NeoDisc version 1.7.0

Florian Huber, Michal Bassani-Sternberg

March 18, 2025

Contents

1 License **3**

2 NeoDisc installation **4**
2.1 Requirements . . . 4
2.2 Extract NeoDisc . . . 4
2.3 Download Resources . . . 4
2.4 Download proprietary software . . . 5
2.5 Install proprietary software in NeoDisc container . . . 6

3 Prepare Input data **8**
3.1 Format NGS reads filenames (fastq) . . . 8
3.2 Immunopeptidomics MS files . . . 8
3.3 Configuration file . . . 9
3.3.1 Configuration files variables . . . 9

4 Running NeoDisc **14**
4.1 NeoDisc Modules . . . 14
4.2 NeoDisc running Modes . . . 15
4.3 fastq Mode . . . 15
4.3.1 Requirements . . . 15
4.3.2 Running NeoDisc default fastq mode . . . 15
4.3.3 Running NeoDisc sensitive fastq mode . . . 17
4.4 panel mode . . . 17
4.4.1 Requirements . . . 17
4.4.2 Running NeoDisc in panel Mode . . . 17
4.5 Combining Individual Analysis (Clonality Analysis) . . . 18
4.5.1 Requirements . . . 18
4.5.2 Combining Individual Analysis . . . 18
4.6 Test cases . . . 20
4.6.1 Gene Panel . . . 20
4.6.2 fastq . . . 21

5 NeoDisc Results **23**
5.1 Prioritization.xlsx . . . 23
5.1.1 MS_peptides (Optional) . . . 23
5.1.2 Expressed_TAAs_peptides (Optional) . . . 25
5.1.3 VIRUS_RNA_HC_peptides (Optional) . . . 26
5.1.4 Class_I/Class_II . . . 27
5.1.5 Class_I_selection . . . 29
5.1.6 Class_II_selection . . . 29
5.1.7 long_peptides . . . 29
5.1.8 long_peptides_selection . . . 31
5.2 Individual Analysis Results . . . 31

5.3 Combined analysis (clonality analysis) Results 31

6 FAQ **32**

7 Softwares and databases used in NeoDisc **34**

Please note that this PDF contains links to attached files. To open the files, simply click the pink links. Examples of such files available at Configuration files variables. If this function is not supported by your PDF viewer, please use Adobe reader.

1 License

FOR ACADEMIC NON-COMMERCIAL RESEARCH PURPOSES ONLY. See the License

FOR-PROFIT USERS: If you plan to use NeoDisc or any data provided with the script in any for-profit application, you are required to obtain a separate license. To do so, please contact michal.bassani@chuv.ch.

2 NeoDisc installation

2.1 Requirements

1. **A linux computer with sudo rights** - This is only required for the initial installation of NeoDisc. Once the installation is finished, NeoDisc should run on any linux computer which satisfies the next two points.
2. Install **Singularity** - required both on the installation and running computers (see: https://docs.sylabs.io/guides/latest/user-guide/quick_start.html#quick-installation-steps)
3. Make sure that **users namespaces** are enabled on the running computer (see: https://docs.sylabs.io/guides/latest/admin-guide/user_namespace.html)
4. **Minimum 4 CPUs and 64G of RAM** on the running computer

2.2 Extract NeoDisc

This Manual and the NeoDisc_v1.7.0_bf1.tar.gz tarball can be downloaded from https://neodisc.unil.ch, upon registration.
Extract NeoDisc tarball in the installation computer / directory where you have **sudo rights**. It will be possible to move NeoDisc to another computer / directory after the initial installation. Please be patient, this step takes about 20-30 minutes.

```
tar xzvf NeoDisc_v1.7.0_bf1.tar.gz
```

This will create a directory **'NeoDisc_v1.7.0'** which will contain the following script and sub-directories:

- **install.sh**: A script for the installation of NeoDisc - see below
- **PROPRIETARY_SOFTWARE**: A directory where the user has to save the proprietary software required for NeoDisc installation - see Install proprietary software in NeoDisc container.
- **lib**: A directory required for the installation of NeoDisc

2.3 Download Resources

Download the resource data required by NeoDisc. The Resources_1.7.0.tar.gz tarball can be downloaded from https://neodisc.unil.ch and does not require registration. Extract the content of the Resources_1.7.0.tar.gz tarball directly within the **'NeoDisc_v1.7.0'** folder:

```
tar xzvf Resources_1.7.0.tar.gz -C path/to/NeoDisc_v1.7.0/
```

This will create a **'Resources'** sub-directory within the **'NeoDisc_v1.7.0'** directory. This directory is used in each NeoDisc analysis. Make sure to preserve timestamps of Resources files if you move this directory.

2.4 Download proprietary software

NeoDisc relies on proprietary software which can not be shared. Consequently, academic users should download their own copies. Save all the software and data in the *PROPRIETARY_SOFTWARE* subdirectory. All software tarballs should be saved in the same directory, a script will install them automatically in the container (see Install proprietary software in NeoDisc container). Note that the version of the downloaded software has to be identical to avoid any errors in the installation process or when runing NeoDisc. Also, make sure that the filenames are identical to those listed below. All the links to the software provided below are for academic research, non-commercial, or educational purposes only.

1. **HLA-HD - 1.7.0**

   Follow the registration process at: https://w3.genome.med.kyoto-u.ac.jp/HLA-HD/download-request/

2. **Varscan - 2.4.6**

   Please read first the description/license at: https://github.com/dkoboldt/varscan/blob/master/VarScan.v2.4.6.description.txt

   Download the jar directly from GitHub:

```
wget https://github.com/dkoboldt/varscan/raw/master/VarScan.v2.4.6.jar
```

3. **MSFragger - 3.8**

   Follow the registration process at: http://msfragger-upgrader.nesvilab.org/upgrader/

4. **IonQuant - 1.9.8**

   Follow the registration process at: https://msfragger.arsci.com/ionquant/

5. **netchop - 3.1d**

   Follow the registration process at: https://services.healthtech.dtu.dk/cgi-bin/sw_request?software=netchop&version=3.1&packageversion=3.1d&platform=Linux

6. **netMHCstabpan - 1.0b**

   Follow the registration process at: https://services.healthtech.dtu.dk/cgi-bin/sw_request?software=netMHCstabpan&version=1.0&packageversion=1.0b&platform=Linux

7. **netMHCstabpan - 1.0 data**

   Download file and rename to 'netMHCstabpan-1.0.data.tar.gz'

```
wget https://services.healthtech.dtu.dk/services/NetMHCstabpan-1.0/data.tar.gz -O netMHCstabpan-1.0.data.tar.gz
```

8. **netMHCpan - 4.1b**

   Follow registration process at: https://services.healthtech.dtu.dk/cgi-bin/sw_request?software=netMHCpan&version=4.1&packageversion=4.1b&platform=Linux

9. **netMHCpan - 2.8a**

   Follow registration process at: https://services.healthtech.dtu.dk/cgi-bin/sw_request?software=netMHCpan&version=2.8&packageversion=2.8a&platform=Linux

10. **netMHCpan - 4.1 data**

    Download file and rename to 'netMHCpan-4.1b.data.tar.gz'

```
wget https://services.healthtech.dtu.dk/services/NetMHCpan-4.1/data.tar.gz -O
netMHCpan-4.1b.data.tar.gz
```

11. **netMHCpan - 2.8 data**

    Download file and rename to 'netMHC2.8.data.tar.gz':

```
wget https://services.healthtech.dtu.dk/services/NetMHCpan-2.8/data.tar.gz -O
netMHC2.8.data.tar.gz
```

12. **netCTLpan - 1.1b**

    Follow registration process at: https://services.healthtech.dtu.dk/cgi-bin/sw_request?software=netCTLpan&version=1.1&packageversion=1.1b&platform=Linux

13. **netCTLpan - 1.1 data**

    Download file

```
wget https://www.cbs.dtu.dk/services/NetCTLpan-1.1/data_netCTLpan-1.1.tar.Z
```

14. **netMHCpan - 2.3 data**

    Download file

```
wget https://www.cbs.dtu.dk/services/NetCTLpan-1.1/data_netMHCpan-2.3.tar.Z
```

2.5 Install proprietary software in NeoDisc container

The following section describes how to install the proprietary software in NeoDisc container and setup the NeoDisc pipeline.

1. Edit the following line in **NeoDisc_v1.7.0/install.sh** and provide the full path to the extracted NeoDisc folder (i.e. full path to NeoDisc_v1.7.0/).

```
NEODISC_INSTALLATION_PATH="/FULL/PATH/TO/NeoDisc"
```

2. (Optional) If you're using a software management tool, please load singularity.

3. Configure NeoDisc

   This step will create a container **NeoDisc.sif** with proprietary software installed as well as a wrapper script **NeoDisc.sh**. Go to the installation folder and Run (this may take some time, between 5 and 30 minutes, depending on your system. Please note that tee is optional, but allows to save a copy of the log):

```
sudo ./install.sh 2>&1 | tee install.log
```

4. (Optional) Move **NeoDisc.sh**, **NeoDisc.sif** and **Resources** to the location of your choice - No sudo rights required from that point. **Important:** When moving/transfering files from one location to another, please make sure to **preserve timestamps of Resources files**. Some of the tools check if index files are correct and may otherwise return an error. We suggest using *rsync -aP* command to transfer files. Alternatively, consider transfering the NeoDisc.tar.gz and extract the Resources folder to ensure maintaining timestamps.

```
mv NeoDisc.sif NeoDisc.sh Resources /PATH/TO/NEW/LOCATION/
```

When moving files, ensure to edit the following lines in **NeoDisc.sh**:

```
# Singularity loading command (optional, if using a software manager)
neodiscpath=...
```

5. (Optional) If you're planning to run NeoDisc on an HPC with a queuing system, don't forget to edit the **NeoDisc.sh** script.

```
# Singularity loading command (optional, if using a software manager)
```

We suggest providing a minimum of 24 CPUs and 120G of RAM per job. Note that some software included in NeoDisc (Fragpipe in particular) are particularly memory-consuming. Increasing the memory may be necessary in some cases. If you're not planning to use a software manager, options related to cpus and memory is described at fastq Mode.

