Tutorial for Running PECAN on a Grid Engine Cluster

Ying Sonia Ting (sonia810@uw.edu)

Sep 18, 2016

Table of Contents

Introduction

PECAN (PEptide-Centric Analysis) is a tool for peptide detection directly from DIA data without the need of a spectral library. PECAN takes a list of peptide sequences and query each peptide against the DIA data (in centroid mzML format) and reports the best evidence of detection for each peptide. The PECAN report is designed for Percolator to separate correct from incorrect matches with false discovery rate (FDR) control.

In the current implementation, PECAN demands a lot of memory. When querying with the human UniProt Swiss-Prot database, the memory requirement for all peptides in one 5*m/z*-wide isolation window is approximately 16G. (~50G if the DIA windows are 20*m/z*-wide). This memory request is also proportional to the "length" of the data. For example, data acquired with longer LC time typically have more MS1 and MS2 spectra, and thus require more memory and processing time.

Thus, a computer cluster is an easy solution for PECAN analysis. Below are the instructions for running the PECAN pipeline on a Grid Engine Cluster.

Workflow: pecan and pecanpie

In current implementation, each *pecan* run query one DIA isolation window. Only precursor ions of the query peptides that falls within the targeted isolation window are kept and queried against the slice of DIA data that was acquired with the isolation window. If a DIA method has 20 isolation windows, *pecan* needs to be run with each of the 20 windows to finish querying the data. *pecan* outputs a tab-delimited table called a feature file that contains the reported evidence of detection and the associated auxiliary scores.

pecanpie is script that generates a pipeline of jobs for submission to a Grid Engine cluster. *pecanpie* takes a list of mzML files, a list of target peptides, and a list of isolation windows, and then generates array jobs that pipelines three *pecan, percolator,* and *pecan2blib* (a spectral library builder for pecan). These array jobs are set up so that each mzML file is queried sequentially with every isolation window, and the resulting feature files are concatenated for *percolator* analysis.

Installation Instruction

System requirement

Before install the PECAN package, make sure your system is complied with the following requirement:

1. Linux OS
2. Grid Engine Scheduler SGE6.2u5p2 or newer*
3. Python 2.7.4 or newer (currently not compatible with Python 3)
4. Python package NumPy v1.6 or newer
5. Python package pymzML v0.7.4 or newer
6. Percolator v2.08 or newer

*If your computer resource is managed by a different scheduling system, such as Condor, certain changes will need to be made in the **pecanpie** job-generator. Please ask your system administrator to contact us for further assistance.

Install PECAN package

1. Download the PECAN release and decompress the file

2. Set up your environment configuration for PECAN using the config file located at:

   ```
   PECAN/PecanUtil/config
   ```

   The config file contains four sections, [env], [sge], [percolator], and [species]. Lines start with # are comments and will be ignored. All variables on the left of the equals signs are not to be changed.
   The first section [env] has four variables, specifying the paths and other environmental settings required for PECAN.

   ```
   [env]
   # path to the scripts
   # if local installation is used, please specify like this
   # PECAN = /local/install/pecan
   # PECAN2BLIB = /local/install/pecan2blib
   PECAN = pecan
   PECAN2BLIB = pecan2blib

   # path to the species proteome databases
   # this is the folder containing all files in the [species] section
   DBDIR = /data/peoteome_database

   # if using module, load the necessary modules here
   # otherwise, leave it empty like this
   # MODULELOAD =
   MODULELOAD = modules, modules-init, python/2.7.2, numpy/1.6.1
   ```

   The second section [sge] contains one variable MEM_STRING, used to request for memory for each job. This section should be tailored to your cluster scheduler settings. If system needed, adding other variables is possible with corresponding changes in the **pecanpie** job-generator.

   ```
   [sge]
   # memory request phrase for your grid engine system
   MEM_STRING = m_mem_free
   ```

   The third section [percolator] contains two variables.

