

# Package ‘nanotatoR’

April 12, 2022

**Title** Next generation structural variant annotation and classification

**Version** 1.10.0

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

## Description

Whole genome sequencing (WGS) has successfully been used to identify single-nucleotide variants (SNV), small insertions and deletions (INDELs) and, more recently, small copy number variants (CNVs). However, due to utilization of short reads, it is not well suited for identification of structural variants (SV). Optical mapping (OM) from Bionano Genomics, utilizes long fluorescently labeled megabase size DNA molecules for de novo genome assembly and identification of SVs with a much higher sensitivity than WGS. Nevertheless, currently available SV annotation tools have limited number of functions. NanotatoR is an R package written to provide a set of annotations for SVs identified by OM. It uses Database of Genomic Variants (DGV), Database of Chromosomal Imbalance and Phenotype in Humans Using Ensembl Resources (DECIPHER) as well as a subset (154 samples) of 1000 Genome Project to calculate the population frequencies of the SVs (an optional internal cohort SV frequency calculation is also available). NanotatoR creates a primary gene list (PG) from NCBI databases based on proband’s phenotype specific keywords and compares the list to the set of genes overlapping/near SVs. The output is given in an Excel file format, which is subdivided into multiple sheets based on SV type (e.g., INDELs, Inversions, Translocations). Users then have a choice to filter SVs using the provided annotations for de novo (if parental samples are available) or inherited rare variants.

**Depends** R (>= 4.1),

**Imports** hash(>= 2.2.6), openxlsx(>= 4.0.17), rentrez(>= 1.1.0), stats, rlang, stringr, knitr, testthat, utils, AnnotationDbi, httr, GenomicRanges, tidyverse, VarfromPDB, org.Hs.eg.db, curl, dplyr, XML, XML2R

**Suggests** rmarkdown, yaml

**VignetteBuilder** knitr

**License** file LICENSE

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

**Encoding** UTF-8

**RoxygenNote** 7.1.1.9001

**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\_14

**git\_last\_commit** 821a44d

**git\_last\_commit\_date** 2021-10-26

**Date/Publication** 2022-04-12

**Maintainer** Surajit Bhattacharya <[sbhattach2@childrensnational.org](mailto:sbhattach2@childrensnational.org)>

## R topics documented:

BNDBfrequency . . . . .	3
buildrunBNBedFiles . . . . .	5
clinvar_gene . . . . .	6
Decipherfrequency . . . . .	6
DGVfrequency . . . . .	8
extract_clinvar_mod . . . . .	9
FamilyInfoPrep . . . . .	10
gene_extraction . . . . .	11
gene_list_generation . . . . .	11
gtr_gene . . . . .	13
internalFrequencyTrio_Duo . . . . .	13
internalFrequency_solo . . . . .	16
makeInternalBNDatabase . . . . .	19
mergingSMAP_SE . . . . .	20
mergingSMAP_SVMerge . . . . .	20
merging_SE_SVMerge . . . . .	21
nanotatoR . . . . .	22
nanotatoR_Duo_SVmerge . . . . .	23
nanotatoR_main_Duo_SE . . . . .	28
nanotatoR_main_Solo_SE . . . . .	33
nanotatoR_main_Solo_SVmerge . . . . .	38
nanotatoR_main_Trio_SE . . . . .	43
nanotatoR_SVmerge_Trio . . . . .	47
nonOverlapGenes . . . . .	52
nonOverlappingDNGenes . . . . .	54
nonOverlappingUPGenes . . . . .	55
nonOverlapRNAseq . . . . .	55
nonOverlapRNAseq_solo . . . . .	56
omim_gene . . . . .	57
overlapGenes . . . . .	58
overlapnearestgeneSearch . . . . .	59
overlappingGenes . . . . .	60
OverlapRNAseq . . . . .	61
OverlapRNAseq_solo . . . . .	62

phenoextractHPO_mod . . . . .	63
readBNBedFiles . . . . .	63
reading_GTR . . . . .	64
reading_mim2gene . . . . .	65
readSMap . . . . .	65
readSMap_DLE . . . . .	66
RNAseqcombine . . . . .	66
RNAseqcombine_solo . . . . .	67
run_bionano_filter_SE_duo . . . . .	68
run_bionano_filter_SE_solo . . . . .	69
run_bionano_filter_SE_Trio . . . . .	72
run_bionano_filter_SVMerge_duo . . . . .	74
run_bionano_filter_SVMerge_solo . . . . .	76
run_bionano_filter_SVMerge_Trio . . . . .	78
SVexpression_duo_trio . . . . .	80
SVexpression_solo . . . . .	81

**Index****84**


---

BNDBfrequency	<i>Calculates the internal frequencies of BNDB cohorts</i>
---------------	--

---

**Description**

Calculates the internal frequencies of BNDB cohorts

**Usage**

```
BNDBfrequency(
  internalBNDB,
  smappath,
  smap,
  buildBNInternalDB = FALSE,
  smapdata,
  input_fmt_SV = c("Text", "dataFrame"),
  dbOutput = c("dataframe", "text"),
  BNDBpath,
  BNDBpattern,
  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"),
  EnzymeType = c("SVmerge", "SE")
)
```

**Arguments**

internalBNDB	character. Path to the merged SV files.
smappath	character. path to the query smap file.
smap	character. File name for the smap
buildBNInternalDB	boolean. Checking whether the merged BNDB file database exist.
smapdata	dataframe. smapdata in the form of dataframe.
input_fmt_SV	character. Choice between Text and DataFrame.
dbOutput	character. database output type. Options dataframe or text.
BNDBpath	character. Path to the BNDB file database.
BNDBpattern	character. pattern of the file names to merge.
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. SV size limit.
transconf	Numeric. Threshold for translocation Score.
returnMethod	character. Choice between Text and DataFrame.
EnzymeType	Character. Type of enzyme. Options SVmerge and SE.

**Value**

Text file or data frames containing internalFrequency data.

**Examples**

```

path <- system.file("extdata", "Bionano_config/", package = "nanotatoR")
pattern <- "*_hg19_*"
smapName="GM24385_Ason_DLE1_VAP_trio5.smap"
smap = system.file("extdata", smapName, package="nanotatoR")
BNDBfrequency(smap = smap,
  buildBNInternalDB=TRUE,
  input_fmt_SV = "Text",
  dbOutput="dataframe",
  BNDBpath = path,
  BNDBpattern = pattern,
  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("dataFrame"),  
    EnzymeType = c("SE"))
```

---

buildrunBNBedFiles      *Reads BED files to produce bionano Bed files*

---

## Description

Reads BED files to produce bionano Bed files

## Usage

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

## Arguments

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

## Value

Data Frame or text file. Contains the gene information.

## Examples

```
bedFile <- system.file("extdata", "HomoSapienGRCH19_lift37.bed",  
  package="nanotator")  
bed<-buildrunBNBedFiles(bedFile,returnMethod="dataFrame")
```

---

clinvar_gene	<i>Extracting genes from clinvar database NCBI.</i>
--------------	---

---

### Description

Extracting genes from clinvar database NCBI.

### Usage

```
clinvar_gene(terms, clinvar, downloadClinvar, omimID = NULL)
```

### Arguments

terms	Single or Multiple Terms.
clinvar	character clinvar database location.
downloadClinvar	boolean If TRUE, download the gtr database. Default FALSE.
omimID	numeric Omim Id for disease.

### Value

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

### Examples

```
terms="Liver cirrhosis"
clinvar = system.file("extdata", "localPDB/", package="nanotator")
downloadClinvar = FALSE
ge <- clinvar_gene(terms = terms, clinvar = clinvar,
downloadClinvar = downloadClinvar,
omimID = "OMIM:118980")
```

---

Decipherfrequency	<i>Frequency calculation of variants compared to Decipher.</i>
-------------------	--

---

### Description

Frequency calculation of variants compared to Decipher.

**Usage**

```
Decipherfrequency(
  decipherpath,
  smap,
  smap_data,
  win_indel = 10000,
  perc_similarity = 0.5,
  returnMethod = c("Text", "dataFrame"),
  input_fmt_SV = c("Text", "dataFrame"),
  EnzymeType = c("SVMerge", "SE"),
  outpath
)
```

**Arguments**

decipherpath	character. Decipher Text file.
smap	character Filepath for smap.
smap_data	Dataset containing smap data.
win_indel	character indel window. Default 10000.
perc_similarity	Numeric . ThresholdPercentage similarity of the query SV and reference SV.
returnMethod	character. Choice between text or data frame as the output.
input_fmt_SV	boolean . Options SE and SVMerge.
EnzymeType	boolean . Options SE and SVMerge.
outpath	character. Path where gene lists are saved.

**Value**

dataframe containing decipher data. are stored as text files.

**Examples**

```
decipherpath = system.file("extdata", "population_cnv.txt",
  package="nanotatoR")
smappath=system.file("extdata", "GM24385_Ason_DLE1_VAP_trio5.smap",
  package="nanotatoR")
datdecipher <- Decipherfrequency (decipherpath = decipherpath,
  smap = smappath, win_indel = 10000,
  EnzymeType= "SE",
  perc_similarity = 0.5,returnMethod="dataFrame",
  input_fmt_SV = "Text")
datdecipher[1,]
```

---

DGVfrequency

*Frequency calculation of variants compared to DGV.*


---

### Description

Frequency calculation of variants compared to DGV.

### Usage

```
DGVfrequency(
  hgpath,
  smap,
  smap_data,
  win_indel_DGV = 10000,
  win_inv_trans_DGV = 50000,
  perc_similarity_DGV = 0.5,
  input_fmt_SV = c("Text", "dataframe"),
  returnMethod = c("Text", "dataFrame"),
  outpath,
  EnzymeType = c("SVMerge", "SE")
)
```

### Arguments

hgpath	character. Path to Database of Genomic Variants (DGV) Text file.
smap	character. File name for smap textfile.
smap_data	dataframe. Dataset containing smap data.
win_indel_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.
input_fmt_SV	boolean . Options Text and dataframe.
returnMethod	character. Choice between text or data frame as the output.
outpath	character. Path where gene lists are saved.
EnzymeType	boolean . Options SE and SVMerge.

### Value

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



**Examples**

```

hgpath=system.file("extdata", "GRCh37_hg19_variants_2016-05-15.txt", package="nanotatoR")
smappath=system.file("extdata", "GM24385_Ason_DLE1_VAP_trio5.smmap", package="nanotatoR")
datDGV <- DGVfrequency (hgpath = hgpath,
smap = smappath,
win_indel_DGV = 10000,
EnzymeType = "SE",
input_fmt_SV = "Text",
perc_similarity_DGV = 0.5,returnMethod="dataFrame")

```

---

extract_clinvar_mod	<i>Extract the genes and variants related to a genetic disorder from ClinVar</i>
---------------------	--

---

**Description**

Extract the genes and variants related to a genetic disorder from ClinVar

**Usage**

```

extract_clinvar_mod(
  keyword,
  localPDB.path,
  type = "both",
  HPO.disease = NULL,
  genelist = NULL,
  OMIM = NULL
)

```

**Arguments**

keyword	character. character string: keyword, to search a disease, a clinical feature, or a phenotype.
localPDB.path	character. the path of localized public data bases. The default value is set in the working directory.
type	character. the type of the information to extract, must be one of "gene", "variant", "both"(default).
HPO.disease	character. MIM number of the disease. The default value is NULL, which means that all the OMIM number of the disease in HPO are added. localized public data bases. The default value is set in the working directory.
genelist	character. the gene(s) associated to the disease, or the genes you are interested.
OMIM	character. whether use the information from OMIM database. The default value is NULL. It can be set 'yes' when you make sue you have a OMIM API key.

**Value**

subset of the file `gene_condition_source_id`, which include all the information about genes and phenotypes in ClinVar and subset of the file `variant_summary.txt`, but added several columns which describe the phenotype from GeneReview, MedGen, and OMIM databases. Function modified from `extract_clinvar` function `VarFromPDB`.

**Examples**

```
keyword = "retinoblastoma"
extract_clinvar_mod(keyword,
  localPDB.path = system.file("extdata", "localPDB", package="nanotatoR"),
  type = "both", HPO.disease = NULL,
  genelist = NULL, OMIM = NULL)
```

---

FamilyInfoPrep

*Mapping Relationship to unique nanoIDs*

---

**Description**

Mapping Relationship to unique nanoIDs

**Usage**

```
FamilyInfoPrep(
  Samplecodes = "X:/Hayks_Materials/BNG/Projects/nanotatoR_sample_codes.csv",
  mergeKey = "X:/Hayks_Materials/BNG/Projects/MergeKey.csv",
  outMode = c("Text", "dataframe"),
  outpath = "X:/Hayks_Materials/BNG/Projects/VAP_DLE1_solo_SMAPs/Merged"
)
```

**Arguments**

<code>Samplecodes</code>	character. File containing relations and IDs associated to them.
<code>mergeKey</code>	character. File containing sample ID and relation.
<code>outMode</code>	character. The output mode. Choices, dataframe or Text.
<code>outpath</code>	character. Path where the dual labelled merged samples are kept. Is mandatory if <code>outMode</code> is Text.

**Value**

Text files containing merged smaps from different samples

**Examples**

```
FamilyInfoPrep(
  Samplecodes = system.file("extdata", "nanotatoR_sample_codes.csv", package="nanotatoR"),
  mergeKey = system.file("extdata", "nanotatoR_control_sample_codes.csv", package="nanotatoR"),
  outMode = c("dataframe"))
```

---

gene_extraction	<i>Extracting genes from gene database NCBI.</i>
-----------------	--

---

**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="Liver cirrhosis"  
ge <- gene_extraction(terms)
```

---

gene_list_generation	<i>Extracting genes for phenotype/diseases from NCBI.</i>
----------------------	---

---

**Description**

Extracting genes for phenotype/diseases from NCBI.

**Usage**

```
gene_list_generation(  
  method_entrez = c("Single", "Multiple", "Text"),  
  termPath,  
  omimID = NULL,  
  term,  
  outpath,  
  thresh = 5,  
  returnMethod = c("Text", "dataFrame"),  
  omim,  
  clinvar,  
  gtr,  
  removeClinvar = FALSE,
```

```

removeGTR = FALSE,
downloadClinvar = FALSE,
downloadGTR = FALSE,
url_gtr
)

```

### Arguments

method_entrez	character. Input Method for terms. Choices are "Single", "Multiple" and "Text".
termPath	character. Path and file name for textfile. FileName should be in the following format "SampleID_Keywords.csv".
omimID	numeric. mimID for disease. Default is NULL.
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.
omim	character. omim2gene file name and location.
clinvar	character. clinvar file name and location.
gtr	character. gtr file name and location.
removeClinvar	logical. Deletes the Clinvar database if TRUE.
removeGTR	logical. Deletes the GTR database if TRUE.
downloadClinvar	logical. Downloads the Clinvar database if TRUE.
downloadGTR	logical. Downloads the GTR database if TRUE.
url_gtr	character. url for GTR.

### Value

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

### Examples

```

terms="CIRRHOSIS, FAMILIAL"
genes <- gene_list_generation(
  method_entrez = c("Single"),
  term = terms,
  returnMethod=c("dataFrame"),
  omimID = "OMIM:118980",
  omim = system.file("extdata", "mim2gene.txt", package="nanotatoR"),
  clinvar = system.file("extdata", "localPDB/", package="nanotatoR"),
  gtr = system.file("extdata", "gtrDatabase.txt", package="nanotatoR"),
  downloadClinvar = FALSE, downloadGTR = FALSE)

```

---

gtr_gene	<i>Extracting genes from gtr database NCBI.</i>
----------	---

---

**Description**

Extracting genes from gtr database NCBI.

**Usage**

