

Package ‘SimRAD’

October 12, 2022

Type Package

Title Simulations to Predict the Number of RAD and GBS Loci

Version 0.96

Date 2015-11-04

Maintainer Olivier Lepais <olepais@st-pee.inra.fr>

Depends Biostrings, ShortRead, zlibbioc

Imports stats, graphics

Suggests seqinr

Description Provides a number of functions to simulate restriction enzyme digestion, library construction and fragments size selection to predict the number of loci expected from most of the Restriction site Associated DNA (RAD) and Genotyping By Sequencing (GBS) approaches. SimRAD estimates the number of loci expected from a particular genome depending on the protocol type and parameters allowing to assess feasibility, multiplexing capacity and the amount of sequencing required.

License GPL-2

NeedsCompilation no

Author Olivier Lepais [aut, cre],
Jason Weir [aut]

Repository CRAN

Date/Publication 2016-01-06 17:22:25

R topics documented:

SimRAD-package	2
adapt.select	3
exclude.seqsite	5
insilico.digest	8
ref.DNAseq	10
sim.DNAseq	12
size.select	14

Index	17
--------------	-----------

SimRAD-package	<i>Simulations to predict the number of loci expected in RAD and GBS approaches.</i>
----------------	--------------------------------------------------------------------------------------

Description

SimRAD provides a number of functions to simulate restriction enzyme digestion, library construction and fragments size selection to predict the number of loci expected from most of the Restriction site Associated DNA (RAD) and Genotyping By Sequencing (GBS) approaches. SimRAD aims to provide an estimation of the number of loci expected from a given genome depending on protocol type and parameters allowing to assess feasibility, multiplexing capacity and the amount of sequencing required.

Details

Package:	SimRAD
Type:	Package
Version:	0.96
Date:	2015-11-04
License:	GPL-2

This package provides a number of functions to simulate restriction enzyme digestion ([insilico.digest](#)), library construction ([adapt.select](#)) and fragments size selection ([size.select](#)) to predict the number of loci expected from most of the Restriction site Associated DNA (RAD) and Genotyping By Sequencing (GBS) approaches.

Built with non-model species in mind, a function can simulate DNA sequence when no reference genome sequence is available ([sim.DNAseq](#)) providing genome size and CG content is known. Alternatively, reference genomic sequences can be used as an input ([ref.DNAseq](#)). SimRAD can accommodate various Reduced Representation Libraries methods that may combine single or double restriction enzyme digestion (RAD and GBS), fragment size selection (ddRAD and ezRAD) and restriction site based fragment exclusion (RESTseq).

Author(s)

Olivier Lepais and Jason Weir

Maintainer: Olivier Lepais <olepais@st-pee.inra.fr>

References

Lepais O & Weir JT. 2014. SimRAD: an R package for simulation-based prediction of the number of loci expected in RADseq and similar genotyping by sequencing approaches. *Molecular Ecology Resources*, 14, 1314-1321. DOI: 10.1111/1755-0998.12273.

Examples

```

### Example 1: a single digestion (RAD)
simseq <- sim.DNAseq(size=100000, GCfreq=0.51)
#Restriction Enzyme 1
#SbfI
cs_5p1 <- "CCTGCA"
cs_3p1 <- "GG"
simseq.dig <- insilico.digest(simseq, cs_5p1, cs_3p1, verbose=TRUE)

### Example 2: a single digestion (GBS)
simseq <- sim.DNAseq(size=100000, GCfreq=0.51)
#Restriction Enzyme 1
# ApeKI : G|CWGC which is equivalent of either G|CAGC or G|CTGC
cs_5p1 <- "G"
cs_3p1 <- "CAGC"
cs_5p2 <- "G"
cs_3p2 <- "CTGC"
simseq.dig <- insilico.digest(simseq, cs_5p1, cs_3p1, cs_5p2, cs_3p2, verbose=TRUE)

### Example3: a double digestion (ddRAD)
# simulating some sequence:
simseq <- sim.DNAseq(size=1000000, GCfreq=0.433)

#Restriction Enzyme 1
#TaqI
cs_5p1 <- "T"
cs_3p1 <- "CGA"
#Restriction Enzyme 2
#MseI #
cs_5p2 <- "T"
cs_3p2 <- "TAA"

simseq.dig <- insilico.digest(simseq, cs_5p1, cs_3p1, cs_5p2, cs_3p2, verbose=TRUE)
simseq.sel <- adapt.select(simseq.dig, type="AB+BA", cs_5p1, cs_3p1, cs_5p2, cs_3p2)

# wide size selection (200-270):
wid.simseq <- size.select(simseq.sel, min.size = 200, max.size = 270, graph=TRUE, verbose=TRUE)

# narrow size selection (210-260):
nar.simseq <- size.select(simseq.sel, min.size = 210, max.size = 260, graph=TRUE, verbose=TRUE)

#the resulting fragment characteristics can be further examined:
boxplot(list(width(simseq.sel), width(wid.simseq), width(nar.simseq)), names=c("All fragments",
"Wide size selection", "Narrow size selection"), ylab="Locus size (bp)")

