MassPred 0.41

Goran "CHUPCKO" Lazić

Content

1. Description....2
  1.1. List of supported predictors....2
    1.1.1. Disorder predictors....2
    1.1.2. MHC binding predictors....3
    1.1.3. Hydropathy scale....3
    1.1.4. Disorder-binding predictor....3
2. Quick installation and test....4
3. Installation....5
  3.1. MassPred installation....5
  3.2. Predictors installation....5
    3.2.1. NR....6
    3.2.2. ANCHOR-1.0....6
    3.2.3. DisEMBL-1.4....7
    3.2.4. disopred-2.43....7
    3.2.5. IsUnstruct-2.02....7
    3.2.6. iupred-1.0....7
    3.2.7. netMHC-3.0c....8
    3.2.8. netMHC-3.4a....8
    3.2.9. netMHCII-2.2....8
    3.2.10. netMHCIIpan-1.0b....8
    3.2.11. netMHCIIpan-2.0b....9
    3.2.12. netMHCIIpan-3.0c....9
    3.2.13. netMHCIIpan-3.1a....9
    3.2.14. netMHCpan-2.0c....9
    3.2.15. netMHCpan-2.4a....9
    3.2.16. netMHCpan-2.8a....10
    3.2.17. OnD-1.0....10
    3.2.18. predisorder-1.1....10
    3.2.19. RONN-3.1....10
    3.2.20. VSL-2....11
4. Configuration....12
5. Running....16
6. Results....17
  6.1. SQL....17
  6.2. Command....20
7. Tests....21
8. Uninstall....22
9. Update....23
10. Possible problems....24

1. Description

MassPred is a command line (shell) oriented system which enables multiple (massive) and parallel execution of predictors with group FASTA files. Results are filtered and stored in a form suitable for loading into SQL. Also MassPred processing multiple proteins in the FASTA file. It can to execute any command on a set of proteins.

MassPred is a set of tools that provides easy predictor installation, application of predictors to input data and filtering of the results of predictor action. MassPred provides scripts for automated installation of the desired predictors and their preparation for automatic execution. Predictors (any of the previously mentioned four types) are applied to the protein dataset, which can be stored in one or more files or directories. Each file can include one or more proteins in FASTA format. MassPred takes the contents of input files or directories, extracts every single protein and applies the desired predictors to the extracted proteins, creating separate jobs for every pair (protein, predictor). The created jobs can be simultaneously executed on a symmetric multiprocessor computer, or on computers with a multicore/multithread processor architecture. MassPred itself does not perform a parallel execution of a single predictor application to a single protein.

MassPred collects the results and filters them in a CSV file format in order to prepare the results in a form that can be used as an input in a LOAD utility program for loading results in RDBMS tables. By default, the results are filtered for IBM DB2 RDBMS.

The MassPred mechanism for generating jobs for simultaneous execution is not restricted to supported predictors. In fact, any program that takes a single protein in FASTA format can be massively applied to a set of proteins.

1.1. List of supported predictors

Some predictors are obsolete, meaning that they are not available to download, but some required files can still download.

1.1.1. Disorder predictors

- VSL2b
  `http://www.ist.temple.edu/disprot/predictorVSL2.php`
- IUPred (Long, Short) version 1.0
  `http://iupred.enzim.hu/`
- DisEMBL (Coin coils, Hot loops, Remark465) version 1.4
  `http://dis.embl.de/`
- RONN version 3.1
  `http://www.bioinformatics.nl/~berndb/ronn.html`
- Predisorder version 1.1
  `http://sysbio.rnet.missouri.edu/multicom_toolbox/tools.html`
- IsUnstruct version 2.02
  `http://bioinfo.protres.ru/IsUnstruct/`
- Disopred version 2.43
  `http://bioinfadmin.cs.ucl.ac.uk/downloads/DISOPRED/`
- OnD-CRF version 1.0

http://babel.ucmp.umu.se/ond-crf/

1.1.2. MHC binding predictors

- NetMHCpan 2.0 version 2.0c (Obsolete)
  http://www.cbs.dtu.dk/services/NetMHCpan-2.0/
- NetMHCpan 2.4 version 2.4a (Obsolete)
  http://www.cbs.dtu.dk/services/NetMHCpan-2.4/
- NetMHCpan 2.8 version 2.8a
  http://www.cbs.dtu.dk/services/NetMHCpan-2.8/
- NetMHCIIpan 1.0 version 1.0b (Obsolete)
  http://www.cbs.dtu.dk/services/NetMHCIIpan-1.0/
- NetMHCIIpan 2.0 version 2.0b
  http://www.cbs.dtu.dk/services/NetMHCIIpan-2.0/
- NetMHCIIpan 3.0 version 3.0c (Obsolete)
  http://www.cbs.dtu.dk/services/NetMHCIIpan-3.0/
- NetMHCIIpan 3.1 version 3.1a
  http://www.cbs.dtu.dk/services/NetMHCIIpan-3.1/
- NetMHC 3.0 version 3.0c
  http://www.cbs.dtu.dk/services/NetMHC-3.0/
- NetMHC 3.4 version 3.4a
  http://www.cbs.dtu.dk/services/NetMHC-3.4/
- NetMHCII 2.2
  http://www.cbs.dtu.dk/services/NetMHCII-2.2/

