

# Package ‘sapFinder’

October 18, 2017

**Type** Package

**Title** A package for variant peptides detection and visualization in shotgun proteomics.

**Version** 1.14.0

**Date** 2014-11-21

**Author** Shaohang Xu, Bo Wen

**Maintainer** Shaohang Xu <xsh.skye@gmail.com>, Bo Wen <wenbo@genomics.cn>

**Depends** R (>= 3.0.0),rTANDEM (>= 1.3.5)

**Suggests** RUnit, BiocGenerics, BiocStyle

**Imports** pheatmap,Rcpp (>= 0.10.6),graphics,grDevices,stats, utils

**biocViews** MassSpectrometry, Proteomics, SNP, RNASeq, Visualization, ReportWriting

**Description** sapFinder is developed to automate  
(1) variation-associated database construction,  
(2) database searching,  
(3) post-processing,  
(4) HTML-based report generation in shotgun proteomics.

**License** GPL-2

**LazyLoad** yes

**LinkingTo** Rcpp

**NeedsCompilation** yes

## R topics documented:

dbCreator . . . . .	2
easyRun . . . . .	3
parserGear . . . . .	4
reportCreator . . . . .	5
runTandem . . . . .	6

<b>Index</b>	<b>8</b>
--------------	----------

dbCreator

*dbCreator***Description**

An integrated function to generate variation-associated database based on sample-specific NGS data or public SNV data.

**Usage**

```
dbCreator(vcf = NULL, annotation = NULL, refseq = NULL, outdir = "./",
          prefix = "test", xmx = NULL, xref = "noxref")
```

**Arguments**

vcf	Input VCF file name. This file contains the information of gene sequence variations.
annotation	Input annotation file name. It contains the gene annotation information and can be downloaded from UCSC Genome Browser. Currently it supports RefSeq genes and ENSEMBL genes annotation file.
refseq	Input mRNA sequences file with FASTA format. It can be downloaded from UCSC Genome Browser.
outdir	Output directory.
prefix	The prefix of output file.
xmx	The maximum Java heap size. The unit is "G".
xref	Optional external cross-reference file, generally it's downloaded through BioMart. If this file is provided, the final html report will present some relevant protein id or description.

**Value**

A vector containing two file names. One is a FASTA format file contains the mutated peptides, the normal protein sequences and their reverse versions, and the other is a tab-delimited file contains detailed variation information.

**Examples**

```
vcf      <- system.file("extdata/sapFinder_test.vcf",
                       package="sapFinder")
annotation <- system.file("extdata/sapFinder_test_ensGene.txt",
                          package="sapFinder")
refseq    <- system.file("extdata/sapFinder_test_ensGeneMrna.fa",
                          package="sapFinder")
xref      <- system.file("extdata/sapFinder_test_BioMart.Xref.txt",
                          package="sapFinder")
outdir    <- "db_dir"
prefix    <- "sapFinder_test"
db.files  <- dbCreator(vcf=vcf, annotation=annotation,
                      refseq=refseq, outdir=outdir,
                      prefix=prefix, xref=xref)
```

easyRun

*easyRun***Description**

This function is used to automate the variation-associated database construction, MS/MS searching, post-processing and HTML-based report generation.

**Usage**

```
easyRun(vcf = NULL, annotation = NULL, refseq = NULL, outdir = "./",
        prefix = "sapFinder_test", spectra = "", cpu = 1, enzyme = "[KR][X]",
        tol = 10, tolu = "ppm", itol = 0.6, itolu = "Daltons",
        varmod = NULL, fixmod = NULL, miss = 2, maxCharge = 8, ti = FALSE,
        alignment = 1, xref = "noxref", xmx = NULL, ...)
```

