

RAId: User Guide

Gelio Alves, Aleksey Ogurtsov, and Yi-Kuo Yu *

National Center for Biotechnology Information, National Library of Medicine,
National Institutes of Health, Bethesda, MD 20894.

(Dated: February 10, 2015)

Contents

I. RAId	2
A. Software Package	3
B. Syntax	4
C. Options	5
D. RAId Enhanced Organism Databases Status	10
Figure 1. - Information-preserved protein clustering example	11
Figure 2. - Structure of Enhanced Database.	11
Figure 3. - Illustration of Database Compression.	12
E. Database Formatting	13
F. User Enhanced Database Formatting	14
G. Post-Translation Modifications (PTMs) File	15
II. RAId Command Line Execution Examples	16
A. RAId_DbS Command Line Examples	16
B. RAId_aPS Command Line Examples	19
C. RAId Command Line Protein Identification Example	22
References	23

* to whom correspondence should be addressed: alves@ncbi.nlm.nih.gov, ogurtsov@ncbi.nlm.nih.gov, yyu@ncbi.nlm.nih.gov

I. RAId

RAId is a software designed to analyze MS/MS spectra. The software is written in C++ and implemented to execute in parallel in a single operating system taking advantage of multi-logical-cores when available. One of the features of RAId is that for each identified peptide it reports an *E-value*. RAId also has a unique database structure which incorporates scientific information from already observed post-translational modifications (PTMs), single amino acid polymorphisms (SAPs) and their associated diseases when available². Annotated databases from various species that can be used by RAId are available for download from ftp://ftp.ncbi.nlm.nih.gov/pub/qmbp/qmbp_ms/RAId/RAId_Databases/. RAId's unique database structure also allows users to construct specialized databases based on the user's own knowledge related to PTMs, SAPs and diseases.

RAId can assign statistical significance to identified peptides by three different methods. RAId_DbS¹ statistical significance is computed using a theoretically derived parametric distribution based on the central limit theorem or it can also compute statistical significance using extreme-value-distribution (EVD). While RAId_aPS³ assigns statistical significance using a probability distribution generated by scoring all possible peptides⁵. The scoring functions available with RAId_aPS are: RAId_DbS's scoring function (Rscore)¹, Hyperscore⁷, XCorr⁶ and Kscore⁸. When multiple scoring functions are selected RAId computes a combined database *P-value*⁴ which in principle can increase peptide identification confidence.

A. Software Package

Software Site:

<http://www.ncbi.nlm.nih.gov/CBBresearch/Yu/downloads/raid.html>

Installation:

To install unzip and untar the file RAID.tar.gz.

```
$ gunzip RAID.tar.gz
$ tar -xvf RAID.tar
```

Compiling:

```
$ make -f RAID.mak
```

Executable:

```
$RAID
```

Databases:

RAID can perform searches in specially formatted protein databases. Special annotated and formatted protein databases for different organisms which are searchable by RAID are available for download from the above ftp site. User can also generate their own database using the *-fp* option available or they can create their own enhanced databases using the perl UserDb.pl provided.

The FASTA file used by RAID has to have the following format. **Fasta file** example:

```
> |key|Id_Seq1(sequence identifier)| sequence description.
MLLATLLLLLLGGALAHDPRIIFPNHACEDPPAVLLEVQGTLQRPLVRDSRTSPANCTWLILGSKEQTVT
...
> |key|Id_Seq2(sequence identifier)| sequence description.
MTGSERLTLRSPLQPLISLCEAPPSPLQLPGGNVTITYSYAGARAPMGQGFLSYSQDWLMC
```

,where the allowed values for *key* are: gi, sp, tr, ref, pdb.

Example file:

The bash file raid_example.sh is an example of how to execute RAID. It contains someone of the syntax (parameters) that can be used to customize searches. One can also execute the bash file to verify that RAID is properly installed.

```
$/raid_example.sh
```

B. Syntax

[-cg],
 [-daa], [-db], [-dsv], [-dt],
 [-ect], [-ed], [-evc], [-ex], [-exf], [-ez],
 [-fl], [-fp], [-fps],
 [-ip],
 [-mc], [-mw],
 [-nc], [-nd], [-ng], [-nmcs],
 [-of], [-op],
 [-pie], [-pt], [-pfd], [-pfc],
 [-rap], [-ras], [-rnp],
 [-sm], [-ssr], [-ssk], [-ssh], [-ssx],
 [-v].