1.1.3. Hydropathy scale

- Kyte Doolittle
  http://gcat.davidson.edu/rakarnik/KD.html
- Hopp Woods
  http://web.expasy.org/protscale/pscale/Hphob.Woods.html

1.1.4. Disorder-binding predictor

- ANCHOR version 1.0
  http://anchor.enzim.hu/

2. Quick installation and test

For quick installation and test MassPred, download **`masspred-x.xx.tar.gz`** and put into some directory. Enter in that directory and execute:

```
tar xzf masspred-x.xx.tar.gz
cd masspred
./work.sh test/configuration.ish test/cancer.faa
cd test/cancer.faa.out
ls -la
```

3. Installation

Package installation is done in two steps: MassPred installation and installation of predictors that are necessary for work.

3.1. MassPred installation

Unpack MassPred distribution file with command **`tar xzf masspred-x.xx.tar.gz`**. Name of directory which includes unpacked version should includes only letters, digits, minus, underline, point, or commas. Other symbols in the path may cause a problem in MassPred's work. Recommended directory for installation is **`/usr/local/masspred`**.

3.2. Predictors installation

First step in predictor installation process is install the tools necessary in for the installation process and operation of predictors. Tools includes specific version of Python and runtime C/C++ libraries. For Ubuntu linux (tested on version 15.10) use the following command:

```
sudo apt-get install build-essential gawk python-dev tcsh default-jre libstdc++5 libstdc++5:i386 libstdc++6:i386 libc6:i386 libncurses5:i386 lib32stdc++6 lib32ncurses5 lib32z1
```

MassPred directory after unpacking contains sub directory named **`predictors`**. This directory includes sub directory **`source`** with the following structure:

- **`clean_all.sh`** – shell script which removes all installed predictors;
- **`clean_current.sh`** – shell script which removes all currents installed predictors;
- **`clean_obsolete.sh`** – shell script which removes all obsolete installed predictors;
- **`data`** – directory with source of predictors, downloaded from predictors URL;
- **`data.md5`** – MD5 sum of predictors sources files;
- **`download`** – directory with shell scripts for automatic download (it is available) of predictor source file;
- **`download_all.sh`** – shell script which downloads all available predictors sources files;
- **`download_current.sh`** – shell script which downloads currents available predictors sources files;
- **`download_obsolete.sh`** – shell script which downloads obsolete available predictors sources files;
- **`install`** – directory with shell scripts for installing predictors;
- **`install_all.sh`** – shell script which installs all predictors;
- **`install_current.sh`** – shell script which installs currents predictors;
- **`install_obsolete.sh`** – shell script which installs obsolete predictors;
- **`md5sum_data.sh`** – shell script which calculates MD5 sums of predictors sources files, output must be the same as **`data.md5`**;
- **`patch`** – directory with patch files for some predictors;
- **`test`** – directory with shell scripts for testing predictors;

- **`test_all.sh`** – shell script which tests all predictors
- **`test_current.sh`** – shell script which tests currents predictors
- **`test_obsolete.sh`** – shell script which tests obsolete predictors

Predictors sources files are not included because of the licenses, but some predictors or some of their components are on permanent URL (listed above in list of supported predictors) which is used by download scripts.

Instruction for the predictors installation are:

- Open terminal window and go to the **`predictors`** directory;
- Execute **`source/download_all.sh`** to download all available predictors;
- If some of predictors requires additional files, downlad it as described in its own installation requirements;
- Optionally execute **`source/md5sum_data.sh > source/data_new.md5`** to calculate MD5 sums;
- Optionally compare **`source/data.md5`** and **`source/data_new.md5`**, generally they should be equal;
- Execute **`source/install_all.sh`** to install all predictors;
- Optionally execute **`source/test_all.sh`** to test all predictors;

The specific information about download individual predictors and their test after installation are listed in the rest of this chapter.

3.2.1. NR

For download execute **`source/download/nr.sh`**

File list with appropriate MD5 sum:

```
1572cc751fdd583be37daff6006abcb8  data/nr/nr.00.tar.gz
e720992d2bd11b94039bcf02df5434c4  data/nr/nr.01.tar.gz
0a74580abaf821f598d1d20aaa6a3484  data/nr/nr.02.tar.gz
2e6ea9ec585f98db5ec219a1ca680c60  data/nr/nr.03.tar.gz
684e769b7d1bbebddfbac85aecff249b  data/nr/nr.04.tar.gz
c9b3919d15c6ab52d035e9088af08018  data/nr/nr.05.tar.gz
3ce6adb2b55f23a08e5380b2a3bc1bbe  data/nr/nr.06.tar.gz
1eec6dd9f30ac9f3627b44e8f2c2ea69  data/nr/nr.07.tar.gz
3d8818ed159efed790d2a8f49aee4196  data/nr/nr.08.tar.gz
98f165bb761bcb7f960493429811ae16  data/nr/nr.09.tar.gz
8b04abebe763115fc40c11f7b153a356  data/nr/nr.10.tar.gz
```

For installation execute **`source/install/nr.sh`**
Remark: predictors disopred and predisorder use NR database. It is also possible that some MD5 sums changed.

