# MHC Scripts

**Automatic Classification of HLA Peptides**

This set of scripts classifies a list of peptides with know HLA type and visualizes the results as a heatmap.

**1. Predict Affinities**

**peptides\_matrix\_netMHC.txt**

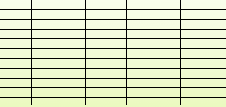
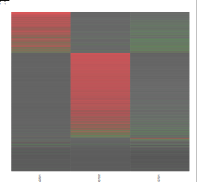
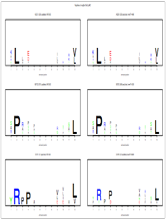
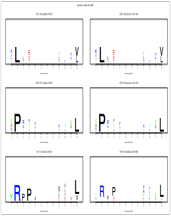
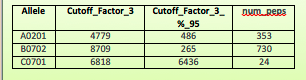
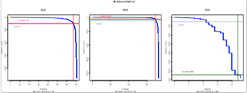
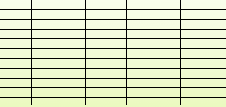
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sample** | **Alleles** | **Peptide** | **Accession** | **A0201** | **B0702** | **C0701** |
| peptides | A0201;B0702;C0701 | KLQNLMIFL | A0AVT1 | 50 | 3002 | 85723 |
| peptides | A0201;B0702;C0701 | SPRSSTVSL | A0JNW5 | 45634 | 32 | 7688 |
| peptides | A0201;B0702;C0701 | RPLQGTTL | A0PJW6 | 500 | 10345 | 6003 |
| ... | ... | ... | ... | ... | ... | ... |
| WB Threshold | WB Threshold | WB Threshold | WB Threshold | 500 | 500 | 500 |
| SB Threshold | SB Threshold | SB Threshold | SB Threshold | 50 | 50 | 50 |

peptides.txt

|  |  |
| --- | --- |
| **Peptide** | **Accession** |
| KLQNLmIFL | A0AVT1 |
| SPRSSTVSL | A0JNW5 |
| RPLQGT[+79]TL | A0PJW6 |
| ... | ... |

alleles

A02,B07,C07



**factor.txt**

new\_cutoffs.txt

distribution.pdf

motifs\_9er.pdf

motifs\_all.pdf

heatmap.pdf

heatmap.txt

**Cutoff Estimation**

**Heatmap**

**Motifs**

**2. Classification and Visualization**

**1. Predict Affinities**

From a peptide list from a sample with known alleles, we obtain a new table including the **binding affinity predicted by netMHC for each of these alleles**. This netMHC matrix output will be used in the next step to automatically classify the peptides.

It also includes the original affinity thresholds used for weak and strong binders (two last rows).

The length of the peptides should be between **8 and 12** amino acids, otherwise they will be filtered out. **Duplicated** peptide sequences will be reported only once and the protein accession IDs will be combined using a semicolon.

Peptide **modifications** (ex. oxidation) are usually ignored and their naked sequences are used in the output.

In case no alleles selection is provided, a prediction will be done for all netMHC available alleles. See note below for adding new alleles to netMHC.

