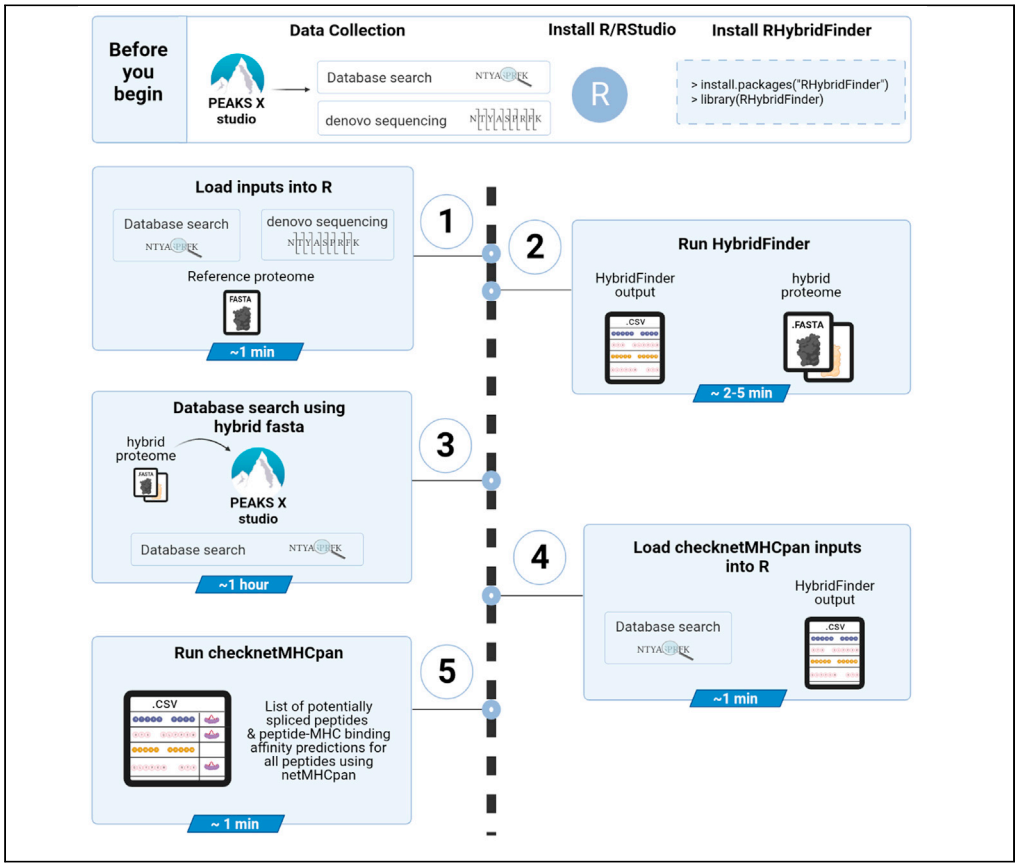
STAR Protocols

CellPress
OPEN ACCESS

Protocol

RHybridFinder: An R package to process immunopeptidomic data for putative hybrid peptide discovery



Identification of proteasomal spliced peptides (PSPs) by mass spectrometry (MS) is not possible with traditional search engines. Here, we provide a protocol for running RHybridFinder (RHF), an R package for the computational inference of putative PSPs detected by MS. RHF extracts high confidence scored de novo sequenced peptides identified by PEAKS software. Those peptides are then matched to protein databases to infer cis- or trans-spliced MHC-associated peptides. RHF is relatively fast and straightforward. PSPs have to be validated experimentally.

Frederic Saab, David J. Hamelin, Qing Ma, ..., Anthony W. Purcell, Peter Kubiniok, Etienne Caron

peterkubiniok@gmail.com (P.K.)
etienne.caron@umontreal.ca (E.C.)

Highlights

RHybridFinder (RHF) is an improved R package for the discovery of spliced peptides

RHF builds upon the algorithm published in Faridi et al. (2018)

RHF uses MS data analyzed in PEAKS

The spliced peptide candidates generated by RHF need to be validated experimentally

Saab et al., STAR Protocols 2, 100875
December 17, 2021
https://doi.org/10.1016/j.xpro.2021.100875

STAR Protocols

CellPress
OPEN ACCESS

Protocol

RHybridFinder: An R package to process immunopeptidomic data for putative hybrid peptide discovery

Frederic Saab,[1,5] David J. Hamelin,[1] Qing Ma,[4] Kevin A. Kovalchik,[1] Isabelle Sirois,[1] Pouya Faridi,[2] Chen Li,[2] Anthony W. Purcell,[2] Peter Kubiniok,[1,5,*] and Etienne Caron[1,3,6,*]

[1]CHU Sainte-Justine Research Center, Montreal, QC H3T 1C5, Canada

[2]Infection and Immunity Program and Department of Biochemistry and Molecular Biology, Biomedicine Discovery Institute, Monash University, Clayton, VIC 3800, Australia

[3]Department of Pathology and Cellular Biology, Faculty of Medicine, Université de Montréal, Montreal, QC H3T 1J4, Canada

[4]School of Electrical Engineering and Computer Science, Faculty of Engineering, University of Ottawa, Ottawa, ON K1N 6N5, Canada

[5]Technical contact

[6]Lead contact

*Correspondence: peterkubiniok@gmail.com (P.K.), etienne.caron@umontreal.ca (E.C.)
https://doi.org/10.1016/j.xpro.2021.100875

SUMMARY

Identification of proteasomal spliced peptides (PSPs) by mass spectrometry (MS) is not possible with traditional search engines. Here, we provide a protocol for running RHybridFinder (RHF), an R package for the computational inference of putative PSPs detected by MS. RHF extracts high confidence scored de novo sequenced peptides identified by PEAKS software. Those peptides are then matched to protein databases to infer cis- or trans-spliced major histocompatibility complex (MHC)-associated peptides. RHF is relatively fast and straightforward. PSPs have to be validated experimentally.

For complete details on the use and execution of the original protocol, please refer to Faridi et al. (2018).

BEFORE YOU BEGIN

The proteasome is recognized as the core enzymatic machinery of the antigen processing and presentation pathway wherein peptides derived from proteasomal proteolysis are selectively presented on the cell surface by MHC (major histocompatibility complex)-I molecules (Neefjes et al., 2011). In 2004, Hanada et al. discovered that the proteasome could cleave and splice peptide fragments to generate immunogenic epitopes presented by MHC class I molecules (Hanada et al., 2004). Following this groundbreaking discovery, other research groups have been able to uncover additional T cell spliced epitopes generated by the proteosome, referred in this protocol as proteasomal spliced peptides (PSPs) (Berkers et al., 2015; Dalet et al., 2011; Ebstein et al., 2016; Michaux et al., 2014; Vigneron et al., 2004).

