# Package 'rKOMICS'

June 30, 2023

**Title** Minicircle Sequence Classes (MSC) Analyses

**Version** 1.3

**Date** 2023-06-28

**Description** This is a analysis toolkit to streamline the analyses of minicircle sequence diversity in population-scale genome projects. rKOMICS is a user-friendly R package that has simple installation requirements and that is applicable to all 27 trypanosomatid genera. Once minicircle sequence alignments are generated, rKOMICS allows to examine, summarize and visualize minicircle sequence diversity within and between samples through the analyses of minicircle sequence clusters. We showcase the functionalities of the (r)KOMICS tool suite using a whole-genome sequencing dataset from a recently published study on the history of diversification of the Leishmania braziliensis species complex in Peru. Analyses of population diversity and structure highlighted differences in minicircle sequence richness and composition between Leishmania subspecies, and between subpopulations within subspecies. The rKOMICS package establishes a critical framework to manipulate, explore and extract biologically relevant information from mitochondrial minicircle assemblies in tens to hundreds of samples simultaneously and efficiently. This should facilitate research that aims to develop new molecular markers for identifying species-specific minicircles, or to study the ancestry of parasites for complementary insights into their evolutionary history. ***** !! WARNING: this package relies on dependencies from Bioconductor. For Mac users, this can generate errors when installing rKOMICS. Install Bioconductor and ComplexHeatmap at advance: install.packages(``BiocManager''); BiocManager::install(``ComplexHeatmap'') *****.

**License** GPL

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**biocViews**

**Imports** ggplot2, ape, circlize, ComplexHeatmap, reshape2, utils, stats, dplyr, factoextra, FactoMineR, ggpubr, magrittr, stringr

**Suggests** viridis

**Depends** R (>= 2.10)

**NeedsCompilation** no

**Author** Frederik Van den Broeck [aut],
Manon Geerts [aut, cre]

**Maintainer** Manon Geerts <geertsmanon@gmail.com>

**Repository** CRAN

**Date/Publication** 2023-06-29 22:40:03 UTC

# R topics documented:



---

`exData` *Example dataset*

---

## Description

A dataset containing example inputs.

## Usage

```
exData
```

## Format

A list with seven objects

**strain** a character vector containing the strain names.

**subspecies** a factor specifying to which subspecie the strains belong to.

**HCN** a numerical vector with their corresponding median genome wide coverage.

**fastafiles** a character vector containing the file names of the minicircle sequences in fasta format.

**ucs** a character vector containing the file names of the cluster information in uc format.

**mapstats** a character vector containing the files names of the mapping statistics in text format.

**depthstats** a character vector containing the files names of the depth statistics in text format.

---

`matrices` *Example cluster matrices*

---

## Description

A list containing 15 example cluster matrices with percent identities of 80, 85 and 88:100.

## Usage

```
matrices
```

## Format

a list of different cluster matrices.

---

`msc.depth` *Check the read depth of assembled minicircles*

---

## Description

The msc.depth function allows you to analyze the read depth of assembled minicircles using depth statistics generated by KOMICS. These statistics provide information such as the average, median, minimum, and maximum read depth per site for each minicircle contig. By standardizing the median read depths per minicircle contig to the median genome-wide read depths, you can estimate minicircle copy numbers.

## Usage

```
msc.depth(depthstats, groups, HCN = NULL)
```

## Arguments

| | |
|---|---|
| `depthstats` | character vector containing the file names of the depth statistics generated by KOMICS. Each file should be in the format "sample_name.depthstats.txt" (e.g., sampleA.depthstats.txt, sampleB.depthstats.txt,...). |
| `groups` | a vector specifying the groups (e.g., species) to which the samples belong. |
| `HCN` | an optional numeric vector containing haploid copy numbers of the corresponding samples. This argument is set to null by default. |

## Value

| | |
|---|---|
| all | a table that merges the depth statistics of all samples. The table includes the average, median, minimum, and maximum per site read depth. |
| plots | a plot per sample that visualizes the median read depth distribution. |
| medianRD | a graph summarizing the median read depth distribution of all samples. |
| CN | a graph summarizing the copy number (if HCN is not null) of all samples. |