   ```
   [percolator]
   VERSION = 2.08
   # if using module, load the necessary modules for percolator here
   # otherwise, leave it empty like this
   # MODULELOAD =
   ```

```
MODULELOAD = boost/1.52.0, mpc/0.8.2, mpfr/3.0.0, gmp/5.0.2, gcc/4.7.0,
percolator/2.08.01
```

The last section [species] can contain as many variable as you wanted.

```
[species]
# name the species proteome databases in the form of peptide list
# these files should located in the folder specified by
# variable DBDIR in the [env] section
HUMAN = human_20150911_uniprot_sp_digested_Mass600to4000.txt
YEAST = yeast_20150911_uniprot_sp_digested_Mass600to4000.txt
MOUSE = mouse_20150911_uniprot_sp_digested_Mass600to4000.txt
ECOLI = ecoli_20150911_uniprot_sp_digested_Mass600to4000.txt
```

3. Install PECAN with (root permission required)

```
[user@server]$ python setup.py install
```

4. After installation, if you wish to make changes in the config setting, go to the relative location for your default python2.7 site-package, such as:

```
/usr/local/lib/python2.7/site-packages/PECAN/PecanUtil/config
```

Species database should contain two columns named Proten_Name and Sequences:

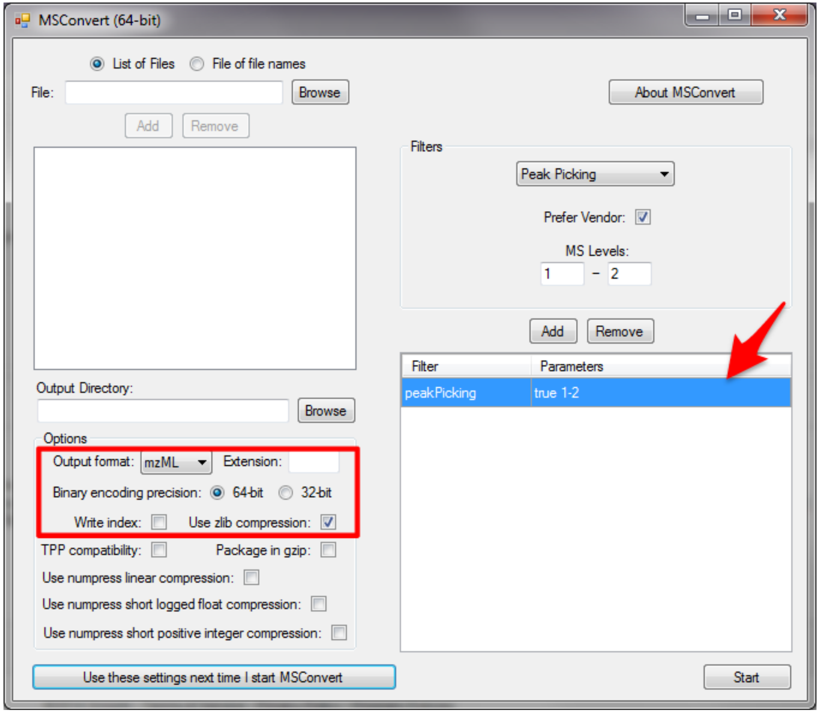
Protein_Name	Sequence
sp|P31946|1433B_HUMAN	MTMDKSELVQK
sp|P31946|1433B_HUMAN	SELVQKAK
sp|P31946|1433B_HUMAN	LAEQAERYDDMAAAMK
sp|P31946|1433B_HUMAN	YDDMAAAMK
sp|P0DMP2|SRG2B_HUMAN	QRLMEMYNNVFCPPMK
sp|P0DMP2|SRG2B_HUMAN	LMEMYNNVFCPPMK
sp|P0DMP2|SRG2B_HUMAN	FEFQPHMGDMASQLCAQQPVQSELLQR
sp|P0DMP2|SRG2B_HUMAN	STVSETFMSKPSIAK
sp|P0DMP2|SRG2B_HUMAN	RANQQETEQFYFTVR
sp|A6NKU9|SPDE3_HUMAN	IHFFLALYLANDMEEDDEAPKQK
sp|A6NKU9|SPDE3_HUMAN	QKIFYFLYGK
sp|A6NKU9|SPDE3_HUMAN	IFYFLYGK

PECAN Workflow

STEP 1: Convert native files to mzML files

The mzML format is an open, XML-based format for mass spectrometer output files, developed with the full participation of vendors and researchers in order to create a single open format that would be supported by all software. To convert vender native mass spectrometry files to mzML files, we recommend the MSConvert included in the ProteoWizard Library and Tools, an open-source, cross-platform tools and software libraries that facilitate proteomics data analysis, available at http://proteowizard.sourceforge.net/.