More recently, MS-based immunopeptidomics has been used to expedite the identification of PSPs in a systematic manner, including cis- and trans-spliced peptides (Berkers et al., 2015; Faridi et al., 2018; Liepe et al., 2010, 2016; Rolfs et al., 2019; Specht et al., 2020). However, MS-based studies using different computational approaches have led to a debate around the proportion of those PSPs in the MHC class I immunopeptidome (Lichti, 2021; Mylonas et al., 2018; Wilhelm et al., 2021).

Here, we provide a protocol to run RHybridFinder (RHF), an open access and improved R package built upon the computational workflow developed by Faridi et al. (2018) for the analysis of MS data to

systematically identify putative PSPs (Faridi et al., 2018). High speed performance is the main strength of RHF in addition to be relatively straightforward to run. The main limitation is that the PSPs identified by RHF may not be genuily spliced by the proteasome in vivo. Their source and presentation should therefore be validated experimentally to move the debate forward (Figure 1).

RHybridFinder is available on CRAN (https://cran.r-project.org/package=RHybridFinder) to enable more researchers to explore those debated peptides.

Data collection

For demonstration of the output of the different RHybridFinder functions, we have used datasets from the HLA Ligand Atlas (Marcu et al., 2021) deposited in PRIDE (Proteomics IDentification Database) PXD019643.

1. Download the following mzML files and analyzed them in PEAKS:
   171002_AM_AUT01-DN17_Liver_W6-32_10%_DDA_3_400-650mz_msms4,
   171002_AM_AUT01-DN17_Liver_W6-32_10%_DDA_3_400-650mz_msms5,
   171002_AM_AUT01-DN17_Liver_W6-32_10%_DDA_3_400-650mz_msms6.
2. Analyze these files in PEAKS.

Installing Rstudio/R

RHybridFinder package has been developed in RStudio and implemented in R programming language.

3. Download & install Rstudio if not already installed: (https://www.rstudio.com/products/rstudio/download/).

Installing and loading RHybridFinder

Below are the lines needed to install the RHybridFinder package from CRAN (the Comprehensive R Archive Network) and then load it.

4. Install and load RHybridFinder by typing "install.packages("RHybridFinder") in the R console.

```
> install.packages(``RHybridFinder``)
```

5. Load RHybridFinder by typing "library(RHybridFinder)" in the R console

```
> library(RHybridFinder)
```

⚠ CRITICAL: if you copy the lines of code from here, keep in mind that you might have to re-write the quotation marks yourself.

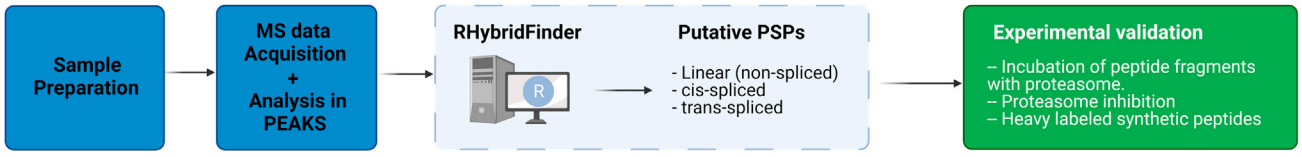


Figure 1. Overview of suggested workflow for the discovery of PSPs

We propose a four-step workflow for the identification of PSPs. The first three steps (blue squares: sample preparation, MS data acquisition and RHbridFinder enable computational exploration of putative PSPs followed by experimental validations (green square). A non-exhaustive list of possible experiments is shown for validating/gaining confidence in the identification of MHC-I peptides that are genuinely catalyzed by proteasomal splicing.

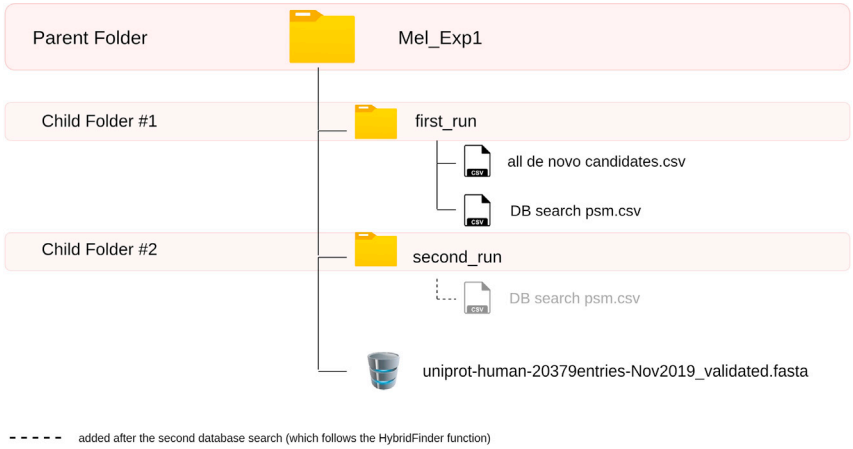


Figure 2. Recommended folder structure

The parent folder includes two child folders. The child folders include the various files that are necessary for running RHybridFinder. The dotted line (second_run) indicates that the DB search psm.csv file is added after the second DB search.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Human liver sample from autologous donor 17 - HLA Ligand Atlas	(Marcu et al., 2021)	PXD019643
Software and algorithms		
RStudio (version 1.3.1093)	RStudio website https://www.rstudio.com	SCR_000432
R (>3.5.0)	R statistical software (https://www.r-project.org/)	SCR_001905
PEAKS (PEAKS X studio)	PEAKS website: https://www.bioinfor.com//	N/A
RHybridFinder (v.0.2.0)	https://cran.r-project.org/web/packages/RHybridFinder/index.html	N/A
seqinr (v. 4.2-5)	CRAN - (Charif and Lobry, 2007)	N/A
foreach (v.1.5.1), doParallel (v. 1.0.16)	CRAN	N/A
netMHCpan (v. 4.0 &4.1)	DTU health tech: https://services.healthtech.dtu.dk (Reynisson et al., 2020)	SCR_018182
hybrid finder	Faridi et al. (2018) (workflow on which the package is based)	N/A

STEP-BY-STEP METHOD DETAILS

Step 1: Load inputs into R

Timing: 1 min

Before running HybridFinder, the inputs need to be loaded into R. We propose the following way of loading the files into R in order to facilitate the process Figure 2.

1. Create an object (folder_Exp1) for the path to the parent folder (Mel_Exp1) (but both can be named otherwise).

```
> folder_Exp1 <- file.path(''/Users/YOURUSERNAME/Desktop/Mel_Exp1'')
```

2. Import the *de novo* sequencing as well as the database results, both of which are located in the first_run child folder.
   a. *de novo* sequencing results file

```
> denovo_Exp1 <- read.csv(file = file.path(folder_Exp1, ''first_run'', "all_denovo_candida-
tes.csv"), header=TRUE, sep=",", stringsAsFactors = FALSE)
```

   b. database search results file

```
> db_search_Exp1<- read.csv (file=file.path(folder_Exp1, ''first_run'',''DB seach psm.csv''),
header=TRUE, sep='','', stringsAsFactors=FALSE)
```

3. Create an object for the path to the proteome file, located in the parent folder (folder_Exp1) (see refproteome_Exp1, in the example below). The fasta proteome will be imported in R during the HybridFinder function.

```
> refproteome_Exp1 <- file.path(folder_Exp1, ''uniprothuman-20379entries-Nov2019_
validated.fasta'')
```

⚠ CRITICAL: Please note that if you copy the file access path (in windows), you will need to switch the backslash (''\'') to a normal slash (''/'').

