

# Package ‘nanotatoR’

October 16, 2019

**Title** nanotatoR: next generation structural variant annotation and classification

**Version** 1.0.0

**Author** Surajit Bhattacharya, Hayk Barsheghyan, Emmanuele C Delot and Eric Vilain

**Maintainer** Surajit Bhattacharya <sbhattach2@childrensnational.org>

**Description** Whole genome sequencing (WGS) has successfully been used to identify single-nucleotide variants (SNV), small insertions and deletions and, more recently, small copy number variants. However, due to utilization of short reads, it is not well suited for identification of structural variants (SV) and the majority of SV calling tools having high false positive and negative rates. Optical next-generation mapping (NGM) utilizes long fluorescently labeled native-state DNA molecules for de novo genome assembly to overcome the limitations of WGS. NGM allows for a significant increase in SV detection capability. However, accuracy of SV annotation is highly important for variant classification and filtration to determine pathogenicity. Here we create a new tool in R, for SV annotation, including population frequency and gene function and expression, using publicly available datasets. We use DGV (Database of Genomic Variants), to calculate the population frequency of the SVs identified by the Bionano SVCaller in the NGM dataset of a cohort of 50 undiagnosed patients with a variety of phenotypes. The new annotation tool, nanotatoR, also calculates the internal frequency of SVs, which could be beneficial in identification of potential false positive or common calls. The software creates a primary gene list (PG) from NCBI databases based on patient phenotype and compares the list to the set of genes overlapping the patient’s SVs generated by SVCaller, providing analysts with an easy way to identify variants affecting genes in the PG. The output is given in an Excel file format, which is subdivided into multiple sheets based on SV type. Users then have a choice to filter SVs using the provided annotation for identification of inherited, de novo or rare variants. nanotatoR provides integrated annotation and the expression patterns to enable users to identify potential pathogenic SVs with greater precision and faster turnaround times.

**Depends** R (>= 3.6)

**Imports** hash(>= 2.2.6), openxlsx(>= 4.0.17), rentrez(>= 1.1.0), stats, grDevices, graphics, stringr, knitr, testthat, utils, AnnotationDbi, httr, org.Hs.eg.db, rtracklayer

**Suggests** rmarkdown, yaml

**VignetteBuilder** knitr

**License** file LICENSE

**biocViews** Software, WorkflowStep, GenomeAssembly, VariantAnnotation

**Encoding** UTF-8

**RoxygenNote** 6.1.1

**URL** <https://github.com/VilainLab/Nanotator>

**BugReports** <https://github.com/VilainLab/Nanotator/issues>

**git\_url** <https://git.bioconductor.org/packages/nanotatoR>

**git\_branch** RELEASE\_3\_9

**git\_last\_commit** 7d7934f

**git\_last\_commit\_date** 2019-05-02

**Date/Publication** 2019-10-15

## R topics documented:

buildrunBNBedFiles . . . . .	2
clinvar_gene . . . . .	3
cohortFrequency . . . . .	4
compSmapped . . . . .	5
Decipher_extraction . . . . .	6
DGV_extraction . . . . .	7
gene_extraction . . . . .	8
gene_list_generation . . . . .	8
gtr_gene . . . . .	9
internalFrequency . . . . .	9
makeMergedSmappedData . . . . .	11
makeMergedSVData . . . . .	11
nanotatoR . . . . .	12
nanotatoR_main . . . . .	12
nonOverlapGenes . . . . .	15
omim_gene . . . . .	16
overlapGenes . . . . .	16
readBNBedFiles . . . . .	17
readSMap . . . . .	18
run_bionano_filter . . . . .	18
<b>Index</b>	<b>20</b>

---

buildrunBNBedFiles	<i>Reads BED files to produce bionano Bed files</i>
--------------------	---

---

### Description

Reads BED files to produce bionano Bed files

### Usage

```
buildrunBNBedFiles.bedFile, returnMethod = c("Text", "dataFrame"),
  outdir)
```

**Arguments**

bedFile            character. Path to UCSC Bed File.  
returnMethod      character. Path to output directory.  
outdir             character. Path to output directory.

