**ProteoMapper tutorial**

Luis Mendoza
Institute for Systems Biology
Last Revision: May 25, 2021

**About**

This tutorial covers the downloading, installation, and basic usage of the **ProteoMapper** tools through various scenarios. Please contact the author if you have any questions or have any issues with this tutorial.

**Introduction**

**ProteoMapper** is a set of tools that enable the mapping of peptide sequences to protein sequences using a full reverse index of segments. The software supports protein sequence variants (specified via the *PEFF* keyword *VariantSimple*), as well as wildcard and fuzzy mapping, multiple output formats, and is very quick.

For more information, visit: http://www.tppms.org/pm/

**Prerequisites**

The software is written in the *Perl* language, and thus it must be installed and available in the system. No other special modules or dependencies are required.

Also, since the software operates on a sequence database file, at least one must be available, in either *fasta* or *PEFF* format.

**Download and test installation**

1. Go to <http://www.tppms.org/tools/pm/> to download the *proteomapper.zip* bundle
2. Unzip the file and make sure you have two components: **clips** and **promast**; clips is the indexer, and promast is the mapper
3. Open a command window (or a shell in Linux) and type the name of the **clips** command and hit [enter]. You should get a short usage statement. If you see any errors, verify that *Perl* is installed and in your path. Do the same for **promast**.
4. From the same website, download and uncompress the file **nextprot\_all.peff.gz** . This is a recent version of the Human protein database from neXtProt in *PEFF* format. For a more recent version, go to: <https://download.nextprot.org/pub/current_release/peff/>

**Indexing and Mapping 1: no variants**

1. Create a segments index of the nextprot file, discarding variants, by issuing the following command:
**clips –V nextprot\_all.peff**This should only take under one minute on a moderately-powered system; notice the diagnostic and progress information given by the indexer.
2. Examine the directory to confirm the presence of the new index file called **nextprot\_all.peff.pep.idx** . This is the segments index file, and can be renamed (more on that later). It should be about 50% larger than the original file.
3. Find the protein mapping of a single peptide, PPLPKSR:
**promast nextprot\_all.peff PPLPKSR**How many proteins does it map to? Note that the start position is also indicated in the output from ProMaST.
4. Use the fuzzy mapping function in ProMaST to map a peptide sequence from a de novo search. Let’s say that you ran a de novo algorithm on a spectrum, and got the following sequence as the best match: **EDDSLSPASANDDK**
- Does this sequence map to any proteins?

- Use the “fuzzy mapping” option in promast, with one unknown amino acid with the following command:
**promast –f 1 nextprot\_all.peff EDDSLSPASANDDK**Notice that there are no mappings (all results are “UNMAPPED”), even though promast generated 253 versions of the peptide by replacing every amino acid at every position of the peptide. Since we are only interested in actual mappings, we will use the –U option to skip those that do not map.

- Now let’s use a fuzzy mapping of two unknown amino acids: **promast –U –f 2 nextprot\_all.peff EDDSLSPASANDDK**Are there any matching proteins? What was the actual peptide sequence that was matched to the protein? Do you expect this to be common when doing de novo sequencing? Why/why not?

**Indexing and Mapping 2: PEFF variants**

1. Before creating a new index, let’s save the current index by renaming it:
**mv nextprot\_all.peff.pep.idx nextprot\_all.peff.noVars.pep.idx**
2. Create a segments index file that also includes all permutations of known amino acid variants (which should be specified via the *VariantSimple* keyword in the PEFF file):
**clips nextprot\_all.peff**This will take significantly longer than the previous index, around 15 minutes or so.
3. Examine the directory; you should have a new index file that is about 12 times larger than the original protein sequence file.
4. Find the protein mapping the peptide, PPLPKSR:
**promast nextprot\_all.peff PPLPKSR**Note that this is the same sequence that we used in step 7 above. Does it map to new protein sequences? Look up any interesting proteins in PeptideAtlas or neXtProt to find out which variants are involved in the new mapping.