Here's an example of a header compatible with SLURM:

```
#SBATCH --job-name=NeoDisc
#SBATCH --output=%x_%j.out
#SBATCH --error=%x_%j.err
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=24
#SBATCH --mem=120G
#SBATCH --time=72:00:00
#SBATCH --export=NONE
```

6. Check if the wrapper script and the container work by typing the following command. This should show the help message of NeoDisc.
Click this link to see the expected output of the command

```
./NeoDisc.sh
```

7. Try the test case example (Gene Panel and/or fastq)

3 Prepare Input data

NeoDisc input data:

	fastq Mode	panel mode	Related section
Paired-end WES/WGS data	Required		Format NGS reads filenames (fastq)
Paired-end RNA sequencing data	Required		Format NGS reads filenames (fastq)
Gene Panel data	Optional	Required	Configuration files variables
HLA Typing data	Optional	Required	Configuration files variables
MS immunopeptidomics data	Optional	Optional	Immunopeptidomics MS files

3.1 Format NGS reads filenames (fastq)

NeoDisc requires specifically formatted fastQ filenames for both WES/WGS and RNAseq fastq files.

1. Reads should be **paired end**.
2. Reads should be in gzipped fastq format **fastq.gz**.
3. All samples should have only two fastQ files **R1 and R2**. If sequencing reads are spread across multiple files, please concatenate them into a single file (ensure consistent ordering of the concatenated files in R1 and R2 fastQ).
4. **Fastq filenames** should be formatted as: *patient-sampleid_tag_R1.fastq.gz* and *patient-sampleid_tag_R2.fastq.gz*.

   Where: *patient*, *sampleid*, and *tag* should all only contain alphanumeric characters (no space, dash, lower dash, or other special characters). *patient* corresponds to the patient name (should be shared if multiple samples are available), *sampleid* corresponds to a sample code (should be unique to each sample), and *tag* could be any of **"DNA", "WES", or "WGS" for WES/WGS reads** and should be set to **"RNA" for RNAseq reads**. **R1** and **R2** are paired-end identifiers. Please note the dash and lower dash separators, which are mandatory as well.
5. All WES/WGS fastq.gz files should be placed in the same directory. All RNAseq fastq.gz files should be placed in their own directory or in the same directory as WES/WGS and RNAseq fastq.gz. Importantly, WES/WGS and RNAseq directories have to be accessible and mountable on the container. Path to the WES/WGS and RNAseq reads has to be defined in the configuration file.

   Examples: WES/WGS reads: NPC1-PBMCs_DNA_R1.fastq.gz, NPC1-PBMCs_DNA_R2.fastq.gz
   RNAseq reads: NPC1-PBMCs_RNA_R1.fastq.gz, NPC1-PBMCs_RNA_R2.fastq.gz

3.2 Immunopeptidomics MS files

NeoDisc only supports **Thermo Orbitrap .raw** files at the moment. All **.raw** files should be placed in a single directory (that is accessible and can be mounted on the container). Path to the **.raw** files has to be defined in the configuration file.

Alternatively, it is also possible to provide the MS identifications in both **.mgf** and **.mzML** format (i.e., provide both the .mgf and .mzML for each raw file). If you want to use this option, please use the following **msconvert parameters** for the conversion of .raw files:

- **DDA → MGF**: *–mgf –filter "peakPicking true 2-" –filter "zeroSamples removeExtra 2-" –filter "msLevel 2-"*
- **DDA → mzML**: *–mzML –64 –zlib –filter "peakPicking true 1-"*
- **DIA → MGF**: *–mgf –filter "peakPicking true 2-" –filter "zeroSamples removeExtra 2-" –filter "msLevel 2-" –filter "demultiplex minWindowSize=2"*

- **DIA → mzML**: *–mzML –64 –zlib –filter "peakPicking true 1-" –filter "demultiplex minWindowSize=2"*

Immunopeptidomics filenames don't have a specific synthax, but have to end with the suffix **.raw**, **.mgf**, or **.mzML**. See *MS_SPECTRA_SAMPLES* in the config file.

3.3 Configuration file

NeoDisc requires a configuration file with each run. The configuration file specifies information related to the sample/patient being analyzed, including the location of the input and output data, run-specific parameters, and sample/patient-specific information. The configuration file is formatted as VARIABLE="Value" pairs defined on separate lines. Note that **all fields are mandatory** and many have **limited possible values**. **Values have to be in between quotes**, without spacing. It is also important to ensure that **NeoDisc configuration filenames end with ".config"**. Comments are allowed, by using a # character.

To get a configuration file template (*template.config*), which is easily editable using *sed* replacements, run:

```
./NeoDisc.sh GetConfigTemplate > template.config
```

3.3.1 Configuration files variables

Here are the variables used in the NeoDisc configuration files:

- **PATIENT_ID**: Patient identifier.
- **GENDER**: Gender of the patient (e.g., "FEMALE").
  Must be one of: ["MALE", "FEMALE", "UNKNOWN"]
- **TISSUE_GTEX**: GTex healthy tissue to use (e.g., "Cervix - Ectocervix").
  Click this link to see available values.
  It is also possible to get the list of available GTEx tissues:

```
./NeoDisc.sh Show_GTEx
```

- **TCGA_CANCERTYPE**: TCGA cancer type to use (e.g., "Cervical Squamous Cell Carcinoma and Endocervical Adenocarcinoma").
  Click this link to see available values.
  It is also possible to get the list of available TCGA cancertypes:

```
./NeoDisc.sh Show_TCGA
```

- **HLA_I**: MixMHCpred alleles to use for class-I predictions, comma-separated and without spacing. Set to "None" if you want NeoDisc to assign it from NGS-typing
  (e.g., "A2402,A3101,B3543,B5101,C0102,C1502").
  Click this link to see available values.
  It is also possible to get the list of available HLA_I:

```
./NeoDisc.sh Show_HLA_I
```

- **HLA_I_NETMHCPAN**: netMHCpan alleles to use for class-I predictions, comma-separated and without spacing. Set to "None" if you want NeoDisc to assign it from NGS-typing (e.g., "HLA-A24:02,HLA-A31:01,HLA-B35:43,HLA-B51:01,HLA-C01:02,HLA-C15:02").
  Click this link to see available values.
  It is also possible to get the list of available HLA_I_NETMHCPAN:

```
./NeoDisc.sh Show_HLA_I_NETMHCPAN
```

- **HLA_II**: MixMHCIIpred alleles to use for class-II predictions, comma-separated and without spacing. Set to "None" if you want NeoDisc to assign it from NGS-typing (e.g., "DPA1_01_03__DPB1_04_01„DRB1_04_04").
  Click this link to see available values.
  It is also possible to get the list of available HLA_II:

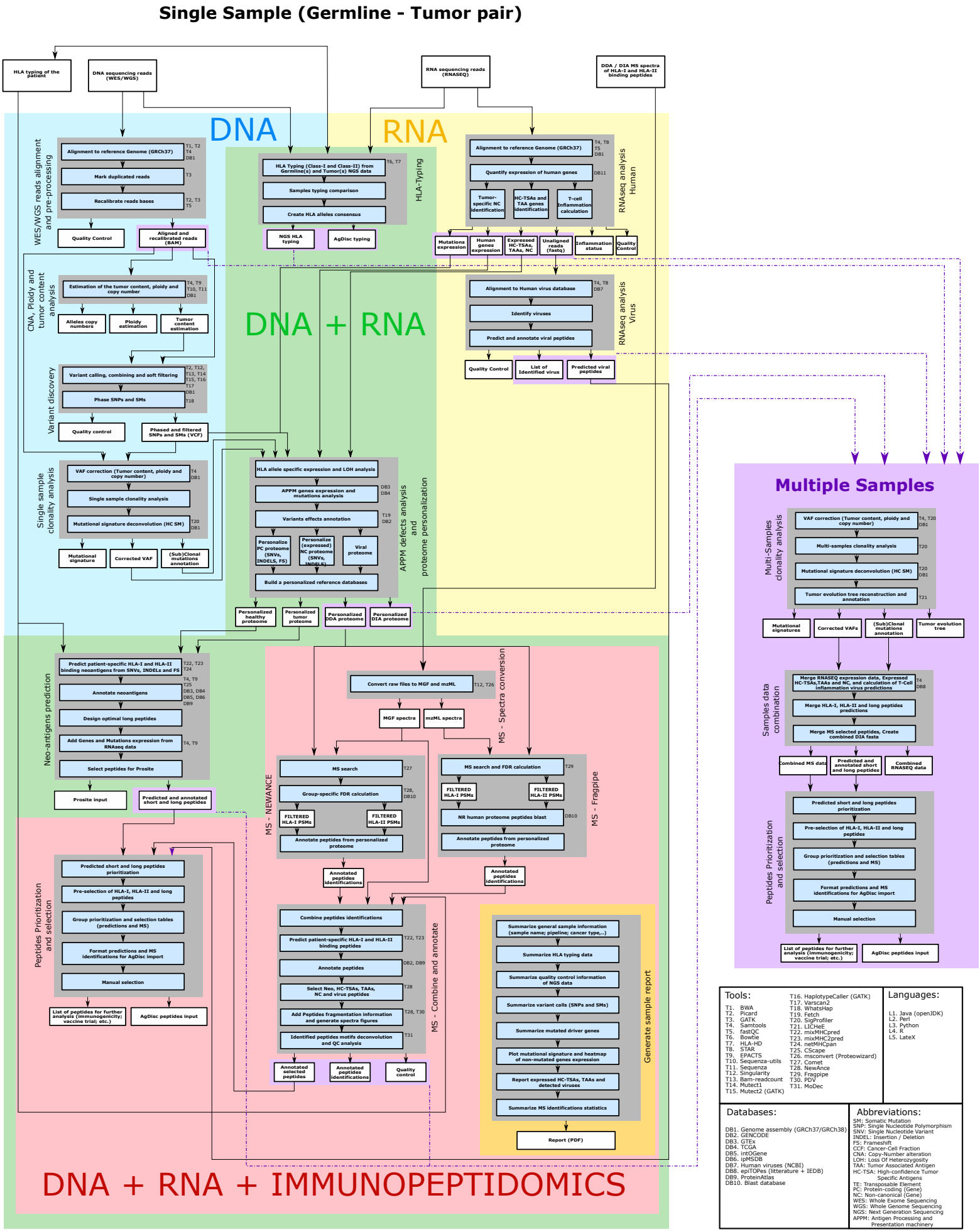
```
./NeoDisc.sh Show_HLA_II
```

- **MAXLONGPEPSIZE**: Integer value, defines the maximum length (in amino acids) for long peptides design (e.g., "25").
  Should be set, at a minimum, to the smallest CI_PEPSIZES/CII_PEPSIZES.
- **CI_PEPSIZES**: Defines the length (in amino-acids) of class-I peptides to predict. Space-separated integer values (e.g., "8 9 10 11 12").
  Min=8, Max=14.
- **CII_PEPSIZES**: Defines the length (in amino-acids) of class-I peptides to predict. Space-separated integer values (e.g., "12 13 14 15 16 17 18 19").
  Min=12, Max=21.
- **NB_CI_SELECTION**: Integer value, Number of class-I peptides to be automatically selected by NeoDisc. Note that a few additional peptides of interest may be added to the selection (e.g., "100").
- **NB_CII_SELECTION**: Integer value, Number of class-II peptides to be automatically selected by NeoDisc. Note that a few additional peptides of interest may be added to the selection (e.g., "20").
- **NB_LONG_SELECTION**: Integer value, Number of long peptides to be automatically selected by NeoDisc. Note that a few additional peptides of interest may be added to the selection (e.g., "15").
- **THRESHOLD_CI_SELECTION**: Float-point value, defines the threshold for predicted class-I peptides selection (MixMHCpred predicted binding Rank). Note that this only affects the selection of predicted neo-antigens (e.g., "2.0").
- **THRESHOLD_CII_SELECTION**: Float-point value, defines the threshold for predicted class-II peptides selection (MixMHC2pred predicted binding Rank). Note that this only affects the selection of predicted neo-antigens (e.g., "2.0").

- **BAD_NTER_RESIDUES**: Space separated amino-acids or amino-acids sequences, defining which amino-acids (sequences) to avoid at N-ter of long peptide sequences. Set to "None" to ignore this option during long peptide design (e.g. for mRNA vaccines). (e.g. "E Q HP").
- **BAD_CTER_RESIDUES**: Space separated amino-acids or amino-acids sequences, defining which amino-acids (sequences) to avoid at C-ter of long peptide sequences. Set to "None" to ignore this option during long peptide design (e.g. for mRNA vaccines). (e.g. "H P C").
- **RESULTS_PATH**: Path to NeoDisc results folder (e.g., "/path/to/NeoDisc/Results/"). The path has to be valid.
- **GENEPANEL_MUTATIONS**: Gene panel mutations to include, comma-separated without spacing. Set to "None" if not available (e.g., "GP1|3|7-140453136-A-T,GP2|4|12-46285701-G-G,GP3|6|11-65373268-G-A").
  Mutations have to be formatted as "GP1|3|7-140453136-A-T" where: GP1 = unique identifier starting by 'GP'; 3 = variant allele frequency; 7-140453136-A-T = chromosome-position-reference_base-mutant_base in GRCH37 coordinates, on the forward (+) strand.
- **DNA_FASTQ_PATH**: Path to the folder that contains the DNA (WES/WGS) fastq files of the germline(s) and tumor(s) (e.g., "/path/to/NeoDisc/fastq/sample/WES").
- **CAPTURE_KIT**: Capture kit used for WES/WGS library preparation. If using whole-genome sequencing data, use "gencode_v43" (recommended) or "wgs". If the capture kit is not available in this list, the use "gencode_v43" is recommended as it will cover the entire annotated genome. (e.g., "Twist_HCEP_V1").
  Must be one of: "SureSelect_Exome_V7", "SureSelect_Exome_V6_COSMIC_r2", "SureSelect_Exome_V5", "xGen_IDT_V1", "xGen_IDT_V2", "Twist_HCEP_V1", "wgs", "gencode_v43".
- **RNASEQ_FASTQ_PATH**: Path to the folder that contains the DNA (WES/WGS) fastq files of the tumor(s) (e.g., "/path/to/NeoDisc/fastq/sample/RNA").
- **RNASEQ_KIT**: Capture kit used for rnaseq library preparation (e.g., "Truseq_Stranded_RNA"). Set to "None" if unknown.
  Must be one of: ["Truseq_RNA", "Truseq_Stranded_RNA", "NEB_Ultra_Directional_RNA", "Agilent_SureSelect_Strand-Specific", "Directional_Illumina", "ScriptSeq_v2_RNA-Seq", "SMARTer_Stranded", "Encore_Complete", "NuGEN_SoLo"].
- **MS_PATH**: Path to the folder that contains the MS raw files (e.g., "/path/to/NeoDisc/fastq/sample/MS").
- **MS_SPECTRA_SAMPLES**: Mapping of ALL raw files sample names, comma-separated and without spacing. Multiple sample names may be present, and sample names don't have to match the ones used as input in NeoDisc. Have to be formatted as: ms_file.raw:samplename (e.g., "CESC1-A_DDA_HLAIIp_R01.raw:CESC1-A1,CESC1-A_DDA_HLAIIp_R02.raw:CESC1-A2,CESC1-A_DDA_HLAIp_R01.raw:CESC1-A1,CESC1-A_DDA_HLAIp_R02.raw:CESC1-A2,CESC1-A_DIA_HLAIIp_R01.raw:CESC1-A1,CESC1-A_DIA_HLAIIp_R02.raw:CESC1-A2,CESC1-A_DIA_HLAIp_R01.raw:CESC1-A1,CESC1-A_DIA_HLAIp_R02.raw:CESC1-A2").
- **MS_SPECTRA_DDA_HLA_I**: List of DDA HLA-I MS raw files, comma-separated and without spacing (e.g., "CESC1-A_DDA_HLAIp_R01.raw,CESC1-A_DDA_HLAIp_R02.raw").
- **MS_SPECTRA_DDA_HLA_II**: List of DDA HLA-II MS raw files, comma-separated and without spacing (e.g., "CESC1-A_DDA_HLAIIp_R01.raw,CESC1-A_DDA_HLAIIp_R02.raw").
- **MS_SPECTRA_DIA_HLA_I**: List of DIA HLA-I MS raw files, comma-separated and without spacing (e.g., "CESC1-A_DIA_HLAIp_R01.raw,CESC1-A_DIA_HLAIp_R02.raw").
- **MS_SPECTRA_DIA_HLA_II**: List of DIA HLA-II MS raw files, comma-separated and without spacing (e.g., "CESC1-A_DIA_HLAIIp_R01.raw,CESC1-A_DIA_HLAIIp_R02.raw").

- **PROJECTS**: Name of the project - only used for traceability, in the PDF report (e.g., "NeoDisc").
- **WES_SEQUENCING_FACILITY**: Name of the facility where WES was sequenced - only used for traceability, in the PDF report (e.g., "Microsynth").
- **RNA_SEQUENCING_FACILITY**: Name of the facility where RNA was sequenced - only used for traceability, in the PDF report (e.g., "Health 2030 Genome Center-Geneva").
- **AIMOFANALYSIS**: Description of the goal of the analysis - only used for traceability, in the PDF report (e.g., "Example of NeoDisc analysis").
- **NOT_AVAILABLE_HLA_I**: Optional, Set to "None". Only used for traceability, in the PDF report. List of HLA alleles that are not available for class-I predictions, comma-separated and without spacing. Is filled in by NeoDisc from NGS-typing. (e.g., "A1234,B1234").
- **NOT_AVAILABLE_HLA_II**: Optional, Set to "None". Only used for traceability, in the PDF report. List of HLA alleles that are not available for class-II predictions, comma-separated and without spacing. Is filled in by NeoDisc from NGS-typing. (e.g., "DPA1_12_34,DRB1_12_34").