```
gtr_gene(terms, gtr, url_gtr, downloadGTR = TRUE)
```

**Arguments**

terms	Single or Multiple Terms.
gtr	character gtr database location.
url_gtr	character url for gtr database.
downloadGTR	boolean If TRUE, download the gtr database. Default FALSE.

**Value**

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

**Examples**

```
terms="Liver cirrhosis"
gtr = system.file("extdata", "gtrDatabase.txt", package="nanotatoR")
ge <- gtr_gene(terms = terms,gtr = gtr, downloadGTR = FALSE)
```

---

internalFrequencyTrio_Duo	<i>Calculates the internal frequencies of SV in internal cohorts, for SVMerge</i>
---------------------------	---

---

**Description**

Calculates the internal frequencies of SV in internal cohorts, for SVMerge

**Usage**

```

internalFrequencyTrio_Duo(
  mergedFiles,
  smappath,
  smap,
  buildSVInternalDB = FALSE,
  smapdata,
  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"),
  perc_similarity_parents = 0.9,
  returnMethod = c("Text", "dataFrame"),
  input_fmt_SV = c("Text", "dataFrame"),
  indexfile,
  EnzymeType = c("SVmerge", "SE"),
  labelType = c("SVMerge", "SE", "Both"),
  SVMerge_path,
  SVMerge_pattern,
  SE_path,
  SE_pattern,
  Samplecodes,
  mergeKey,
  mergedKeyoutpath,
  mergedKeyFname
)

```

**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.
path	character. Path to the solo file database.
pattern	character. pattern of the file names to merge.

outpath	character. Path where the merged samples are kept.
win_indel	N umeric. 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 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.
transconf	Numeric. Threshold for translocation confidence. Default 0.1.
dbOutput	character. Output of merged bionano data.
perc_similarity_parents	Numeric . ThresholdPercentage similarity for zygoty determination. Default 0.9.
returnMethod	character. Choice between Text and DataFrame. Required if you want to calculate internal frequency.
input_fmt_SV	Format in which data is provided as an input to thefunction.
indexfile	File containing connection between sample and nanoIDs
EnzymeType	Character. Type of enzyme. Options Dual and DLE.
labelType	character. Type of labels used for mapping. Choices are Dual, DLE and Both.
SVMerge_path	character. Path for the Dual labelled cmap
SVMerge_pattern	character. pattern of the dual files.
SE_path	character. Path for the Dual labelled cmap
SE_pattern	character. pattern of the dual files.
Samplecodes	character. File containing relations and IDs associated to them.
mergeKey	character. File containing sample ID and relation.
mergedKeyoutputpath	character. File path storing sample name and nanoID key information.
mergedKeyFname	character. File name storing sample name and nanoID key information.

**Value**

Calculated internal frequency in dataframe or text.

**Examples**

```

smapName="GM24385_Ason_DLE1_VAP_trio5.smap"
smap = system.file("extdata", smapName, package="nanotatoR")
indelconf = 0.5; invconf = 0.01;transconf = 0.1;input_fmt_SV="Text";
datInf <- internalFrequency_solo( smap = smap,
buildSVInternalDB = FALSE, win_indel=10000,
win_inv_trans=50000, EnzymeType = "SE",
mergedFiles = system.file("extdata", "nanotatoRControl.txt", package="nanotatoR"),
perc_similarity_parents =0.9,
indexfile = system.file("extdata", "Sample_index.csv", package="nanotatoR"),
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", input_fmt_SV = "Text")

```

---

internalFrequency\_solo

*Calculates the internal frequencies of SV in internal cohorts, for SE*

---

**Description**

Calculates the internal frequencies of SV in internal cohorts, for SE

**Usage**

```

internalFrequency_solo(
  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"),

```



```

input_fmt_SV = c("Text", "dataFrame"),
indexfile,
perc_similarity_parents = 0.9,
EnzymeType = c("SVmerge", "SE"),
labelType = c("SVMerge", "SE", "Both"),
SVMerge_path,
SVMerge_pattern,
SE_path,
SE_pattern,
Samplecodes,
mergeKey,
mergedKeyoutpath,
mergedKeyFname
)

```

### 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 where the merged samples are kept.
win_indel	N umeric. 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 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.
transconf	Numeric. Threshold for translocation confidence. Default 0.1.

dbOutput	character. Output of merged bionano data.
returnMethod	character. Choice between Text and DataFrame. Required if you want to calculate internal frequency.
input_fmt_SV	character. Choice between Text and DataFrame.
indexfile	File containing connection between sample and nanoIDs
perc_similarity_parents	Numeric . ThresholdPercentage similarity for zygosity determination. Default 0.9.
EnzymeType	Character. Type of enzyme. Options Dual and DLE.
labelType	character. Type of labels used for mapping. Choices are Dual, DLE and Both.
SVMerge_path	character. Path for the Dual labelled cmap
SVMerge_pattern	character. pattern of the dual files.
SE_path	character. Path for the Dual labelled cmap
SE_pattern	character. pattern of the dual files.
Samplecodes	character. File containing relations and IDs associated to them.
mergeKey	character. File containing sample ID and relation.
mergedKeyoutpath	character. File path storing sample name and nanoID key information.
mergedKeyFname	character. File name storing sample name and nanoID key information.

**Value**

Calculated internal frequency in dataframe or text.

**Examples**

```

smapName = "NA12878_DLE1_VAP_solo5.smap"
smap = system.file("extdata", smapName, package="nanotatoR")
indelconf = 0.5; invconf = 0.01;transconf = 0.1;input_fmt="Text";
datInf <- internalFrequency_solo( smap = smap,
buildSVInternalDB = FALSE, win_indel=10000,
win_inv_trans=50000, EnzymeType = "SE",
mergedFiles = system.file("extdata", "nanotatoRControl.txt", package="nanotatoR"),
perc_similarity_parents =0.9,
indexfile = system.file("extdata", "Sample_index.csv", package="nanotatoR"),
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", input_fmt_SV = "Text")

```

---

`makeInternalBNDDatabase`*Merges Solo SV files to one common SV file*

---

## Description

Merges Solo SV files to one common SV file

## Usage

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

## Arguments

<code>path</code>	character. Path to the solo files.
<code>pattern</code>	character. file name pattern for solo files.
<code>outpath</code>	character. file path for the output file.
<code>fname</code>	character. file name for the output file.
<code>dbOutput</code>	character. Output option database or text.

## Value

Text file containing all the solo SMAP files.

## Examples

```
path <- system.file("extdata", "Bionano_config/", package = "nanotatoR")  
pattern <- "_hg19.txt"  
mergedSmap <- makeInternalBNDDatabase(path = path,  
  pattern = pattern, dbOutput = "dataframe")  
mergedSmap[1,]
```

---

mergingSMAP\_SE      *Merging DLE labelled smaps*

---

**Description**

Merging DLE labelled smaps

**Usage**

```
mergingSMAP_SE(path, pattern, outMode = c("Text", "dataframe"), outpath)
```

**Arguments**

path	character. Path to the solo files directory.
pattern	character. Pattern for the solo files.
outMode	character. The output mode. Choices, dataframe or Text.
outpath	character. Path where the dual labelled merged samples are kept. Is mandatory if outMode is Text.

**Value**

Text files containing merged smaps from different samples

**Examples**

```
mergedSmap <- mergingSMAP_SE (  
  path = system.file("extdata", "SoloFile/", package="nanotatoR"),  
  pattern = "*_DLE1_*", outMode = "dataframe",  
  outpath = system.file("extdata", "Merged/", package="nanotatoR"))
```

---

mergingSMAP\_SVMerge      *Merging dual labelled smaps*

---

**Description**

Merging dual labelled smaps

**Usage**

```
mergingSMAP_SVMerge(path, pattern, outMode = c("dataframe", "Text"), outpath)
```

**Arguments**

path	character. Path to the solo files directory.
pattern	character. Patternna for the solo files.
outMode	character. The ouput mode. Choices, dataframe or Text.
outpath	character. Path where the dual labelled merged samples are kept. Is mandatory if outMode is Text.

**Value**

Text files containg merged smaps from different samples

**Examples**

```
a <- mergingSMAP_SVMerge(
  path = system.file("extdata", "SoloFile/", package="nanotatoR"),
  pattern = "*.txt", outMode = "dataframe",
  outpath = system.file("extdata", "SoloFile/", package="nanotatoR"))
```

---

merging\_SE\_SVMerge      *Merging Dual and DLE, and adding nanotatoR relation ID*

---

**Description**

Merging Dual and DLE, and adding nanotatoR relation ID

**Usage**

```
merging_SE_SVMerge(
  labelType = c("SVMerge", "SE", "Both", "SE_Cancer"),
  SVMerge_path,
  SVMerge_pattern,
  SE_path,
  SE_pattern,
  Samplecodes,
  mergeKey,
  outpath,
  mergedKeyoutpath,
  mergedKeyfname,
  filename,
  outputMode = c("dataframe", "Text")
)
```

**Arguments**

labelType	character. Type of labels used for mapping. Choices are Dual, DLE and Both.
SVMerge_path	character. Path for the Dual labelled cmap
SVMerge_pattern	character. pattern of the dual files.
SE_path	character. Path for the Dual labelled cmap
SE_pattern	character. pattern of the dual files.
Samplecodes	character. File containing relations and IDs associated to them.
mergeKey	character. File containing sample ID and relation.
outpath	character. Path where the merged samples are kept.
mergedKeyoutpath	character. File path storing sample name and nanoID key information.
mergedKeyFname	character. File name storing sample name and nanoID key information.
filename	character. Output file name.
outputMode	character. Mode of databse output. Text or dataframe.

**Value**

Text files containg merged smaps from different samples

**Examples**

```
dat1 <- merging_SE_SVMerge (
  labelType = c("SE"),
  SE_path = system.file("extdata", "SoloFile/", package="nanotatoR"),
  SE_pattern = "*_DLE1_*",
  Samplecodes = system.file("extdata", "nanotatoR_sample_codes.csv", package="nanotatoR"),
  mergeKey = system.file("extdata", "nanotatoR_control_sample_codes.csv", package="nanotatoR"),
  outpath = system.file("extdata", package="nanotatoR"),
  mergedKeyoutpath = system.file("extdata", package="nanotatoR"),
  mergedKeyFname = "Sample_index.csv",
  filename= "nanotatoRControl.txt",
  outputMode = "dataframe")
```

---

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 <- makeInternalBNDatabase(path = path,
  pattern = pattern, dbOutput = "dataframe")
mergedSmap[1,]
```

---

nanotatoR\_Duo\_SVmerge *Annotation and visualisation of Bionano SV, of SVMerge Duo samples.*

---

## Description

Annotation and visualisation of Bionano SV, of SVMerge Duo samples.

## Usage

```
nanotatoR_Duo_SVmerge(  
  smap,  
  bed,  
  inputfmtBed = c("bed", "BNBed"),  
  n = 3,  
  buildBNInternalDB = TRUE,  
  mergedFiles,  
  smappath,  
  buildSVInternalDB = FALSE,  
  path,  
  pattern,  
  win_indel_INF = 10000,  
  win_inv_trans_INF = 50000,  
  perc_similarity_INF = 0.5,  
  indelconf = 0.5,  
  invconf = 0.01,  
  transconf = 0.1,  
  perc_similarity_INF_parents = 0.9,  
  hgpath,  
  win_indel_DGV = 10000,  
  win_inv_trans_DGV = 50000,  
  perc_similarity_DGV = 0.5,  
  method_entrez = c("Single", "Multiple", "Text"),  
  termPath,  
  term,  
  thresh = 5,  
  limsize = 1000,  
  EnzymeType = c("SVmerge", "SE"),  
  labelType = c("SVMerge", "SE", "Both"),  
  SVMerge_path,  
  SVMerge_pattern,
```

```

SE_path,
SE_pattern,
Samplecodes,
mergeKey,
mergedKeyoutpath,
mergedKeyFname,
RNAseqcombo = TRUE,
RNASeqDir,
returnMethod = "dataFrame",
RNASeqData,
RNASeqPATH,
pattern_Proband = NA,
pattern_Mother = NA,
pattern_Father = NA,
outpath,
outputFilename = "",
termListPresent = TRUE,
internalBNDB,
clinvar,
InternaldatabasePresent = TRUE,
RNASeqDatasetPresent = TRUE,
geneListPresent = TRUE,
omim,
gtr,
removeClinvar = FALSE,
removeGTR = FALSE,
downloadClinvar = FALSE,
downloadGTR = FALSE,
url_gtr,
omimID,
RZIPpath,
directoryName,
fileprefix,
datGeneListPath,
decipherpath,
indexfile,
primaryGenesPresent = TRUE,
outputType = c("Excel", "csv")
)

```

### Arguments

smap	character. File name for the smap
bed	Text Bionano Bed file.
inputfmtBed	character Whether the bed input is UCSC bed or Bionano bed.
n	numeric Number of genes to report which are nearest to the breakpoint. Default is 3.



buildBNInternalDB	boolean. Checking whether the merged BNDB file database exist.
mergedFiles	character. Path to the merged SV files.
smappath	character. Path and file name for textfile.
buildSVInternalDB	boolean. Checking whether the merged solo file database exist.
path	character. Path to the solo file database.
pattern	character. pattern of the file names to merge.
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.
perc_similarity_INF_parents	Numeric . ThresholdPercentage similarity for parent zygoty calculation. Default threshold 0.9.
hgpath	character. Path to Database of Genomic Variants (DGV) Text file.
win_indel_DGV	Numeric. Insertion and deletion error window for DGV.
win_inv_trans_DGV	Numeric. Inversion and translocation error window for DGV.
perc_similarity_DGV	Numeric . ThresholdPercentage similarity of the query SV and reference SV, for DGV..
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.
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.
limsize	Numeric. Minimum size for SV. Default 1000.
EnzymeType	Character. Type of enzyme. Options Dual and DLE.
labelType	character. Type of labels used for mapping. Choices are Dual, DLE and Both.
SVMerge_path	character. Path for the Dual labelled cmap
SVMerge_pattern	character. pattern of the dual files.
SE_path	character. Path for the Dual labelled cmap
SE_pattern	character. pattern of the dual files.
Samplecodes	character. File containing relations and IDs associated to them.
mergeKey	character. File containing sample ID and relation.

mergedKeyoutputpath	character. File path storing sample name and nanoID key information.
mergedKeyFname	character. File name storing sample name and nanoID key information.
RNAseqcombo	boolean whether RNASeq datasets are combined or not.
RNASeqDir	boolean Directory for RNASeq.
returnMethod	character. Choice between text or data frame as the output.
RNASeqData	dataFrame. RNAseq data with gene names.
RNASeqPATH	character. RNAseq dataset path .
pattern_Proband	character. Pattern for proband.
pattern_Mother	character. Pattern to identify the mother reads.
pattern_Father	character. Pattern to identify the father reads.
outputpath	Character Directory to the output file.
outputFilename	Character Output filename.
termListPresent	logical Checks whether term list is provided by the user.
internalBNDB	character. internak Bionano merged databse.
clinvar	character. clinvar file name and location.
InternaldatabasePresent	boolean. Checking whether internal DB present.
RNASeqDatasetPresent	boolean. Checking whether RNASeq database present or not.
geneListPresent	logical Checks whether gene list is provided by the user.
omim	character. omim2gene file name and location.
gtr	character. gtr file name and location.
removeClinvar	logical. Deletes the Clinvar database if TRUE.
removeGTR	logical. Deletes the GTR database if TRUE.
downloadClinvar	logical. Downloads the Clinvar database if TRUE.
downloadGTR	logical. Downloads the GTR database if TRUE.
url_gtr	character. url for GTR.
omimID	character. Omim ID.
RZIPpath	character. Path to RZippath.
directoryName	Directory name where individual SV files will be stored.
fileprefix	character Prefix to use for each of the files in the directory.
datGeneListPath	Character Path for genelist.
decipherpath	character. Decipher database path.
indexfile	character. indexfile containing nano ID and sample relation.
primaryGenesPresent	logical Checks whether primarygene list is provided by the user.
outputType	Variants in excel tabs or in different csv files. Options Excel or csv.