**Arguments**

vcf	Input VCF file name. This file contains the information of gene sequence variations.
annotation	Input annotation file name. It contains the gene annotation information and can be downloaded from UCSC Genome Browser. Currently it supports RefSeq genes and ENSEMBL genes annotation file.
refseq	Input mRNA sequences file with FASTA format. It can be downloaded from UCSC Genome Browser.
outdir	Output directory.
prefix	The prefix of output file.
spectra	MS/MS peak list file
cpu	The number of CPU used for X!Tandem search. Default is 1.
enzyme	Specification of specific protein cleavage sites. Default is "[KR][X]".
varmod	Specification of potential modifications of residues.
fixmod	Specification of modifications of residues.
tol	Parent ion mass tolerance (monoisotopic mass).
tolu	Parent ion M+H mass tolerance window units.
itol	Fragment ion mass tolerance (monoisotopic mass).
itolu	Unit for fragment ion mass tolerance (monoisotopic mass).
miss	The number of missed cleavage sites. Default is 2.
maxCharge	The Maximum parent charge, default is 8
ti	anticipate carbon isotope parent ion assignment errors. Default is false.
alignment	0 or 1 to determine if peptide should be alignment or not. Default is 0.
xmx	The maximum Java heap size. The unit is "G".
xref	Optional external cross-reference file,generally it's downloaded through BioMart.If this file is provided,the final html report will present some relevant protein id or description.
...	Additional arguments

## Examples

```
vcf      <- system.file("extdata/sapFinder_test.vcf",
                        package="sapFinder")
annotation <- system.file("extdata/sapFinder_test_ensGene.txt",
                          package="sapFinder")
refseq   <- system.file("extdata/sapFinder_test_ensGeneMrna.fa",
                        package="sapFinder")
mgf.path <- system.file("extdata/sapFinder_test.mgf",
                        package="sapFinder")
xref     <- system.file("extdata/sapFinder_test_BioMart.Xref.txt",
                        package="sapFinder")
easyRun(vcf=vcf,annotation=annotation,refseq=refseq,outdir="test",
        prefix="sapFinder_test",spectra=mgf.path,cpu=0,tol=10, tolu="ppm", itol=0.1,
        itolu="Daltons",alignment=1,xref=xref)
```

---

parserGear

*parserGear*

---

## Description

This function is mainly for q-value calculation, protein inference and variant peptides spectra annotation.

## Usage

```
parserGear(file = NULL, db = NULL, outdir = "parser_outdir",
           prefix = "sapFinder_test", mutPrefix = "VAR", decoyPrefix = "###REV###",
           alignment = 1, xmx = NULL, thread = 1)
```

## Arguments

file	MS/MS search file. Currently, only XML format file of X!Tandem and DAT result of Mascot are supported.
db	A FASTA format database file used for MS/MS searching. Usually, it is from the output of the function dbCreator.
outdir	Output directory.
prefix	The prefix of output file.
mutPrefix	The prefix of variant peptides ID. Default is "VAR". "VAR" is the prefix which used by function dbCreator.
decoyPrefix	The prefix of decoy sequences ID. Default is "###REV###". "###REV###" is the prefix which used by function dbCreator.
alignment	0 or 1 to determine if peptide should be alignment or not. Default is 1.
thread	This parameter is used to specify the number of threads. "0" represents that all of the available threads are used; "1" represents one thread is used; "2" represents two threads are used, and so on. Default is 1.
xmx	The maximum Java heap size. The unit is "G".

**Examples**

```

## Step 1. Variation-associated database construction
vcf      <- system.file("extdata/sapFinder_test.vcf",
                        package="sapFinder")
annotation <- system.file("extdata/sapFinder_test_ensGene.txt",
                          package="sapFinder")
refseq    <- system.file("extdata/sapFinder_test_ensGeneMrna.fa",
                          package="sapFinder")
xref      <- system.file("extdata/sapFinder_test_BioMart.Xref.txt",
                          package="sapFinder")

outdir    <- "db_dir"
prefix    <- "sapFinder_test"
db.files  <- dbCreator(vcf=vcf, annotation=annotation,
                      refseq=refseq, outdir=outdir,
                      prefix=prefix, xref=xref)

## Step 2. MS/MS searching
mgf.path  <- system.file("extdata/sapFinder_test.mgf",
                          package="sapFinder")
fasta.path <- db.files[1]
xml.path  <- runTandem(spectra=mgf.path, fasta=fasta.path, outdir=".",
                      tol=10, tolu="ppm", itol=0.1, itolu="Daltons")

## Step 3. Post-processing
parserGear(file=xml.path, db=fasta.path, prefix=prefix,
           outdir="parser_outdir", alignment=1)

```

---

reportCreator

*reportCreator*


---

**Description**

This function is used for HTML-based report writing

**Usage**

```
reportCreator(indir = ".", outdir = .REPORT.DIR, db = NULL,
             prefix = NULL, varInfor = NULL)
```

**Arguments**

indir	The directory of output files of function parserGear.
outdir	Output directory for this report
db	A FASTA format database file used for MS/MS searching. Usually, it is from the output of the function dbCreator.
prefix	It must be set the same with the parameter of "prefix" in function parserGear.
varInfor	It is a tab-delimited file contains detailed variation information and is from the output of the function dbCreator.