1. General instructions for converting files. For specifics on the Fusion and SCIEX data, see below. PECAN takes mzML files generated using 64-bit, zlib compression, and with centroid peaks rocessed by Filters > Peak Picking > Prefer Vendor > MS Levels 1-2 > Add (as shown below). After file format conversion, upload the mzML files to your desired location, such as: /data/instrument/mzML/project_folder



MSConvert (64-bit)

● List of Files ○ File of file names

File: Browse

About MSConvert

Filters: Peak Picking

Prefer Vendor: ☑

MS Levels: 1 - 2

Filter	Parameters
peakPicking	true 1-2

Output Directory:

Options

Output format: mzML Extension:

Binary encoding precision: ● 64-bit ○ 32-bit

Write index: □ Use zlib compression: ☑

TPP compatibility: □ Package in gzip: □

Use numpress linear compression: □

Use numpress short logged float compression: □

Use numpress short positive integer compression: □

2. Orbitrap Fusion (Lumos) instruments. When SIM scans are used for MS1, proteoWizard GUI cannot correctly convert the raw files to mzML. Use the ProteoWizard command-line interface to convert the raw files to mzML format.

```
msconvert.exe -c msconvert-SIMMS1.config raw_files
```

msconvert-SIMMS1.config file should include:

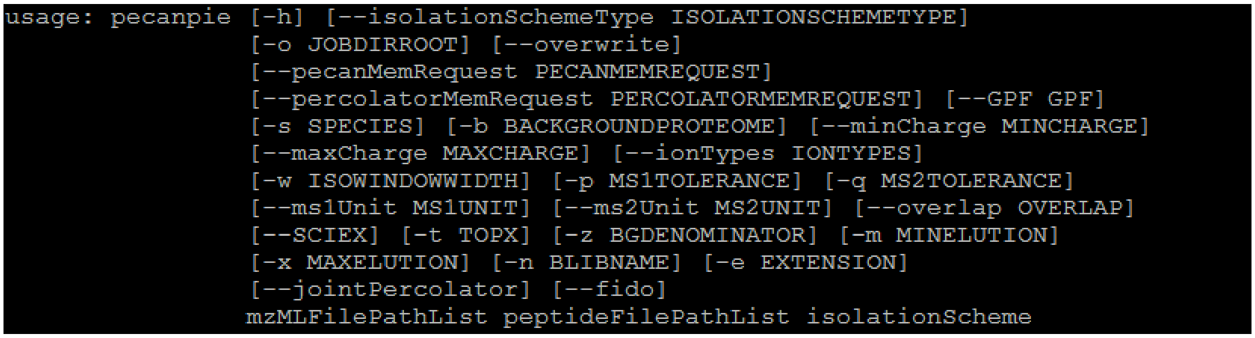
```
##################
outdir=mzMLs
mzML=true
zlib=true
mz64=true
inten64=true
simAsSpectra=true
filter="peakPicking vendor msLevel=1-2"
##################
```

3. Data from SCIEX instruments should be converted using SCIEX MS Data Converter for best results.

STEP 2: Prepare your input files

pecanpie is the pipeline generator for PECAN analysis. It generates folders and scripts that are ready to submit for SGE cluster. ** If other types of cluster are used, such as condor, some modifications are necessary for the submission scripts to work properly (e.g. allocate the memory).

pecanpie takes three input files, **mzMLFilePathList, peptideFilePathList,** and **isolationScheme**. We recommend users named their input files in a more meaningful way.



```
usage: pecanpie [-h] [--isolationSchemeType ISOLATIONSCHEMETYPE]
                [-o JOBDIRROOT] [--overwrite]
                [--pecanMemRequest PECANMEMREQUEST]
                [--percolatorMemRequest PERCOLATORMEMREQUEST] [--GPF GPF]
                [-s SPECIES] [-b BACKGROUNDPROTEOME] [--minCharge MINCHARGE]
                [--maxCharge MAXCHARGE] [--ionTypes IONTYPES]
                [-w ISOWINDOWWIDTH] [-p MS1TOLERANCE] [-q MS2TOLERANCE]
                [--ms1Unit MS1UNIT] [--ms2Unit MS2UNIT] [--overlap OVERLAP]
                [--SCIEX] [-t TOPX] [-z BGDENOMINATOR] [-m MINELUTION]
                [-x MAXELUTION] [-n BLIBNAME] [-e EXTENSION]
                [--jointPercolator] [--fido]
                mzMLFilePathList peptideFilePathList isolationScheme
```