## Examples

```
require(ggpubr)
data(exData)

### run function

depth <- msc.depth(depthstats = system.file("extdata",
                  exData$depthstats, package = "rKOMICS"), groups = exData$species,
                  HCN = exData$medGWD/2)

### visualize results
hist(depth$all[,"MEDIAN.DEPTH"], breaks=100,
    main="Global median depth distribution",xlab = (''))

### alter plot
annotate_figure(depth$plots$CUM29A1, fig.lab = "CUM29A1",
               fig.lab.pos = "bottom.right", fig.lab.face = 'italic')
```

---

# msc.heatmap *Visualization of cluster matrices*

---

## Description

The msc.heatmap function generates a heatmap that summarizes the presence or absence of Minicircle Sequence Classes (MCSs) between groups of samples. It takes an input cluster matrix, generated using the msc.matrix function, and visualizes the clustering patterns of MCSs.

## Usage

```
msc.heatmap(clustmatrix, samples, groups)
```

## Arguments

| | |
|---|---|
| clustmatrix | a cluster matrix generated with the msc.matrix function. This matrix represents the presence or absence of MCSs in each sample. |
| samples | a vector containing the sample names. |
| groups | a vector specifying the groups (e.g., species) to which the samples belong. |

## Value

a heatmap that visualizes the clustering patterns of MCSs between sample groups. The heatmap provides an overview of the presence or absence of MCSs and helps identify shared or distinct MCS patterns among the groups.

## Examples

```
data(exData)
data(matrices)

### run function
msc.heatmap(matrices[["id80"]], groups = exData$species,
            samples = exData$samples )

### run function on every cluster matrix with subset of samples
### you will be asked to confirm
table(exData$species)
hybrid <- which(exData$species=="hybrid")
# msc.heatmap(matrices[["id97"]], groups = exData$species[hybrid],
#             samples = exData$samples[hybrid])
```

---

# msc.length *Length of minicircles*

---

## Description

The msc.length function allows you to check the length of minicircle sequences based on a single FASTA file. This function helps determine the size distribution of minicircle sequences.

## Usage

```
msc.length(file, samples, groups)
```

## Arguments

| | |
|---|---|
| file | the name of the FASTA file that contains all the minicircle sequences. The file should be in the format "all.minicircles.circ.fasta". |
| samples | a character vector containing the sample names. |
| groups | a vector of the same length as the samples, specifying the groups (e.g., subspecies) to which the samples belong. |

## Value

| | |
|---|---|
| length | a numerical vector containing the lengths of the minicircle sequences. Each element corresponds to the length of a specific minicircle sequence. |
| plot | a histogram that visualizes the frequency distribution of minicircle sequence lengths. The histogram provides an overview of the length distribution of the minicircles. |

## Examples

```
require(ggplot2)
require(ggpubr)

### run function
bf <- msc.length(file = system.file("extdata", "all.minicircles.fasta", package="rKOMICS"),
                 samples = exData$samples, groups = exData$subspecies)
af <- msc.length(file = system.file("extdata", "all.minicircles.circ.fasta", package="rKOMICS"),
                 samples = exData$samples, groups = exData$subspecies)

length(which(bf$length<800))
length(which(bf$length>1400))

### visualize results
hist(af$length, breaks=50)

### alter plot
ggarrange(bf$plot + labs(caption = "Before filtering"),
          af$plot + labs(caption = "After filtering"), nrow=2)
```

---

# msc.matrix *Build cluster matrix*

---

## Description

The msc.matrix function reads the output of clustering analyses (UC file) for specified minimum percent identity (MPI) values and organizes the data into a matrix format. This matrix represents the presence or absence of Minicircle Sequence Classes (MSCs) in each sample. The resulting matrix simplifies downstream analyses and visualizations by eliminating the need for manual data manipulation and reformatting.

## Usage

```
msc.matrix(files, samples, groups)
```

### Arguments

| | |
|---|---|
| files | a character vector containing the names of the UC files generated by the VSEARCH tool. Each file represents the output of clustering analysis for a specific minimum percent identity (MPI), such as all.minicircles.circ.id70.uc, all.minicircles.circ.id80.uc, and so on. Please ensure that your file names end with 'idxx.uc' for this function to work properly. |
| samples | a character vector containing the sample names. |
| groups | a vector of the same length as the samples, specifying the groups (e.g., species) to which the samples belong. |

### Value

a a list that contains one cluster matrix per percent identity. Each matrix represents the presence or absence of MSCs in each sample. In the cluster matrix, a value of 0 indicates that the MSC is not present in the sample, while a value higher than 0 indicates that the MSC is found at least once in the sample.