**Value**

Data Frame or text file. Contains the gene information.

**Examples**

```
bedFile <- system.file("extdata", "Homo_sapiens.Hg19.bed",  
package="nanotator")  
bed<-buildrunBNBedFiles(bedFile,returnMethod="dataFrame")
```

---

clinvar\_gene            *Extracting genes from clinvar database NCBI.*

---

**Description**

Extracting genes from clinvar database NCBI.

**Usage**

```
clinvar_gene(terms)
```

**Arguments**

terms                Single or Multiple Terms.

**Value**

Dataframe returned containing gene lists in entrezid and Gene Symbols, and terms associated with it

**Examples**

```
terms="Muscle Weakness"  
ge <- clinvar_gene(terms)
```

---

cohortFrequency      *Calculates the internal frequencies of SV in bionano cohorts*

---

### Description

Calculates the internal frequencies of SV in bionano cohorts

### Usage

```
cohortFrequency(internalBNDB, smapath, smap, buildBNInternalDB = FALSE,
  smapdata, input_fmt = c("Text", "dataFrame"),
  dbOutput = c("dataframe", "text"), BNDBPath, BNDBPattern, fname,
  outpath, win_indel = 10000, win_inv_trans = 50000,
  perc_similarity = 0.5, indelconf = 0.5, invconf = 0.01,
  limsize = 1000, transconf = 0.1, returnMethod = c("Text",
  "dataFrame"))
```

### Arguments

internalBNDB	character. Path to the merged BN files.
smapath	character. path to the query smap file.
smap	character. File name for the smap
buildBNInternalDB	boolean. Checking whether the merged bionano file database exist.
smapdata	character. SV data in dataframe.
input_fmt	character. Choice between Text and DataFrame.
dbOutput	character. Output of merged bionano data.
BNDBPath	Path of Bionano database files.
BNDBPattern	Pattern of Bionano database files.
fname	character. Filename in case dbOutput = Text.
outpath	character. Path to merged SV solo datasets.
win_indel	Numeric. Insertion and deletion error window.
win_inv_trans	Numeric. Inversion and translocation error window.
perc_similarity	Numeric . ThresholdPercentage similarity of the query SV and reference SV.
indelconf	Numeric. Threshold for insertion and deletion Score.
invconf	Numeric. Threshold for inversion Score.
limsize	Numeric . Minimum size to consider for a SV.
transconf	Numeric. Threshold for translocation Score.
returnMethod	character. Choice between Text and DataFrame.

### Value

Text file or data frames containing internalFrequency data.

**Examples**

```
mergedFiles <- system.file("extdata", "BNSOLO2_merged.txt",
  package = "nanotatoR")
smapName <- "F1.1_TestSample1_solo_hg19.smap"
smappath <- system.file("extdata", package = "nanotatoR")
win_indel = 10000; win_inv_trans = 50000; perc_similarity = 0.5;
indelconf = 0.5; invconf = 0.01;transconf = 0.1
cohortFreq<-cohortFrequency(internalBNDB = mergedFiles , smappath ,
  smap=smapName, input_fmt ="Text",
  buildBNInternalDB=FALSE, win_indel, win_inv_trans,
  perc_similarity , indelconf, invconf,
  transconf,returnMethod="dataFrame")
```

---

 compSmapped

*Extracts gene information from bed files*


---

**Description**

Extracts gene information from bed files

**Usage**

```
compSmapped(smap, bed, inputfmtBed = c("BED", "BNBED"), filepath, n = 3,
  returnMethod_bedcomp = c("Text", "dataFrame"))
```

**Arguments**

smmap	character. Path to SMAP file.
bed	Text. Normal Bed files or Bionano Bed file.
inputfmtBed	character Whether the bed input is UCSC bed or Bionano bed. Note: extract in bed format to be read by bedsv: <code>awk 'print \$1,\$4,\$5,\$18,\$7' gencode.v19.annotation.gtf&gt;HomoSapien</code>
filepath	character Path for the output files.
n	numeric Number of genes to report which are nearest to the breakpoint. Default is 3.
returnMethod_bedcomp	Character. Type of output Dataframe or in Text format.

