Procedures for using PepQuery

Contents

1 Introduction

PepQuery (www.pepquery.org/) is a peptide-centric search engine for novel peptide identification and validation. It is available as a standalone application as well as a web-application. Users can find the detailed procedures for using PepQuery in this document. Completing all examples will take about less than 45 minutes.

2 Required resources

2.1 For web server

Web server address: http://pepquery2.pepquery.org/ .
Input data for testing: please find the input sequences data in appendix section.
Support web browser:
Windows: Google Chrome 56.0 or later, Internet Explorer 10.

Mac: Google Chrome 56.0, Safari 10.0 or later.
We strongly recommend upgrading to the latest version of the supported browsers.

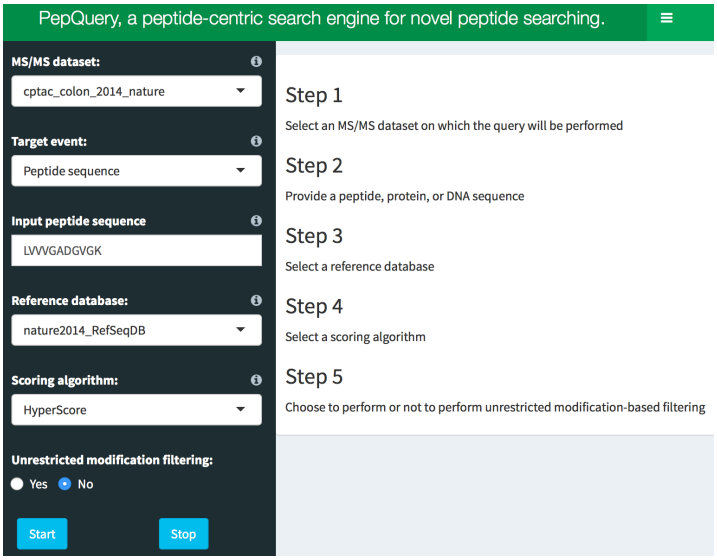
2.2 For standalone version

Software download: please download the latest version of PepQuery: http://pepquery.org/data/PepQuery_v1.0.0.tar.gz .
Java version: 1.8 or later
OS: Windows, Linux and Mac OS
Hardware: 2 CPUs (more is better), 4 Gb memory
Input data for testing: The testing data is included in the software package (http://pepquery.org/data/PepQuery_v1.0.0.tar.gz).

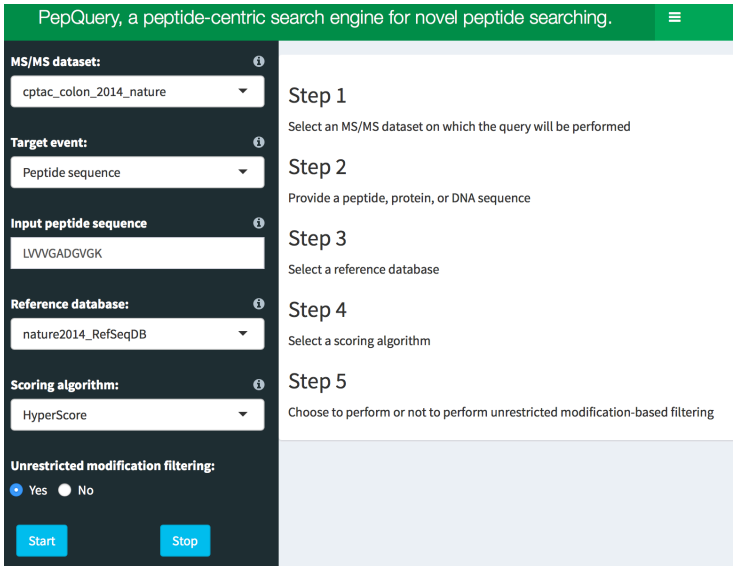
3 Using the web server version of PepQuery