|  |
| --- |
| $ head peptides\_test.txt  Peptide Accession  KLQNLNIFL A0AVT1  SPRSSTVSL A0JNW5  RPLQGTTL A0PJW6  ...  $ **matrix\_netMHC\_MHCI.py peptides\_test.txt A02,B07,C07**  0. Parameters, input files and format handling:  - Input file: peptides\_test.txt  - Output file: peptides\_test\_matrix\_netMHC.txt  - Fixing input file to ensure linux end of lines  - Executing: perl -pi -e 's/\r\n/\n/g' peptides\_test.txt  - Executing: perl -pi -e 's/\r/\n/g' peptides\_test.txt  1. Match the given alleles to the netMHC available ones ...  - Alleles to predict selected (3): A0201, B0702, C0701  ...DONE!  2. Parsing the input table ...  - Peptide length allowed: [8 - 12]  - Read 26 input lines, including 0 duplicates  - Found 26 unique peptides: ILDFQPPEL, SPRAPFYRPL, LILMGNALIL... TMADQIVTV  ...DONE!  3. Processing peptides with netMHC...  ...DONE!  4. Write output file...  - Written 26 peptide rows to peptides\_test\_matrix\_netMHC.txt  - Done, added 2 threshold rows to peptides\_test\_matrix\_netMHC.txt  ...DONE!  $ cat peptides\_test\_matrix\_netMHC.txt  Sample Alleles Peptide Accession A0201 B0702 C0701  peptides\_test A0201;B0702;C0701 ILDFQPPEL A4D174 32 19503 6355  peptides\_test A0201;B0702;C0701 SPRAPFYRPL A6NED2 27294 9 22048  ...  WB Threshold WB Threshold WB Threshold WB Threshold 500 500 500  SB Threshold SB Threshold SB Threshold SB Threshold 50 50 50 |
| ***★NOTE: “How to add more alleles to netMHC”***  *The standalone version of netMHC installed in Brutus does not work straightforward due to the presence of not allowed symbols in the names of the folders that contain the prediction information for the available alleles.*  * Directory Structure (simplified):*   * /IMSB/ra/lespona/html/bin/mhc\_i/   + - method/       * netMHC-3.4/ * **netMHC-3.4\_fix** (custom executable to produce a tsv table) * etc/ * **NN.list** * net/ (folder with the available allele subfolders)   + - **to\_fix**/ (folder with yet not fixed alleles) * **rename\_synlist.pl** * Allele/   + - * bl50/         + SYN/ * **synlist**   + - * sparse/         + SYN/ * **synlist**   * To see all the netMHC options execute:* netMHC-3.4\_fix –h  * To get the list of currently available alleles execute:* netMHC-3.4\_fix –A  * To add an allele, first check that the subfolder(s) exists in the to\_fix folder. Then follow this steps:*   1. *Move up allele subfolders from* ***to\_fix*** *to* ***net*** *directory and* ***rename*** *them to a suitable name (no “-“, “/”, etc, “\_” is allowed, ex. “*HLA-A02/02\_8mer*”* “A0202\_8mer*”)* 2. *Edit* ***NN.list*** *file and add the exact new allele folder name* 3. *Edit the script* ***rename\_synlist.pl*** *so that it performs the replacement of the new allele name into all the data files. Then run the script from the net folder:*   $ cd /IMSB/ra/lespona/html/bin/mhc\_i/method/netMHC-3.4/etc/net  $ ./rename\_synlist.pl   1. *Edit the* ***matrix\_netMHC\_MHCI.py*** *script and add the new added allele name to the list in the variable ALLELES\_LIST\_AVAIL at the beginning of the script*   * To test the new allele addition execute:*  $ cd $SONAS/bin/mhc\_i/method/netMHC-3.4/test  *# show command examples*  $ cat commands\_test.txt  ../netMHC -a A0201 test.fsa > test\_fsa\_A0201.out  ../netMHC -p test.pep -a A0201 > test\_pep\_A0201.out  *# test for new allele*  $ ../netMHC -a XXXXX test.fsa > test\_fsa\_XXXX.out  $ **../netMHC -p test.pep -a XXXXX > test\_pep\_XXXX.out**  *# check outputs*  $ cat test\_pep\_XXXX.out  pos peptide score affinity(nM) Bind Level Protein Name Allele Method SB Threshold WB Threshold  0 AAAWYLWEV 0.823 6 SB Peptides XXXXX ANN 50 500  1 AAGLQDCTM 0.071 23033 NA Peptides XXXXX ANN 50 500  […] |

**2. Classification and Visualization**

The R script ‘**matrix\_netMHC\_analysis\_batch.R**’ classifies the peptides, generates the heatmap and estimates new cutoffs for netMHC performs this step.

It needs to call functions from two other R scripts: ‘**allele\_distributions.R**’ and ‘**sequence\_motiv.R**’. These scripts should be located in the directory defined by the environment variable ‘R\_MHCI\_SCRIPTS’. To define this variable execute:

|  |
| --- |
| $ export R\_MHCI\_SCRIPTS=$SONAS/bin/MHC\_scripts |

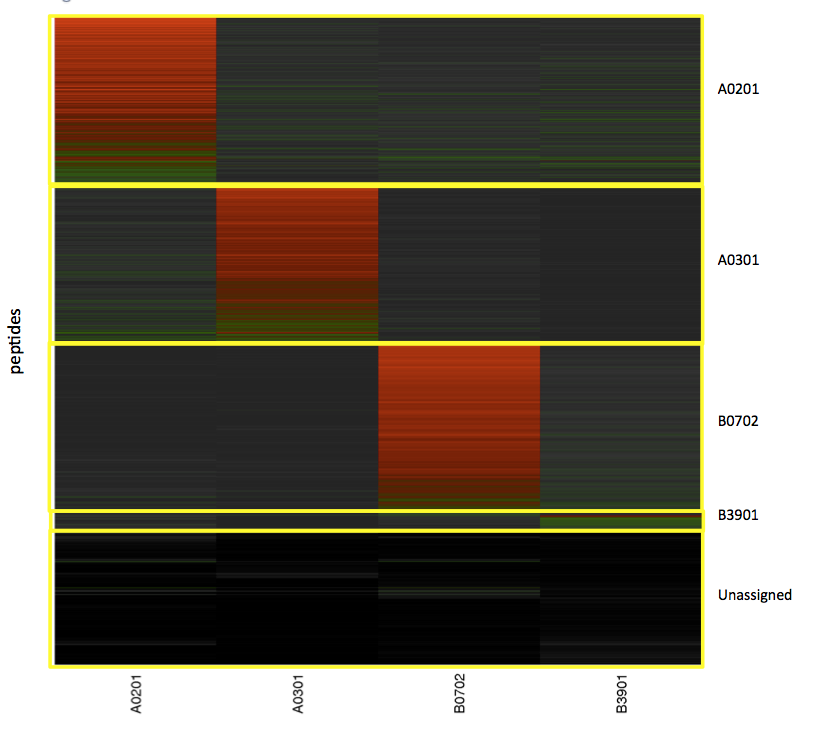
The script performs the following analysis steps:

* First a value called **‘factor’** is calculated as the fold change between the best and the second best affinity score. The higher this factor, the more confident the assignment of a peptide to an allele will be. The minimum to be considered as a valid assignment is 3, independently of the affinity value. This value can be changed on the R code, variable ‘FACTOR\_TH\_MIN’. The column factor is added to the original table and written to the output table.

This way even peptides with a predicted affinity not passing the netMHC weak binder threshold (500 nM) can be confidently assigned to an allele when the scores for the other HLA types are significantly worse. Also, peptides with a high score for an allele won’t be classified as good binders if the score for one of the other possible HLA types has a similar value.

* Afterwards the new **cutoffs for netMHC are estimated** for each allele by taking the 95 percentile of the assigned peptides using the defined factor.
* Optionally (set variable ‘DO\_MOTIFS’) it displays the **motifs** of the peptides assigned to each allele using the new and the standard weak binder netMHC cutoff. Two PDF files are generated, one containing all lengths and other just for the 9-mers.
* The last step is to visualize the classification results on a **heatmap** (PDF file). The heatmap values are also stored in a tab-separated file.

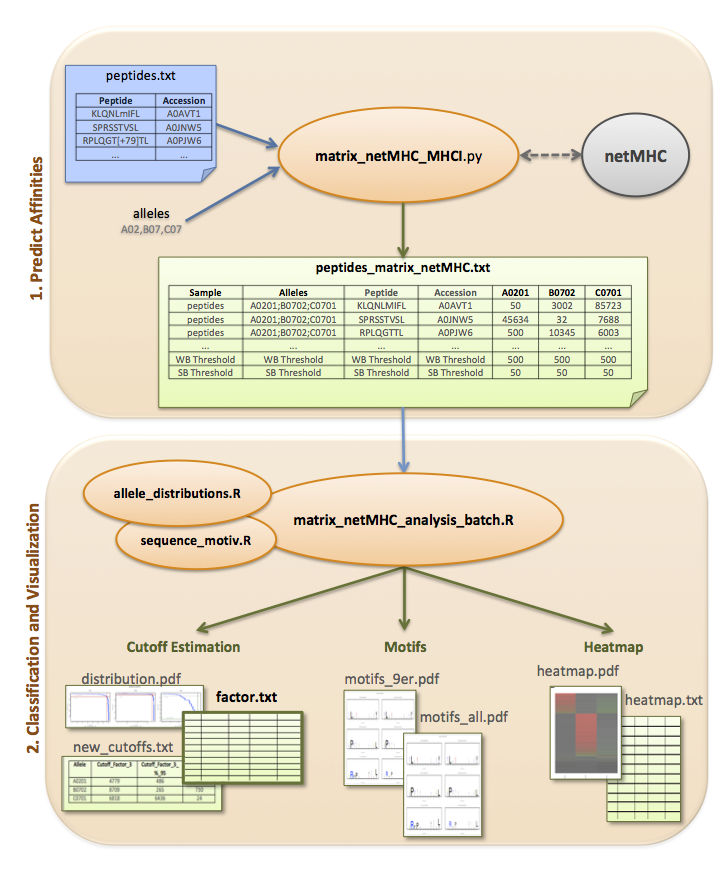
The alleles with **no peptide assigned** will be removed from the outputs (except heatmap which will show everything).

