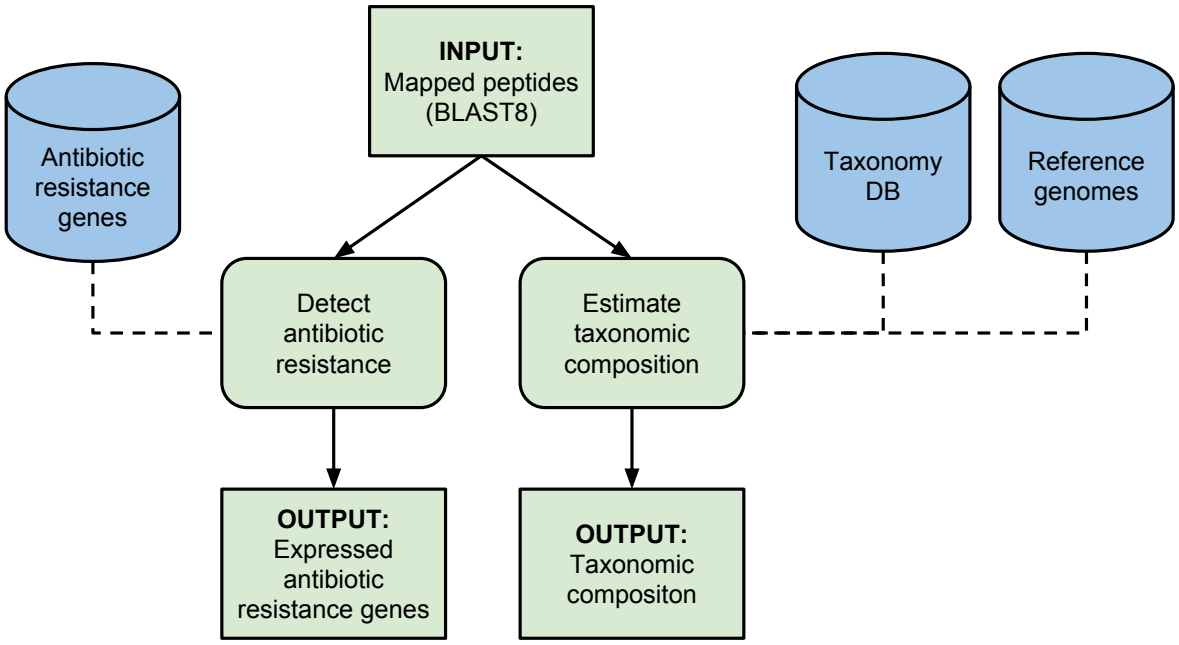
TCUP Documentation

Release 1.1.0

Fredrik Boulund

Sep 06, 2022

Contents



TCUP uses peptides generated by mass spectrometry proteomics to:

- identify and estimate the taxonomic composition of a microbial sample.
- detect expressed antibiotic resistance proteins.

TCUP does not require a priori information about the contents of a sample and is suitable for analysis of both pure cultures, mixed samples, and clinical samples.

Read *Introduction and overview* for more details on how it works. If you prefer to throw yourself head-first into TCUP without first knowing the details of how it works, check out *Installation* right away to get started!

CHAPTER 1

About

Authors Fredrik Boulund,

Contact fredrik.boulund@chalmers.se

This is the documentation for TCUP version 1.1.0, last updated Sep 06, 2022. The documentation is available online at

http://tcup.readthedocs.org

TCUP is published as open-source under the ISC license and you are welcome to look at, suggest improvements, or download and improve/contribute to the code at the project's Bitbucket page.

CHAPTER 2

Citing TCUP

If you find TCUP useful, please cite the following paper that describes the method:

Fredrik Boulund, Roger Karlsson, Lucia Gonzales-Siles, Anna Johnning, Nahid Karami, Omar AL-Bayati, Christina Ahren, Edward R. B. Moore, and Erik Kristiansson

TCUP: Typing and characterization of bacteria using bottom-up tandem mass spectrometry proteomics