Executing mode option:

[-ex]

Enzyme option:

[-ect],[ez]

Cysteine modification option:

[-mc]

Molecular weight options:

[-cg], [-dt], [-mw], [-ng], [-pie], [-pt]

Amino acid residue (PTMs,SAPs) options:

[-daa], [-rap], [-ras], [-rnp]

Database options:

[-db], [-fp],

Scoring options:

[-dsv], [-evc], [-sm],

Scoring series for different scoring functions:

[-ssr], [-ssk], [-ssh], [-ssx],

Output and Input file options:

[-exf], [-ip], [-op]

Number of logical cores:

[-nc]

C. Options

–cg

Chemical group attached to peptide *C-terminal*.

Default value: `–cg 17.002739`

`–cg 17.002739` = Free Acid

`–cg 16.01872` = Amide

user can specify any molecular weight after the option `–cg`.

–daa

This option is used with RAId_aPS.

A list of the allowed residues to be used with RAId_aPS to generate the score histogram.

Default value:

`–daa [A00, G00, V00, L00, I00, P00, F00, Y00, W00, S00, T00, C00, M00, N00, Q00, D00, E00, K00, R00, H00]`.

Any of the amino acids and modifications presented in the file RAId_PTM_file are allowed choices .

The example below includes 2 PTMs G01 and G02

`–daa [A00, G00, G01, G02, V00, L00, I00, P00, F00, Y00, W00, S00, T00, C00, M00, N00, Q00, D00, E00, K00, R00, H00]`

–db

This option is used to specified the protein database to be searched.

`–db /path/database.name`

–dsv

- Scoring function to score peptides using RAId_DbS or RAId_aPS algorithm.

Any combination of the different scoring functions separated by comma are allowed parameters for RAId_aPS. While for RAId_DbS selecting of only one single scoring function is allowed.

Default value: `–dsv 1`.

Allowed options

`–dsv 0` = Rscore.

`–dsv 1` = Kscore.

`–dsv 2` = Hyperscore.

`–dsv 3` = XCorr.

–dt

Daughter fragment ions mass accuracy δm (Da.).

Default value: `–dt 0.2`.

–ect

Enzyme cleavage type. Peptide is fully-cleavaged or partially-cleavaged.

Default value: `–ect 0`

`–ect 0` = Fully-cleavaged

`–ect 1` = Partially-cleavaged

–ed

User can used the `–ed` to specify any experimental details to be included in the final output file.

Users must use quotation marks to specify the experimental details.

Example:

`–ed "Human liver cancer cell line study using ETD"`

–evc

Maximum allowed peptide *E-value*.

Default value: `–evc 10`.

–ex

RAId operation mode.

Default value: `–ex 1`

-ex 0 = RAId_aPS mode. Generates the total number of possible peptides for a given precursor ion.
-ex 1 = RAId_DbS database search mode. Score statistics using saddle point approximation.
-ex 2 = RAId_aPS database search mode. Score statistics using all possible peptides.
-ex 3 = RAId_aPS mode. Generates the score distribution by scoring all possible possible peptides for a given precursor ion.
-ex 4 = RAId_DbS database search mode. Score statistics using extreme-value-theory.
-ex 5 = RAId protein identification mode.

-exf

Extracting MS/MS spectrum file option.

The *-exf* option will generate single MS/MS spectrum from a file containing multiple MS/MS spectra.

-exf file_name

-ez

Enzyme option.

Default value: *-ez 1*

-ez 1 = Trypsin (K,R)
-ez 2 = Lys-C (K)
-ez 3 = Arg-C (R)
-ez 4 = GluC-Phosphate (E,D)
-ez 5 = GluC-Bicarbonate (E)
-ez 6 = PepsinA (L,F)
-ez 7 = Chymotrypsin (F,Y,W,L)
-ez 8 = Cyanogen bromide (M)
-ez 9 = Cyanogen bromide + Trypsin (M,K,R)
-ez 10 = Chymotrypsin + Trypsin (F,Y,W,L,K,R)
-ez 11 = V8-DE (N,D,E,Q)
-ez 12 = V8-E (E,Q)
-ez 13 = Trypsin + PepsinA (K,R,F,L)
-ez 14 = Lys-N (K)
-ez 15 = Asp-N (D,N)
-ez 16 = Asp-N_ambic (D,E)

-fl

Specifying a list of files (separated by comma) of peptide identification done by RAId. The file list is used for protein identification.

-fl file1,file2,...,filen

-fp

Formatting database option.

The option *-fp* will generate a database that can be used by RAId_DbS and RAId_aPS from a file of protein sequences in FASTA format.

-fp /path/input_database_name /path/output_database_name

-ip

Input MS/MS spectrum file name together with directory path.

-ip /path/msms_filename

-mc

Cysteine modification options.

