

TagGraph

Defining complex proteomes with string

TagGraph is the first algorithm of its kind which leverages the efficiency of FM-indexed sequence databases to rapidly and sensitively interpret large mass spectrometry data sets. One major advantage of this approach is its ability to discover and localize post-translational modifications and other amino acid alterations within peptide sequences, without anticipating them *a priori*.

The following describes basic installation and operation procedures for the TagGraph software, version 1.6.

SET UP LINUX ENVIRONMENT	1
I. INSTALL TAGGRAPH.....	3
II. INDEX FASTA FILE FOR FM-INDEX SEARCH.....	5
III. GATHER INPUT FILES	6
A. PROCESS MASS SPECTRA DATA FILES	6
B. DE NOVO SEARCH RESULTS.....	6
1. <i>CSV file format</i>	8
2. <i>pepXML file format</i>	8
IV. SET UP TAGGRAPH RUNNING PARAMETERS	9
A. TAGGRAPH PARAMETER FILE:	9
V. RUN TAGGRAPH	12
VI. OUTPUT DESCRIPTION	13
VII. KNOWN ISSUES	17

Set up linux environment

1. If you are already running a compatible form of linux (so far, we've tested CentOS v 7), skip to step 1.5
2. Install virtual machine infrastructure (E.g., hyper-v on windows 10)
 - a. Follow general instructions here:
<https://www.tenforums.com/tutorials/2087-hyper-v-virtualization-setup-use-windows-10-a.html#Part1>
 - b. Follow supplemental instructions here:
<https://www.tenforums.com/tutorials/2291-hyper-v-vm-install-centos-linux-windows-10-a.html#Part1>
3. Download CentOS operating system

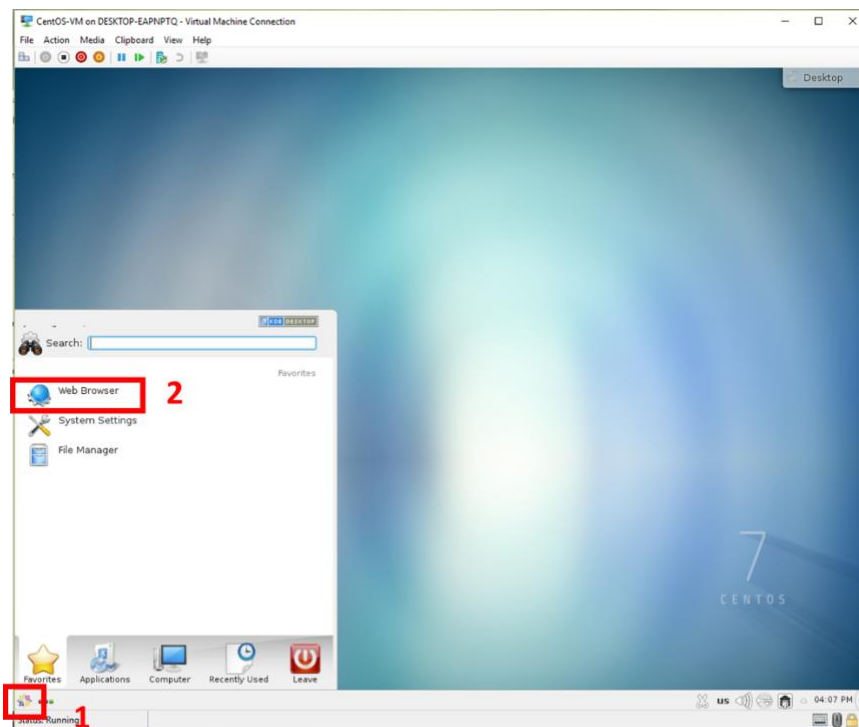
<https://www.centos.org/download/>

NB: Full download ~ 4.1 GB as of CentOS7.1611, March 21, 2017

4. Create CentOS virtual machine by continuing with the tutorial noted in Step 2b.

NB: be sure to follow instructions pertaining to internet connectivity, as this will be needed to install prerequisites for TagGraph, and TagGraph itself.

1. If your host computer is behind a firewall, you will need to register your virtual machine with the network. This can probably be accomplished by pointing the KDE web browser to any web page.