**Value**

Data Frame and Text file. Contains the smmap with additional Gene Information.

**Examples**

```
smmapName="F1.1_TestSample1_solo_hg19.smap"
smmap = system.file("extdata", smmapName, package="nanotatoR")
bedFile <- system.file("extdata", "Homo_sapiens.Hg19_BN_bed.txt",
  package="nanotatoR")
filepath <- system.file("extdata", package="nanotatoR")
datcomp<-compSmapped(smmap, bed=bedFile, inputfmtBed = "BNBED", n = 3,
  returnMethod_bedcomp = c("dataFrame"))
datcomp[1,]
```

---

Decipher\_extraction     *Frequency calculation of variants compared to DGV.*

---

### Description

Frequency calculation of variants compared to DGV.

### Usage

```
Decipher_extraction(decipherpath, smappath, smap, smap_data,
  input_fmt = c("Text", "dataFrame"), win_indel = 10000,
  perc_similarity = 0.5, returnMethod = c("Text", "dataFrame"))
```

### Arguments

decipherpath	character. Path to DECIPHER Text file.
smappath	character. path to the query smap file.
smap	character. File name for the smap.
smap_data	character. Dataframe if input type chosen as dataframe.
input_fmt	character. Choice between text or data frame as an input to the DGV frequency calculator.
win_indel	Numeric. Insertion and deletion error window. Default 10000.
perc_similarity	Numeric . ThresholdPercentage similarity of the query SV and reference SV. Default 0.5.
returnMethod	character. Choice between text or data frame as the output.

### Value

Text and character vector containing gene list and terms associated with them are stored as text files.

### Examples

```
decipherpath <- system.file("extdata", "population_cnv.txt",
  package = "nanotatoR")
smapName <- "F1.1_TestSample1_solo_hg19.smap"
smappath <- system.file("extdata", package = "nanotatoR")
win_indel=10000;win_inv_trans=50000;perc_similarity=0.5
decipherext<-Decipher_extraction (decipherpath, input_fmt = "Text", smappath,
  smap= smapName,
  win_indel = 10000, perc_similarity = 0.5, returnMethod="dataFrame")
```

---

DGV_extraction	<i>Frequency calculation of variants compared to DGV.</i>
----------------	---

---

### Description

Frequency calculation of variants compared to DGV.

### Usage

```
DGV_extraction(hgpath, smappath, smap, smap_data,
  input_fmt_DGV = c("Text", "dataFrame"), win_inde1_DGV = 10000,
  win_inv_trans_DGV = 50000, perc_similarity_DGV = 0.5,
  returnMethod = c("Text", "dataFrame"), outpath, usample = 54946)
```

### Arguments

hgpath	character. Path to Database of Genomic Variants (DGV) Text file.
smappath	character. Path for smap textfile.
smap	character. File name for smap textfile.
smap_data	dataframe. Dataset containing smap data.
input_fmt_DGV	character. Choice between text or data frame as an input to the DGV frequency calculator.
win_inde1_DGV	Numeric. Insertion and deletion error window.Default 10000 bases.
win_inv_trans_DGV	Numeric. Inversion and translocation error window. Default 50000 bases.
perc_similarity_DGV	Numeric. ThresholdPercentage similarity of the query SV and reference SV. Default 0.5.
returnMethod	character. Choice between text or data frame as the output.
outpath	character. Path where gene lists are saved.
usample	Numeric. Number of unique samples.

### Value

Text and character vector containg gene list and terms associated with them are stored as text files.

