Manual

1. Copyright
2. Running environment
3. Data preparation
4. Install and compile
5. Configure File Setting
6. Regular Search
7. Post translational modification search
8. Stable isotope probing search
9. Isobaric chemical labeling search
10. Bug report

1. Copyright

2. Running environment

Running Python scripts requires Python 2.7.2 or above Python 2 versions. Running R scripts requires R environment. Compiling Sipros source code requires a C++ compiler such as g++ for Linux or MinGW g++ for Windows.

3. Data preparation

Raxport can be used to Thermo Raw files to FT1 and FT2 text files with the following command.

raxport.exe -w WorkingDirectory

mzML2FT can be used to transfer mzML files to FT1 and FT2 text files with the following command.

python mzML2FT.py -w WorkingDirectory

Please note that running mzML2FT requires pymzml library. Both software are available at http://code.google.com/p/raxport/downloads/list.

4. Install and compile

Window

Double click

Sipros-ProRata_V3.0_Setup.exe

Linux

tar -zxzf Sipros-ProRata_V3.0_Linux.tar.gz

cd Sipros-ProRata_V3.0_Linux/src

if Sipros will be used on a desktop, you just need Openmp feature. Please execute

make

You will get binary file Sipros_openmp

If you need the MPI/OpenMP feature of Sipros, please execute

make -f MPIMakefile

You will get binary file Sipros_openmp_mpi

In this wiki, we will use Sipros to refer to the binary file. Please note that the current makefile uses g++ by default. If you want to switch to other compilers or add flags, please update the corresponding makefile.

5. Configure File Setting.

is for comments.

[] is used for section name, e.g., [Section Name].

= is used for assigning features, e.g., Search_Type = Regular

{} is used for specifying key value, e.g., PTM{!} = NQR

Currently, there are 35 symbols available for specifying ptms, which are

~ ! @ $ % ^ & * ( ) _ + ` - | \ : " ; ' < > ? . /

1 2 3 4 5 6 7 8 9 0

Please don't use these reserved symbols: { } # [ ] = ,

Neutral loss can be specified by PTM{1to2}, e.g., PTM{>to|} = ST. If symbol2 is nothing, it can be specified by PTM{1to}, e.g., PTM{>to} = ST.

6. Regular Search

6.1. **Create configure file**

Please refer to "Configure File Setting" for technical details. An example is available at http://code.google.com/p/sipros/source/browse/trunk/Sipros3.0/Sipros-src/SiprosConfigRegular.cfg

6.2. **FT1 and FT2 files**

Please refer to http://code.google.com/p/raxport/ for generating FT1 and FT2 files

6.3. **Pre-processing**

The current version of scripts has been tested using Python 2.7.2, so if you are using different versions of Python (2.6.X or 3.X), you are encouraged to try with Python 2.7.2. If you want to use reverseseq.py, Biopython is required.

cd Sipros-ProRata_V3.0_Linux/scripts

6.4. **Generate reverse sequence**

python reverseseq.py -i original_database_file -o output_database_file

The step will generate a new database file with reverse sequences.

6.5 **Run Sipros**

The easiest way for running Sipros is

./Sipros -c configurefilename -w workingdirectory

Sipros will use all .FT2 files in the working directory. Results (.sip files) will be saved on working directory by default. You can specify output directory by specifying -o. If you just want to specify one FT2 file, you can use -f like,

./Sipros -c configurefilename -f FT2filename

If you want to slice screen output, please add -s. In Sipros_openmp_mpi, you are allowing to specify -g like

./Sipros -g configurefiledirectory -w workingdirectory

if you have many configure files.

Please note that you can get help information by flag -h.

