
TTT Proteotyping Pipeline Documentation

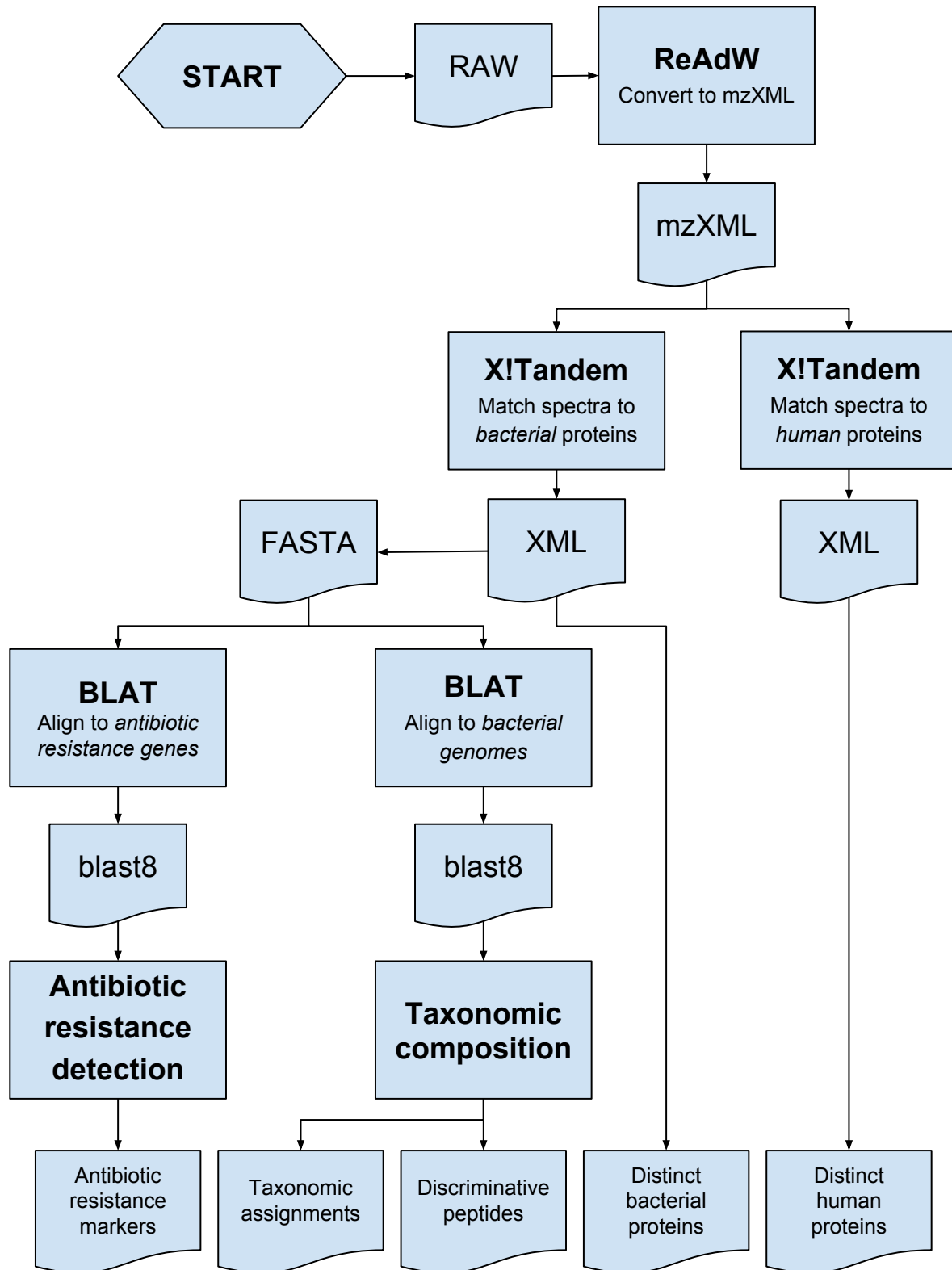
Release 1.0

Fredrik Boulund

Apr 25, 2017

Contents

1	Installation	3
1.1	Download the code	3
1.2	ReAdW	3
1.3	X!Tandem	5
2	Running	7
2.1	Work directory	7
2.2	Run the Snakemake workflow	7
2.3	Automatic invocation via crontab	8
3	Indices and tables	9



Contents:

CHAPTER 1

Installation

TTT Proteotyping Pipeline consists of several steps:

- raw to mzXML conversion with ReAdW.
- peptide identification with X!Tandem.
- X!Tandem xml output to FASTA conversion with `convert_tandem_xml_2_fasta.py`.
- Peptide to reference database alignment with BLAT.
- Antibiotic resistance detection with Proteotyping.
- Taxonomic composition estimation with Proteotyping.
- Listing distinct/unique proteins in X!Tandem output with `create_unique_protein_list.py`.

ReAdW, X!Tandem, and BLAT are external programs. This documentation will give brief instructions on how to install and use them together with TTT Proteotyping Pipeline. The Python programs are a part of this Python package, available in the `TTT_proteotyping_pipeline` folder in the repository.

