# Package 'RHybridFinder'

July 21, 2025

**Type** Package

**Title** Identification of Hybrid Peptides in Immunopeptidomic Analyses

**Version** 0.2.0

**Maintainer** Frederic Saab <frederic.saab@umontreal.ca>

**Description** Tool for the analysis Mass Spectrometry (MS) data in the context of immunopeptidomic analysis for the identification of hybrid peptides and the predictions of binding affinity of all peptides using 'netMHCpan' <doi:10.1093/nar/gkaa379> while providing a summary of the netMHCpan output. 'RHybridFinder' (RHF) is destined for researchers who are looking to analyze their MS data for the purpose of identification of potential spliced peptides. This package, developed mainly in base R, is based on the workflow published by Faridi et al. in 2018 <doi:10.1126/sciimmunol.aar3947>.

**Imports** doParallel, foreach, seqinr

**Depends** R (>= 3.5.0)

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**LazyData** True

**LazyDataCompression** bzip2

**Author** Frederic Saab [aut, cre],
Peter Kubiniok [aut]

**Repository** CRAN

**Date/Publication** 2021-08-17 16:30:24 UTC

# Contents

checknetMHCpan . . . . . 2
db_Human_Liver_AUTD17 . . . . . 3
db_rerun_Human_Liver_AUTD17 . . . . . 4
denovo_Human_Liver_AUTD17 . . . . . 5
export_checknetmhcpan_results . . . . . 6
export_HybridFinder_results . . . . . 7
export_step2_results . . . . . 8
HybridFinder . . . . . 9
mhc_check . . . . . 11
netmhcpan_list_alleles . . . . . 12
step2_wo_netMHCpan . . . . . 13

**Index** **15**

---

`checknetMHCpan` *checknetMHCpan*

---

## Description

checknetMHCpan, utilizes the file from the second (PEAKS) run and analyzes the data with netMHCpan in order to provide the peptide binding affinity to different HLA/MHC alleles.

## Usage

```
checknetMHCpan(
  netmhcpan_directory,
  netmhcpan_alleles,
  peptide_rerun,
  HF_step1_output,
  export_files = FALSE,
  export_dir = NULL
)
```

## Arguments

`netmhcpan_directory`
: the directory in which the netMHCpan file is.

`netmhcpan_alleles`
: vector of comma-separated alleles for which these peptides should be analyzed (i.e HLA_alleles_Exp1<- c("HLA-A*02:01", "HLA-A*03:01", "HLA-A24:02"))

`peptide_rerun`
: dataframe containing the results of the second run

`HF_step1_output`
: the HybridFinder output containing the potential splicing categorizations obtained with the HybridFinder function (HybridFinder) based on the matching of fragment pairs of peptides in 1 or 2 proteins. This parameter can be provided either by loading the .csv exported file, or if the results #' object still is in the

| | |
|---|---|
| | global environment (i.e results_HF_Exp1), then it can be accessed by simply writing "results_HF_Exp1[[1]]". |
| export_files | a boolean parameter for exporting the dataframes into files in the next parameter for the output directory, Default: FALSE |
| export_dir | export_dir the output directory for the results files if export_files=TRUE, Default: NULL |

## Details

The ability to check the peptide binding affinity to the different MHC/HLA molecules is essential for assessing the antigenicity of all peptides. This function thus uses netMHCpan (Reynisson et al., 2020) for the generation of binding affinty results.

## Value

1. netMHCpan results pertaining to the binding affinity of all peptides in the database search results (in long- and wide- format, with data tidying in the wide format in order to compute the amount of HLA molecules to which a peptide is strong/weak/non-binder binder)(dataframe)
2. netMHCpan results pertaining to the binding affinity of the hybrid peptides to the MHC molecules (in long- and wide- format, with data tidying in the wide format in order to compute the amount of HLA molecules to which a peptide is strong/weak/non-binder binder) (dataframe)
3. the database search rerun with the categorizations already determined in step1 (HybridFinder Function)

(datafrane)