- a. Click Kickoff button (1)
- b. Select Web Browser (2)
- c. Enter any valid address in URL field (e.g., www.stanford.edu)

d. Register virtual computer with network if prompted.

5. Install non-standard packages with “yum” or pip commands which TagGraph will need

A. pip (python installer)

```
[jelias@sr17-fa2fb197bf ~]$ sudo easy_install pip
```

B. lxml, numpy, pyteomics

```
[jelias@sr17-fa2fb197bf ~]$ sudo pip install lxml numpy pyteomics
```

C. sqlalchemy

```
[jelias@sr17-fa2fb197bf ~]$ sudo pip install sqlalchemy
```

D. pandas

```
[jelias@sr17-fa2fb197bf ~]$ sudo pip install pandas
```

E. pympler

```
[jelias@sr17-fa2fb197bf ~]$ sudo pip install pympler
```

F. MySQL-python

```
[jelias@sr17-fa2fb197bf ~]$ sudo yum install MySQL-python
```

Enter ‘y’ when prompted

G. networkx

```
[jelias@sr17-fa2fb197bf ~]$ sudo pip install networkx
```

H. pymzml

```
sudo pip install pymzml
```

I. matplotlib

```
sudo pip install matplotlib
```

I. Install TagGraph

A. Download TagGraph program

A. Use web browser to navigate to taggraph repository:

<https://sourceforge.net/projects/taggraph/>

B. Download:

- i. Source code (TagGraph.1.6.tar.gz
- ii. Sample input (for testing): sampleInputFiles.2.tar.gz
- iii. Sample output (for verification): sampleOutputFiles.tar.gz

C. Open Konsole terminal:

- i. Kickoff button -> Search for “Konsole”
- ii. Select Konsole



B. Verify software was installed correctly

A. Verify taggraph file was saved: by typing “ls”

```
[jelias@sr17-fa2fb197bf ~]$ ls
Desktop  Downloads  Pictures  TagGraph-1.0.0.tar.gz  Videos
Documents Music      Public   Templates
```

B. Unzip TagGraph file:

```
[jelias@sr17-fa2fb197bf ~]$ tar -xzf TagGraph-1.0.0.tar.gz
```

C. Navigate to the new folder that was created

D. Verify all required modules have been installed:

Run python checkPythonPackages.py

```
sqlalchemy exist  
pympler exist  
numpy exist  
networkx exist  
pymzml exist  
MySQLdb exist  
pyteomics exist
```

- a. if any packages are missing, run the indicated command to install it or seek assistance from your system administrator.

II. Index fasta file for FM-index search

- A. Download or transfer valid fasta-formatted protein sequence file
 - a. E.g., the file "human_uniprot_12092014_crap.fasta" used in the TagGraph manuscript, and included in the SampleOutputFiles.tar.gz download
 - b. Download other fasta-formatted protein sequences from <http://www.uniprot.org>
- B. Run FM indexer:

python Build_FMIndex_new.py <fasta location> -o <desired output location; default is to save the fasta file to the fmindexer's working directory>

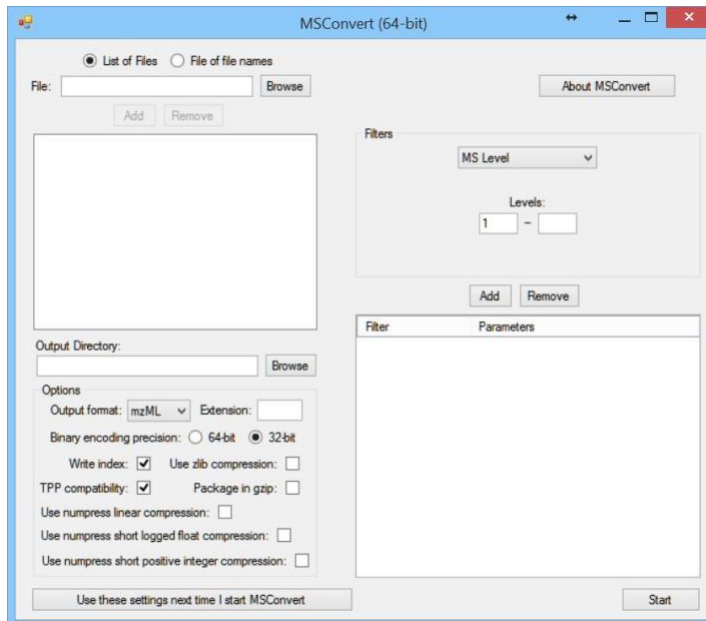
```
building index.
- remapping alphabet.
- creating cumulative counts C[].
- performing bwt.
- sample SA locations.
- creating bwt output.
- create RRR wavelet tree over bwt.
build FM-Index done. (0.021 sec)
space usage:
- remap_reverse: 21 bytes (0.10%)
- C: 1028 bytes (4.66%)
- Suffixes: 1484 bytes (6.72%)
- Positions: 1488 bytes (6.74%)
- Sampled: 1276 bytes (5.78%)
- T_bwt: 16512 bytes (74.78%)
input Size n = 23738 bytes

index Size = 22081 bytes (0.93 n)
writing FM Index to file 'protein.fm.1'
```

