# Package ‘RepertoiR’

January 20, 2025

**Title** Repertoire Graphical Visualization

**Version** 0.0.1

**Description** Visualization platform for T cell receptor repertoire analysis output results. It includes comparison of sequence frequency among samples, network of similar sequences and convergent recombination source between species. Currently repertoire analysis is in early stage of development and requires new approaches for repertoire data examination and assessment as we intend to develop. No publication is available yet (will be available in the near future), Efroni (2021) <https:>.

**License** MIT + file LICENSE

**URL** https://github.com/systemsbiomed/RepertoiR

**BugReports** https://github.com/systemsbiomed/RepertoiR/issues

**Imports** circlize, grDevices, igraph, reshape2, stringdist, stringi, stringr

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**NeedsCompilation** no

**Author** Ido Hasson [aut, cre] (<https://orcid.org/0000-0001-8382-5638>),
Sol Efroni [aut],
Hagit Philip [aut],
Alona Zilberberg [aut]

**Maintainer** Ido Hasson <idoh@systemsbiomed.org>

**Repository** CRAN

**Date/Publication** 2021-10-25 07:00:21 UTC

# Contents

cr_source . . . . . 2
cr_source.default . . . . . 3
network . . . . . 4
network.default . . . . . 5
sunflower . . . . . 6
sunflower.default . . . . . 6

**Index** **8**

---

`cr_source` *Visualized for CR Sources*

---

## Description

Visualization of Two clones for their convergent recombination (CR) sources. Each sequence (NT) is represented as a colored bar (red for A, yellow for G, blue for T and green for C) linked to its translated amino acid sequence by a colored line, red for the first clone and blue for the second.

## Usage

```
cr_source(clone1, clone2, ...)
```

## Arguments

| | |
|---|---|
| `clone1` | First vector of sequences, string-length is the same for each nucleotide sequence ('A', 'G', 'T', 'C'). |
| `clone2` | Second vector of sequences, same string-length as for the first vector. |
| `...` | Any other arguments. |

## Value

No return value.

## Examples

```
nt <- c("A", "G", "C", "T")
seq_len <- 15
seq_n <- c(12, 7)

# Create data
c1 <- replicate(seq_n[1],
                paste(sample(nt, seq_len, replace = TRUE), collapse = ''))
c2 <- replicate(seq_n[2],
                paste(sample(nt, seq_len, replace = TRUE), collapse = ''))

cr_source(c1, c2)
```

## cr_source.default *Visualized for CR Sources*

### Description

Visualization of Two clones for their convergent recombination (CR) sources. Each sequence (NT) is represented as a colored bar (red for A, yellow for G, blue for T and green for C) linked to its translated amino acid sequence by a colored line, red for the first clone and blue for the second.

### Usage

```
## Default S3 method:
cr_source(clone1, clone2, ...)
```

### Arguments

| | |
|---|---|
| clone1 | First vector of sequences, string-length is the same for each nucleotide sequence ('A', 'G', 'T', 'C'). |
| clone2 | Second vector of sequences, same string-length as for the first vector. |
| ... | Any other arguments. |

### Value

No return value.

### Examples

```
nt <- c("A", "G", "C", "T")
seq_len <- 15
seq_n <- c(12, 7)

# Create data
c1 <- replicate(seq_n[1],
                paste(sample(nt, seq_len, replace = TRUE), collapse = ''))
c2 <- replicate(seq_n[2],
                paste(sample(nt, seq_len, replace = TRUE), collapse = ''))

cr_source(c1, c2)
```

network *Sequences distance network*

## Description

Computes pairwise string distances among repertoire's sequences and visualize similar pairs as connected nodes, each sized by its frequency.

## Usage

```
network(dataset, by, nrow, method, ...)
```

## Arguments

| | |
|---|---|
| dataset | A matrix or a data frame includes row names which are used as the compared sequences. Data set's numeric values determine node-size. |
| by | Index of column to set its values as node-size. first column is default (1). |
| nrow | Number of nodes to display. Default is 1000 nodes. |
| method | stringdist method to perform for distance dissimilarity calculation: "osa", "lv", "dl", "hamming", "lcs", "qgram", "cosine", "jaccard", "jw", "soundex". Default is Levenshtein distance ("lv"). |
| ... | Any additional arguments needed by the specialized methods. |

## Value

No return value.