**Value**

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

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

**Examples**

```
## Not run:
smapName="NA12878_DLE1_VAP_solo5.smap"
smap = system.file("extdata", smapName, package="nanotatoR")
bedFile <- system.file("extdata", "HomoSapienGRCH19_lift37.bed", package="nanotatoR")
hgpath=system.file("extdata", "GRCh37_hg19_variants_2016-05-15.txt", package="nanotatoR")
decipherpath = system.file("extdata", "population_cnv.txt", package="nanotatoR")
omim = system.file("extdata", "mim2gene.txt", package="nanotatoR")
clinvar = system.file("extdata", "localPDB/", package="nanotatoR")
gtr = system.file("extdata", "gtrDatabase.txt", package="nanotatoR")
labelType = c("SE")
SE_path = system.file("extdata", "SoloFile/", package="nanotatoR")
SE_pattern = "*_DLE1_*"
Samplecodes = system.file("extdata", "nanotatoR_sample_codes.csv", package="nanotatoR")
mergeKey = system.file("extdata", "nanotatoR_control_sample_codes.csv", package="nanotatoR")
mergedKeyoutpath = system.file("extdata", package="nanotatoR")
indexfile = "Sample_index.csv"
RNASeqDir = system.file("extdata", "NA12878_P_Blood_S1.genes.results", package="nanotatoR")
path = system.file("extdata", "Bionano_config/", package = "nanotatoR")
pattern = "_hg19.txt"
outputFilename <- "GM24385_DLE-1_P_trio_hg19_out"
outpath <- system.file("extdata", smapName, package = "nanotatoR")
RZIPpath <- system.file("extdata", "zip.exe", package = "nanotatoR")
nanotatoR_Duo_SVmerge(
  smap = smap, bed = bedFile, inputfmtBed = c("bed"),
  n=3,EnzymeType = c("SVMerge"),
  buildBNInternalDB=TRUE,
  path = path , pattern = pattern,
  buildSVInternalDB = TRUE,
  labelType = c("SE"),
  SE_path = SE_path, SE_pattern = SE_pattern,
  win_indel_INF = 10000, win_inv_trans_INF = 50000,
  perc_similarity_INF= 0.5, indelconf = 0.5, invconf = 0.01,
  transconf = 0.1, perc_similarity_INF_parents = 0.9,
  hgpath = hgpath, win_indel_DGV = 10000, win_inv_trans_DGV = 50000,
  perc_similarity_DGV = 0.5,
  RNASeqDatasetPresent = FALSE,
  RNASeqcombo = TRUE,
  RNASeqDir = RNASeqDir, returnMethod = "dataFrame",
  limsize = 1000,
  pattern_Proband = "*_P_*",
  outpath = outpath, primaryGenesPresent = FALSE,
  outputFilename = outputFilename,
  termListPresent = FALSE,
  InternaldatabasePresent = TRUE,
  outputType = c("Excel"), RZIPpath = RZIPpath)
```

```
## End(Not run)
```

---

nanotatoR\_main\_Duo\_SE *Annotation and visualisation of Bionano SV, of Single enzyme Duo samples.*

---

## Description

Annotation and visualisation of Bionano SV, of Single enzyme Duo samples.

## Usage

```
nanotatoR_main_Duo_SE(  
  smap,  
  bed,  
  inputfmtBed = c("bed", "BNBed"),  
  n = 3,  
  buildBNInternalDB = TRUE,  
  mergedFiles,  
  smappath,  
  buildSVInternalDB = FALSE,  
  path,  
  pattern,  
  win_indel_INF = 10000,  
  win_inv_trans_INF = 50000,  
  perc_similarity_INF = 0.5,  
  indelconf = 0.5,  
  invconf = 0.01,  
  transconf = 0.1,  
  perc_similarity_INF_parents = 0.9,  
  hgpath,  
  win_indel_DGV = 10000,  
  win_inv_trans_DGV = 50000,  
  perc_similarity_DGV = 0.5,  
  method_entrez = c("Single", "Multiple", "Text"),  
  termPath,  
  term,  
  thresh = 5,  
  limsize = 1000,  
  EnzymeType = c("SVmerge", "SE"),  
  labelType = c("SVMerge", "SE", "Both"),  
  SVMerge_path,  
  SVMerge_pattern,  
  SE_path,  
  SE_pattern,  
  Samplecodes,  
  mergeKey,
```

```

mergedKeyoutpath,
mergedKeyFname,
RNAseqcombo = TRUE,
RNASeqDir,
returnMethod = "dataFrame",
RNASeqData,
RNASeqPATH,
pattern_Proband = NA,
pattern_Mother = NA,
pattern_Father = NA,
outpath,
outputFilename = "",
termListPresent = TRUE,
internalBNDB,
clinvar,
InternaldatabasePresent = TRUE,
RNASeqDatasetPresent = TRUE,
geneListPresent = TRUE,
omim,
gtr,
removeClinvar = FALSE,
removeGTR = FALSE,
downloadClinvar = FALSE,
downloadGTR = FALSE,
url_gtr,
omimID,
RZIPpath,
directoryName,
fileprefix,
datGeneListPath,
decipherpath,
indexfile,
primaryGenesPresent = TRUE,
outputType = c("Excel", "csv")
)

```

### Arguments

smap	character. File name for the smap
bed	Text Bionano Bed file.
inputfmtBed	character Whether the bed input is UCSC bed or Bionano bed.
n	numeric Number of genes to report which are nearest to the breakpoint. Default is 3.
buildBNInternalDB	boolean. Checking whether the merged BNDB file database exist.
mergedFiles	character. Path to the merged SV files.
smappath	character. Path and file name for textfile.

buildSVInternalDB      boolean. Checking whether the merged solo file database exist.  
 path                    character. Path to the solo file database.  
 pattern                 character. pattern of the file names to merge.  
 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.  
 perc\_similarity\_INF\_parents    Numeric . ThresholdPercentage similarity for parent zygoty calculation. Default threshold 0.9.  
 hgpath                 character. Path to Database of Genomic Variants (DGV) Text file.  
 win\_indel\_DGV         Numeric. Insertion and deletion error window for DGV.  
 win\_inv\_trans\_DGV     Numeric. Inversion and translocation error window for DGV.  
 perc\_similarity\_DGV    Numeric . ThresholdPercentage similarity of the query SV and reference SV, for DGV..  
 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.  
 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.  
 limsize                Numeric. Minimum size for SV. Default 1000.  
 EnzymeType            Character. Type of enzyme. Options Dual and DLE.  
 labelType             character. Type of labels used for mapping. Choices are Dual, DLE and Both.  
 SVMerge\_path         character. Path for the Dual labelled cmap  
 SVMerge\_pattern       character. pattern of the dual files.  
 SE\_path                character. Path for the Dual labelled cmap  
 SE\_pattern             character. pattern of the dual files.  
 Samplecodes           character. File containing relations and IDs associated to them.  
 mergeKey              character. File containing sample ID and relation.  
 mergedKeyoutpath      character. File path storing sample name and nanoID key information.  
 mergedKeyFname        character. File name storing sample name and nanoID key information.  
 RNAseqcombo           boolean whether RNASeq datasets are combined or not.

RNASeqDir	boolean Directory for RNASeq.
returnMethod	character. Choice between text or data frame as the output.
RNASeqData	dataFrame. RNAseq data with gene names.
RNASeqPATH	character. RNAseq dataset path .
pattern_Proband	character. Pattern for proband.
pattern_Mother	character. Pattern to identify the mother reads.
pattern_Father	character. Pattern to identify the father reads.
outpath	Character Directory to the output file.
outputFilename	Character Output filename.
termListPresent	logical Checks whether term list is provided by the user.
internalBNDB	character. internak Bionano merged databse.
clinvar	character. clinvar file name and location.
InternaldatabasePresent	boolean. Checking whether internal DB present.
RNASeqDatasetPresent	boolean. Checking whether RNASeq database present or not.
geneListPresent	logical Checks whether gene list is provided by the user.
omim	character. omim2gene file name and location.
gtr	character. gtr file name and location.
removeClinvar	logical. Deletes the Clinvar database if TRUE.
removeGTR	logical. Deletes the GTR database if TRUE.
downloadClinvar	logical. Downloads the Clinvar database if TRUE.
downloadGTR	logical. Downloads the GTR database if TRUE.
url_gtr	character. url for GTR.
omimID	character. Omim ID.
RZIPpath	character. Path to RZippath.
directoryName	Directory name where individual SV files will be stored.
fileprefix	character Prefix to use for each of the files in the directory.
datGeneListPath	Character Path for genelist.
decipherpath	character. Decipher database path.
indexfile	character. indexfile containing nano ID and sample relation.
primaryGenesPresent	logical Checks whether primarygene list is provided by the user.
outputType	Variants in excel tabs or in different csv files. Options Excel or csv.

**Value**

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

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

**Examples**

```
## Not run:
terms="Muscle Weakness"
smapName="NA12878_DLE1_VAP_solo5.smap"
smap = system.file("extdata", smapName, package="nanotatoR")
bedFile <- system.file("extdata", "HomoSapienGRCH19_lift37.bed", package="nanotatoR")
hgpath=system.file("extdata", "GRCh37_hg19_variants_2016-05-15.txt", package="nanotatoR")
decipherpath = system.file("extdata", "population_cnv.txt", package="nanotatoR")
omim = system.file("extdata", "mim2gene.txt", package="nanotatoR")
clinvar = system.file("extdata", "localPDB/", package="nanotatoR")
gtr = system.file("extdata", "gtrDatabase.txt", package="nanotatoR")
labelType = c("SE")
SE_path = system.file("extdata", "SoloFile/", package="nanotatoR")
SE_pattern = "*_DLE1_*"
Samplecodes = system.file("extdata", "nanotatoR_sample_codes.csv", package="nanotatoR")
mergeKey = system.file("extdata", "nanotatoR_control_sample_codes.csv", package="nanotatoR")
mergedKeyoutpath = system.file("extdata", package="nanotatoR")
mergedKeyFname = "Sample_index.csv"
RNASeqDir = system.file("extdata", "NA12878_P_Blood_S1.genes.results", package="nanotatoR")
path = system.file("extdata", "Bionano_config/", package = "nanotatoR")
pattern = "_hg19.txt"
outputFilename <- "GM24385_DLE-1_P_trio_hg19_out"
outpath <- system.file("extdata", smapName, package = "nanotatoR")
RZIPpath <- system.file("extdata", "zip.exe", package = "nanotatoR")
nanotatoR_main_Duo_SE(
  smap = smap, bed = bedFile, inputfmtBed = c("bed"), limsize = 1000,
  n=3,EnzymeType = c("SE"),
  buildBNInternalDB=TRUE,
  path = path , pattern = pattern,
  buildSVInternalDB = TRUE,
  labelType = c("SE"),
  SE_path = SE_path, SE_pattern = SE_pattern,
  win_indel_INF = 10000, win_inv_trans_INF = 50000,
  perc_similarity_INF= 0.5, indelconf = 0.5, invconf = 0.01,
  transconf = 0.1, perc_similarity_INF_parents = 0.9,
  hgpath = hgpath, win_indel_DGV = 10000, win_inv_trans_DGV = 50000,
  perc_similarity_DGV = 0.5,
  method_entrez=c("Single"),
  term = "Liver cirrhosis",
  omim = omim, clinvar = clinvar, gtr = gtr,
  removeClinvar = TRUE, removeGTR = TRUE,
  downloadClinvar = FALSE, downloadGTR = FALSE,
  RNASeqDatasetPresent = FALSE,
  RNAseqcombo = TRUE,
  RNASeqDir = RNASeqDir, returnMethod = "dataFrame",
  pattern_Proband = "*_P_*",
  outpath = outpath,
```



```

outputFilename = outputFilename,
termListPresent = FALSE, primaryGenesPresent = FALSE,
InternaldatabasePresent = TRUE,
outputType = c("Excel"))

## End(Not run)

```

---

nanotatoR\_main\_Solo\_SE

*Annotation and visualisation of Bionano SV, of DLE Solo samples.*

---

## Description

Annotation and visualisation of Bionano SV, of DLE Solo samples.

## Usage

```

nanotatoR_main_Solo_SE(
  smap,
  bed,
  inputfmtBed = c("bed", "BNBed"),
  n = 3,
  buildBNInternalDB = TRUE,
  mergedFiles,
  smappath,
  buildSVInternalDB = FALSE,
  path,
  pattern,
  win_indel_INF = 10000,
  win_inv_trans_INF = 50000,
  perc_similarity_INF = 0.5,
  indelconf = 0.5,
  invconf = 0.01,
  transconf = 0.1,
  hgpath,
  win_indel_DGV = 10000,
  win_inv_trans_DGV = 50000,
  perc_similarity_DGV = 0.5,
  method_entrez = c("Single", "Multiple", "Text"),
  termPath,
  term,
  thresh = 5,
  limsize = 1000,
  EnzymeType = c("SVmerge", "SE"),
  labelType = c("SVMerge", "SE", "Both"),
  SVMerge_path,
  SVMerge_pattern,

```

```

SE_path,
SE_pattern,
Samplecodes,
mergeKey,
mergedKeyoutpath,
mergedKeyFname,
RNAseqcombo = TRUE,
RNASeqDir,
returnMethod = "dataFrame",
RNASeqData,
RNASeqPATH,
pattern_Proband = NA,
outpath,
outputFilename = "",
termListPresent = TRUE,
internalBNDB,
clinvar,
InternaldatabasePresent = TRUE,
RNASeqDatasetPresent = TRUE,
datGeneListPresent = TRUE,
omim,
gtr,
removeClinvar = FALSE,
removeGTR = FALSE,
downloadClinvar = FALSE,
downloadGTR = FALSE,
url_gtr,
omimID,
RZIPpath,
directoryName,
fileprefix,
datGeneListPath,
decipherpath,
indexfile,
primaryGenesPresent = TRUE,
outputType = c("Excel", "csv")
)

```

### Arguments

smap	character. File name for the smap
bed	Text Bionano Bed file.
inputfmtBed	character Whether the bed input is UCSC bed or Bionano bed.
n	numeric Number of genes to report which are nearest to the breakpoint. Default is 3.
buildBNInternalDB	boolean. Checking whether the merged BNDB file database exist.
mergedFiles	character. Path to the merged SV files.

smappath	character. Path and file name for textfile.
buildSVInternalDB	boolean. Checking whether the merged solo file database exist.
path	character. Path to the solo file database.
pattern	character. pattern of the file names to merge.
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.
hgpath	character. Path to Database of Genomic Variants (DGV) Text file.
win_indel_DGV	Numeric. Insertion and deletion error window for DGV.
win_inv_trans_DGV	Numeric. Inversion and translocation error window for DGV.
perc_similarity_DGV	Numeric . ThresholdPercentage similarity of the query SV and reference SV, for DGV..
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.
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.
limsize	Numeric. Minimum size for SV. Default 1000.
EnzymeType	Character. Type of enzyme. Options Dual and DLE.
labelType	character. Type of labels used for mapping. Choices are Dual, DLE and Both.
SVMerge_path	character. Path for the Dual labelled cmap
SVMerge_pattern	character. pattern of the dual files.
SE_path	character. Path for the Dual labelled cmap
SE_pattern	character. pattern of the dual files.
Samplecodes	character. File containing relations and IDs associated to them.
mergeKey	character. File containing sample ID and relation.
mergedKeyoutpath	character. File path storing sample name and nanoID key information.
mergedKeyFname	character. File name storing sample name and nanoID key information.
RNAseqcombo	boolean whether RNASeq datasets are combined or not.
RNASeqDir	boolean Directory for RNASeq.

returnMethod	character. Choice between text or data frame as the output.
RNASeqData	dataFrame. RNAseq data with gene names.
RNASeqPATH	character. RNAseq dataset path .
pattern_Proband	character. Pattern for proband.
outpath	Character Directory to the output file.
outputFilename	Character Output filename.
termListPresent	logical Checks whether term list is provided by the user.
internalBNDB	character. internak Bionano merged databse.
clinvar	character. clinvar file name and location.
InternaldatabasePresent	boolean. Checking whether internal DB present.
RNASeqDatasetPresent	boolean. Checking whether RNASeq database present or not.
datGeneListPresent	logical Checks whether gene list is provided by the user.
omim	character. omim2gene file name and location.
gtr	character. gtr file name and location.
removeClinvar	logical. Deletes the Clinvar database if TRUE.
removeGTR	logical. Deletes the GTR database if TRUE.
downloadClinvar	logical. Downloads the Clinvar database if TRUE.
downloadGTR	logical. Downloads the GTR database if TRUE.
url_gtr	character. url for GTR.
omimID	character. Omim ID.
RZIPpath	character. Path to RZippath.
directoryName	Directory name where individual SV files will be stored.
fileprefix	character Prefix to use for each of the files in the directory.
datGeneListPath	Character Path for genelist.
decipherpath	character. Decipher database path.
indexfile	character. indexfile containing nano ID and sample relation.
primaryGenesPresent	logical Checks whether primarygene list is provided by the user.
outputType	Variants in excel tabs or in different csv files. Options Excel or csv.