Access the datasets included in the R package

The RHybridFinder package also includes demonstration datasets from the HLA Ligand Atlas that have already been analyzed in PEAKS. These datasets include PEAKS de novo sequencing results and PEAKS database search results.

```
# access denovo dataset

> data(package= ''RHybridFinder'', ''denovo_Human_Liver_AUTD17'')

# access database search dataset

> data(package=''RHybridFinder'', ''db_Human_Liver_AUTD17'')
```

Note: that due to size constraints the proteome database (.fasta) file is not included in the package. It can be downloaded from the Uniprot database.

Note: In the environment tab, the denovo_Human_Liver_AUTD17 and db_Human_Liver_AUTD17 should appear. Note that if you see <promise>, after clicking on the objects, the data would appear.

Step 2: Run HybridFinder

Timing: 2–5 min (with parallelism, 8 cores) - 10–15 min (without parallelism)

In order to have a relatively short runtime, we have implemented an option to use parallel computing. However, please note that because parallel computing requires a certain amount of processing units for proper functioning, it has been made possible to also run HybridFinder without parallel computing.

Based on default parameters in the HybridFinder function, the ''all de novo candidates.csv'' file contains 16,286 peptide sequences and the runtime (parallelism with 8 cores) is of 2 min 17 s ~5 min are required for double the number of peptides. Without parallelism, the runtime ranged between 10 and 15 min for 16,286 peptide sequences.

4. Run Hybridfinder (Please refer to Table 1 in order to know more about the inputs needed) and export the results in the parent folder.

```
> HybridFinder_results_Exp1<- HybridFinder(denovo_candidates = denovo_Exp1, db_search =
db_search_Exp1, proteome_db = refproteome_Exp1,customALCcutoff = NULL, with_parallel=-
TRUE, customCores = 8, export_files= TRUE, export_dir = folder_Exp1)
```

⚠ CRITICAL: if you use the datasets included in the package, please note that they are named differently so for instance the "denovo_candidates" and "db_search" parameters should be set to the datasets loaded from the package: denovo_Human_Liver_AUTD17 and db_Human_Liver_AUTD17, respectively.

⚠ CRITICAL: Make sure to store the HybridFinder results in an object (i.e HybridFinder_results_Exp1), as the HybridFinder output dataframe will come in handy in the second function.

Note: At the end of the hybrid proteome will be the concatenated hybrid fake proteins with the name pattern 'sp|denovo_HF_fake_protein_[#]'.

Note: with_parallel is activated if set to true and if the PC has more than 5 cores.

⚠ CRITICAL: Please ensure to have a minimal number of other windows open and to save any work in other softwares prior to using HybridFinder with parallelism.

The function will output a list (Figure 3) containing: (1) the HybridFinder output containing all the denovo peptides along with their potential splice type explanation cis-/trans-, (2) a list of the step1 hybrid candidate peptides, (3) the hybrid proteome (merged proteome: the original user proteome along with the hybrid proteome composed of the concatenated candidate hybrid peptide sequences).

Note: In the example above, export_files have been set to TRUE and the export_dir has been defined which means that the files are also automatically exported. If these two parameters were not specified or were set to FALSE & NULL, the results are only stored in the Exp1_HybridFinder_results. In this case, you can still use "export_HybridFinder_results" as in the code below, where HybridFinder_results_Exp1 is the object created above for the storage of HybridFinder results.

```
> export_HybridFinder_results(HybridFinder_results_Exp1, export_dir= folder_Exp1)
```

Table 1. HybridFinder function parameters

Parameter	Description	Default value
de novo_candidates	the dataframe containing the *de novo* sequencing results	No defaults. Necessary input.
db_search	the data frame containing the database search results	No defaults. Necessary input.
db_search	the data frame containing the database search results	No defaults. Necessary input.
proteome_db	the file path to the proteome used for the database search	No defaults. Necessary input.
(Optional) customALCcutoff	A custom score cutoff that can be set by the user as long as it would be at least 85	NULL. (ALC cutoff calculated automatically as median of matching peptide sequences of assigned spectra). If set manually, minimum is 85.
with_parallel : boolean (True or False)	representing whether parallel computing should be employed for running the function.	TRUE
(Optional) customCores	If with_parallel is set to TRUE and the PC has >5 cores, the user can set a custom amount of cores to be used by the function.	6
(Optional) export_files : boolean (True or False)	by default it is set to False, however, if set to True, then the following input is essential.	FALSE
(Optional) export_dir	file path to the directory where the output files should be stored. This parameter is necessary for the export.	NULL

Figure 3. Screenshot of the HybridFinder function results

In the results list you will find 3 items: 1) a dataframe containing the HybridFinder output. 2) a character vector containing the candidate spliced peptides. 3) a list which is in a seqinr class (Charif and Lobry, 2007) containing the merged hybrid proteome.

Pause point: If you would like to conduct the rest of the protocol at a later time, either use the export functionality and then load the HybridFinder output in order to use it for the second step. Alternatively, save the objects in R in a .rda file as follows and once you want to use it again for the step 4, load checknetMHCpan inputs into R.

```
> save (HybridFinder_results_Exp1, file=file.path(folder_Exp1, ''HybridFinder_results_Exp1.rda'')

>load (file.path(folder_Exp1, ''HybridFinder_results_Exp1.rda''))
```

Step 3: Database search using hybrid Fasta

Timing: 1 h

An essential interim step must follow the HybridFinder function and consists of running a database search in PEAKS with the merged proteome. Importantly, now that a merged hybrid proteome has been obtained from the HybridFinder function, it can be used to obtain potential PSPs whose quality is comparable with all other database search peptides while filtering all peptides at the same FDR (False Discovery Rate) cutoff which can be adjusted by the users in PEAKS. In the original workflow by Faridi et al. (2018), the database search peptides in both runs were filtered in PEAKS at a 1% FDR.

5. Perform a database search in PEAKS using the original raw MS file (while using the same settings as in the beginning) however, this time while using the merged hybrid proteome (.fasta) file generated with the HybridFinder function.

Step 4: Load checknetMHCpan inputs into R

Timing: 1 min

Prior to running checknetMHCpan, please ensure that netMHCpan (versions 4.0 or 4.1) is installed. checknetMHCpan is the last step of the hybrid finder workflow, the function uses the database search results from the second PEAKS analysis and provides the binding affinity results of all the peptides along with their categorizations.