### Examples

```
## Not run:
smap <- "F1.1_TestSample1_solo_hg19.smap"
smappath <- system.file("extdata", package = "nanotatoR")
hgpath <- system.file("extdata", "GRCh37_hg19_variants_2016-05-15.txt",
  package = "nanotatoR")
win_inde1_DGV=10000;win_inv_trans_DGV=50000;perc_similarity_DGV=0.5;
usample = 54946
dgv <- DGV_extraction (hgpath, input_fmt_DGV = "Text",smap=smap,
  smappath, win_inde1_DGV = 10000, win_inv_trans_DGV = 50000,
  perc_similarity_DGV = 0.5,returnMethod="dataFrame",usample = 54946)

## End(Not run)
```

---

gene\_extraction      *Extracting genes from gene database NCBI.*

---

### Description

Extracting genes from gene database NCBI.

### Usage

```
gene_extraction(terms)
```

### Arguments

terms                  Single or Multiple Terms.

### Value

Dataframe returned containing gene lists in entrezid and Gene Symbols, and terms associated with it

### Examples

```
terms="Muscle Weakness"
ge <- gene_extraction(terms)
```

---

gene\_list\_generation      *Extracting genes for phenotype/diseases from NCBI.*

---

### Description

Extracting genes for phenotype/diseases from NCBI.

### Usage

```
gene_list_generation(method_entrez = c("Single", "Multiple", "Text"),
  termPath, term, outpath, thresh = 5, returnMethod = c("Text",
  "dataFrame"))
```

### Arguments

method\_entrez      character. Input Method for terms. Choices are "Single","Multiple" and "Text".

termPath            character. Path and file name for textfile.

term                character. Single or Multiple Terms.

outpath            character. Path where gene lists are saved.

thresh             integer. Threshold for the number of terms sent to entrez. Note if large lists are sent to ncbi, it might fail to get processed. Default is 5.

returnMethod      Method of returning output. Options, Text or data.frame.



**Value**

Text files containing gene list and terms associated with them are stored as text files.

**Examples**

```
terms="Muscle Weakness"
genes <- gene_list_generation(method="Single", term=terms, returnMethod="dataFrame")
```

---

gtr\_gene

*Extracting genes from gtr database NCBI.*


---

**Description**

Extracting genes from gtr database NCBI.

**Usage**

```
gtr_gene(terms)
```

**Arguments**

terms                    Single or Multiple Terms.

**Value**

Dataframe returned containing gene lists in entrezid and Gene Symbols, and terms associated with it

**Examples**

```
terms="Muscle Weakness"
ge <- gtr_gene(terms)
```

---

internalFrequency

*Calculates the internal frequencies of SV in internal cohorts*


---

**Description**

Calculates the internal frequencies of SV in internal cohorts

**Usage**

```
internalFrequency(mergedFiles, smappath, smap, buildSVInternalDB = FALSE,
  smapdata, input_fmt = c("Text", "dataFrame"), path, pattern, outpath,
  win_indel = 10000, win_inv_trans = 50000, perc_similarity = 0.5,
  indelconf = 0.5, invconf = 0.01, fname, limsize = 1000,
  win_indel_parents = 5000, win_inv_trans_parents = 40000,
  transconf = 0.1, dbOutput = c("dataframe", "text"),
  returnMethod = c("Text", "dataFrame"))
```

**Arguments**

mergedFiles	character. Path to the merged SV files.
smappath	character. path to the query smap file.
smap	character. File name for the smap
buildSVInternalDB	boolean. Checking whether the merged solo file database exist.
smapdata	character. Dataframe if input type chosen as dataframe.
input_fmt	Format in which data is provided as an input to the function.
path	character. Path to the solo file database.
pattern	character. pattern of the file names to merge.
outpath	character. Path to merged SV solo datasets.
win_indel	Numeric. Insertion and deletion error window. Default 10000.
win_inv_trans	Numeric. Inversion and translocation error window. Default 50000.
perc_similarity	Numeric . ThresholdPercentage similarity of the query SV and reference SV. Default 0.5.
indelconf	Numeric. Threshold for insertion and deletion confidence. Default 0.5
invconf	Numeric. Threshold for inversion confidence. Default 0.01.
fname	character. Filename in case dbOutput = Text.
limsize	Numeric. Minimum size of SV that can be determined accurately by the Bionano SV caller. Default 1000.
win_indel_parents	Numeric. Insertion and deletion error window to determine zygosity in case of parents. Default 5000.
win_inv_trans_parents	Numeric. Inversion and translocation error window to determine zygosity in case of parents. Default 40000.
transconf	Numeric. Threshold for translocation confidence. Default 0.1.
dbOutput	character. Output of merged bionano data.
returnMethod	character. Choice between Text and DataFrame.