3.2.2. ANCHOR-1.0

For download in directory **`source/data/anchor-1.0`** download file:

`ANCHOR.tar.gz` from URL:

http://anchor.enzim.hu/Downloads.php

File list with appropriate MD5 sum:

```
e8625a57d9eaa5d112a779c3ee5168a5  data/anchor-1.0/ANCHOR.tar.gz
```

For installation execute **source/install/anchor-1.0.sh**
For test execute **source/test/anchor-1.0.sh**

3.2.3. DisEMBL-1.4

For download execute **source/download/disembl-1.4.sh**

File list with appropriate MD5 sum:

```
9174d0371b23fc026b2b76e62197c37c  data/disembl-1.4/DisEMBL-1.4.tgz
733e585a5125b272618e7e4f98580c5d  data/disembl-1.4/TISEAN_3.0.1.tar.gz
8539f1761483187a04da9bf7f499a21f  data/disembl-1.4/biopython-1.60.tar.gz
0ab72b3b83528a7ae79c6df9042d61c6  data/disembl-1.4/numpy-1.7.1.tar.gz
```

For installation execute **source/install/disembl-1.4.sh**
For test execute **source/test/disembl-1.4.sh**

3.2.4. disopred-2.43

For download execute **source/download/disopred-2.43.sh**

File list with appropriate MD5 sum:

```
875be33b3b4a7f3a3612843bed80545f  data/disopred-2.43/blast-2.2.26-ia32-linux.tar.gz
809798a912f4fb37f62406201456df67  data/disopred-2.43/blast-2.2.26-x64-linux.tar.gz
d082af598bc6160d67e183fc25d7f057  data/disopred-2.43/disopred2.43.tar.gz
```

For installation execute **source/install/disopred-2.43.sh**
For test execute **source/test/disopred-2.43.sh**

3.2.5. IsUnstruct-2.02

For download execute **source/download/isunstruct-2.02**

File list with appropriate MD5 sum:

```
f1acc1b6c55601f0070326e22cb9cee1  data/isunstruct-2.02/IsUnstruct_2.02.tar.gz
```

For installation execute **source/install/isunstruct-2.02.sh**
For test execute **source/test/isunstruct-2.02.sh**

3.2.6. iupred-1.0

For download in directory **source/data/iupred-1.0** download file:

iupred.tar.gz from URL:

http://iupred.enzim.hu/Downloads.php

File list with appropriate MD5 sum:

```
4cfe70b929f2b6e46c321b385adb292f  data/iupred-1.0/iupred.tar.gz
```

For installation execute **source/install/iupred-1.0.sh**
For test execute **source/test/iupred-1.0.sh**

3.2.7. netMHC-3.0c

For download in directory **`source/data/netmhc-3.0c`** download file:

`netMHC-3.0c.Linux.tar.Z` from URL:

`http://www.cbs.dtu.dk/cgi-bin/sw_request?netMHC+3.0`

File list with appropriate MD5 sum:

```
9952dd7aec7f0471e490b36ee3697734  data/netmhc-3.0c/netMHC-3.0c.Linux.tar.Z
```

For installation execute **`source/install/netmhc-3.0c.sh`**
For test execute **`source/test/netmhc-3.0c.sh`**

3.2.8. netMHC-3.4a

For download execute **`source/download/netmhc-3.4a.sh`**

Then in directory **`source/data/netmhc-3.4a`** download file:

`netMHC-3.4a.Linux.tar.gz` from URL:

`http://www.cbs.dtu.dk/cgi-bin/sw_request?netMHC`

File list with appropriate MD5 sum:

```
c77bcc0216cddf1be671ecd69f213bed  data/netmhc-3.4a/net.tar.gz
c76e6c042e4f4b3f57dbe970552260a6  data/netmhc-3.4a/netMHC-3.4a.Linux.tar.gz
```

For installation execute **`source/install/netmhc-3.4a.sh`**
For test execute **`source/test/netmhc-3.4a.sh`**

3.2.9. netMHCII-2.2

For download execute **`source/download/netmhcii-2.2.sh`**

Then in directory **`source/data/netmhcii-2.2`** download file:

`netMHCII-2.2.Linux.tar.Z` from URL:

`http://www.cbs.dtu.dk/cgi-bin/sw_request?netMHCII`

File list with appropriate MD5 sum:

```
11579b61d3bfe13311f7b42fc93b4dd8  data/netmhcii-2.2/data.tar.gz
918b7108a37599887b0725623d0974e6  data/netmhcii-2.2/netMHCII-2.2.Linux.tar.Z
```

For installation execute **`source/install/netmhcii-2.2.sh`**
For test execute **`source/test/netmhcii-2.2.sh`**

3.2.10. netMHCIIpan-1.0b

This version is obsolete.