6. Create an object for the location of the netMHCpan executable

```
> netmhcpan_dir <- file.path(''/usr/local/bin'')
```

7. Create an object (vector) for storing the HLA-I alleles that you would like to have binding affinity predictions for.

```
> alleles_Exp1 <- c(''HLA-A*02:01'', ''HLA-A*03:01'', ''HLA-B*07:02'')
```

8. Retrieve the HybridFinder output from the HybridFinder function results

```
> HF_output_Exp1 <- HybridFinder_results_Exp1[[1]]
```

9. Import the database search results (from step 3: Database search using hybrid fasta)

```
> rerun_db_search_Exp1 <- read.csv(file.path(folder_Exp1, ''second_run'', ''DB search psm.csv''), sep='','', head = TRUE, stringsAsFactors = FALSE)
```

Note: in case your computer's OS is "Windows" (netMHCpan is not compatible with Windows) the web version of netMHCpan (http://www.cbs.dtu.dk/services/NetMHCpan-4.1/instructions.php) would come in handy. In this case, we propose to use a separate function from this package instead (step2_wo_netmhcpan) which outputs a netMHCpan-ready input of sequences in .pep format.

Access the datasets included in the R package

The demonstration datasets from the HLA Ligand Atlas included in this package also include datasets for the checknetMHCpan/step2_wo_netMHCpan functions. After having run the HybridFinder function and stored the results in HyrbidFinder_results_Exp1, PEAKS was run using the merged hybrid proteome. Below is a way to retrieve the second PEAKS run dataset included in the package:

```
> data(package= ''RHybridFinder'', ''db_rerun_Human_Liver_AUTD17'')
```

Note: The merged proteome used for the second database search is based on the custom ALCcutoff being set to NULL (default parameter value).

⚠ CRITICAL: The merged proteome database would change between different samples, and if the customALCcutoff parameter is changed. The same merged hybrid proteome cannot be used for separate analyses.

Step 5: Run checknetMHCpan

Timing: ~ 1 min

The checknetMHCpan function embodies the second major step of the workflow. The categorizations of the hybrid peptides from the HybridFinder output are retrieved for matched peptides found in the second PEAKS database results. Then, peptide-MHC class I binding predictions for the entire database search results (for peptides between 9 and 12 amino acids) are computed using netMHCpan and are tidied in order to summarize the results.

10. Run checknetMHCpan using the code below (Please refer to Table 2 in order to know more about the inputs needed) and export the results in the same folder:

```
> checknetMHCpan_results_Exp1 <- checknetMHCpan(netmhcpan_directory = netmhcpan_dir, netmhcpan_alleles = alleles_Exp1, peptide_rerun = rerun_db_search_Exp1, HF_step1_output = HF_output_Exp1, export_files= TRUE, export_dir = folder_Exp1)
```

Table 2. checknetMHCpan function parameters

Parameter	Description	Default value
netmhcpan_directory	the directory where netMHCpan is installed (i.e., '/usr/bin' or '/usr/local/bin', depending on where you have it installed)	No defaults. Necessary input.
netmhcpan_alleles	a vector composed of the alleles the peptides will be tested against.	No defaults. Necessary input.
peptide_rerun	the database search results from the second peaks run	No defaults. Necessary input.
HF_step1_output	the data frame from the HybridFinder function of the containing the spliced peptide potential explanations as well as RT, m/z, ALC, Scan & Fraction	No defaults. Necessary input.
(Optional) export_files : boolean (True or False)	by default it is set to False, however, if set to True, then the following input is essential.	FALSE
(Optional) export_dir	file path to the directory where the output files should be stored. This parameter is necessary for the export.	NULL

Note: checknetMHCpan is compatible with the exports from both netMHCpan 4.0 & netMHCpan 4.1.

⚠ CRITICAL: if you use the datasets included in the package, please note that they are named differently so for instance the "peptide_rerun" parameter should be set to dataset loaded from the package db_rerun_Human_Liver_AUTD17.

After running the code above, a results list should be returned (Figure 4).

These results are also exportable with the export_checknetMHCpan_results function.

```
> export_checknetMHCpan_results(step2_RHF_results_Exp1 , export_dir = folder_Exp1)
```

Note: If you intend on using the web version of netMHCpan (especially useful for windows OS users) or another software for peptide binding affinity, the step2_wo_netMHCpan function does the same as checknetMHCpan but without running netMHCpan. The function should return a list (Figure 5) containing the updated database search results as well as a list of the peptides which can be used as input in the web version of netMHCpan.

EXPECTED OUTCOMES

HybridFinder

The HybridFinder function follows the same rationale as indicated in Faridi et al. (2018). After high-confidence *de novo* peptides are extracted, these are searched sequentially for an exact hit, followed by a search of pair fragments within one protein and then within two proteins (Figure 6). Finally, the sequences of all hybrid peptides are concatenated to create fake proteins, which are added at the bottom of the proteome database in order to constitute a merged hybrid proteome.

Typically, when the HybridFinder function is run, 3 messages are printed representing each major stage of the algorithm and finally 'Done!' is printed once the processing is finished. The function returns a list containing 3 items: the HybridFinder output (Figure 7) where the predicted splice type is displayed, a character vector containing only the list of hybrid candidates (Figure 8) and finally the

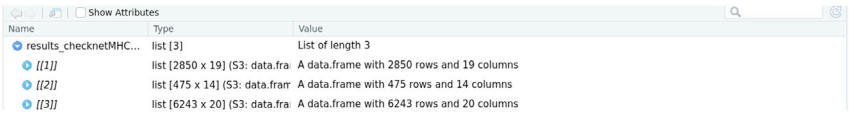


Figure 4. Screenshot of the checknetMHCpan results list

In the results list you will find 3 items: 1) a dataframe containing the netMHCpan results. 2) a dataframe containing the tidied netMHCpan results. 3) the database search results with the "Potential_spliceType" for the hybrid peptides retrieved from step1.

Figure 5. Screenshot of the step2_wo_netMHCpan results list

In the results list you will find 2 items: 1) a character vector containing the netMHCpan-ready input. 2) the database search results with the "Potential_spliceType" for the hybrid peptides retrieved from step1.

merged hybrid proteome (Figure 9) where the hybrid peptide candidates have been concatenated as fake proteins.

The results might differ if the customALCcutoff score parameter is changed. If the results are exported, these are stored in a folder as .csv files and the merged proteome database is saved as .fasta file. The peptide sequences predicted as spliced are considered as preliminary candidates. Performing the rest of the steps is essential in order to obtain the final list.

checknetMHCpan and step2_wo_netMHCpan

The checknetMHCpan & step2_wo_netMHCpan functions represent the last step in Faridi et al.'s (2018) workflow. After a database search is performed using the merged hybrid proteome in step 1, these two functions can be used. Both of these functions retrieve the potential splice type categorization established in step 1. However, with checknetMHCpan the user can directly obtain MHC-I binding affinity predictions computed for all peptides between 9 and 12 amino acids using netMHCpan (Jurtz et al., 2017; Reynisson et al., 2020).