III. Gather input files

A. Generate peak lists from raw mass spectra data files

TagGraph natively interprets mzXML and mzML formats. Several tools are available for converting vendor-specific raw data formats to either of these standards. We have tested and recommend using msconvert, a component of the proteowizard suite of tools:
<http://proteowizard.sourceforge.net/downloads.shtml>



1. Select raw data file(s) to be converted
2. Select local output directory
3. Select 32-bit, and check write index, TPP compatibility. Do not check Use zlib compression or Package in gzip
4. Click “Start”

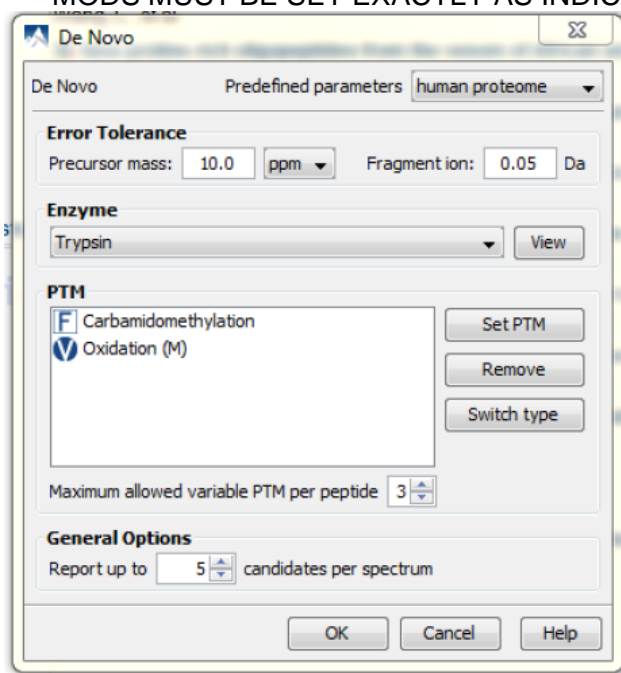
B. Generate de novo search results

TagGraph should in theory be able to consider any de novo sequence as input. We have found that PEAKS (ver 5 or later) produces strong results, in part due to its ability to consider a wide mass range and non-tryptic peptide options. TagGraph currently natively parse de novo results in the PEAKS .csv output file format (detailed below) or the pepXML standard. Converting output from other de novo sequencing engines to either of these formats should be compliant with TagGraph operation, provided that the following search parameters are approximately followed:

PEAKS de novo search parameters:

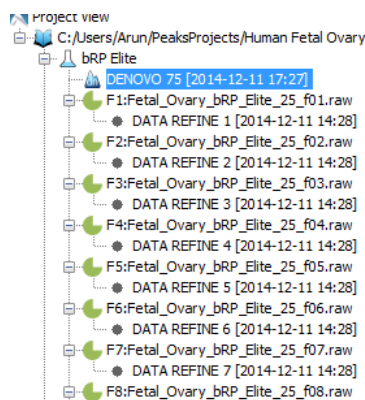
1. Set enzyme to enzyme used in your sample (only Trypsin and LysC supported now)

MODS MUST BE SET EXACTLY AS INDICATED!



