Package 'epitopeR'

February 15, 2023

Type Package

Title Predict Peptide-MHC Binding

Version 1.0.0

Description A suite of tools to predict peptide MHC (major histocompatibility complex) presentation in the context of both human and mouse. Polymorphic peptides between self and foreign proteins are identified. The ability of peptides to bind self MHC is assessed and scored. Based on half maximal inhibitory concentration as queried through the immune epitope database API <http://tools.iedb.org> using user defined methods, the foreign peptides most likely to be presented are output along with their predicted binding strength, amino acid position, the protein from which each peptide was derived, and the presenting allele. ``References:" Vita R, Mahajan S, Overton JA, Dhanda SK, Martini S, Cantrell JR, Wheeler DK, Sette A, Peters B. <doi:10.1093/nar/gky1006>.

Depends R (>= 4.2.0)

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0), ggseqlogo, ggVennDiagram

VignetteBuilder knitr

Imports dplyr, fs, httr, janitor, purrr, rlang, stringr, tibble, tidyverse, utils, readr, Biostrings, seqinr

Config/testthat/edition 3

NeedsCompilation no

Author Joan Zhang [aut, cre], Aileen Johnson [aut], Christian P Larsen [cph, aut]

Maintainer Joan Zhang <joan.zhang@emory.edu>

Repository CRAN

Date/Publication 2023-02-15 12:50:13 UTC

R topics documented:

`core_mut` *Determine presence of mutation in core binding sequence*

Description

The core_mut() function appends a new column to the peptide dataframe, identifying those that have a mutation in the core binding sequence. Since MHCII has an open binding pocket, it presents peptides that may be several amino acids longer than the core sequence pattern required to bind a particular MHC. In some cases, the user may want to filter their results, in order to keep only peptides with a mutation in the core binding sequence (when compared to the equivalent self peptide). In order to achieve this, the stimulating and self antigens are aligned using a multiple sequence alignment tool, from the bioconductor package msa. The sequence positions of the core in the stimulating peptide are determined, and sequences are kept only if there is a sequence difference between stimulating and self peptides at that position.

Usage

```
core_mut(dat_in, ag_stim, ag_self)
```

Arguments

`dat_in`	data frame, output of mhcII(). The stimulating peptide, core pattern, start, and end positions will be pulled from this dataframe.
`ag_stim`	string, amino acid sequence of the stimulating antigen
`ag_self`	string, amino acid sequence of the self antigen

Value

data frame, peptide with flag of whether or not a mutation is in the core binding sequence

mhcI *Prediction of Nonself Peptide Presentation on Human MHC Class I*

Description

It determines which non-self peptides can be presented by a given HLA class I allele. This function takes a sequence for a stimulating antigen and the corresponding self antigen, and given a defined sequence length, queries the IEDB API with the user's choice of peptide binding prediction method. The set of peptides present in the results for the stimulating antigen but not the self antigen are then carried forward as non-self peptides. If desired, the user can adjust the default thresholds (by IC50 binding affinity or percentile rank) used to define "strong" and "weak" binders. The output is a dataframe of non-self peptides that are predicted to bind to the presenting allele.

Usage

```
mhcI(
  ag_present,
  ag_stim,
  ag_self = "",
  seq_len = "9",
  fd_out = as.character(paste0(tempdir(), "/", "outputs", "/")),
  method = "consensus",
  thold_ic50 = c(50, 500),
  thold_pect_rank = c(2, 10),
  url_iedb = "http://tools-cluster-interface.iedb.org/tools_api/mhci/",
  noneself = TRUE
)
```

Arguments

ag_present	character vector, presenting allele, formatting examples - A*01:01 or HLA-A*01:01
ag_stim	character vector, stimulating antigen, can either be an HLA class I allele entered in the same format as ag_present, or a character vector of the amino acid sequence of the protein
ag_self	character, self antigen, can either be an HLA class I allele entered in the same format as ag_present, or a character vector of the amino acid sequence of the protein
seq_len	string, length of peptides to consider
fd_out	string, output folder name, set to tempdir()
method	string, IEDB prediction method to be used. Options are ann, comblib_sidney2008, consensus, netmhccons, netmhcpan, netmhcpan_ba netmhcpan_el, netmhcstabpan, pickpocket, recommended, smm, smmpmbec. Default is netmhcstabpan

thold_ic50	vector. Defines the thresholds required to be included in results, and to be labeled, "strong" or "weak" binder. Multiple prediction methods are used, each of which provide different raw outputs (i.e. IC50, "strength", "score"). Our justification for the default thresholds is listed in the mhcI_hu vignette, however the user may choose to specify alternate cutoffs if desired.
thold_pect_rank	vector, IEDB adjusts all outputs in comparison to a set of random natural peptides in order to determine an normalized adjusted percentile rank. With normalized ranks, the same thresholds can be used across different methods. Default thresholds are <2% for strong binders and <10% for weak binders.
url_iedb	string, iedb api url
noneself	logic, whether to subtract self peptide from stim. If TRUE|T, self peptides will be subtracted from stim. If FALSE|F no self subtraction.