File list with appropriate MD5 sum:

```
eac6371f3eb612827377a9049e8c2e21  data/netmhciipan-1.0b/netMHCIIpan-1.0b.Linux.tar.Z
```

For instalation execute **`source/install/netmhciipan-1.0b.sh`**
For test execute **`source/test/netmhciipan-1.0b.sh`**

3.2.11. netMHCIIpan-2.0b

For download execute **`source/download/netmhciipan-2.0b.sh`**

Then in directory **`source/data/netmhciipan-2.0b`** download file:

`netMHCIIpan-2.0b.Linux.tar.Z` from URL:

`http://www.cbs.dtu.dk/cgi-bin/sw_request?netMHCIIpan+2.0`

File list with appropriate MD5 sum:

```
42416a2974ef78650fe016461041ecd3  data/netmhciipan-2.0b/data.tar.gz
6c245f991f0a169ea8b0b8dc98d3b241  data/netmhciipan-2.0b/netMHCIIpan-2.0b.Linux.tar.Z
```

For installation execute **`source/install/netmhciipan-2.0b.sh`**
For test execute **`source/test/netmhciipan-2.0b.sh`**

3.2.12. netMHCIIpan-3.0c

This version is obsolete.

File list with appropriate MD5 sum:

```
9ebbed9fa7648cb28313ebe67a27c199  data/netmhciipan-3.0c/data.tar.gz
da25b29413770565cb9aa1886538e7cd  data/netmhciipan-3.0c/netMHCIIpan-3.0c.Linux.tar.gz
```

For installation execute **`source/install/netmhciipan-3.0c.sh`**
For test execute **`source/test/netmhciipan-3.0c.sh`**

3.2.13. netMHCIIpan-3.1a

For download execute **`source/download/netmhciipan-3.1a.sh`**

Then in directory **`source/data/netmhciipan-3.1a`** download file:

`netMHCIIpan-3.1a.Linux.tar.gz` from URL:

`http://www.cbs.dtu.dk/cgi-bin/sw_request?netMHCIIpan`

File list with appropriate MD5 sum:

```
f833df245378e60ca6e55748344a36f6  data/netmhciipan-3.1a/data.tar.gz
0962ce799f7a4c9631f8566a55237073  data/netmhciipan-3.1a/netMHCIIpan-3.1a.Linux.tar.gz
```

For installation execute **`source/install/netmhciipan-3.1a.sh`**
For test execute **`source/test/netmhciipan-3.1a.sh`**

3.2.14. netMHCpan-2.0c

This version is obsolete.

File list with appropriate MD5 sum:

```
227ac3b6b0c432253d22251d1dcbbf1c  data/netmhcpan-2.0c/netMHCpan-2.0c.Linux.tar.Z
```

For installation execute **`source/install/netmhcpan-2.0c.sh`**
For test execute **`source/test/netmhcpan-2.0c.sh`**

3.2.15. netMHCpan-2.4a

This version is obsolete.

File list with appropriate MD5 sum:

```
9a9a5063d2c1e4738e829eb408e8a980  data/netmhcpan-2.4a/data.tar.gz
9d6f860009f12b0246a1599c77289697  data/netmhcpan-2.4a/netMHCpan-2.4a.Linux.tar.Z
```

For installation execute **`source/install/netmhcpan-2.4a.sh`**
For test execute **`source/test/netmhcpan-2.4a.sh`**

3.2.16. netMHCpan-2.8a

For download execute **`source/download/netmhcpan-2.8a.sh`**

Then in directory **`source/data/netmhcpan-2.8a`** download file:

`netMHCpan-2.8a.Linux.tar.gz` from URL:

`http://www.cbs.dtu.dk/cgi-bin/sw_request?netMHCpan`

File list with appropriate MD5 sum:

```
32780a965561a77bdb2fb46db6829d0c  data/netmhcpan-2.8a/data.tar.gz
f6597248e77adcdf626cb1e129c6ba79  data/netmhcpan-2.8a/netMHCpan-2.8a.Linux.tar.gz
```

For installation execute **`source/install/netmhcpan-2.8a.sh`**
For test execute **`source/test/netmhcpan-2.8a.sh`**

3.2.17. OnD-1.0

For download execute **`source/download/ond-1.0.sh`**

File list with appropriate MD5 sum:

```
255fae52b8362c14eb86118849d098c1  data/ond-1.0/OnD.tar.gz
```

For installation execute **`source/install/ond-1.0.sh`**
For test execute **`source/test/ond-1.0.sh`**

3.2.18. predisorder-1.1

For download execute **`source/download/predisorder-1.1.sh`**

File list with appropriate MD5 sum:

```
f824ddfd6f098f2a25ae740ee037c385  data/predisorder-1.1/blast-2.2.17-ia32-linux.tar.gz
dec7bccb1800b622ce410324e7a2aa93  data/predisorder-1.1/predisorder1.1.tar.gz
1613b0edb5b5477c5f132e4687b8f4bb  data/predisorder-1.1/sspro4.1.tar.gz
```