**Value**

Text file or data frames containing internalFrequency data.

**Examples**

```
mergedFiles <- system.file("extdata", "nanotatoR_merged.txt", package = "nanotatoR")
smapName <- "F1.1_TestSample1_solo_hg19.smap"
smappath <- system.file("extdata", package = "nanotatoR")
indelconf = 0.5; invconf = 0.01;transconf = 0.1;input_fmt="Text";
internalFrequency(mergedFiles = mergedFiles, smappath = smappath , smap = smapName,
buildSVInternalDB=FALSE, win_indel=10000,
win_inv_trans=50000,
perc_similarity=0.5, indelconf=0.5, invconf=0.01,
transconf=0.1, limsize=1000, win_indel_parents=5000,input_fmt="Text",
win_inv_trans_parents=40000,
returnMethod="dataFrame")
```

---

makeMergedSmapData      *Merges Bionano SV files to one common SV file*

---

### Description

Merges Bionano SV files to one common SV file

### Usage

```
makeMergedSmapData(path, pattern, outpath, fname,
  dbOutput = c("dataframe", "text"))
```

### Arguments

path	character. Path to the solo files.
pattern	character. file name pattern for solo files.
outpath	character. path for output file if dbOutput = text.
fname	character. filename if dbOutput = text.
dbOutput	character. Output type. Default text and dataframe.

### Value

Text file containing all the solo SMAP files.

### Examples

```
path <- system.file("extdata", "Bionano_config/", package = "nanotatoR")
pattern <- "_hg19.txt"
mergedSmap <- makeMergedSmapData(path, pattern, dbOutput = "dataframe")
```

---

makeMergedSVDData      *Merges Solo SV files to one common SV file*

---

### Description

Merges Solo SV files to one common SV file

### Usage

```
makeMergedSVDData(path, pattern, outpath, fname, dbOutput = c("dataframe",
  "text"))
```

### Arguments

path	character. Path to the solo files.
pattern	character. file name pattern for solo files.
outpath	character. path for output file if dbOutput = text.
fname	character. filename if dbOutput = text.
dbOutput	character. Output type. Default text and dataframe.

**Value**

Text file containing all the solo SMAP files.

**Examples**

```
path <- system.file("extdata", "SoloFile/", package = "nanotatoR")
pattern <- "_hg19.smap"
mergedFiles <- makeMergedSVData(path = path, pattern = pattern,
dbOutput = "dataframe")
mergedFiles[1,]
```

---

nanotatoR

*nanotatoR: Annotation package for Bionano Data*

---

**Description**

Annotation of Bionano data using available databases

**Examples**

```
path <- system.file("extdata", "Bionano_config/", package = "nanotatoR")
pattern <- "_hg19.txt"
mergedSmap <- makeMergedSmapData(path, pattern, dbOutput = "dataframe")
```

---

nanotatoR\_main

*Annotation of Bionano SV.*

---

**Description**

Annotation of Bionano SV.