### Value

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

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

## Examples

```

smapName="NA12878_DLE1_VAP_solo5.smap"
smap = system.file("extdata", smapName, package="nanotatoR")
bedFile <- system.file("extdata", "HomoSapienGRCH19_lift37.bed", package="nanotatoR")
hgpath=system.file("extdata", "GRCh37_hg19_variants_2016-05-15.txt", package="nanotatoR")
decipherpath = system.file("extdata", "population_cnv.txt", package="nanotatoR")
omim = system.file("extdata", "mim2gene.txt", package="nanotatoR")
clinvar = system.file("extdata", "localPDB/", package="nanotatoR")
gtr = system.file("extdata", "gtrDatabase.txt", package="nanotatoR")
labelType = c("SE")
SE_path = system.file("extdata", "SoloFile/", package="nanotatoR")
SE_pattern = "*_DLE1_*"
Samplecodes = system.file("extdata", "nanotatoR_sample_codes.csv", package="nanotatoR")
mergeKey = system.file("extdata", "nanotatoR_control_sample_codes.csv", package="nanotatoR")
mergedKeyoutpath = system.file("extdata", package="nanotatoR")
mergedKeyFname = "Sample_index.csv"
RNASeqDir = system.file("extdata", "NA12878_P_Blood_S1.genes.results", package="nanotatoR")
path = system.file("extdata", "Bionano_config/", package = "nanotatoR")
pattern = "_hg19.txt"
outputFilename <- "NA12878_DLE1_VAP_solo5_out"
outpath <- system.file("extdata", smapName, package = "nanotatoR")
RZIPpath <- system.file("extdata", "zip.exe", package = "nanotatoR")
nanotatoR_main_Solo_SE(
  smap = smap, bed = bedFile, inputfmtBed = c("bed"),
  n=3,EnzymeType = c("SE"),
  buildBNInternalDB=TRUE,
  path = path , pattern = pattern,
  buildSVInternalDB = TRUE,
  labelType = c("SE"), decipherpath = decipherpath,
  SE_path = SE_path, SE_pattern = SE_pattern,
  win_indel_INF= 10000, win_inv_trans_INF = 50000,
  perc_similarity_INF= 0.5, indelconf = 0.5, invconf = 0.01,
  transconf = 0.1,
  hgpath = hgpath, win_indel_DGV = 10000,
  win_inv_trans_DGV = 50000,
  perc_similarity_DGV = 0.5, limsize = 1000,
  method_entrez=c("Single"),
  term = "Liver cirrhosis", RZIPpath = RZIPpath,
  omim = omim, clinvar = clinvar, gtr = gtr,
  removeClinvar = TRUE, removeGTR = TRUE,
  downloadClinvar = FALSE, downloadGTR = FALSE,
  RNASeqDatasetPresent = TRUE,
  RNASeqcombo = TRUE, datGeneListPresent = FALSE,
  RNASeqDir = RNASeqDir, returnMethod = "dataFrame",
  pattern_Proband = "*_P_*",
  outpath = outpath,
  indexfile = system.file("extdata", "Sample_index.csv", package="nanotatoR"),
  primaryGenesPresent = FALSE,
  outputFilename = outputFilename,
  termListPresent = FALSE,
  InternaldatabasePresent = TRUE,
  outputType = c("Excel"))

```

---

nanotatoR\_main\_Solo\_SVmerge

*Annotation and visualisation of Bionano SV, of Solo SVMerge samples.*

---

## Description

Annotation and visualisation of Bionano SV, of Solo SVMerge samples.

## Usage

```
nanotatoR_main_Solo_SVmerge(  
  smap,  
  bed,  
  inputfmtBed = c("bed", "BNBed"),  
  n = 3,  
  buildBNInternalDB = TRUE,  
  mergedFiles,  
  smappath,  
  buildSVInternalDB = FALSE,  
  path,  
  pattern,  
  win_indel_INF = 10000,  
  win_inv_trans_INF = 50000,  
  perc_similarity_INF = 0.5,  
  indelconf = 0.5,  
  invconf = 0.01,  
  transconf = 0.1,  
  hgpath,  
  win_indel_DGV = 10000,  
  win_inv_trans_DGV = 50000,  
  perc_similarity_DGV = 0.5,  
  method_entrez = c("Single", "Multiple", "Text"),  
  termPath,  
  term,  
  thresh = 5,  
  limsize = 1000,  
  EnzymeType = c("SVmerge", "SE"),  
  labelType = c("SVMerge", "SE", "Both"),  
  SVMerge_path,  
  SVMerge_pattern,  
  SE_path,  
  SE_pattern,  
  Samplecodes,  
  mergeKey,  
  mergedKeyoutputpath,  
  mergedKeyfname,  
  RNAseqcombo = TRUE,
```

```

RNASeqDir,
returnMethod = "dataFrame",
RNASeqData,
RNASeqPATH,
pattern_Proband = NA,
outpath,
outputFilename = "",
termListPresent = TRUE,
internalBNDB,
clinvar,
InternaldatabasePresent = TRUE,
RNASeqDatasetPresent = TRUE,
geneListPresent = TRUE,
omim,
gtr,
removeClinvar = FALSE,
removeGTR = FALSE,
downloadClinvar = FALSE,
downloadGTR = FALSE,
url_gtr,
omimID,
RZIPpath,
directoryName,
fileprefix,
datGeneListPath,
decipherpath,
indexfile,
primaryGenesPresent = TRUE,
outputType = c("Excel", "csv")
)

```

### Arguments

smap	character. File name for the smap
bed	Text Bionano Bed file.
inputfmtBed	character Whether the bed input is UCSC bed or Bionano bed.
n	numeric Number of genes to report which are nearest to the breakpoint. Default is 3.
buildBNInternalDB	boolean. Checking whether the merged BNDB file database exist.
mergedFiles	character. Path to the merged SV files.
smappath	character. Path and file name for textfile.
buildSVInternalDB	boolean. Checking whether the merged solo file database exist.
path	character. Path to the solo file database.
pattern	character. pattern of the file names to merge.

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.
hgpath	character. Path to Database of Genomic Variants (DGV) Text file.
win_indel_DGV	Numeric. Insertion and deletion error window for DGV.
win_inv_trans_DGV	Numeric. Inversion and translocation error window for DGV.
perc_similarity_DGV	Numeric . ThresholdPercentage similarity of the query SV and reference SV, for DGV..
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.
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.
limsize	Numeric. Minimum size for SV. Default 1000.
EnzymeType	Character. Type of enzyme. Options Dual and DLE.
labelType	character. Type of labels used for mapping. Choices are Dual, DLE and Both.
SVMerge_path	character. Path for the Dual labelled cmap
SVMerge_pattern	character. pattern of the dual files.
SE_path	character. Path for the Dual labelled cmap
SE_pattern	character. pattern of the dual files.
Samplecodes	character. File containing relations and IDs associated to them.
mergeKey	character. File containing sample ID and relation.
mergedKeyoutpath	character. File path storing sample name and nanoID key information.
mergedKeyFname	character. File name storing sample name and nanoID key information.
RNAseqcombo	boolean whether RNASeq datasets are combined or not.
RNASeqDir	boolean Directory for RNASeq.
returnMethod	character. Choice between text or data frame as the output.
RNASeqData	dataFrame. RNAseq data with gene names.
RNASeqPATH	character. RNAseq dataset path .
pattern_Proband	character. Pattern for proband.



outpath	Character Directory to the output file.
outputFilename	Character Output filename.
termListPresent	logical Checks whether term list is provided by the user.
internalBNDB	character. internak Bionano merged databse.
clinvar	character. clinvar file name and location.
InternaldatabasePresent	boolean. Checking whether internal DB present.
RNASeqDatasetPresent	boolean. Checking whether RNASeq database present or not.
geneListPresent	logical Checks whether gene list is provided by the user.
omim	character. omim2gene file name and location.
gtr	character. gtr file name and location.
removeClinvar	logical. Deletes the Clinvar database if TRUE.
removeGTR	logical. Deletes the GTR database if TRUE.
downloadClinvar	logical. Downloads the Clinvar database if TRUE.
downloadGTR	logical. Downloads the GTR database if TRUE.
url_gtr	character. url for GTR.
omimID	character. Omim ID.
RZIPpath	character. Path to RZippath.
directoryName	Directory name where individual SV files will be stored.
fileprefix	character Prefix to use for each of the files in the directory.
datGeneListPath	Character Path for genelist.
decipherpath	character. Decipher database path.
indexfile	character. indexfile containing nano ID and sample relation.
primaryGenesPresent	logical Checks whether primarygene list is provided by the user.
outputType	Variants in excel tabs or in different csv files. Options Excel or csv.

**Value**

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

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

**Examples**

```

smapName="NA12878_Q.S_VAP_SVmerge_solo5.txt"
smap = system.file("extdata", smapName, package="nanotatoR")
bedFile <- system.file("extdata", "HomoSapienGRCH19_lift37.bed", package="nanotatoR")
hgpath=system.file("extdata", "GRCh37_hg19_variants_2016-05-15.txt", package="nanotatoR")
decipherpath = system.file("extdata", "population_cnv.txt", package="nanotatoR")
omim = system.file("extdata", "mim2gene.txt", package="nanotatoR")
clinvar = system.file("extdata", "localPDB/", package="nanotatoR")
gtr = system.file("extdata", "gtrDatabase.txt", package="nanotatoR")
labelType = c("SE")
SE_path = system.file("extdata", "SoloFile/", package="nanotatoR")
SE_pattern = "*_DLE1_*"
Samplecodes = system.file("extdata", "nanotatoR_sample_codes.csv", package="nanotatoR")
mergeKey = system.file("extdata", "nanotatoR_control_sample_codes.csv", package="nanotatoR")
mergedKeyoutpath = system.file("extdata", package="nanotatoR")
mergedKeyFname = "Sample_index.csv"
RNASeqDir = system.file("extdata", "NA12878_P_Blood_S1.genes.results", package="nanotatoR")
path = system.file("extdata", "Bionano_config/", package = "nanotatoR")
pattern = "_hg19.txt"
outputFilename <- "NA12878_Q.S_VAP_SVmerge_solo5_out"
outpath <- system.file("extdata", smapName, package = "nanotatoR")
RZIPpath <- system.file("extdata", "zip.exe", package = "nanotatoR")
nanotatoR_main_Solo_SVmerge(
  smap = smap, bed = bedFile, inputfmtBed = c("bed"),
  n=3,EnzymeType = c("SVMerge"),
  buildBNInternalDB=TRUE,
  path = path , pattern = pattern,
  buildSVInternalDB = TRUE,
  labelType = c("SE"), decipherpath = decipherpath,
  SE_path = SE_path, SE_pattern = SE_pattern,
  win_indel_INF= 10000, win_inv_trans_INF = 50000,
  perc_similarity_INF= 0.5, indelconf = 0.5, invconf = 0.01,
  transconf = 0.1,
  hgpath = hgpath, win_indel_DGV = 10000,
  win_inv_trans_DGV = 50000,
  perc_similarity_DGV = 0.5, limsize = 1000,
  method_entrez=c("Single"),
  term = "Liver cirrhosis", RZIPpath = RZIPpath,
  omim = omim, clinvar = clinvar, gtr = gtr,
  removeClinvar = TRUE, removeGTR = TRUE,
  downloadClinvar = FALSE, downloadGTR = FALSE,
  RNASeqDatasetPresent = TRUE,
  RNASeqcombo = TRUE, geneListPresent = FALSE,
  RNASeqDir = RNASeqDir, returnMethod = "dataFrame",
  pattern_Proband = "*_P_*",
  outpath = outpath,
  indexfile = system.file("extdata", "Sample_index.csv", package="nanotatoR"),
  primaryGenesPresent = FALSE,
  outputFilename = outputFilename,
  termListPresent = FALSE,
  InternaldatabasePresent = TRUE,
  outputType = c("Excel"))

```

---

`nanotatoR_main_Trio_SE`*Annotation and visualisation of Bionano SV, of DLE Trio samples.*

---

**Description**

Annotation and visualisation of Bionano SV, of DLE Trio samples.

**Usage**

```
nanotatoR_main_Trio_SE(  
  smap,  
  bed,  
  inputfmtBed = c("bed", "BNBed"),  
  n = 3,  
  buildBNInternalDB = TRUE,  
  mergedFiles,  
  smappath,  
  buildSVInternalDB = FALSE,  
  path,  
  pattern,  
  win_indel_INF = 10000,  
  win_inv_trans_INF = 50000,  
  perc_similarity_INF = 0.5,  
  indelconf = 0.5,  
  invconf = 0.01,  
  transconf = 0.1,  
  perc_similarity_INF_parents = 0.9,  
  hgpath,  
  win_indel_DGV = 10000,  
  win_inv_trans_DGV = 50000,  
  perc_similarity_DGV = 0.5,  
  method_entrez = c("Single", "Multiple", "Text"),  
  termPath,  
  term,  
  thresh = 5,  
  limsize = 1000,  
  EnzymeType = c("SVmerge", "SE"),  
  labelType = c("SVMerge", "SE", "Both"),  
  SVMerge_path,  
  SVMerge_pattern,  
  SE_path,  
  SE_pattern,  
  Samplecodes,  
  mergeKey,  
  mergedKeyoutpath,  
  mergedKeyFname,
```

```

RNAseqcombo = TRUE,
RNASeqDir,
returnMethod = "dataFrame",
RNASeqData,
RNASeqPATH,
pattern_Proband = NA,
pattern_Mother = NA,
pattern_Father = NA,
outpath,
outputFilename = "",
termListPresent = TRUE,
internalBNDB,
clinvar,
InternaldatabasePresent = TRUE,
RNASeqDatasetPresent = TRUE,
geneListPresent = TRUE,
omim,
gtr,
removeClinvar = FALSE,
removeGTR = FALSE,
downloadClinvar = FALSE,
downloadGTR = FALSE,
url_gtr,
omimID,
RZIPpath,
directoryName,
fileprefix,
datGeneListPath,
decipherpath,
indexfile,
primaryGenesPresent = TRUE,
outputType = c("Excel", "csv")
)

```

### Arguments

smap	character. File name for the smap
bed	Text Bionano Bed file.
inputfmtBed	character Whether the bed input is UCSC bed or Bionano bed.
n	numeric Number of genes to report which are nearest to the breakpoint. Default is 3.
buildBNInternalDB	boolean. Checking whether the merged BNDB file database exist.
mergedFiles	character. Path to the merged SV files.
smappath	character. Path and file name for textfile.
buildSVInternalDB	boolean. Checking whether the merged solo file database exist.

path	character. Path to the solo file database.
pattern	character. pattern of the file names to merge.
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.
perc_similarity_INF_parents	Numeric . ThresholdPercentage similarity for parent zygosity calculation. Default threshold 0.9.
hgpath	character. Path to Database of Genomic Variants (DGV) Text file.
win_indel_DGV	Numeric. Insertion and deletion error window for DGV.
win_inv_trans_DGV	Numeric. Inversion and translocation error window for DGV.
perc_similarity_DGV	Numeric . ThresholdPercentage similarity of the query SV and reference SV, for DGV..
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.
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.
limsize	Numeric. Minimum size for SV. Default 1000.
EnzymeType	Character. Type of enzyme. Options Dual and DLE.
labelType	character. Type of labels used for mapping. Choices are Dual, DLE and Both.
SVMerge_path	character. Path for the Dual labelled cmap
SVMerge_pattern	character. pattern of the dual files.
SE_path	character. Path for the Dual labelled cmap
SE_pattern	character. pattern of the dual files.
Samplecodes	character. File containing relations and IDs associated to them.
mergeKey	character. File containing sample ID and relation.
mergedKeyoutpath	character. File path storing sample name and nanoID key information.
mergedKeyFname	character. File name storing sample name and nanoID key information.
RNAseqcombo	boolean whether RNASeq datasets are combined or not.
RNASeqDir	boolean Directory for RNASeq.
returnMethod	character. Choice between text or data frame as the output.