The checknetMHCpan function returns two formats of the netMHCpan results and the updated database search results from the second run with the potential splice type. The first format of the

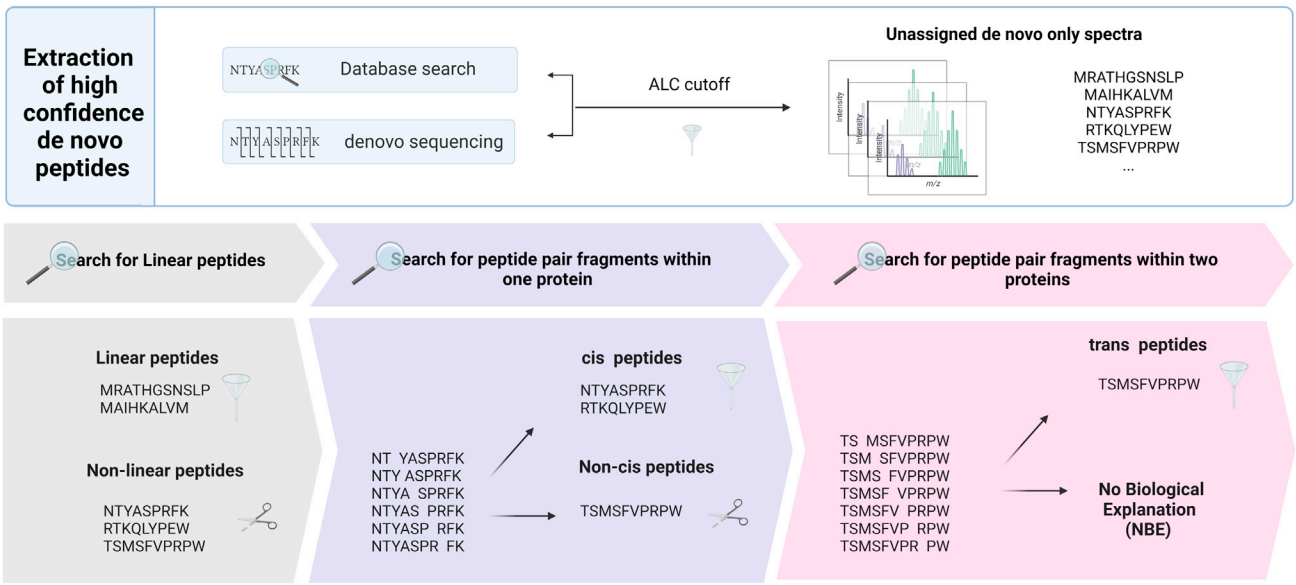


Figure 6. HybridFinder function

HybridFinder extracts high confidence *de novo* peptides by using a ALC cutoff based on the median ALC of common spectrum groups & sequence of peptides between the *de novo* and the database search. The ALC cutoff is used to filter unassigned *de novo* spectrum groups in order to obtain high confidence *de novo* spectra. All sequences are then searched in the proteome for the entire sequence, those that match are filtered and considered "Linear", the remainder of the peptide spectrum groups are "cut" in order to create peptide fragment combinations. These are then searched in the proteome for whether fragment combinations exist within a same protein, matches are considered as cis-spliced and further filtered. Finally, fragment combinations are created from those that didn't match in the previous step and are searched whether they exist in two proteins. If there is a match, these are considered as trans-spliced peptides. The remaining uncategorized spectrum groups are considered not to have a biological explanation (NBE) and are therefore discarded.

Fraction	Scan	m/z	RT	Peptide	Length	Potential_spliceType	ALC	proteome_database_used
1	F1:17511	575.7929	78.01	SYLEHLFEL	9	Linear	83	uniprot human-20379entries-Nov2019_validated.fasta
2	F2:10733	603.2902	54.40	LYTEKFEEF	9	Linear	93	uniprot human-20379entries-Nov2019_validated.fasta
1	F1:15697	575.7930	72.95	SYLEHLFEL	9	Linear	92	uniprot human-20379entries-Nov2019_validated.fasta
3	F3:7391	581.7984	42.96	LLYYASRNY	9	trans	81	uniprot human-20379entries-Nov2019_validated.fasta
2	F2:6862	560.7658	40.40	FSVHMVTHF	9	cis	91	uniprot human-20379entries-Nov2019_validated.fasta

Figure 7. Screenshot of the HybridFinder output dataframe

(5 rows), The *Fraction* column represents the LC-MS run, the *Scan* column is a number representing a unique index for the tandem mass spectra (F[Fraction#]:Scan#), *m/z* is the precursor mass-to-charge ratio, *RT* is the Retention Time (elution time) for the spectrum, *Peptide* corresponds to the peptide sequences. The *Length* column represents the number of amino acids for a given peptide, *ALC* (Average Local Confidence), is a score calculated in PEAKS as the total of the residue local confidence scores in the peptide divided by the peptide length. These columns are not provided by the HybridFinder function, they are columns found in any PEAKS *de novo* sequencing export. For more information, please visit the PEAKS user manual. The *Potential_spliceType* corresponds to the resulting categorization from the HybridFinder function. Finally, the *proteome_database_used* is the filename of the fasta proteome provided by the user (this column is mainly for helping the user keep track of the proteome used) in the HybridFinder function.

netMHCpan represents the results as they are (Figure 10). The second format is a tidied version of the netMHCpan results (Figure 11), where the rows are summarized into different columns, to allow quick analysis of the netMHCpan results (especially when more than one HLA-I allele is used); in these columns are summed the number of HLA-I alleles that a given peptide is a strong or weak binder to as well as the corresponding alleles. Finally, the database search results dataframe (from the second PEAKS run) updated with the potential splice type determined in the HybridFinder function for each peptide (Figure 12). Additionally, any sequence not identified in the HybridFinder output and solely attributed to the fake proteins created is removed. If exported, these are stored in a folder containing 2 .csv files and a .tsv (tab-separated values) corresponding to these different outputs.

The step2_wo_netMHCpan is the equivalent of checknetMHCpan with the exception of computing binding affinity. The function returns a netMHCpan-ready list of peptides (Figure 13), as well as the updated the database search results (Figure 14). If exported, the results are exported into a folder containing a .pep file and a .csv file.

V1
KAVNLLLSY
AKVNLLLSY
KLADLFRLY
NYGELFEKF
DYGELFEKF

Figure 8. Screenshot of the HybridFinder hybrid peptide candidates vector (5 rows)

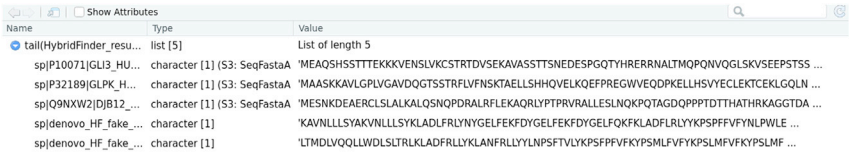


Figure 9. Screenshot of the bottom of the HybridFinder merged hybrid proteome (5 proteins)

After running checknetMHCpan or step2_wo_netMHCpan the final list of hybrid candidate peptides should be explored for further experimental validation (Figure 1).