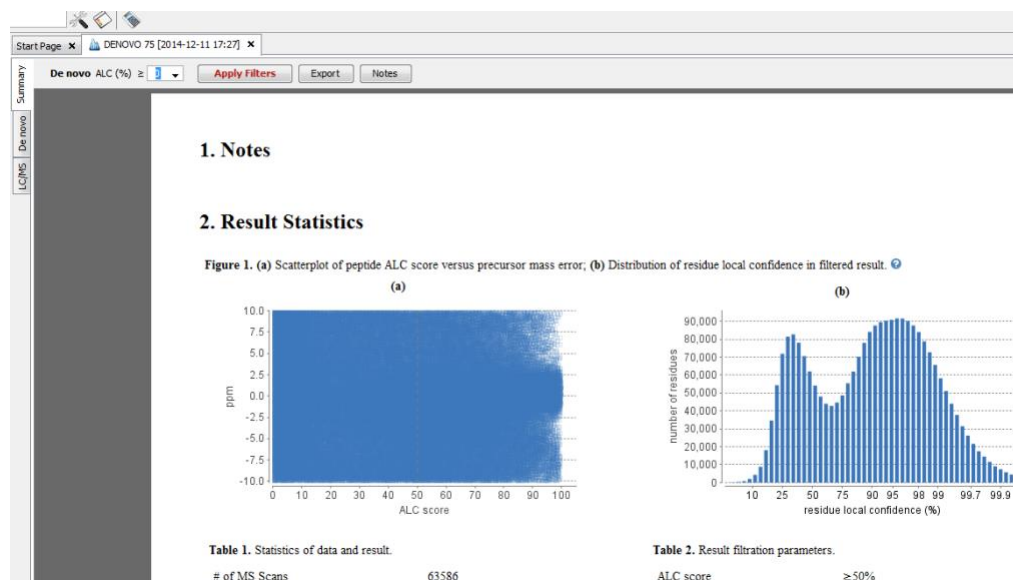
Export Step 1:

Double click the DENOVO entry under the project to open the de novo results



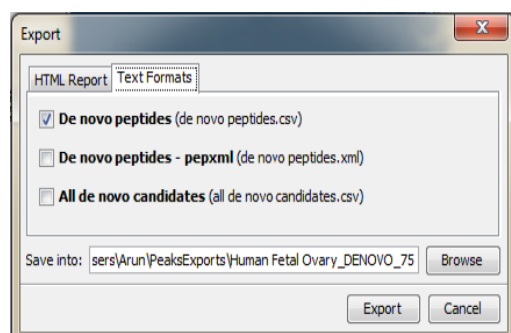
Export Step 2:

Change the ALC cutoff from 50 to 0 and then click the “Apply Filters” button (in red). This is important so that PEAKS exports all de novo results, not just the ones it thinks are high quality.



Export Step 3:

Click the “Export” button next to “Apply Filters”. Export using the settings shown here, upload the “de novo peptides.csv” file to the web server along with the corresponding mzXMLfile and watch the magic happen.



1. CSV file format

See PEAKS\de_novo_peptides_mzXML.csv within SampleInputFiles.2.tar.gz tarball for example.

2. pepXML file format

NB: If raw (as opposed to mzXML or mzML) data files are searched directly by de novo software, you will need to edit the de novo output file (.csv or .pepXML) so that the file extensions they list match the TagGraph input (mzXML or mzML).

e.g., for Thermo raw files:

sed 's/\.raw/\.mzXML/g' original de novo file > altered de novo file

IV. Set up TagGraph running parameters

- A. TagGraph parameter file: (see TG\A375mzXML.params in sampleInputFiles.2.tar.gz tarball)

General Settings

[General]

Not used at this time, place-holder for the capabilities to only run the TG step, or the EM step if TG was run previously

runTG = True

runEM = True

If set to True the copies of the input mzML/mzXML input files are removed from the output tree when the run has finished.

cleanInputDataFilesFromOutput = True

cleanIntermediateFiles = False

output file

generatePepXML = False # Boolean: print tab-delimited .txt file only, or pepXML
format too?

outputPerFraction = False # possibly useful for large data sets: produce per-fraction
output in addition to aggregated output across all
fractions?

FDRCutoff = 0.01 # FDR Threshold to apply to output

logEMCutoff = 2 # p-value threshold to apply to output (equivalent to p=0.99
that identification is correct)

DisplayProteinNum = 3 # maximum number of proteins to list which contain
identified peptide

END General Settings

TagGraph Settings

[TagGraph]

An integer number of data files to be used as fractions, and the path to each
(mz[X]ML) data file with 2-digit counter appended to 'fraction' as key

dataDirectory = A375_Dataset/ #give full path to target data set directory

Name to be used for this experiment and its output files and directories. Must not
contain spaces.