## Examples

```
## Step 1. Variation-associated database construction
vcf      <- system.file("extdata/sapFinder_test.vcf",
                        package="sapFinder")
annotation <- system.file("extdata/sapFinder_test_ensGene.txt",
                          package="sapFinder")
refseq    <- system.file("extdata/sapFinder_test_ensGeneMrna.fa",
                          package="sapFinder")
xref      <- system.file("extdata/sapFinder_test_BioMart.Xref.txt",
                          package="sapFinder")

outdir    <- "db_dir"
prefix    <- "sapFinder_test"
db.files  <- dbCreator(vcf=vcf, annotation=annotation,
                      refseq=refseq, outdir=outdir,
                      prefix=prefix, xref=xref)

## Step 2. MS/MS searching
mgf.path  <- system.file("extdata/sapFinder_test.mgf",
                          package="sapFinder")

fasta.path <- db.files[1]
xml.path  <- runTandem(spectra=mgf.path, fasta=fasta.path, outdir=".",
                      tol=10, tolu="ppm", itol=0.1, itolu="Daltons")

## Step 3. Post-processing
parserGear(file=xml.path, db=fasta.path, prefix=prefix,
           outdir="parser_outdir")

## Step 4. HTML-based report generation
reportCreator(indir="parser_outdir", outdir="report", db=fasta.path,
             prefix=prefix, varInfor=db.files[2])
```

---

runTandem

*run xtandem*

---

## Description

run xtandem

## Usage

```
runTandem(spectra = "", fasta = "", outdir = ".", cpu = 1,
          enzyme = "[KR]|[X]", tol = 10, tolu = "ppm", itol = 0.6,
          itolu = "Daltons", varmod = NULL, fixmod = NULL, miss = 2,
          maxCharge = 8, ti = FALSE)
```

## Arguments

spectra	MS/MS peak list file
fasta	Protein database file for searching.
outdir	The output directory.
cpu	The number of CPU used for X!Tandem search. Default is 1.
enzyme	Specification of specific protein cleavage sites. Default is "[KR] [X]".

varmod	Specification of potential modifications of residues.
fixmod	Specification of modifications of residues.
tol	Parent ion mass tolerance (monoisotopic mass).
tolu	Parent ion M+H mass tolerance window units.
itol	Fragment ion mass tolerance (monoisotopic mass).
itolu	Unit for fragment ion mass tolerance (monoisotopic mass).
miss	The number of missed cleavage sites. Default is 2.
maxCharge	The Maximum parent charge, default is 8
ti	anticipate carbon isotope parent ion assignment errors. Default is false.

**Value**

The search result file path

**Examples**

```
# Variation-associated database construction
vcf      <- system.file("extdata/sapFinder_test.vcf",
                       package="sapFinder")
annotation <- system.file("extdata/sapFinder_test_ensGene.txt",
                          package="sapFinder")
refseq    <- system.file("extdata/sapFinder_test_ensGeneMrna.fa",
                          package="sapFinder")
xref      <- system.file("extdata/sapFinder_test_BioMart.Xref.txt",
                          package="sapFinder")

outdir    <- "db_dir"
prefix    <- "sapFinder_test"
db.files  <- dbCreator(vcf=vcf, annotation=annotation,
                      refseq=refseq, outdir=outdir,
                      prefix=prefix, xref=xref)

# MS/MS searching
mgf.path  <- system.file("extdata/sapFinder_test.mgf",
                          package="sapFinder")
runTandem(spectra=mgf.path, fasta=db.files[1],
          tol=10, tolu="ppm", itol=0.1, itolu="Daltons")
```

# Index

dbCreator, [2](#)

easyRun, [3](#)

parserGear, [4](#)

reportCreator, [5](#)

runTandem, [6](#)