## Examples

```
## Not run:
  results_checknetmhcpan_Exp1<- checknetMHCpan('/usr/local/bin', alleles,
  peptide_rerun, Exp1_HF_results[[1]])
  results_checknetmhcpan_Exp1 <- checknetMHCpan('/usr/local/bin', alleles,
  peptide_rerun, Exp1_HF_results_denovo_w_spliced)

## End(Not run)
```

---

# db_Human_Liver_AUTD17 *db_Human_Liver_AUTD17*

---

## Description

database search results from the first run using PEAKS software, on the raw dataset from the HLA ligand Atlas Human Liver sample of AutDonor17

## Usage

```
db_Human_Liver_AUTD17
```

## Format

A data frame with 6108 rows and 18 variables:

`Peptide` character Peptide

`X.10lgP` double X.10lgP

`Mass` double Mass

`Length` integer Peptide length

`ppm` double ppm

`m.z` double mass-to-charge

`Z` integer charge

`RT` double Retention Time

`Area` double Area

`Fraction` integer Fraction

`Id` integer Id

`Scan` character Scan

`from.Chimera` character from.Chimera

`Source.File` character Source.File

`Accession` character Accession

`PTM` character Post Translational Modification

`AScore` character AScore

`Found.By` character Found.By

## Details

database search results from the first run using PEAKS software, on the raw dataset from the HLA ligand Atlas Human Liver sample of AutDonor17

---

`db_rerun_Human_Liver_AUTD17`

*db_rerun_Human_Liver_AUTD17*

---

## Description

database search results from the second run using PEAKS software, using the raw file from HLA ligand ATLAS, Human Liver AutDonor 17

## Usage

```
db_rerun_Human_Liver_AUTD17
```

## Format

A data frame with 6315 rows and 18 variables:

`Peptide` character Peptide

`X.10lgP` double X.10lgP

`Mass` double Mass

`Length` integer Peptide sequence

`ppm` double ppm

`m.z` double mass-to-charge

`Z` integer charge

`RT` double Retention Time

`Area` double Area

`Fraction` integer Fraction

`Id` integer Id

`Scan` character Scan

`from.Chimera` character from.Chimera

`Source.File` character Source.File

`Accession` character Accession

`PTM` character Post-Translational Modification

`AScore` character AScore

`Found.By` character Found.By

## Details

database search results from the second run using PEAKS software, using the raw file from HLA ligand ATLAS, Human Liver AutDonor 17

---

`denovo_Human_Liver_AUTD17`

*denovo_Human_Liver_AUTD17*

---

## Description

denovo sequencing results obtained using PEAKS software, and the raw file of Human liver from autDonor 17

## Usage

```
denovo_Human_Liver_AUTD17
```

## Format

A data frame with 50114 rows and 18 variables:

`Fraction` integer Fraction

`Source.File` character Source.File

`Feature` character Feature

`Peptide` character Peptide

`Scan` character Scan

`Tag.length` integer Tag.length

`Denovo.score` integer Denovo.score

`ALC....` integer Average Local Confidence

`Length` integer peptide length

`m.z` double mass-to-charge(m/z)

`z` integer charge

`RT` double Retention Time

`Predict.RT` character Predict.RT

`Mass` double Mass

`ppm` double ppm

`local.confidence....` character confidence score per residue

`tag....0..` character tag

`mode` character mode

## Details

denovo sequencing results obtained using PEAKS software, and the raw file of Human liver from autDonor 17

---

`export_checknetmhcpan_results`

*export_checknetmhcpan_results*

---

## Description

this function allows to export the results generated from checknetMHCpan()

## Usage

```
export_checknetmhcpan_results(list_checknetMHCpan_results, export_dir)
```

### Arguments

| | |
|---|---|
| list_checknetMHCpan_results | the results generated from running checknetMHCpan() |
| export_dir | the export directory where the results .csv files should be exported. |

### Details

In order to be able to have the checknetMHCpan() function results exported, this function will come in handy. Please note that this function is also part of the checknetMHCpan() function (if export_files is set to TRUE and a valid export directory is indicated)