3.1 Taking a novel peptide as input

1. Open the web server: http://pepquery2.pepquery.org/. The web server looks like below:



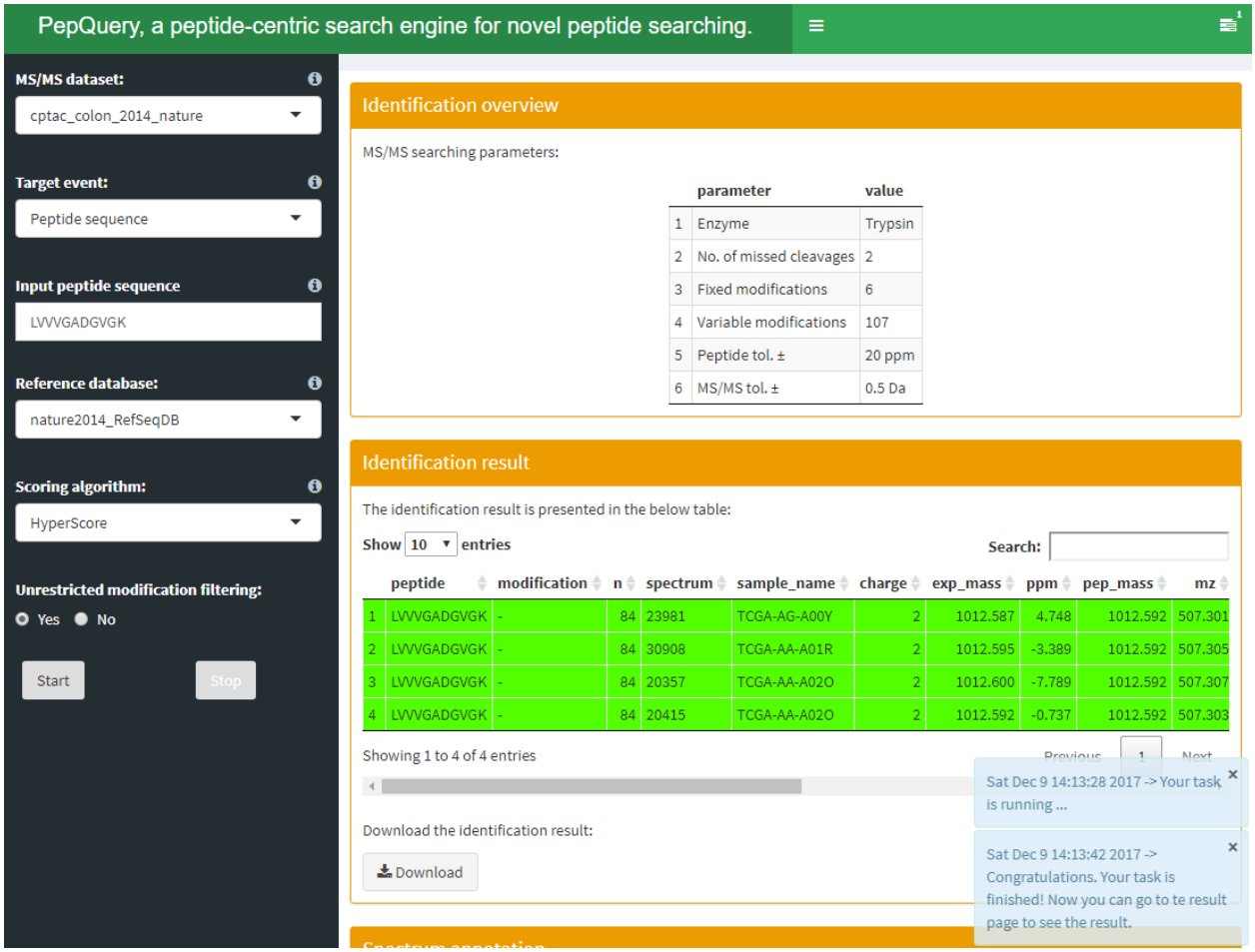
2. Set parameters. Please find detailed description of each parameter in the website of PepQuery: http://pepquery.org/document.html . Please set all the parameters like below:



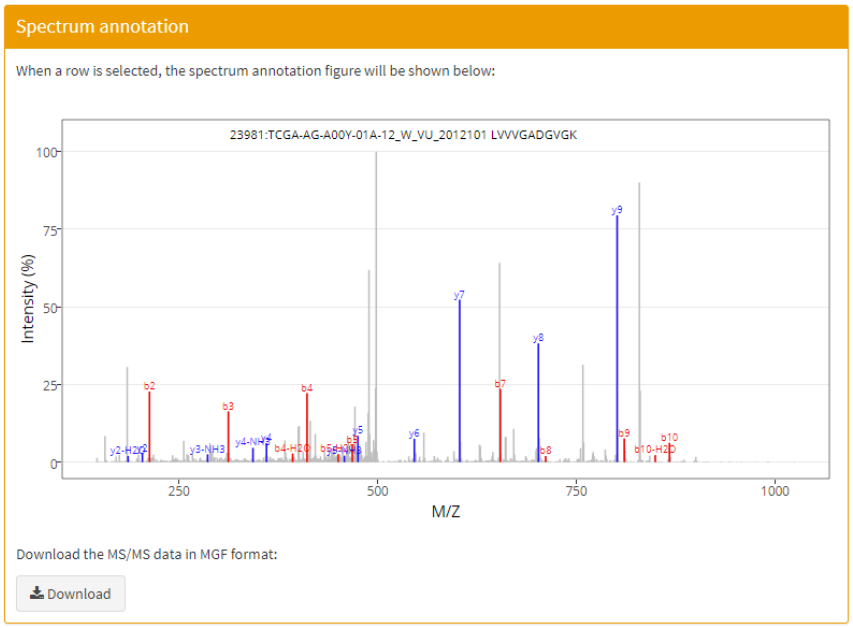
Please select "Yes" for unrestricted modification filtering so that unrestricted modification filtering is used.

The MS/MS dataset used in this testing is a large-scale colon tumor dataset which contains MS/MS data from 95 colon tumor samples. This MS/MS dataset contains 12,941,421 MS/MS spectra. Please go to this web page to find more information about this dataset if interested: http://www.pepquery.org/document.html#webdataset.