histograms_alleles_factor_3_95.pdf

It uses Rscript to run in batch mode (not interactively), like a normal command line program in any machine with R installed. Command is described below.

|  |
| --- |
| **$ Rscript $R\_MHCI\_SCRIPTS/matrix\_netMHC\_analysis\_batch.R "input\_file=\"peptides\_test\_matrix\_netMHC.txt\"" "output\_file=\"peptides\_test\_matrix\_netMHC\_factor.tsv\""**  [1] " - Parameters: \* input\_file: ' peptides\_test\_matrix\_netMHC.tsv '"  [1] " \* output\_file: ' peptides\_test\_matrix\_netMHC\_factor.tsv '"  [1] " - Weak Binder Threshold: 500"  [1] " - Calculating minimum and factors..."  [1] " \* Writing factor to output file: 'peptides\_test\_matrix\_netMHC\_factor.tsv'"  [1] " - Calculating new affinity cutoffs (95%)..."  [1] " \* Minimum factor: 3"  [1] " \* Printing distributions to PDF file:'peptides\_test\_allele\_distributions.pdf'"  Allele Cutoff\_Factor\_3 **Cutoff\_Factor\_3\_%\_95** num\_peps  A0201 A0201 4779 **486** 353  B0702 B0702 8709 **265** 730  C0701 C0701 6818 **6436** 24  [1] " \* Output: factor\_df\_file: 'peptides\_test\_new\_netMHC\_cutoff\_3.tsv'"  [1] " - Displaying motifs..."  [1] " \* Printing all length motifs to PDF file: 'peptides\_test\_motifs\_all.pdf'"  [1] " \* Printing length 9 motifs to PDF file: 'peptides\_test \_motifs\_9ers.pdf'"  [1] " - Creating heatmap..."  [1] " \* Writing heatmap table to file: 'peptides\_test\_heatmap\_table.tsv'"  [1] " \* Printing heatmap to PDF file: 'peptides\_test\_heatmap\_Rplot.pdf'"  [1] " \* Classification of the peptides: A0201(353), B0702(730), C0701(24),  UNCLASSIFIED(268)” |

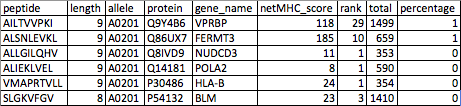
**netMHC Peptide Ranking**

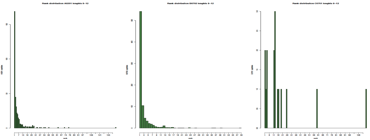
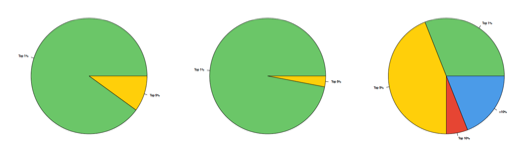
This set of scripts ranks peptides inside each protein based on their affinity to the previously assigned allele (classification using factor). This rank is displayed as distribution histograms and pie charts.

database.fasta

**1. Peptide Ranking**

**peptides\_matrix\_netMHC\_factor.txt**





**Rank Pie Charts**

*per length [8-12] & all*

**2. Ranking Visualization**

**Distribution Histograms**

**1. Peptide Ranking**

The python script ‘**peptides\_rank\_netMHC\_MHCI.py**’ ranks the peptides assigned to each allele comparing their affinity to all the peptides from the same protein. The **length** of these peptides is restricted from 8 to 12 aminoacids.

The protein information is provided by the **database** in fasta format selected as second input parameter. The previous accession information is ignored; the peptide-protein assignment is done from sequence matching.

