Package 'Neoantimon'

June 27, 2020

Type Package

Title Neoantimon: A multifunctional R package for identification of tumor-specific neoantigens

Version 2.1.1

Date 2020-6-24

Author Takanori Hasegawa

Maintainer Takanori Hasegawa <t-hasegw@ims.u-tokyo.ac.jp>

Description This Package has been developed to calculate candidates neoantigens from Mutation Data (.vcf,.txt,string).

License MIT + file LICENSE

VignetteBuilder knitr

Encoding UTF-8

Depends R (>= 3.3.0)

biocViews

Imports ensemblVEP, devtools, graphics, grDevices, stats, utils, biomaRt

Suggests data.table, knitr, rmarkdown

LazyData FALSE

RoxygenNote 7.1.0

R topics documented:

Export_Summary_Entire_Fragments 2
Export_Summary_Fragments 3
Export_Summary_IndelSV 4
Export_Summary_IndelSV_perFragments 5
Export_Summary_SNV 6
MainEntireRegionClass1 7
MainEntireRegionClass2 9
MainINDELClass1 11
MainINDELClass2 14
MainSeqFragmentClass1 18
MainSeqFragmentClass2 20
MainSNVClass1 22
MainSNVClass2 26
MainSVFUSIONClass1 30

MainSVFUSIONClass2 . . . 33
sample_copynum . . . 37
sample_hla_table_c1 . . . 37
sample_hla_table_c2 . . . 37
sample_refFlat.grch37 . . . 38
sample_refMrna.grch37.fa . . . 38
sample_result_INDEL_CLASS1_ALL . . . 38
sample_result_INDEL_CLASS2_ALL . . . 39
sample_result_SeqFragment_CLASS1_ALL . . . 39
sample_result_SeqFragment_CLASS2_ALL . . . 39
sample_result_SNV_CLASS1_ALL . . . 39
sample_result_SNV_CLASS2_ALL . . . 40
sample_result_SVFusion_CLASS1_ALL . . . 40
sample_result_SVFusion_CLASS2_ALL . . . 40
sample_rna_exp . . . 40
sample_sv_bnd . . . 41
sample_vcf.annovar . . . 41
sample_vcf.snps . . . 41
sample_vcf.vep . . . 42
TestAnalysis . . . 42

Index **43**

`Export_Summary_Entire_Fragments` *Export Summary Count from Indel/SV Results*

Description

Export Summary Count from Indel/SV Results

Usage

```
Export_Summary_Entire_Fragments(
  Input,
  Mut_IC50_th = NA,
  Mut_Rank_th = NA,
  Total_RNA_th = NA,
  Tumor_RNA_th = NA,
  MutRatio_th = NA,
  WriteLongIndel = NA,
  DupCount = FALSE
)
```

Arguments

`Input`	Input file generated from MainSNVClass1,2.
`Mut_IC50_th`	The threshold for mutant peptide to be neoantigen by IC50.
`Mut_Rank_th`	The threshold for mutant peptide to be neoantigen by Rank.
`Total_RNA_th`	The total RNA expression threshold.
`Tumor_RNA_th`	The tumor specific RNA expression threshold.

MutRatio_th	The mutation ratio threshold.
WriteLongIndel	If setting a file name, Write Long Indels of which the p-value is less than 0.05.
DupCount	Count for each different HLA type

Value

Num_Alteration The number of evaluated alterations.

Num_Alteration_Generating_NeoAg The number of evaluated alterations that can generate neoantigen.

Num_Peptide The number of evaluated peptifdes.

Num_Peptide_Generating_NeoAg The number of evaluated peptides that can be neoantigen.

Export_Summary_Fragments

Export Summary Count from Indel/SV Results

Description

Export Summary Count from Indel/SV Results

Usage

```
Export_Summary_Fragments(
  Input,
  Mut_IC50_th = NA,
  Mut_Rank_th = NA,
  Total_RNA_th = NA,
  Tumor_RNA_th = NA,
  MutRatio_th = NA,
  WriteLongIndel = NA,
  DupCount = FALSE
)
```

Arguments

Input	Input file generated from MainSNVClass1,2.
Mut_IC50_th	The threshold for mutant peptide to be neoantigen by IC50.
Mut_Rank_th	The threshold for mutant peptide to be neoantigen by Rank.
Total_RNA_th	The total RNA expression threshold.
Tumor_RNA_th	The tumor specific RNA expression threshold.
MutRatio_th	The mutation ratio threshold.
WriteLongIndel	If setting a file name, Write Long Indels of which the p-value is less than 0.05.
DupCount	Count for each different HLA type

Value

Num_Alteration The number of evaluated alterations.

Num_Alteration_Generating_NeoAg The number of evaluated alterations that can generate neoantigen.

Num_Peptide The number of evaluated peptifdes.

Num_Peptide_Generating_NeoAg The number of evaluated peptides that can be neoantigen.

`Export_Summary_IndelSV` *Export Summary Count from Indel/SV Results*

Description

Export Summary Count from Indel/SV Results

Usage

```
Export_Summary_IndelSV(
  Input,
  Mut_IC50_th = NA,
  Mut_Rank_th = NA,
  Total_RNA_th = NA,
  Tumor_RNA_th = NA,
  MutRatio_th = NA,
  Weight = NA,
  WriteLongIndel = NA,
  IgnoreLongIndel = 0,
  DupCount = FALSE
)
```

Arguments

`Input`	Input file generated from MainSNVClass1,2.
`Mut_IC50_th`	The threshold for mutant peptide to be neoantigen by IC50.
`Mut_Rank_th`	The threshold for mutant peptide to be neoantigen by Rank.
`Total_RNA_th`	The total RNA expression threshold.
`Tumor_RNA_th`	The tumor specific RNA expression threshold.
`MutRatio_th`	The mutation ratio threshold.
`Weight`	The weight for alterations.
`WriteLongIndel`	If setting a file name, Write Long Indels of which the p-value is less than 0.05.
`IgnoreLongIndel`	Ignore Indels of which p-value is less than the indicated value for counting.
`DupCount`	Count for each different HLA type

Value

Num_Alteration The number of evaluated alterations.

Num_Alteration_Generating_NeoAg The number of evaluated alterations that can generate neoantigen.

Num_Peptide The number of evaluated peptifdes.

Num_Peptide_Generating_NeoAg The number of evaluated peptides that can be neoantigen.

`Export_Summary_IndelSV_perFragments` *Export Summary Count from Indel/SV Results*

Description

Export Summary Count from Indel/SV Results

Usage

```
Export_Summary_IndelSV_perFragments(
  Input,
  Mut_IC50_th = NA,
  Mut_Rank_th = NA,
  Total_RNA_th = NA,
  Tumor_RNA_th = NA,
  MutRatio_th = NA,
  Weight = NA,
  WriteLongIndel = NA,
  IgnoreLongIndel = 0,
  DupCount = FALSE
)
```

Arguments

`Input`	Input file generated from MainSNVClass1,2.
`Mut_IC50_th`	The threshold for mutant peptide to be neoantigen by IC50.
`Mut_Rank_th`	The threshold for mutant peptide to be neoantigen by Rank.
`Total_RNA_th`	The total RNA expression threshold.
`Tumor_RNA_th`	The tumor specific RNA expression threshold.
`MutRatio_th`	The mutation ratio threshold.
`Weight`	The weight for alterations.
`WriteLongIndel`	If setting a file name, Write Long Indels of which the p-value is less than 0.05.
`IgnoreLongIndel`	Ignore Indels of which p-value is less than the indicated value for counting.
`DupCount`	Count for each different HLA type

Value

Num_Alteration The number of evaluated alterations.

Num_Alteration_Generating_NeoAg The number of evaluated alterations that can generate neoantigen.

Num_Peptide The number of evaluated peptifdes.

Num_Peptide_Generating_NeoAg The number of evaluated peptides that can be neoantigen.

`Export_Summary_SNV` *Export Summary Count from SNV Results*

Description

Export Summary Count from SNV Results

Usage

```
Export_Summary_SNV(
  Input,
  Mut_IC50_th = NA,
  Mut_Rank_th = NA,
  Wt_IC50_th = NA,
  Wt_Rank_th = NA,
  Total_RNA_th = NA,
  Tumor_RNA_th = NA,
  MutRatio_th = NA,
  DupCount = FALSE
)
```

Arguments

`Input`	Input file generated from MainSNVClass1,2.
`Mut_IC50_th`	The threshold for mutant peptide to be neoantigen.
`Mut_Rank_th`	The threshold for mutant peptide to be neoantigen.
`Wt_IC50_th`	The threshold for wt peptide to be neoantigen.
`Wt_Rank_th`	The threshold for wt peptide to be neoantigen.
`Total_RNA_th`	The total RNA expression threshold.
`Tumor_RNA_th`	The tumor specific RNA expression threshold.
`MutRatio_th`	The mutation ratio threshold.
`DupCount`	Count for each different HLA type

Value

Num_Alteration The number of evaluated alterations.

Num_Alteration_Generating_NeoAg The number of evaluated alterations that can generate neoantigen.

Num_Peptide The number of evaluated peptifdes.

Num_Peptide_Generating_NeoAg The number of evaluated peptides that can be neoantigen.

MainEntireRegionClass1 *Calculate A Set All Neoantigen Candidates from A Given Gene Symbol and nm_id for MHC Class1 (Not yet stably available)*

Description

Calculate A Set All Neoantigen Candidates from A Given Gene Symbol and nm_id for MHC Class1 (Not yet stably available)

Usage

```
MainEntireRegionClass1(
  input_nm_id,
  group_ids = seq(1:length(input_nm_id)),
  hla_file = "here_is_a_table",
  hla_types = NA,
  file_name_in_hla_table = NA,
  refflat_file = paste(hmdir, "lib/refFlat.txt", sep = "/"),
  refmrna_file = paste(hmdir, "lib/refMrna.fa", sep = "/"),
  hmdir = getwd(),
  job_id = "ID",
  export_dir = paste("result", job_id, "EntireRegion1", sep = "."),
  netMHCpan_dir = paste(hmdir, "lib/netMHCpan-4.0/netMHCpan", sep = "/"),
  peptide_length = c(8, 9, 10, 11, 12, 13),
  reading_frame = 1,
  CalculateIC50 = FALSE,
  ignore_short = TRUE
)
```

Arguments

input_nm_id	(Required) An input amino acid sequence indicated as NM_ID
group_ids	flag to cluster the same group
hla_file	A tab separated file indicating HLA types. The 1st column is input_file name, and the following columns indicate HLA types. See by data(sample_hla_table_c1); sample_hla_table_c1;
hla_types	Set a list of HLA types
file_name_in_hla_table	If the name (1st column) in HLA table is not the same as input_file, indicate the corresponding name (Default=input_file).
refflat_file	refFlat file to be used in constructing peptide. (Default=paste(hmdir, "lib/refFlat.txt", sep=""). See "https://github.com/hase62/Neoantimon"
refmrna_file	refMrna file to be used in constructing peptide (Default=paste(hmdir, "lib/refMrna.fa", sep=""). See "https://github.com/hase62/Neoantimon"
hmdir	Home directory for the analysis (Default = getwd()).

job_id	Job-id to be attached in output files (Default = "NO_job_id").
export_dir	The directory will be stored results (Default = "paste("result", file_name_in_hla_table, job_id, sep=".")")
netMHCpan_dir	The file directory to netMHCpan (Default="lib/netMHCpan-4.0/netMHCpan").
peptide_length	Peptide Length to be generated (Default = 8,9,10,11,12,13).
reading_frame	The starting frame of the input sequence (Default = 1)
CalculateIC50	Whether Calculate IC50 by NetMHCpan or not.
ignore_short	Ignore to output results of Short Peptide Less Than min(peptide_length)

Value

void (Calculated Neoantigen Files will be generated as .tsv files.):

HLA: HLA type used to calculate neoantigen.