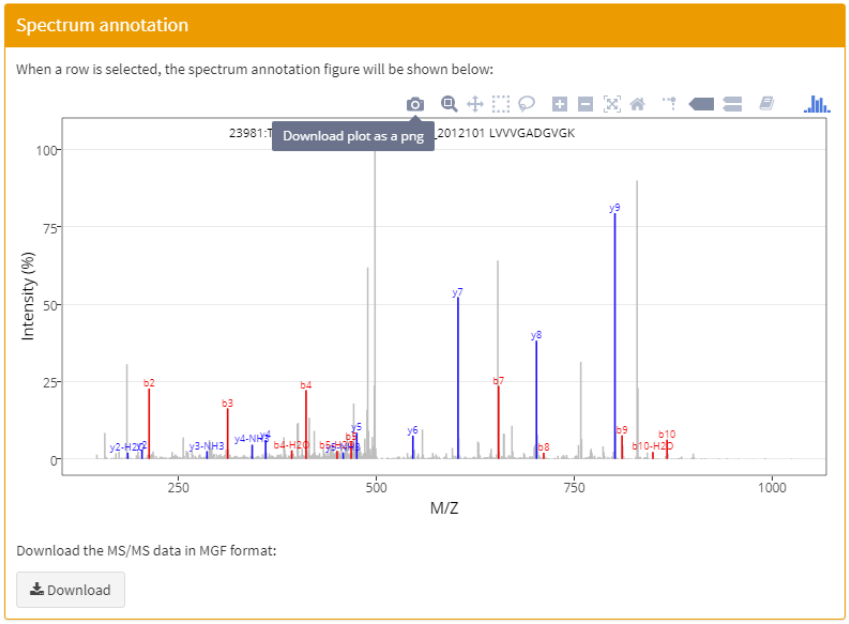
3. Click "Start" to submit the task. Analysis will be completed in about 30 seconds, and the search result will look like below:



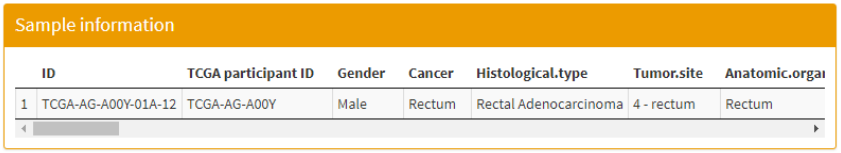
4. From the result page, users will find a table listing a total of 4 spectra with confident match (p-value <= 0.01 and n_ptm = 0, n_ptm is the number of better matched modification peptides when performing the unrestricted modification searching.) to the input novel peptide. If the value in the column of "Confident" is "Yes", it means that the identification is confident with cutoff of p-value <= 0.01 and n_ptm = 0. Users can download the identification result table by clicking the "Download" button. When users click a row in the table, the annotated spectrum of this peptide-spectrum match (PSM) will be shown in the "Spectrum annotation" panel as shown below:



Click the “Download” button in this panel will download the MS/MS spectrum of this match in MGF format. Click the camera button as shown below will download the annotated spectrum in the png format:

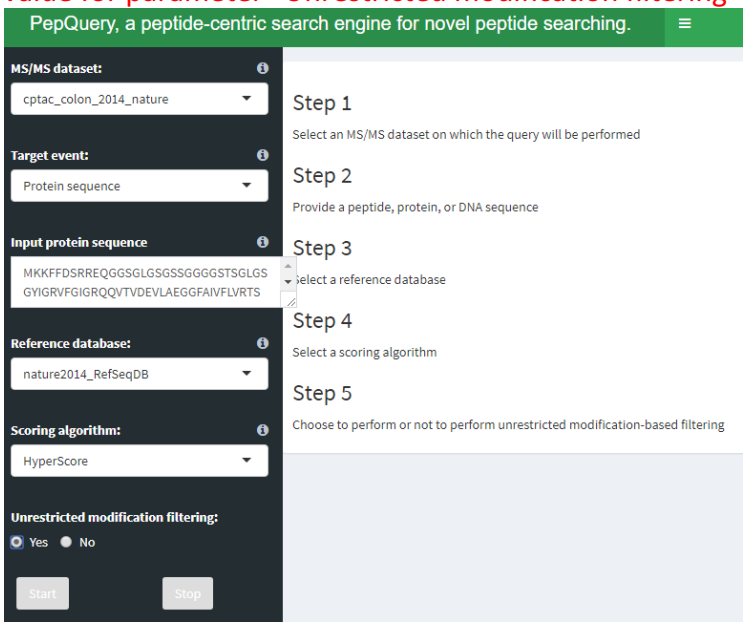


Sample information for the identified spectra can be found in the panel of “Sample information” as shown below:

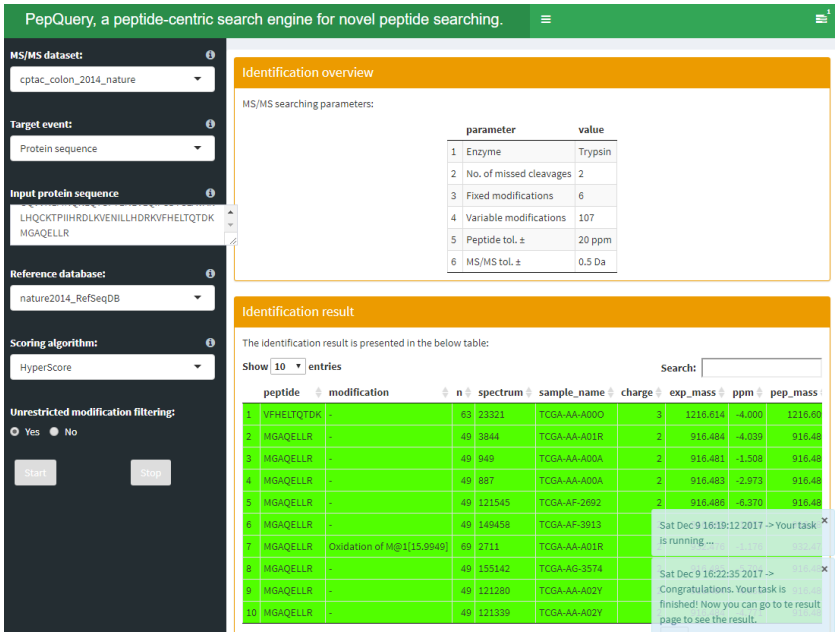


3.2 Taking a novel protein sequence as input

Please use the protein sequence in appendix section as an example. Because the input is a protein sequence, so users need to select the "Target event" as "Protein sequence". Actually, after users set the value of "Target event" as "Protein sequence", the default sequence in the "Input protein sequence" is the testing novel protein sequence used in this document. After setting the parameters as shown below, a click on the "Start" button will submit the job. Please make sure that the value for parameter "Unrestricted modification filtering" is set as "Yes".

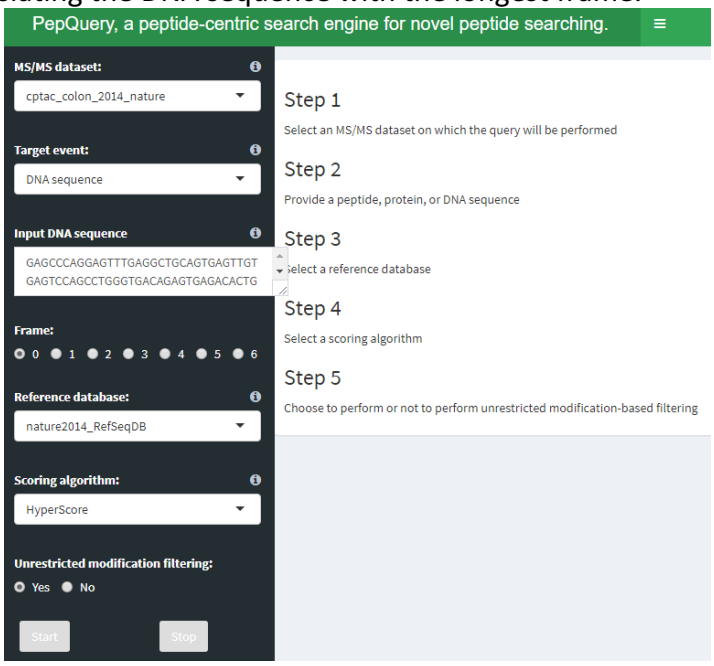


The job will take about 3 or 4 minutes. The result page will look like below. In total, 35 PSMs are identified, including 26 confident PSMs corresponding to 3 unique peptides (MGAQELLR, VFHELTQTDK and VENILLHDRK, highlighted in green color) and 9 PSMs that are not confident matches. The cutoff here is the same with the previous section (p-value <= 0.01 and n_ptm = 0).