### Examples

```
data(exData)

### run function

matrices <- msc.matrix(files = system.file("extdata", exData$ucs, package="rKOMICS"),
                       samples = exData$samples,
                       groups = exData$species)


### or:
data(matrices)

### show matrix with id 95%
matrices[["id95"]]
rowSums(matrices[["id95"]]) # --> frequency of MSC across all samples
colSums(matrices[["id95"]]) # --> number of MSC per sample
```

---

## msc.pca *Prinicple Component Analysis based on MSC*

---

### Description

The msc.pca function allows you to perform Principle Component Analysis (PCA) to summarize the variation of Minicircle Sequence Classes (MSCs) in all samples or in a subset of samples.

### Usage

```
msc.pca(clustmatrix, samples, groups, n = 20, labels = TRUE, title = NULL)
```

## Arguments

| | |
|---|---|
| clustmatrix | a cluster matrix obtained from the msc.matrix function. The cluster matrix represents the presence or absence of MSCs in each sample, where rows represent MSCs and columns represent samples. |
| samples | a vector containing the names of the samples. This can include all samples or a subset of samples that you want to analyze. |
| groups | a vector specifying the groups (e.g., species) to which the samples belong. |
| n | the number of clusters to select with the highest contribution to PCA. By default, it is set to 20. |
| labels | a logical parameter indicating whether to use labels on the PCA plot or not. If set to TRUE (default), the plot will display sample labels. |
| title | the title of the graph. You can provide a title for the PCA plot if desired. |

## Value

| | |
|---|---|
| plot | a PCA plot that visualizes the clustering of samples based on the presence/absence of MSCs. The plot helps identify clusters and patterns of similarity or dissimilarity between samples. |
| eigenvalues | a barplot showing the percentage of explained variances by each principal component. This plot provides insights into the contribution of each principal component to the overall variation in the data. |
| clustnames | a A list of cluster names with the highest contribution to PCA. This list helps identify the MSC clusters that have the most influence on the PCA results. |

## Examples

```
data(matrices)
data(exData)

### run function with all samples
res.pca <- lapply(matrices, function(x) msc.pca(x, samples = exData$samples,
                  groups = exData$species, n=30, labels=FALSE, title=NULL))

res.pca$id95$eigenvalues
res.pca$id95$plot

### use clusters with highest contribution to visualize in a heatmap
msc.heatmap(matrices[["id95"]][res.pca$id95$clustnames,], samples = exData$samples,
            groups = exData$species)

### run function with a subset of samples
### you will be asked to confirm
table(exData$species)
hybrid <- which(exData$species=="hybrid")
# pca.subset <- msc.pca(clustmatrix = matrices[["id97"]],
#                       samples = exData$samples[hybrid],
#                       groups = exData$species[hybrid], labels = TRUE,
#                       title = "PCA only with hybrids")
```

`msc.quality` *Check the quality of the assembly*

## Description

The msc.quality function allows you to summarize mapping statistics generated by KOMICS to assess the quality of minicircle assembly and mapping. It focuses on examining the frequencies of various mapped read categories, including read frequency, mapped read frequency, high-quality mapped read frequency, and the proportion of (near-)perfect alignments of CSB3-containing reads.