```

Description

Given a vector of sequences representing DNA fragments digested by two restriction enzymes (RE1 and RE2), the function will return fragments flanked by either the same restriction site (E1-E1 fragments typically for RESTseq method) or different restriction sites (RE1 and RE2, for ezRAD and ddRAD methods).

Usage

```
adapt.select(sequences, type = "AB+BA", cut_site_5prime1, cut_site_3prime1,
             cut_site_5prime2, cut_site_3prime2)
```

Arguments

sequences	a vector of DNA sequences representing DNA fragments after digestion by two restriction enzymes, typically the output of the insilico.digest function.
type	type of fragments to return. Options are "AA": fragments flanked by two restriction enzyme sites, "AB+BA": the default, fragments flanked by two different restriction enzyme sites, "BB": fragments flanked by two restriction enzyme sites. For flexibility, "AB" and "BA" options are also available and will return directional digested fragments but such fragments are of no use in current GBS methods as yet.
cut_site_5prime1	5 prime side (left side) of the recognized restriction site of the restriction enzyme 1.
cut_site_3prime1	3 prime side (right side) of the recognized restriction site of the restriction enzyme 1.
cut_site_5prime2	5 prime side (left side) of the recognized restriction site of the restriction enzyme 2.
cut_site_3prime2	3 prime side (right side) of the recognized restriction site of the restriction enzyme 2.

Details

This function will be generally needed when simulating double digest ddRAD methods where fragments flanked by two different restriction site are selected during library construction (type="AB+BA").

In addition, when simulating RESTseq method, type = "AA" can be used to exclude fragments that contained the restriction site of enzyme 2, as an alternative or in complement to subsequent DNA fragment exclusion by additional restriction site (see function [exclude.seqsites](#)).

Value

A vector of DNA fragment sequences.

Author(s)

Olivier Lepais

References

Lepais O & Weir JT. 2014. SimRAD: an R package for simulation-based prediction of the number of loci expected in RADseq and similar genotyping by sequencing approaches. *Molecular Ecology Resources*, 14, 1314-1321. DOI: 10.1111/1755-0998.12273.

Peterson et al. 2012. Double Digest RADseq: an inexpensive method for de novo SNP discovery and genotyping in model and non-model species. *PLoS ONE* 7: e37135. doi:10.1371/journal.pone.0037135

Stolle & Moritz 2013. RESTseq - Efficient benchtop population genomics with RESTriction fragment SEQuencing. *PLoS ONE* 8: e63960. doi:10.1371/journal.pone.0063960

See Also

[insilico.digest](#), [exclude.seqsite](#), [size.select](#).

Examples

```
# simulating some sequence:
simseq <- sim.DNAseq(size=10000, GCfreq=0.433)

#Restriction Enzyme 1
#PstI
cs_5p1 <- "CTGCA"
cs_3p1 <- "G"

#Restriction Enzyme 2
#MseI
cs_5p2 <- "T"
cs_3p2 <- "TAA"

# double digestion:
simseq.dig <- insilico.digest(simseq, cs_5p1, cs_3p1, cs_5p2, cs_3p2, verbose=TRUE)
#selection of AB type fragments
simseq.selected <- adapt.select(simseq.dig, type="AB+BA", cs_5p1, cs_3p1, cs_5p2, cs_3p2)
# number of selected fragments:
length(simseq.selected)
```

Description

Given a vector of sequences representing DNA fragments digested by restriction enzyme, the function return the DNA fragments that do not contain a specified restriction site, which is typically used to reduce the number of loci in the RESTseq method. The function can be use repeatedly for excluding fragments containing several restriction sites.