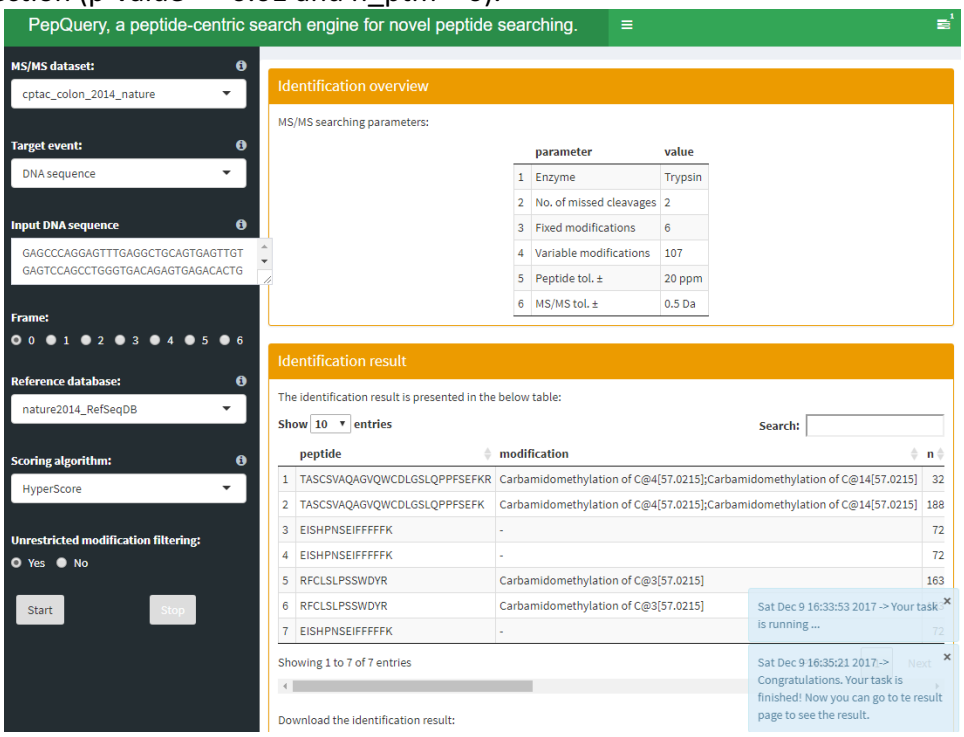


3.3 Taking a novel DNA sequence as input

Please use the DNA sequence in appendix section as an example. Because the input is a novel DNA sequence, so users need to select the "Target event" as "DNA sequence". Actually, after users set the value of "Target event" as "DNA sequence", the default sequence in the "Input DNA sequence" is the testing novel DNA sequence used in this document. After setting the parameters as shown below, a click on the "Start" button will submit the job. Please make sure that the value for parameter "Unrestricted modification filtering" is set as "Yes". Selecting 0 for "Frame" means translating the DNA sequence with the longest frame.



The job will take about 2 or 3 minutes. The result page will look like below. Several matches can be found but none of them are confident matches. The cutoff here is the same with the previous section (p-value <= 0.01 and n_ptm = 0).



4 Using the standalone version of PepQuery

We recommend users to test the standalone version in Linux system. However, it should work in Windows. After download the standalone version of PepQuery from this link: http://www.pepquery.org/data/PepQuery_v1.0.0.tar.gz, please unzip it to a folder:
In Linux system, users can use the following command line to unzip:

tar xvzf PepQuery_v1.0.0.tar.gz

Then the folder of "PepQuery_v1.0.0" looks like below:

|-- **example/**
| |-- create_proteindb_index.sh
| |-- **iPRG2015**/
| | |-- iPRG2015_no6.fasta
| | |-- JD_06232014_sample1-A.mgf
| | `-- peplist.txt
| |-- run_input_a_peptide.sh
| |-- run_input_a_protein.sh
| |-- run_input_peptidelistfile.sh
| |-- run_input_vcf.sh
| |-- test_PGA.R

```
|   `-- vcf
|       |-- Ref_Hsapiens_GRCh37
|       |   |-- exon_anno.RData
|       |   |-- hg19_refGeneCDS.fa
|       |   |-- hg19_refGenePro.fa
|       |   |-- ids.RData
|       |   |-- procodingseq.RData
|       |   |-- proseq.RData
|       |   |-- splicemax.RData
|       |   `-- txdb.sqlite
|       |-- TCGA-A6-3807-01A_f0106.mgf
|       `-- test.vcf
|-- lib/
|-- modification_list.txt
|-- pepquery.jar
`-- PDV/
```

The file pepquery.jar is the main program of PepQuery. The data in the example folder will be used by this testing procedures. One dataset (**iPRG2015**) is from the iPRG-2015 study (http://pubs.acs.org/doi/abs/10.1021/acs.jproteome.6b00881). The other one (**vcf**) is from the CPTAC study (https://www.ncbi.nlm.nih.gov/pubmed/25043054). The folder "PDV" contains the PDV software (http://pdv.zhang-lab.org) that can be used to visualize PepQuery results.

Firstly, in order to speed up the database searching, we can build a SQL-based index database for each reference protein database using the following command line (create_proteindb_index.sh):

```
# Change directory to the example folder
cd example/
# Please note that all the following command lines will be run in the example folder
java -Xmx2G -cp ../pepquery.jar main.java.util.createProDB -db iPRG2015/iPRG2015_no6.fasta -fixMod 6 -varMod 107,142,143,15 -maxVar 3 -e 1 -c 2
```

But please note this step is optional. It will take about 4 minutes. The description of the above parameters is listed below:

Parameter	Description
-db	A reference protein database
-fixMod	Fixed modification. Each modification is represented by a number. Please find the complete modification list in file "modification_list.txt". "6" is Carbamidomethylation of C.
-varMod	Variable modification. 15: Acetylation of peptide N-term; 107: Oxidation of M; 142: Deamidation of Q; 143: Deamidation of N

-maxVar	Max number of variable modifications per peptide
-e	Enzyme used for protein digestion. 1:Trypsin (default), 2:Trypsin (no P rule), 3:Arg-C, 4:Arg-C (no P rule), 5:Arg-N, 6:Glu-C, 7:Lys-C
-c	The max missed cleavages, default is 2

4.1 Taking a novel peptide as input

Please find the detailed description for each parameter of PepQuery here: http://www.pepquery.org/document.html#saparameter . Users can use "-pep" to specify the input peptide (run_input_a_peptide.sh).

java -jar -Xmx2G ../pepquery.jar -db iPRG2015/iPRG2015_no6.fasta -fixMod 6 -varMod 107,142,143,15 -maxVar 3 -cpu 4 -fragmentMethod 1 -itol 0.02 -tol 10 -ms iPRG2015/JD_06232014_sample1-A.mgf -pep AVVQDPALKPLALVYGEATSR -m 1 -minScore 10 -n 1000 -um -o out

The above search will take about less than 10 seconds. One spectrum is identified for the input peptide with cutoff of p-value<=0.01 and n_ptm==0. The final result is saved in the file out/psm_rank.txt. Users can find the description for each column of this file here: http://www.pepquery.org/document.html#saoutput.