ExperimentName = A375

Location of input de novo search results exported from de novo sequencing program in
.csv or .pepXML format.

de_novo = A375_Dataset/de_novo_peptides_mzXML.csv

Path to folder TagGraph will create to store output. Should not already exist.

Output = A375_Dataset/EM_output

Initialization file used to configure TAG-GRAPH. See below for detailed information.

init = resources/TAG_GRAPH_Tryp_CysCarbam_MetOx.ini

Location and name of fminindex to search. This fminindex location and name should
describe the root file name and extension prefix (lacking any numbers).

Fmindex = resources/FMIndices/20141209_UniHUMAN_cRAP_ILEq.fm

Expected standard deviation in ppm error distributions of fragment ions. Recommend 5 for HCD 30,000 resolution

Ppmstd = 10

Maximum absolute deviation (Da) between experimental and database modification mass for TagGraph to consider modification as a candidate match. Recommend 0.1.
Modtolerance = 0.1

Maximum number of times a de novo-produced substring can occur in the protein sequence database for TagGraph to consider it as an unmodified peptide match.
Recommend less than 1,000 for single organism database (i.e., human uniprot) and 5,000 for nr or 6-frame translations
Maxcounts = 400

Maximum number of times a de novo-produced substring can occur in the protein sequence database for TagGraph to consider it as a modified peptide match.
Recommend 200 for single organism database (i.e. human uniprot) and 1,000 for nr or 6-frame translations
modmaxcounts = 200

Location of pickled (python-serialized) unimod dictionary
unimoddict = resources/unimodDict_noLabels_20160724.pck

Path to pickled (python-serialized) probabilistic model file.
model =
resources/AllChargeDist_posOnlyDependence_20150808_HumanProt500000.pck

Path to pickled (python-serialized) model configuration file.
Config = resources/AllChargeDist_posOnlyDependence_20150808.txt
End TagGraph Settings

EM Settings

[EM]

Number of iterations in initial EM overall results. Recommend 20 iterations.
initIterations = 20

Maximum number of expectation maximization iterations for FDR assignment.
Recommend 100 iterations.
maxIterations = 100

Filename Prefix to use for the output EM results files. Must not contain spaces.
resultsPrefix = EM_Results
End EM Settings

B. TagGraph initialization file (see TG\Tag_Graph_Tryp_MetOx.ini in sampleInputFiles.2.tar.gz tarball)

; Add in default values for command line parameters here

[Parameters]

; Add enzyme info here

; Specificity is regular expression with a semicolon used to indicate cleavage site

[Enzyme]

Name: Tryp

Specificity: K|R;.*

; AA_name: AA_1_letter_abbrev AA_3_letter_abbrev AA_Elemental_Comp
AA_monoisotopic_mass AA_avg_mass

; Add any additional amino acids other than the twenty original ones here

[Amino Acids]

Selenocysteine: U Sec C₃H₇NO₂Se 150.953636 151.0388

Pyrrolysine: O Pyr C₁₂H₂₁N₃O₃ 237.147727 237.3018

Xeucine: J Xle C₆H₁₁ON 113.08406 113.1594

; mod_name: AA mod_mass

; use N-Term for N-terminus and C-Term for C-terminus

[Static Mods]

; mod_name: AA(can be list of AAs such as STY, etc.) mod_mass override_static_mod
mod_symbol

; mod_symbol optional and will be chosen automatically if not given

; override_static_mod is either 0 or 1, 1 means add mod_mass to original AA mass, not
statically modified mass

[Diff Mods]