Pos: The position of a fraction of peptide used to be evaluated from the full-length peptide.

Gene: Gene symbol used to be evaluated in NetMHCpan.

Evaluated_Mutant_Peptide: The mutant peptide to be evaluated.

Evaluated_Mutant_Peptide_Core: The core peptide of the mutant peptide to be evaluated in NetMHCpan.

Mut_IC50: IC50 value for evaluated mutant peptide.

Mut_Rank: Rank value for evaluated mutanat peptide.

Chr: Chromosome Number of the mutation.

NM_ID: NM_ID used to construct peptides from the mutation.

Change: The annotation to be described in .vcf file.

Ref: reference type nucleic acid base.

Alt: alternative type nucleic acid base.

Prob: A probability of reference nucleic acid base described in .vcf file.

Mutation_Prob: A probability of alternative nucleic acid base described in .vcf file.

Exon_Start: The exon start position of the corrsponding NM_ID.

Exon_End: The exon end position of the corrsponding NM_ID.

Mutation_Position: The mutation position of the corrsponding NM_ID.

Total_Depth: The sum depth of the reference and alternative nucleic acid base.

Tumor_Depth: The depth of the alternative nucleic acid base.

Wt_Peptide: The full-length of the wild-type peptide.

Mutant_Peptide: The full-length of the mutant peptide.

Total_RNA: The expression amount of the corresponding RNA.

Tumor_RNA_Ratio: The variant allele frequency of the corresponding RNA.

Tumor_RNA: The modified amount of the corresponding RNA level based on RNA Reads.

Tumor_RNA_based_on_DNA: The modified amount of the corresponding RNA level based on DNA Reads.

MutRatio: The mean value of the cancer cell fraction probability.

MutRatio_Min: The 1% percentile of the cancer cell fraction probability.

MutRatio_Max: The 99% percentile of the cancer cell fraction probability.

`MainEntireRegionClass2` *Calculate A Set All Neoantigen Candidates from A Given Gene Symbol and nm_id for MHC Class2 (Not yet stably available)*

Description

Calculate A Set All Neoantigen Candidates from A Given Gene Symbol and nm_id for MHC Class2 (Not yet stably available)

Usage

```
MainEntireRegionClass2(
  input_nm_id,
  group_ids = seq(1:length(input_nm_id)),
  hla_file = "here_is_a_table",
  hla_types = NA,
  file_name_in_hla_table = NA,
  refflat_file = paste(hmdir, "lib/refFlat.txt", sep = "/"),
  refmrna_file = paste(hmdir, "lib/refMrna.fa", sep = "/"),
  hmdir = getwd(),
  job_id = "ID",
  export_dir = paste("result", job_id, "EntireRegion2", sep = "."),
  netMHCIIpan_dir = paste(hmdir, "lib/netMHCIIpan-3.1/netMHCIIpan", sep = "/"),
  peptide_length = c(15),
  reading_frame = 1,
  CalculateIC50 = FALSE,
  ignore_short = TRUE
)
```

Arguments

`input_nm_id`	(Required) An input amino acid sequence indicated as NM_ID
`group_ids`	flag to cluster the same group
`hla_file`	A tab separated file indicating HLA types. The 1st column is input_file name, and the following columns indicate HLA types. See by data(sample_hla_table_c1); sample_hla_table_c1;
`hla_types`	Set a list of HLA types
`file_name_in_hla_table`	If the name (1st column) in HLA table is not the same as input_file, indicate the corresponding name (Default=input_file).
`refflat_file`	refFlat file to be used in constructing peptide. (Default=paste(hmdir, "lib/refFlat.txt", sep=""). See "https://github.com/hase62/Neoantimon"
`refmrna_file`	refMrna file to be used in constructing peptide (Default=paste(hmdir, "lib/refMrna.fa", sep=""). See "https://github.com/hase62/Neoantimon"
`hmdir`	Home directory for the analysis (Default = getwd()).

job_id	Job-id to be attached in output files (Default = "NO_job_id").
export_dir	The directory will be stored results (Default = "paste("result", file_name_in_hla_table, job_id, sep=".")")
netMHCIIpan_dir	The file directory to netMHCpan (Default="lib/netMHCIIpan-3.2/netMHCIIpan").
peptide_length	Peptide Length to be generated (Default = 8,9,10,11,12,13).
reading_frame	The starting frame of the input sequence (Default = 1)
CalculateIC50	Whether Calculate IC50 by NetMHCpan or not.
ignore_short	Ignore to output results of Short Peptide Less Than min(peptide_length)

Value

void (Calculated Neoantigen Files will be generated as .tsv files.):

HLA: HLA type used to calculate neoantigen.

Pos: The position of a fraction of peptide used to be evaluated from the full-length peptide.

Gene: Gene symbol used to be evaluated in NetMHCpan.

Evaluated_Mutant_Peptide: The mutant peptide to be evaluated.

Evaluated_Mutant_Peptide_Core: The core peptide of the mutant peptide to be evaluated in NetMHCpan.

Mut_IC50: IC50 value for evaluated mutant peptide.

Mut_Rank: Rank value for evaluated mutanat peptide.

Chr: Chromosome Number of the mutation.

NM_ID: NM_ID used to construct peptides from the mutation.

Change: The annotation to be described in .vcf file.

Ref: reference type nucleic acid base.

Alt: alternative type nucleic acid base.

Prob: A probability of reference nucleic acid base described in .vcf file.

Mutation_Prob: A probability of alternative nucleic acid base described in .vcf file.

Exon_Start: The exon start position of the corrsponding NM_ID.

Exon_End: The exon end position of the corrsponding NM_ID.

Mutation_Position: The mutation position of the corrsponding NM_ID.

Total_Depth: The sum depth of the reference and alternative nucleic acid base.

Tumor_Depth: The depth of the alternative nucleic acid base.

Wt_Peptide: The full-length of the wild-type peptide.

Mutant_Peptide: The full-length of the mutant peptide.

Total_RNA: The expression amount of the corresponding RNA.

Tumor_RNA_Ratio: The variant allele frequency of the corresponding RNA.

Tumor_RNA: The modified amount of the corresponding RNA level based on RNA Reads.

Tumor_RNA_based_on_DNA: The modified amount of the corresponding RNA level based on DNA Reads.

MutRatio: The mean value of the cancer cell fraction probability.

MutRatio_Min: The 1% percentile of the cancer cell fraction probability.

MutRatio_Max: The 99% percentile of the cancer cell fraction probability.

MainINDELClass1 *Calculate Neoantigen Candidates on INDELs for MHC Class1*

Description

Calculate Neoantigen Candidates on INDELs for MHC Class1

Usage

```
MainINDELClass1(
  input_annovar_format_file = NA,
  input_vep_format_file = NA,
  input_vcf_format_file_and_vep = NA,
  hla_file = "here_is_a_table",
  hla_types = NA,
  file_name_in_hla_table = "sample",
  refflat_file = paste(hmdir, "lib/refFlat.txt", sep = "/"),
  refmrna_file = paste(hmdir, "lib/refMrna.fa", sep = "/"),
  hmdir = getwd(),
  job_id = "ID",
  export_dir = paste("result", job_id, "INDEL1", sep = "."),
  rnaexp_file = NA,
  rnabam_file = NA,
  cnv_file = NA,
  purity = 1,
  netMHCpan_dir = paste(hmdir, "lib/netMHCpan-4.0/netMHCpan", sep = "/"),
  MHCflurry = NA,
  refdna_file = NA,
  samtools_dir = "samtools",
  bcftools_dir = NA,
  chr_column = NA,
  mutation_start_column = NA,
  mutation_end_column = NA,
  mutation_ref_column = NA,
  mutation_alt_column = NA,
  nm_id_column = NA,
  depth_normal_column = NA,
  depth_tumor_column = NA,
  ambiguous_between_exon = 0,
  ambiguous_codon = 0,
  peptide_length = c(8, 9, 10, 11, 12, 13),
  ignore_short = TRUE,
  SNPs = NA,
  multiple_variants = FALSE
)
```

Arguments

`input_annovar_format_file`
: An input vcf file annotated by ANNOVAR (http://annovar.openbioinformatics.org/en/latest/). You can directly indicate a matrix, which is the same as annovar format vcf file, as input.

See by data(sample_vcf.annovar); sample_vcf.annovar.txt;

`input_vep_format_file`
: An input file annotated by Ensembl Variant Effect Predictor (VEP). You can directly indicate a matrix, which is the same as annovar format VEP file, as input.
See by data(sample_vcf.vep); sample_vcf.vep.txt;

`input_vcf_format_file_and_vep`
: An input vcf file and path to Ensembl Variant Effect Predictor (VEP). Before using this option, please install vep according to the official cite ("https://asia.ensembl.org/info/docs/tool

`hla_file`
: A tab separated file indicating HLA types. The 1st column is input_file name, and the following columns indicate HLA types.
See by data(sample_hla_table_c1); sample_hla_table_c1;

`hla_types`
: Set a list of HLA types

`file_name_in_hla_table`
: If the name (1st column) in HLA table is not the same as input_file, indicate the corresponding name.

`refflat_file`
: refFlat file to be used in constructing peptide. (Default=paste(hmdir, "lib/refFlat.txt", sep="").
See "https://github.com/hase62/Neoantimon"

`refmrna_file`
: refMrna file to be used in constructing peptide (Default=paste(hmdir, "lib/refMrna.fa", sep="").
See "https://github.com/hase62/Neoantimon"

`hmdir`
: Home directory for the analysis (Default = getwd()).