For installation execute **`source/install/predisorder-1.1.sh`**
For test execute **`source/test/predisorder-1.1.sh`**

3.2.19. RONN-3.1

For download in directory **`source/data/ronn-3.1`** download file:

`RONNv3_1.tar.gz` from URL:

`http://www.strubi.ox.ac.uk/RONN/`

File list with appropriate MD5 sum:

```
54cdfa97cdf087b667accac00719091a  data/ronn-3.1/RONNv3_1.tar.gz
```

For installation execute **`source/install/ronn-3.1.sh`**
For test execute **`source/test/ronn-3.1.sh`**

3.2.20. VSL-2

For download execute **`source/download/vsl-2.sh`**

File list with appropriate MD5 sum:

```
04a34314afab1f5fc3acd4b83f7295eb  data/vsl-2/VSL2.tar.gz
```

For installation execute **`source/install/vsl-2.sh`**
For test execute **`source/test/vsl-2.sh`**

4. Configuration

Before running MassPred, it is necessary to make a configuration file with lines in the following form:

`variable=value`

Comments are from **#** to the end of the line.
File **`configuration.ish`** in installed directory is an example of configuration file.

Variables are:

- **`CPU_NUMBER`**
  Specify maximal number of simultaneously executed jobs. Parameter **`value`** is number, with default **4**.
- **`WORK_SQL`**
  Specify if SQL files with insert statements should be generated. Parameter **`value`** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **`WORK_NUMERIC`**
  Specify generation of alternative form results with numeric value for each AA. Parameter **`value`** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **`WORK_COMMAND`**
  Specify execution of some command for each protein. Parameter **`value`** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **`COMMAND`**
  This variable has meaning only if **`WORK_COMMAND`** is set. Specify command which executes for each protein. Parameter **`value`** is string with default value "-" (when report error, because "-"is nonexistent command).
- **`WORK_HYDRO`**
  Specify generation of table **`hydro`** for each AA in protein. Parameter **`value`** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **`WORK_ANCHOR`**
  Specify execution of ANCHOR predictor. Parameter **`value`** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **`WORK_DISEMBL`**
  Specify execution of DisEMBL predictor. Parameter **`value`** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **`WORK_DISOPRED`**
  Specify execution of disopred predictor. Parameter **`value`** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **`WORK_ISUNSTRUCT`**
  Specify execution of IsUnstruct predictor. Parameter **`value`** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **`WORK_IUPRED_LONG`**
  Specify execution of iupred predictor, long variant. Parameter **`value`** is string with default value "**no**". Possible values are "**yes**" or "**no**".

- **WORK_IUPRED_SHORT**
  Specify execution of iupred predictor, short variant. Parameter **value** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **WORK_OND**
  Specify execution of OnD predictor. Parameter **value** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **WORK_PREDISORDER**
  Specify execution of predisorder predictor. Parameter **value** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **WORK_RONN**
  Specify execution of RONN predictor. Parameter **value** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **WORK_VSL2**
  Specify execution of VSL2 predictor. Parameter **value** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **WORK_NETMHC_1_30C**
  Specify execution of netMHC predictor, version 3.0c. Parameter **value** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **NETMHC_1_30C_ALLELE_FILE**
  This variable has meaning only if **WORK_NETMHC_1_30C** is set. Specify file with allele. Path is specified absolute or relative with regard to configuration file. Parameter **value** is string with default value "**NetMhc.3.0c.pseudo**".
- **NETMHC_1_30C_LENGTH_FROM**
  This variable has meaning only if **WORK_NETMHC_1_30C** is set. Specify from which length of peptide predictor works. Parameter **value** is number, with default **8**.
- **NETMHC_1_30C_LENGTH_TO**
  This variable has meaning only if **WORK_NETMHC_1_30C** is set. Specify to which length of peptide predictor works. Parameter **value** is number, with default **11**.
- **WORK_NETMHC_1_34A**
  Specify execution of netMHC predictor, version 3.4a. Parameter **value** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **NETMHC_1_34A_ALLELE_FILE**
  This variable has meaning only if **WORK_NETMHC_1_34A** is set. Specify file with allele. Path is specified absolute or relative with regard to configuration file. Parameter **value** is string with default value "**NetMhc.3.4a.pseudo**".
- **NETMHC_1_34A_LENGTH_FROM**
  This variable has meaning only if **WORK_NETMHC_1_34A** is set. Specify from which length of peptide predictor works. Parameter **value** is number, with default **8**.
- **NETMHC_1_34A_LENGTH_TO**
  This variable has meaning only if **WORK_NETMHC_1_34A** is set. Specify to which length of peptide predictor works. Parameter **value** is number, with default **11**.
- **WORK_NETMHC_2_22**
  Specify execution of netMHCII predictor, version 2.2. Parameter **value** is string with default value "**no**". Possible values are "**yes**" or "**no**".