6.6. **Post-processing**

The current version of scripts has been tested using Python 2.7.2, so if you are using different versions of Python (2.6.X or 3.X), you are encouraged to try with Python 2.7.2.

cd Sipros-ProRata_V3.0_Linux/scripts

6.6.1. Peptide filtering

python sipros_peptides_filtering.py -c configurefile -w workingdirectory

All sip files, output files of Sipros, should be in the working directory. The step will generate related psm.txt and pep.txt

6.6.2. Peptide assembling

python sipros_peptides_assembling.py -c configurefile -w workingdirectory

The psm.txt and pep.txt files generated by the filtering step will be used in this step. And the step will generate pro.txt, pro2pep.txt, and pro2psm.txt files

7. Post translational modification search

7.1. **Create configure file**

Please refer to "Configure File Setting" for technical details. An example is available at http://code.google.com/p/sipros/source/browse/trunk/Sipros3.0/Sipros-src/SiprosConfigPTM.cfg

7.2. **FT1 and FT2 files**

Please refer to http://code.google.com/p/raxport/ for generating FT1 and FT2 files.

7.3. **Pre-processing**

The current version of scripts has been tested using Python 2.7.2, so if you are using different versions of Python (2.6.X or 3.X), you are encouraged to try with Python 2.7.2. If you want to use reverseseq.py, Biopython is required.

cd Sipros-ProRata_V3.0_Linux/scripts

7.4. **Generate reverse sequence**

python reverseseq.py -i original_database_file -o output_database_file

The step will generate a new database file with reverse sequences

7.5. Run Sipros

The easiest way for running Sipros is

./Sipros -c configurefilename -w workingdirectory

Sipros will use all .FT2 files in the working directory. Results (.sip files) will be saved on working directory by default. You can specify output directory by specifying -o. If you just want to specify one FT2 file, you can use -f like,

./Sipros -c configurefilename -f FT2filename

If you want to slice screen output, please add -s. In Sipros_openmp_mpi, you are allowing to specify -g like

./Sipros -g configurefiledirectory -w workingdirectory

if you have many configure files.

Please note that you can get help information by flag -h.

7.6. Post-processing

The current version of scripts has been tested using Python 2.7.2, so if you are using different versions of Python (2.6.X or 3.X), you are encouraged to try with Python 2.7.2. If you want to use pro2ptm.py, Biopython is required.

cd Sipros-ProRata_V3.0_Linux/scripts

7.6.1. Peptide filtering

python sipros_peptides_filtering.py -c configurefile -w workingdirectory

All sip files, output files of Sipros, should be in the working directory. The step will generate related psm.txt and pep.txt

7.6.2. Peptide assembling

python sipros_peptides_assembling.py -c configurefile -w workingdirectory

The psm.txt and pep.txt files generated by the filtering step will be used in this step. And the step will generate pro.txt, pro2pep.txt, and pro2psm.txt files.

Pro2ptm (Biopython needs to be installed)

python pro2ptm.py -c configurefile -w workingdirectory

Script pro2ptm.py will generate pro2ptm.txt file

If you also need pro2ptm.prorata.txt file, please run pro2ptm.py first and add ProRata_LabelFree_Peptide.txt, a output file of ProRata, to the working directory. Then run

python prorata_pro2ptm.py -w workingdirectory

This step will generate pro2ptm.prorata.txt file.

8. Stable isotope probing search

8.1. **Create configure file**

Please refer to "Configure File Setting" for technical details. An example is available at http://code.google.com/p/sipros/source/browse/trunk/Sipros3.0/Sipros-src/SiprosConfig.N15_SIP.cfg

8.2**. FT1 and FT2 files**

Please refer to http://code.google.com/p/raxport/ for generating FT1 and FT2 files.

8.3. **Pre-processing**

The current version of scripts has been tested using Python 2.7.2, so if you are using different versions of Python (2.6.X or 3.X), you are encouraged to try with Python 2.7.2. If you want to use reverseseq.py, Biopython is required.

cd Sipros-ProRata_V3.0_Linux/scripts

8.4. **Generate reverse sequence**

python reverseseq.py -i original_database_file -o output_database_file

The step will generate a new database file with reverse sequences.

8.5. Generate new configure files

python sip.py -c configurefilename -w workingdirectory

New configure files will be generated in the working directory.