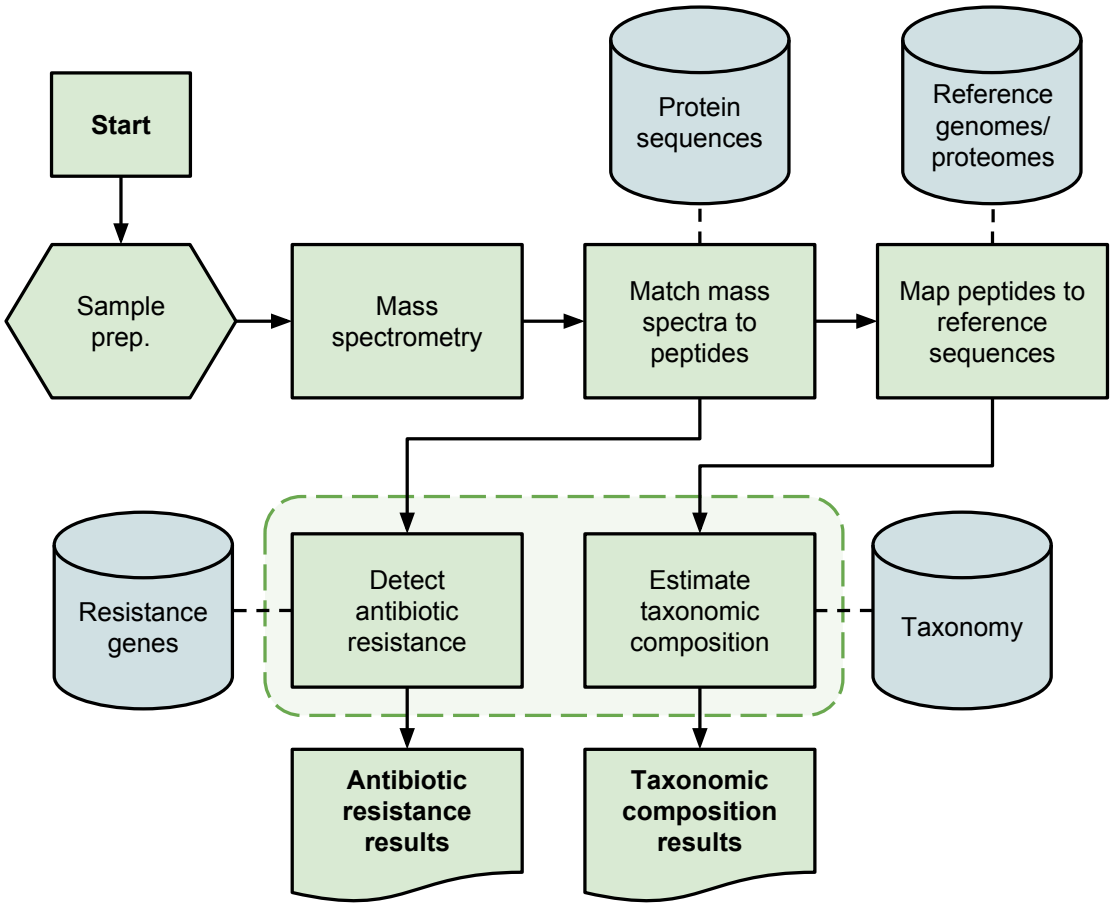
Mol Cell Proteomics mcp.M116.061721. First Published on April 18, 2017, doi:10.1074/mcp.M116.061721

CHAPTER 3

Documentation

3.1 Introduction and overview

TCUP can estimate the taxonomic composition of a sample of microorganisms using peptides generated from mass spectrometry. The programs contained in the TCUP package are meant to be used in a larger proteotyping workflow like this:



In the above picture, the components of TCUP are represented as the two boxes inside the dashed light green area in the lower center of the picture. Before TCUP can be used to estimate the taxonomic composition or detect expressed antibiotic resistance proteins in a sample, the following steps must be performed after the mass spectrometry:

1. Match tandem mass spectra with peptide sequences (i.e. protein identification).
2. Map the identified peptides to a database of reference genome sequences.

Step 1 can be performed using any mass spectrometry search engine, such as X!Tandem, Mascot, MyriMatch, Comet, Tide, etc. It is recommended to use the largest possible database of non-redundant protein sequences in this step, to allow for maximal similarity between the identified peptide sequences and the actual peptides in the sample. Step 2 can be performed using any sequence aligner/mapper capable of mapping protein sequences to nucleotide references, producing results in blast8 tabular format. We recommend using BLAT (or pBLAT).

TCUP uses several reference data sources (the two main ones depicted in the picture above) to create its estimate of a sample's taxonomic composition and detecting expressed antibiotic resistance proteins. More information on how to prepare these reference databases is available in *Preparing databases for use with TCUP*.

3.2 Installation

TCUP is written in Python 3.5 and runs on both Windows and Linux. The installation instructions in this document aim to serve both these platforms, but examples are given in Linux environments. Where notable difference between Windows and Linux install procedure exists, they will be pointed out. This guide assumes the user is reasonably familiar with the command prompt/terminal of these operating systems.

Note: We recommend using Anaconda Python. Anaconda Python runs on Windows and Linux and requires no administrator privileges, and can be installed in your user's home directory. Download and install Anaconda Python 3.5 (or Miniconda Python 3.5) using the instructions on their website.

If you are unable to use Mercurial (hg) in your environment, you can download a zipfile with the most recent version of TCUP from the Bitbucket repository. If you download TCUP using the zipfile in the above link, just skip any Mercurial (hg) related instructions below.