Supplementary Figure 1



4 Running NeoDisc

It is strongly advised to use the **NeoDisc.sh** wrapper script for running NeoDisc as it will take care of mounting all folders required for the analysis inside of the singularity container. Because of the runtime of NeoDisc, if you're planning to run NeoDisc on a single computer (i.e. not an HPC) using **tmux** is recommended for running NeoDisc in fastq mode (fastq Mode), or running multiple samples in parallel. It is possible to show the NeoDisc help message by running the wrapping script **NeoDisc.sh** without any argument:

```
./NeoDisc.sh
```

A test case example is provided at Gene Panel and/or fastq

4.1 NeoDisc Modules

NeoDisc pipeline is composed of multiple modules, running sequentially, each relying on the data produced by the upstream module(s). All modules are listed below, with a brief explanation about their function. Note that modules in **underlined** do not run by default (see the requirements in brackets). This information should be in context with the NeoDisc running Modes described below:

- **alignbam-gl**: Aligns to the reference genome and recalibrates germline DNA fastq reads (fastq)
- **alignbam-t**: Aligns to the reference genome and recalibrates tumor DNA fastq reads (fastq)
- **tumor_content**: Tumor content and copy-numbers analysis (fastq)
- **hlatyping**: HLA-typing and HLA-LOH analysis on germline DNA and tumor DNA and RNA (if available) fastq reads (fastq)
- **rnaseqquantification**: Aligns tumor RNA fastq reads to the reference genome; quantification of genes expression; selection of expressed tumor-specific genes (TAAs, Non-canonical); Viral infection analysis; Prediction and prioritization of expressed TAAs and viral antigens (fastq) [requires -rna flag]
- **haplotypecaller**: Variant calling for SNPs and SM identification with haplotypecaller
- **mutect1**: Variant calling for SMs identification with Mutect v1
- **mutect2**: Variant calling for SNPs and SM identification with Mutect v2
- **varscan**: Variant calling for SNPs and SM identification with Varscan2
- **variantscombination**: Combines and determines phasing of all variants identified. (fastq)
- **clonality**: Analysis of SMs frequency, clonal and subclonal classification(fastq)
- **personalizedproteomes**: Aggregates information from tumor_content, hlatyping, *rnaseqquantification*, variantscombination, clonality modules for the creation of personalized proteomes. Includes APPM defects analysis. (fastq, panel)
- **neoprediction**: Predicts class-I and class-II neoantigens, designs optimal long peptide sequences (fastq, panel)
- **convertmsspectra**: Convert raw MS spectra to .mgf and .mzML [requires -ms flag] (fastq)
- **cometms**: COMET + NewAnce DDA MS analysis [requires -ms flag] (fastq)

- **fragpipems**: FragPipe DDA and DIA MS analysis [requires -ms flag] (fastq)
- **combinems**: Combine cometms and fragpipems peptides identifications. Annotates identified peptides. Selects MS identified tumor-specific antigens. Generates MS QC plots. [requires -ms flag] (fastq)
- **prioritizepeptides**: Prioritizes predicted long peptides, class-I and class-II neoantigens. Combines neoantigens, TAAs, viral predictions, and MS-identified TSAs in a single Excel file (fastq, panel)
- **report**: Generates a PDF report of the NeoDisc analysis (fastq, panel)

4.2 NeoDisc running Modes

NeoDisc offers two running modes, each running a subset of the modules described below:

- **fastq**: This mode starts from sequencing fastq files and runs all NeoDisc modules except RNAseq and MS-related modules, which are Optional.
- **panel**: This mode will run predictions from gene-panel data only.

NeoDisc analysis starts with matched germline-tumor samples. The user should submit one analysis for each tumor sample. NeoDisc offers a function to combine individual analyses ran in fastq mode (see Combining Individual Analysis (Clonality Analysis)). The basic syntax for a NeoDisc analysis looks like:

```
./NeoDisc.sh RunPipeline <mode> -p <patient>
                                -t <tumor_sampleid> -g <germline_sampleid>
                                -c </full/path/to/sample.config> [options]
```

4.3 fastq Mode

The fastq mode of NeoDisc is designed to start the analysis pipeline from fastq reads. Note that it is possible to include gene-panel data in the fastq mode by filling up the GENEPANEL_MUTATIONS parameter in the configuration file.

4.3.1 Requirements

Paired-end fastq files for both tumor and germline samples following the standard naming convention described in Format NGS reads filenames (fastq). All NGS fastq and MS files are stored in the folder(s) specified in in the config file.

4.3.2 Running NeoDisc default fastq mode

To run NeoDisc in fastq mode, use the following command (for **WGS** data analysis, the *-wgs* flag should be set (described below):

```
./NeoDisc.sh RunPipeline fastq -p <patient>
                               -t <tumor_sampleid> -g <germline_sampleid>
                               -c <path_to_configfile> [options]
```

Replace the placeholders with your specific details:

- <patient>: A unique identifier for your patient (matching the *patient* described in Format NGS reads filenames (fastq)).
- <tumor_sampleid>: The name of your tumor sample (formatted as *patient-code*, where **patient** is identical to <patient> and **code** is a unique tumor identifier). Note it has to match the tumor sample fastq.gz filename (see Format NGS reads filenames (fastq)).
- <germline_sampleid>: The name of your germline sample (formatted as *patient-code*, where **patient** is identical to <patient> and **code** is a unique germline identifier). Note it has to match the germline sample fastq.gz filename (see Format NGS reads filenames (fastq)).
- <path_to_configfile>: The file path to your configuration file.

Options can be used as needed:

- -rna <tumor_rnasampleid>: Specify if RNA-seq data for the tumor sample is available (following the same format as described above for the <tumor_sampleid>). Multiple samples can be provided, space-separated, and don't have to be identical to <tumor_sampleid>. This option is required to enable RNA-seq analysis.
- -wgs: Specify WGS input data (will tune some parameters for WGS analysis)
- -runid <run_identifier>: Specify a run identifier.
- -ms: Include this flag to turn on MS immunopeptidomics analysis.
- -resume: Include this flag to resume an analysis. Modules which did not run will be restarted. The sample names and runid have to be identical to the original analysis.
- -cpu: Integer, Set the maximum number of CPUs, minimum should be 4 (if using job manager, should match the specifications) - default: 16.
- -mem: Integer, Set the maximum amount of memory available for **fragpipe**, in Go, minimum should be 64 - default: 64. Please note that some software are particularly memory consuming (e.g. Fragpipe) and may require increasing the amount of memory.
- -hlai_loh <keep/discard>: Define wether to keep or discard HLA-I alleles subject to LOH (default is "keep"; recommended for research applications).
- -hlaii_loh <keep/discard>: Define wether to keep or discard HLA-II alleles subject to LOH (default is "keep"; recommended for most applications - except if analyzing cell lines for translational research).
- -sensitive: Include this flag for sensitive mutation calling (Instead of predicting neoantigens identified by a minimum of two variant calling algorithms, predict neoantigens identified by any algorithm - Note that this option will increase the number of false-positive identifications).
- -cleandnafq <light/full>: Include this flag for pre-processing of DNA fastq (only works with paired-end sequencing data; only to be used if errors in WES alignment. light mode only reformats headers (faster), full mode performs deduplication in addition)
- -cleanrnafq <light/full>: Include this flag for pre-processing of RNA fastq (only works with paired-end sequencing data; only to be used if errors in WES alignment. light mode only reformats headers (faster), full mode performs deduplication in addition)

Command line examples: *One patient (Mel1), One germline sample (Mel1-GL), One WES tumor sample (Mel1-Tum1), two RNA-seq tumor samples (Mel1-Tum1a+Mel1-Tum1b), and MS data available*

```
./NeoDisc.sh RunPipeline \
                    fastq \
                    -p Mel1 \
                    -t Mel1-Tum1 \
                    -g Mel1-GL \
                    -c /path/to/Mel1.config \
                    -rna Mel1-Tum1a Mel1-Tum1b \
                    -ms \
                    -runid exampleFASTQrun
```

4.3.3 Running NeoDisc sensitive fastq mode

Running NeoDisc in sensitive mode is actually the same as running it in fastq, and adding the **"-sensitive"** option:

```
./NeoDisc.sh RunPipeline \
                    fastq \
                    -p Mel1 \
                    -t Mel1-Tum1 \
                    -g Mel1-GL \
                    -c /path/to/Mel1.config \
                    -rna Mel1-Tum1a Mel1-Tum1b \
                    -ms \
                    -sensitive \
                    -runid exampleFASTQSENSITIVErun
```

4.4 panel mode

The panel mode in NeoDisc is designed for processing and analyzing **gene-panel only** data. It is possible to include gene-panel data in the fastq mode by filling up the GENEPANEL_MUTATIONS parameter in the configuration file.

4.4.1 Requirements

The **GENEPANEL_MUTATIONS** parameter is set in the configuration file, following the standard format: "GP1|3|7-140453136-A-T" where: **GP1** = unique identifier starting by 'GP'; **3** = variant allele frequency; **7-140453136-A-T** = chromosome-position-reference_base-mutant_base in GRCH37 coordinates, on the forward (+) strand. Multiple gene identifiers can be provided as comma-separated without spaces.

4.4.2 Running NeoDisc in panel Mode

To run NeoDisc in panel mode, use the following command:

```
./NeoDisc.sh RunPipeline panel -p <patient>
                               -t <tumor_sampleid> -g <germline_sampleid>
                               -c <path_to_configfile> [options]
```

Replace the placeholders with your specific details:

- <patient>: A unique identifier for your patient (matching the *patient* described in Format NGS reads filenames (fastq)).
- <tumor_sampleid>: The name of your tumor sample (formatted as *patient-code*, where **patient** is identical to <patient> and **code** is a unique tumor identifier, see Format NGS reads filenames (fastq)).
- <germline_sampleid>: The name of your germline sample (formatted as *patient-code*, where **patient** is identical to <patient> and **code** is a unique germline identifier, see Format NGS reads filenames (fastq)).
- <path_to_configfile>: The file path to your configuration file.