Value

data frame, MHC I binding prediction result table

Examples

```
mhcI(ag_present=c("HLA-A*03:01"),
ag_stim=c("A_01_01","A_02_06"),
ag_self=c("B_07_03"),
method = "rec")
```

---

mhcII *Prediction of Nonself Peptide Presentation on Human MHC Class II*

Description

It determines which non-self peptides can be presented by a given HLA class II allele. This function takes a sequence for a stimulating antigen and the corresponding self antigen, and given a defined sequence length, queries the IEDB API with the user's choice of peptide binding prediction method. The set of peptides present in the results for the stimulating antigen but not the self antigen are then carried forward as non-self peptides. If desired, the user can adjust the default thresholds (by IC50 binding affinity or percentile rank) used to define "strong" and "weak" binders. The output is a dataframe of non-self peptides that are predicted to bind to the presenting allele.

Usage

```
mhcII(
  ag_present,
  ag_stim,
  ag_self = "",
  seq_len = "15",
```

```
  fd_out = as.character(paste0(tempdir(), "/", "outputs", "/")),
  method = "netmhciipan",
  cutoff_score = list(cutoff_netpan = c(50, 500), cutoff_comblib = c(50, 500),
   cutoff_nn_align = c(50, 500), cutoff_smm_align = c(50, 500), cutoff_sturniolo = c(2),
    cutoff_el = c(2, 10)),
  cutoff_rank = c(2, 10),
  url_iedb = "http://tools-cluster-interface.iedb.org/tools_api/mhcii/",
  noneself = TRUE,
  nm_self = "",
  nm_stim = ""
)
```

Arguments

ag_present	character vector, presenting allele, formatted with either "_", "*", or":" separating loci, antigen, and allele. For example, "DRB1_08_01".
ag_stim	character vector, stimulating antigen, can either be an HLA class II allele entered in the same format as ag_present, or a character vector of the amino acid sequence of the protein
ag_self	character, self antigen, can either be an HLA class II allele entered in the same format as ag_present, or a character vector of the amino acid sequence of the protein
seq_len	string, length of peptides to consider
fd_out	string, output folder name; default output is current working directory
method	string, IEDB prediction method to be used. Options are "netmhciipan', "netmhciipan_el" or "recommended." Default is netmhciipan.
cutoff_score	list of vectors. Defines the thresholds required to be included in results, and to be labeled, "strong" or "weak" binder. Multiple prediction methods are used, each of which provide different raw outputs (i.e. IC50, "strength", "score"). Our justification for the default thresholds is listed in the mhcII_hu vignette, however the user may choose to specify alternate cutoffs if desired.
cutoff_rank	vector, IEDB adjusts all outputs in comparison to a set of random natural peptides in order to determine an normalized adjusted percentile rank. With normalized ranks, the same thresholds can be used across different methods. Default thresholds are <2% for strong binders and <10% for weak binders.
url_iedb	string, iedb api url
noneself	logic, whether to subtract self peptide from stim. If TRUE|T, self peptides will be subtracted from stim. If FALSE|F no self subtraction.
nm_self	string of the name of self antigen you want use in the output table
nm_stim	string of the name of stime antigen yiuo want use in the output table

Value

data frame, MHC II binding prediction result table

utils *Basic functions of the package*

Description

val_ag_name() validation of name of binding locus

Usage

```
val_ag_name(ag_present)

preproc(allele_in, link)

pull_seq(alleles_in, tbl_ref_in)

comb_pred_tbl(nm_method, nm_sht, nm_fd, thold_score, thold_rank)

comb_pred_tbl_mhcI(nm_method, nm_fd, thold_score, thold_rank)

find_nonself(dat_in)

pull_ag_self(vec_in)

find_core_mut(dat_in)

align_seq(seq1, seq2, gapopening = 0, gapextension = 8)

pull_obj_name(x)
```

Arguments

ag_present	a vector of locus name(s) to make binding predictions for
	preproc() format and validate allele names
allele_in	a vector contains allele name(s)
link	a string of url of MHC I or II api
	pull_seq() pull out sequence of each allele based on ref table
alleles_in	vector, allele names
tbl_ref_in	dataframe, reference table, default is human_all.csv from github
	comb_pred_tbl() combine individual prediction tables by method, exclude none-binders and keep strong and weak binders only
nm_method	string, prediction method used for IEDB prediction
nm_sht	string, short name of alleles
nm_fd	string, folder name which contains predict tables from IEDB
thold_score	list of vectors, binder thresholds by ic50 score

thold_rank	vector, binder thresholds by percentile rank find_nonself() find nonself binding peptides
dat_in	dataframe with pep_stim, core, pep_self selected from pull_ag_self align_seq() align protein sequences
vec_in	dataframe with pep_stim, core, aligned ag_stim and ag_self columns find_core_mut() find mutation position to core
seq1	string, unaligned sequence of ag_stim
seq2	string, unaligned sequence of ag_self
gapopening	numeric, the cost for opening a gap in the alignment.
gapextension	numeric, the incremental cost incurred along the length of the gap in the alignment pull_obj_name()
x,	name of an object

Index