Usage

```
exclude.seqsite(sequences, site, verbose=TRUE)
```

Arguments

sequences	a vector of DNA sequences representing DNA fragments after digestion by restriction enzyme(s), typically the output of another function such as adapt.select , insilico.digest or exclude.seqsite .
site	restriction site to target DNA fragments to exclude. This typically corresponds to recognition site of a frequent cutter restriction enzyme.
verbose	If TRUE (the default), returns the number of fragments excluded and kept. FALSE makes the function silent to be used in a loop.

Details

Frequent cutter restriction enzyme can be easily used to further reduce the number of fragments as demonstrated by RESTseq method. This approach looks interesting in some species with complex genomes as it allows removing parts of the genomes containing highly repetitive CG or / and AT rich sequences.

This function can be used directly after a single enzyme digestion using [insilico.digest](#) function to remove fragments containing restriction site of a second enzyme. An equivalent alternative would be to simulate a double digestion using [insilico.digest](#) followed by [adapt.select](#) with type = "AA", which would remove fragments containing restriction site of the enzyme 2 (see example below).

An unlimited number of exclusion steps using different restriction enzyme can be simulated by running the function with the output of a previous execution of the function (see example below).

Value

A vector of DNA fragment sequences.

Author(s)

Olivier Lepais

References

Lepais O & Weir JT. 2014. SimRAD: an R package for simulation-based prediction of the number of loci expected in RADseq and similar genotyping by sequencing approaches. *Molecular Ecology Resources*, 14, 1314-1321. DOI: 10.1111/1755-0998.12273.

Stolle & Moritz 2013. RESTseq - Efficient benchtop population genomics with RESTriction fragment SEQuencing. PLoS ONE 8: e63960. doi:10.1371/journal.pone.0063960

See Also

[adapt.select](#), [size.select](#).

Examples

```
### Example 1:
# simulating some sequence:
simseq <- sim.DNAseq(size=1000000, GCfreq=0.433)

#Restriction Enzyme 1
#PstI
cs_5p1 <- "CTGCA"
cs_3p1 <- "G"

#Restriction Enzyme 2
#MseI #
cs_5p2 <- "T"
cs_3p2 <- "TAA"
# hence, recognition site: "TTAA"

# single digestion:
simseq.dig <- insilico.digest(simseq, cs_5p1, cs_3p1, cs_5p1, cs_3p1, verbose=TRUE)
# excluding fragments containing restriction site of the enzyme 2
simseq.exc <- exclude.seqsites(simseq.dig, "TTAA")

## which is equivalent to:
simseq.dig2 <- insilico.digest(simseq, cs_5p1, cs_3p1, cs_5p2, cs_3p2, verbose=TRUE)
simseq.selectAA <- adapt.select(simseq.dig2, type="AA", cs_5p1, cs_3p1, cs_5p2, cs_3p2)
length(simseq.selectAA)

### Example 2:
simseq <- sim.DNAseq(size=1000000, GCfreq=0.51)

#Restriction Enzyme 1
#TaqI
cs_5p1 <- "T"
cs_3p1 <- "CGA"

simseq.dig <- insilico.digest(simseq, cs_5p1, cs_3p1, cs_5p1, cs_3p1, verbose=TRUE)

# removing fragments containing restriction sites of MseI ("TTAA"), MliCI ("AATT"),
# HaellI ("GGCC"), MspI ("CCGG") and HinP1I ("GCGC"):
excl1 <- exclude.seqsites(simseq.dig, "TTAA")
excl2 <- exclude.seqsites(excl1, "AATT")
excl3 <- exclude.seqsites(excl2, "GGCC")
excl4 <- exclude.seqsites(excl3, "CCGG")
```

```
excl5 <- exclude.seqsites(excl4, "GCGC")
# which can be followed by size selection step.
```

insilico.digest	<i>Function to digest DNA sequence using one or two restriction enzyme(s).</i>
-----------------	--------------------------------------------------------------------------------

Description

Perform an in silico digestion of a DNA sequence using one or two restriction enzyme(s).