3.2.1 Install with Anaconda Python

Use Mercurial (hg) to clone the repository available via Bitbucket page to download the source code to be able to install the package:

```
hg clone https://bitbucket.org/chalmersmathbioinformatics/tcup
```

This creates a folder `tcup` in your current directory containing the source code for TCUP. Change into the directory you downloaded (e.g. `cd tcup`). Now, create a Conda environment into which you will install all dependencies required to run TCUP. The following commands will create a conda environment named "tcup", containing all the required dependencies, into which you will also install TCUP itself. Replace `<platform>` with the platform you are installing TCUP on (`Linux` or `Windows`):

```
conda env create -f conda_environment_<platform>.yml
```

After creating the conda environment, activate the environment and install TCUP into it (Windows users activate by typing `activate tcup`, omitting `source`):

```
source activate tcup
cd tcup
pip install .
```

Now you can read the section *Running TCUP* for information on how to use TCUP, or have a look at the *Tutorial* for a more brief introduction.

3.2.2 Download source code for TCUP

The complete source code for TCUP can be downloaded from the project's Bitbucket page, either by downloading a zipfile of the most recent version from the webpage, or using Mercurial (hg):

```
hg clone https://bitbucket.org/chalmersmathbioinformatics/tcup
```

This is useful if you want to help improve TCUP. Consult the `CONTRIBUTING` file in the project root folder for information on how to contribute to TCUP.

3.3 Tutorial

In this tutorial we will run TCUP on a sample input file that is available for download from here:

http://bioinformatics.math.chalmers.se/tcup/tutorial/tcup_tutorial_sample.zip

The input file contains 1500 peptides in FASTA format, randomly subsampled from a real tandem mass spectrometry sample, sarched using X!Tandem against a large peptide database of bacterial proteins.

3.3.1 Preparations

Follow the installation instructions in *Installation* to install TCUP. When you have completed the installation, download the following databases that are required to run TCUP:

- Reference genome database. BLAST or BLAT reference databases [FASTA or BLASTDB]. About 21 GB uncompressed.
- Taxonomy ("taxref") database. Links reference genome sequences to taxonomy structure [sqlite3]. About 340 MB uncompressed.
- Annotation database. Contains annotations for all genome sequences included in the reference genome database [sqlite3]. About 5 GB uncompressed.
- Resistance gene database. Based on ResFinder [FASTA or BLASTDB and sqlite3]. About 1 MB uncompressed.

Example versions of these databases can be downloaded from the links in the list above. Note that the databases are quite large. The total download size is approximately 10.3 GB. There are instructions on how to create your own databases in *Preparing databases for use with TCUP*. Do not forget to download the sample (linked in the section above).

You also need to install either BLAT (Linux) or BLAST (Windows), depending on your OS. Please refer to their respective installation instructions for installing them.

3.3.2 Running TCUP

A simple way to run TCUP is to use the included `run_tcup.py` script that is installed as part of the TCUP package. Running `run_tcup` without any arguments (or with `-h`/`--help`) produces this helpful output:

```
$ run_tcup.py
usage: run_tcup.py [-h] -t TAXREF_DB -a ANNOTATION_DB -r RESISTANCE_DB
                   SAMPLE GENOME_DB RESISTANCE_DB

TCUP wrapper; align peptides to reference databases in parallel and run TCUP
on alignment results. Fredrik Boulund 2016

positional arguments:
  SAMPLE                FASTA file with peptides from tandem MS.
  GENOME_DB             Reference bacterial genome db (FASTA or blastdb format
                        depending on OS).
  RESISTANCE_DB         Antibiotic resistance gene db (FASTA or blastdb format
                        depending on OS).

optional arguments:
  -h, --help            show this help message and exit

Taxonomic composition:
  -t TAXREF_DB, --taxref-db TAXREF_DB
                        Path to taxref db (sqlite3).
  -a ANNOTATION_DB, --annotation-db ANNOTATION_DB
                        Path to annotation db (sqlite3).

Antibiotic resistance:
  -r RESISTANCE_DB, --resistance-db RESISTANCE_DB
                        Path to resistance db (sqlite3).
```

As indicated by the help text, we need to supply a number of arguments to `run_tcup` in order to run TCUP. The program requires the `SAMPLE` file containing the sample peptides as the first argument. The second argument is the

path to the GENOME_DB. The third argument is the path to the RESISTANCE_DB. In addition to these positional arguments, three additional arguments are required: -t specifies the path to the taxref.sqlite3 file, -a is the path to the annotation_db.sqlite3 file, and -r is the path to the resfinder.sqlite3 file.

Assuming you have downloaded and extracted all the databases listed above, and downloaded the tutorial sample FASTA file, into a folder like this:

```
tutorial/
    tcup_tutorial_sample.fasta
    databases/
        annotation_db.sqlite3
        reference_genomes.00.nhr
        reference_genomes.00.nin
        reference_genomes.00.nsq
        reference_genomes.01.nhr
        reference_genomes.01.nin
        reference_genomes.01.nsq
        reference_genomes.01.fasta
        reference_genomes.02.nhr
        reference_genomes.02.nin
        reference_genomes.02.nsq
        reference_genomes.02.fasta
        reference_genomes.03.nhr
        reference_genomes.03.nhr
        reference_genomes.03.nin
        reference_genomes.04.nsq
        reference_genomes.04.nin
        reference_genomes.04.nsq
        reference_genomes.nal
        resfinder.fasta
        resfinder.phr
        resfinder.pin
        resfinder.psq
        resfinder.sqlite3
        taxref.sqlite3
```