### Value

exports a folder containing three files

1. netMHCpan results in long format (the original output)(.csv file)
2. netMHCpan results tidied (in wide format) so as to summarize the information per peptide (.tsv tab-separated file)
3. the updated database search results which contain the categorizatiosn of the peptides found in common between the 2nd database search and the HybridFinder function (.csv file)

### Examples

```
## Not run:
 export_checknetmhcpan_results(results_checknetMHCpan_Human_Liver_AUTD17, folder_Human_Liver_AUTD17)

## End(Not run)
```

---

## export_HybridFinder_results

*export_HybridFinder_results*

---

### Description

this function allows to export the results list obtained in the HybridFinder() function.

### Usage

```
export_HybridFinder_results(results_list, export_dir)
```

### Arguments

| | |
|---|---|
| results_list | the results list obtained with the HybridFinder() function. |
| export_dir | the export directory |

## Details

In order to be able to have the HybridFinder() results list exported, this function will come in handy. Please note that this function is also part of the HybridFinder() function, therefore if you set export_files=TRUE and you indicate the export directory in export_dir in the HybridFinder() function, you would have the exact same outcome.

## Value

exports a folder containing three files

1. the HybridFinder output - the spectra that made it to the end with their respective columns (ALC, m/z, RT, Fraction, Scan) and a categorization column which denotes their potential splice type (-cis, -trans) or whether they are linear (the entire sequence was matched in proteins in the proteome database). Potential cis- & trans-spliced peptide are peptides whose fragments were matched with fragments within one protein, or two proteins, respectively.
2. list of potential hybrid peptides (excluding the linear peptides) (.csv file)
3. the merged proteome consisting of the reference proteome along with the hybrid proteome added at the end of the file with the sequence names following the pattern "sp|denovo_HF_fake_protein" along with a digit at the end (1,2,3,4,4,etc.) (.fasta file)

## See Also

write.fasta

## Examples

```
## Not run:
 export_results(results_HybridFinder_Human_Liver_AUTD17,folder_Human_Liver_AUTD17)

## End(Not run)
```

---

# export_step2_results *export_step2_results*

---

## Description

this function allows to export the results generated from step2_wo_netmhcpan.

## Usage

```
export_step2_results(step2_RHF_results_Exp1, export_dir)
```

## Arguments

| | |
|---|---|
| step2_RHF_results_Exp1 | the results generated from running step2_wo_netMHCpan() |
| export_dir | the export directory where you would like to have the .csv file saved. |

## Details

Since netMHCpan is not compatible with Windows OS, the package offers an alternative by outputting the input for netMHCpan and as well the database results with their respective categorizations (cis, trans) established in step1.

## Value

exports a folder containing 2 files

1. the peptide list to be entered in a netMHCpan-ready format,(.csv)
2. the updated database search results which contain the categorizatiosn of the peptides found in common between the 2nd database search and the HybridFinder function (.csv file)

## Examples

```
## Not run:
 export_step2_results(results_step2_Human_Liver_AUTD17, folder_Human_Liver_AUTD17)

## End(Not run)
```

---

`HybridFinder` *HybridFinder*

---

## Description

This function takes in three mandatory inputs: (1) all denovo candidates (2) database search results and (3) the corresponding proteome fasta file. The function's role is to extract high confidence de novo peptides and to search for their existence in the proteome, whether the entire peptide sequence or its pair fragments (in one or two proteins).