`job_id`
: Job-id to be attached in output files (Default = "NO_job_id").

`export_dir`
: The directory will be stored results (Default = "paste("result", file_name_in_hla_table, job_id, sep=".")")

`rnaexp_file`
: A file including RNA expressions (Default=NA). The 1st, 2nd and 3rd columns are "GeneSymbol Chr:Exonstart-Exonend (locus) ExpressionAmount", respectively. The 1st row should be any header.
See by data(sample_rna_exp); sample_rna_exp;

`rnabam_file`
: RNA bam file to calculate variant allele frequency of RNA at each mutation (Default=NA).

`cnv_file`
: A file including copy number variation to calculate cancer cell fraction probability (CCFP) (Default=NA). The format is according to ASCAT output files. The columns are "SNPName Chromosome Position LogR segmentedLogR BAF segmentedBAF CopyNumber MinorAllele RawCopyNumber" The 1st row should be the above header.
See data(sample_copynum); sample_copynum;

`purity`
: Tumor purity or tumor contents ratio required to calculate CCFP (Default=1).

`netMHCpan_dir`
: The file directory to netMHCpan (Default="lib/netMHCpan-4.0/netMHCpan").

`MHCflurry`
: Output MHCflurry results. Return a list of both results (Default=FALSE).

`refdna_file`
: refdna_file information to be used to calculate RNA VAF (Default=NA).
See "https://github.com/hase62/Neoantimon"

`samtools_dir`
: The file directory to samtools_0_x_x (Default="samtools"). It shouled be indicated when you indicate RNA-bam and try to calculate RNA VAF.

`bcftools_dir` The file directory to netMHCpan (Default="bcftools"). It shouled be indicated when you indicate RNA-bam and try to calculate RNA VAF. samtools 0_x_x includes bcftools in the directory.

`chr_column` The column number describing chromosome number in input_file (Default=NA, but will automatically search "Chr" in header).

`mutation_start_column`
The column number describing mutation start Position in input_file (Default=NA, but will automatically search "Start" in header) .

`mutation_end_column`
The column number describing mutation end Position in input_file (Default=NA, but will automatically search "End" in header).

`mutation_ref_column`
The column number describing mutation Ref in input_file (Default=NA, but will automatically search "Ref" in header).

`mutation_alt_column`
The column number describing mutation Alt in input_file (Default=NA, but will automatically search "Alt" in header).

`nm_id_column` The column number describing NM IDs in input_file such as "SLCO1C1:NM_001145944:exon7:c.692_693insG:p.L231fs" (Default=NA).

`depth_normal_column`
The column number describing the read count from normal cells (Default = NA).

`depth_tumor_column`
The column number describing the read count from tumor cells (Default = NA).

`ambiguous_between_exon`
The maximum number to permit the differences between Exon-Lengths from refFlat and refMrna (Default=0).

`ambiguous_codon`
The maximum number to permit the differences between inputfile- and refMrna-oriented translation start/end position (Default=0).

`peptide_length` Peptide Length to be generated (Default = 8,9,10,11,12,13).

`ignore_short` Ignore to output results of short peptide less than min (peptide_length)

`SNPs` Apply indivisual SNPs on peptides by indicate a vcf file.

`multiple_variants`
Reflect multiple variants on a peptide, e.g., SNVs on frameshift region.

Value

void (Calculated Neoantigen Files will be generated as .tsv files.):

HLA: HLA type used to calculate neoantigen.

Pos: The position of a fraction of peptide used to be evaluated from the full-length peptide.

Gene Gene symbol used to be evaluated in NetMHCpan.

Evaluated_Mutant_Peptide: The mutant peptide to be evaluated.

Evaluated_Mutant_Peptide_Core: The core peptide of the mutant peptide to be evaluated in NetMHCpan.

Mut_IC50: IC50 value for evaluated mutant peptide.

Mut_Rank: Rank value for evaluated mutanat peptide.

Chr: Chromosome Number of the mutation.

NM_ID: NM_ID used to construct peptides from the mutation.

Change: The annotation to be described in .vcf file.

Ref: reference type nucleic acid base.

Alt: alternative type nucleic acid base.

Prob: A probability of reference nucleic acid base described in .vcf file.

Mutation_Prob: A probability of alternative nucleic acid base described in .vcf file.

Exon_Start: The exon start position of the corrsponding NM_ID.

Exon_End: The exon end position of the corrsponding NM_ID.

Mutation_Position: The mutation position of the corrsponding NM_ID.

Total_Depth: The sum depth of the reference and alternative nucleic acid base.

Tumor_Depth: The depth of the alternative nucleic acid base.

Wt_Peptide: The full-length of the wild-type peptide.

Mutant_Peptide: The full-length of the mutant peptide.

Total_RNA: The expression amount of the corresponding RNA.

Tumor_RNA_Ratio: The variant allele frequency of the corresponding RNA.

Tumor_RNA: The modified amount of the corresponding RNA level based on RNA Reads.

Tumor_RNA_based_on_DNA: The modified amount of the corresponding RNA level based on DNA Reads.

MutRatio: The mean value of the cancer cell fraction probability.

MutRatio_Min: The 1% percentile of the cancer cell fraction probability.

MutRatio_Max: The 99% percentile of the cancer cell fraction probability.

P_I: Priority score using the IC50. Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

P_R: Priority score using the percentage of rank affinity. Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

P: Priority score implemented in MuPeXI (Bjerregaard et al. 2017). Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

`MainINDELClass2` *Calculate Neoantigen Candidates on INDELs for MHC Class2*

Description

Calculate Neoantigen Candidates on INDELs for MHC Class2

Usage

```
MainINDELClass2(
  input_annovar_format_file = NA,
  input_vep_format_file = NA,
  input_vcf_format_file_and_vep = NA,
  hla_file = "here_is_a_table",
  hla_types = NA,
  file_name_in_hla_table = "sample",
```

```
  refflat_file = paste(hmdir, "lib/refFlat.txt", sep = "/"),
  refmrna_file = paste(hmdir, "lib/refMrna.fa", sep = "/"),
  hmdir = getwd(),
  job_id = "ID",
  export_dir = paste("result", job_id, "INDEL2", sep = "."),
  rnaexp_file = NA,
  rnabam_file = NA,
  cnv_file = NA,
  purity = 1,
  netMHCIIpan_dir = paste(hmdir, "lib/netMHCIIpan-3.1/netMHCIIpan", sep = "/"),
  refdna_file = NA,
  samtools_dir = "samtools",
  bcftools_dir = NA,
  chr_column = NA,
  mutation_start_column = NA,
  mutation_end_column = NA,
  mutation_ref_column = NA,
  mutation_alt_column = NA,
  nm_id_column = NA,
  depth_normal_column = NA,
  depth_tumor_column = NA,
  ambiguous_between_exon = 0,
  ambiguous_codon = 0,
  peptide_length = c(15),
  ignore_short = TRUE,
  SNPs = NA,
  multiple_variants = FALSE
)
```

Arguments

`input_annovar_format_file`
: An input vcf file annotated by ANNOVAR (http://annovar.openbioinformatics.org/en/latest/). You can directly indicate a matrix, which is the same as annovar format vcf file, as input.

  See by data(sample_vcf.annovar); sample_vcf.annovar.txt;

`input_vep_format_file`
: An input file annotated by Ensembl Variant Effect Predictor (VEP). You can directly indicate a matrix, which is the same as annovar format VEP file, as input.

  See by data(sample_vcf.vep); sample_vcf.vep.txt;

`input_vcf_format_file_and_vep`
: An input vcf file and path to Ensembl Variant Effect Predictor (VEP). Before using this option, please install vep according to the official cite ("https://asia.ensembl.org/info/docs/tool

`hla_file`
: A tab separated file indicating HLA types. The 1st column is input_file name, and the following columns indicate HLA types.

  See by data(sample_hla_table_c1); sample_hla_table_c1;

`hla_types`
: Set a list of HLA types

`file_name_in_hla_table`
: If the name (1st column) in HLA table is not the same as input_file, indicate the corresponding name.

refflat_file
: refFlat file to be used in constructing peptide. (Default=paste(hmdir, "lib/refFlat.txt", sep="").
See "https://github.com/hase62/Neoantimon"

refmrna_file
: refMrna file to be used in constructing peptide (Default=paste(hmdir, "lib/refMrna.fa", sep="").
See "https://github.com/hase62/Neoantimon"

hmdir
: Home directory for the analysis (Default = getwd()).

job_id
: Job-Id to be attached in output files (Default = "NO_job_id").

export_dir
: The directory will be stored results (Default = "paste("result", file_name_in_hla_table, job_id, sep=".")")

rnaexp_file
: A file including RNA expressions (Default=NA). The 1st, 2nd and 3rd columns are "GeneSymbol Chr:Exonstart-Exonend (locus) ExpressionAmount", respectively. The 1st row should be any header.
See by data(sample_rna_exp); sample_rna_exp;

rnabam_file
: RNA bam file to calculate variant allele frequency of RNA at each mutation (Default=NA).

cnv_file
: A file including copy number variation to calculate cancer cell fraction probability (CCFP) (Default=NA). The format is according to ASCAT output files. The columns are "SNPName Chromosome Position LogR segmentedLogR BAF segmentedBAF CopyNumber MinorAllele RawCopyNumber" The 1st row should be the above header.
See data(sample_copynum); sample_copynum;

purity
: Tumor purity or tumor contents ratio required to calculate CCFP (Default=1).

netMHCIIpan_dir
: The file directory to netMHCpan (Default="lib/netMHCIIpan-3.2/netMHCpan").

refdna_file
: refdna_file information to be used to calculate RNA VAF (Default=NA).
See "https://github.com/hase62/Neoantimon"

samtools_dir
: The file directory to samtools_0_x_x (Default="samtools"). It shouled be indicated when you indicate RNA-bam and try to calculate RNA VAF.

bcftools_dir
: The file directory to netMHCpan (Default="bcftools"). It shouled be indicated when you indicate RNA-bam and try to calculate RNA VAF. samtools 0_x_x includes bcftools in the directory.

chr_column
: The column number describing chromosome number in input_file (Default=NA, but will automatically search "Chr" in header).

mutation_start_column
: The column number describing mutation start Position in input_file (Default=NA, but will automatically search "Start" in header) .

mutation_end_column
: The column number describing mutation end Position in input_file (Default=NA, but will automatically search "End" in header).

mutation_ref_column
: The column number describing mutation Ref in input_file (Default=NA, but will automatically search "Ref" in header).

mutation_alt_column
: The column number describing mutation Alt in input_file (Default=NA, but will automatically search "Alt" in header).