To run TCUP on Windows, type the following command line (without linebreaks):

```
> run_tcup.exe
    -t databases\taxref.sqlite3
    -a databases\annotation_db.sqlite3
    -r databases\resfinder.sqlite3
    tcup_tutorial_sample.fasta
    databases\reference_genomes
    databases\resfinder
```

To run TCUP on Linux, type the following command line (without linebreaks):

```
$ run_tcup
    -t databases/taxref.sqlite3
    -a databases/annotation_db.sqlite3
    -r databases/resfinder.sqlite3
    tcup_tutorial_sample.fasta
    databases/reference_genomes
    databases/resfinder.fasta
```

Running TCUP on the tutoral sample will take some time, sometimes up to a couple of hours, depending on your computer's speed and amount of memory. After completing, TCUP will produce the following output files:

```
tcup_tutorial_sample.fasta.genomes.blast8
tcup_tutorial_sample.fasta.ar.blast8
tcup_tutorial_sample.fasta.antibiotic_resistance.txt
tcup_tutorial_sample.fasta.taxonomic_composition.txt
tcup_tutorial_sample.fasta.taxonomic_composition.xslx
```

The mapping output files `*.blast8*` contain the raw mapping results in BLAST tabular format (BLAST actually calls this blast6). The `*.txt` and `*.xlslx` files contain the output from TCUP.

Note: TCUP is actually not intended to be run via the 'run_tcup' script as described in this section. The script is provided as a convenience to easily try out TCUP to see how it works, but for real world use of TCUP, please refer to *Running TCUP*.

In the next section we will analyze the output from TCUP.

3.3.3 Analysis of the results

Note: NCBI BLAST produces more false positives than BLAT, and TCUP has only been optimized for use with BLAT at this time. The use of BLAST together with TCUP to determine taxonomic composition or expressed antibiotic resistance peptides is currently not recommended. Thus, if you are running TCUP on Windows, keep in mind that the results likely will contain a higher number of false positive assignments, both for taxonomic affiliation and antibiotic resistance detection.

Complete details on how to interpret TCUP output is available in *Example output*.

Taxonomic composition

First off, let's have a look at the taxonomic composition of the sample. The taxonomic composition estimation is presented in two formats: plain text and as an Excel spreadsheet. They both contain the same information regarding the taxonomic composition estimation of the sample, but the Excel file also includes a sheet with information on hits to annotated regions of the reference sequences.

The table in the first sheet of `tcup_tutorial_sample.fasta.taxonomic_composition.xslx` shows columns containing:

```
Cumulative  Count   Percentage  Rank    Spname
```

The leftmost column, `Cumulative`, shows the number of peptides that are discriminative at the taxonomic rank specified in the `Rank` column. This forms a cumulative sum as you look at ranks higher up in the taxonomic hierarchy. If e.g. the rank of superkingdom was included in the results, it would contain the total cumulative sum of the number of discriminative peptides at all taxa in the bacterial tree.

The `Percentage` column shows the relative proportion of peptides classified to the species given in the `Spname` column. This number is relative to all other entries of the same taxonomic rank, e.g. the sum of all the percentages across all *species* would sum to 100%.

The Excel format makes it easy to use the filtering functions in Excel to look at the most interesting parts of the results, e.g. to filter out only matches to the *genus* or *species* levels.

The second sheet in the Excel file contains a listing of all hits to regions in the reference genome sequences that were matched by any discriminative peptide.

Antibiotic resistance

Second, let's have a look at the antibiotic resistance results. These are presented in a text file. The output contains four columns:

```
Disc.  Hits   %    Family
```

The first column, `Disc.`, shows the number of discriminative peptides that matched to the resistance gene family listed in the `Family` column. The `Hits` column shows how many separate matches the discriminative peptides produced to the family in question. The `%` column shows the proportion of peptides that matched to each family.

Congratulations, you have now completed the tutorial. There is more detailed information on how to use TCUP in the *Running TCUP* section.

3.4 Preparing databases for use with TCUP

TCUP relies on several reference databases that need to be prepared in order to analyze a sample.

For creating the databases required for taxonomic composition estimation, the following reference data sources are required:

- Reference genome sequences (FASTA)
- Reference genome annotations (GFF)

The headers of the genome sequences along with their taxonomic affiliations are combined with NCBI Taxonomy to form an SQLite3 database used to map genome sequences to the taxonomy, along with species names etc. The genome annotations are put into a separate SQLite3 database, used to retrieve annotation information if required. These SQLite3 databases are referred to as *taxref DB* and *annotation DB* throughout this documentation. How to prepare these reference databases are described in *Taxref DB* and *Annotation DB*.

Additionally, for creating the database used for antibiotic resistance detection, some kind of reference data sources for antibiotic resistance genes is required, we recommend something like the publicly available ResFinder database. A version of the ResFinder database suitable for use with TCUP is available from our website.