Usage

```
insilico.digest(DNAseq, cut_site_5prime1, cut_site_3prime1,
               cut_site_5prime2=NULL, cut_site_3prime2=NULL,
               cut_site_5prime3=NULL, cut_site_3prime3=NULL,
               cut_site_5prime4=NULL, cut_site_3prime4=NULL,
               verbose = TRUE)
```

Arguments

DNaseq	A DNA sequence (character string) as output for instance, but not exclusively, by sim.DNaseq or ref.DNaseq functions.
cut_site_5prime1	5 prime side (left side) of the recognized restriction site of the restriction enzyme 1.
cut_site_3prime1	3 prime side (right side) of the recognized restriction site of the restriction enzyme 1.
cut_site_5prime2	5 prime side (left side) of the recognized restriction site of the restriction enzyme 2 (optional).
cut_site_3prime2	3 prime side (right side) of the recognized restriction site of the restriction enzyme 2 (optional).
cut_site_5prime3	5 prime side (left side) of the recognized restriction site of the restriction enzyme 3 (optional).
cut_site_3prime3	3 prime side (right side) of the recognized restriction site of the restriction enzyme 3 (optional).
cut_site_5prime4	5 prime side (left side) of the recognized restriction site of the restriction enzyme 4 (optional).

cut_site_3prime4	3 prime side (right side) of the recognized restriction site of the restriction enzyme 4 (optional).
verbose	If TRUE (the default), returns the number of restriction sites (if only one restriction enzyme is used) or the number of restriction sites for the two enzymes and the number of fragments flanked by either restriction site of the enzyme 1 or 2 and flanked by restriction fragments of two different enzymes. FALSE makes the function silent to be used in a loop.

Details

Restriction site with alternative bases such as ApeKI used for GBS by Elshire et al. (2011) can be simulated by specifying the two alternative recognition motifs of the enzyme as parameter for enzyme 1 and enzyme 2 (see example below).

Value

A vector of DNA fragments resulting from the digestion.

Author(s)

Olivier Lepais and Jason Weir

References

- Lepais O & Weir JT. 2014. SimRAD: an R package for simulation-based prediction of the number of loci expected in RADseq and similar genotyping by sequencing approaches. *Molecular Ecology Resources*, 14, 1314-1321. DOI: 10.1111/1755-0998.12273.
- Baird et al. 2008. Rapid SNP discovery and genetic mapping using sequenced RAD markers. *PLoS ONE* 3: e3376. doi:10.1371/journal.pone.0003376
- Elshire et al. 2011. A robust, simple Genotyping-By-Sequencing (GBS) approach for high diversity species. *PLoS ONE* 6: e19379. doi:10.1371/journal.pone.0019379
- Peterson et al. 2012. Double Digest RADseq: an inexpensive method for de novo SNP discovery and genotyping in model and non-model species. *PLoS ONE* 7: e37135. doi:10.1371/journal.pone.0037135
- Poland et al. 2012. Development of high-density genetic maps for barley and wheat using a novel two-enzyme Genotyping-By-Sequencing approach. *PLoS ONE* 7: e32253. doi:10.1371/journal.pone.0032253
- Stolle & Moritz 2013. RESTseq - Efficient benchtop population genomics with RESTriction fragment SEQuencing. *PLoS ONE* 8: e63960. doi:10.1371/journal.pone.0063960
- Toonen et al. 2013. ezRAD: a simplified method for genomic genotyping in non-model organisms. *PeerJ* 1:e203 <http://dx.doi.org/10.7717/peerj.203>

See Also

[sim.DNAseq](#), [ref.DNAseq](#).

Examples

```

### Example 1: a single digestion (RAD)
simseq <- sim.DNAseq(size=1000000, GCfreq=0.51)
#Restriction Enzyme 1
#SbfI
cs_5p1 <- "CCTGCA"
cs_3p1 <- "GG"
simseq.dig <- insilico.digest(simseq, cs_5p1, cs_3p1, verbose=TRUE)

### Example 2: a single digestion (GBS)
simseq <- sim.DNAseq(size=1000000, GCfreq=0.51)
#Restriction Enzyme 1
# ApeKI : G|CWGC which is equivalent of either G|CAGC or G|CTGC
cs_5p1 <- "G"
cs_3p1 <- "CAGC"
cs_5p2 <- "G"
cs_3p2 <- "CTGC"
simseq.dig <- insilico.digest(simseq, cs_5p1, cs_3p1, cs_5p2, cs_3p2, verbose=TRUE)

### Example 3: a double digestion (ddRAD)
# simulating some sequence:
simseq <- sim.DNAseq(size=1000000, GCfreq=0.433)
#Restriction Enzyme 1
#PstI
cs_5p1 <- "CTGCA"
cs_3p1 <- "G"
#Restriction Enzyme 2
#MseI #
cs_5p2 <- "T"
cs_3p2 <- "TAA"
simseq.dig <- insilico.digest(simseq, cs_5p1, cs_3p1, cs_5p2, cs_3p2, verbose=TRUE)

```

ref.DNAseq

Function to import a reference DNA sequence from a Fasta file.