RNASeqData	dataFrame. RNAseq data with gene names.
RNASeqPATH	character. RNAseq dataset path .
pattern_Proband	character. Pattern for proband.
pattern_Mother	character. Pattern to identify the mother reads.
pattern_Father	character. Pattern to identify the father reads.
outpath	Character Directory to the output file.
outputFilename	Character Output filename.
termListPresent	logical Checks whether term list is provided by the user.
internalBNDB	character. internak Bionano merged databse.
clinvar	character. clinvar file name and location.
InternaldatabasePresent	boolean. Checking whether internal DB present.
RNASeqDatasetPresent	boolean. Checking whether RNASeq database present or not.
geneListPresent	logical Checks whether gene list is provided by the user.
omim	character. omim2gene file name and location.
gtr	character. gtr file name and location.
removeClinvar	logical. Deletes the Clinvar database if TRUE.
removeGTR	logical. Deletes the GTR database if TRUE.
downloadClinvar	logical. Downloads the Clinvar database if TRUE.
downloadGTR	logical. Downloads the GTR database if TRUE.
url_gtr	character. url for GTR.
omimID	character. Omim ID.
RZiPpath	character. Path to RZiPpath.
directoryName	Directory name where individual SV files will be stored.
fileprefix	character Prefix to use for each of the files in the directory.
datGeneListPath	Character Path for genelist.
decipherpath	character. Decipher database path.
indexfile	character. indexfile containing nano ID and sample relation.
primaryGenesPresent	logical Checks whether primarygene list is provided by the user.
outputType	Variants in excel tabs or in different csv files. Options Excel or csv.

### Value

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

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

## Examples

```

smapName="GM24385_Ason_DLE1_VAP_trio5.smap"
smap = system.file("extdata", smapName, package="nanotatoR")
bedFile <- system.file("extdata", "HomoSapienGRCH19_lift37.bed", package="nanotatoR")
hgpath=system.file("extdata", "GRCh37_hg19_variants_2016-05-15.txt", package="nanotatoR")
decipherpath = system.file("extdata", "population_cnv.txt", package="nanotatoR")
omim = system.file("extdata", "mim2gene.txt", package="nanotatoR")
clinvar = system.file("extdata", "localPDB/", package="nanotatoR")
gtr = system.file("extdata", "gtrDatabase.txt", package="nanotatoR")
mergedFiles = system.file("extdata", "nanotatoRControl.txt", package="nanotatoR")
indexfile = system.file("extdata", "Sample_index.csv", package="nanotatoR")
RNASeqDir = system.file("extdata", "NA12878_P_Blood_S1.genes.results", package="nanotatoR")
path = system.file("extdata", "Bionano_config/", package = "nanotatoR")
pattern = "_hg19.txt"
outputFilename <- "NA12878_DLE1_VAP_solo5_out"
outpath <- system.file("extdata", smapName, package = "nanotatoR")
RZIPpath <- system.file("extdata", "zip.exe", package = "nanotatoR")
nanotatoR_main_Trio_SE(
  smap = smap, bed = bedFile, inputfmtBed = c("bed"),
  n=3,EnzymeType = c("SE"),
  buildBNInternalDB=TRUE,
  path = path , pattern = pattern,
  buildSVInternalDB = FALSE,
  decipherpath = decipherpath,
  win_indel_INF = 10000, win_inv_trans_INF = 50000,
  perc_similarity_INF= 0.5, indelconf = 0.5, invconf = 0.01,
  transconf = 0.1, perc_similarity_INF_parents = 0.9,
  hgpath = hgpath, win_indel_DGV = 10000,
  win_inv_trans_DGV = 50000,
  perc_similarity_DGV = 0.5, limsize = 1000,
  method_entrez=c("Single"),
  term = "Liver cirrhosis", RZIPpath = RZIPpath,
  omim = omim, clinvar = clinvar, gtr = gtr,
  removeClinvar = TRUE, removeGTR = TRUE,
  downloadClinvar = FALSE, downloadGTR = FALSE,
  RNASeqDatasetPresent = TRUE,
  RNASeqcombo = TRUE, geneListPresent = FALSE,
  RNASeqDir = RNASeqDir, returnMethod = "dataFrame",
  pattern_Proband = "*_P_*",
  outpath = outpath,
  indexfile = system.file("extdata", "Sample_index.csv", package="nanotatoR"),
  primaryGenesPresent = FALSE,
  outputFilename = outputFilename,
  termListPresent = FALSE,
  InternaldatabasePresent = TRUE,
  outputType = c("Excel"))

```

---

nanotatoR\_SVmerge\_Trio

*Annotation and visualisation of Bionano SV, of DLE Trio samples.*

---

**Description**

Annotation and visualisation of Bionano SV, of DLE Trio samples.

**Usage**

```
nanotatoR_SVmerge_Trio(  
  smap,  
  bed,  
  inputfmtBed = c("bed", "BNBed"),  
  n = 3,  
  buildBNInternalDB = TRUE,  
  mergedFiles,  
  smappath,  
  buildSVInternalDB = FALSE,  
  path,  
  pattern,  
  win_indel_INF = 10000,  
  win_inv_trans_INF = 50000,  
  perc_similarity_INF = 0.5,  
  indelconf = 0.5,  
  invconf = 0.01,  
  transconf = 0.1,  
  perc_similarity_INF_parents = 0.9,  
  hgpath,  
  win_indel_DGV = 10000,  
  win_inv_trans_DGV = 50000,  
  perc_similarity_DGV = 0.5,  
  method_entrez = c("Single", "Multiple", "Text"),  
  termPath,  
  term,  
  thresh = 5,  
  limsize = 1000,  
  EnzymeType = c("SVmerge", "SE"),  
  labelType = c("SVMerge", "SE", "Both"),  
  SVMerge_path,  
  SVMerge_pattern,  
  SE_path,  
  SE_pattern,  
  Samplecodes,  
  mergeKey,  
  mergedKeyoutpath,  
  mergedKeyfname,  
  RNAseqcombo = TRUE,  
  RNASeqDir,  
  returnMethod = "dataFrame",  
  RNASeqData,  
  RNASeqPATH,  
  pattern_Proband = NA,
```



```

pattern_Mother = NA,
pattern_Father = NA,
outpath,
outputFilename = "",
termListPresent = TRUE,
internalBNDB,
clinvar,
InternaldatabasePresent = TRUE,
RNASeqDatasetPresent = TRUE,
geneListPresent = TRUE,
omim,
gtr,
removeClinvar = FALSE,
removeGTR = FALSE,
downloadClinvar = FALSE,
downloadGTR = FALSE,
url_gtr,
omimID,
RZIPpath,
directoryName,
fileprefix,
datGeneListPath,
decipherpath,
indexfile,
primaryGenesPresent = TRUE,
outputType = c("Excel", "csv")
)

```

### Arguments

smap	character. File name for the smap
bed	Text Bionano Bed file.
inputfmtBed	character Whether the bed input is UCSC bed or Bionano bed.
n	numeric Number of genes to report which are nearest to the breakpoint. Default is 3.
buildBNInternalDB	boolean. Checking whether the merged BNDB file database exist.
mergedFiles	character. Path to the merged SV files.
smappath	character. Path and file name for textfile.
buildSVInternalDB	boolean. Checking whether the merged solo file database exist.
path	character. Path to the solo file database.
pattern	character. pattern of the file names to merge.
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.
perc_similarity_INF_parents	Numeric . ThresholdPercentage similarity for parent zygosity calculation. Default threshold 0.9.
hgpath	character. Path to Database of Genomic Variants (DGV) Text file.
win_indel_DGV	Numeric. Insertion and deletion error window for DGV.
win_inv_trans_DGV	Numeric. Inversion and translocation error window for DGV.
perc_similarity_DGV	Numeric . ThresholdPercentage similarity of the query SV and reference SV, for DGV..
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.
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.
limsize	Numeric. Minimum size for SV. Default 1000.
EnzymeType	Character. Type of enzyme. Options Dual and DLE.
labelType	character. Type of labels used for mapping. Choices are Dual, DLE and Both.
SVMerge_path	character. Path for the Dual labelled cmap
SVMerge_pattern	character. pattern of the dual files.
SE_path	character. Path for the Dual labelled cmap
SE_pattern	character. pattern of the dual files.
Samplecodes	character. File containing relations and IDs associated to them.
mergeKey	character. File containing sample ID and relation.
mergedKeyoutpath	character. File path storing sample name and nanoID key information.
mergedKeyFname	character. File name storing sample name and nanoID key information.
RNAseqcombo	boolean whether RNASeq datasets are combined or not.
RNASeqDir	boolean Directory for RNASeq.
returnMethod	character. Choice between text or data frame as the output.
RNASeqData	dataFrame. RNAseq data with gene names.
RNASeqPATH	character. RNAseq dataset path .
pattern_Proband	character. Pattern for proband.
pattern_Mother	character. Pattern to identify the mother reads.

pattern_Father	character. Pattern to identify the father reads.
outputpath	Character Directory to the output file.
outputFilename	Character Output filename.
termListPresent	logical Checks whether term list is provided by the user.
internalBNDB	character. internak Bionano merged databse.
clinvar	character. clinvar file name and location.
InternaldatabasePresent	boolean. Checking whether internal DB present.
RNASeqDatasetPresent	boolean. Checking whether RNASeq database present or not.
geneListPresent	logical Checks whether gene list is provided by the user.
omim	character. omim2gene file name and location.
gtr	character. gtr file name and location.
removeClinvar	logical. Deletes the Clinvar database if TRUE.
removeGTR	logical. Deletes the GTR database if TRUE.
downloadClinvar	logical. Downloads the Clinvar database if TRUE.
downloadGTR	logical. Downloads the GTR database if TRUE.
url_gtr	character. url for GTR.
omimID	character. Omim ID.
RZIPpath	character. Path to RZippath.
directoryName	Directory name where individual SV files will be stored.
fileprefix	character Prefix to use for each of the files in the directory.
datGeneListPath	Character Path for genelist.
decipherpath	character. Decipher database path.
indexfile	character. indexfile containing nano ID and sample relation.
primaryGenesPresent	logical Checks whether primarygene list is provided by the user.
outputType	Variants in excel tabs or in different csv files. Options Excel or csv.

**Value**

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

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

**Examples**

```

## Not run:
smapName="NA12878_Q.S_VAP_SVmerge_solo5.txt"
smap = system.file("extdata", smapName, package="nanotatoR")
bedFile <- system.file("extdata", "HomoSapienGRCH19_lift37.bed", package="nanotatoR")
hgpath=system.file("extdata", "GRCh37_hg19_variants_2016-05-15.txt", package="nanotatoR")
labelType = c("SE")
SE_path = system.file("extdata", "SoloFile/", package="nanotatoR")
SE_pattern = "*_DLE1_*"
Samplecodes = system.file("extdata", "nanotatoR_sample_codes.csv", package="nanotatoR")
mergeKey = system.file("extdata", "nanotatoR_control_sample_codes.csv", package="nanotatoR")
mergedKeyoutpath = system.file("extdata", package="nanotatoR")
mergedKeyFname = "Sample_index.csv"
RNASeqDir = system.file("extdata", "NA12878_P_Blood_S1.genes.results", package="nanotatoR")
path = system.file("extdata", "Bionano_config/", package = "nanotatoR")
pattern = "_hg19.txt"
outputFilename <- "GM24385_DLE-1_P_trio_hg19_out"
outpath <- system.file("extdata", smapName, package = "nanotatoR")
RZIPpath <- system.file("extdata", "zip.exe", package = "nanotatoR")
nanotatoR_main_Solo_SE(
  smap = smap, bed = bedFile, inputfmt = c("bed"),
  n=3,
  buildBNInternalDB=TRUE,
  path = path , pattern = pattern,
  buildSVInternalDB = TRUE,
  EnzymeType = c("SVMerge"),
  labelType = c("SVMerge"),
  SE_path = SE_path, SE_pattern = SE_pattern,
  win_indel_INF = 10000, win_inv_trans_INF = 50000,
  perc_similarity_INF= 0.5, indelconf = 0.5, invconf = 0.01,
  transconf = 0.1,
  hgpath = hgpath, win_indel_DGV = 10000, win_inv_trans_DGV = 50000,
  perc_similarity_DGV = 0.5,
  RNAseqcombo = TRUE, perc_similarity_INF_parents = 0.9,
  RNASeqDir = RNASeqDir, returnMethod = "dataFrame",
  pattern_Proband = "*_P_*",
  outpath = outpath,
  outputFilename = outputFilename,
  termListPresent = FALSE,
  InternaldatabasePresent = FALSE,
  primaryGenesPresent = FALSE,
  outputType = c("Excel"))

## End(Not run)

```

---

nonOverlapGenes

*Calculates Genes that are near to the SV region*


---

**Description**

Calculates Genes that are near to the SV region

**Usage**

```
nonOverlapGenes(
  bed,
  chrom,
  startpos,
  chrom2,
  endpos,
  svid,
  n = 3,
  SVTyp,
  bpererrorindel = 3000,
  bpererrorinvtrans = 50000
)
```

**Arguments**

bed	Text Bionano Bed file.
chrom	character SVmap chromosome.
startpos	numeric starting position of the breakpoints.
chrom2	character SVmap 2nd chromosome.
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.
SVTyp	Character. Type of SV.
bpererrorindel	Numeric. base pair error indel.
bpererrorinvtrans	Numeric. base pair error invtranslocation.