## Usage

```
HybridFinder(
  denovo_candidates,
  db_search,
  proteome_db,
  customALCcutoff = NULL,
  with_parallel = TRUE,
  customCores = 6,
  export_files = FALSE,
  export_dir = NULL
)
```

## Arguments

| | |
|---|---|
| denovo_candidates | dataframe containing all denovo candidate peptides |
| db_search | dataframe containing the database search peptides |
| proteome_db | path to the proteome FASTA file |
| customALCcutoff | the default is calculated based on the median ALC of the assigned spectrum groups (spectrum groups that match in the database search results and in the denovo sequencing results) where also the peptide sequence matches, Default: NULL |
| with_parallel | for faster results, this function also utilizes parallel computing (please read more on parallel computing in order to be sure that your computer does support this), Default: TRUE |
| customCores | custom amount of cores strictly higher than 5, Default: 6 |
| export_files | a boolean parameter for exporting the dataframes into files in the next parameter for the output directory, Default: FALSE, Default: FALSE |
| export_dir | the output directory for the results files if export_files=TRUE, Default: NULL, Default: NULL |

## Details

This function is based on the published algorithm by Faridi et al. (2018) for the identification and categorization of hybrid peptides. The function described here adopts a slightly modified version of the algorithm for computational efficiency. The function starts by extracting unassigned denovo spectra where the Average Local Confidence (assigned by PEAKS software), is equivalent to the ALC cutoff which is based on the median of the assigned spectra (between denovo and database search). The sequences of all peptides are searched against the reference proteome. If there is a hit then, then, the peptide sequence within a spectrum group considered as being linear and each spectrum group is is then filtered so as to keep the highest ALC-ranking spectra. Then, the rest of the spectra (spectra that did not contain any sequence that had an entire match in the proteome database) then undergo a "cutting" procedure where each sequence yields n-2 sequences (with n being the length of the peptide. That is if the peptide contains 9 amino acids i.e NTYASPRFK, then the sequence is cut into a combination of 7 sequences of 2 fragment pairs each i.e fragment 1: NTY and fragment 2: ASPRFK, etc).These are then searched in the proteome for hits of both peptide fragments within a same protein, spectra in which sequences have fragment pairs that match within a same protein, these are considerent to be potentially cis-spliced. Potentially cis-spliced spectrum groups are then filtered based on the highest ranking ALC. Spectrum groups not considered to be potentially cis-spliced are further checked for potential trans-splicing. The peptide sequences are cut again in the same fashion, however, this time peptide fragment pairs are searched for matches in two proteins. Peptide sequences whose fragment pairs match in 2 proteins are considerend to be potentially trans-spliced. The same filtering for the highest ranking ALC within each peptide spectrum group. The remaining spectra that were neither assigned as linear nor potentially spliced (neither cis- nor trans-) are then discarded. The result is a list of spectra along with their categorizations (Linear, potentially cis- and potentially trans-) Potentially cis- and trans-spliced peptides are then concatenated and then broken into several "fake" proteins and added to the bottom of the reference proteome. The point of this last step is to create a merged proteome (consisting of the reference

proteome and the hybrid proteome) which would be used for a second database search. After the second database search the checknetmhcpan function or the step2_wo_netMHCpan function can be used in order to obtain the final list of potentially spliced peptides. Article: Faridi P, Li C, Ramarathinam SH, Vivian JP, Illing PT, Mifsud NA, Ayala R, Song J, Gearing LJ, Hertzog PJ, Ternette N, Rossjohn J, Croft NP, Purcell AW. A subset of HLA-I peptides are not genomically templated: Evidence for cis- and trans-spliced peptide ligands. Sci Immunol. 2018 Oct 12;3(28):eaar3947. <doi: 10.1126/sciimmunol.aar3947>. PMID: 30315122.

## Value

The output is a list of 3 dataframes containing:

1. the HybridFinder output (dataframe) - the spectra that made it to the end with their respective columns (ALC, m/z, RT, Fraction, Scan) and a categorization column which denotes their potential splice type (-cis, -trans) or whether they are linear (the entire sequence was matched in proteins in the proteome database). Potential cis- & trans-spliced peptide are peptides whose fragments were matched with fragments within one protein, or two proteins, respectively.
2. character vector containing potentially hybrid peptides (cis- and trans-)
3. list containing the reference proteome and the "fake" proteins added at the end with a patterned naming convention (sp|denovo_HF_fake_protein) made up of the concatenated potential hybrid peptides.