Oxidation: M 15.994915 0 #

V. Run TagGraph

Command to run the taggraph:

cd TagGraph.1.4

python runTG.py pathToYourParameterFile

```
slin@kronos TagGraph.1.4]$ python runTG.py ~/sampleInputFiles/TG/A375mzML.params
**** start TagGraph process: 2017-11-13 14:11:06.535310

*****
** START TAGGRAPH FROM CONFIG FILE **
*****

/home/slin/sampleInputFiles/TG/A375mzML.params

-----
Using Configuration File: /home/slin/sampleInputFiles/TG/A375mzML.params
-----
* Found Required Numeric TagGraph Parameter 'ppmstd' : '10'
* Found Required Numeric TagGraph Parameter 'modtolerance' : '0.1'
* Found Required Numeric TagGraph Parameter 'maxcounts' : '400'
* Found Required Numeric TagGraph Parameter 'modmaxcounts' : '200'
* Found Required Readable File for TagGraph Parameter 'unimoddict' : 'resources/unimodDict_noLabels_20160724.pck'
* Found Required Readable File for TagGraph Parameter 'model' : 'resources/AllChargeDist_posOnlyDependence_20150808_HumanProt50'
* Found Required Readable File for TagGraph Parameter 'config' : 'resources/AllChargeDist_posOnlyDependence_20150808.txt'
* Found Required Readable File for TagGraph Parameter 'init' : '/home/slin/sampleInputFiles/TG/TAG_GRAPH_Tryp_CysCarbam_MetOx.i'
* Found Required Readable File for TagGraph Parameter 'de_novo' : '/home/slin/sampleInputFiles/PEAKS/de_novo_peptides_mzML.csv'
* Found Required Createable Directory for TagGraph Parameter 'output' : '/home/slin/sampleInputFiles/TG/mzML_EM_output'
* Found Required TagGraph Parameter ExperimentName: 'A375MZML'
dataDirectory: /home/slin/sampleInputFiles/mzML/
fileFractionMapping: [('01', 'ath017216_F01.mzML'), ('02', 'ath017217_F02.mzML')]
* Created temporary sym-link Directory for TagGraph mz[X]ML files '/tmp/A375MZML_1731/'
  * Created symLink '/tmp/A375MZML_1731/A375MZML_f01.mzML' to data file '/home/slin/sampleInputFiles/mzML/ath017216_F01.mzML'
  * Created symLink '/tmp/A375MZML_1731/A375MZML_f02.mzML' to data file '/home/slin/sampleInputFiles/mzML/ath017217_F02.mzML'
```

VI. Output description

Output Overview

Taggraph writes the output to the output directory specified within the parameter file.

Description of files and folders in the output

Results files:

Taggraph outputs a run log: runReport.log. The log lists if the taggraph run was successful using three parameters, reported towards the end of this file:

- 1)Spectrum score $x^2 > 1$? Yes: PASS. No: FAIL - Match between high-scoring peptides and spectra is unusually low
- 2)Tag length $x^2 > 1$? Yes: PASS. No: FAIL - Unexpectedly few long de novo alignments with database
- 3)Mod size $x^2 < 1$? Yes: PASS. No: FAIL - TagGraph had to insert far more large modifications than would be expected.

Results are written out to three files listing the peptides assigned to each scan.

results.db – raw output file with no filtering. Open only with appropriate viewer for .db files

<SAMPLE>_TopResults.tdv – unformatted output of top hits after EM-score based filtering

<SAMPLE>_TopResults.txt – Formatted version of the .tdv output, listing out every modification in separate columns. Output fields in this file described below:

Output Fields

File: Sample

ScanF	Charge	Retention Time	Obs M+H	Theo M+H	PPM	PPM Upper Bound	PPM Lower Bound	PPM In Range
FL0016200.mzXML:76771	2	139.03	1035.58	1035.59	-2.8849	1.8	-4.6	Yes
FL0016200.mzXML:84366	2	148.42	1496.82	1496.82	-1.2811	1.8	-4.6	Yes
FL0016200.mzXML:80488	2	144.12	1808	1808.01	-1.846	1.8	-4.6	Yes
FL0016200.mzXML:71335	2	131.81	1172.53	1172.53	-2.9744	1.8	-4.6	Yes
FL0016200.mzXML:32843	2	77.44	1022.58	1022.58	-2.5695	1.8	-4.6	Yes
FL0016200.mzXML:18835	2	51.88	813.461	813.462	-1.4845	1.8	-4.6	Yes
FL0016200.mzXML:31552	2	75.22	1492.66	1492.66	-0.5196	1.8	-4.6	Yes
FL0016200.mzXML:29790	2	72.19	1521.74	1521.74	-0.0404	1.8	-4.6	Yes
FL0016200.mzXML:80159	4	143.72	2611.37	2611.37	-2.1408	1.8	-4.6	Yes

#ScanF

Number of peptide's MS2 fragmentation scan listed as: Filename.format:Scan#

#Charge

Charge state of precursor ion that gave rise to the peptide.