mzMLFilePathList contains a list of path to the mzML files to be analyzed together by PECAN. For example, for a list of a HeLa experiment acquired on 01/01/2016, this input file could be named **20160101_QE_10mzDIA_HeLa_mzmls.txt** and contains:

```
/data/instrument/mzML/project_folder/HeLa_A1.mzML
/data/instrument/mzML/project_folder/HeLa_A2.mzML
/data/instrument/mzML/project_folder/HeLa_A3.mzML
/data/instrument/mzML/project_folder/HeLa_A4.mzML
```

peptideFilePathList contains a list of paths to the “peptide list(s)” to be queried. For example, if the user wish to query the DIA data with only mitochondrial proteins and transcription factors, this input file could be named **query_human_mt_tf_heavystd.txt** and contains:

```
/home/user/proj/project_folder/human_mitochondria_peptides.txt
/home/user/proj/project_folder/human_transcriptionfactors_peptides.txt
/home/user/proj/project_folder/heavy_spikedin_standards.txt
```

Each peptideFile in the peptideFilePathList should contain two specific columns named **Protein_Name** and **Sequence**. A peptide list file could contain as many other columns as wanted, as long as the two mandatory columns are present. PECAN accepts modifications annotated in the format of **aa[±123.4567890]** as shown in the STVSE**T[+97.9769]**FMSKPSIAK example below. Noted that only the peptides specified are queried. In the following example, both unmodified STVSETFMSKPSIAK and STVSE**T[+97.9769]**FMSKPSIAK will be used to query DIA data respectively, but not the other potential forms of phosphorylation such as STV**S[+97.9769]** ETFMSKPSIAK because they are not specified in the file.

Protein_Name	Sequence
sp|P31946|1433B_HUMAN	LAEQAERYDDMAAAMK
sp|P31946|1433B_HUMAN	YDDMAAAMK
sp|P0DMP2|SRG2B_HUMAN	QRLMEMYNNVFCPPMK
sp|P0DMP2|SRG2B_HUMAN	FEFQPHMGDMASQLCAQQPVQSELLQR
sp|P0DMP2|SRG2B_HUMAN	STVSE**T[+97.9769]**FMSKPSIAK
sp|P0DMP2|SRG2B_HUMAN	STVSETFMSKPSIAK
sp|P0DMP2|SRG2B_HUMAN	RANQQETEQFYFTVR
sp|A6NKU9|SPDE3_HUMAN	IHFFLALYLANDMEEDDEAPKQK
sp|A6NKU9|SPDE3_HUMAN	QKIFYFLYGK
sp|A6NKU9|SPDE3_HUMAN	IFYFLYGK

isolationScheme contains the DIA precursor isolation windows used, represented in two modes TARGET / BORDER mode. No column name is required.

[TARGET mode] TARGET mode is used with the following options (only suitable for fixed-window-width DIA).

```
--isolationSchemeType TARGET -w isolationWindowWidth
```

For the TARGET mode, the isolationScheme file should contain only one column: the precursor *m/z* of the center of each isolation window. This file can be named as **isoW_10mz_400to450.txt** and contains:

405.0
415.0
425.0
435.0
445.0

[BORDER mode] BORDER mode is used with the following options (suitable for both fixed-window-width and variable-window-width DIA).

```
--isolationSchemeType BOARDER
```

For the BORDER mode, the isolationScheme file should contain two columns: the precursor *m/z* of the start and end of each isolation window. This file could be named as **isoW_var_400to450.txt** is an example.

400.0	420.0
420.0	435.0
435.0	445.0
445.0	450.0

STEP 3: Set parameters

** Optional parameters that are most likely to change every data set **

`[-o JOBDIRROOT]`
Name your PECAN job directory

`[-w ISOWINDOWWIDTH]`
Isolation window width in *m/z* used for DIA. Only applicable when using TARGET isolation scheme mode

`[-s SPECIES]`
Use the default background proteomes by specifying species (ecoli/yeast/mouse/human) defined in the config file

`[-b BACKGROUNDPROTEOME]`
Overwrites `[-s SPECIES]`. Use user specified background proteome by providing an absolute path to the peptide list. Background proteome file should be a tab-delimited file that contains two columns named “Proten_Name” and “Sequence”.