## See Also

read.fasta,s2c

## Examples

```
## Not run:
 hybridFinderResult_list <- HybridFinder(denovo_candidates, db_search,
 proteome, export = TRUE, output_dir)
 hybridFinderResult_list <- HybridFinder(denovo_candidates, db_search,
 proteome)
 hybridFinderResult_list <- HybridFinder(denovo_candidates, db_search,
 proteome, export = FALSE)

## End(Not run)
```

---

# mhc_check *mhc_check*

---

## Description

this function only contains the alleles list, read by netMHCpan, the list was retrieved by reading the file exported from netMHCpan, using the following command line "netMHCpan -listMHC"

## Usage

```
mhc_check(netmhcpan_alleles)
```

### Arguments

`netmhcpan_alleles`
: the netmhcpan alleles to be used for the netmhcpan call.

### Details

a custom error is printed in case the allele is not written correctly

### Value

- returns a custom error message if MHC/HLA allele(s) are not written correctly
- returns nothing if there are no issues. If HLA alleles are not written correctly

### Examples

```
if (interactive()) {
 mhc_check("HLA-A02:01")
 mhc_check("HLA-A0201")
}
```

---

## `netmhcpan_list_alleles` *netmhcpan_list_alleles*

---

### Description

the list of alleles in the acceptable format for netMHCpan

### Usage

```
netmhcpan_list_alleles
```

### Format

A data frame with 1024 rows and 1 variables:

`V1` character the alleles

### Details

the list of alleles in the acceptable format for netMHCpan

step2_wo_netMHCpan *step2_wo_netMHCpan*

## Description

This function helps retrieve the categorizations for the peptides from step 1 and apply them to those that are matched in the second database search.

## Usage

```
step2_wo_netMHCpan(
  peptide_rerun,
  HF_step1_output,
  export_files = FALSE,
  export_dir = NULL
)
```

## Arguments

| | |
|---|---|
| peptide_rerun | dataframe containing the results of the second database PEAKS search. |
| HF_step1_output | the HybridFinder output containing the potential splicing categorizations obtained with the HybridFinder function (HybridFinder) based on the matching of fragment pairs of peptides in 1 or 2 proteins. This parameter can be provided either by loading the .csv exported file, or if the results #' object still is in the global environment (i.e results_HF_Exp1), then it can be accessed by simply writing "results_HF_Exp1[[1]]". |
| export_files | a boolean parameter for exporting the dataframes into files in the next parameter for the output directory, Default: FALSE |
| export_dir | export_dir the output directory for the results files if export_files=TRUE, Default: NULL |

## Details

In special cases where the PC runs on windows OS, since it would only be possible to use the web version of netMHCpan, this function returns the peptide input file for the webversion of netMHCpan. Also, this function outputs the database search rerun results with their categorizations (into potentially cis and potentially trans) obtained from the first step (HybridFinder).

## Value

1. the input file for the web version of netMHCpan (dataframe)
2. the database search rerun with the categorizations already determined in the previous step. (character vector)

## Examples

```
if (interactive()) {
data(package="RHybridFinder", "db_rerun_Human_Liver_AUTD17")
results_checknetmhcpan_Human_Liver_AUTD17<- step2_wo_netMHCpan(db_rerun_Human_Liver_AUTD17,
    results_HybridFinder_Human_Liver_AUTD17[[1]])
}
```

# Index

∗ **datasets**
db_Human_Liver_AUTD17, 3
db_rerun_Human_Liver_AUTD17, 4
denovo_Human_Liver_AUTD17, 5
netmhcpan_list_alleles, 12

checknetMHCpan, 2

db_Human_Liver_AUTD17, 3
db_rerun_Human_Liver_AUTD17, 4
denovo_Human_Liver_AUTD17, 5

export_checknetmhcpan_results, 6
export_HybridFinder_results, 7
export_step2_results, 8

HybridFinder, 9

mhc_check, 11

netmhcpan_list_alleles, 12

read.fasta, *11*

s2c, *11*
step2_wo_netMHCpan, 13

write.fasta, *8*