- **NETMHC_2_22_ALLELE_FILE**
  This variable has meaning only if **WORK_NETMHC_2_22** is set. Specify file with allele. Path is specified absolute or relative with regard to configuration file. Parameter **value** is string with default value "**NetMhcII.2.2.pseudo**".
- **NETMHC_2_22_LENGTH_FROM**
  This variable has meaning only if **WORK_NETMHC_2_22** is set. Specify from which length of peptide predictor works. Parameter **value** is number, with default **9**.
- **NETMHC_2_22_LENGTH_TO**
  This variable has meaning only if **WORK_NETMHC_2_22** is set. Specify to which length of peptide predictor works. Parameter **value** is number, with default **15**.
- **WORK_NETMHC_PAN_1_20C**
  Specify execution of netMHCpan predictor, version 2.0c. Parameter **value** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **NETMHCPAN_1_20C_ALLELE_FILE**
  This variable has meaning only if **WORK_NETMHC_PAN_1_20C** is set. Specify file with allele. Path is specified absolute or relative with regard to configuration file. Parameter **value** is string with default value "**NetMhcPan.2.0c.pseudo**".
- **NETMHCPAN_1_20C_LENGTH_FROM**
  This variable has meaning only if **WORK_NETMHC_PAN_1_20C** is set. Specify from which length of peptide predictor works. Parameter **value** is number, with default **8**.
- **NETMHCPAN_1_20C_LENGTH_TO**
  This variable has meaning only if **WORK_NETMHC_PAN_1_20C** is set. Specify to which length of peptide predictor works. Parameter **value** is number, with default **11**.
- **WORK_NETMHCPAN_1_24A**
  Specify execution of netMHCpan predictor, version 2.4a. Parameter **value** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **NETMHCPAN_1_24A_ALLELE_FILE**
  This variable has meaning only if **WORK_NETMHCPAN_1_24A** is set. Specify file with allele. Path is specified absolute or relative with regard to configuration file. Parameter **value** is string with default value "**NetMhcPan.2.4a.pseudo**".
- **NETMHCPAN_1_24A_LENGTH_FROM**
  This variable has meaning only if **WORK_NETMHCPAN_1_24A** is set. Specify from which length of peptide predictor works. Parameter **value** is number, with default **8**.
- **NETMHCPAN_1_24A_LENGTH_TO**
  This variable has meaning only if **WORK_NETMHCPAN_1_24A** is set. Specify to which length of peptide predictor works. Parameter **value** is number, with default **11**.
- **WORK_NETMHCPAN_1_28A**
  Specify execution of netMHCpan predictor, version 2.8a. Parameter **value** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **NETMHCPAN_1_28A_ALLELE_FILE**
  This variable has meaning only if **WORK_NETMHCPAN_1_28A** is set. Specify file with allele. Path is specified absolute or relative with regard to configuration file. Parameter **value** is string with default value "**NetMhcPan.2.8a.pseudo**".
- **NETMHCPAN_1_28A_LENGTH_FROM**

This variable has meaning only if **WORK_NETMHCPAN_1_28A** is set. Specify from which length of peptide predictor works. Parameter **value** is number, with default **8**.

- **NETMHCPAN_1_28A_LENGTH_TO**
  This variable has meaning only if **WORK_NETMHCPAN_1_28A** is set. Specify to which length of peptide predictor works. Parameter **value** is number, with default **11**.
- **WORK_NETMHCPAN_2_10B**
  Specify execution of netMHCIIpan predictor, version 1.0b. Parameter **value** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **NETMHCPAN_2_10B_ALLELE_FILE**
  This variable has meaning only if **WORK_NETMHCPAN_2_10B** is set. Specify file with allele. Path is specified absolute or relative with regard to configuration file. Parameter **value** is string with default value "**NetMhcIIPan.1.0b.pseudo**".
- **NETMHCPAN_2_10B_LENGTH_FROM**
  This variable has meaning only if **WORK_NETMHCPAN_2_10B** is set. Specify from which length of peptide predictor works. Parameter **value** is number, with default **9**.
- **NETMHCPAN_2_10B_LENGTH_TO**
  This variable has meaning only if **WORK_NETMHCPAN_2_10B** is set. Specify to which length of peptide predictor works. Parameter **value** is number, with default **15**.
- **WORK_NETMHCPAN_2_20B**
  Specify execution of netMHCIIpan predictor, version 2.0b. Parameter **value** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **NETMHCPAN_2_20B_ALLELE_FILE**
  This variable has meaning only if **WORK_NETMHCPAN_2_20B** is set. Specify file with allele. Path is specified absolute or relative with regard to configuration file. Parameter **value** is string with default value "**NetMhcIIPan.2.0b.pseudo**".
- **NETMHCPAN_2_20B_LENGTH_FROM**
  This variable has meaning only if **WORK_NETMHCPAN_2_20B** is set. Specify from which length of peptide predictor works. Parameter **value** is number, with default **9**.
- **NETMHCPAN_2_20B_LENGTH_TO**
  This variable has meaning only if **WORK_NETMHCPAN_2_20B** is set. Specify to which length of peptide predictor works. Parameter **value** is number, with default **15**.
- **WORK_NETMHCPAN_2_30C**
  Specify execution of netMHCIIpan predictor, version 3.0c. Parameter **value** is string with default value "**no**". Possible values are "**yes**" or "**no**".
- **NETMHCPAN_2_30C_ALLELE_FILE**
  This variable has meaning only if **WORK_NETMHCPAN_2_30C** is set. Specify file with allele. Path is specified absolute or relative with regard to configuration file. Parameter **value** is string with default value "**NetMhcIIPan.3.0c.pseudo**".
- **NETMHCPAN_2_30C_LENGTH_FROM**
  This variable has meaning only if **WORK_NETMHCPAN_2_30C** is set. Specify from which length of peptide predictor works. Parameter **value** is number, with default **9**.
- **NETMHCPAN_2_30C_LENGTH_TO**
  This variable has meaning only if **WORK_NETMHCPAN_2_30C** is set. Specify to which length of peptide predictor works. Parameter **value** is number, with default **15**.