Description

This function read a Fasta file containing a genome reference DNA sequence in the form of several contigs (or alternatively a single sequence) and can optionally randomly sub-select a fraction of the contigs sequence to allow for faster computation and avoid memory saturation.

Usage

```
ref.DNAseq(FASTA.file, subselect.contigs = TRUE, prop.contigs = 0.1)
```

Arguments

FASTA.file	Full patch to a Fasta file (e.g. "C:/Users/Me/Documents/RAD/refGenome.fa").
subselect.contigs	If TRUE (the default) and if the Fasta file contains more than one sequence, sub-select a fraction of the contig sequences. If FALSE or if the Fasta file contains only one continuous DNA sequence, all data contain in the Fasta file is imported.
prop.contigs	Proportion of contigs to randomly sub-select for subsequent analyses.

Details

To limit memory usage and speed up computation time, it is strongly recommended to sub-select a fraction of the available reference DNA contigs. Usually, a proportion of 0.10 of the reference contigs should be representative enough to give a good idea of the genome characteristics of the species. The ratio of the length of contig sequence kept to the estimated total size of the genome can then be used to estimate the number of loci by cross-multiplication (see example below).

Contig sequences will be randomly concatenated to form a chimeric unique continuous sequence (comparable to simulated sequence by [sim.DNAseq](#) function). This may create some bias as two restriction site located in two different contigs may thus form a fragments that would not exist in the real genome. Yet, contigs do not represent real entities and keeping them apart would also create biases by randomly generating false digested restriction site which would heavily inflate the number of sites and loci particularly for draft reference genome with hundred of thousand of contigs. On the contrary, the choice to concatenate contigs may only slightly increase the number of restriction site and the number of fragments (loci).

Value

A single continuous DNA sequence (character string).

Author(s)

Olivier Lepais

References

Lepais O & Weir JT. 2014. SimRAD: an R package for simulation-based prediction of the number of loci expected in RADseq and similar genotyping by sequencing approaches. *Molecular Ecology Resources*, 14, 1314-1321. DOI: 10.1111/1755-0998.12273.

See Also

[sim.DNAseq](#), [insilico.digest](#).

Examples

```
# generating a Fasta file for the example:
sq<-c()
for(i in 1:20){
```

```

sq <- c(sq, sim.DNAseq(size=rpois(1, 10000), GCfreq=0.444))
}
sq <- DNASTringSet(sq)
writeFasta(sq, file="SimRAD-exampleRefSeq.fa", mode="w")

# importing the Fasta file and sub selecting 25% of the contigs
rfsq <- ref.DNAseq("SimRAD-exampleRefSeq.fa", subselect.contigs = TRUE, prop.contigs = 0.25)

# length of the reference sequence:
width(rfsq)

# ratio for the cross-multiplication of the number of fragments and loci at the genomes scale:
genome.size <- 100000000 # genome size: 100Mb
ratio <- genome.size/width(rfsq)
ratio

# computing GC content:
require(seqinr)
GC(s2c(rfsq))

# simulating random generated DNA sequence with characteristics equivalent to
# the sub-selected reference genome for comparison purpose:
smsq <- sim.DNAseq(size=width(rfsq), GCfreq=GC(s2c(rfsq)))

```

sim.DNAseq

Function to simulate randomly generated DNA sequence.

Description

The function randomly generated DNA sequence of a given length and with fixed GC content to simulate DNA sequence representing a (proportion of the) genome for non-model species without available reference genome sequence.

Usage

```
sim.DNAseq(size = 10000, GCfreq = 0.46)
```

Arguments

size	DNA sequence length to generate in bp. Could be the known or estimated size of the species genome or more conveniently a fraction large enough to be representative of the full length size to limit memory usage and speed up computations.
GCfreq	GC content expressed as the proportion of G and C bases in the genome.