(Options) can be added as needed:

- -runid <run_identifier>: Specify a run identifier.
- -cpu: Integer, Set the maximum number of CPUs, minimum should be 4 (if using job manager, should match the specifications) - default: 16. (the default value can be lowered down in panel mode)

Command line example: Note that the -p and -t parameters are mandatory and can be set to anything by the user. *One patient (Mel1) with gene-panel data included in /path/to/Mel1.config*

```
./NeoDisc.sh RunPipeline \
                    panel \
                    -p Mel1 \
                    -t Mel1-GP \
                    -g Mel1-GL \
                    -c /path/to/Mel1.config \
                    -runid exampleGPrun
```

4.5 Combining Individual Analysis (Clonality Analysis)

NeoDisc supports data integration of multiple lesions as well as longitudinally collected samples from the patient. This allows the combination of individually identified variants, human and viral gene expression, predicted and MS-identified tumor-specific antigens, and additionally provides analysis of tumor heterogeneity and evolution.

4.5.1 Requirements

All samples were analyzed individually in **fastq** mode, see fastq Mode.

4.5.2 Combining Individual Analysis

To combine individual analysis, use the following command:

NeoDisc supports data integration of multiple lesions as well as longitudinally collected samples from the patient. This allows the combination of individually identified variants, human and viral gene expression, predicted and MS-identified tumor-specific antigens and additionally provides analysis of tumor heterogeneity and evolution.

Requirements: All samples were analyzed individually in **fastq** mode.

Combining individual analysis:

```
./NeoDisc.sh MergeSamples <patient_name> <path_to_configfile>
                          <number_cpus> <samples_data>
```

Replace the placeholders with your specific details:

- <patient>: A unique identifier for your merged sample.
- <path_to_configfile>: The file path to your configuration file. This configuration file has to contain HLA typing information. You should, therefore, use one of the config files produced by NeoDisc analysis (see the Results section - Individual analysis - `PATIENT.config`).
- <number_cpus>: Number of CPUs to use for the job.
- <samples_data>: Space-separated data of each of the sample to merge. For each sample, comma-separated **patient, runid, germline_sampleid, tumor_sampleid, tumor_rnasampleid**. Note that if multiple **tumor_rnasampleid** samples were provided, you can include them all as "+" separated (note the different formatting compared to the single sample format).

Command line example: *One patient (Mel1), One germline sample (Mel1-GL), Two WES tumor samples (Mel1-Tum1 and Mel1-Tum2), having one and two RNAseq samples (Mel1-Tum1a+Mel1-Tum1b, and Mel1-Tum2, respectively)*

```
./NeoDisc.sh MergeSamples \
               Mel1 \
               .../Results/Mel1/examplerun/Mel1-GL__Mel1-Tum1/Mel1.config \
               12 \
               Mel1,examplerun,Mel1-GL,Mel1-Tum1,Mel1-Tum1a+Mel1-Tum1b \
               Mel1,examplerun,Mel1-GL,Mel1-Tum2,Mel1-Tum2
```

4.6 Test cases

4.6.1 Gene Panel

A test case is provided for testing NeoDisc. This test case is limited to illustrating the pipeline's functionality in 'panel' mode. It's important to note that the results obtained in this mode are not as comprehensive as those achievable in the 'fastq' mode of the pipeline (see NeoDisc Results for more information).

- Start by retrieving the config file example included in NeoDisc. This config file contains fictive data, including a total of 55 gene panel variants. **Mutations GP1, GP2 and GP3 are synonymous mutations** and **Mutations GP54 and GP55 are premature stop mutations**, which are consequently not actionable mutations and therefore will not be included in the prioritization results.

```
# Replace "/path/to/" by an actual path
./NeoDisc.sh GetConfigExample > /path/to/Example.config
```

- In */path/to/Example.config*, replace the **__RESULTS_PATH__** with a local path (the path has to be valid).
- Run NeoDisc. This analysis is expected to finish within 5-15 minutes, depending on the computer (Please note that tee is optional, but allows to save a copy of the log).

```
./NeoDisc.sh RunPipeline \
                    panel \
                    -p Example \
                    -t Example-GP \
                    -g Example-germline \
                    -c /path/to/Example.config \
                    -runid exampleGPrun \
                    -cpu 4 2>&1 | tee exampleGPrun.log
```

- Check the results folder defined in your configuration file (**RESULTS_PATH**). Within this folder, the following subfolders should have been created: *.../Example/exampleGPrun/Example-germline__Example-GP*. NeoDisc results for this analysis will be in this folder (for more details about the folder structure, please see Individual Analysis Results).
  You should see the following files:
  - *.../Example/exampleGPrun/Example-germline__Example-GP/***Example-GP_report.pdf** : NeoDisc report, summarizing the analysis.
  - *.../Example/exampleGPrun/Example-germline__Example-GP/*Prioritization/**Example-GP_Prioritization.xlsx** : NeoDisc prioritization file (for more details about the content, see Prioritization.xlsx).

  Expected log and result files are attached below (click the links):
  - Log
  - PDF Report
  - Excel prioritization

4.6.2 fastq

A test case is provided for testing NeoDisc. Due to the sensitive nature of sequencing data, such data can't be provided for testing. Consequently, sequencing data have to be downloaded by the user. We provide here the procedure for the download and analysis of a publicly available data, deposited on the Sequence Read Archive (SRA) by Steven A. Rosenberg laboratory. The dataset derives from a melanoma metastasis sample (2369) in which three immunogenic peptide sequences reported by Gartner et al. 2021:

Gene	Mutation	Immunogenic mutant peptide
PLEKHM2	p.H902Y	LTDDRLFTCY
PPP1R3B	p.P176H	YTDFHCQYV
DOP1B	p.P2168L	FPPDKMLLF

Please note that germline and tumor WES data don't cover HLA-I and -II regions and therefore won't be typed. However RNAseq reads do cover HLA-I and-II regions and will therefore be used for HLA typing.

- Download and install sra-toolkit (see: https://github.com/ncbi/sra-tools/wiki/02.-Installing-SRA-Toolkit)
- Download WES and RNAseq data from SRA

SRA Run	Sample type	Sequencing strategy
SRR5128254	Tumor	RNAseq
SRR5122419	Tumor	WES
SRR5122514	Germline	WES

```
for id in SRR5128254 SRR5122514 SRR5122419; do
  ./fastq-dump --split-3 --gzip ${id} &
done
wait
```

- Once download finished, create directory structure and rename fastq files

```
mkdir -p 2369/DNA 2369/RNA

mv SRR5128254_1.fastq.gz 2369/RNA/2369-tumor_RNA_R1.fastq.gz
mv SRR5128254_2.fastq.gz 2369/RNA/2369-tumor_RNA_R2.fastq.gz

mv SRR5122419_1.fastq.gz 2369/DNA/2369-tumor_WES_R1.fastq.gz
mv SRR5122419_2.fastq.gz 2369/DNA/2369-tumor_WES_R2.fastq.gz

mv SRR5122514_1.fastq.gz 2369/DNA/2369-germline_WES_R1.fastq.gz
mv SRR5122514_2.fastq.gz 2369/DNA/2369-germline_WES_R2.fastq.gz
```

- Get NeoDisc configuration file

```
./NeoDisc.sh GetTestConfig > /path/to/2369.config
```

- In */path/to/2369.config*, replace **__RESULTS_PATH__** with a local path; replace **__DNA_FASTQ_PATH__** and **__RNA_FASTQ_PATH__** by the local path to 2369/DNA and 2369/RNA folders, respectively (all path have to be valid).
- Run NeoDisc. The analysis should finish in less than 12h (tested with 24 cpu and 64G of RAM). (Please note that tee is optional, but allows to save a copy of the log).

```
./NeoDisc.sh RunPipeline \
                    fastq \
                    -p 2369 \
                    -t 2369-tumor \
                    -g 2369-germline \
                    -rna 2369-tumor \
                    -c /path/to/2369.config \
                    -runid exampleFQrun \
                    -cpu 24 2>&1 | tee exampleFQrun.log
```

Expected log and result files (truncated for safety reasons) are attached below (click the links):

- Log
- PDF Report
- Excel prioritization

5 NeoDisc Results

NeoDisc generates numerous data derived from genomics, transcriptomics, and immunopeptidomics. Importantly, NeoDisc generates mostly two important files, which are a PDF report (**PATIENT-TID_report.pdf**) summarizing the analysis and an Excel file (**PATIENT-TID_Prioritization.xlsx**) reporting prioritized lists of tumor-specific antigens. Reports contain a copy of the Excel prioritization tables. A more detailed list of the data and their structure is provided below. The report of an Epidermoid cervical carcinoma sample is provided as example. This example shows how viral infections are reported and prioritized by NeoDisc. In addition, it also shows how immunopeptidomics data are reported. The excel prioritization tables for this sample are provided at the end of the report and are limiter to the first 15 rows, for safety reasons.