`[-n BLIBNAME]`
The output spectral library (.blib) name

`[-m MINELUTION]`
How many times (MS/MS) at least do you expect that each peptide was sampled (default 6)

`[--ionTypes IONTYPES]`
Fragment ion types (b, y, or by) (suggesting HCD: y, reCID: by)

** Other optional parameters **

Specify additional memory requested for each job. Recommended if you are querying a lot of proteins/peptides.

```
[--pecanMemRequest PECANMEMREQUEST]
[--percolatorMemRequest PERCOLATORMEMREQUEST]
```

PECAN peptide settings (default 2-3)

```
[--minCharge MINCHARGE]
[--maxCharge MAXCHARGE]
```

PECAN experiment/extraction settings

`[--GPF GPF]`
Divide isolation window list by the number of injections (Only works for multiple injections that divides the isolation window list into equal parts)

`[--jointPercolator]`
Run Percolator jointly with results from all input mzML files

`[-p MS1PPM]`
MS extraction window ± ppm (default 10)

`[-q MS2PPM]`
MS/MS extraction window ± ppm (default 10)

`[-x MAXELUTION]`

How many times (MS/MS) at most do you expect that each peptide was sampled? (Default is greedy [-m MINELUTION] + 1)

`[-t TOPX]`
Default only top one evidence per query is reported for feature generation

`[--SCIEX]`
Include this flag if your mzML files were converted by AB SCIEX MS Data Converter

`[-e EXTENTION]`
Default raw (use wiff for SCIEX data)

`[--fido]`
Include protein inference using FIDO in percolator

`[--IDOTP]`
An option that can be used to filter peptide evidence prior to scoring if the observed isotopic pattern from the MS1 data does not match the theoretical pattern calculated based on the peptide sequence. The score used for filtering is a weighted dot product between the observed and theoretical MS1 isotope pattern. For users who wish to apply the filter, any evidence with the weighted-average isotopic dot product less than the IDOTP cutoff will be disqualified. By default, PECAN does not use any MS1 filtering (default is 0.0).

STEP 4: Run pecanpie and submit the jobs to cluster

Here is an example of pecanpie for analyzing a HeLa DIA dataset acquired on a Q-Exactive with 10 *m/z*-wide isolation windows.

```
[user@server]$ pecanpie -w 10 -s human -o 20160101_HeLa_DIA_pecan
20160101_QE_10mzDIA_HeLa_mzmls.txt query_human_mt_tf_heavystd.txt
isoW_10mz_400to450.txt
```

The optional parameters can be in any order as long as each flag (e.g. `-w`) is followed by the corresponding parameter. The three input files have to be in the correct order. The above command will generate a folder named 20160101_HeLa_DIA_pecan that contains three subfolders: pecan, percolator and pecan2blib, as such:

```
20160101_HeLa_DIA_pecan/
|-- pecan
|   |-- log
|   |-- pecan_HeLa_A1.job
|   |-- pecan_HeLa_A2.job
|   |-- pecan_HeLa_A3.job
|   `-- pecan_HeLa_A4.job
|-- pecan2blib
|   |-- log
|   `-- pecan2blib.job
|-- pecanpie.call.log
|-- percolator
|   |-- log
|   |-- percolator_HeLa_A1.job
|   |-- percolator_HeLa_A2.job
|   |-- percolator_HeLa_A3.job
|   `-- percolator_HeLa_A4.job
`-- run_search.sh
```

Each subfolder contains the jobs the tool will run and a log folder for recording runtime errors. To submit the search, navigate to the created folder and execute the **run_search.sh** by:

```
[user@server]$ ./20160101_HeLa_DIA_pecan/run_search.sh
```

Or by:

```
[user@server]$ cd 20160101_HeLa_DIA_pecan
[user@server]$ ./run_search.sh
```

Your PECAN jobs have been submitted. Please use SGE cluster tools to check for your job status. For example:

```
[user@server]$ qstat -u user
```

Your final results will be in pecan library format
named **20160101_HeLa_DIA_pecan_library.blib** inside of the **pecan2blib** subfolder.

STEP 5: PECAN outputs

Three important files are generated after PECAN analysis. The first is called a feature file in tab-delimited format, where all PECAN reported evidence of detection (i.e. peaks) and the corresponding auxiliary scores for both targets and decoys are recorded. The second is the percolator result file in tab-delimited format, where percolator-assigned *q*-values and posterior error probabilities (PEPs) for all targets are recorded. The third is a spectral library in the .blib format, a simple sqlite database, where the PECAN reported evidence of detection with default *q*-value < 0.01 are recorded (precursor m/z, charge, fragment m/z, retention time). Currently the spectra in PECAN output .blib contains only uniform intensities. This is in part because we intend to add in transition selection (of fragment ions) based on interference in the near future.