**Value**

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

**Examples**

```
smapName="GM24385_Ason_DLE1_VAP_trio5.smap"
smap = system.file("extdata", smapName, package="nanotatoR")
bedFile <- system.file("extdata", "HomoSapienGRCH19_lift37.bed",
  package="nanotatoR")
bed<-buildrunBNBedFiles(bedFile,returnMethod="dataFrame")
smap<-readSMap_DLE(smap, input_fmt_smap = "Text")
chrom<-smap$RefcontigID1
chrom2 <- smap$RefcontigID2
startpos<-smap$RefStartPos
endpos<-smap$RefEndPos
if (length(grep("SVIndex",names(smap)))>0){
  svid <- smap$SVIndex
```

```

    }else{
      svid <- smap$SmapEntryID
    }
  SVTyp <- smap$Type
  n<-3
  nonOverlapGenes.bed = bed, chrom = chrom, startpos = startpos,
  endpos = endpos, svid = svid, chrom2 = chrom2, SVTyp = SVTyp,
  bperrorindel = 3000, bperrorinvtrans = 50000, n = 3)

```

---

nonOverlappingDNGenes *Extracting terms for genes that overlap SVs*

---

### Description

Extracting terms for genes that overlap SVs

### Usage

```
nonOverlappingDNGenes(rr, dngene)
```

### Arguments

rr                    character. dataframe with primary genes and terms associated.  
dngene                character. genes that overlap the SV.

### Value

Dataframe with overlapping genes and terms.

### Examples

```

terms= c("steroid_Gene","steroid synthesis_Gene")
genes <- c("NR1H3", "ABCC4")
rr <- data.frame(Genes = genes, Terms = terms)
smapName="GM24385_Ason_DLE1_VAP_trio5.smap"
smap = system.file("extdata", smapName, package="nanotatoR")
bedFile <- system.file("extdata", "HomoSapienGRCH19_lift37.bed", package="nanotatoR")
outpath <- system.file("extdata", package="nanotatoR")
datcomp<-overlapnearestgeneSearch(smap = smap,
bed=bedFile, inputfmtBed = "bed", outpath,
n = 3, returnMethod_bedcomp = c("dataFrame"),
input_fmt_SV = "Text",
EnzymeType = "SE",
bperrorindel = 3000, bperrorinvtrans = 50000)
dngene <- as.character(datcomp$Downstream_nonOverlapGenes_dist_kb)
dataPGDN <- nonOverlappingDNGenes (rr, dngene)

```

---

nonOverlappingUPGenes *Extracting terms for genes that overlap SVs*

---

**Description**

Extracting terms for genes that overlap SVs

**Usage**

```
nonOverlappingUPGenes(rr, upgene)
```

**Arguments**

rr                    character. dataframe with primary genes and terms associated.  
 upgene                character. genes that overlap the SV.

**Value**

Dataframe with overlapping genes and terms.

**Examples**

```
terms= c("steroid_Gene","steroid synthesis_Gene")
genes <- c("NR1H3", "ABCC4")
rr <- data.frame(Genes = genes, Terms = terms)
smapName="GM24385_Ason_DLE1_VAP_trio5.smap"
smap = system.file("extdata", smapName, package="nanotatoR")
bedFile <- system.file("extdata", "HomoSapienGRCH19_lift37.bed", package="nanotatoR")
outpath <- system.file("extdata", package="nanotatoR")
datcomp<-overlapnearestgeneSearch(smap = smap,
bed=bedFile, inputfmtBed = "bed", outpath,
n = 3, returnMethod_bedcomp = c("dataFrame"),
input_fmt_SV = "Text",
EnzymeType = "SE",
bperrorindel = 3000, bperrorinvtrans = 50000)
upgene <- as.character(datcomp$Upstream_nonOverlapGenes_dist_kb)
dataPGUP <- nonOverlappingUPGenes (rr, upgene)
```

---

nonOverlapRNAseq            *Extract Read counts for genes that are near SVs.*

---

**Description**

Extract Read counts for genes that are near SVs.

**Usage**

```
nonOverlapRNAseq(
  gnsNonOverlap,
  SVID,
  RNASeqData,
  pattern_Proband = NA,
  pattern_Mother = NA,
  pattern_Father = NA
)
```

**Arguments**

`gnsNonOverlap` character. genes that are upstream and/or downstream of SV.  
`SVID` character. ID of the SVs.  
`RNASeqData` character. Expression of the genes.  
`pattern_Proband` character. Pattern to identify the proband reads.  
`pattern_Mother` character. Pattern to identify the mother reads.  
`pattern_Father` character. Pattern to identify the father reads.

**Value**

Text or Dataframe containing TPM read counts of genes in the family.

**Examples**

```
RNASeqDir = system.file("extdata", package="nanotatoR")
returnMethod="dataFrame"
datRNASeq <- RNAseqcombine(RNASeqDir = RNASeqDir,
returnMethod = returnMethod)
gnsNonOverlap <- c("DDX11L1", "MIR1302-2HG", "OR4G4P")
SVID = 397
datgnonovrlap <- nonOverlapRNAseq(gnsNonOverlap = gnsNonOverlap,
SVID = SVID, RNASeqData = datRNASeq,
pattern_Proband = "*_P_*")
```

---

`nonOverlapRNAseq_solo` *Annotating the Non-Overlapping genes with RNAseq expression*

---

**Description**

Annotating the Non-Overlapping genes with RNAseq expression

**Usage**

```
nonOverlapRNAseq_solo(gnsNonOverlap, SVID, RNASeqData, pattern_Proband = NA)
```



**Arguments**

gnsNonOverlap    character. Vector containing non-overlapping genes.  
 SVID             character. SV Index ID.  
 RNASeqData      dataframe. RNAseq data with gene names.  
 pattern\_Proband    character. Pattern for proband.

**Value**

Dataframe containing TPM read counts of overlapping genes.

**Examples**

```
RNASeqDir = system.file("extdata", package="nanotatoR")
returnMethod="dataFrame"
datRNASeq <- RNAseqcombine_solo(RNASeqDir = RNASeqDir,
returnMethod = returnMethod)
gnsNonOverlap <- c("DDX11L1", "MIR1302-2HG", "OR4G4P")
SVID = 397
datgnonovrlap <- nonOverlapRNAseq_solo(gnsNonOverlap = gnsNonOverlap,
SVID = SVID, RNASeqData = datRNASeq,
pattern_Proband = "*_P_*")
```

---

omim\_gene

*Extracting genes from OMIM database NCBI.*


---

**Description**

Extracting genes from OMIM database NCBI.

**Usage**

```
omim_gene(terms, omim)
```

**Arguments**

terms             character Single or Multiple Terms.  
 omim             character omim database location.

**Value**

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

**Examples**

```
terms="Liver cirrhosis"
omim = system.file("extdata", "mim2gene.txt", package="nanotatoR")
ge <- omim_gene(terms = terms, omim = omim)
```

---

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,
  chrom2,
  SVTyp,
  bperrorindel = 3000,
  bperrorinvtrans = 50000
)
```

### 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).
chrom2	character SVmap chromosome number 2.
SVTyp	Character. Type of SV.
bperrorindel	Numeric. base pair error indel.
bperrorinvtrans	Numeric. base pair error invtranslocation.

### Value

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

### Examples

```
smapName="GM24385_Ason_DLE1_VAP_trio5.smap"
smap = system.file("extdata", smapName, package="nanotatoR")
bedFile <- system.file("extdata", "HomoSapienGRCH19_lift37.bed",
  package="nanotatoR")
bed<-buildrunBNBedFiles(bedFile,returnMethod="dataFrame")
```

```

smap<-readSMap_DLE(smap, input_fmt_smap = "Text")
chrom<-smap$RefcontigID1
chrom2 <- smap$RefcontigID2
startpos<-smap$RefStartPos
endpos<-smap$RefEndPos
if (length(grep("SVIndex",names(smap)))>0){
  svid <- smap$SVIndex
}else{
  svid <- smap$SmapEntryID
}
SVTyp <- smap$Type
overlapGenes.bed = bed, chrom = chrom, startpos = startpos,
  endpos = endpos, chrom2 = chrom2, svid = svid,
  SVTyp = SVTyp,
  bperrorindel = 3000, bperrorinvtrans = 50000)

```

---

overlapnearestgeneSearch

*Extracts gene information from bed files*

---

## Description

Extracts gene information from bed files

## Usage

```

overlapnearestgeneSearch(
  smap,
  bed,
  inputfmtBed = c("bed", "BNBED"),
  EnzymeType = c("SVMerge", "SE"),
  outpath,
  n = 3,
  returnMethod_bedcomp = c("Text", "dataFrame"),
  input_fmt_SV = c("Text", "dataFrame"),
  bperrorindel = 3000,
  bperrorinvtrans = 50000
)

```

## Arguments

smap	character or dataFrame depending on the input_fmt_SV argument. If input_fmt_SV='Text', it is path to SMAP file. If input_fmt_SV='dataFrame', it is a dataframe.
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 ' if(\$3 == "gene" &amp;&amp; \$13 == "gene_status") print \$1,\$4,\$5,\$16,\$7 else if (\$3 == "gene" &amp;&amp; \$13 == "gene_name") print \$1,\$4,\$5,\$14,\$7 ' gencode.v33.annotation.gtf &gt;HomoSapienGRCH19.bed</code>

EnzymeType	Character. Type of enzyme. Options SVMerge and SE.
outpath	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.
input_fmt_SV	Character. Type of output Dataframe or in Text format.
bperrorindel	Numeric. base pair error indel.
bperrorinvtrans	Numeric. base pair error invtranslocation.

**Value**

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

**Examples**

```

smapName="GM24385_Ason_DLE1_VAP_trio5.smap"
smap = system.file("extdata", smapName, package="nanotatoR")
bedFile <- system.file("extdata", "HomoSapienGRCH19_lift37.bed", package="nanotatoR")
outpath <- system.file("extdata", package="nanotatoR")
datcomp<-overlapnearestgeneSearch(smap = smap,
bed=bedFile, inputfmtBed = "bed", outpath,
n = 3, returnMethod_bedcomp = c("dataFrame"),
input_fmt_SV = "Text",
EnzymeType = "SE",
bperrorindel = 3000, bperrorinvtrans = 50000)
datcomp[1,]

```

---

overlappingGenes      *Extracting terms for genes that overlap SVs*

---

**Description**

Extracting terms for genes that overlap SVs

**Usage**

```
overlappingGenes(rr, ogene)
```

**Arguments**

rr	character. dataframe with primary genes and terms associated.
ogene	character. genes that overlap the SV.

**Value**

Dataframe with overlapping genes and terms.

**Examples**

```

terms= c("steroid_Gene","steroid synthesis_Gene")
genes <- c("NR1H3", "ABCC4")
rr <- data.frame(Genes = genes, Terms = terms)
smapName="GM24385_Ason_DLE1_VAP_trio5.smap"
smap = system.file("extdata", smapName, package="nanotatoR")
bedFile <- system.file("extdata", "HomoSapienGRCH19_lift37.bed", package="nanotatoR")
outpath <- system.file("extdata", package="nanotatoR")
datcomp<-overlapnearestgeneSearch(smap = smap,
bed=bedFile, inputfmtBed = "bed", outpath,
n = 3, returnMethod_bedcomp = c("dataFrame"),
input_fmt_SV = "Text",
EnzymeType = "SE",
bperrorindel = 3000, bperrorinvtrans = 50000)
ogene <- as.character(datcomp$OverlapGenes_strand_perc)
datogenes <- overlappingGenes (rr, ogene)

```

OverlapRNAseq

*Extract Read counts for genes that overlap SVs.***Description**

Extract Read counts for genes that overlap SVs.

**Usage**

```

OverlapRNAseq(
  gnsOverlap,
  SVID,
  RNASeqData,
  pattern_Proband = NA,
  pattern_Mother = NA,
  pattern_Father = NA
)

```

**Arguments**

<code>gnsOverlap</code>	character. genes that overlap SV.
<code>SVID</code>	character. ID of the SVs.
<code>RNASeqData</code>	character. Expression of the genes.
<code>pattern_Proband</code>	character. Pattern to identify the proband reads.
<code>pattern_Mother</code>	character. Pattern to identify the mother reads.
<code>pattern_Father</code>	character. Pattern to identify the father reads.

**Value**

Text or Dataframe containing TPM read counts of genes in the family.

**Examples**

```

RNASeqDir = system.file("extdata", package="nanotatoR")
returnMethod="dataFrame"
datRNASeq <- RNAseqcombine(RNASeqDir = RNASeqDir,
returnMethod = returnMethod)
gnsOverlap <- c("AGL")
SVID = 397
datgnovrlap <- OverlapRNAseq(gnsOverlap = gnsOverlap,
SVID = SVID, RNASeqData = datRNASeq,
pattern_Proband = "*_P_*")

```

---

OverlapRNAseq\_solo      *Annotating the Overlapping genes with RNAseq expression*

---

**Description**

Annotating the Overlapping genes with RNAseq expression

**Usage**

```
OverlapRNAseq_solo(gnsOverlap, SVID, RNASeqData, pattern_Proband = NA)
```

**Arguments**

gnsOverlap	character. Vector containing overlapping genes.
SVID	character. SV Index ID.
RNASeqData	dataFrame. RNAseq data with gene names.
pattern_Proband	character. Pattern for proband.

**Value**

Dataframe containing TPM read counts of overlapping genes.

**Examples**

```

RNASeqDir = system.file("extdata", package="nanotatoR")
returnMethod="dataFrame"
datRNASeq <- RNAseqcombine_solo(RNASeqDir = RNASeqDir,
returnMethod = returnMethod)
gnsOverlap <- c("AGL")
SVID = 397
datgnovrlap <- OverlapRNAseq_solo(gnsOverlap = gnsOverlap,
SVID = SVID, RNASeqData = datRNASeq,
pattern_Proband = "*_P_*")

```

---

phenoextractHPO\_mod     *Extract the genes related to a disease or disease alias from HPO database.*

---

### Description

Extract the genes related to a disease or disease alias from HPO database.

### Usage

```
phenoextractHPO_mod(keyword, localPDB.path)
```

### Arguments

keyword	character. character string: keyword, to search a disease, a clinical feature, or a phenotype.
localPDB.path	character. the path of localized public data bases. The default value is set in the working directory.

### Value

subset of HPO and extract the genes and alias for a disease(phenotype), or a clinical feature. Function modified from pheno\_extract\_HPO function VarFromPDB.

### Examples

```
HPO.phenotype = phenoextractHPO_mod("retinoblastoma",
localPDB.path = system.file("extdata", "localPDB", package="nanotatoR"))
```

---

readBNBedFiles     *Reads Bionano Bedfiles*

---

### 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", "HomoSapienGRCH19_lift37_BN.bed", package="nanotatoR")
bed<-readBNBedFiles(BNFile)
```

---

reading_GTR	<i>Reading and parsing gtr database.</i>
-------------	--

---

## Description

Reading and parsing gtr database.

## Usage

```
reading_GTR(  
  gtr,  
  downloadGTR,  
  url_gtr = "ftp://ftp.ncbi.nlm.nih.gov/pub/GTR/data/test_condition_gene.txt"  
)
```

## Arguments

gtr	character gtr database location.
downloadGTR	logical if true, downloads gtr database, . and store data in the gtr location, else reads dataset from gtr location.
url_gtr	character url for gtr database.

## Value

Dataframe representation of gtr database.

## Examples

```
a <- reading_GTR(  
  gtr = system.file("extdata", "gtrDatabase.txt", package="nanotatoR"),  
  downloadGTR = FALSE)
```



---

reading_mim2gene	<i>Reading and parsing OMIM database.</i>
------------------	---

---

**Description**

Reading and parsing OMIM database.

**Usage**

```
reading_mim2gene(omim)
```

**Arguments**

omim                    character omim database location.

**Value**

Dataframe returned containing Omim ID and gene IDs.

**Examples**

```
omim = system.file("extdata", "mim2gene.txt", package="nanotator")
a <- reading_mim2gene(omim = omim)
```

---

readSMAP	<i>Reads SMAP files to extract information from SVMerge</i>
----------	---

---

**Description**

Reads SMAP files to extract information from SVMerge

**Usage**

```
readSMAP(smmap, smmapdata, input_fmt_smmap = c("Text", "dataFrame"))
```

**Arguments**

smmap                    character. Path to SMAP file.  
smmapdata                dataframe. variable for smmap dataset.  
input\_fmt\_smmap        character. input format for smmap text or dataframe.