It considers the **classification** performed by the previous scripts and filters out the peptides classified with too low confidence defined by the ‘factor’ value (minimum cutoff defined by variable ‘**MIN\_FACTOR’** in the code).

|  |
| --- |
| **$ peptides\_rank\_netMHC\_MHCI.py peptides\_matrix\_netMHC\_factor.txt database.fasta**  0. Parameters, input files and format handling:  - Factor matrix file: peptides\_matrix\_netMHC\_factor.txt  - Database fasta file: database.fasta  - Output file: peptides\_matrix\_netMHC\_factor\_rank.txt  1. Read the input list...  - Read 1166 input lines, selected 1069 rows (factor minimum 3):  \* A0201: 314 peptides, scores [2...5299]  \* B0702: 742 peptides, scores [6...11104]  \* C0701: 13 peptides, scores [9...7406]  2. Reading the database...  - Read 20204 proteins, (Q86T20-C6orf1, P78411-IRX5,... Q16534-HLF)  - Ignored 62 ambiguous proteins  3. Matching peptides to the proteins...  - Processing fasta for each allele:  \* A0201: 311/314 peptides matched (99%), 363 proteins  \* B0702: 742/742 peptides matched (100%), 738 proteins  \* C0701: 13/13 peptides matched (100%), 16 proteins  - Non matching peptides: 3  \* KVAAALTKA not matched!  \* FIDPKKQPAT not matched!  \* SLDKFLANV not matched!  - Selected 1061 proteins from initial fasta  \* Removed 19143 proteins from initial list, 1061 proteins left  \* Written 1061 selected proteins to database\_rank\_selection.fasta for  optimizing next analysis  4. Processing proteins with netMHC and writing output table...  - Processing allele A0201...  - Processing allele B0702...  - Processing allele C0701...  - Written 1235 peptide rows to peptides\_matrix\_netMHC\_factor\_rank.txt  ...DONE!  **$ cat peptides\_matrix\_netMHC\_factor\_rank.txt**  peptide length allele protein gene\_name netMHC\_score rank total percentage  AILTVVPKI 9 A0201 Q9Y4B6 VPRBP 118 29 1499 1  SLSDTVEKL 9 A0201 Q14203 DCTN1 43 6 1270 0  […]  ALLDKLYAL 9 A0201 Q9NV31 IMP3 4 1 176 0  NLLTHIENV 9 A0201 A8MW92 PHF20L1 13 2 1009 0 |