LIMITATIONS

The presented package was developed and optimized for exports from PEAKS software. Therefore, results from other search engines or *de novo* sequencing softwares might not work while using this package. Limitations related to the workflow include the possible introduction of bias towards having results containing a higher proportion of Leucine residues. This is due to the workaround proposed by Faridi et al. (2018) which is also used in this package, entailing a switch of all Isoleucines to Leucines in the database search and the proteome since *de novo* sequencing does not differentiate between Isoleucine and leucine. As mentioned above, it is also important to emphasize that this protocol does not enable the direct identification of high-confidence PSPs that are genuinely spliced by the proteasome in vivo. However, this protocol enables the computational identification of putative PSPs, which should then be validated experimentally in a rigorous manner as shown in Figure 1.

TROUBLESHOOTING

Problem 1

While installing the .tar.gz file for the package, in case you run into the following error: "Error in install.packages : type == "both" cannot be used with 'repos = NULL'"

Potential solution

The solution would be to simply invoke the install.packages function while specifying where the package is located, setting the repository (repos) to NULL and setting the type as source (source package).

```
> install.packages("~/Downloads/RHybridFinder_0.1.0.tar", repos = NULL, Type="source")
```

Problem 2

While running HybridFinder, in case you run into the following error: "Error in prepare_input_for_HF(de novo_candidates, db_search): Please make sure you have the right input. N.B: The *de novo* results data frame should be the first input"

Pos	HLA	Peptide	Core	Of	Gp	Gl	Ip	Il	Icore	Identity	Score	Aff(nM)	%Rank	BindLevel	strongBinder	weakBinder	noneBinder	Potential_spliceType
1	HLA-A*03:01	RVFVVGVGMTK	RVFGVGMTK	0	3	2	0	0	RVFVVGVGMTK	PEPLIST	0.6371920	50.7	0.1899	Strong binder	HLA-A*03:01			Linear
1	HLA-A*03:01	PTTKTYFPHF	TTKTYFPHF	1	0	0	0	0	TTKTYFPHF	PEPLIST	0.0519360	28505.2	30.4401	Non binder			HLA-A*03:01	Linear
1	HLA-A*03:01	APVFRDYVF	APVFRDYVF	0	0	0	0	0	APVFRDYVF	PEPLIST	0.0396010	32575.1	41.8781	Non binder			HLA-A*03:01	Linear
1	HLA-A*03:01	YYFEGLKQTF	YYFGLKQTF	0	3	1	0	0	YYFEGLKQTF	PEPLIST	0.0531360	28137.5	29.5946	Non binder			HLA-A*03:01	Linear
1	HLA-A*03:01	EYLPLGGLAEF	YLLGGLAEF	1	2	1	0	0	YLPLGGLAEF	PEPLIST	0.0342630	34511.9	48.9703	Non binder			HLA-A*03:01	Linear

Figure 10. Screenshot of the checknetMHCpan netMHCpan results

(5 rows) *HLA/MHC* is the allele, *Peptide* is the amino acid sequence of the potential ligand, *Core* is the minimal 9 amino acid sequence core to enable HLA binding, *Of* is the starting position of the Core within the peptide, *Gp* and *Gl* are the position and the length of the deletions (respectively), if any. *Ip* and *Il* are the position and the length of the insertions (respectively), if any. *Icore* is the interaction core, *Identity* is PEPLIST (which indicates that peptides were used as input as opposed to proteins in fasta-format). *Score* is the raw prediction, *Aff(nM)* is the predicted IC50 value in nanoMolar units, *%Rank* is the percentile rank of the predicted affinity compared to a set of random natural ligands. *BindLevel* is designated by 3 qualifiers: Strong binder, Weak binder, None binder. *Potential_spliceType* is the categorization retrieved from the HybridFinder output on the potential splice type explanation of the peptide (i.e., linear, cis, trans).

Peptide	strongBinder	weakBinder	%Rank.HLA-C*16:01	%Rank.HLA-B*45:01	%Rank.HLA-B*35:03	%Rank.HLA-A*03:01	%Rank.HLA-C*04:01	%Rank.HLA-A*24:02	strongBinder_count	weakBinder_count	noneBinder_count	Potential_spliceType
AYTLLLHTW	HLA-A*24:02		15.1854	31.4452	31.0508	56.3778	8.3913	0.0893	1	0	5	Linear
VFPKAVSMPSF	HLA-A*24:02		15.0174	68.1566	2.1079	41.4260	2.8719	0.2910	1	0	5	Linear
SATLSFRLY		HLA-C*16:01	1.6557	19.6211	6.7228	6.1922	12.8342	18.3819	0	1	5	Linear
YQSRDYYNF	HLA-A*24:02	HLA-C*04:01	2.5753	10.3557	5.3714	34.7392	1.6679	0.1706	1	1	4	Linear
DYGELFEKF	HLA-A*24:02		30.4847	69.5422	20.7801	85.8340	3.4712	0.1806	1	0	5	cis

Figure 11. Screenshot of the checknetMHCpan tidied netMHCpan results

(5 rows) *Peptide* is the amino acid sequence of the potential ligand, the *strongBinder, weakBinder, noneBinder* (this column not shown in this figure) columns correspond to the alleles to which a given peptide is a strong/weak/none binder to, respectively. If more than one allele, these are separated by commas. For each peptide, there will be %Rank columns per allele (e.g., If 3 alleles were specified in the checknetMHCpan command, then each peptide will have 3%Rank columns). *strongBinder_count, weakBinder_count, noneBinder_count* represent the number of alleles to which a peptide is a strong/weak/none binder to. Lastly, the *Potential_spliceType* column is the categorization retrieved from the HybridFinder output on the potential splice type explanation of the peptide (i.e., linear, cis, trans).

Potential solution

Verify the *de novo* data frame has been correctly imported. Since the *de novo* results file is in .csv format, the separator should be a comma ",", stringsAsFactors should be set to FALSE and lastly the header should be set to TRUE. Please refer to step1: Loading inputs into R.

Verify that the HybridFinder parameters are properly typed. The *de novo* sequencing results data frame is indicated first and then the database search results. Alternatively, write the parameters and assigned them their appropriate objects (i.e., de novo_candidates = de novo_results_human_liver_Exp1). Please refer to step2: Run HybridFinder.

Problem 3

While running HybridFinder, in case you run into the following error: "Error in $<-.dataframe'('*tmp', "db_id", value = character (0)) : replacement has 0 rows, data has[…]"

Potential solution

Verify the database search data frame, make sure it has been correctly imported. Since the database search results file is in .csv format the separator should be a comma ",", stringsAsFactors should be set to FALSE and lastly the header should be set to TRUE. Please refer to step1: Loading inputs into R.