**Value**

Data Frame or text file. Contains the SMAP information.

**Examples**

```
smapName="NA12878_Q.S_VAP_SVmerge_solo5.txt"
smap = system.file("extdata", smapName, package="nanotatoR")
readSMap(smap, input_fmt_smap = "Text")
```

---

readSMap_DLE	<i>Reads DLE SMAP files to extract information</i>
--------------	--

---

**Description**

Reads DLE SMAP files to extract information

**Usage**

```
readSMap_DLE(smap, smapdata, input_fmt_smap = "Text")
```

**Arguments**

smap                    character. Path to SMAP file.  
smapdata                dataframe. variable for smap dataset.  
input\_fmt\_smap        character. input format for smap text or dataframe.

**Value**

Data Frame or text file. Contains the SMAP information.

**Examples**

```
smapName="GM24385_Ason_DLE1_VAP_trio5.smap"
smap = system.file("extdata", smapName, package="nanotatoR")
readSMap_DLE(smap, input_fmt_smap = "Text")
```

---

RNAseqcombine	<i>Combining the RNAseq reads of family members in a single file.</i>
---------------	---

---

**Description**

Combining the RNAseq reads of family members in a single file.

**Usage**

```
RNAseqcombine(  
  RNASeqDir,  
  returnMethod = c("Text", "dataFrame"),  
  outpath = "",  
  outFileName = ""  
)
```

**Arguments**

RNASeqDir        character. Directory containing RNAseq reads.  
 returnMethod    character. Method of returning Data.  
 outpath         character. Contains file path if Method of return is chosen as Text.  
 outFileNames    character. Output file name.

**Value**

Text or Dataframe containing TPM read counts of genes in the family.

**Examples**

```
## Not run:
RNASeqDir = system.file("extdata", package="nanotatoR")
returnMethod="dataFrame"
datRNASeq <- RNAseqcombine(RNASeqDir = RNASeqDir,
returnMethod = returnMethod)

## End(Not run)
```

---

RNAseqcombine\_solo        *Combining the RNAseq reads of family members in a single file.*

---

**Description**

Combining the RNAseq reads of family members in a single file.

**Usage**

```
RNAseqcombine_solo(
  RNASeqDir,
  returnMethod = c("Text", "dataFrame"),
  outpath = "",
  outFileNames = ""
)
```

**Arguments**

RNASeqDir        character. Directory containing RNAseq reads.  
 returnMethod    character. Method of returning Data.  
 outpath         character. Contains file path if Method of return is chosen as Text.  
 outFileNames    character. Output file name.

**Value**

Text or Dataframe containing TPM read counts of genes in the family.

**Examples**

```
RNASeqDir = system.file("extdata", package="nanotatoR")
returnMethod="dataFrame"
datRNASeq <- RNAseqcombine_solo(RNASeqDir = RNASeqDir,
returnMethod = returnMethod)
```

---

```
run_bionano_filter_SE_duo
```

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

---

**Description**

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

**Usage**

```
run_bionano_filter_SE_duo(
  primaryGenesPresent = TRUE,
  input_fmt_geneList = c("Text", "dataFrame"),
  input_fmt_SV = c("Text", "dataFrame"),
  smap = NULL,
  svData,
  dat_geneList,
  fileName,
  outpath,
  outputFilename = "",
  RZIPpath,
  EnzymeType = c("SVMerge", "SE"),
  outputType = c("Excel", "csv"),
  directoryName,
  fileprefix
)
```

**Arguments**

primaryGenesPresent	boolean Checks whether the primary gene List is present or not.
input_fmt_geneList	character. Choice of gene list input Text or Dataframe.
input_fmt_SV	character. Choice of gene list input Text or Dataframe.
smap	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.

outputpath	Character Directory to the output file.
outputFilename	Character Output filename.
RZIPpath	Character Path for the Rtools Zip package.
EnzymeType	Character. Enzyme type used. Options SVMerge or SE.
outputType	Character. Variants in excel tabs or in different csv files. Options Excel or csv.
directoryName	Character. Directory name where individual SV files will be stored.
fileprefix	Character. fileprefix to use for each of the files in the directory.

**Value**

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

**Examples**

```
## Not run:
smapName <- "GM24385_DLE-1_P_trio_hg19.smap"
outputFilename <- "GM24385_DLE-1_P_trio_hg19_out"
smappath <- system.file("extdata", smapName, package = "nanotatoR")
outputpath <- system.file("extdata", smapName, package = "nanotatoR")
RZIPpath <- system.file("extdata", "zip.exe", package = "nanotatoR")
run_bionano_filter_SE_duo (input_fmt_geneList = c("Text"),
input_fmt_SV = c("Text"),
smap = smappath,
dat_geneList = dat_geneList,
RZIPpath = RZIPpath, EnzymeType = c("SE"),
outputType = c("Excel"),
primaryGenesPresent = FALSE,
outputFilename = outputFilename,
outputpath = outputpath) #'
## End(Not run)
```

---

```
run_bionano_filter_SE_solo
```

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

---

**Description**

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

**Usage**

```
run_bionano_filter_SE_solo(
  input_fmt_geneList = c("Text", "dataFrame"),
  input_fmt_SV = c("Text", "dataFrame"),
  EnzymeType = c("SE", "SVMerge"),
  smap = NULL,
```

```

svData,
dat_geneList,
fileName,
outpath,
outputFilename = "",
RZIPpath,
primaryGenesPresent = TRUE,
outputType = c("Excel", "csv"),
directoryName,
fileprefix
)

```

### Arguments

input_fmt_geneList	character. Choice of gene list input Text or Dataframe.
input_fmt_SV	character. Choice of gene list input Text or Dataframe.
EnzymeType	Character. Enzyme type used. Options SVMerge or SE.
smap	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.
primaryGenesPresent	boolean Checks whether the primary gene List is present or not.
outputType	Character. Variants in excel tabs or in different csv files. Options Excel or csv.
directoryName	Character. Directory name where individual SV files will be stored.
fileprefix	Character. fileprefix to use for each of the files in the directory.

### Value

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

### Examples

```

smapName <- "NA12878_DLE1_VAP_solo5.smap"
outputFilename <- "NA12878_DLE1_VAP_solo5_out"
smappath <- system.file("extdata", smapName, package = "nanotatoR")
outpath <- system.file("extdata", smapName, package = "nanotatoR")
RZIPpath <- system.file("extdata", "zip.exe", package = "nanotatoR")
smap = system.file("extdata", smapName, package="nanotatoR")
bedFile <- system.file("extdata", "HomoSapienGRCH19_lift37.bed", package="nanotatoR")
outpath <- system.file("extdata", package="nanotatoR")
datcomp<-overlapnearestgeneSearch(smap = smap,

```

```

bed=bedFile, inputfmtBed = "bed", outpath,
n = 3, returnMethod_bedcomp = c("dataFrame"),
input_fmt_SV = "Text",
EnzymeType = "SE",
bperrorindel = 3000, bperrorinvtrans = 50000)
hgpath=system.file("extdata", "GRCh37_hg19_variants_2016-05-15.txt", package="nanotatoR")
datDGV <- DGVfrequency (hgpath = hgpath,
  smap_data = datcomp,
  win_indel_DGV = 10000,
  input_fmt_SV = "dataFrame", EnzymeType = "SE",
  perc_similarity_DGV = 0.5,returnMethod="dataFrame")
indelconf = 0.5; invconf = 0.01;transconf = 0.1;
datInf <- internalFrequency_solo(smapdata = datDGV,
  buildSVInternalDB=TRUE, win_indel=10000,
  win_inv_trans=50000,
  labelType = c("SE"), EnzymeType = "SE",
  SE_path = system.file("extdata", "SoloFile/", package="nanotatoR"),
  SE_pattern = "*_DLE1_*", perc_similarity_parents =0.9,
  Samplecodes = system.file("extdata", "nanotatoR_sample_codes.csv", package="nanotatoR"),
  mergeKey = system.file("extdata", "nanotatoR_control_sample_codes.csv", package="nanotatoR"),
  mergedKeyoutpath = system.file("extdata", package="nanotatoR"),
  mergedKeyFname = "Sample_index.csv",
  indexfile = system.file("extdata", mergedKeyFname, package="nanotatoR"),
  perc_similarity = 0.5, indelconf = 0.5, invconf = 0.01,
  transconf = 0.1, limsize = 1000, win_indel_parents = 5000,
  win_inv_trans_parents=40000,
  returnMethod="dataFrame", input_fmt_SV = "dataFrame")
path <- system.file("extdata", "Bionano_config/", package = "nanotatoR")
pattern <- "*_hg19_*"
directoryName <- system.file("extdata", package="nanotatoR")
datBNDB <- BNDBfrequency(smapdata = datInf,
  buildBNInternalDB=TRUE,
  input_fmt_SV = "dataFrame",
  dbOutput="dataframe",
  BNDBpath = path,
  BNDBpattern = pattern,
  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("dataFrame"),
  EnzymeType = c("SE"))
decipherpath = system.file("extdata", "population_cnv.txt", package="nanotatoR")
datdecipher <- Decipherfrequency (decipherpath = decipherpath,
  smap_data = datBNDB, win_indel = 10000,
  perc_similarity = 0.5,returnMethod="dataFrame",
  input_fmt_SV = "dataFrame", EnzymeType = c("SE"))
run_bionano_filter_SE_solo (input_fmt_geneList = c("Text"),
  input_fmt_SV = c("dataFrame"),

```

```
svData = datdecipher,
dat_geneList = dat_geneList,
RZIPpath = RZIPpath, EnzymeType = c("SE"),
outputType = c("csv"),
primaryGenesPresent = FALSE,
directoryName = directoryName,
fileprefix = "AnnotatedSamplesNA12878_DLE")
```

---

run\_bionano\_filter\_SE\_Trio

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

---

### Description

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

### Usage

```
run_bionano_filter_SE_Trio(
  primaryGenesPresent = TRUE,
  input_fmt_geneList = c("Text", "dataFrame"),
  input_fmt_SV = c("Text", "dataFrame"),
  smap = NULL,
  svData,
  dat_geneList,
  fileName,
  outpath,
  outputFilename = "",
  RZIPpath,
  outputType = c("Excel", "csv"),
  directoryName,
  fileprefix,
  EnzymeType = c("SVMerge", "SE")
)
```

### Arguments

primaryGenesPresent	boolean Checks whether the primary gene List is present or not.
input_fmt_geneList	character. Choice of gene list input Text or Dataframe.
input_fmt_SV	character. Choice of gene list input Text or Dataframe.
smap	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.
outputpath	Character Directory to the output file.
outputFilename	Character Output filename.
RZIPpath	Character Path for the Rtools Zip package.
outputType	Character. Variants in excel tabs or in different csv files. Options Excel or csv.
directoryName	Character. Directory name where individual SV files will be stored.
fileprefix	Character. fileprefix to use for each of the files in the directory.
EnzymeType	Character. Enzyme type used. Options SVmerge or SE.

### Value

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

### Examples

```

smapName <- "GM24385_Ason_DLE1_VAP_trio5.smap"
outputFilename <- "GM24385_Ason_DLE1_VAP_trio5_out"
smappath <- system.file("extdata", smapName, package = "nanotatoR")
outputpath <- system.file("extdata", smapName, package = "nanotatoR")
RZIPpath <- system.file("extdata", "zip.exe", package = "nanotatoR")
smap = system.file("extdata", smapName, package="nanotatoR")
bedFile <- system.file("extdata", "HomoSapienGRCH19_lift37.bed", package="nanotatoR")
outputpath <- system.file("extdata", package="nanotatoR")
directoryName <- system.file("extdata", package="nanotatoR")
datcomp<-overlapnearestgeneSearch(smap = smap,
  bed=bedFile, inputfmtBed = "bed", outputpath,
  n = 3, returnMethod_bedcomp = c("dataFrame"),
  input_fmt_SV = "Text",
  EnzymeType = "SE",
  berrorindel = 3000, berrorinvtrans = 50000)
hgpath=system.file("extdata", "GRCh37_hg19_variants_2016-05-15.txt", package="nanotatoR")
datDGV <- DGVfrequency (hgpath = hgpath,
  smap_data = datcomp,
  win_indel_DGV = 10000,
  input_fmt_SV = "dataFrame",EnzymeType = "SE",
  perc_similarity_DGV = 0.5,returnMethod="dataFrame")
indelconf = 0.5; invconf = 0.01;transconf = 0.1;
datInf <- internalFrequencyTrio_Duo(smapdata = datDGV,
  buildSVInternalDB=TRUE, win_indel=10000,
  win_inv_trans=50000,
  labelType = c("SE"), EnzymeType = "SE",
  SE_path = system.file("extdata", "SoloFile/", package="nanotatoR"),
  SE_pattern = "*_DLE1_*", perc_similarity_parents =0.9,
  Samplecodes = system.file("extdata", "nanotatoR_sample_codes.csv", package="nanotatoR"),
  mergeKey = system.file("extdata", "nanotatoR_control_sample_codes.csv", package="nanotatoR"),
  mergedKeyoutputpath = system.file("extdata", package="nanotatoR"),
  mergedKeyFname = "Sample_index.csv",
  indexfile = system.file("extdata", mergedKeyFname, package="nanotatoR"),
  perc_similarity = 0.5, indelconf = 0.5, invconf = 0.01,
  transconf = 0.1, limsize = 1000, win_indel_parents = 5000,

```

```

win_inv_trans_parents=40000,
returnMethod="dataFrame", input_fmt_SV = "dataFrame")
path <- system.file("extdata", "Bionano_config/", package = "nanotatoR")
pattern <- "*_hg19_*"
datBNDB <- BNDBfrequency(smadata = datInf,
  buildBNInternalDB=TRUE,
  input_fmt_SV = "dataFrame",
  dbOutput="dataframe",
  BNDBpath = path,
  BNDBpattern = pattern,
  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("dataFrame"),
  EnzymeType = c("SE"))
decipherpath = system.file("extdata", "population_cnv.txt", package="nanotatoR")
datdecipher <- Decipherfrequency (decipherpath = decipherpath,
  smap_data = datBNDB, win_indel = 10000,
  perc_similarity = 0.5,returnMethod="dataFrame",
  input_fmt_SV = "dataFrame", EnzymeType = c("SE"))
run_bionano_filter_SE_Trio (input_fmt_geneList = c("Text"),
  input_fmt_SV = c("dataFrame"),
  svData = datdecipher,
  dat_geneList = dat_geneList,
  RZIPpath = RZIPpath, EnzymeType = c("SE"),
  outputType = c("csv"),
  primaryGenesPresent = FALSE,
  directoryName = directoryName,
  fileprefix = "AnnotatedSamplesGM24385")

```

---

```
run_bionano_filter_SVMerge_duo
```

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

---

## Description

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

## Usage

```

run_bionano_filter_SVMerge_duo(
  input_fmt_geneList = c("Text", "dataFrame"),
  input_fmt_SV = c("Text", "dataFrame"),
  smap = NULL,

```

```

svData,
dat_geneList,
fileName,
outpath,
outputFilename = "",
RZIPpath,
outputType = c("Excel", "csv"),
primaryGenesPresent = TRUE,
fileprefix,
directoryName,
EnzymeType = c("SVMerge", "SE")
)

```

### Arguments

input_fmt_geneList	character. Choice of gene list input Text or Dataframe.
input_fmt_SV	character. Choice of gene list input Text or Dataframe.
smap	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.
outputType	Character. Variants in excel tabs or in different csv files. Options Excel or csv.
primaryGenesPresent	boolean Checks whether the primary gene List is present or not.
fileprefix	Character. fileprefix to use for each of the files in the directory.
directoryName	Character. Directory name where individual SV files will be stored.
EnzymeType	Character. Enzyme type used. Options SVMerge or SE.