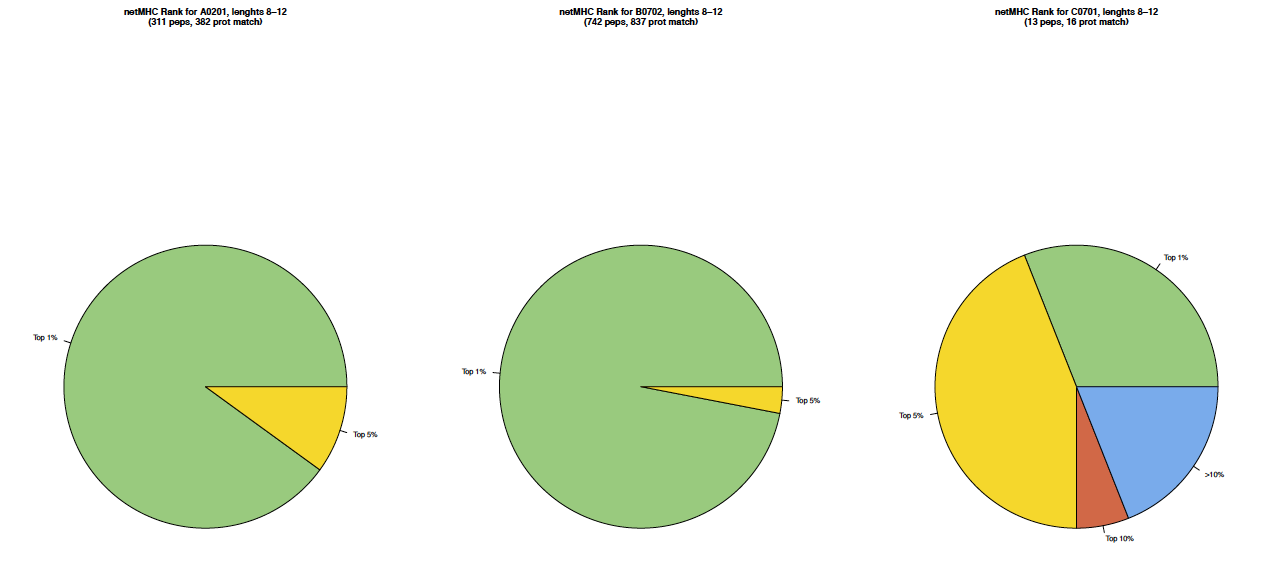
**2. Ranking Visualization**

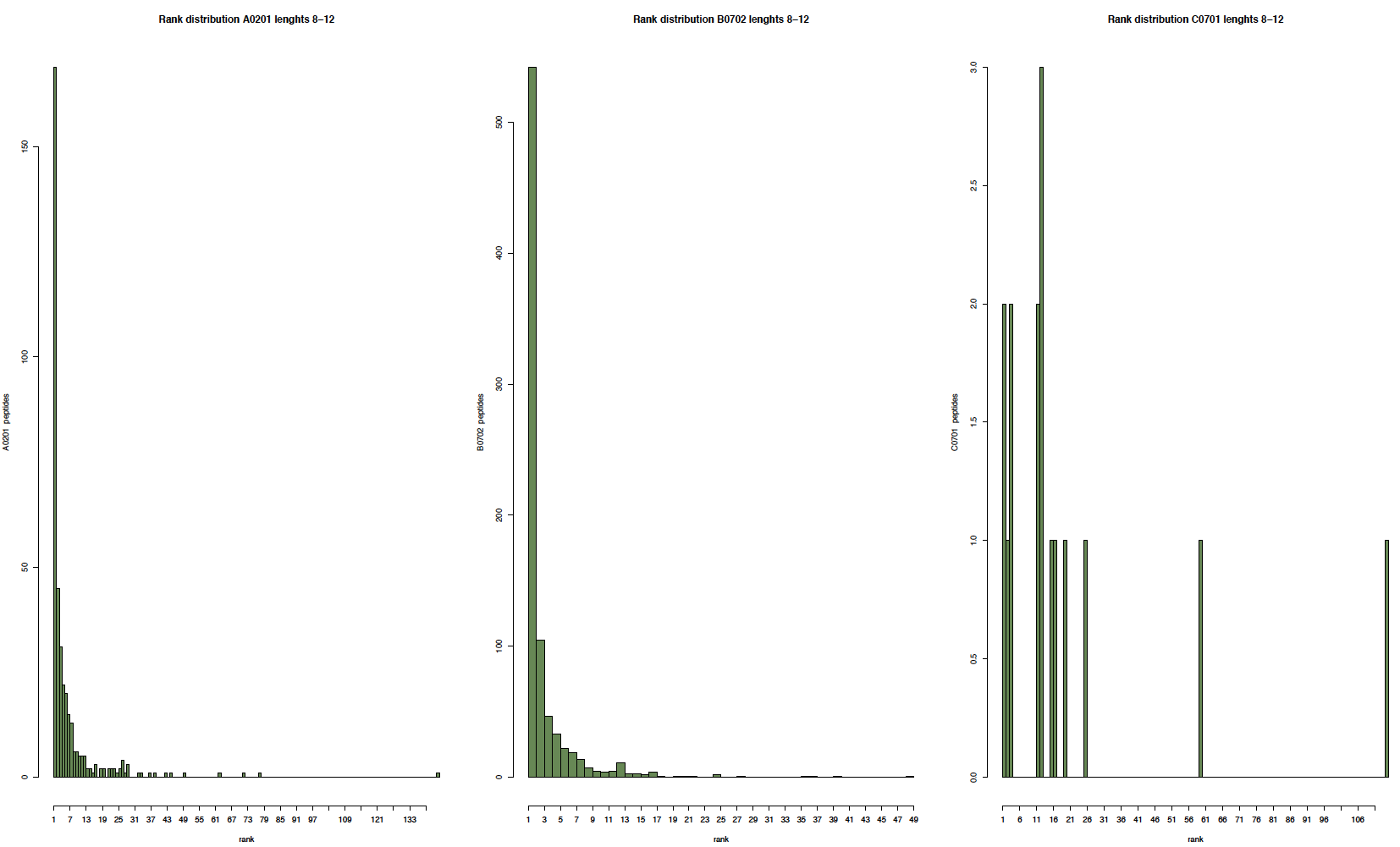
The R script ‘**peptides\_rank\_netMHC\_batch.R**’ generates the pie chart plots and distribution histograms. It uses Rscript to run in batch mode (not interactively), like a normal command line program in any machine with R installed.

