Package 'seqGen'

June 18, 2018

Type Package

Title Generates random DNA-String with introduced shifts.

Version 1.0

Date 2018-06-11

Author Robert Deibel

Maintainer Robert Deibel <deibel.robert@googlemail.com>

Description This is a package for creation of random DNA-
strings with shifts of different lengths and occurences. Strings are output in IUPAC notation.

License The Unlicense

R topics documented:

---

seqGen-package   *Generates random DNA-String with introduced shifts.*

Description

This is a package for creation of random DNA-strings with shifts of different lengths and occurences. Strings are output in IUPAC notation.

Details

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

Index: This package was not yet installed at build time.
Use generateRandBioString() to generate a random DNA string for a specified length with introduced shifts. Output is in IUPAC notation.

1

Author(s)

Robert Deibel

Maintainer: Robert Deibel <deibel.robert@googlemail.com>

Examples

```
generateRandBioString(150)
```

---

combineSeq	*Combine Sequences*

Description

Given two sequences as vectors, combine those to IUPAC notation.

Usage

```
combineSeq(seq1, seq2)
```

Arguments

seq1	A DNA sequence as a character vector.
seq2	A DNA sequence as a character vector.
	The combined sequence in IUPAC notation.
	Robert Deibel
	a <- c("A","T","T","C","G") b <- c("A","C","A","G","C") comb <- combineSeq(a,b)
	datagen

---

generateRandBioString	*Generate Random Bio-String*

Description

Generate a random DNA string with introduced shifts at a given place in IUPAC notation.

Usage

```
generateRandBioString(length, dist = runif(1, 0, 4), exact = TRUE, occurrence = "m")
```

Arguments

length	Length of the string (before shift). One of `c(150,300,700)`.
dist	The distribution used to generate the sequence.
exact	Whether the shift is to be introduced exactly or with some randomisation.
occurrence	Occurrence of the shift. One of `c("m","b","q1","q3")` indicating the middle, beginning, first quater and third quater of the sequence.

Value

A string in IUPAC notation.

Examples

```
biostring <- generateRandBioString(150)
```

---

generateSeq   *Generate DNA sequence*

Description

Generate a DNA sequence of a specified length.

Usage

```
generateSeq(length, dist = runif(1, 0, 4))
```

Arguments

length   One of `c(150,300,700)`. Specifies the length of the generated sequence.

dist   The distribution by which the sequence is generated.

Value

A randomly generated DNA sequence .

Author(s)

Robert Deibel

Examples

```
seq <- generateSeq(150)
```

---

introduceShift   *Shift sequence*

Description

Shifts a given sequence at specified occurrence.

Usage

```
introduceShift(seqence, exact = TRUE, occurrence = "m")
```

Arguments

seqence	A vector of character values representing a DNA sequence.
exact	`logical` indicating whether the occurrence is to be at the exact place or if some randomisation should take place.
occurrence	One of `c("m","b","q1","q3")` indicating the middle beginning first quater and third quater of the sequence. Where the shift is to occur.

Value

The combined sequence in IUPAC notation with introduced shift.

Author(s)

Robert Deibel

Examples

```
sequence <- character(150)
sequence[] <- "A"
shiftedSeq <- introduceShift(sequence)
print(shiftedSeq)
shiftedSeq <- introduceShift(sequence,FALSE)
print(shiftedSeq)
shiftedSeq <- introduceShift(sequence,occurrence="b")
print(shiftedSeq)
```

---

iupacNotation   *IUPAC Notation*

Description

Returns the IUPAC notation of a vector of two characters.

Usage

```
iupacNotation(chars)
```

Arguments

chars	A vector of two characters of "A", "C", "G" or "T".

Value

A character representing the IUPAC notation of one or two characters,

Author(s)

Robert Deibel

Examples

```
chars <- c("A","G")
iupac <- iupacNotation(chars)
print(iupac)
```

---

randomBase	*Randomized Base*

Description

Returns a random Base with specified distribution. If not further stated the uniform distribution is used.

Usage

```
randomBase(dist = runif(1, 0, 4))
```

Arguments

dist
The distribution the base is derived from. Lower bound should be 0 and upper bound should always be 4.

Value

A character of either "A", "C", "G" or "T".

Author(s)

Robert Deibel

Examples

```
base <- randomBase()
print(base)
```

Index