Problem 4

While running checknetMHCpan, if the following error is displayed: "Error in checknetMHCpan[…]: Please provide the proper input"

Potential solution

Verify that the *de novo* and database search data frames are not switched. Please refer to step 5: Run checknetMHCpan.

Problem 5

While running checknetMHCpan, if the following error is displayed: "Please check the input alleles: […]"

Peptide	X.10lgP	Mass	Length	ppm	m.z	Z	RT	Area	Fraction	Id	Scan	from.Chimera	Source.File	Accession	PTM	AScore	Found.By	Peptide_no_mods	Potential_spliceType
SYM(+15.99)GHFDLL	24.02	1097.485	9	1.1	549.7504	2	62.22	4902300	3	42227	F3:13106	No	171002_AM_BD-ZH17_Liver_W_10%_DDA_#3_400-650mz_msms6.mzML	Q9BWJ5	Oxidation (M)	M3:Oxidation (M):1000.00	PEAKS DB	SYMGHFDLL	Linear
VVYPWTQRF	29.34	1194.619	9	0.9	598.3171	2	61.91	5517900	2	24509	F2:13529	No	171002_AM_BD-ZH17_Liver_W_10%_DDA_#2_400-650mz_msms5.mzML	P68871			PEAKS DB	VVYPWTQRF	Linear
DYLEKYYKF	41.47	1267.612	9	0.6	634.8138	2	55.70	513300	2	23160	F2:11165	No	171002_AM_BD-ZH17_Liver_W_10%_DDA_#2_400-650mz_msms5.mzML	P40261			PEAKS DB	DYLEKYYKF	Linear
LLYYASNRY	37.29	1161.582	9	0.6	581.7985	2	43.23	238520	2	20603	F2:7207	No	171002_AM_BD-ZH17_Liver_W_10%_DDA_#2_400-650mz_msms5.mzML	denovo_HF_fake_protein2			PEAKS DB	LLYYASNRY	trans
KLADFRLLY	29.40	1137.655	9	0.4	569.8348	2	55.92	4620500	3	40877	F3:11153	No	171002_AM_BD-ZH17_Liver_W_10%_DDA_#3_400-650mz_msms6.mzML	denovo_HF_fake_protein1			PEAKS DB	KLADFRLLY	cis

Figure 12. Screenshot of the checknetMHCpan database search results updated with the Potential_spliceType column

(5 rows) *Peptide* is the amino acid sequence of the potential ligand, *X.log10P* represents the best -10logP identification score for the corresponding peptide. *Mass* represents the monoisotopic mass of the peptide, *Length* is the number of amino acid residues that constitute the given peptide, *ppm* is the precursor mass error, the *m.z* is the precursor mass-to-charge ratio, *Z* is the precursor charge, *RT* is the Retention Time (elution time) for the spectrum, *Area* represents the area underthe curve of the peptide feature found at the same m/z and retention time as the MS/MS scan, *Fraction* is the LC-MS run, *id* represents the precursor ID associated with the PSM, *Scan* is a number representing a unique index for tandem mass spectra (F[Fraction#]:Scan#), *from.Chimera* (this column is not shown in this figure) displays whether the identified peptide is from chimeric spectra, *Source.File* is the mzML/mzXML file used in the PEAKS analysis, *PTM* is the type of the post-translational modification, *Ascore* is the localization score assigned to modifications on the peptide, *Found.By* represents the analysis (in this case PEAKS DB). *Peptide_no_mods* represents the peptide sequence without modifications, *Potential_spliceType* is linear, cis or trans and is retrieved from the HybridFinder function.

V1
LYPDSFTVL
LDFPKPLLA
YYTPLTPHL
LYEPNFLFF
VAHVDDMPNAL

Figure 13. Screenshot of the step2_wo_netMHCpan netMHCpan-ready input (5 rows)

Potential solution

Ensure that the alleles are in the right format, or that the allele is written correctly (i.e., HLA-A03:01, HLA-A*03:01). Please refer to step 4: Load checknetMHCpan inputs into R.

Problem 6

While running checknetMHCpan, if the path to netMHCpan is not correct, the following error might appear: sh: 1: [/temporary directory/netMHCpan]: not found error in running command

Potential solution

The issue could either be that the directory does not contains the netMHCpan file or that the directory was not well written I.e('usr/bin/' vs. '/usr/bin' or '/usr/bin/, where the first example is wrong and the other two are correct). Please refer to step 4: Load checknetMHCpan inputs into R

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Etienne Caron etienne.caron@umontreal.ca.

Materials availability

This study did not generate new unique reagents.

Data and code availability

The package is available on CRAN and includes data (PEAKS analyses) from HLA Ligand Atlas (Marcu et al., 2021) deposited in PRIDE (Proteomics IDentification Database) PXD019643 (were analyzed in PEAKS and used in this protocol for demonstration purposes only).

Peptide	X.10lgP	Mass	Length	ppm	m.z	Z	RT	Area	Fraction	Id	Scan	from.Chimera	Source.File	Accession	PTM	AScore	Found.By	Peptide_no_mods	Potential_spliceType
SYM(+15.99)GHFDLL	24.02	1097.485	9	1.1	549.7504	2	62.22	4902300	3	42227	F3:13106	No	171002_AM_BD-ZH17_Liver_W_10%_DDA_#3_400-650mz_msms6.mzML	Q9BWJ5	Oxidation (M)	M3:Oxidation (M):1000.00	PEAKS DB	SYMGHFDLL	Linear
VVYPWTQRF	29.34	1194.619	9	0.9	598.3171	2	61.91	5517900	2	24509	F2:13529	No	171002_AM_BD-ZH17_Liver_W_10%_DDA_#2_400-650mz_msms5.mzML	P68871			PEAKS DB	VVYPWTQRF	Linear
DYLEKYYKF	41.47	1267.612	9	0.6	634.8138	2	55.70	513300	2	23160	F2:11165	No	171002_AM_BD-ZH17_Liver_W_10%_DDA_#2_400-650mz_msms5.mzML	P40261			PEAKS DB	DYLEKYYKF	Linear
LLYYASNRY	37.29	1161.582	9	0.6	581.7985	2	43.23	238520	2	20603	F2:7207	No	171002_AM_BD-ZH17_Liver_W_10%_DDA_#2_400-650mz_msms5.mzML	denovo_HF_fake_protein2			PEAKS DB	LLYYASNRY	trans
KLADFRLLY	29.40	1137.655	9	0.4	569.8348	2	55.92	4620500	3	40877	F3:11153	No	171002_AM_BD-ZH17_Liver_W_10%_DDA_#3_400-650mz_msms6.mzML	denovo_HF_fake_protein1			PEAKS DB	KLADFRLLY	cis

Figure 14. Screenshot of the checknetMHCpan database search results updated with the Potential_spliceType column
(5 rows) The dataframe contains the same columns as in Figure 13

ACKNOWLEDGMENTS

This work was supported by funding from the Funds de recherche du Québec - Santé (FRQS), the Cole Foundation, CHU Sainte-Justine and the Charles-Bruneau Foundations, Canada Foundation for Innovation, the National Sciences and Engineering Research Council (NSERC) (#RGPIN-2020-05232), and the Canadian Institutes of Health Research (CIHR) (#174924). K.A.K. is a recipient of IVADO's postdoctoral scholarship (#3879287150). C.L. is currently supported by a National Health and Medicine Research Council (NHMRC) of Australia CJ Martin Early Career Research Fellowship (1143366). A.W.P. is supported by a NHMRC Principal Research fellowship (1137739). P.F. is supported by a Victorian Cancer Agency (Australia) Mid-Career Fellowship.