- **`WORK_NETMHCPAN_2_31A`**
  Specify execution of netMHCIIpan predictor, version 3.1a. Parameter **`value`** is string with default value "**`no`**". Possible values are "**`yes`**" or "**`no`**".
- **`NETMHCPAN_2_31A_ALLELE_FILE`**
  This variable has meaning only if **`WORK_NETMHCPAN_2_31A`** is set. Specify file with allele. Path is specified absolute or relative with regard to configuration file. Parameter **`value`** is string with default value "**`NetMhcIIPan.3.1a.pseudo`**".
- **`NETMHCPAN_2_31A_LENGTH_FROM`**
  This variable has meaning only if **`WORK_NETMHCPAN_2_31A`** is set. Specify from which length of peptide predictor works. Parameter **`value`** is number, with default **`9`**.
- **`NETMHCPAN_2_31A_LENGTH_TO`**
  This variable has meaning only if **`WORK_NETMHCPAN_2_31A`** is set. Specify to which length of peptide predictor works. Parameter **`value`** is number, with default **`15`**.

5. Running

MassPred can be started with command in installed directory:

```
./work.sh <configuration_file> <input>
```

Input is a directory with FASTA files or a single FASTA file with multiple FASTA formats..

For each protein in each FASTA file, MassPred execute required predictors and/or command if the command is given.

6. Results

After MassPred's successful ending, the result will be found in the output directory that is printed at the end of execution.
The Name of the output directory is calculated as the name of the input with added "**`.out`**". If command is specified, in the output directory will be found the sub directory **`command`** with the results of the execution command for each protein. Other files are the results of predictors.

6.1. SQL

The results of the predictors can be found in the output directory. Depending on which predictors are applied, the appropriate file with the results is generated. The files are gziped. The process always generate the LOAD files, while generating the SQL files is optional. The FAIL files appear if the error occurred during the execution for some predictors (the most common example is execution of VSL2b predictor which can not be applied on proteins with letters denotes ambiguous AAs).

Possible files in the output directory:

- **`epitope_fail.load.gz`**
  gziped LOAD file with list of netMHC failed predictors. This file is generated if the error occurred in the execution of netMHC predictors.
- **`epitope_fail.sql.gz`**
  gziped SQL file with list of netMHC failed predictors. This file is generated if the error occurred in the execution of netMHC predictors.
- **`region_fail.load.gz`**
  gziped LOAD file with list of disorder failed predictors. This file is generated if the error occurred in the execution of disorder predictors.
- **`region_fail.sql.gz`**
  gziped SQL file with list of disorder failed predictors. This file is generated if the error occurred in the execution of disorder predictors.
- **`hydro.load.gz`**
  gziped LOAD file with the results of hydrophobia.
- **`hydro.sql.gz`**
  gziped SQL file with the results of hydrophobia.
- **`epitope_success.load.gz`**
  gziped LOAD file with the results of netMHC predictors.
- **`epitope_success.sql.gz`**
  gziped SQL file with the results of netMHC predictors.
- **`epitope_success_numeric.load.gz`**
  gziped LOAD file with numeric results of netMHC predictors. This file is generated if **`WORK_NUMERIC`** option is given.
- **`epitope_success_numeric.sql.gz`**
  gziped SQL file with numeric results of netMHC predictors. This file is generated if **`WORK_NUMERIC`** option is given.
- **`region_success.load.gz`**
  gziped LOAD file with the results of disorder predictors.
- **`region_success.sql.gz`**

gziped SQL file with the results of disorder predictors.

- **`region_success_numeric.load.gz`**
  gziped LOAD file with numeric results of disorder predictors. This file is generated if **`WORK_NUMERIC`** option is given.
- **`region_success_numeric.sql.gz`**
  gziped SQL file with numeric results of disorder predictors. This file is generated if **`WORK_NUMERIC`** option is given.