`nm_id_column` The column number describing NM IDs in input_file such as "SLCO1C1:NM_001145944:exon7:c.692_693insG:p.L231fs" (Default=NA).

`depth_normal_column` The column number describing the read count from normal cells (Default = NA).

`depth_tumor_column` The column number describing the read count from tumor cells (Default = NA).

`ambiguous_between_exon` The maximum number to permit the differences between Exon-Lengths from refFlat and refMrna (Default=0).

`ambiguous_codon` The maximum number to permit the differences between inputfile- and refMrna-oriented translation start/end position (Default=0).

`peptide_length` Peptide Length to be generated (Default = 15 in HLA Class2).

`ignore_short` Ignore to output results of short peptide less than min (peptide_length)

`SNPs` Apply indivisual SNPs on peptides by indicate a vcf file.

`multiple_variants` Reflect multiple variants on a peptide, e.g., SNVs on frameshift region.

Value

void (Calculated Neoantigen Files will be generated as .tsv files.):

HLA: HLA type used to calculate neoantigen.

Pos: The position of a fraction of peptide used to be evaluated from the full-length peptide.

Gene Gene symbol used to be evaluated in NetMHCpan.

Evaluated_Mutant_Peptide: The mutant peptide to be evaluated.

Evaluated_Mutant_Peptide_Core: The core peptide of the mutant peptide to be evaluated in NetMHCpan.

Mut_IC50: IC50 value for evaluated mutant peptide.

Mut_Rank: Rank value for evaluated mutanat peptide.

Chr: Chromosome Number of the mutation.

NM_ID: NM_ID used to construct peptides from the mutation.

Change: The annotation to be described in .vcf file.

Ref: reference type nucleic acid base.

Alt: alternative type nucleic acid base.

Prob: A probability of reference nucleic acid base described in .vcf file.

Mutation_Prob: A probability of alternative nucleic acid base described in .vcf file.

Exon_Start: The exon start position of the corrsponding NM_ID.

Exon_End: The exon end position of the corrsponding NM_ID.

Mutation_Position: The mutation position of the corrsponding NM_ID.

Total_Depth: The sum depth of the reference and alternative nucleic acid base.

Tumor_Depth: The depth of the alternative nucleic acid base.

Wt_Peptide: The full-length of the wild-type peptide.

Mutant_Peptide: The full-length of the mutant peptide.

Total_RNA: The expression amount of the corresponding RNA.

Tumor_RNA_Ratio: The variant allele frequency of the corresponding RNA.

Tumor_RNA: The modified amount of the corresponding RNA level based on RNA Reads.

Tumor_RNA_based_on_DNA: The modified amount of the corresponding RNA level based on DNA Reads.

MutRatio: The mean value of the cancer cell fraction probability.

MutRatio_Min: The 1% percentile of the cancer cell fraction probability.

MutRatio_Max: The 99% percentile of the cancer cell fraction probability.

P_I: Priority score using the IC50. Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

P_R: Priority score using the percentage of rank affinity. Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

P: Priority score implemented in MuPeXI (Bjerregaard et al. 2017). Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

`MainSeqFragmentClass1` *Calculate Neoantigen Candidates from A Given Sequence for MHC Class1*

Description

Calculate Neoantigen Candidates from A Given Sequence for MHC Class1

Usage

```
MainSeqFragmentClass1(
  input_sequence = NA,
  group_ids = seq(1:length(reference_nm_id)),
  hla_file = "here_is_a_table",
  hla_types = NA,
  file_name_in_hla_table = NA,
  refflat_file = paste(hmdir, "lib/refFlat.txt", sep = "/"),
  refmrna_file = paste(hmdir, "lib/refMrna.fa", sep = "/"),
  hmdir = getwd(),
  job_id = "ID",
  export_dir = paste("result", job_id, "SeqFragment1", sep = "."),
  netMHCpan_dir = paste(hmdir, "lib/netMHCpan-4.0/netMHCpan", sep = "/"),
  peptide_length = c(8, 9, 10, 11, 12, 13),
  reference_nm_id = NA,
  reference_gene_symbol = NA,
  ignore_short = TRUE
)
```

Arguments

`input_sequence`	(Required) An input amino acid sequence
`group_ids`	flag to cluster the same group
`hla_file`	A tab separated file indicating HLA types. The 1st column is input_file name, and the following columns indicate HLA types. See by data(sample_hla_table_c1); sample_hla_table_c1;

`hla_types` Set a list of HLA types

`file_name_in_hla_table`
If the name (1st column) in HLA table is not the same as input_file, indicate the corresponding name (Default=input_file).

`refflat_file` refFlat file to be used in constructing peptide. (Default=paste(hmdir, "lib/refFlat.txt", sep="").
See "https://github.com/hase62/Neoantimon"

`refmrna_file` refMrna file to be used in constructing peptide (Default=paste(hmdir, "lib/refMrna.fa", sep="").
See "https://github.com/hase62/Neoantimon"

`hmdir` Home directory for the analysis (Default = getwd()).

`job_id` Job-Id to be attached in output files (Default = "NO_job_id").

`export_dir` The directory will be stored results (Default = "paste("result", file_name_in_hla_table, job_id, sep=".")")

`netMHCpan_dir` The file directory to netMHCpan (Default="lib/netMHCpan-4.0/netMHCpan").

`peptide_length` Peptide Length to be generated (Default = 8,9,10,11,12,13).

`reference_nm_id`
Corresponding original sequences that the input sequence is generated. If franctions of peptides generated from the input are included in the indicated protein, such peptides are removed. It can be indicated when gene_symbol is not NA.

`reference_gene_symbol`
Corresponding original sequences that the input sequence is generated. If franctions of peptides generated from the input are included in the indicated protein, such peptides are removed. It can be indicated when nm_id is not NA.

`ignore_short` Ignore to output results of short peptide less than min (peptide_length)

Value

void (Calculated Neoantigen Files will be generated as .tsv files.):

HLA: HLA type used to calculate neoantigen.

Pos: The position of a fraction of peptide used to be evaluated from the full-length peptide.

Gene Gene symbol used to be evaluated in NetMHCpan.

Evaluated_Mutant_Peptide: The mutant peptide to be evaluated.

Evaluated_Mutant_Peptide_Core: The core peptide of the mutant peptide to be evaluated in NetMHCpan.

Mut_IC50: IC50 value for evaluated mutant peptide.

Mut_Rank: Rank value for evaluated mutanat peptide.

Chr: Chromosome Number of the mutation.

NM_ID: NM_ID used to construct peptides from the mutation.

Change: The annotation to be described in .vcf file.

Ref: reference type nucleic acid base.

Alt: alternative type nucleic acid base.

Prob: A probability of reference nucleic acid base described in .vcf file.

Mutation_Prob: A probability of alternative nucleic acid base described in .vcf file.

Exon_Start: The exon start position of the corrsponding NM_ID.

Exon_End: The exon end position of the corrsponding NM_ID.

Mutation_Position: The mutation position of the corrsponding NM_ID.

Total_Depth: The sum depth of the reference and alternative nucleic acid base.

Tumor_Depth: The depth of the alternative nucleic acid base.

Wt_Peptide: The full-length of the wild-type peptide.

Mutant_Peptide: The full-length of the mutant peptide.

Total_RNA: The expression amount of the corresponding RNA.

Tumor_RNA_Ratio: The variant allele frequency of the corresponding RNA.

Tumor_RNA: The modified amount of the corresponding RNA level based on RNA Reads.

Tumor_RNA_based_on_DNA: The modified amount of the corresponding RNA level based on DNA Reads.

MutRatio: The mean value of the cancer cell fraction probability.

MutRatio_Min: The 1% percentile of the cancer cell fraction probability.

MutRatio_Max: The 99% percentile of the cancer cell fraction probability.

P_I: Priority score using the IC50. Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

P_R: Priority score using the percentage of rank affinity. Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

P: Priority score implemented in MuPeXI (Bjerregaard et al. 2017). Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

`MainSeqFragmentClass2` *Calculate Neoantigen Candidates from A Given Sequence for MHC Class2*

Description

Calculate Neoantigen Candidates from A Given Sequence for MHC Class2

Usage

```
MainSeqFragmentClass2(
  input_sequence = NA,
  group_ids = seq(1:length(reference_nm_id)),
  hla_file = "here_is_a_table",
  hla_types = NA,
  file_name_in_hla_table = NA,
  refflat_file = paste(hmdir, "lib/refFlat.txt", sep = "/"),
  refmrna_file = paste(hmdir, "lib/refMrna.fa", sep = "/"),
  hmdir = getwd(),
  job_id = "ID",
  export_dir = paste("result", job_id, "SeqFragment2", sep = "."),
  netMHCIIpan_dir = paste(hmdir, "lib/netMHCIIpan-3.1/netMHCIIpan", sep = "/"),
  peptide_length = c(15),
  reference_nm_id = NA,
```

```
  reference_gene_symbol = NA,
  ignore_short = TRUE
)
```

Arguments

input_sequence	(Required) An input amino acid sequence
group_ids	flag to cluster the same group
hla_file	A tab separated file indicating HLA types. The 1st column is input_file name, and the following columns indicate HLA types. See by data(sample_hla_table_c1); sample_hla_table_c1;
hla_types	Set a list of HLA types
file_name_in_hla_table	If the name (1st column) in HLA table is not the same as input_file, indicate the corresponding name (Default=input_file).
refflat_file	refFlat file to be used in constructing peptide. (Default=paste(hmdir, "lib/refFlat.txt", sep=""). See "https://github.com/hase62/Neoantimon"
refmrna_file	refMrna file to be used in constructing peptide (Default=paste(hmdir, "lib/refMrna.fa", sep=""). See "https://github.com/hase62/Neoantimon"
hmdir	Home directory for the analysis (Default = getwd()).
job_id	Job-Id to be attached in output files (Default = "NO_job_id").
export_dir	The directory will be stored results (Default = "paste("result", file_name_in_hla_table, job_id, sep=".")")
netMHCIIpan_dir	The file directory to netMHCpan (Default="lib/netMHCIIpan-3.2/netMHCIIpan").
peptide_length	Peptide Length to be generated (Default = 8,9,10,11,12,13).
reference_nm_id	Corresponding original sequences that the input sequence is generated. If franctions of peptides generated from the input are included in the indicated protein, such peptides are removed. It can be indicated when gene_symbol is not NA.
reference_gene_symbol	Corresponding original sequences that the input sequence is generated. If franctions of peptides generated from the input are included in the indicated protein, such peptides are removed. It can be indicated when nm_id is not NA.
ignore_short	Ignore to output results of short peptide less than min (peptide_length)

Value

void (Calculated Neoantigen Files will be generated as .tsv files.):

HLA: HLA type used to calculate neoantigen.