**Usage**

```
nanotatoR_main(smap, bed, inputfmtBed = c("BED", "BNBED"), n = 3,
InternalDBpath, InternalDBpattern, dbOutput_Int, fname_Int, dbOutput_BN,
fname_BN, buildSVInternalDB = FALSE, soloPath, solopattern,
returnMethod_bedcomp = c("Text", "dataFrame"), mergedFiles_BN,
win_indel_INF = 10000, win_inv_trans_INF = 50000,
perc_similarity_INF = 0.5, indelconf = 0.5, invconf = 0.01,
transconf = 0.1, returnMethod_DGV = c("Text", "dataFrame"), hgpath,
win_indel_DGV = 10000, win_inv_trans_DGV = 50000,
perc_similarity_DGV = 0.5, returnMethod_Internal = c("Text",
"dataFrame"), input_fmt_DGV = c("Text", "dataFrame"),
input_fmt_BN = c("Text", "dataFrame"), input_fmt_INF = c("Text",
"dataFrame"), input_fmt_decipher = c("Text", "dataFrame"),
input_fmt_svMap = c("Text", "dataFrame"), dat_geneList, decipherpath,
input_fmt_geneList = c("Text", "dataFrame"),
returnMethod_GeneList = c("Text", "dataFrame"),
buildBNInternalDB = FALSE, returnMethod_BNCohort = c("Text",
"dataFrame"), mergedFiles_INF, returnMethod_decipher = c("Text",
```

```
"dataFrame"), method_entrez = c("Single", "Multiple", "Text"),
smapName, termPath, term, thresh = 5, limsize = 1000,
win_indel_parents = 5000, win_inv_trans_parents = 40000, outpath,
outputFilename = "", RZIPpath)
```

## Arguments

smap	character. Path to SMAP file.
bed	Text Choice between UCSC bed or Bionano bed.
inputfmtBed	character. Choice between Text and DataFrame as input for bed file.
n	numeric Number of genes to report which are nearest to the breakpoint. Default is 3.
InternalDBpath	character. Path to the BNFile file database.
InternalDBpattern	character. pattern of the BNFile names to merge.
dbOutput_Int	character. Output of solo bionano data.
fname_Int	character. Filename in case dbOutput_Int = Text.
dbOutput_BN	character. Output of merged bionano data.
fname_BN	character. Filename in case dbOutput_BN = Text.
buildSVInternalDB	boolean. Checking whether the merged solo file database exist or you need to build it. Default= TRUE.
soloPath	character. Path to the solo file database.
solopattern	character. pattern of the file names to merge.
returnMethod_bedcomp	character. Return Methods from the compSmapbed function, choice between Text and Dataframe.
mergedFiles_BN	character. Path to the merged BN SV files.
win_indel_INF	Numeric. Insertion and deletion error window.
win_inv_trans_INF	Numeric. Inversion and translocation error window.
perc_similarity_INF	Numeric . ThresholdPercentage similarity of the query SV and reference SV.
indelconf	Numeric. Threshold for insertion and deletion confidence.
invconf	Numeric. Threshold for inversion confidence.
transconf	Numeric. Threshold for translocation confidence.
returnMethod_DGV	character. Return Methods from the DGV_extraction function, choice between Text and Dataframe.
hgpath	character. Path to Database of Genomic Variants (DGV) Text file.
win_indel_DGV	Numeric. Insertion and deletion error window.
win_inv_trans_DGV	Numeric. Inversion and translocation error window.
perc_similarity_DGV	Numeric . ThresholdPercentage similarity of the query SV and reference SV.

returnMethod\_Internal  
character. Return Methods from the internalFrequency function, choice between Text and Dataframe.

input\_fmt\_DGV character. Choice between Text and DataFrame for input to DGV.

input\_fmt\_BN character. Choice of Bionano dataset input Text or Dataframe.

input\_fmt\_INF character. Choice between Text and DataFrame for input to INF.

input\_fmt\_decipher  
character. Choice of gene list input Text or Dataframe.

input\_fmt\_svMap  
character. Choice of SVmap input for final step Text or Dataframe.

dat\_geneList Dataframe Input data containing geneList data.

decipherpath character. Path to DECIPHER. Text file.

input\_fmt\_geneList  
character. Choice of gene list input Text or Dataframe.

returnMethod\_GeneList  
character. Return Methods from the gene\_list\_generation function, choice between Text and Dataframe.

buildBNInternalDB  
boolean. Checking whether the merged Bionano file database exist or you need to build it. Default= TRUE.

returnMethod\_BNCohort  
character. Return Methods from the Bionano function, choice between Text and Dataframe.