Details

If no reference genome sequence or some reference contigs are available for a species, knowledge of GC content and genome size are needed to generate random DNA sequence representative of a species. If reference genome sequence (even draft or unassembled contigs) are available, a randomly generated sequence can be simulate following CG content and length of an import proportion of the reference sequence for comparison purpose (see example below).

Value

A single continuous DNA sequence (character string).

Author(s)

Olivier Lepais

References

Lepais O & Weir JT. 2014. SimRAD: an R package for simulation-based prediction of the number of loci expected in RADseq and similar genotyping by sequencing approaches. *Molecular Ecology Resources*, 14, 1314-1321. DOI: 10.1111/1755-0998.12273.

See Also

[ref.DNAseq](#), [insilico.digest](#).

Examples

```
## example 1: simulation of random sequence for non-model species without reference genome:
# generating 1Mb of DNA sequence with 44.4% GC content:
smsq <- sim.DNAseq(size=1000000, GCfreq=0.444)

# length:
width(smsq)

# GC content:
require(seqinr)
GC(s2c(smsq))

## example 2: simulating random sequence with parameter following a known reference genome sequence:

# generating a Fasta file for the example:
sq<-c()
for(i in 1:10){
  sq <- c(sq, sim.DNAseq(size=rpois(1, 1000), GCfreq=0.444))
}
sq <- DNASTringSet(sq)
writeFasta(sq, file="SimRAD-exampleRefSeq-Fasta.fa", mode="w")

# importing the Fasta file and sub-selecting 25% of the contigs
```

```

rfsq <- ref.DNAseq("SimRAD-exampleRefSeq-Fasta.fa", subselect.contigs = TRUE, prop.contigs = 0.25)

# length of the reference sequence:
width(rfsq)

# computing GC content:
require(seqinr)
GC(s2c(rfsq))

# simulating random generated DNA sequence with characteristics equivalent to
# the sub-selected reference genome for comparison purpose:
smsq <- sim.DNAseq(size=width(rfsq), GCfreq=GC(s2c(rfsq)))

```

size.select

Function to select fragments according to their size.

Description

Given a vector of sequences representing DNA fragments digested by one or several restriction enzymes, the function will return fragments within a specified size range which will simulate the size selection step typical of ddRAD, RESTseq and ezRAD methods.

Usage

```
size.select(sequences, min.size, max.size, graph = TRUE, verbose = TRUE)
```

Arguments

sequences	a vector of DNA sequences representing DNA fragments after digestion, typically the output of the <code>insilico.digest</code> or <code>adapt.select</code> functions.
min.size	minimum fragment size.
max.size	maximum fragment size.
graph	if TRUE (the default) the function returns a histogram of distribution of fragment size (in grey) and selected fragments within the specified size range (in red). This may be useful to further adjust the selected size windows to increase or decrease the targeted number of loci. If FALSE, the histogram is not plotted.
verbose	if TRUE (the default) the function returns the number of loci selected. If FALSE, the function is silent.

Details

Size selection is usually performed after adaptor ligation in real life, but as adaptors are not simulated here (because they are specific to the sequencing platform and the protocol used) the user should remember to account for the adaptor length when comparing size selection in the lab and in silico. For instance, size selection of 210-260 in silico correspond to size selection of 300-350 in the lab for adaptors total length of 90bp.

Value

A vector of DNA fragment sequences.

Author(s)

Olivier Lepais

References

Lepais O & Weir JT. 2014. SimRAD: an R package for simulation-based prediction of the number of loci expected in RADseq and similar genotyping by sequencing approaches. *Molecular Ecology Resources*, 14, 1314-1321. DOI: 10.1111/1755-0998.12273.

Peterson et al. 2012. Double Digest RADseq: an inexpensive method for de novo SNP discovery and genotyping in model and non-model species. *PLoS ONE* 7: e37135. doi:10.1371/journal.pone.0037135

Stolle & Moritz 2013. RESTseq - Efficient benchtop population genomics with RESTriction fragment SEQuencing. *PLoS ONE* 8: e63960. doi:10.1371/journal.pone.0063960

Toonen et al. 2013. ezRAD: a simplified method for genomic genotyping in non-model organisms. *PeerJ* 1:e203 <http://dx.doi.org/10.7717/peerj.203>

See Also

[adapt.select](#), [exclude.seqsites](#).