## Usage

```
msc.quality(mapstats, groups)
```

## Arguments

| | |
|---|---|
| `mapstats` | a character vector containing the file names of mapping statistics generated by KOMICS. These files provide information about the mapping and quality of the assembled minicircles. |
| `groups` | a vector specifying the groups (e.g., species) to which the samples belong. |

## Value

| | |
|---|---|
| `all` | a table merging the mapping statistics of all samples. The mapping statistics include the number of mapped reads (MR), mapped reads with high quality (MR_HQ), CSB3-containing mapped reads (MR_CSB3), and CSB3-containing mapped reads with high quality (MR_CSB3_HQ). |
| `proportions` | a list of tables containing the proportions of the mentioned mapping statistics. It provides insights into the relative frequencies and proportions of each category. |
| `plots` | barplots visualizing the mapping statistics and proportions. These plots help visualize and compare the mapping quality across samples. |

## Examples

```
data(exData)

### run function
map <- msc.quality(mapstats = system.file("extdata", exData$mapstats, package = "rKOMICS"),
                   exData$species)

lapply(map$proportions, mean)$MR_HQ
lapply(map$proportions, mean)$MR_CSB3_HQ

### visualize results
barplot(map$proportions$MR)
```

`msc.richness` *Minicircle Sequence Cluster richness*

## Description

The msc.richness function calculates the measure of minicircle richness per sample by estimating the number of Minicircle Sequence Classes (MSCs) in each sample. It takes into account different minimum percent identities (MPIs) and returns a table of richness estimates per sample and per MPI. Additionally, it generates a boxplot that illustrates the minicircle richness across samples based on the estimated MSCs over a range of MPIs.

## Usage

```
msc.richness(clustmatrices, samples, groups)
```

## Arguments

| | |
|---|---|
| `clustmatrices` | a list of cluster matrices obtained from the msc.matrix function. Each cluster matrix represents the presence or absence of MSCs in each sample. |
| `samples` | a vector containing the names of the samples. |
| `groups` | a vector specifying the group (e.g., species) to which each sample belongs. |

## Value

| | |
|---|---|
| `table` | a table summarizing the number of MSCs per sample at different percent identities. The table provides an overview of the estimated minicircle richness in each sample across the MPIs. |
| `plot` | a boxplot visualizing the minicircle richness across samples. The boxplot represents the distribution of richness estimates in each sample over the range of MPIs considered. |

## Examples

```
require(ggplot2)
data(matrices)
data(exData)

#### run function
richness <- msc.richness(matrices, samples = exData$samples, groups = exData$species)

apply(richness$table[which(richness$table$group=="L. peruviana"),-(1:2)], 2, mean)
apply(richness$table[which(richness$table$group=="L. braziliensis"),-(1:2)], 2, mean)
apply(richness$table[which(richness$table$group=="hybrid"),-(1:2)], 2, mean)

#### visualize results
barplot(richness$table[,"id93"], names.arg = richness$table[,1],
        las=2, cex.names=0.4, main="N of MSC at id 93")
```

```
#### adjust plot
richness$plot + ggtitle("MSC richness across % id") +
                theme(axis.text.x = element_text(angle=45, hjust=1))

### show results of subset
table(exData$species)
hybrid <- which(exData$species=="hybrid")
# richness.subset <- msc.richness(matrices, samples = exData$samples[hybrid],
#                                 groups = exData$species[hybrid])
```

---

`msc.seqs` *Retrieve sequences*

---

## Description

The msc.seqs function retrieves the DNA sequence of a Minicircle Sequence Classes (MSC) together with all its hit sequences from a FASTA file and a corresponding UC file. This function is useful for extracting and analyzing specific MSCs and their associated hit sequences.

## Usage

```
msc.seqs(fastafile, ucfile, clustnumbers, writeDNA = TRUE)
```

## Arguments

| | |
|---|---|
| `fastafile` | the name of the FASTA file containing all minicircle sequences. |
| `ucfile` | the name of the UC file. |
| `clustnumbers` | a character vector containing the cluster numbers (in the format "C0", "C1", etc.) of the MSCs for which you want to retrieve the sequences. These cluster numbers specify the MSCs and their associated hit sequences that need to be extracted from the FASTA file and UC file. |
| `writeDNA` | a logical parameter that is set to TRUE by default. When set to TRUE, this parameter will write the extracted sequences into separate FASTA files in the current directory. |

## Value

a table that summarizes the number of hit sequences found in each MSC, the MSC names, and the samples where the MSCs are present. This table provides an overview of the extracted sequences and their distribution across samples.

one FASTA file per MSC with all its hit sequences. These FASTA files can be further used for downstream analyses or sequence comparisons.