mergedFiles\_INF  
character. Path to the merged BN SV files.

returnMethod\_decipher  
character. Return Methods from the decipher Frequency function, choice between Text and Dataframe.

method\_entrez character. Input Method for terms for entrez. Choices are "Single","Multiple" and "Text".

smapName character. Name of the smap file.

termPath character. Path and file name for textfile for terms.

term character. Single or Multiple Terms as vectord.

thresh integer. Threshold for the number of terms sent to entrez. Note if large lists are sent to ncbi, it might fail to get processed. Default is 5.

lmsize Numeric. Minimum size of SV that can be determined accurately by the Bionano SV caller. Default 1000.

win\_indel\_parents  
Numeric. Insertion and deletion error window to determine zygoty in case of parents. Default 5000.

win\_inv\_trans\_parents  
Numeric. Inversion and translocation error window to determine zygoty in case of parents. Default 40000.

outpath Character Directory to the output file.

outputFilename Character Output filename for the annotated smap.

RZIPpath Character. Path for the Rtools zip.exe

**Value**

Excel file containing the annotated SV map, tabs divided based on type of SVs.

---

nonOverlapGenes	<i>Calculates Genes that are near to the SV region</i>
-----------------	--

---

### Description

Calculates Genes that are near to the SV region

### Usage

```
nonOverlapGenes.bed, chrom, startpos, endpos, svid, n = 3)
```

### Arguments

bed	Text Bionano Bed file.
chrom	character SVmap chromosome.
startpos	numeric starting position of the breakpoints.
endpos	numeric end position of the breakpoints.
svid	numeric Structural variant identifier (Bionano generated).
n	numeric Number of genes to report which are nearest to the breakpoint. Default is 3.

### Value

Data Frame. Contains the SVID, Gene name, strand information and Distance from the SV covered.

### Examples

```
smapName="F1.1_TestSample1_solo_hg19.smap"
smap = system.file("extdata", smapName, package="nanotator")
bedFile <- system.file("extdata", "Homo_sapiens.Hg19.bed",
package="nanotator")
bed<-buildrunBNBedFiles(bedFile,returnMethod="dataFrame")
smap<-readSMap(smap)
chrom<-smap$RefcontigID1
startpos<-smap$RefStartPos
endpos<-smap$RefEndPos
if (length(grep("SVIndex",names(smap)))>0){
  svid <- smap$SVIndex
}else{
  svid <- smap$SmapEntryID
}
n<-3
nonOverlapGenes.bed, chrom, startpos, endpos, svid,n)
```

---

omim_gene	<i>Extracting genes from OMIM database NCBI.</i>
-----------	--

---

**Description**

Extracting genes from OMIM database NCBI.

**Usage**

```
omim_gene(terms)
```

**Arguments**

terms	Single or Multiple Terms.
-------	---------------------------

**Value**

Dataframe returned containing gene lists in entrezid and Gene Symbols, and terms associated with it

**Examples**

```
terms="Muscle Weakness"
ge <- omim_gene(terms)
```

---

overlapGenes	<i>Calculates Genes that overlap the SV region</i>
--------------	--

---

**Description**

Calculates Genes that overlap the SV region

**Usage**

```
overlapGenes.bed, chrom, startpos, endpos, svid)
```

**Arguments**

bed	Text Bionano Bed file.
chrom	character SVmap chromosome.
startpos	numeric starting position of the breakpoints.
endpos	numeric end position of the breakpoints.
svid	numeric Structural variant identifier (Bionano generated).

**Value**

Data Frame. Contains the SVID, Gene name, strand information and percentage of SV covered.