4.2 Taking a novel protein sequence as input

Users can use "-i" and "-t" to specify a protein sequence as input (run_input_a_protein.sh):

java -jar -Xmx2G ../pepquery.jar -db iPRG2015/iPRG2015_no6.fasta -fixMod 6 -varMod 107,142,143,15 -maxVar 3 -cpu 4 -fragmentMethod 1 -itol 0.02 -tol 10 -ms iPRG2015/JD_06232014_sample1-A.mgf -m 1 -minScore 10 -n 1000 -um -o out -i MSHHWGYGKHNGPEHWHKDFPIANGERQSPVDIDTKAVVQDPALKPLALVYGEATSRRMVNNGHSFNV EYDDSQDKAVLKDGPLTGTYRLVQFHFHWGSSDDQGSEHTVDRKKYAAELHLVHWNTKYGDFGTAAQQP DGLAVVGVFLKVGDANPALQKVLDALDSIKTKGKSTDFPNFDPGSLLPNVLDYWTYPGSLTTPPLLESVTWIV LKEPISVSSQQMLKFRTLNFNAEGEPELLMLANWRPAQPLKNRQVRGFPK -t 1

The above search will take about 10 seconds. The final result is saved in the file out/psm_rank.txt. In total, 7 PSMs for 6 unique peptides are identified with cutoff of p-value<=0.01 and n_ptm==0. Users can also take a DNA sequence as input.

4.3 Taking multiple peptide sequences as input

If users have multiple novel peptides, a file with multiple peptides can be used for parameter "-pep". Below is an example (run_input_peptidelistfile.sh):

java -jar -Xmx2G ../pepquery.jar -db iPRG2015/iPRG2015_no6.fasta -fixMod 6 -varMod 107,142,143,15 -maxVar 3 -cpu 4 -fragmentMethod 1 -itol 0.02 -tol 10 -ms

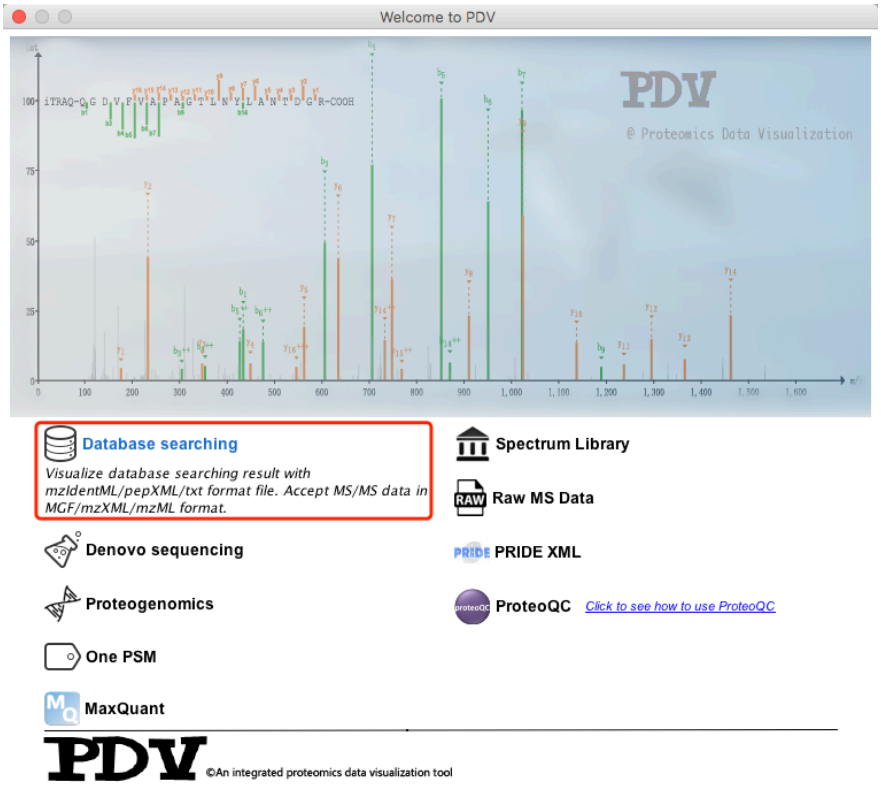
iPRG2015/JD_06232014_sample1-A.mgf -pep iPRG2015/peplist.txt -m 1 -minScore 10 -n 1000 -um -o out

In the input file “iPRG2015/peplist.txt”, it contains 4 peptides which do not exist in the reference protein database (iPRG2015/iPRG2015_no6.fasta). Please note each row corresponds to one peptide. The content of file “iPRG2015/peplist.txt” looks like below:

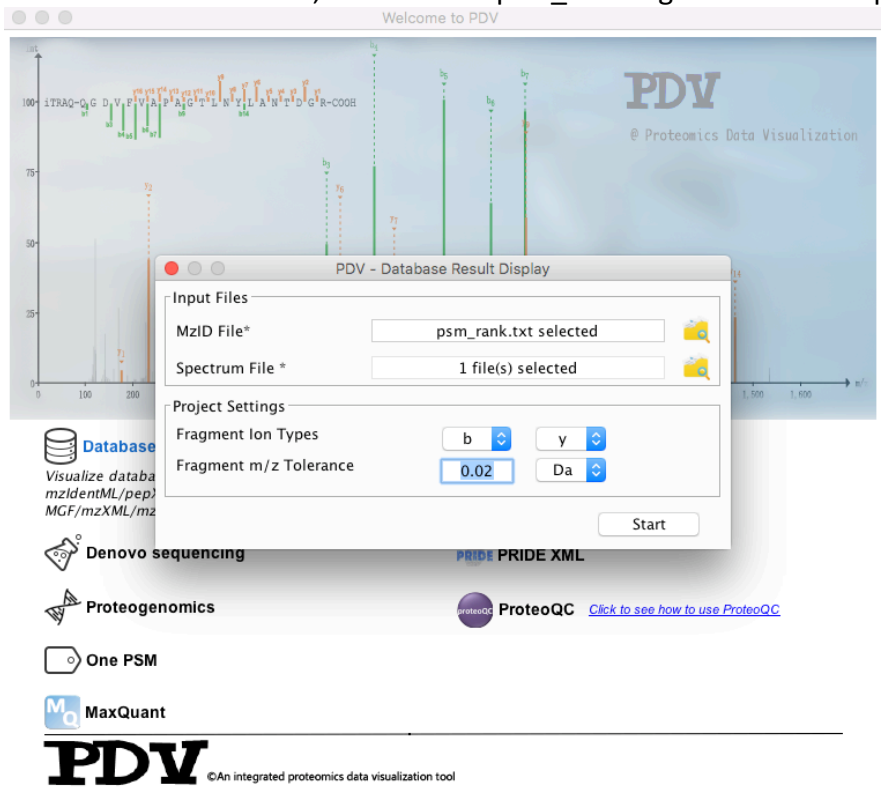
LLSYVDDEAFIRDVAK
AVVQDPALKPLALVYGEATSR
DFPIANGER
VGDANPALQK

The above search will take about 10 seconds. The final result is saved in the file out/psm_rank.txt, including 5 spectra that are confidently matched to the 4 peptides with cutoff of p-value<=0.01 and n_ptm==0. Please note that the result file (psm_rank.txt) can be visualized by PDV (http://pdv.zhang-lab.org) using the following step by step procedure:

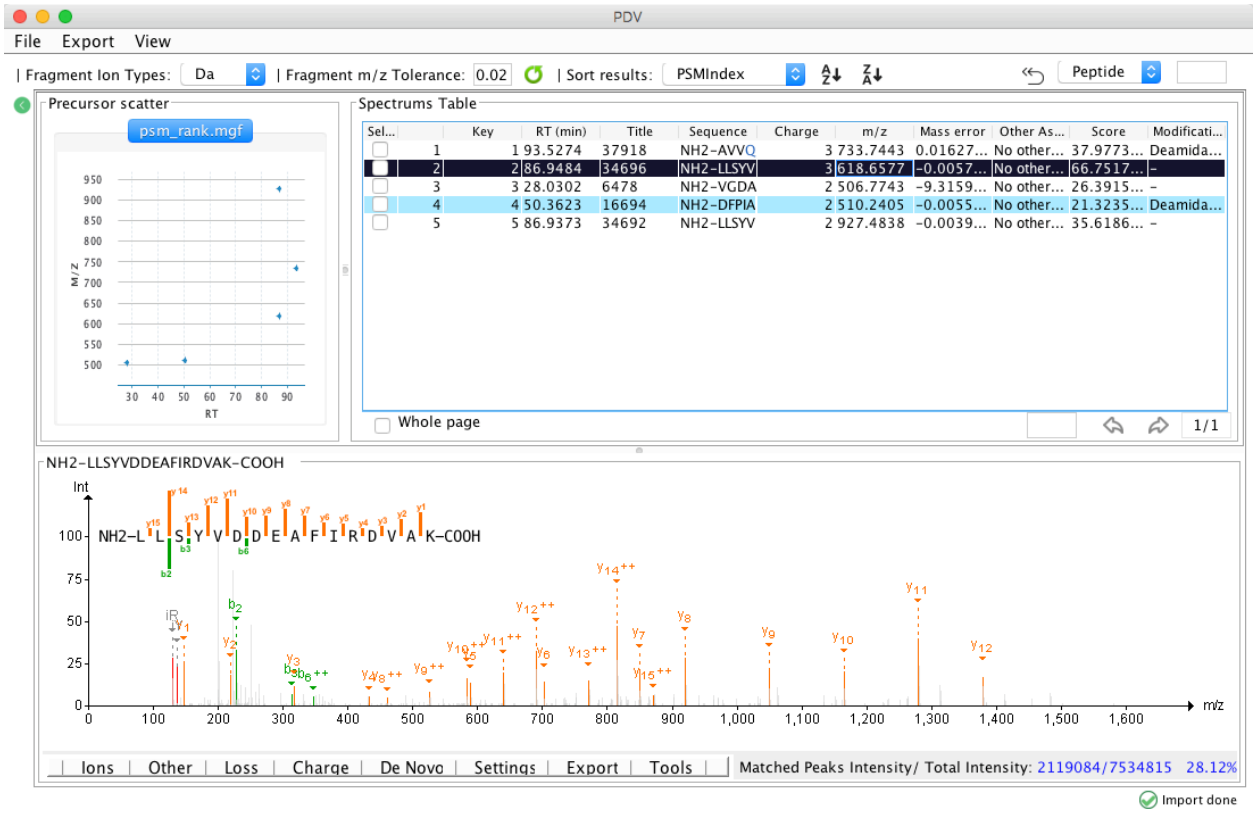
(1). Please launch PDV by double clicking the file “PDV-1.2.beta.jar” in the “PDV” folder. The main interface is shown below:



(2). Click “Database searching” in the above panel, then import data and set parameters as shown below. For visualization, only two files from the PepQuery output are required. The file “psm_rank.txt” is used for “MzID File”, and the file “psm_rank.mgf” is used for “Spectrum File”.