Pos: The position of a fraction of peptide used to be evaluated from the full-length peptide.

Gene Gene symbol used to be evaluated in NetMHCpan.

Evaluated_Mutant_Peptide: The mutant peptide to be evaluated.

Evaluated_Mutant_Peptide_Core: The core peptide of the mutant peptide to be evaluated in NetMHCpan.

Mut_IC50: IC50 value for evaluated mutant peptide.

Mut_Rank: Rank value for evaluated mutanat peptide.

Chr: Chromosome Number of the mutation.

NM_ID: NM_ID used to construct peptides from the mutation.

Change: The annotation to be described in .vcf file.

Ref: reference type nucleic acid base.

Alt: alternative type nucleic acid base.

Prob: A probability of reference nucleic acid base described in .vcf file.

Mutation_Prob: A probability of alternative nucleic acid base described in .vcf file.

Exon_Start: The exon start position of the corrsponding NM_ID.

Exon_End: The exon end position of the corrsponding NM_ID.

Mutation_Position: The mutation position of the corrsponding NM_ID.

Total_Depth: The sum depth of the reference and alternative nucleic acid base.

Tumor_Depth: The depth of the alternative nucleic acid base.

Wt_Peptide: The full-length of the wild-type peptide.

Mutant_Peptide: The full-length of the mutant peptide.

Total_RNA: The expression amount of the corresponding RNA.

Tumor_RNA_Ratio: The variant allele frequency of the corresponding RNA.

Tumor_RNA: The modified amount of the corresponding RNA level based on RNA Reads.

Tumor_RNA_based_on_DNA: The modified amount of the corresponding RNA level based on DNA Reads.

MutRatio: The mean value of the cancer cell fraction probability.

MutRatio_Min: The 1% percentile of the cancer cell fraction probability.

MutRatio_Max: The 99% percentile of the cancer cell fraction probability.

P_I: Priority score using the IC50. Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

P_R: Priority score using the percentage of rank affinity. Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

P: Priority score implemented in MuPeXI (Bjerregaard et al. 2017). Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

`MainSNVClass1` *Calculate Neoantigen Candidates on SNVs for MHC Class1*

Description

Calculate Neoantigen Candidates on SNVs for MHC Class1

Usage

```
MainSNVClass1(
  input_annovar_format_file = NA,
  input_vep_format_file = NA,
  input_vcf_format_file_and_vep = NA,
  hla_file = "here_is_a_table",
  hla_types = NA,
  file_name_in_hla_table = "sample",
  refflat_file = paste(hmdir, "lib/refFlat.txt", sep = "/"),
  refmrna_file = paste(hmdir, "lib/refMrna.fa", sep = "/"),
  hmdir = getwd(),
  job_id = "ID",
  export_dir = paste("result", job_id, "SNV1", sep = "."),
  rnaexp_file = NA,
  rnabam_file = NA,
  cnv_file = NA,
  purity = 1,
  netMHCpan_dir = paste(hmdir, "lib/netMHCpan-4.0/netMHCpan", sep = "/"),
  MHCflurry = NA,
  refdna_file = NA,
  samtools_dir = "samtools",
  bcftools_dir = NA,
  chr_column = NA,
  mutation_start_column = NA,
  mutation_end_column = NA,
  mutation_ref_column = NA,
  mutation_alt_column = NA,
  nm_id_column = NA,
  depth_normal_column = NA,
  depth_tumor_column = NA,
  ambiguous_between_exon = 0,
  ambiguous_codon = 0,
  peptide_length = c(8, 9, 10, 11, 12, 13),
  ignore_short = TRUE,
  SNPs = NA,
  multiple_variants = FALSE
)
```

Arguments

`input_annovar_format_file`
: An input vcf file annotated by ANNOVAR (http://annovar.openbioinformatics.org/en/latest/). You can directly indicate a matrix, which is the same as annovar format vcf file, as input.

  See by data(sample_vcf.annovar); sample_vcf.annovar.txt;

`input_vep_format_file`
: An input file annotated by Ensembl Variant Effect Predictor (VEP). You can directly indicate a matrix, which is the same as annovar format VEP file, as input.

  See by data(sample_vcf.vep); sample_vcf.vep.txt;

`input_vcf_format_file_and_vep`
: A list of (1) An input vcf file, (2) path to Ensembl Variant Effect Predictor

(VEP), and (3) cache file for VEP. Before using this option, please install vep according to the official cite ("https://asia.ensembl.org/info/docs/tools/vep/index.html").

`hla_file` A tab separated file indicating HLA types. The 1st column is input_file name, and the following columns indicate HLA types.
See by data(sample_hla_table_c1); sample_hla_table_c1;

`hla_types` Set a list of HLA types

`file_name_in_hla_table`
If the name (1st column) in HLA table is not the same as input_file, indicate the corresponding name.

`refflat_file` refFlat file to be used in constructing peptide. (Default=paste(hmdir, "lib/refFlat.txt", sep="").
See "https://github.com/hase62/Neoantimon"

`refmrna_file` refMrna file to be used in constructing peptide (Default=paste(hmdir, "lib/refMrna.fa", sep="").
See "https://github.com/hase62/Neoantimon"

`hmdir` Home directory for the analysis (Default = getwd()).

`job_id` Job-id to be attached in output files (Default = "NO_job_id").

`export_dir` The directory will be stored results (Default = "paste("result", file_name_in_hla_table, job_id, sep=".")")

`rnaexp_file` A file including RNA expressions (Default=NA). The 1st, 2nd and 3rd columns are "GeneSymbol Chr:Exonstart-Exonend (locus) ExpressionAmount", respectively. The 1st row should be any header.
See by data(sample_rna_exp); sample_rna_exp;

`rnabam_file` RNA bam file to calculate variant allele frequency of RNA at each mutation (Default=NA).

`cnv_file` A file including copy number variation to calculate cancer cell fraction probability (CCFP) (Default=NA). The format is according to ASCAT output files. The columns are "SNPName Chromosome Position LogR segmentedLogR BAF segmentedBAF CopyNumber MinorAllele RawCopyNumber" The 1st row should be the above header.
See data(sample_copynum); sample_copynum;

`purity` Tumor purity or tumor contents ratio required to calculate CCFP (Default=1).

`netMHCpan_dir` The file directory to netMHCpan (Default="lib/netMHCpan-4.0/netMHCpan").

`MHCflurry` Output MHCflurry results. Return a list of both results (Default=FALSE).

`refdna_file` refdna_file information to be used to calculate RNA VAF (Default=NA).
See "https://github.com/hase62/Neoantimon"

`samtools_dir` The file directory to samtools_0_x_x (Default="samtools"). It shouled be indicated when you indicate RNA-bam and try to calculate RNA VAF.

`bcftools_dir` The file directory to netMHCpan (Default="bcftools"). It shouled be indicated when you indicate RNA-bam and try to calculate RNA VAF . samtools 0_x_x includes bcftools in the directory.

`chr_column` The column number describing chromosome number in input_file (Default=NA, but will automatically search "Chr" in header).

`mutation_start_column`
The column number describing mutation start Position in input_file (Default=NA, but will automatically search "Start" in header) .

`mutation_end_column`
: The column number describing mutation end Position in input_file (Default=NA, but will automatically search "End" in header).

`mutation_ref_column`
: The column number describing mutation Ref in input_file (Default=NA, but will automatically search "Ref" in header).

`mutation_alt_column`
: The column number describing mutation Alt in input_file (Default=NA, but will automatically search "Alt" in header).

`nm_id_column`
: The column number describing NM IDs in input_file such as "SLCO1C1:NM_001145944:exon7:c.692_693insG:p.L231fs" (Default=NA).

`depth_normal_column`
: The column number describing the read count from normal cells (Default = NA).

`depth_tumor_column`
: The column number describing the read count from tumor cells (Default = NA).

`ambiguous_between_exon`
: The maximum number to permit the differences between Exon-Lengths from refFlat and refMrna (Default=0).

`ambiguous_codon`
: The maximum number to permit the differences between inputfile- and refMrna-oriented translation start/end position (Default=0).

`peptide_length`
: Peptide Length to be generated (Default = 8,9,10,11,12,13).

`ignore_short`
: Ignore to output results of short peptide less than min (peptide_length)

`SNPs`
: Apply indivisual SNPs on peptides by indicate a vcf file.

`multiple_variants`
: Reflect multiple variants on a peptide, e.g., SNVs on frameshift region.

Value

void (Calculated Neoantigen Files will be generated as .tsv files.)

HLA: HLA type used to calculate neoantigen.

Pos: The position of a fraction of peptide used to be evaluated from the full-length peptide.

Gene Gene symbol used to be evaluated in NetMHCpan.

Evaluated_Mutant_Peptide: The mutant peptide to be evaluated.

Mut_IC50: IC50 value for evaluated mutant peptide.

Mut_Rank: Rank value for evaluated mutanat peptide.

Evaluated_Wt_Peptide: The wild-type peptide to be evaluated.

Wt_IC50: IC50 value for evaluated wild-type peptide.

Wt_Rank: Rank value for evaluated wild-type peptide.

Chr: Chromosome Number of the mutation.

NM_ID: NM_ID used to construct peptides from the mutation.

Change: The annotation to be described in .vcf file.

Ref: reference type nucleic acid base.

Alt: alternative type nucleic acid base.

Prob: A probability of reference nucleic acid base described in .vcf file.

Mutation_Prob: A probability of alternative nucleic acid base described in .vcf file.

Exon_Start: The exon start position of the corrsponding NM_ID.

Exon_End: The exon end position of the corrsponding NM_ID.

Mutation_Position: The mutation position of the corrsponding NM_ID.

Total_Depth: The sum depth of the reference and alternative nucleic acid base.

Tumor_Depth: The depth of the alternative nucleic acid base.

Wt_Peptide: The full-length of the wild-type peptide.

Mutant_Peptide: The full-length of the mutant peptide.

Total_RNA: The expression amount of the corresponding RNA.

Tumor_RNA_Ratio: The variant allele frequency of the corresponding RNA.

Tumor_RNA: The modified amount of the corresponding RNA level based on RNA Reads.

Tumor_RNA_based_on_DNA: The modified amount of the corresponding RNA level based on DNA Reads.

MutRatio: The mean value of the cancer cell fraction probability.

MutRatio_Min: The 1% percentile of the cancer cell fraction probability.