The steps can be run manually, or preferably via the included Snakemake script.

Download the code

To download the code, clone the repository:

```
$ hg clone https://bitbucket.org/chalmersmathbioinformatics/TTT_proteotyping_pipeline
```

This will clone the entire repository to a folder called *TTT_proteotyping_pipeline* in your current directory.

ReAdW

ReAdW is meant to be run under Windows, but can be run under Linux using Wine, see instructions below. Running ReAdW in Wine requires a Linux system with a working 32-bit Wine installation.

Note: It is important that you use 32-bit Wine, as ReAdW cannot be run under 64-bit Wine. As of this writing, 32-bit Wine is only available for RedHat Enterprise Linux 6 and below. Support for 32-bit Wine was removed in RHEL 7.

Get ReAdW

ReAdW can be downloaded from the [ReAdW Github repository](#). Either clone the entire repository or download the binary suitable for your system. Note the information about the dependencies on three Windows DLL files: XRawfile2.dll, fileio.dll, fregistry.dll. These files are **NOT** supplied with this pipeline.

Create a 32-bit Wine prefix

1. Install Wine. It is important that a 32-bit version of Wine is installed, this normally means packages named `<package>.i686` instead of `<package>.x86_64`. In RHEL/CentOS it can be installed like this:

```
yum install wine
```

2. Create a Win32 prefix from which to run ReAdW. Make sure to set and export `WINEARCH=win32` during the creation of the wine prefix. Modify the command below to a path of your choice. Note that this step likely requires working X11-forwarding:

```
export WINEARCH=win32
export WINEPREFIX=/path/to/your/desired/wineprefix
winecfg
```

Click OK in any configuration windows that pop up.

3. Download *winetricks* to install the required Visual Studio C++ runtimes. `vcrun2010` is required for ReAdW and `vcrun2008` is required for the Thermo DLL's. Again, note that this step requires X11-forwarding to be enabled:

```
wget https://raw.githubusercontent.com/Winetricks/winetricks/master/src/winetricks
sh winetricks vcrun2010 vcrun2008
```

Click through any installation prompts that pop up, and after they complete, finish by registering XRawfile2.dll in your wine prefix:

```
wine regsvr32 XRawfile2.dll
```

Running ReAdW

Make sure to set the `WINEPREFIX` environment variable to the correct path (same directory you specified when creating the 32 bit wine prefix), then run ReAdW from your Linux command prompt via Wine:

```
export WINEPREFIX=/path/to/your/desired/wineprefix
wine /path/to/ReAdW.201510.xcalibur.exe [options] /path/to/sample.raw
```

Now it should run.

X!Tandem

Download and install X!Tandem according to the instructions on the [X!Tandem homepage](#). To run a sample in X!Tandem, several xml-files must be prepared. There is a Python program in TTT Proteotyping Pipeline called `run_xtandem.py` that will automatically create the required input files and run X!Tandem for you. It makes it very easy to use X!Tandem:

```
run_xtandem.py --output OUTFILE --db /PATH/TO/FASTA --threads N --xtandem /PATH/TO/  
↪TANDEM.EXE
```


CHAPTER 2

Running

The TTT Proteotyping Pipeline can be run either by manually running each of the steps, or it can be controlled automatically using [Snakemake](#). The automation ensures that each RAW input file is taken through all the required steps to produce the final output. Together with the supplied `snakemake_crontab_script.sh` it can be used as a completely hands-off automated way of analyzing proteomics samples.

Note: The Snakemake workflow can only be run on Linux computers, as it depends on some Linux command line features.

Work directory

The Snakemake workflow requires a work directory containing the following folder structure:

```
0.raw
1.mzXML
2.xml
3.fasta
4.blast8
5.results
```

The reference data required to run the entire workflow is usually put in a single directory (or symlinked there), but they can (in theory) be located anywhere in the file system. The position of all the required files must be specified in the `TTT_pipeline_snakemake_config.yaml` file. This file must be specified on the command line when invoking the workflow.

Run the Snakemake workflow

To run the Snakemake workflow, ensure that a suitable Python/Conda environment is activated in which all the proteotyping programs and scripts are available in `PATH`. The minimal command line required to start the workflow is

this:

```
snakemake --snakefile SNAKEFILE --configfile CONFIGFILE
```

As the work directory is specified in the configfile, the command can in theory be run anywhere in the file system. It is recommended, however, that the Snakemake workflow is invoked via the use of the included `snakemake_crontab_script.sh` which sets some environment parameters to ensure reliable operation. It uses the linux command `flock` to ensure that only one instance of the workflow is ever run at the same time.

Automatic invocation via crontab

The workflow can be invoked automatically at set times using the Linux built-in `crontab`. To edit your personal user's crontab, type `crontab -e` at the command prompt. Add something like the following line to make the Snakemake workflow check for new files to analyze three times daily (00:00, 12:00, 18:00):

```
0 0,12,18 * * * /bin/bash /PATH/TO/snakemake_crontab_script.sh
```

Make sure to modify the configfile (`TTT_pipeline_snakemake_config.yaml`) and the crontab script file (`snakemake_crontab_script.sh`) to match your environment.

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`