MassPred collects the results and filters them in a CSV file format in order to prepare the results in a form that can be used as an input in a *load* utility program for loading results in RDBMS tables. By default, the results are filtered for IBM DB2 RDBMS
Load files includes data in CSV like format that can be used as an input in a LOAD utility program for loading results in RDBMS tables. Structure of tables that can be used are:

```
epitope_fail
(
  protein_id         VARCHAR(64),
  protein_reference  VARCHAR(64),
  protein_file_name  VARCHAR(64),
  type               VARCHAR(32),
  allele             VARCHAR(32),
  length             INTEGER
)

region_fail
(
  protein_id         VARCHAR(64),
  protein_reference  VARCHAR(64),
  protein_file_name  VARCHAR(64),
  type               VARCHAR(32)
)

hydro
(
  protein_id         VARCHAR(64),
  protein_reference  VARCHAR(64),
  protein_file_name  VARCHAR(64),
  position           INTEGER,
  aa                 CHAR(1),
  hydro_kd           DECIMAL,
  hydro_hw           DECIMAL
)

epitope
(
  protein_id         VARCHAR(64),
  protein_reference  VARCHAR(64),
  protein_file_name  VARCHAR(64),
  position           INTEGER,
```

```
  epitope           VARCHAR(32),
  pos               INTEGER,
  core              VARCHAR(32),
  aff_log           DECIMAL,
  aff               DECIMAL,
  rank              DECIMAL,
  binding           CHAR(2),
  type              VARCHAR(32),
  allele            VARCHAR(32),
  length            INTEGER
)

epitope_numeric
(
  protein_id        VARCHAR(64),
  protein_reference VARCHAR(64),
  protein_file_name VARCHAR(64),
  position          INTEGER,
  epitope           VARCHAR(32),
  pos               INTEGER,
  core              VARCHAR(32),
  aff_log           DECIMAL,
  aff               DECIMAL,
  rank              DECIMAL,
  binding           CHAR(2),
  type              VARCHAR(32),
  allele            VARCHAR(32),
  length            INTEGER
)

region
(
  protein_id        VARCHAR(64),
  protein_reference VARCHAR(64),
  protein_file_name VARCHAR(64),
  begin             INTEGER,
  end               INTEGER,
  order             CHAR(1),
  type              VARCHAR(32)
)

region_numeric
(
  protein_id        VARCHAR(64),
  protein_reference VARCHAR(64),
  protein_file_name VARCHAR(64),
  position          INTEGER,
  aa                CHAR(1),
  value             DECIMAL,
  order             CHAR(1),
```

```
  type                 VARCHAR(32)
)
```

6.2. Command

If the option **WORK_COMMAND** is set, then in directory **command**, for each protein from input FASTA files, can be found three files with file name in form:
<name>.<number>.<suffix>

Name is the name of input file.
Number is the position of the protein in FASTA file.
Suffix is:

- "**rc**" - file with result code of executed command,
- "**out**" - file with standard output of executed command,
- "**err**" - file with standard error output of executed command.

The command gets in standard input content of FASTA file (which refers to the specific protein in the order).
Before the execution of command, the next environment variables are set:

- **MASSPRED_INPUT** - name of input FASTA file,
- **MASSPRED_INPUT_POSITION** - position of protein in input FASTA file,
- **MASSPRED_OUTPUT** - prefix of file name output files,
- **MASSPRED_FASTA_DB** - first field from header of input FASTA file,
- **MASSPRED_FASTA_ID** - second field from header of input FASTA file,
- **MASSPRED_FASTA_REFERENCE_DB** - third field from header of input FASTA file,
- **MASSPRED_FASTA_REFERENCE** - fourth field from header of input FASTA file,
- **MASSPRED_FILE_NAME** - base part of input FASTA file name.

7. Tests

Calling MassPeed is very simple. On the command line from directory where MassPred is unpacked (for example in directory **`/usr/local/masspred`**), just execute **`work.sh`** script:

```
./work.sh my_directory/configuration.ish my_directory/my_file.faa
```

Where **`configuration.ish`** is configuration file described in chapter 3, and **`my_file.faa`** is file with FASTA format of proteins. For example, these commands can be used for test of successful installation of MassPred:

```
./work.sh test/configuration.ish test/cancer.faa
./work.sh test/configuration.ish test/iedb_9014-9017.faa
./work.sh test/configuration.ish test/iedb_9000.faa
./work.sh test/configuration.ish test/dir
./work.sh test/configuration.ish test/p03211.faa
```

8. Uninstall

For uninstalling MassPred just delete the directory in which MassPred is installed.

9. Update

The longer and more secure way is to uninstall MassPred first and then install it again.

The shorter way allows retain installed predictors. First all installed predictors from the directories **`predictors`** move to a temporary location, then delete the directory containing the MassPred, after that unpack again MassPred (in the same directory) and replace directory **`predictors`** with the old version (from the temporary location).

It is important that the complete path to the predictors directory is not changed. In order to ensure that, it is essential to always install the MassPred in the same directory (for example **`/usr/local/masspred`**).

10. Possible problems

In case of error in execution, or interruption it is necessary to remove all temporary directories, which have form:
`/tmp/masspred-dir-<number>-<number>`.

When the execution is normal, the MassPred clean all the temporary directories, but in case breaking of execution, it is possible that some temporary directories will not be properly removed.