MutRatio_Max: The 99% percentile of the cancer cell fraction probability.

P_I: Priority score using the IC50. Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

P_R: Priority score using the percentage of rank affinity. Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

P: Priority score implemented in MuPeXI (Bjerregaard et al. 2017). Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

`MainSNVClass2` *Calculate Neoantigen Candidates on SNVs for MHC Class2*

Description

Calculate Neoantigen Candidates on SNVs for MHC Class2

Usage

```
MainSNVClass2(
  input_annovar_format_file = NA,
  input_vep_format_file = NA,
  input_vcf_format_file_and_vep = NA,
  hla_file = "here_is_a_table",
  hla_types = NA,
  file_name_in_hla_table = "sample",
  refflat_file = paste(hmdir, "lib/refFlat.txt", sep = "/"),
  refmrna_file = paste(hmdir, "lib/refMrna.fa", sep = "/"),
  hmdir = getwd(),
  job_id = "ID",
  export_dir = paste("result", job_id, "SNV2", sep = "."),
  rnaexp_file = NA,
```

```
  rnabam_file = NA,
  cnv_file = NA,
  purity = 1,
  netMHCIIpan_dir = paste(hmdir, "lib/netMHCIIpan-3.2/netMHCIIpan", sep = "/"),
  refdna_file = NA,
  samtools_dir = "samtools",
  bcftools_dir = NA,
  chr_column = NA,
  mutation_start_column = NA,
  mutation_end_column = NA,
  mutation_ref_column = NA,
  mutation_alt_column = NA,
  nm_id_column = NA,
  depth_normal_column = NA,
  depth_tumor_column = NA,
  ambiguous_between_exon = 0,
  ambiguous_codon = 0,
  peptide_length = c(15),
  ignore_short = TRUE,
  SNPs = NA,
  multiple_variants = FALSE
)
```

Arguments

`input_annovar_format_file`
: An input vcf file annotated by ANNOVAR (http://annovar.openbioinformatics.org/en/latest/). You can directly indicate a matrix, which is the same as annovar format vcf file, as input. See by data(sample_vcf.annovar); sample_vcf.annovar.txt;

`input_vep_format_file`
: An input file annotated by Ensembl Variant Effect Predictor (VEP). You can directly indicate a matrix, which is the same as annovar format VEP file, as input.

  See by data(sample_vcf.vep); sample_vcf.vep.txt;

`input_vcf_format_file_and_vep`
: A list of (1) An input vcf file, (2) path to Ensembl Variant Effect Predictor (VEP), and (3) cache file for VEP. Before using this option, please install vep according to the official cite ("https://asia.ensembl.org/info/docs/tools/vep/index.html").

`hla_file`
: A tab separated file indicating HLA types. The 1st column is input_file name, and the following columns indicate HLA types.

  See by data(sample_hla_table_c1); sample_hla_table_c1;

`hla_types`
: Set a list of HLA types

`file_name_in_hla_table`
: If the name (1st column) in HLA table is not the same as input_file, indicate the corresponding name.

`refflat_file`
: refFlat file to be used in constructing peptide. (Default=paste(hmdir, "lib/refFlat.txt", sep="").

  See "https://github.com/hase62/Neoantimon"

`refmrna_file`
: refMrna file to be used in constructing peptide (Default=paste(hmdir, "lib/refMrna.fa", sep="").

  See "https://github.com/hase62/Neoantimon"

`hmdir` Home directory for the analysis (Default = getwd()).

`job_id` Job-Id to be attached in output files (Default = "NO_job_id").

`export_dir` The directory will be stored results (Default = "paste("result", file_name_in_hla_table, job_id, sep=".")")

`rnaexp_file` A file including RNA expressions (Default=NA). The 1st, 2nd and 3rd columns are "GeneSymbol Chr:Exonstart-Exonend (locus) ExpressionAmount", respectively. The 1st row should be any header.

See by data(sample_rna_exp); sample_rna_exp;

`rnabam_file` RNA bam file to calculate variant allele frequency of RNA at each mutation (Default=NA).

`cnv_file` A file including copy number variation to calculate cancer cell fraction probability (CCFP) (Default=NA). The format is according to ASCAT output files. The columns are "SNPName Chromosome Position LogR segmentedLogR BAF segmentedBAF CopyNumber MinorAllele RawCopyNumber" The 1st row should be the above header.

See data(sample_copynum); sample_copynum;

`purity` Tumor purity or tumor contents ratio required to calculate CCFP (Default=1).

`netMHCIIpan_dir`
The file directory to netMHCpan (Default="lib/netMHCIIpan-3.2/netMHCpan").

`refdna_file` refdna_file information to be used to calculate RNA VAF (Default=NA).

See "https://github.com/hase62/Neoantimon"

`samtools_dir` The file directory to samtools_0_x_x (Default="samtools"). It shouled be indicated when you indicate RNA-bam and try to calculate RNA VAF .

`bcftools_dir` The file directory to netMHCpan (Default="bcftools"). It shouled be indicated when you indicate RNA-bam and try to calculate RNA VAF . samtools 0_x_x includes bcftools in the directory.

`chr_column` The column number describing chromosome number in input_file (Default=NA, but will automatically search "Chr" in header).

`mutation_start_column`
The column number describing mutation start Position in input_file (Default=NA, but will automatically search "Start" in header) .

`mutation_end_column`
The column number describing mutation end Position in input_file (Default=NA, but will automatically search "End" in header).

`mutation_ref_column`
The column number describing mutation Ref in input_file (Default=NA, but will automatically search "Ref" in header).

`mutation_alt_column`
The column number describing mutation Alt in input_file (Default=NA, but will automatically search "Alt" in header).

`nm_id_column` The column number describing NM IDs in input_file such as "SLCO1C1:NM_001145944:exon7:c.692_693insG:p.L231fs" (Default=NA).

`depth_normal_column`
The column number describing the read count from normal cells (Default = NA).

`depth_tumor_column`
The column number describing the read count from tumor cells (Default = NA).

`ambiguous_between_exon`
: The maximum number to permit the differences between Exon-Lengths from refFlat and refMrna (Default=0).

`ambiguous_codon`
: The maximum number to permit the differences between inputfile- and refMrna-oriented translation start/end position (Default=0).

`peptide_length`
: Peptide Length to be generated (Default = 15 in HLA Class2).

`ignore_short`
: Ignore to output results of short peptide less than min (peptide_length)

`SNPs`
: Apply indivisual SNPs on peptides by indicate a vcf file.

`multiple_variants`
: Reflect multiple variants on a peptide, e.g., SNVs on frameshift region.

Value

void (Calculated Neoantigen Files will be generated as .tsv files.):

HLA: HLA type used to calculate neoantigen.

Pos: The position of a fraction of peptide used to be evaluated from the full-length peptide.

Gene Gene symbol used to be evaluated in NetMHCpan.

Evaluated_Mutant_Peptide: The mutant peptide to be evaluated.

Mut_IC50: IC50 value for evaluated mutant peptide.

Mut_Rank: Rank value for evaluated mutanat peptide.

Evaluated_Wt_Peptide: The wild-type peptide to be evaluated.

Wt_IC50: IC50 value for evaluated wild-type peptide.

Wt_Rank: Rank value for evaluated wild-type peptide.

Chr: Chromosome Number of the mutation.

NM_ID: NM_ID used to construct peptides from the mutation.

Change: The annotation to be described in .vcf file.

Ref: reference type nucleic acid base.

Alt: alternative type nucleic acid base.

Prob: A probability of reference nucleic acid base described in .vcf file.

Mutation_Prob: A probability of alternative nucleic acid base described in .vcf file.

Exon_Start: The exon start position of the corrsponding NM_ID.

Exon_End: The exon end position of the corrsponding NM_ID.

Mutation_Position: The mutation position of the corrsponding NM_ID.

Total_Depth: The sum depth of the reference and alternative nucleic acid base.

Tumor_Depth: The depth of the alternative nucleic acid base.

Wt_Peptide: The full-length of the wild-type peptide.

Mutant_Peptide: The full-length of the mutant peptide.

Total_RNA: The expression amount of the corresponding RNA.

Tumor_RNA_Ratio: The variant allele frequency of the corresponding RNA.

Tumor_RNA: The modified amount of the corresponding RNA level based on RNA Reads.

Tumor_RNA_based_on_DNA: The modified amount of the corresponding RNA level based on DNA Reads.

MutRatio: The mean value of the cancer cell fraction probability.

MutRatio_Min: The 1% percentile of the cancer cell fraction probability.

MutRatio_Max: The 99% percentile of the cancer cell fraction probability.

P_I: Priority score using the IC50. Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

P_R: Priority score using the percentage of rank affinity. Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

P: Priority score implemented in MuPeXI (Bjerregaard et al. 2017). Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

`MainSVFUSIONClass1` *Calculate Neoantigen Candidates on SV fusions for MHC Class1*

Description

Calculate Neoantigen Candidates on SV fusions for MHC Class1

Usage

```
MainSVFUSIONClass1(
  input_file,
  hla_file = "here_is_a_table",
  hla_types = NA,
  file_name_in_hla_table = input_file,
  refflat_file = paste(hmdir, "lib/refFlat.txt", sep = "/"),
  refmrna_file = paste(hmdir, "lib/refMrna.fa", sep = "/"),
  hmdir = getwd(),
  job_id = "ID",
  export_dir = paste("result", job_id, "SV1", sep = "."),
  rnaexp_file = NA,
  rnabam_file = NA,
  cnv_file = NA,
  purity = 1,
  netMHCpan_dir = paste(hmdir, "lib/netMHCpan-4.0/netMHCpan", sep = "/"),
  refdna_file = NA,
  samtools_dir = NA,
  bcftools_dir = NA,
  chr_column = NA,
  mutation_start_column = NA,
  mutation_end_column = NA,
  mutation_ref_column = NA,
  mutation_alt_bnd_column = NA,
  depth_normal_column = NA,
  depth_tumor_column = NA,
  nm_id_column = NA,
  ambiguous_between_exon = 0,
  ambiguous_codon = 0,
  peptide_length = c(8, 9, 10, 11, 12, 13),
  gene_symbol_column = NA,
```

```
  mate_id_column = NA,
  ignore_short = TRUE
)
```