Examples

```
### Example: a double digestion (ddRAD)
# simulating some sequence:
simseq <- sim.DNAseq(size=1000000, GCfreq=0.433)

#Restriction Enzyme 1
#TaqI
cs_5p1 <- "T"
cs_3p1 <- "CGA"
#Restriction Enzyme 2
#MseI #
cs_5p2 <- "T"
cs_3p2 <- "TAA"

simseq.dig <- insilico.digest(simseq, cs_5p1, cs_3p1, cs_5p2, cs_3p2, verbose=TRUE)
simseq.sel <- adapt.select(simseq.dig, type="AB+BA", cs_5p1, cs_3p1, cs_5p2, cs_3p2)

# wide size selection (200-270):
wid.simseq <- size.select(simseq.sel, min.size = 200, max.size = 270, graph=TRUE, verbose=TRUE)

# narrow size selection (210-260):
nar.simseq <- size.select(simseq.sel, min.size = 210, max.size = 260, graph=TRUE, verbose=TRUE)

#the resulting fragment characteristics can be further examined:
```

```
boxplot(list(width(simseq.sel), width(wid.simseq), width(nar.simseq)), names=c("All fragments",  
"Wide size selection", "Narrow size selection"), ylab="Locus size (bp)")
```

Index

- * **GBS**
 - insilico.digest, 8
 - ref.DNAseq, 10
 - sim.DNAseq, 12
 - SimRAD-package, 2
 - size.select, 14
- * **RAD**
 - insilico.digest, 8
 - ref.DNAseq, 10
 - sim.DNAseq, 12
 - SimRAD-package, 2
 - size.select, 14
- * **RESTseq**
 - adapt.select, 3
 - exclude.seqsites, 5
 - insilico.digest, 8
 - ref.DNAseq, 10
 - sim.DNAseq, 12
 - SimRAD-package, 2
 - size.select, 14
- * **adaptator ligation**
 - adapt.select, 3
 - exclude.seqsites, 5
 - insilico.digest, 8
 - ref.DNAseq, 10
 - sim.DNAseq, 12
 - SimRAD-package, 2
 - size.select, 14
- * **ddRAD**
 - adapt.select, 3
 - insilico.digest, 8
 - ref.DNAseq, 10
 - sim.DNAseq, 12
 - SimRAD-package, 2
 - size.select, 14
- * **double digestion**
 - adapt.select, 3
 - exclude.seqsites, 5
 - insilico.digest, 8
- ref.DNAseq, 10
 - sim.DNAseq, 12
 - SimRAD-package, 2
 - size.select, 14
- * **ezRAD**
 - insilico.digest, 8
 - ref.DNAseq, 10
 - sim.DNAseq, 12
 - SimRAD-package, 2
 - size.select, 14
- * **fragment selection**
 - adapt.select, 3
 - exclude.seqsites, 5
 - insilico.digest, 8
 - ref.DNAseq, 10
 - sim.DNAseq, 12
 - SimRAD-package, 2
 - size.select, 14
- * **library construction**
 - adapt.select, 3
 - exclude.seqsites, 5
 - insilico.digest, 8
 - ref.DNAseq, 10
 - sim.DNAseq, 12
 - SimRAD-package, 2
 - size.select, 14
- * **package**
 - SimRAD-package, 2
- * **restriction enzyme**
 - adapt.select, 3
 - exclude.seqsites, 5
 - insilico.digest, 8
 - ref.DNAseq, 10
 - sim.DNAseq, 12
 - SimRAD-package, 2
 - size.select, 14
- * **restriction exclusion**
 - exclude.seqsites, 5
 - insilico.digest, 8

- ref.DNAseq, 10
- sim.DNAseq, 12
- SimRAD-package, 2
- size.select, 14

adapt.select, 2, 3, 6, 7, 14, 15

exclude.seqsites, 4, 5, 5, 6, 15

insilico.digest, 2, 4-6, 8, 11, 13, 14

ref.DNAseq, 2, 8, 9, 10, 13

sim.DNAseq, 2, 8, 9, 11, 12

SimRAD (SimRAD-package), 2

SimRAD-package, 2

size.select, 2, 5, 7, 14