#Retention time

Peptide retention time (mins)

#Obs M+H

Inferred singly-charged ion mass, based on observed precursor ion's m/z ratio and charge

#Theo M+H

Computed peptide's singly-charged ion mass, based on its amino acid sequence and any additional modifications

#PPM

Parts-per-million mass deviation between observed and theoretical peptide masses

#PPM Upper

Upper bound of precursor mass tolerance based on all files submitted together in TagGraph instance

#PPM Lower

Upper bound of precursor mass tolerance based on all files submitted together in TagGraph instance

#PPM In Range

Specifies if calculated precursor error within specified tolerance [Yes/No]

FDR	EM Probability	1-Ig10 EM	Spectrum Score	Alignment Score	Composite Score	Unique Siblings	Context Mod Variants	Num Mod Occurrences
0	0.99999997	6.51735	9.351939	9	18.351939	350	1	6116
0	0.99999999	8.26582	8.196586	12	20.196586	62	1	6116
0	1	49.0427	8.500118	16	24.500118	66	9	6116
0	0.99999999	7.92573	7.884147	10	17.884147	350	2	6116
0	0.99999977	6.63157	8.833676	9	17.833676	855	1	6116
0	1	9.0336	9.152572	7	16.152572	855	4	6116
0	1	12.0068	8.270654	13	21.270654	125	5	6116
0	1	11.3179	9.382939	13	22.382939	120	2	6116
0	1	51.7194	6.919787	23	29.919787	855	7	6116

#FDR

PSM false discovery rate

#EM Probability

Expectation Maximization (EM)-estimated probability of indicated peptide identification being correct [0-1]

1 = high probability

0 = low probability

#1-Ig10 EM

Log-transformed, inverse probability of indicated peptide identification being correct, as computed by the expectation maximization-optimized Bayesian network; calculated as $\log_{10}(1-EM \text{ probability})$

#Spectrum Score

Score describing the extent to which a peptide agrees with the input spectrum

#Alignment Score

Heuristic score describing the extent to which the de novo sequence agrees with the database context sequence

#Composite Score

Composite score combining the spectrum score and the alignment score

#Unique Siblings

Number of other unique peptides found from the same protein that produced the current peptide-spectrum match

#Context Mod Variants

The number of unique peptides present in the dataset with the same base peptide sequence

#Num Mod Occurrences

The number of times the indicated modification(s) occur in the data set

Context	Mod Context	Mods	Mod Ambig Edges	Mod Ranges	Proteins	De Novo Peptide	De Novo Score	Matching Tag Length	Num Matches	N-terminus	Final Peptide	C-terminus	mod Pos0	mod Name0	mod AA0	mod Mass 0
APEIIFFAK	APEIIFFAK	[]	[]	[]	1:: [sp P02768	APEIIFFAK	99	9	1	Y	APEIIFFAK	R				
TFEIVYEEIII	TFEIVYEEIII	[]	[]	[]	1:: [sp Q14624	TFEIVYEEIII	99	12	1	K	TFEIVYEEIII	R				
ITVIHQDW	ITVIHQDW	[]	[]	[]	3:: [sp P01861	ITVIHQDW	99	16	3	R	ITVIHQDW	E				
IDDFAAFV	IDDFAAFV	[]	[]	[]	1:: [sp P02768	IDDFAAFV	99	10	1	V	IDDFAAFV	C				
IAHWSPAII	IAHWSPAII	[]	[]	[]	1:: [sp P04114	IAHWSPAII	99	9	1	E	IAHWSPAII	I				
TPAIHFK	TPAIHFK	[]	[]	[]	1:: [sp P04114	TPAIHFK	99	7	1	R	TPAIHFK	S				
SCGDICNFI	SCGDICNFI	[]	[]	[]	1:: [sp P04003	SCGDICNFI	99	13	1	R	2]GDIC[57	I	3	methyle	C	57
DEWSVNS	DEWSVNS	[]	[]	[]	1:: [sp P02787	DEWSVNS	99	13	1	R	02]DEWSV	I	2	methyle	C	57
IANIINSEEI	IANIINSEEI	[]	[]	[]	1:: [sp P04114	IANIINSEEI	99	23	1	K	IANIINSEEI	I				

#Context

Peptide sequence from input FASTA sequence database, with flanking amino acids (modifications not specified)

#Mod Context

Peptide sequence from input FASTA sequence database with modified residue designated with “-”.