The pie charts show the percentages of peptides belonging to the top 1, 5, 10 percentages from all the peptides classified as belonging to each allele.

The histograms show the distribution of the peptides over the different rank values.

Both graphs are generated once for all lengths and then separately for each peptide length in the range 8 to 12 aminoacids.





|  |
| --- |
| **$ Rscript $R\_MHCI\_SCRIPTS/peptides\_rank\_netMHC\_batch.R "input\_file=\"peptides\_matrix\_netMHC\_factor\_rank.txt\""**  [1] "input\_file=\"peptides\_matrix\_netMHC\_factor\_rank.txt\""  [1] 1  [1] " - Parameters: \* input\_file: 'peptides\_matrix\_netMHC\_factor\_rank.txt'"  [1] " \* output\_files: './peptides\_matrix\_netMHC\_factor\_rank\*'"  [1] " - Reading the input file..."  peptide length allele netMHC\_score rank total percentage  1 AILTVVPKI 9 A0201 118 29 1499 1  2 SLSDTVEKL 9 A0201 43 6 1270 0  3 ALLSRLEQI 9 A0201 65 2 297 0  4 LLLPGELAKHAV 12 A0201 19 1 115 0  5 ALLGDLTKA 9 A0201 71 27 890 3  6 QLNEKVAQL 9 A0201 131 19 495 3  [1] " \* Found 3 alleles in input table: A0201,B0702,C0701"  [1] " - Plotting the pie chart for all lengths (8-12)..."  null device  1  [1] " \* Writing all lengths pie chart to PDF file: './peptides\_matrix\_netMHC\_factor\_rank\_all\_lengths\_pie.pdf'"  [1] " - Plotting the pie chart for each length (8-12)..."  [1] " \* Writing 8-mers pie chart to PDF file: './peptides\_matrix\_netMHC\_factor\_rank\_length\_8\_pie.pdf'"  [1] " \* Writing 9-mers pie chart to PDF file: './peptides\_matrix\_netMHC\_factor\_rank\_length\_9\_pie.pdf'"  [1] " \* Writing 10-mers pie chart to PDF file: './peptides\_matrix\_netMHC\_factor\_rank\_length\_10\_pie.pdf'"  [1] " \* Writing 11-mers pie chart to PDF file: './peptides\_matrix\_netMHC\_factor\_rank\_length\_11\_pie.pdf'"  [1] " \* Writing 12-mers pie chart to PDF file: './peptides\_matrix\_netMHC\_factor\_rank\_length\_12\_pie.pdf'"  [1] " - Plotting the rank distribution histograms for all lengths (8-12)..."  null device  1  [1] " \* Writing all lengths distribution histogram to PDF file: './peptides\_matrix\_netMHC\_factor\_rank\_all\_lengths\_hist.pdf'"  [1] " - Plotting the distribution histogram for each length (8-12)..."  [1] " \* Writing 8-mers distribution histogram to PDF file: './peptides\_matrix\_netMHC\_factor\_rank\_length\_8\_rank\_hist.pdf'"  [1] " \* Writing 9-mers distribution histogram to PDF file: './peptides\_matrix\_netMHC\_factor\_rank\_length\_9\_rank\_hist.pdf'"  [1] " \* Writing 10-mers distribution histogram to PDF file: './peptides\_matrix\_netMHC\_factor\_rank\_length\_10\_rank\_hist.pdf'"  [1] " \* Writing 11-mers distribution histogram to PDF file: './peptides\_matrix\_netMHC\_factor\_rank\_length\_11\_rank\_hist.pdf'"  [1] " \* Writing 12-mers distribution histogram to PDF file: './peptides\_matrix\_netMHC\_factor\_rank\_length\_12\_rank\_hist.pdf'" |