Arguments

input_file	(Required) An input vcf file (BND format) annotated by, e.g., ANNOVAR (http://annovar.openbioinformatics.org/en/latest/) or other softwares. See by data(sample_sv_bnd); sample_sv_bnd;
hla_file	A tab separated file indicating HLA types. The 1st column is input_file name, and the following columns indicate HLA types. See by data(sample_hla_table_c1); sample_hla_table_c1;
hla_types	Set a list of HLA types
file_name_in_hla_table	If the name (1st column) in HLA table is not the same as input_file, indicate the corresponding name (Default=input_file).
refflat_file	refFlat file to be used in constructing peptide. (Default=paste(hmdir, "lib/refFlat.txt",sep="").) See "https://github.com/hase62/Neoantimon"
refmrna_file	refMrna file to be used in constructing peptide (Default=paste(hmdir, "lib/refMrna.fa", sep="").) See "https://github.com/hase62/Neoantimon"
hmdir	Home directory for the analysis (Default = getwd()).
job_id	Job-Id to be attached in output files (Default = "NO_job_id").
export_dir	The directory will be stored results (Default = "paste("result", file_name_in_hla_table, job_id, sep=".")")
rnaexp_file	A file including RNA expressions (Default=NA). The 1st, 2nd and 3rd columns are "GeneSymbol Chr:Exonstart-Exonend (locus) ExpressionAmount", respectively. The 1st row should be any header. See by data(sample_rna_exp); sample_rna_exp;
rnabam_file	RNA bam file to calculate variant allele frequency of RNA at each mutation (Default=NA).
cnv_file	A file including copy number variation to calculate cancer cell fraction probability (CCFP) (Default=NA). The format is according to ASCAT output files. The columns are "SNPName Chromosome Position LogR segmentedLogR BAF segmentedBAF CopyNumber MinorAllele RawCopyNumber" The 1st row should be the above header. See data(sample_copynum); sample_copynum;
purity	Tumor purity or tumor contents ratio required to calculate CCFP (Default=1).
netMHCpan_dir	The file directory to netMHCpan (Default="lib/netMHCpan-4.0/netMHCpan").
refdna_file	refdna_file information to be used to calculate RNA VAF (Default=NA). See "https://github.com/hase62/Neoantimon"
samtools_dir	The file directory to samtools_0_x_x (Default="samtools"). It shouled be indicated when you indicate RNA-bam and try to calculate RNA VAF.
bcftools_dir	The file directory to netMHCpan (Default="bcftools"). It shouled be indicated when you indicate RNA-bam and try to calculate RNA VAF. samtools 0_x_x includes bcftools in the directory.

`chr_column` The column number describing chromosome number in input_file (Default=NA, but will automatically search "Chr" in header).

`mutation_start_column` The column number describing mutation start Position in input_file (Default=NA, but will automatically search "Start" in header) .

`mutation_end_column` The column number describing mutation end Position in input_file (Default=NA, but will automatically search "End" in header).

`mutation_ref_column` The column number describing mutation Ref in input_file (Default=NA, but will automatically search "Ref" in header).

`mutation_alt_bnd_column` The column number describing mutation Alt (BND format) in input_file (Default=NA, but will automatically search "Alt" in header).

`depth_normal_column` The column number describing the read count from normal cells (Default = NA).

`depth_tumor_column` The column number describing the read count from tumor cells (Default = NA).

`nm_id_column` (Required if gene_symbol_column = NA) The column number describing NM IDs in input_file such as "SLCO1C1:NM_001145944:exon7:c.692_693insG:p.L231fs" (Default=NA).

`ambiguous_between_exon` The maximum number to permit the differences between Exon-Lengths from refFlat and refMrna (Default=0).

`ambiguous_codon` The maximum number to permit the differences between inputfile- and refMrna-oriented translation start/end position (Default=0).

`peptide_length` Peptide Length to be generated (Default = 8,9,10,11,12,13).

`gene_symbol_column` (Required if nm_id_column = NA) The column number describing gene symbol in input_file (Default=NA).

`mate_id_column` (Required) The column indicating mateIDs or svIDs such as "SVMERGE1_1" (Default=NA).

`ignore_short` Ignore to output results of short peptide less than min (peptide_length)

Value

void (Calculated Neoantigen Files will be generated as .tsv files.):

HLA: HLA type used to calculate neoantigen.

Pos: The position of a fraction of peptide used to be evaluated from the full-length peptide.

Gene Gene symbol used to be evaluated in NetMHCpan.

Evaluated_Mutant_Peptide: The mutant peptide to be evaluated.

Evaluated_Mutant_Peptide_Core: The core peptide of the mutant peptide to be evaluated in NetMHCpan.

Mut_IC50: IC50 value for evaluated mutant peptide.

Mut_Rank: Rank value for evaluated mutanat peptide.

Chr: Chromosome Number of the mutation.

NM_ID: NM_ID used to construct peptides from the mutation.

Change: The annotation to be described in .vcf file.

Ref: reference type nucleic acid base.

Alt: alternative type nucleic acid base.

Prob: A probability of reference nucleic acid base described in .vcf file.

Mutation_Prob: A probability of alternative nucleic acid base described in .vcf file.

Exon_Start: The exon start position of the corrsponding NM_ID.

Exon_End: The exon end position of the corrsponding NM_ID.

Mutation_Position: The mutation position of the corrsponding NM_ID.

Total_Depth: The sum depth of the reference and alternative nucleic acid base.

Tumor_Depth: The depth of the alternative nucleic acid base.

Wt_Peptide: The full-length of the wild-type peptide.

Mutant_Peptide: The full-length of the mutant peptide.

Total_RNA: The expression amount of the corresponding RNA.

Tumor_RNA_Ratio: The variant allele frequency of the corresponding RNA.

Tumor_RNA: The modified amount of the corresponding RNA level based on RNA Reads.

Tumor_RNA_based_on_DNA: The modified amount of the corresponding RNA level based on DNA Reads.

MutRatio: The mean value of the cancer cell fraction probability.

MutRatio_Min: The 1% percentile of the cancer cell fraction probability.

MutRatio_Max: The 99% percentile of the cancer cell fraction probability.

P_I: Priority score using the IC50. Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

P_R: Priority score using the percentage of rank affinity. Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

P: Priority score implemented in MuPeXI (Bjerregaard et al. 2017). Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

`MainSVFUSIONClass2` *Calculate Neoantigen Candidates on SV fusions for MHC Class2*

Description

Calculate Neoantigen Candidates on SV fusions for MHC Class2

Usage

```
MainSVFUSIONClass2(
  input_file,
  hla_file = "here_is_a_table",
  hla_types = NA,
  file_name_in_hla_table = input_file,
  refflat_file = paste(hmdir, "lib/refFlat.txt", sep = "/"),
  refmrna_file = paste(hmdir, "lib/refMrna.fa", sep = "/"),
```

```
  hmdir = getwd(),
  job_id = "ID",
  export_dir = paste("result", job_id, "SV2", sep = "."),
  rnaexp_file = NA,
  rnabam_file = NA,
  cnv_file = NA,
  purity = 1,
  netMHCIIpan_dir = paste(hmdir, "lib/netMHCIIpan-3.1/netMHCIIpan", sep = "/"),
  refdna_file = NA,
  samtools_dir = NA,
  bcftools_dir = NA,
  chr_column = NA,
  mutation_start_column = NA,
  mutation_end_column = NA,
  mutation_ref_column = NA,
  mutation_alt_bnd_column = NA,
  depth_normal_column = NA,
  depth_tumor_column = NA,
  nm_id_column = NA,
  ambiguous_between_exon = 0,
  ambiguous_codon = 0,
  peptide_length = c(15),
  gene_symbol_column = NA,
  mate_id_column = NA,
  ignore_short = TRUE
)
```

Arguments

input_file	(Required) An input vcf file (BND format) annotated by, e.g., ANNOVAR (http://annovar.openbioinformatics.org/en/latest/) or other softwares. See by data(sample_sv_bnd); sample_sv_bnd;
hla_file	A tab separated file indicating HLA types. The 1st column is input_file name, and the following columns indicate HLA types. See by data(sample_hla_table_c1); sample_hla_table_c1;
hla_types	Set a list of HLA types
file_name_in_hla_table	If the name (1st column) in HLA table is not the same as input_file, indicate the corresponding name (Default=input_file).
refflat_file	refFlat file to be used in constructing peptide. (Default=paste(hmdir, "lib/refFlat.txt",sep="". See "https://github.com/hase62/Neoantimon"
refmrna_file	refMrna file to be used in constructing peptide (Default=paste(hmdir, "lib/refMrna.fa", sep="". See "https://github.com/hase62/Neoantimon"
hmdir	Home directory for the analysis (Default = getwd()).
job_id	Job-Id to be attached in output files (Default = "NO_job_id").
export_dir	The directory will be stored results (Default = "paste("result", file_name_in_hla_table, job_id, sep=".")")

`rnaexp_file` A file including RNA expressions (Default=NA). The 1st, 2nd and 3rd columns are "GeneSymbol Chr:Exonstart-Exonend (locus) ExpressionAmount", respectively. The 1st row should be any header.

See by data(sample_rna_exp); sample_rna_exp;

`rnabam_file` RNA bam file to calculate variant allele frequency of RNA at each mutation (Default=NA).

`cnv_file` A file including copy number variation to calculate cancer cell fraction probability (CCFP) (Default=NA). The format is according to ASCAT output files. The columns are "SNPName Chromosome Position LogR segmentedLogR BAF segmentedBAF CopyNumber MinorAllele RawCopyNumber" The 1st row should be the above header.

See data(sample_copynum); sample_copynum;

`purity` Tumor purity or tumor contents ratio required to calculate CCFP (Default=1).

`netMHCIIpan_dir`

The file directory to netMHCpan (Default="lib/netMHCIIpan-3.2/netMHCpan").

`refdna_file` refdna_file information to be used to calculate RNA VAF (Default=NA).

See "https://github.com/hase62/Neoantimon"

`samtools_dir` The file directory to samtools_0_x_x (Default="samtools"). It shouled be indicated when you indicate RNA-bam and try to calculate RNA VAF .

`bcftools_dir` The file directory to netMHCpan (Default="bcftools"). It shouled be indicated when you indicate RNA-bam and try to calculate RNA VAF . samtools 0_x_x includes bcftools in the directory.

`chr_column` The column number describing chromosome number in input_file (Default=NA, but will automatically search "Chr" in header).

`mutation_start_column`

The column number describing mutation start Position in input_file (Default=NA, but will automatically search "Start" in header) .

`mutation_end_column`

The column number describing mutation end Position in input_file (Default=NA, but will automatically search "End" in header).

`mutation_ref_column`

The column number describing mutation Ref in input_file (Default=NA, but will automatically search "Ref" in header).

`mutation_alt_bnd_column`

The column number describing mutation Alt (BND format) in input_file (Default=NA, but will automatically search "Alt" in header).

`depth_normal_column`

The column number describing the read count from normal cells (Default = NA).