Note that the detection of expressed antibiotic resistance proteins and the estimation of taxonomic composition are actually two independent programs in the TCUP package. This means that if you only require one of them, you only need to create the databases associated with the program you want to use.

Note: All examples of commands below are for Linux environments. Slight changes to the displayed commands might be required if working in a Windows environment.

3.4.1 Taxref DB

The *taxref DB* is central to determining the taxonomic composition of samples. The database is built from NCBI Taxonomy and contains information linking the reference genome sequences to nodes in the taxonomy.

To create the *taxref DB*, a single user-created input file is required, called *header_mappings*. This file is a tab separated text file with two columns:

```
<FASTA_HEADER>  <TAXID>
```

Note: <FASTA_HEADER> should only contain the first part of the FASTA header up to the first space (also excluding the ">" angle bracket symbol). For example, for a FASTA header like this:

```
>gi|158333233|ref|NC_009925.1| Acaryochloris marina MBIC11017 chromosome, complete␣
↪genome
```

the corresponding line in the *header_mappings* file should be (without the extra spaces around the TAB character):

```
gi|158333233|ref|NC_009925.1| <TAB> 329726
```

Header mappings can be created for NCBI RefSeq sequences using the *efetch_taxids.py* subprogram described in *Header mappings* below. It is also possible to create this mapping yourself, however you want to. The only important thing to note is to make sure that the first part of the FASTA header (up to the first space) is mapped to a valid NCBI taxid. As the accuracy of the taxonomic composition estimation is wholly dependent on correct assignments in the *taxref DB*, it is imperative that these assignments are as correct as possible.

The simplest invocation to create a *taxref DB* looks like this:

```
taxref_db  <HEADER_MAPPINGS>
```

where `<HEADER_MAPPINGS>` is the path to a *header_mappings* file as described above. The program allows multiple header mapping files to be specified on the command line. The program will automatically download the most recent version of NCBI Taxonomy to build the database from. Run `taxref_db --help` for a listing of all the available options.

Header mappings

A *header_mappings* file is required to create a *taxref DB*. For NCBI RefSeq sequences, it can be created using `efetch_taxids.py` to look up taxids via NCBI E-utils. This example works on Linux only:

```
$ grep ">" path/to/reference_genomes.fasta > reference_genome_headers.txt
$ efetch_taxids reference_genome_headers.txt > header_mappings.tab
```

It is also perfectly possible to manually create this file, or use some other method for pairing up the sequence headers to their respective taxid. Learn more about taxids on NCBI Taxonomy.

3.4.2 Annotation DB

The *annotation DB* is used after estimating the taxonomic composition to present what annotated regions in the reference genome sequences were matched by discriminative peptides in the sample. The annotation matches are presented in the xlsx (Excel) output, and can optionally be printed in the text file output using the `--print-annotations` flag to `taxonomic_composition`.

To create the *annotation DB*, the database creation program `annotation_db` parses GFF files (General Feature Format). The simplest invocation to create an *annotation DB* looks like this:

```
annotation_db  <TAXREF_DB>  <GFF_DIR>
```

where `<TAXREF_DB>` is the path to a pre-made *Taxref DB*, and `<GFF_DIR>` is the path to a directory containing GFF files to parse. The program allows multiple directories to be specified on the command line, and will recursively search subdirectories for GFF files as well.

Note: Creating an *annotation DB* requires that you have already completed a *taxref DB*. Also note that the program won't do anything with GFF files for reference genome sequences that were not included in the creation of the *taxref DB*.

3.4.3 Antibiotic resistance DB

To create your own antibiotic resistance gene database for use with TCUP. Use the `construct_resfinder_db.py` program supplied with TCUP. A typical invocation might look like this:

```
$ construct_resfinder_db.py --sequences <SEQUENCES> --notes <NOTES>
```

where `<SEQUENCES>` is the path to a FASTA file with antibiotic resistance gene sequences. `<NOTES>` is the path to the `notes.txt` file included with the ResFinder distribution.

Note: A manually curated version of ResFinder suitable for use with TCUP is available for download from our website.

3.5 Running TCUP

Before running TCUP, ensure that the requisite databases have been created (see *Preparing databases for use with TCUP*). The input data to TCUP are mappings of peptides from against the reference genomes database you used to create the *taxref DB* (see *Preparing databases for use with TCUP*).

Note: **IMPORTANT:** The FASTA headers of the peptides need to contain the peptide length. It should be encoded at the end of the header as a single integer after an underscore character ("_") at the end of the first space separated part of the header. E.g. the following header:

```
>peptidename_15 some other information
```

belongs to a peptide called *peptidename* that is 15 amino acids long.

The mappings of peptides to reference genome sequences must be in blast8 tabular format (without column headers or comments). We recommend using BLAT or pBLAT. On Windows, BLAST can be used.

Note: All examples of command invocations in this section are aimed towards Linux. Where notable differences exist, they will be pointed out.