### Value

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

### Examples

```

## Not run:
smapName <- "GM24385_DLE-1_P_trio_hg19.smap"
outputFilename <- "GM24385_DLE-1_P_trio_hg19_out"
smappath <- system.file("extdata", smapName, package = "nanotatoR")
outpath <- system.file("extdata", smapName, package = "nanotatoR")
RZIPpath <- system.file("extdata", "zip.exe", package = "nanotatoR")
run_bionano_filter_SVMerge_duo (input_fmt_geneList = c("Text"),
input_fmt_SV = c("Text"),

```

```

smap = smappath,
dat_geneList = dat_geneList,
RZIPpath = RZIPpath, EnzymeType = c("SVMerge"),
outputType = c("Excel"),
primaryGenesPresent = FALSE,
outputFilename = outputFilename,
outpath = outpath)

## End(Not run)

```

---

```
run_bionano_filter_SVMerge_solo
```

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

---

## Description

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

## Usage

```

run_bionano_filter_SVMerge_solo(
  input_fmt_geneList = c("Text", "dataFrame"),
  input_fmt_SV = c("Text", "dataFrame"),
  EnzymeType = c("SE", "SVMerge"),
  smap = NULL,
  svData,
  dat_geneList,
  fileName,
  outpath,
  outputFilename = "",
  RZIPpath,
  primaryGenesPresent = TRUE,
  outputType = c("Excel", "csv"),
  directoryName,
  fileprefix
)

```

## Arguments

input_fmt_geneList	character. Choice of gene list input Text or Dataframe.
input_fmt_SV	character. Choice of gene list input Text or Dataframe.
EnzymeType	Character. Enzyme type used. Options SVMerge or SE.
smap	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.  
 primaryGenesPresent  
                   boolean Checks whether the primary gene List is present or not.  
 outputType       Character. Variants in excel tabs or in different csv files. Options Excel or csv.  
 directoryName    Character. Directory name where individual SV files will be stored.  
 fileprefix       Character. fileprefix to use for each of the files in the directory.

### Value

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

### Examples

```

smapName <- "NA12878_Q.S_VAP_SVmerge_solo5.txt"
outputFilename <- "NA12878_Q.S_VAP_SVmerge_solo5_out"
smappath <- system.file("extdata", smapName, package = "nanotatoR")
outpath <- system.file("extdata", smapName, package = "nanotatoR")
RZIPpath <- system.file("extdata", "zip.exe", package = "nanotatoR")
smap = system.file("extdata", smapName, package="nanotatoR")
bedFile <- system.file("extdata", "HomoSapienGRCH19_lift37.bed", package="nanotatoR")
outpath <- system.file("extdata", package="nanotatoR")
directoryName <- system.file("extdata", package="nanotatoR")
datcomp<-overlapnearestgeneSearch(smap = smap,
  bed=bedFile, inputfmtBed = "bed", outpath,
  n = 3, returnMethod_bedcomp = c("dataFrame"),
  input_fmt_SV = "Text",
  EnzymeType = "SVMerge",
  bperrorindel = 3000, bperrorinvtrans = 50000)
hgpath=system.file("extdata", "GRCh37_hg19_variants_2016-05-15.txt", package="nanotatoR")
datDGV <- DGVfrequency (hgpath = hgpath,
  smap_data = datcomp,
  win_indel_DGV = 10000,
  EnzymeType = "SVMerge",
  input_fmt_SV = "dataFrame",
  perc_similarity_DGV = 0.5,returnMethod="dataFrame")
indelconf = 0.5; invconf = 0.01;transconf = 0.1;
datInf <- internalFrequency_solo(smapdata = datDGV,
  buildSVInternalDB=TRUE, win_indel=10000,
  win_inv_trans=50000,
  labelType = c("SE"),
  EnzymeType = "SVMerge",
  SE_path = system.file("extdata", "SoloFile/", package="nanotatoR"),
  SE_pattern = "*_DLE1_*", perc_similarity_parents =0.9,
  Samplecodes = system.file("extdata", "nanotatoR_sample_codes.csv", package="nanotatoR"),
  mergeKey = system.file("extdata", "nanotatoR_control_sample_codes.csv", package="nanotatoR"),

```

```

mergedKeyouthpath = system.file("extdata", package="nanotatoR"),
mergedKeyFname = "Sample_index.csv",
indexfile = system.file("extdata", mergedKeyFname, package="nanotatoR"),
perc_similarity = 0.5, indelconf = 0.5, invconf = 0.01,
transconf = 0.1, limsize = 1000, win_indel_parents = 5000,
win_inv_trans_parents=40000,
returnMethod="dataFrame", input_fmt_SV = "dataFrame")
path <- system.file("extdata", "Bionano_config/", package = "nanotatoR")
pattern <- "*_hg19_*"
datBNDB <- BNDBfrequency(smapdata = datInf,
  buildBNInternalDB=TRUE,
  input_fmt_SV = "dataFrame",
  dbOutput="dataframe",
  EnzymeType = "SVMerge",
  BNDBpath = path,
  BNDBpattern = pattern,
  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("dataFrame"))
decipherpath = system.file("extdata", "population_cnv.txt", package="nanotatoR")
datdecipher <- Decipherfrequency (decipherpath = decipherpath,
  smap_data = datBNDB, win_indel = 10000,
  perc_similarity = 0.5,returnMethod="dataFrame",
  input_fmt_SV = "dataFrame", EnzymeType = c("SVMerge"))
run_bionano_filter_SVMerge_solo (input_fmt_geneList = c("Text"),
  input_fmt_SV = c("dataFrame"),
  svData = datdecipher,
  dat_geneList = dat_geneList,
  RZIPpath = RZIPpath, EnzymeType = c("SVMerge"),
  outputType = c("csv"),
  primaryGenesPresent = FALSE,
  directoryName = directoryName,
  fileprefix = "AnnotatedSamplesNA12878_SVMerge")

```

---

run\_bionano\_filter\_SVMerge\_Trio

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

---

## Description

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

**Usage**

```
run_bionano_filter_SVMerge_Trio(
  input_fmt_geneList = c("Text", "dataFrame"),
  input_fmt_SV = c("Text", "dataFrame"),
  EnzymeType = c("SVMerge", "SE"),
  smap = NULL,
  svData,
  dat_geneList,
  fileName,
  outpath,
  outputFilename = "",
  RZIPpath,
  primaryGenesPresent = TRUE,
  outputType = c("Excel", "csv"),
  directoryName,
  fileprefix
)
```

**Arguments**

input_fmt_geneList	character. Choice of gene list input Text or Dataframe.
input_fmt_SV	character. Choice of gene list input Text or Dataframe.
EnzymeType	Character. Enzyme type used. Options SVMerge or SE.
smap	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.
primaryGenesPresent	boolean Checks whether the primary gene List is present or not.
outputType	Character. Variants in excel tabs or in different csv files. Options Excel or csv.
directoryName	Character. Directory name where individual SV files will be stored.
fileprefix	Character. fileprefix to use for each of the files in the directory.

**Value**

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

**Examples**

```
## Not run:
smapName <- "GM24385_DLE-1_P_trio_hg19.smap"
outputFilename <- "GM24385_DLE-1_P_trio_hg19_out"
smappath <- system.file("extdata", smapName, package = "nanotatoR")
outpath <- system.file("extdata", smapName, package = "nanotatoR")
RZIPpath <- system.file("extdata", "zip.exe", package = "nanotatoR")
run_bionano_filter_SVMerge_Trio (input_fmt_geneList = c("Text"),
input_fmt_SV = c("Text"),
smap = smappath,
dat_geneList = dat_geneList,
RZIPpath = RZIPpath, EnzymeType = c("SVMerge"),
outputType = c("Excel"),
primaryGenesPresent = FALSE,
outputFilename = outputFilename,
outpath = outpath)##'
## End(Not run)
```

---

SVexpression\_duo\_trio *Extract Read counts for genes that are near or overlapping SVs.*

---

**Description**

Extract Read counts for genes that are near or overlapping SVs.

**Usage**

```
SVexpression_duo_trio(
  input_fmt_SV = c("Text", "dataFrame"),
  smapdata,
  smappath,
  input_fmt_RNASeq = c("Text", "dataFrame"),
  RNASeqData,
  RNASeqPATH,
  outputfmt = c("Text", "datFrame"),
  pattern_Proband = NA,
  pattern_Mother = NA,
  pattern_Father = NA,
  EnzymeType = c("SVMerge", "SE")
)
```

**Arguments**

input_fmt_SV	character. genes that are upstream and/or downstream of SV. input_fmt_RNASeq
smapdata	dataframe. smap data in dataframe format.
smappath	character. smap path.



```

input_fmt_RNASeq      character. input format of RNASEQ data. Text or dataframe.
RNASeqData            character. Expression of the genes.
RNASeqPATH            character. RNASEQ path.
outputfmt             character. Output format choice dataframe or text.
pattern_Proband       character. Pattern to identify the proband reads.
pattern_Mother        character. Pattern to identify the mother reads.
pattern_Father        character. Pattern to identify the father reads.
EnzymeType            character. Enzyme used. option "Dual" or "DLE".

```

**Value**

Text or Dataframe containing TPM read counts of genes in the family.

**Examples**

```

## Not run:
RNASeqDir = system.file("extdata", package="nanotatoR")
returnMethod="dataFrame"
datRNASeq <- RNAseqcombine(RNASeqDir = RNASeqDir,
returnMethod = returnMethod)
smapName="NA12878_DLE1_VAP_solo5.smap"
smap = system.file("extdata", smapName, package="nanotatoR")
datcomp<-overlapnearestgeneSearch(smap = smap,
  bed=bedFile, inputfmtBed = "BED", outpath,
  n = 3, returnMethod_bedcomp = c("dataFrame"),
  input_fmt_SV = "Text",
  EnzymeType = "SVMerge",
  bperrorindel = 3000,
  bperrorinvtrans = 50000)
datRNASeq1 <- SVexpression(
  input_fmt_SV=c("dataFrame"),
  input_fmt_RNASeq=c("dataFrame"),
  RNASeqData = datRNASeq,
  outputfmt=c("datFrame"),
  pattern_Proband = "*_P_*", EnzymeType = c("SVMerge"))

## End(Not run)

```

---

SVexpression_solo	<i>Annotating the Overlapping and Non-Overlapping genes with RNAseq expression</i>
-------------------	--

---

**Description**

Annotating the Overlapping and Non-Overlapping genes with RNAseq expression

**Usage**

```
SVexpression_solo(
  input_fmt_SV = c("Text", "dataFrame"),
  smapdata,
  smappath,
  input_fmt_RNASeq = c("Text", "dataFrame"),
  RNASeqData,
  RNASeqPATH,
  outputfmt = c("Text", "datFrame"),
  pattern_Proband = NA,
  EnzymeType = c("SVMerge", "SE")
)
```

**Arguments**

input_fmt_SV	character. Input format of the SV data. Options "Text" or "DataFrame".
smapdata	dataframe. SV data dataframe.
smappath	character. smap path.
input_fmt_RNASeq	character. Input format of the RNASeq data. Options "Text" or "DataFrame"..
RNASeqData	dataFrame. RNAseq data with gene names.
RNASeqPATH	character. RNAseq dataset path .
outputfmt	character. Output format of the result. Options "Text" or "DataFrame"..
pattern_Proband	character. Pattern for proband.
EnzymeType	character. Enzyme used. option "SVMerge" or "SE".

**Value**

Dataframe Annotated dataframe with RNASeq data.

**Examples**

```
RNASeqDir = system.file("extdata", package="nanotatoR")
returnMethod="dataFrame"
datRNASeq <- RNAseqcombine_solo(RNASeqDir = RNASeqDir,
returnMethod = returnMethod)
smapName="NA12878_DLE1_VAP_solo5.smap"
smap = system.file("extdata", smapName, package="nanotatoR")
smap = system.file("extdata", smapName, package="nanotatoR")
bedFile <- system.file("extdata", "HomoSapienGRCH19_lift37.bed", package="nanotatoR")
outpath <- system.file("extdata", package="nanotatoR")
datcomp<-overlapnearestgeneSearch(smap = smap,
  bed=bedFile, inputfmtBed = "bed", outpath,
  n = 3, returnMethod_bedcomp = c("dataFrame"),
  input_fmt_SV = "Text",
  EnzymeType = "SE",
  bperrorindel = 3000,
```

```
bperrorinvtrans = 50000)
datRNASeq1 <- SVexpression_solo (input_fmt_SV=c("dataFrame"),
  smapdata = datcomp,
  input_fmt_RNASeq=c("dataFrame"),
  RNASeqData = datRNASeq,
  outputfmt=c("datFrame"),
  pattern_Proband = "*_P_*", EnzymeType = c("SE"))
datRNASeq1[1,]
```

# Index

[BNDBfrequency](#), [3](#)  
[buildrunBNBedFiles](#), [5](#)

[clinvar\\_gene](#), [6](#)

[Decipherfrequency](#), [6](#)  
[DGVfrequency](#), [8](#)

[extract\\_clinvar\\_mod](#), [9](#)

[FamilyInfoPrep](#), [10](#)

[gene\\_extraction](#), [11](#)  
[gene\\_list\\_generation](#), [11](#)  
[gtr\\_gene](#), [13](#)

[internalFrequency\\_solo](#), [16](#)  
[internalFrequencyTrio\\_Duo](#), [13](#)

[makeInternalBNDatabase](#), [19](#)  
[merging\\_SE\\_SVMerge](#), [21](#)  
[mergingSMAP\\_SE](#), [20](#)  
[mergingSMAP\\_SVMerge](#), [20](#)

[nanotatoR](#), [22](#)  
[nanotatoR\\_Duo\\_SVmerge](#), [23](#)  
[nanotatoR\\_main\\_Duo\\_SE](#), [28](#)  
[nanotatoR\\_main\\_Solo\\_SE](#), [33](#)  
[nanotatoR\\_main\\_Solo\\_SVmerge](#), [38](#)  
[nanotatoR\\_main\\_Trio\\_SE](#), [43](#)  
[nanotatoR\\_SVmerge\\_Trio](#), [47](#)  
[nonOverlapGenes](#), [52](#)  
[nonOverlappingDNGenes](#), [54](#)  
[nonOverlappingUPGenes](#), [55](#)  
[nonOverlapRNAseq](#), [55](#)  
[nonOverlapRNAseq\\_solo](#), [56](#)

[omim\\_gene](#), [57](#)  
[overlapGenes](#), [58](#)  
[overlapnearestgeneSearch](#), [59](#)  
[overlappingGenes](#), [60](#)

[OverlapRNAseq](#), [61](#)  
[OverlapRNAseq\\_solo](#), [62](#)

[phenoextractHPO\\_mod](#), [63](#)

[readBNBedFiles](#), [63](#)  
[reading\\_GTR](#), [64](#)  
[reading\\_mim2gene](#), [65](#)  
[readSMap](#), [65](#)  
[readSMap\\_DLE](#), [66](#)  
[RNAseqcombine](#), [66](#)  
[RNAseqcombine\\_solo](#), [67](#)  
[run\\_bionano\\_filter\\_SE\\_duo](#), [68](#)  
[run\\_bionano\\_filter\\_SE\\_solo](#), [69](#)  
[run\\_bionano\\_filter\\_SE\\_Trio](#), [72](#)  
[run\\_bionano\\_filter\\_SVMerge\\_duo](#), [74](#)  
[run\\_bionano\\_filter\\_SVMerge\\_solo](#), [76](#)  
[run\\_bionano\\_filter\\_SVMerge\\_Trio](#), [78](#)

[SVexpression\\_duo\\_trio](#), [80](#)  
[SVexpression\\_solo](#), [81](#)