## Examples

```
data(exData)

### select a subset of MSC
Lpe <- which(exData$species == "L. peruviana")
specific <- msc.subset(matrices[[7]], subset = Lpe)

### run function
seq <- msc.seqs(fastafile = system.file("extdata", "all.minicircles.circ.fasta", package="rKOMICS"),
                ucfile = system.file("extdata", exData$ucs, package="rKOMICS")[7],
                clustnumbers = specific$clustnumbers, writeDNA = FALSE)
```

---

# msc.similarity *Minicircle Sequence Classes similarity*

---

## Description

The function msc.similarity returns a measure of minicircle sequence composition within and between groups of samples. Specifically, it estimates the absolute and relative number of Minicircle Sequence Classes (MSCs) that are unique to each group or shared between two or more groups. The function returns tables and barplots that summarize the number of unique or shared MSCs for each minimum percent identity (MPI) separately or combined over all MPIs.

## Usage

```
msc.similarity(clustmatrices, samples, groups)
```

## Arguments

| | |
|---|---|
| clustmatrices | a list of cluster matrices. |
| samples | a vector containing the names of the samples. This can include all samples or it can be a subset. |
| groups | a vector, of equal length as samples, specifying to which group (e.g. species) the samples belong to. |

## Value

| | |
|---|---|
| absfreq | a list per percent identity containing absolute frequency values of shared and unique MSCs. |
| absfreq.plot | a list of barplots visualizing previous results. |
| relfreq | a list per percent identity containing relative frequency values of shared and unique MSCs. |
| relfreq.plot | one barplot visualizing previous results. |

## Examples

```
require(viridis)
data(matrices)
data(exData)

### run function
sim <- msc.similarity(matrices, samples = exData$samples,
                      groups = exData$species)

### visualize results (absolute frequencies)
barplot(sim$absfreq$id93)

### adjust plot (relative frequencies)
sim$relfreq.plot + scale_fill_viridis(discrete = TRUE)

sim$relfreq$id97["2"]*100
sim$relfreq$id97["3"]*100

### reduce number of groups
groups <- exData$species
levels(groups)[levels(groups)!='hybrid'] <- "non-hybrid"
sim.red <- msc.similarity(matrices, samples = exData$samples, groups = groups)
sim.red$relfreq.plot + scale_fill_viridis(discrete = TRUE)
```

---

# msc.subset *Specific Minicircle Sequence Classes*

---

## Description

The msc.subset function allows you to identify specific Minicircle Sequence Classes (MSCs) for a subset of samples based on the output of the msc.matrix function. It helps in extracting and analyzing MSCs that are present in a particular subset of samples.

## Usage

```
msc.subset(clustmatrix, subset)
```

## Arguments

| | |
|---|---|
| clustmatrix | a cluster matrix obtained from the msc.matrix function. The cluster matrix represents the presence or absence of MSCs in each sample. |
| subset | a numerical vector indicating the subset of samples for which you want to identify specific MSCs. The values in the subset vector correspond to the indices of the samples to be included. |

## Value

| | |
|---|---|
| clustnumbers | a vector containing the names of the specific MSCs present in the subset of samples. These are the MSCs that are found in the indicated subset. |
| freq | frequency values indicating the occurrence of the specific MSCs in the subset of samples. These values represent the number of times each MSC appears in the subset. |
| matrix | a subset of the cluster matrix containing only the specific MSCs found in the subset of samples. For samples not included in the subset, the values in the matrix should have the value 0, indicating the absence of the MSC. |
| sum | the total number of MSC found in the indicated subset of samples. |

## Examples

```
data(matrices)
data(exData)

### selecting a group of samples e.g. all L. peruviana species
Lpe <- which(exData$species == "L. peruviana")

### run function
specific <- msc.subset(matrices[["id97"]], subset = Lpe)

### visualize results (check if it is indeed specific)
heatmap(specific$matrix) # or:
msc.heatmap(specific$matrix, samples = exData$samples, groups = exData$species)

### find specific MSC with highest frequency
which.max(specific$freq)
```

---

# msc.uc *Cluster Analyses*

---

## Description

The function msc.uc reads the output of the clustering analyses (UC file) for each specified minimum percent identity (MPI) into a single list, which will be analyzed automatically to calculate and visualize, per MPI, the number of minicircle sequence classes (MSCs), the proportion of perfect alignments (i.e. alignments without any insertion/deletion, but allowing point mutations) and the number of alignment gaps. Gaps are defined by i) the number of insertions/deletions and ii) the length in base pairs of each individual insertion/deletion. It also issues a warning when large gaps (>500 bp) are found, which points the user to anomalous alignments due to e.g. artificial dimers introduced by the assembly process. This allows the user to make an informed decision about the MPI (or MPI's) that best captures minicircle sequence richness within a (group of) sample(s) while minimizing the number and length of alignment gaps.