3.5.1 BLAT/pBLAT

The peptides in the sample need to mapped to the reference genomes. We recommend the use of BLAT for this, as it performs very well with translated mapping to large genome sequences. It is important to tell BLAT to do protein-to-translated-dna mapping with the `-t=dnax -q=prot` arguments. We have successfully used the following command line options with tandem mass spectrometry data:

```
blat <ref_db.fasta> <sample.fasta> -t=dnax -q=prot -minScore=10 -stepSize=5 -
↪tileSize=5 -minIdentity=90 -out=blast8 <outfilename.blast8>
```

For the antibiotic resistance detection, the high sensitivity afforded by the above settings are not required, also the mapping is now protein-to-protein. Instead of using the above command line for antibiotic resistance detection, we recommend using the default settings, like so:

```
blat <resfinder.fasta> <sample.fasta> -out=blast8 <outfilename.blast8>
```

3.5.2 BLAST

We do not recommend using BLAST, but if no other option is available, it can be done using the following settings for taxonomic composition estimation:

```
tblastn.exe -query <query> -db <db> -out <outfile> -outfmt 6
```

For antibiotic resistance detection, we need to do protein-to-protein mapping using the following command line:

```
blastp.exe -query <query> -db <db> -out <outfile> -outfmt 6
```

Note: TCUP has not yet been optimized for use with BLAST, and the results might be be unreliable if BLAST is used for peptide to genome mapping.

3.5.3 Taxonomic composition

A typical invocation might look something like this:

```
taxonomic_composition \
    --taxref-db <TAXREF_DB> \
    --annotation-db <ANNOTATION_DB> \
    --write-xlsx <OUTPUT_XLSX_FILENAME> \
    --output <OUTPUT_TXT_FILENAME>
    <INPUT_BLAST8_FILENAME>
```

where `<TAXREF_DB>` is the path to the taxonomy reference database file, `<ANNOTATION_DB>` is the path to the annotation database file, `<OUTPUT_XLSX_FILENAME>` is desired filename for the output in Excel format, `<OUTPUT_TXT_FILENAME>` is the desired filename for the output in text format, and `<INPUT_BLAST8_FILENAME>` is the filename of a file in BLAST8 format containing the mapping results of sample peptides against the reference genomes.

Running `taxonomic_composition --help` will show a full list of all available options and their default values.

3.5.4 Antibiotic resistance

A typical invocation might look something like this:

```
antibiotic_resistance \
    --resfinder <RESFINDER_DB> \
    --output <OUTPUT_TXT_FILENAME> \
    <INPUT_BLAST8_FILENAME>
```

where `<RESFINDER_DB>` is the path to the ResFinder sqlite3 database file, `<OUTPUT_TXT_FILENAME>` is the desired filename for the text format output, and `<INPUT_BLAST8_FILENAME>` is the filename of a file in BLAST8 format containing the mapping results of peptides against the ResFinder database.

Running `antibiotic_resistance --help` will show a full list of all available options and their default values.

3.6 Example output

This section contains some examples of what the output of TCUP looks like. TCUP can output results in several formats.

3.6.1 Taxonomic composition

The main output format intended for human consumption for taxonomic composition estimation is `xlsx`, i.e. Microsoft Excel format. The Excel file is not produced by default, and has to be requested on the command line using `--write-xlsx XLSX_FILE`. The Excel output contains two sheets: *Taxonomic composition* and *Hits to annotated regions*. These two outputs are described in detail below.

Taxonomic composition

A summary table of all discriminative peptide assignments are listed in this sheet. The output can look like this (shortened for brevity):

```
Cumulative  Discriminative count    Percentage  Rank    Description
921         738                     98.29%      genus   Streptococcus
953         32                      97.94%      family  Streptococcaceae
171         157                     94.34%      species Streptococcus pneumoniae
7           7                       25.93%      no rank Streptococcus pneumoniae 70585
5           5                       18.52%      no rank Streptococcus pneumoniae␣
↪CGSP14
2           2                       7.41%       no rank Streptococcus oralis Uo5
1           1                       3.70%       no rank [Eubacterium] eligens ATCC␣
↪27750
1           1                       3.70%       no rank Campylobacter concisus 13826
1           0                       3.70%       no rank Clostridiales incertae sedis
1           1                       3.70%       no rank Mycobacterium rhodesiae NBB3
[...]
```

The columns, from left to right, are:

- `Cumulative`: The number of discriminative fragments at the reported Rank. These figures also include all peptides below the indicated Rank (hence *Cumulative*).
- `Discriminative count`: The number of discriminative peptides only at the indicated Rank. This figure does **not** include peptides on any other Rank.
- `Percentage`: The proportion of all discriminative peptides at the indicated Rank that belong to this identification. Note that these numbers are only comparable within a given Rank; i.e. the sum to 100% within each Rank and have no meaning when comparing two different ranks.
- `Rank`: The rank at which peptides were discriminative.
- `Description`: The strain, species, genus, family, etc.