## Examples

```
aa <- c(
  "G", "A", "V", "L", "I", "P", "F", "Y", "W", "S",
  "T", "N", "Q", "C", "M", "D", "E", "H", "K", "R"
)
data <- matrix(rexp(1 / 2, n = 1000), ncol = 4)
cons <- sample(aa, 10)
aavec <- c()

while (length(aavec) < nrow(data)) {
  aaseq <- cons
  index <- sample(length(aaseq), sample(length(aaseq) / 3, 1))
  aaseq[index] <- sample(aa, length(index), replace = TRUE)
  aaseq <- paste0(aaseq, collapse = "")
  aavec <- unique(append(aavec, aaseq))
}

rownames(data) <- aavec
colnames(data) <- LETTERS[1:ncol(data)]

network(data, by = 3, nrow = 100)
```

network.default *Sequences distance network*

## Description

Computes pairwise string distances among repertoire's sequences and visualize similar pairs as connected nodes, each sized by its frequency.

## Usage

```
## Default S3 method:
network(dataset, by = 1, nrow = 1000, method = "lv", ...)
```

## Arguments

| | |
|---|---|
| dataset | A matrix or a data frame includes row names which are used as the compared sequences. Data set's numeric values determine node-size. |
| by | Index of column to set its values as node-size. first column is default (1). |
| nrow | Number of nodes to display. Default is 1000 nodes. |
| method | stringdist method to perform for distance dissimilarity calculation: "osa", "lv", "dl", "hamming", "lcs", "qgram", "cosine", "jaccard", "jw", "soundex". Default is Levenshtein distance ("lv"). |
| ... | Any additional arguments needed by the specialized methods. |

## Value

No return value.

## Examples

```
aa <- c(
  "G", "A", "V", "L", "I", "P", "F", "Y", "W", "S",
  "T", "N", "Q", "C", "M", "D", "E", "H", "K", "R"
)
data <- matrix(rexp(1 / 2, n = 1000), ncol = 4)
cons <- sample(aa, 10)
aavec <- c()

while (length(aavec) < nrow(data)) {
  aaseq <- cons
  index <- sample(length(aaseq), sample(length(aaseq) / 3, 1))
  aaseq[index] <- sample(aa, length(index), replace = TRUE)
  aaseq <- paste0(aaseq, collapse = "")
  aavec <- unique(append(aavec, aaseq))
}

rownames(data) <- aavec
colnames(data) <- LETTERS[1:ncol(data)]
```

```
network(data)
```

---

`sunflower` *Sunflower repertoire graph*

---

## Description

Sequence frequency visualization among samples, displayed as rings of nodes inside each other.

## Usage

```
sunflower(dataset, ...)
```

## Arguments

| | |
|---|---|
| `dataset` | Input object: a matrix or a data frame.<br>First column is located as the outer ring, the second is right after and so on to the last column as the inmost ring. Cell's numeric value determines node size. |
| `...` | Any other arguments. |

## Value

No return value.

## Examples

```
data <- matrix(rexp(400,1/4), ncol = 4)
sunflower(data)
```

---

`sunflower.default` *Default graph*

---

## Description

Default visualization of sequence frequencies among samples as rings inside each other.

## Usage

```
## Default S3 method:
sunflower(dataset, ...)
```

### Arguments

| | |
|---|---|
| `dataset` | Input object: a matrix or a data frame.<br>First column is located as the outer ring, the second is right after and so on to the last column as the inmost ring. Cell's numeric value determines node size. |
| `...` | Any other arguments. |

### Value

No return value.

### Examples

```
data <- matrix(rexp(400,1/4), ncol = 4)
sunflower(data)
```

# Index

cr_source, 2
cr_source.default, 3

network, 4
network.default, 5

sunflower, 6
sunflower.default, 6