## Examples

```
smapName="F1.1_TestSample1_solo_hg19.smap"
smap = system.file("extdata", smapName, package="nanotatoR")
bedFile <- system.file("extdata", "Homo_sapiens.Hg19.bed",
  package="nanotatoR")
bed<-buildrunBNBedFiles(bedFile,returnMethod="dataFrame")
smap<-readSMap(smap)
chrom<-smap$RefcontigID1
startpos<-smap$RefStartPos
endpos<-smap$RefEndPos
if (length(grep("SVIndex",names(smap)))>0){
  svid <- smap$SVIndex
}else{
  svid <- smap$SmapEntryID
}
overlapGenes(bed, chrom, startpos, endpos, svid)
```

---

readBNBedFiles	<i>Reads Bionano Bedfiles</i>
----------------	-------------------------------

---

## Description

Reads Bionano Bedfiles

## Usage

```
readBNBedFiles(BNFile)
```

## Arguments

BNFile            character. Path to Bionano Bed File.

## Value

Data Frame Contains the gene information.

## Examples

```
BNFile <- system.file("extdata", "Homo_sapiens.Hg19_BN_bed.txt",
  package="nanotatoR")
bed<-readBNBedFiles(BNFile)
```

---

readSMap	<i>Reads SMAP files to extract information</i>
----------	--

---

**Description**

Reads SMAP files to extract information

**Usage**

```
readSMap(smap)
```

**Arguments**

smap                    character. Path to SMAP file.

**Value**

Data Frame or text file. Contains the SMAP information.

**Examples**

```
smapName="F1.1_TestSample1_solo_hg19.smap"
smap = system.file("extdata", smapName, package="nanotator")
readSMap(smap)
```

---

run_bionano_filter	<i>Getting the data from annotated smaps to extract SV information based on type of variants.</i>
--------------------	---

---

**Description**

Getting the data from annotated smaps to extract SV information based on type of variants.

**Usage**

```
run_bionano_filter(input_fmt_geneList = c("Text", "dataFrame"),
  input_fmt_svMap = c("Text", "dataFrame"), SVFile = NULL, svData,
  dat_geneList, fileName, outpath, outputFilename = "", RZIPpath)
```

**Arguments**

input\_fmt\_geneList                    character. Choice of gene list input Text or Dataframe.

input\_fmt\_svMap                        character. Choice of gene list input Text or Dataframe.

SVFile                                 character. SV file name.

svData                                 Dataframe Input data containing SV data.

dat\_geneList                          Dataframe Input data containing geneList data.

fileName                                Character Name of file containing Gene List data.

outpath           Character Directory to the output file.  
outputFilename   Character Output filename.  
RZIPpath          Character Path for the Rtools Zip package.

**Value**

Excel file containing the annotated SV map, tabs divided based on type of SVs.

**Examples**

```
## Not run:  
terms <- "Muscle Weakness"  
gene <- gene_list_generation(  
  method = "Single", term = terms,  
  returnMethod_GeneList = "dataFrame"  
)  
RzipFile = "zip.exe"  
RZIPpath <- system.file("extdata", RzipFile, package = "nanotatoR")  
smapName <- "F1.1_TestSample1_solo_hg19.smap"  
smappath <- system.file("extdata", smapName, package = "nanotatoR")  
smappath1 <- system.file("extdata", package = "nanotatoR")  
run_bionano_filter(input_fmt_geneList = c("dataframe"), input_fmt_svMap = c("Text"),  
  SVFile = smappath, dat_geneList = gene, outpath = smappath1, outputFilename = "test",  
  RZIPpath = RZIPpath)  
  
## End(Not run)
```

# Index

[buildrunBNBedFiles](#), 2

[clinvar\\_gene](#), 3

[cohortFrequency](#), 4

[compSmapped](#), 5

[Decipher\\_extraction](#), 6

[DGV\\_extraction](#), 7

[gene\\_extraction](#), 8

[gene\\_list\\_generation](#), 8

[gtr\\_gene](#), 9

[internalFrequency](#), 9

[makeMergedSmappedData](#), 11

[makeMergedSVData](#), 11

[nanotatoR](#), 12

[nanotatoR-package \(nanotatoR\)](#), 12

[nanotatoR\\_main](#), 12

[nonOverlapGenes](#), 15

[omim\\_gene](#), 16

[overlapGenes](#), 16

[readBNBedFiles](#), 17

[readSMap](#), 18

[run\\_bionano\\_filter](#), 18