#Mods

Modifications assigned to TagGraph-resolved peptide: Nested series are of the format:

“[(('Mod1 name from Unimod if exists', Mod1 delta mass from Unimod if it exists, Mod1 delta mass vs. Unimod if exists), (Mod1 target amino acid from Unimod if exists, Mod1 target amino acid location on peptide from Unimod if exists), indexed location of Mod1 on peptide sequence counting from zero), (('Mod2 name from Unimod if exists'..., indexed location of Mod2 on peptide sequence counting from zero), (...)]. “

Isobaric substitution” describes mass-neutral differences between the de novo-identified peptide and its database counterpart.

#Mod Ambig Edges

Indication of mass gaps TagGraph needed to reconcile between de novo and database sequences

#Mod Ranges

Indication of the possible modification positions TagGraph needed to consider when choosing the final modification location

#Proteins

List of proteins from FASTA sequence database containing indicated peptide

#De Novo Peptide

Peptide predicted by de novo sequencing algorithm (PEAKS) – derived from the de novo sequencing output .pepXML file(s)

#De Novo Score

Peptide de novo score (ALC) [0-100] by de novo sequencing algorithm (PEAKS). Higher score = higher confidence in de novo prediction

#Matching Tag Length

Length of the longest contiguous string in common between fasta and de novo sequence candidates

#Num Matches

Number of peptide candidates considered from fasta file considering spectrum and de novo peptide with indicated Matching Tag Length.

#N-terminus

Flanking residue on the peptide's N-terminus

#Final Peptide

Peptide assigned to spectrum by Taggraph, incorporating all amino acid substitution, and decimal representation of modification between brackets

#C-terminus

Flanking residue on the peptide's C-terminus

#modPosX

Modification position [X] on peptide

#modNameX

Name of modification at position [X] from Unimod if available

#modAAX

Modified amino acid at position [X]

#modMassX

Modification mass shift at position [X]

Taggraph currently outputs files and folders that summarize models used, modifications listed by rank of occurrences, and per scan scores as reported by TAGGRAPH before and after EM-ranking. These are for auditing purposes only and will be removed from the output in future versions. These are listed below:

Folder <SAMPLE> containing TG parsed files

<SAMPLE>_CHECK.txt.12

fileFractionMapping.pck

TagGraph.pep.xml

File containing user defined TG and EM parameters at input

TG_params_<SAMPLE>.txt

Files listing mods (single and combination, indicating delta mass, position, AA)

<SAMPLE>_addPlausibleMods_poss_combo_mods.tdv

<SAMPLE>_addPlausibleMods_poss_single_mods.tdv

EM results – Models End has a ranked list of mods

EM_Results_EMProbs_BEFORERERANK.tdv

EM_Results_EMProbs_END_TOPONLY.tdv

EM_Results_MODELS_BEFORERERANK.log

EM_Results_MODELS_END.log

EM_Results_RERANKSTATS_ONE.tdv

EM_Results_RERANKSTATS_TWO.tdv

VII. Known issues

Peptides that are found with long matching strings spanning hundreds or thousands of proteins (e.g., MHC genes) are not reported

```
Counts exceed max counts for peptide WANDIKK at scan number 6316 - counts: 1153 match length: 5
Counts exceed max counts for peptide ITSTQMSHR at scan number 5043 - counts: 1104 match length: 5
Counts exceed mod max counts for inexact matching peptide YVNVTKHAVVAVVNK at scan number 15095 - counts: 223 match length: 6
Counts exceed mod max counts for inexact matching peptide NCSAMIIIR at scan number 7432 - counts: 285 match length: 5
Counts exceed mod max counts for inexact matching peptide APKVYTIPGHK at scan number 12736 - counts: 252 match length: 5
Counts exceed max counts for peptide IIQITTR at scan number 11521 - counts: 1251 match length: 6
Counts exceed mod max counts for inexact matching peptide VMEEDETEPAK at scan number 3648 - counts: 211 match length: 5
```