(3). Click on the “Start” button in the above panel. After about 1 or 2 seconds, users will see the visualization panel as shown below:



4.4 Taking VCF/BED/GTF as input

Processing VCF/BED/GTF files is complicated. Currently, PepQuery relies on PGA (http://bioconductor.org/packages/PGA/) to translate the genomics information in VCF/BED/GTF to protein sequences. If users want to take VCF, BED or GTF as input, please install R (https://www.r-project.org) and the PGA package and the BSgenome.Hsapiens.UCSC.hg19 package (http://bioconductor.org/packages/BSgenome.Hsapiens.UCSC.hg19/). Please make sure that the R environment in users' computer is compatible with PGA. Then users need to prepare annotation data according to the instruction in the manual of PGA (http://bioconductor.org/packages/devel/bioc/vignettes/PGA/inst/doc/PGA.pdf). Users need to set parameters "-i", "-anno" and "-t" when they want to take VCF, BED or GTF as input.

After R, PGA and BSgenome.Hsapiens.UCSC.hg19 are installed in users' computer, please make sure that R has been added in the environment path. Please run the script test_PGA.R in example folder to make sure PGA works well in users' computer:

Rscript test_PGA.R

If the following three files are generated in the "out" folder, then it indicates PGA works well:

pga_snv.fasta
pga_snv.tab
pga_txFinder.fasta

Below is an example to run PepQuery with taking VCF file as input (run_input_peptidelistfile.sh):

java -jar -Xmx2G ../pepquery.jar -i vcf/test.vcf -anno vcf/Ref_Hsapiens_GRCh37/ -t 3 -cpu 4 -db vcf/Ref_Hsapiens_GRCh37/hg19_refGenePro.fa -varMod 107 -fixMod 6 -tol 10 -itol 0.6 -o out -m 1 -ms vcf/TCGA-A6-3807-01A_f0106.mgf

The above search will take about 1 minutes. The final result is saved in the file out/psm_rank.txt, there is no confident match with cutoff of p-value <= 0.01.

5 Appendix

5.1 A novel peptide

LVVVGADGVGK

5.2 A novel protein

MKKFFDSRREQGGSGLGSGSSGGGGSTSGLGSGYIGRVFGIGRQQVTVDEVLAEGGFAIVFLVRTSNGMKC
ALKRMFVNNEHDLQVCKREIQIMRDLSGHKNIVGYIDSSINNVSSGDVWEVLILMDFCRGGQVVNLMNQR
LQTGFTENEVLQIFCDTCEAVARLHQCKTPIIHRDLKVENILLHDRKVFHELTQTDKMGAQELLR

5.3 A novel DNA sequence

GAGCCCAGGAGTTTGAGGCTGCAGTGAGTTGTGAGTCCAGCCTGGGTGACAGAGTGAGACACTGTCTC
CAAAAATATAATAATAAAAATAAAGCCTGTATCTCAATGGGAAGATTTCTGCTAAGGCAGTTCAACTCAC
TGGAAAGAACTGCTGCATTAGGTTTCAACTTTAATGCTTTCTGTTACTTCTAGTAAAGGTTAAGTGATTTT
TCCATTAGCTGTAGAAGTTTGGAGAGCTATTTACCAAGCCAGATCAATGATTTAAAAATTATTGGAAATT
CATCTAAGAATCAAGTCTGAATGCCCAATTCTATTGCGGTTGATTAGGTGTGATATTCTTTAAAGTTCAG
GAATATTGGCAGTAAAAAATGAGCAGCTACTTTTCAATACTTTGTCCTTTTTTGGTGGTCTCTGCCTATTT
CAAATGTCCTGATCAAAAGATAAATAATTGGCACTGTGCCAAGGTTTGGTTTTCCAACTAAGGTTTCAAC
TGTGCCAGAAACCTATGTCCTTCACTTTGGTGGATGCTAAATGTTATTCTAAGAATATGCTTTTTCCCAAT
TCTCCTTTCTGATTTTTATGTATTAGTGGATGCAAAATTGTCTTTCTAGTTGAATGAATAATTTCGGCTAAT
GCACGTGGAACTTTGCACCCCAGATTCTTCCCATGGTCATTATCAAGTGAAGCCCTCAAAAACATGAGCG
AAGAGCCTAGAAATACTCAGGGAGATTTCTCACCCCAACTCAGAAATTTTTTTTTTTTTTTCAAGACGGC
GTCTTGCTCTGTCGCTCAGGCGGGAGTGCAGTGGTGTGATCTCGGCTCACTGCAACCTCCATTTTCCGAG
TTCAAGCGATTCTGCCTCAGCCTCCCGAGTAGCTGGGATTATAGGCACACACCACCGCGCCTGGCTAATT
TTTGTATTCTTAGTAGAGATGGGGTTTCACCATGTTGGCCAG