8.6. Run Sipros

The easiest way for running Sipros is

./Sipros -c configurefilename -w workingdirectory

Sipros will use all .FT2 files in the working directory. Results (.sip files) will be saved on working directory by default. You can specify output directory by specifying -o. If you just want to specify one FT2 file, you can use -f like,

./Sipros -c configurefilename -f FT2filename

If you want to slice screen output, please add -s. In Sipros_openmp_mpi, you are allowing to specify -g like

./Sipros -g configurefiledirectory -w workingdirectory

If you have many configure files.

Please note that you can get help information by flag -h.

8.7. Post-processing

The current version of scripts has been tested using Python 2.7.2, so if you are using different versions of Python (2.6.X or 3.X), you are encouraged to try with Python 2.7.2.

cd Sipros-ProRata_V3.0_Linux/scripts

8.7.1. Peptide filtering

python sipros_peptides_filtering.py -c configurefile -w workingdirectory

All sip files, output files of Sipros, should be in the working directory. The step will generate related psm.txt and pep.txt

8.7.2. Peptide assembling

python sipros_peptides_assembling.py -c configurefile -w workingdirectory

The psm.txt and pep.txt files generated by the filtering step will be used in this step. And the step will generate pro.txt, pro2pep.txt, and pro2psm.txt files.

8.7.3. Cluster

python ClusterSip.py -c configurefile -w workingdirectory

This step will generate pro.cluster.txt and pro2psm.cluster.txt files.

9. Isobaric chemical labeling search

9.1. Create configure file

Please refer to "Configure File Setting" for technical details. An example is available at http://code.google.com/p/sipros/source/browse/trunk/Sipros3.0/Sipros-src/SiprosConfigIsobaricTags.cfg.

9.2. **FT1 and FT2 files**

Please refer to http://code.google.com/p/raxport/ for generating FT1 and FT2 files.

9.3. **Pre-processing**

The current version of scripts has been tested using Python 2.7.2, so if you are using different versions of Python (2.6.X or 3.X), you are encouraged to try with Python 2.7.2.

cd Sipros-ProRata_V3.0_Linux/scripts

9.4. **Generate reverse sequence**

python reverseseq.py -i original_database_file -o output_database_file

The step will generate a new database file with reverse sequences

9.5. Run Sipros

The easiest way for running Sipros is

./Sipros -c configurefilename -w workingdirectory

Sipros will use all .FT2 files in the working directory. Results (.sip files) will be saved on working directory by default. You can specify output directory by specifying -o. If you just want to specify one FT2 file, you can use -f like,

./Sipros -c configurefilename -f FT2filename

If you want to slice screen output, please add -s. In Sipros_openmp_mpi, you are allowing to specify -g like

./Sipros -g configurefiledirectory -w workingdirectory

if you have many configure files.

Please note that you can get help information by flag -h.

9.6. Post-processing

The current version of scripts has been tested using Python 2.7.2, so if you are using different versions of Python (2.6.X or 3.X), you are encouraged to try with Python 2.7.2. If you want to use reverseseq.py, Biopython is required.

cd Sipros-ProRata_V3.0_Linux/scripts

9.6.1. Peptide filtering

python sipros_peptides_filtering.py -c configurefile -w workingdirectory

All sip files, output files of Sipros, should be in the working directory. The step will generate related psm.txt and pep.txt

9.6.2. Peptide assembling

python sipros_peptides_assembling.py -c configurefile -w workingdirectory

The psm.txt and pep.txt files generated by the filtering step will be used in this step. And the step will generate pro.txt, pro2pep.txt, and pro2psm.txt files.

9.6.3. Isobaric

python isobaric.py -c configurefile -w workingdirectory

python pro2isobaric.py -c configurefile -w workingdirectory

This step will generate pro.isobaric.txt, psm.isobaric.txt, and pro2psm.isobaric.txt files.

10. Bug report

If you find any bug, please report it at http://code.google.com/p/sipros/issues/list.