- CESC1-A_report.pdf

5.1 Prioritization.xlsx

The Excel file contains multiple sheets, each reporting specific tumor-specific antigens. The different sheets are listed below, and the columns of interest to the user are explained. Of note, unless specified differently, **all peptides predictions (alleles and ranks) derive from MixMHCpred and MixMHC2pred %Rank, for class-I and -II, respectively**.

5.1.1 MS_peptides (Optional)

Only available if mass-spectrometry analysis was turned on **(-ms flag)**. Reports **all tumor-specific antigens identified from immunopeptidomics data analysis**.

Columns of interest:

- **peptide_id**: A unique peptide identifier
- **Samples_Injection**: List of samples in which the peptide was identified by MS
- **ms_pipeline**: List of pipelines that identified the peptide
- **MeasurementMethod**: List of the measurement methods in which the peptide was identified
- **PSMCount**: Number of PSMs of the peptide
- **Max.XCorr**: Maximum PSM XCorr value reported for the peptide
- **Max.DeltaCn**: Maximum PSM DeltaCn value reported for the peptide
- **Max.Hyperscore**: Maximum PSM Hyperscore value reported for the peptide
- **Max.PeptideProphet Probability**: Maximum PSM PeptideProphet Probability value reported for the peptide
- **PrecursorQuant**: Precursor quantifications of all PSMs, reported per sample
- **Peptide**: The peptide identified (with modifications)
- **Sequence**: The peptide sequence without the modifications
- **neo_wt_seq**: WT sequence of the peptide sequence (only for Neoantigens)
- **SequenceCoverage**: Measure of the coverage of the peptide sequence, from the best PSM spectrum

- **PEPTIDE_TYPE**: Type of peptide (TAA = manually reviewed tumor-associated antigen, CANCER-SPECIFIC-PC = unreviewed tumor-associated antigen, NEO-FS = neoantigen derived from a frameshift mutation; NEO-INSERTION = neoantigen derived from an in-frame insertion mutation; NEO-DELETION = neoantigen derived from an inframe deletion mutation; NEO-SNV = neoantigen derived from a single nucleotide variant, NEO-DNV = neoantigen derived from a di-nucleotide variant; NC = non-canonical tumor-specific antigen; NH = viral antigen)
- **HLAType**: Peptide was identified from HLA-I or HLA-II immunopeptidomics data
- **ProteinQuant**: Protein quantification, reported per sample
- **GENES**: Gene names of entries containing the peptide sequences
- **EXTERNAL_IDs**: ENSEMBL and/or Refseq sequences identifier
- **rnaseq_TPM**: Expression of the gene in TPM, from RNAseq data
- **GENES_VARIANTS**: Variants included in the peptide sequences (only for neoantigens)
- **VARIANT_TYPE**: Variant is a SNP or a SM
- **VARIANT_NEODISC_FILTER**: Quality of the variant identified by NeoDisc. INCL = High-quality variant (two or more calling algorithms); EXCL = Low-quality variant (single variant calling algorithm)
- **TRANSCRIPT_COORDINATES**: Transcript coordinates from which the peptide sequence derived (only for NC entries)
- **PREDICTED_BINDER**: HLA allele(s) of the patient with the best-predicted binding affinity to the peptide sequence
- **BINDING_RANK**: Predicted binding rank of the Predicted_Best_alleles to the peptide sequence
- **EPITOPES_PEPTIDE: immunogenicity**: Annotated immunogenicity of the peptide sequence in EpiTOPes database
- **EPITOPES_PEPTIDE: Number validations**: Number of validation of the peptide as immunogenic in EpiTOPes database
- **EPITOPES_PEPTIDE: Patient immunogenic HLA**: HLA alleles of the patient reported as immunogenic with this peptide in EpiTOPes database
- **HumanProteomeIDs**: All protein derived from the non-redundant human blast database that contains the peptide sequence

If the file derived from the combined analysis, the following additional columns are available:

- **Samples**: Lists all samples in which analysis the peptide was identified
- **Nb_Samples**: Number of samples in which analysis the peptide was identified
- **Mutation_Category**: Defines if the mutation was truncal, shared, or clonal (only for neo-antigens).

5.1.2 Expressed_TAAs_peptides (Optional)

Only available if the -rna option was used and tumor-specific genes were found expressed. Reports **prioritized predicted peptides from expressed tumor-specific genes**.

Columns of interest:

- **Rank**: Rule-based ranking of the peptides (HLA-I and -II together)
- **Rank_CI**: Rule-based ranking of the HLA-I peptides
- **Rank_CII**: Rule-based ranking of the HLA-I peptides
- **Type**: Indicates whether the predicted is from the high-confidence curated list (HC-TSAs) or not an unreviewed tumor-associated antigen (TAA)
- **peptide_id**: A unique peptide identifier
- **database_entry**: ENSEMBL gene identifier associated with the entry
- **gene**: Gene name
- **expression**: Expression value of the gene from RNAseq (TPM)
- **wt_seq**: Sequence of the tumor-specific predicted peptide (annotated as wt because the sequence remains wt, but the expression of the gene is tumor-specific)
- **Epitopes_immunogenicity**: Annotated immunogenicity of the peptide sequence in EpiTOPes database
- **Epitopes_validations**: Number of validation of the peptide as immunogenic in EpiTOPes database
- **Epitopes_HLA**: HLA alleles of the patient reported as immunogenic with this peptide in EpiTOPes database
- **Predicted_Best_alleles**: HLA allele(s) of the patient with the best predicted binding affinity to the wt_seq
- **Additional_predicted_alleles**: Additional HLA alleles of the patient predicted to bind wt_seq with a rank $\leqslant 2$
- **Predicted_Best_rank**: Predicted binding rank of the Predicted_Best_alleles to the wt_seq
- **bestWTMatchScore_I/II**: Number of HLA-I / II peptides amino acids in ipMSDB overlapping the wt_seq
- **bestWTMatchOverlap_I/II**: Highest overlapping fraction of the peptide with an HLA-I / II peptide present in ipMSDB. (0: No overlapping peptide; 1: Full overlap)
- **bestWTMatchType_I/II**: Matching type of the peptide with the highest overlapping HLA-I / II peptide present in ipMSDB. (NONE: no overlapping peptide; PARTIAL/PARTIAL_MUT: partial overlap; INCLUDED: a longer version of the peptide is present in ipMSDB; COVER: a shorter version of the peptide is present in ipMSDB; EXACT: The exact same peptide is present in ipMSDB)
- **bestWTMatchScore_I/II**: Number of peptides in ipMSDB overlapping with the peptide

5.1.3 VIRUS_RNA_HC_peptides (Optional)

Only available if the -rna option was used and high-confidence viruses were detected. Reports **prioritized predicted peptides from the identified viral proteome(s)**.

Columns of interest:

- **Rank**: Rule-based ranking of the peptides (HLA-I and -II together)
- **Rank_CI**: Rule-based ranking of the HLA-I peptides
- **Rank_CII**: Rule-based ranking of the HLA-I peptides
- **peptide_id**: A unique peptide identifier
- **Identifier**: Refseq identifier associated with the entry
- **Organism**: Refseq identifier of the virus
- **gene**: Viral gene name
- **expression**: Expression value of the viral gene from RNAseq (TPM)
- **pep_start / pep_end**: Start and end coordinates of the peptide sequence relative to the viral protein sequence
- **pep_sequence**: Viral peptide sequence
- **best_core**: Predicted binding core - only for class-II predicted peptides
- **Final_Peptide_Class/_PRIME**: Peptide is predicted to bind on HLA-I (mixMHCpred / PRIME) and/or -II (mixMHC2pred) allele(s) of the patient
- **best_alleles/_PRIME**: HLA allele(s) of the patient with the best predicted binding affinity to the pep_sequence by mixMHCpred / PRIME and mixMHC2pred
- **other_significant_alleles/_PRIME**: Additional HLA alleles of the patient predicted to bind pep_sequence with a rank $\leqslant 2$ by mixMHCpred / PRIME and mixMHC2pred
- **rank/_PRIME**: Predicted binding rank of the Predicted_Best_alleles to the pep_sequence by mixMHCpred / PRIME and mixMHC2pred
- **EPITOPES**: Peptide immunogenicity: Annotated immunogenicity of the peptide sequence in EpiTOPes database
- **EPITOPES**: Number validations: Number of validation of the peptide as immunogenic in EpiTOPes database
- **EPITOPES**: Patient immunogenic HLA: HLA alleles of the patient reported as immunogenic with this peptide in EpiTOPes database

If the file derived from the combined analysis, the following additional columns are available:

- **Samples**: Lists all samples in which analysis the peptide was predicted
- **Nb_Samples**: Number of samples in which analysis the peptide was predicted

5.1.4 Class_I/Class_II

Reports all **rule-based prioritized predicted class-I and -II neoantigens**.
Columns of interest:

- **Prioritization_rulebased**: Ranking of peptides by the rule-based algorithm
- **Prioritization_ML**: Ranking of peptides by the ML algorithm
- **Selection**: Suggested ranking of peptides for the final selection
- **peptide_id**: A unique peptide identifier
- **mutation_id**: A unique mutation identifier. Describes genomic coordinates (chromosome-position-ref-alt) of the mutation from which the peptide sequence derived.
- **callers**: Lists variant calling algorithms that identified the variant (HC: HaplotypeCaller, M1: Mutect1, M2: Mutect2, VS: Varscan)
- **VAF**: Variant allele frequency of the mutation
- **CCF**: Cancer cell fraction of the mutation
- **Clonality**: Clonality assessment of the mutation derived from the CCF (clonal or subclonal)
- **CopyNumbers**: Chromosomal copy-numbers at the position of the mutation
- **gene**: Mutated gene name
- **mutation**: Mutation at the protein level (reference aa, position, mutant aa)
- **database_entry**: ENSEMBL gene identifier associated with the entry
- **phased_sm / phased_snp**: List of other somatic mutations and SNPs included in the peptide sequence
- **rnaseq_TPM**: Expression value of the gene from RNAseq (TPM)
- **rnaseq_coverage**: Number of RNAseq reads covering the coordinates of the mutation
- **rnaseq_detected_variants**: List of amino acids found at the coordinates of the mutation in RNAseq reads (formatted as aminoacid = Number of occurrence)
- **%rnaseq_ref_support**: Percentage of the RNAseq reads supporting the reference (wt) amino acid at the position of the mutation
- **%rnaseq_alt_support**: Percentage of the RNAseq reads supporting the alt (mutant) amino acid at the position of the mutation
- **CSCAPE_score**: CScape score of the mutation
- **mutant_seq**: Neoantigen sequence
- **wt_seq**: Wild-type counterpart of the sequence. This can be set to "NA" in case a wt sequence can't be derived (e.g. frameshift mutations)
- **mutant_best_core**: Predicted binding core of the mutant_seq - only for class-II predicted peptides
- **Mutation_position_in_Core**: Position of the mutation within the mutant_best_core (only relevant for class-II peptides; 0-based coordinates)
- **Mutation_distance_to_Core**: Distance of the mutation from the mutant_best_core (only relevant for class-II peptides; 0-based coordinates)

- **Final_Peptide_Class/_PRIME/_netMHCpan**: Peptide is predicted by all mixMHCpred / PRIME / netMHCpan or one of them to bind on HLA-I and/or by mixMHC2pred on HLA-II allele(s) of the patient
- **mutant_best_alleles/_PRIME/_netMHCpan**: HLA allele(s) of the patient with the best predicted binding affinity to the mutant_seq by mixMHCpred / PRIME / netMHCpan and mixMHC2pred
- **mutant_other_significant_alleles/_PRIME/_netMHCpan**: Additional HLA alleles of the patient predicted to bind mutant_seq with a rank $\leqslant 2$ by mixMHCpred / PRIME / netMHCpan and mixMHC2pred
- **mutant_rank/_PRIME/_netMHCpan**: Predicted binding rank of the mutant_best_alleles to the mutant_seq by mixMHCpred / PRIME / netMHCpan and mixMHC2pred
- **wt_rank/_PRIME/_netMHCpan**: Predicted binding rank of the wt_seq to the mutant_best_alleles (i.e., the same allele as for the mutant sequence)
- **wt_best_alleles/_PRIME/_netMHCpan**: HLA allele(s) of the patient with the best predicted binding affinity to the wt_seq by mixMHCpred / PRIME / netMHCpan and mixMHC2pred
- **wt_best_rank/_PRIME/_netMHCpan**: Predicted binding rank of the mutant_best_alleles to the wt_seq by mixMHCpred / PRIME / netMHCpan and mixMHC2pred
- **Sample_Tissue_expression_GTEx**: Median expression (TPM) of the gene in the associated healthy tissue from GTEx (Sample_Tissue)
- **GTEx_all_tissues_expression_median/_mean**: Median / mean expression (TPM) of the gene in the associated all healthy tissues from GTEx
- **TCGA_Cancer_expression**: Median expression (TPM) of the gene in the associated cancer type in TCGA (Cancer_Type)
- **INTRACELLULAR_LOCATIONS / EXTRACELLULAR_LOCATIONS**: Annotation of the intra/extra cellular location of the protein
- **gene_driver_Intogen**: Intogen annotation describing whether the gene is a driver
- **mutation_driver_statement_Intogen**: Intogen annotation describing whether the mutation is a driver
- **bestWTMatchScore_I/II**: Number of HLA-I / II peptides amino acids in ipMSDB overlapping the wt_seq
- **bestWTMatchOverlap_I/II**: Highest overlapping fraction of the peptide with an HLA-I / II peptide present in ipMSDB (0: No overlapping peptide; 1: Full overlap)
- **bestWTMatchType_I/II**: Matching type of the peptide with the highest overlapping HLA-I / II peptide present in ipMSDB (NONE: no overlapping peptide; PARTIAL/PARTIAL_MUT: partial overlap; INCLUDED: a longer version of the peptide is present in ipMSDB; COVER: a shorter version of the peptide is present in ipMSDB; EXACT: The exact same peptide is present in ipMSDB)
- **bestWTMatchScore_I/II**: Number of peptides in ipMSDB overlapping with the peptide

If the file derived from the combined analysis, the following additional columns are available:

- **Samples**: Lists all samples in which the analysis the peptide was predicted
- **Nb_Samples**: Number of samples in which the analysis the peptide was predicted
- **Mutation_Category**: Defines if the mutation was truncal, shared, or clonal.

5.1.5 Class_I_selection

Reports the suggested selection of the **top N** (defined in the configuration file) predicted **class-I neoantigens, prioritized by the ML algorithm**. Note that MS identified neoantigens are placed at the top of the list, while non-predictable neoantigens and neoantigens derived from driver mutations (not covered in the top N) are reported at the bottom of the list. All the columns are a subset of the columns present in the **Class_I** (described above)

5.1.6 Class_II_selection

Reports the suggested selection of the **top N** (defined in the configuration file) predicted **class-II neoantigens, prioritized by the rule-based algorithm**. Note that MS identified neoantigens are placed at the top of the list, while non-predictable neoantigens and neoantigens derived from driver mutations (not covered in the top N) are reported at the bottom of the list. All the columns are a subset of the columns present in the **Class_II** (described above)

5.1.7 long_peptides

Reports all **rule-based prioritized long neoantigens sequences**.
Columns of interest:

- **Prioritization_rulebased**: Ranking of peptides by the rule-based algorithm
- **Prioritization_ML**: Ranking of peptides by the ML algorithm
- **Selection**: Suggested ranking of peptides for the final selection
- **peptide_id**: A unique peptide identifier
- **LONGPEPTIDE_TAG**: Annotation about the long peptide (BEST: Longest optimal sequence covering the mutation, ADDITIONAL: Shorter optimal sequences covering the mutation).
- **mutation_id**: A unique mutation identifier. Describes genomics coordinates (chromosome-position-ref-alt) of the mutation from which the peptide sequence derived.
- **callers**: Lists variant calling algorithms identified the variant (HC: HaplotypeCaller, M1: Mutect1, M2: Mutect2, VS: Varscan)
- **CCF**: Cancer cell fraction of the mutation
- **Clonality**: Clonality assessment of the mutation derived from the CCF (clonal or subclonal)
- **CopyNumbers**: Chromosomal copy-numbers at the position of the mutation
- **VAF**: Variant allele frequency of the mutation
- **gene**: Mutated gene name
- **mutation**: Mutation at the protein level (reference aa, position, mutant aa)
- **Sample_Tissue_expression_GTEx**: Median expression (TPM) of the gene in the associated healthy tissue from GTEx (Sample_Tissue)
- **TCGA_Cancer_expression**: Median expression (TPM) of the gene in the associated cancer type in TCGA (Cancer_Type)
- **rnaseq_TPM**: Expression value of the gene from RNAseq (TPM)
- **rnaseq_coverage**: Number of RNAseq reads covering the coordinates of the mutation
- **rnaseq_detected_variants**: List of amino-acids found at the coordinates of the mutation in RNAseq reads (formatted as aminoacid = Number of occurrence)

- **%rnaseq_ref_support**: Percentage of the RNAseq reads supporting the reference (wt) amino-acid at the position of the mutation
- **%rnaseq_alt_support**: Percentage of the RNAseq reads supporting the alt (mutant) amino-acid at the position of the mutation
- **mutant_seq**: Long neoantigen sequence
- **wt_seq**: Wild-type counterpart of the sequence (can be set to "NA" in case a wt sequence can't be derived, e.g., frameshift mutations)
- **ALTERNATIVE_MUTANT_LONG_PEPTIDES / ALTERNATIVE_WT_LONG_PEPTIDES**: Lists all other mutant / wt long peptide sequences available for the mutation
- **INCLUDED_SHORT_PEPTIDES**: Lists all short peptides covered by the long sequence
- **MIN_MUT_RANK_CI_MIXMHC/_PRIME/_netMHCpan**: Best predicted binding rank of the included class-I short peptide to any of the patient HLA by mixMHCpred / PRIME / netMHCpan
- **TOP5_MUT_RANK_CI_MIXMHC/_PRIME/_netMHCpan**: Top 5 best predicted binding ranks of the included class-I short peptide to the patient HLA by mixMHCpred / PRIME / netMHCpan
- **MIN_MUT_RANK_ALLELES_CI_MIXMHC/_PRIME/_netMHCpan**: HLA allele with the best predicted binding to any of the short class-I peptide sequence included within the mutant_seq
- **ADDITIONAL_MUT_RANK_ALLELES_CI_MIXMHC/_PRIME/_netMHCpan**: All HLA alleles with a predicted binding $\leqslant 2$ to any of the short class-I peptide sequences included within the mutant_seq
- *The same columns are available for class-II and wt sequences. For class-I, MixMHC2pred was used*
- **INTRACELLULAR_LOCATIONS / EXTRACELLULAR_LOCATIONS**: Annotation of the intra/extra cellular location of the protein
- **CSCAPE_score**: CScape score of the mutation
- **gene_driver_Intogen**: Intogen annotation describing whether the gene is a driver
- **mutation_driver_statement_Intogen**: Intogen annotation describing whether the mutation is a driver
- **GTEx_all_tissues_expression_median/_mean**: Median / mean expression (TPM) of the gene in the associated all healthy tissues from GTEx
- **database_entry**: ENSEMBL gene identifier associated with the entry
- **phased_sm / phased_snp**: List of other somatic mutations and SNPs included in the peptide sequence
- **bestWTMatchScore_I/II**: Number of HLA-I / II peptides amino acids in ipMSDB overlapping the wt_seq
- **bestWTMatchOverlap_I/II**: Highest overlapping fraction of the peptide with an HLA-I / II peptide present in ipMSDB (0: No overlapping peptide; 1: Full overlap)
- **bestWTMatchType_I/II**: Matching type of the peptide with the highest overlapping HLA-I / II peptide present in ipMSDB. (NONE: no overlapping peptide; PARTIAL/PARTIAL_MUT: partial overlap; INCLUDED: a longer version of the peptide is present in ipMSDB; COVER: a shorter version of the peptide is present in ipMSDB; EXACT: The exact same peptide is present in ipMSDB)