Looking at the results displayed above, we can see that this sample most likely contains *Streptococcus pneumoniae*, as indicated by the strong signal at species level (94.34%). This is further strengthened by the strong indicators at both genus and family level. In this shortened output we see some peptides that were discriminative to different strains of *S. pneumoniae*, but also two peptides discriminative to *S. oralis*. Towards the end of the output, several spurious matches to organisms that were probably not in the sample are also visible.

When viewing the output in Microsoft Excel, it is possible to use the filtering tools to filter e.g. the `Rank` column on *species* to display only the results on species level. This makes it very easy to study the results in detail, depending on whatever question is most interesting to you.

Note: The taxonomic composition output can also be written to a text file, using the `--output` command line option. If this option is not used, this output is printed to STDOUT.

Hits to annotated regions

Results in the *Hits to annotated regions* sheet are somewhat harder to interpret than the results in the *Taxonomic composition* sheet. In this sheet, the output might look like this (shortened for brevity):

```
Species                                 Disc. level Count   Product
↪          Features
Streptococcus pneumoniae AP200          family      151     hypothetical protein
↪          ID=cds741;Name=YP_003876344.1;Parent=gene747;Note=Operon 340 Gene 1␣
↪protein supported gi%7C148997148%7Cref%7CZP_01824802.1%7C ribosomal protein S16;
↪Dbxref=Genbank:YP_003876344.1,GeneID:9726765;gbkey=CDS;product=hypothetical protein;
↪protein_id=YP_003876344.1;transl_table=11
Streptococcus pneumoniae R6             genus       96      hypothetical protein
↪          ID=cds26;Name=NP_357627.1;Parent=gene36;Dbxref=Genbank:NP_357627.1,
↪GeneID:933816;gbkey=CDS;product=hypothetical protein;protein_id=NP_357627.1;transl_
↪table=11
Streptococcus pneumoniae CGSP14         genus       81      hypothetical protein
↪          ID=cds32;Name=YP_001834750.1;Parent=gene38;Dbxref=Genbank:YP_001834750.1,
↪GeneID:6216513;gbkey=CDS;product=hypothetical protein;protein_id=YP_001834750.1;
↪transl_table=11
Streptococcus pneumoniae ST556          genus       78      hypothetical protein
↪          ID=cds106;Name=YP_006252305.1;Parent=gene112;Dbxref=Genbank:YP_006252305.
↪1,GeneID:12900645;gbkey=CDS;product=hypothetical protein;protein_id=YP_006252305.1;
↪transl_table=11
Streptococcus pneumoniae TIGR4          genus       78      hypothetical protein
↪          ID=cds29;Name=NP_344583.1;Parent=gene38;Dbxref=Genbank:NP_344583.1,
↪GeneID:929780;gbkey=CDS;product=hypothetical protein;protein_id=NP_344583.1;transl_
↪table=11
Streptococcus mitis B6                  genus       35      Translation elongation␣
↪factor TU    ID=cds854;Name=YP_003446034.1;Parent=gene929;Dbxref=Genbank:YP_
↪003446034.1,GeneID:8797981;gbkey=CDS;product=Translation elongation factor TU;
↪protein_id=YP_003446034.1;transl_table=11
Streptococcus pneumoniae 670-6B         genus       35      translation elongation␣
↪factor Tu    ID=cds793;Name=YP_003878976.1;Parent=gene805;Dbxref=Genbank:YP_
↪003878976.1,GeneID:9729530;gbkey=CDS;product=translation elongation factor Tu;
↪protein_id=YP_003878976.1;transl_table=11
Streptococcus pneumoniae 70585          genus       35      elongation factor Tu
↪          ID=cds1437;Name=YP_002740757.1;Parent=gene1488;Note=EF-Tu%3B promotes GTP-
↪dependent binding of aminoacyl-tRNA to the A-site of ribosomes during protein␣
↪biosynthesis%3B when the tRNA anticodon matches the mRNA codon%2C GTP hydrolysis␣
↪results%3B the inactive EF-Tu-GDP leaves the ribosome and release of GDP is␣
↪promoted by elongation factor Ts%3B many prokaryotes have two copies of the gene␣
↪encoding EF-Tu;Dbxref=Genbank:YP_002740757.1,GeneID:7684439;gbkey=CDS;
↪product=elongation factor Tu;protein_id=YP_002740757.1;transl_table=11
Streptococcus pneumoniae A026           genus       35      elongation factor Tu
↪          ID=cds1209;Name=YP_008730608.1;Parent=gene1336;Note=EF-Tu%3B promotes GTP-
↪dependent binding of aminoacyl-tRNA to the A-site of ribosomes during protein␣
↪biosynthesis%3B when the tRNA anticodon matches the mRNA codon%2C GTP hydrolysis␣
↪results%3B the inactive EF-Tu-GDP leaves the ribosome and release of GDP is␣
↪promoted by elongation factor Ts%3B many prokaryotes have two copies of the gene␣
↪encoding EF-Tu;Dbxref=Genbank:YP_008730608.1,GeneID:17439784;gbkey=CDS;gene=tuf;
↪product=elongation factor Tu;protein_id=YP_008730608.1;transl_table=11
```