`depth_tumor_column`

The column number describing the read count from tumor cells (Default = NA).

`nm_id_column` (Required if gene_symbol_column = NA) The column number describing NM IDs in input_file such as

"SLCO1C1:NM_001145944:exon7:c.692_693insG:p.L231fs" (Default=NA).

`ambiguous_between_exon`

The maximum number to permit the differences between Exon-Lengths from refFlat and refMrna (Default=0).

`ambiguous_codon`

The maximum number to permit the differences between inputfile- and refMrna-oriented translation start/end position (Default=0).

`peptide_length` Peptide Length to be generated (Default = 15 in HLA Class2).

`gene_symbol_column`
(Required if nm_id_column = NA) The column number describing gene symbol in input_file (Default=NA).

`mate_id_column` (Required) The column indicating mateIDs or svIDs such as "SVMERGE1_1" (Default=NA).

`ignore_short` Ignore to output results of short peptide less than min (peptide_length)

Value

void (Calculated Neoantigen Files will be generated as .tsv files.):

HLA: HLA type used to calculate neoantigen.

Pos: The position of a fraction of peptide used to be evaluated from the full-length peptide.

Gene Gene symbol used to be evaluated in NetMHCpan.

Evaluated_Mutant_Peptide: The mutant peptide to be evaluated.

Evaluated_Mutant_Peptide_Core: The core peptide of the mutant peptide to be evaluated in NetMHCpan.

Mut_IC50: IC50 value for evaluated mutant peptide.

Mut_Rank: Rank value for evaluated mutanat peptide.

Chr: Chromosome Number of the mutation.

NM_ID: NM_ID used to construct peptides from the mutation.

Change: The annotation to be described in .vcf file.

Ref: reference type nucleic acid base.

Alt: alternative type nucleic acid base.

Prob: A probability of reference nucleic acid base described in .vcf file.

Mutation_Prob: A probability of alternative nucleic acid base described in .vcf file.

Exon_Start: The exon start position of the corrsponding NM_ID.

Exon_End: The exon end position of the corrsponding NM_ID.

Mutation_Position: The mutation position of the corrsponding NM_ID.

Total_Depth: The sum depth of the reference and alternative nucleic acid base.

Tumor_Depth: The depth of the alternative nucleic acid base.

Wt_Peptide: The full-length of the wild-type peptide.

Mutant_Peptide: The full-length of the mutant peptide.

Total_RNA: The expression amount of the corresponding RNA.

Tumor_RNA_Ratio: The variant allele frequency of the corresponding RNA.

Tumor_RNA: The modified amount of the corresponding RNA level based on RNA Reads.

Tumor_RNA_based_on_DNA: The modified amount of the corresponding RNA level based on DNA Reads.

MutRatio: The mean value of the cancer cell fraction probability.

MutRatio_Min: The 1% percentile of the cancer cell fraction probability.

MutRatio_Max: The 99% percentile of the cancer cell fraction probability.

P_I: Priority score using the IC50. Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

P_R: Priority score using the percentage of rank affinity. Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

P: Priority score implemented in MuPeXI (Bjerregaard et al. 2017). Please use CalculatePriorityScores <- function(result, useRNAvaf = FALSE)

`sample_copynum` *A Format / Sample file for Copy Number Information*

Description

A dataset containing the copy number information obtained by, e.g., ASCAT.

Usage

```
data(sample_copynum)
```

Format

A data frame with 7 rows and 9 variables

`sample_hla_table_c1` *A Format / Sample file for HLA CLASS1 Table*

Description

A dataset containing the HLA types of patients in each row.

Usage

```
data(sample_hla_table_c1)
```

Format

A data frame with 3 rows and at most 7 variables

`sample_hla_table_c2` *A Format / Sample file for HLA CLASS2 Table*

Description

A dataset containing the HLA types of patients in each row.

Usage

```
data(sample_hla_table_c2)
```

Format

A data frame with at least 3 row and at most 10 variables

sample_refFlat.grch37 *A Sample file for refFlat*

Description

A dataset containing a part of refFlat data.

Usage

```
data(sample_refFlat.grch37)
```

Format

A data frame with 11 column.

sample_refMrna.grch37.fa *A Sample file for refSeq RNA*

Description

A dataset containing a part of refSeq RNA.

Usage

```
data(sample_refMrna.grch37.fa)
```

Format

A data frame with 1 column.

sample_result_INDEL_CLASS1_ALL *Analyzed Result for INDEL CLASS1*

Description

Analyzed Result for INDEL CLASS1

Usage

```
data(sample_result_INDEL_CLASS1_ALL)
```

sample_result_INDEL_CLASS2_ALL
Analyzed Result for INDEL CLASS2

Description

Analyzed Result for INDEL CLASS2

Usage

```
data(sample_result_INDEL_CLASS2_ALL)
```

sample_result_SeqFragment_CLASS1_ALL
Analyzed Result for A DNA Fragment CLASS1

Description

Analyzed Result for A DNA Fragment CLASS1

Usage

```
data(sample_result_SeqFragment_CLASS1_ALL)
```

sample_result_SeqFragment_CLASS2_ALL
Analyzed Result for A DNA Fragment CLASS2

Description

Analyzed Result for A DNA Fragment CLASS2

Usage

```
data(sample_result_SeqFragment_CLASS2_ALL)
```

sample_result_SNV_CLASS1_ALL
Analyzed Result for SNV CLASS1

Description

Analyzed Result for SNV CLASS1

Usage

```
data(sample_result_SNV_CLASS1_ALL)
```

sample_result_SNV_CLASS2_ALL *Analyzed Result for SNV CLASS2*

Description

Analyzed Result for SNV CLASS2

Usage

```
data(sample_result_SNV_CLASS2_ALL)
```

sample_result_SVFusion_CLASS1_ALL *Analyzed Result for SV Fusion CLASS1*

Description

Analyzed Result for SV Fusion CLASS1

Usage

```
data(sample_result_SVFusion_CLASS1_ALL)
```

sample_result_SVFusion_CLASS2_ALL *Analyzed Result for SVFusion CLASS2*

Description

Analyzed Result for SVFusion CLASS2

Usage

```
data(sample_result_SVFusion_CLASS2_ALL)
```

sample_rna_exp *A Format / Sample file for RNA Expression Information*

Description

A dataset containing the RNA expression amount of patient for each gene.

Usage

```
data(sample_rna_exp)
```

Format

A data frame with 22 rows and 3 variables

sample_sv_bnd *A Format / Sample file for Annotated vcf file.*

Description

A dataset containing the variant information of a patient.

Usage

```
data(sample_sv_bnd)
```

Format

A data frame with 9 rows and variables including "Chr" "Start" "End" "Ref" "Alt (BND format)" "Func.refGene (exonic, intron, intergenic, ...)" "ExonicFunc.refGene (exonic nonsynonymous, synonymous, insertion, ...)" "mateID (e.g., SVMERGE1_1)"

sample_vcf.annovar *A Format / Sample file for Annotated vcf file basef on Annovar.*

Description

A dataset containing the variant information of a patient.

Usage

```
data(sample_vcf.annovar)
```

Format

A data frame with 9 rows and variables including "Chr" "Start" "End" "Ref" "Alt" "Func.refGene (exonic, intron, intergenic, ...)" "ExonicFunc.refGene (exonic nonsynonymous, synonymous, insertion, ...)" "AAChange.refGene (e.g., SLCO1C1:NM_001145944:exon7:c.692_693insG:p.L231fs ...)"

sample_vcf.snps *A Format / Sample file for snp informatin.*

Description

A dataset containing snps information of a patient.

Usage

```
data(sample_vcf.snps)
```

Format

A data frame with variables including "#CHROM" "POS" "ID" "REF" "ALT" "QUAL" "FILTER" "INFO" "FORMAT".

sample_vcf.vep *A Format / Sample file for Annotated vcf file based on VEP.*

Description

A dataset containing the variant information of a patient.

Usage

```
data(sample_vcf.vep)
```

Format

A data frame with variables including "#Uploaded_variation" "Location" "Allele" "Gene" "Feature" "Feature_type" "Consequence" "cDNA_position" "CDS_position" "Protein_position" "Amino_acids Codons" "Existing_variation" "Extra"

TestAnalysis *Execute Sample Analysis*

Description

Execute Sample Analysis

Usage

```
TestAnalysis()
```

Value

void

Index

∗Topic **datasets**
  sample_copynum, 37
  sample_hla_table_c1, 37
  sample_hla_table_c2, 37
  sample_refFlat.grch37, 38
  sample_refMrna.grch37.fa, 38
  sample_rna_exp, 40
  sample_sv_bnd, 41
  sample_vcf.annovar, 41
  sample_vcf.snps, 41
  sample_vcf.vep, 42
∗Topic **result**
  sample_result_INDEL_CLASS1_ALL, 38
  sample_result_INDEL_CLASS2_ALL, 39
  sample_result_SeqFragment_CLASS1_ALL, 39
  sample_result_SeqFragment_CLASS2_ALL, 39
  sample_result_SNV_CLASS1_ALL, 39
  sample_result_SNV_CLASS2_ALL, 40
  sample_result_SVFusion_CLASS1_ALL, 40
  sample_result_SVFusion_CLASS2_ALL, 40

Export_Summary_Entire_Fragments, 2
Export_Summary_Fragments, 3
Export_Summary_IndelSV, 4
Export_Summary_IndelSV_perFragments, 5
Export_Summary_SNV, 6

MainEntireRegionClass1, 7
MainEntireRegionClass2, 9
MainINDELClass1, 11
MainINDELClass2, 14
MainSeqFragmentClass1, 18
MainSeqFragmentClass2, 20
MainSNVClass1, 22
MainSNVClass2, 26
MainSVFUSIONClass1, 30
MainSVFUSIONClass2, 33

sample_copynum, 37
sample_hla_table_c1, 37
sample_hla_table_c2, 37
sample_refFlat.grch37, 38
sample_refMrna.grch37.fa, 38
sample_result_INDEL_CLASS1_ALL, 38
sample_result_INDEL_CLASS2_ALL, 39
sample_result_SeqFragment_CLASS1_ALL, 39
sample_result_SeqFragment_CLASS2_ALL, 39
sample_result_SNV_CLASS1_ALL, 39
sample_result_SNV_CLASS2_ALL, 40
sample_result_SVFusion_CLASS1_ALL, 40
sample_result_SVFusion_CLASS2_ALL, 40
sample_rna_exp, 40
sample_sv_bnd, 41
sample_vcf.annovar, 41
sample_vcf.snps, 41
sample_vcf.vep, 42

TestAnalysis, 42