## Usage

```
msc.uc(files)
```

## Arguments

| | |
|---|---|
| files | a character vector that includes the file names of UC files (produced by USEARCH or VSEARCH), such as all.minicircles.circ.id70.uc, all.minicircles.circ.id80.uc, and so on. Please ensure that your file names end with 'idxx.uc' for this function to work properly. |

## Value

| | |
|---|---|
| MSCs | a numerical vector containing the number of MSC per MPI. |
| perfect aligments | a numerical vector containing the proportions of perfect alignments per MPI. |
| insertions | a list showing the insertion lengths per MPI. Each element in the list corresponds to a specific MPI, and it provides the lengths of identified insertions. |
| deletions | a list showing the deletions lengths per MPI. Each element in the list corresponds to a specific MPI, and it provides the lengths of identified deletions. |
| insertions summary | a table showing the length and the number of insertions across different MPIs. |
| deletions summary | a table showing the length and the number of deletions across different MPIs. |
| plots | various plots showing previous results. |

## Examples

```
data(exData)

### run function

ucs <- msc.uc(files = system.file("extdata", exData$ucs, package="rKOMICS"))

ucs$MSCs["100"]
ucs$MSCs["97"]

### results
ucs$plots
```

`preprocess` *Filtering of minicircle sequences*

## Description

The preprocess function is used to filter minicircle sequences based on sequence length and circularization success. When minicircle sequences are assembled with KOMICS, individual fasta files are generated for each sample. This function allows you to filter these sequences based on their length and whether they are circularized or not. The filtered sequences are then written into individual FASTA files in the current working directory.

## Usage

```
preprocess(files, groups, circ = TRUE, min = 500, max = 1500, writeDNA = TRUE)
```

## Arguments

| | |
|---|---|
| `files` | a character vector containing the names of the fasta files. Each file corresponds to the minicircle sequences of a specific sample. The file names should be in the format sampleA.minicircles.fasta, sampleB.minicircles.fasta, and so on (output of KOMICS). |
| `groups` | a factor specifying the group (e.g., species) to which each sample belongs. It should have the same length as the list of files, indicating the group assignment for each sample. |
| `circ` | a logical parameter that determines whether non-circularized minicircle sequences should be included or excluded from the filtering process. By default, non-circularized sequences are excluded (circ = TRUE). If you are interested in including non-circularized sequences, you can set the parameter to FALSE. |
| `min` | the minimum length threshold for filtering minicircle sequences. Sequences with a length below this threshold will be excluded. The default value is set to 500. |
| `max` | the maximum length threshold for filtering minicircle sequences. Sequences with a length above this threshold will be excluded. The default value is set to 1500. |
| `writeDNA` | a logical parameter that determines whether the filtered minicircle sequences should be written in FASTA format to the current working directory. By default, the filtered sequences are written (writeDNA = TRUE). If you are only interested in other output values like plots and summary, you can set this parameter to FALSE. |

## Value

| | |
|---|---|
| `samples` | the sample names based on the input files. |
| `N_MC` | a table containing the sample names, the corresponding group assignment, and the number of minicircle sequences (N_MC) before and after filtering. |

| | |
|---|---|
| plot | a barplot visualizing the number of minicircle sequences per sample before and after filtering. |
| summary | the total number of minicircle sequences before and after filtering. |

## Examples

```
require(ggplot2)
data(exData)

### setwd("")

### run function
table(exData$species)
pre <- preprocess(files = system.file("extdata", exData$fastafiles, package="rKOMICS"),
                  groups = exData$species,
                  circ = TRUE, min = 500, max = 1200, writeDNA = FALSE)

pre$summary

### visualize results
barplot(pre$N_MC[,"beforefiltering"],
        names.arg = pre$N_MC[,1], las=2, cex.names=0.4)

### alter plot
pre$plot + labs(caption = paste0('N of MC sequences before and after filtering, ', Sys.Date()))
```

---

# read.uc *Read in uc files*

---

## Description

The read.uc function is used to read the output of clustering analyses from a UC file. The function stores the information in two tables: one for hit records (H) and one for cluster records (C).