(continues on next page)

(continued from previous page)

```
Streptococcus pneumoniae ATCC 700669    genus       35      elongation factor Tu␣
↪          ID=cds1258;Name=YP_002511360.1;Parent=gene1394;Note=EF-Tu%3B promotes GTP-
↪dependent binding of aminoacyl-tRNA to the A-site of ribosomes during protein␣
↪biosynthesis%3B when the tRNA anticodon matches the mRNA codon%2C GTP hydrolysis␣
↪results%3B the inactive EF-Tu-GDP leaves the ribosome and release of GDP is␣
↪promoted by elongation factor Ts%3B many prokaryotes have two copies of the gene␣
↪encoding EF-Tu;Dbxref=Genbank:YP_002511360.1,GeneID:7329336;gbkey=CDS;
↪product=elongation factor Tu;protein_id=YP_002511360.1;transl_table=11
[...]
```

The columns, left to right, are:

- `Species`: The species/strain name of the genome to which the matches occurred.
- `Disc. level`: The taxonomic rank at which the peptide that matched this genome was discriminative.
- `Count`: The number of hits/matches accrued by this annotated region (total across the entire sample). Note that these counts are not quantitative, because each discriminative peptide can produce multiple matches to several different annotated regions in several different reference genome sequences.
- `Product`: Annotation for the protein product of this annotated region.
- `Features`: The raw feature string for this annotated region.

As we can see in this output, there are several hits to regions in different *S. pneumoniae* genomes, to regions annotated with *hypothetical protein* and *elongation factor Tu*. This sheet provides information that can be useful to identify potential biomarkers or help understand what regions of the studied bacteria are discriminative, and at what taxonomic level.

Note: The *Hits to annotated regions* output can also be printed to STDOUT using the `--print-annotations` command line option. However, this is **not** recommended as it is very verbose.

Discriminative peptides

The names and sequences of all discriminative peptides along with the rank at which they were discriminative can be output to a text file using the `--write-discriminative-peptides` command line option.

3.6.2 Antibiotic resistance

Detected expressed antibiotic resistance peptides are summarized in a text file. The output can look like this:

```
-----------------------------------------------------------------------
Results for 124 discriminative peptides in tcup_tutorial_sample.fasta.ar.blast8
Disc.  Hits        %  Family
64     84     0.5114  blaCTX-M
45     45     0.2739  blaTEM
10     10     0.0609  aac(3)-II
5      5      0.0304  mph(A)
```

Here, we see that the sample likely contains four different families of antibiotic resistance mechanisms: blaCTX-M, blaTEM, aac(3)-II, and mph(A). The columns, from left to right, are:

- `Disc.`: The number of peptides that were discriminative to this family, i.e. peptides that did not match any other family.

- `Hits`: The number of aligned regions to this family that the discriminative peptides produced. This number of sometimes higher than the number of discriminative peptides, as each discriminative peptide can sometimes match to several variants within a family, or sometimes even several positions in the same protein.
- `%`: The proportion of peptides in the sample that were discriminative to this family.
- `Family`: The antibotic resistance gene family matched by discriminative peptides.

3.7 Publications

The main publication that describes TCUP is:

```
Fredrik Boulund, Roger Karlsson, Lucia Gonzales-Siles, Anna Johnning, Nahid Karami,␣
↪Omar AL-Bayati, Christina Ahren, Edward R. B. Moore, and Erik Kristiansson
TCUP: Typing and characterization of bacteria using bottom-up tandem mass␣
↪spectrometry proteomics
Mol Cell Proteomics mcp.M116.061721. First Published on April 18, 2017,
DOI: http://dx.doi.org/10.1074/mcp.M116.061721
Keywords: mass spectrometry, proteomics, microbial identification, pathogenic␣
↪bacteria, antibiotic resistance detection, LC-MS/MS
```

Other related publications:

```
Roger Karlsson, Lucia Gonzales-Siles, Fredrik Boulund, Liselott Svensson-Stadler,␣
↪Susann Skovbjerg, Anders Karlsson, Max Davidson, Stefan Hulth, Erik Kristiansson,␣
↪Edward R.B. Moore
Proteotyping: Proteomic characterization, classification and identification of␣
↪microorganisms - A prospectus
Systematic and Applied Microbiology, Volume 38, Issue 4, June 2015, Pages 246-257,␣
↪ISSN 0723-2020
DOI: http://dx.doi.org/10.1016/j.syapm.2015.03.006
Keywords: Proteotypingf Proteomics, Mass spectrometry, Microbial systematics
```

CHAPTER 4

Contributing

If you want to contribute to the development of TCUP, please refer to the *CONTRIBUTING* document in the root of the Bitbucket repository.

CHAPTER 5

Indices and tables

- genindex
- modindex
- search