- **bestWTMatchScore_I/II**: Number of peptides in ipMSDB overlapping with the peptide

If the file derived from the combined analysis, the following additional columns are available:

- **Samples**: Lists all samples in which analysis the peptide was predicted
- **Nb_Samples**: Number of samples in which analysis the peptide was predicted
- **Mutation_Category**: Defines if the mutation was truncal, shared, or clonal.

5.1.8 long_peptides_selection

Reports the selection of the **top N** (defined in the configuration file) predicted **long neoantigens sequences, prioritized by the ML algorithm**. Note that MS identified neoantigens are placed at the top of the list and that neoantigens derived from driver mutations or non-prioritizable with the ML algorithm are placed at the bottom of the list. All the columns are a subset of the columns present in the **long_peptides** (described above).

5.2 Individual Analysis Results

The detailed structure of the results of NeoDisc individual analysis resulting from the **fastq** mode (the **panel** mode results in a subset of the directory and files) is available by clicking this link .
PATIENT, **GL**, **TSID**, and **Runid** refer to the `-s`, `-g`, `-t`, and `-runid` arguments, respectively.
Only the files of interest to the end user are listed. A description of the folders and files is available on the same line.

5.3 Combined analysis (clonality analysis) Results

The detailed structure of the results of NeoDisc combined analysis resulting from the **MergeSamples** command is available by clicking this link.
Only the files of interest to the end user are listed. A description of the folders and files is available on the same line.

6 FAQ

Because of the complexity of the pipeline, most errors are caused by incorrect command line, path to the data, configuration file, or formatting of the data. In case of an error, please start by looking at those.

Next, please look at the log files provided by NeoDisc (in the log folders), starting with the most recent one.

- **NeoDisc does not start and displays the help message:**

  The command provided to NeoDisc is incorrect. Locate the line *COMMAND LINE:* within the *PIPELINE SUBMISSION INFORMATION* section in the **log** and find the error in your command line.

- **Locate where an error occurs:**

  1. Start by looking at the main log from NeoDisc (e.g. exampleFQrun.log) and identify in which module NeoDisc failed.
  2. Find the log file for that module, within the NeoDisc results folder, and identify which command/tool reported the error.
  3. Locate the log of that particular command/tool within the module folder.

- **The index file is older than the data file error:**

  This is typically caused by copy-pasting **Resources** files without preserving timestamps. When moving/transfering files from one location to another, please make sure to **preserve timestamps of Resources files**. Some of the tools check if index files are correct and may otherwise return an error. We suggest using *rsync -aP* command to transfer files. Alternatively, consider transfering the NeoDisc.tar.gz and extract the Resources folder to ensure maintaining timestamps

- **R / Python library not found:**

  It is typically caused by singularity loading local home and environment within the container, which may interfer with those in the container. Try adding options *–cleanenv –no-home* to the singularity command in **NeoDisc.sh** wrapper script:

```
singularity exec --cleanenv \
                 --no-home \
                 -B ${bindpath} \
                 ${containerpath}/NeoDisc.sif ${neodisc_arguments}
```

- **ERROR: GERMLINE WES/WGS ALIGNMENT FAILED:**

  - Check the path to the WES/WGS fastq files in the **configuration file**.
  - Check the formatting of the **WES/WGS fastq filenames** and make sure it is correct and fits with the -t and -g arguments provided.

- **ERROR: HLA-TYPING ANALYSIS FAILED:**

  - Check the path to the WES/WGS and RNAseq fastq files in the **configuration file**.
  - Check the formatting of the **WES/WGS and RNAseq fastq filenames** and make sure it is correct and fits with the -t, -g, and -rna arguments provided.
  - Make sure that you have **paired-end** sequencing WES/WGS and RNAseq files.

- **ERROR: <RNA sample> RNASEQ ANALYSIS FAILED:**

  - Check the path to the RNAseq fastq files in the **configuration file**.

- Check the formatting of the **RNAseq fastq filenames** and make sure it is correct and fits with the ***-rna argument(s)*** provided.

- **ERROR: MS SPECTRA CONVERSION FAILED:**
  - Check the path to the MS files in the **configuration file**.
  - Sometimes MS conversion may fail. Try **re-running** NeoDisc and see if the issue is solved.

- **Restarting a failed analysis:**
  - If you ran your analysis in **fastq** mode, use the ***-resume option***, see fastq Mode.

- **Rerun a specific module:**
  1. **Delete** the output **folder** of the **module**
  2. Use the ***-resume option***, see fastq Mode

Your problem is still not solved?

1. Please have a look at issues in our **github repository**
2. Open a ticket **github repository**

7 Softwares and databases used in NeoDisc

List of softwares included within the NeoDisc container:

Software	Version	Reference	License
bwa	0.7.17	Li H. and Durbin R. (2009)	GPL-3.0
GATK	4.4.0.0	McKenna et al. (2010)	Apache-2.0
Picard	3.0.0	Picard, GitHub Repository	MIT
FastQC	0.11.9	FastQC website	GPL-3.0 or later
Samtools	1.17	Danecek et al. (2021)	MIT/Expat
HTSlib	1.17	Bonfield et al. (2021)	MIT/Expat
bowtie2	2.5.1	Langmead B. and Salzberg S. (2012)	GPL-3.0
Sequenza	3.0.0	Favero F. et al. (2014)	GPL
WhatsHap	1.7	Martin M. et al. (2016)	MIT
MixMHCpred	2.2	Gfeller D. et al. (2023)	Academic
PRIME	2.0	Gfeller D. et al. (2023)	Academic
MixMHC2pred	2.0	Racle J. et al. (2019)	Academic
MoDec	1.2	Racle J. et al. (2019)	Academic
STAR	2.7.10b	Dobin A. et al. (2019)	MIT
RSeQC	5.0.1	Wang L. et al. (2012)	MIT
Subread	2.0.5	Liao Y. et al. (2013)	GPL-3.0*
bam-readcount	1.0.1	Khanna et al. (2022)	MIT
LICHeE	1.0	Popic V. et al. (2015)	MIT
ggtree	3.6.2	Yu G. et al. (2017)	Artistic-2.0
BTLib	0.20	Sourceforge repository	n.a.
CScape	n.a.	Rogers M.F et al. (2017)	n.a.
ProteoWizard	3.0.22132	Chambers M.C. et al. (2012)	Apache-2.0
Comet	2022.01	Eng J.K. et al. (2012)	Apache-2.0
NewAnce	1.7.5	Chong C. et al. (2020)	GPL-2.0 or later
FragPipe	20.0	Yu F. et al. (2023)	GPL-3.0
EasyPQP	0.1.35	EasyPQP, GitHub Repository	BSD-3.0
pyOpenMS	2.7.0	Röst H.L. et al. (2014)	BSD-3.0
Philosopher	5.0.0	da Veiga Leprevost F. et al. (2020)	GPL-3.0
PDV	1.8.0	Li K. et al. (2018)	GPL-3.0
fastq-pair	0.4 (modified)	John A. Edwards, Robert A. Edwards (2019)	MIT

** Included in tarball*

List of softwares NOT included within the NeoDisc container and which require user's download and installation:

Software	Version	Reference	License
HLA-HD	1.17	Kawaguchi S. et al. (2017)	Academic
Varscan	2.4.6	Koboldt DC. et al. (2012)	Academic
MSFragger	3.8	Kong A.T. et al. (2017)	Academic
IonQuant	1.9.8	Yu F. et al. (2021)	Academic
NetMHCpan	2.8a / 4.1b	Reynisson B. et al. (2020)	Academic
netchop	3.1d	Nielsen M. et al. (2005)	Academic
netMHCstabpan	1.0b	Rasmussen M. et al. (2016)	Academic
netCTLpan	1.1b	Stranzl T. et al. (2010)	Academic

List of databases used in NeoDisc:

Database	Version	Reference	License
Ensembl)	GRCh37.p13	Church D.M. et al. (2011)	Apache-2.0
GENCODE	43	Frankish A. et al. (2021)	Terms of use
NCBI Virus	n.a.	Hatcher E.L. et al. (2016)	n.a.
TCGA (open access data)	n.a.	TCGA Research Network	NCI copyright
GTEx (open access data)	7	GTEx Portal	License
Human Protein Atlas	n.a.	Thul P.J. et al. (2017)	CC BY-SA 3.0 DEED