

Package ‘kaos’

October 13, 2022

Type Package

Title Encoding of Sequences Based on Frequency Matrix Chaos Game Representation

Version 0.1.2

Author Dominik Eger and Hannah Franziska Löchel

Maintainer Hannah Franziska Löchel <loechelh@mathematik.uni-marburg.de>

Description Sequences encoding by using the chaos game representation.
Löchel et al. (2019) <[doi:10.1093/bioinformatics/btz493](https://doi.org/10.1093/bioinformatics/btz493)>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports ggplot2, reshape2

NeedsCompilation no

Repository CRAN

Date/Publication 2019-09-27 08:50:07 UTC

R topics documented:

cgr	2
cgr.plot	3
cgr.res	4
HIV	4
vectorize	5
Index	6

cgr *Chaos Game Representation Object*

Description

Chaos Game Representation Object

Usage

```
cgr(data, seq.base = row.names(table(data)), sf = F, res = 100)
```

Arguments

data	Input as list/vector of characters from fasta file or similar
seq.base	By default the included unique elements in data will be used in alphabetical order. It is also possible to define the alphabet explicitly. Predefined alphabets can be used as well: <ul style="list-style-type: none"> • "digits": numbers from 0 to 9 • "AMINO": alphabetical order of the amino acids in capital letters • "amino": alphabetical order of the amino acids in lowercase letters • "DNA": the four bases of DNA ("A","G","T","C") in capital letters • "dna": the four bases of DNA ("a","g","t","c") in lowercase letters • "LETTERS": The alphabetical order of capital letters from A to Z • "letters": The alphabetical order of lowercase Letters from a to z
sf	By default, the scaling factor for fractal polygons is used; the scaling factor can also explicitly set to values between 0 and 1.
res	resolution of the frequency matrix

Details

This function produces a chaos game representation (CGR) object from a sequence (data)

Value

CGR object as list of:

- matrix: frequency matrix with given resolution
- x: x-coordinates for the CGR
- y: y-coordinates for the CGR
- sf: applied scaling factor for the CGR
- res: applied resolution to calculate the FCGR
- base.seq: chars or letters to build the edges of the CGR

Examples

```
###HIV data
data("HIV")

### encoding the sequence
HIV.cgr = cgr(HIV, res = 100)

###plot the sequence
cgr.plot(HIV.cgr, mode = "points")

###plot the FCGR
cgr.plot(HIV.cgr, mode = "matrix")

###change the resolution of matrix from 100x100 to 200x200
cgr.res(HIV.cgr, 200)

### get the FCGR encoded vector
vectorize(HIV.cgr)
```

cgr.plot

Plot from a CGR object

Description

Plot from a CGR object

Usage

```
cgr.plot(data, mode, corners = F, labels = F)
```

Arguments

data	CGR object
mode	character string <ul style="list-style-type: none">"points": CGR plot"matrix": FCGR plot
corners	if true, the corners are added as red dots
labels	if true, the symbol associated with the corner is added

Details

This function plots the chaos game representation as points or as frequency matrix representation

`cgr.res`*Produces a new matrix with new resolution*

Description

Produces a new matrix with new resolution

Usage

```
cgr.res(data, res)
```

Arguments

<code>data</code>	CGR object
<code>res</code>	new resolution

Details

This function adjust the resolution of the FCGR of a CGR object.

- `matrix`: frequency matrix with new given resolution
- `x`: x-coordinates for the CGR
- `y`: y-coordinates for the CGR
- `sf`: applied scaling factor for the CGR
- `res`: applied resolution to calculate the FCGR
- `base.seq`: chars or letters to build the edges of the CGR

`HIV`*Complete HIV 1 genome*

Description

Complete HIV 1 genome

Format

A vector containing the genome of HIV-1 as single characters

Source

NC_001802.1 Human immunodeficiency virus 1, complete genome

vectorize	<i>Vectorizes the FCGR</i>
-----------	----------------------------

Description

Vectorizes the FCGR

Usage

```
vectorize(data)
```

Arguments

data CGR object from cgr.R

Details

This function returns the FCGR as vector.

Value

Vector with FCGR-encoding

Index

cgr, [2](#)
cgr.plot, [3](#)
cgr.res, [4](#)

HIV, [4](#)

vectorize, [5](#)