## Usage

```
read.uc(file)
```

## Arguments

| | |
|---|---|
| file | the name of the UC file that contains the clustering analysis results. The file should be specified with its full name, including the extension (e.g., all.minicircles.circ.id70.uc). |

## Value

| | |
|---|---|
| hits | a table containing all the hit records from the UC file. Each row of the table represents a hit record, providing information about the alignment between a query sequence and a target sequence. |

| | |
|---|---|
| clusters | a table containing all the cluster records from the UC file. Each row of the table represents a cluster record, providing information about the clustering of sequences into clusters. |
| clustnumbers | a vector containing the cluster numbers (0-based). Each element of the vector represents a cluster identified in the clustering analysis. |

---

rKOMICS *Minicircle Sequence Classes (MSC) Analyses*

---

## Description

This is a analysis toolkit to streamline the analyses of minicircle sequence diversity in population-scale genome projects. rKOMICS is a user-friendly R package that has simple installation requirements and that is applicable to all 27 trypanosomatid genera. Once minicircle sequence alignments are generated, rKOMICS allows to examine, summarize and visualize minicircle sequence diversity within and between samples through the analyses of minicircle sequence clusters. We showcase the functionalities of the (r)KOMICS tool suite using a whole-genome sequencing dataset from a recently published study on the history of diversification of the Leishmania braziliensis species complex in Peru. Analyses of population diversity and structure highlighted differences in minicircle sequence richness and composition between Leishmania subspecies, and between subpopulations within subspecies. The rKOMICS package establishes a critical framework to manipulate, explore and extract biologically relevant information from mitochondrial minicircle assemblies in tens to hundreds of samples simultaneously and efficiently. This should facilitate research that aims to develop new molecular markers for identifying species-specific minicircles, or to study the ancestry of parasites for complementary insights into their evolutionary history. ***** !! WARNING: this package relies on dependencies from Bioconductor. For Mac users, this can generate errors when installing rKOMICS. Install Bioconductor and ComplexHeatmap at advance: install.packages("BiocManager"); BiocManager::install("ComplexHeatmap") *****.

## Details

The DESCRIPTION file: This package was not yet installed at build time.

Index of help topics:

```
exData                  Example dataset
matrices                Example cluster matrices
msc.depth               Check the read depth of assembled minicircles
msc.heatmap             Visualization of cluster matrices
msc.length              Length of minicircles
msc.matrix              Build cluster matrix
msc.pca                 Prinicple Component Analysis based on MSC
msc.quality             Check the quality of the assembly
msc.richness            Minicircle Sequence Cluster richness
msc.seqs                Retrieve sequences
msc.similarity          Minicircle Sequence Classes similarity
```

| | |
|---|---|
| msc.subset | Specific Minicircle Sequence Classes |
| msc.uc | Cluster Analyses |
| preprocess | Filtering of minicircle sequences |
| rKOMICS | Minicircle Sequence Classes (MSC) Analyses |
| read.uc | Read in uc files |

## Author(s)

Frederik Van den Broeck <frederik.vandenbroeck@kuleuven.be>

Manon Geerts <mgeerts@itg.be>

## References

Geerts, M., Schnaufer, A. & Van den Broeck, F. rKOMICS: an R package for processing mitochondrial minicircle assemblies in population-scale genome projects. BMC Bioinformatics 22, 468 (2021). doi: 10.1186/s12859021043841

Van den Broeck F, Savill NJ, Imamura H, Sanders M, Maes I, Cooper S, et al. Ecological divergence and hybridization of Neotropical Leishmania parasites. Proc Natl Acad Sci U S A. 2020;117. doi: 10.1073/pnas.1920136117.

## See Also

Github: https://frebio.github.io/

komics-suite: https://frebio.github.io/komics/

# Index