AUTHOR CONTRIBUTIONS

R code, conceptualization, and authorship, F.S. and P.K.; validation and conceptualization, Q.M., D.J.H., K.A.K., and I.S.; conceptualization and authors of the original workflow, P.F., C.L., and A.W.P.; conceptualization and authorship, E.C.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

Berkers, C.R., Jong, A. de, Schuurman, K.G., Linnemann, C., Geenevasen, J.A.J., Schumacher, T.N.M., Rodenko, B., and Ovaa, H. (2015). Peptide splicing in the proteasome creates a novel type of antigen with an isopeptide linkage. J. Immunol. *195*, 4075–4084.

Charif, D., and Lobry, J.R. (2007). SeqinR 1.0-2: a contributed package to the R project for statistical computing devoted to biological sequences retrieval and analysis. In Structural Approaches to Sequence Evolution, Molecules, Networks, Populations, Ugo Bastolla, Markus Porto, H. Eduardo Roman, and Michele Vendruscolo, eds. (Springer Verlag)), pp. 207–232.

Dalet, A., Robbins, P.F., Stroobant, V., Vigneron, N., Li, Y.F., El-Gamil, M., Hanada, K., Yang, J.C., Rosenberg, S.A., and Eynde, B.J.V. den (2011). An antigenic peptide produced by reverse splicing and double asparagine deamidation. Proc. Natl. Acad. Sci. USA *108*, E323–E331.

Ebstein, F., Textoris-Taube, K., Keller, C., Golnik, R., Vigneron, N., Eynde, B.J.V. den, Schuler-Thurner, B., Schadendorf, D., Lorenz, F.K.M., Uckert, W., et al. (2016). Proteasomes generate spliced epitopes by two different mechanisms and as efficiently as non-spliced epitopes. Sci. Rep. *6*, 24032.

Faridi, P., Li, C., Ramarathinam, S.H., Vivian, J.P., Illing, P.T., Mifsud, N.A., Ayala, R., Song, J., Gearing, L.J., Hertzog, P.J., et al. (2018). A subset of HLA-I peptides are not genomically templated: Evidence for cis- and trans-spliced peptide ligands. Sci. Immunol. *3*, eaar3947.

Hanada, K., Yewdell, J.W., and Yang, J.C. (2004). Immune recognition of a human renal cancer antigen through post-translational protein splicing. Nature *427*, 252–256.

Jurtz, V., Paul, S., Andreatta, M., Marcatili, P., Peters, B., and Nielsen, M. (2017). NetMHCpan-4.0: improved peptide–MHC class I interaction predictions integrating eluted ligand and peptide binding affinity data. J. Immunol. *199*, 3360–3368.

Lichti, C.F. (2021). Identification of spliced peptides in pancreatic islets uncovers errors leading to false assignments. Proteomics *21*, e2000176.

Liepe, J., Mishto, M., Textoris-Taube, K., Janek, K., Keller, C., Henklein, P., Kloetzel, P.M., and Zaikin, A. (2010). The 20S proteasome splicing activity discovered by SpliceMet. Plos. Comput. Biol. *6*, e1000830.

Liepe, J., Marino, F., Sidney, J., Jeko, A., Bunting, D.E., Sette, A., Kloetzel, P.M., Stumpf, M.P.H., Heck, A.J.R., and Mishto, M. (2016). A large fraction of HLA class I ligands are proteasome-generated spliced peptides. Science *354*, 354–358.

Marcu, A., Bichmann, L., Kuchenbecker, L., Kowalewski, D.J., Freudenmann, L.K., Backert, L., Mühlenbruch, L., Szolek, A., Lübke, M., Wagner, P., et al. (2021). HLA Ligand Atlas: a benign reference of HLA-presented peptides to improve T-cell-based cancer immunotherapy. J. Immunother. Cancer *9*, e002071.

Michaux, A., Larrieu, P., Stroobant, V., Fonteneau, J.-F., Jotereau, F., Eynde, B.J.V. den, Moreau-Aubry, A., and Vigneron, N. (2014). A spliced antigenic peptide comprising a single spliced amino acid is produced in the proteasome by reverse splicing of a longer peptide fragment followed by trimming. J. Immunol. *192*, 1962–1971.

Mylonas, R., Beer, I., Iseli, C., Chong, C., Pak, H.-S., Gfeller, D., Coukos, G., Xenarios, I., Müller, M., and Bassani-Sternberg, M. (2018). Estimating the contribution of proteasomal spliced peptides to the HLA-I Ligandome. Mol. Cell. Proteomics *17*, 2347–2357.

Neefjes, J., Jongsma, M.L.M., Paul, P., and Bakke, O. (2011). Towards a systems understanding of MHC class I and MHC class II antigen presentation. Nat. Rev. Immunol. *11*, 823–836.

Reynisson, B., Alvarez, B., Paul, S., Peters, B., and Nielsen, M. (2020). NetMHCpan-4.1 and NetMHCIIpan-4.0: improved predictions of MHC antigen presentation by concurrent motif deconvolution and integration of MS MHC eluted ligand data. Nucleic Acids Res. *48*, W449–W454.

Rolfs, Z., Müller, M., Shortreed, M.R., Smith, L.M., and Bassani-Sternberg, M. (2019). Comment on "A subset of HLA-I peptides are not genomically templated: Evidence for cis- and trans-spliced peptide ligands. Sci. Immunol. *4*, eaaw1622.

Specht, G., Roetschke, H.P., Mansurkhodzhaev, A., Henklein, P., Textoris-Taube, K., Urlaub, H., Mishto, M., and Liepe, J. (2020). Large database for the analysis and prediction of spliced and non-spliced peptide generation by proteasomes. Sci. Data *7*, 146.

Vigneron, N., Stroobant, V., Chapiro, J., Ooms, A., Degiovanni, G., Morel, S., Bruggen, P. van der, Boon, T., and Eynde, B.J.V. den (2004). An antigenic peptide produced by peptide splicing in the proteasome. Science *304*, 587–590.

Wilhelm, M., Zolg, D.P., Graber, M., Gessulat, S., Schmidt, T., Schnatbaum, K., Schwencke-Westphal, C., Seifert, P., Krätzig, N. de A., Zerweck, J., et al. (2021). Deep learning boosts sensitivity of mass spectrometry-based immunopeptidomics. Nat. Commun. *12*, 3346.