Default value: *-mc C00* Unmodified Cysteine (103.009186 Da.).

Chemical group attached to the side chain of cysteine.

Other cysteine modifications can be found in the file RAId_PTM_file. If the user cysteine modification is not present in RAId_PTM_file the user can add to RAId_PTM_file the modified cysteine information value.

-mc C00 = Unmodified Cysteine (103.009186 Da.).

-*mc* C31 = Carboxymethylation (161.014649 Da.).
 -*mc* C32 = Carbamidomethylation (160.030646 Da.).
 -*mc* C33 = Pyridylethylation (208.066421 Da.).

-mw

This option is used with RAId_aPS.

Molecular weight (mw) used to compute the total number of possible peptides using RAId_aPS when a MS/MS file is not available.

-*mw* 2354.34 . Will compute the total number of peptides for the requested molecular weight.

The allowed molecular weight range for -*mw* is between [57,5000].

-nc

Number of logical cores to be used. For optimum performance this number should be equal to the number of logical core available in the operation system.

Default value: -*nc* 1

-nd

Number of random/decoy peptides. This used to estimate distribution parameters when employing extreme-value-statistics (EVD). Default value is set to score 100,000 random peptides per spectra to estimate EVD parameters.

-ng

Chemical group attached to peptide *N-terminal*.

Default value: -*ng* 1.007825

User can specify any molecular weight after -*ng*.

Example:

-*ng* 1.007825 = Hydrogen.

-*ng* 43.01838 = Acetyl.

-nmcs

Number of missed-cleaved sites allowed per peptide.

Default value: -*nmcs* 2

-op

Output search results path.

-*op* /path_name/

-of

Suffix used for output file name.

-*of* suffix_name

-pie

Parent monoisotopic mass isotope error.

This option correct for mass shift error when learning the parent ion molecular weight.

Default value: -*pie* 1. -*pie* 0 = will not correct for mass shift error

-*pie* 1 = will correct for mass shift error

-pt

Parent ion mass accuracy δm (Da.).

Default value: -*pt* 1.0.

RAId will look for all masses within $\pm 3 \times$ (Parent ion mass tolerance).

-rap

Users can specify database annotated Post-Translational Modified (PTMs) residues during database searches.

Default value: -*rap* NONE.

-rap [P01] = will allowed annotated PTM of proline P01.
-rap [P02, K03] = will allowed annotated PTMs of proline (P01) and lysine (K03).

-rnp

Users can specify novel Post-Translational Modified (PTMs) (not annotated in RAId's enhanced databases) residues during database searches.

Default value: *-rnp* NONE.

-rnp [P01] = will consider novel PTM of proline P01.

-rnp [P02, K03] = will consider novel PTMs of proline (P01) and lysine (K03).

-ras

Users can specify database annotated Single Amino Acid Polymorphisms (SAPs) residues during database searches.

Default value: *-ras* NONE.

Any of the 20 standard amino acids are allowed as parameter for the *-ras* field.

-ras [P] = will search only for annotated SAP of proline (P), i.e amino acids residues that are annotated in the database with proline as possible SAP.

-ras [P, K] = will search for annotated SAPs of proline and lysine.

-ssr

Fragmentation series used to by RAId scoring function.

Default value: Selected scoring function default files.

Any combination of the following are possible choices:

(b,b-H,b+H,a,a-H,a+H,c,b-NH3,b-NH3+H,b-H2O-H,y,y-H,y+H,x,z,y-NH3,y-H2O)

-sc b,y,a-H = Scoring peptides using the b,y and a-H series.

-sc a,c,y-H2O = Scoring peptides using the a,c and y-H2O series.

-ssh

Fragmentation series used to by RAId(Hyperscore) scoring function.

Default value: Selected scoring function default files.

Any combination of the following are possible choices:

(b,b-H,b+H,a,a-H,a+H,c,b-NH3,b-NH3+H,b-H2O-H,y,y-H,y+H,x,z,y-NH3,y-H2O)

-sc b,y,a-H = Scoring peptides using the b,y and a-H series.

-sc a,c,y-H2O = Scoring peptides using the a,c and y-H2O series.

-ssx

Fragmentation series used to by RAId(XCorr) scoring function.

Default value: Selected scoring function default files.

Any combination of the following are possible choices:

(b,b-H,b+H,a,a-H,a+H,c,b-NH3,b-NH3+H,b-H2O-H,y,y-H,y+H,x,z,y-NH3,y-H2O)

-sc b,y,a-H = Scoring peptides using the b,y and a-H series.

-sc a,c,y-H2O = Scoring peptides using the a,c and y-H2O series.

-ssk

Fragmentation series used to by RAId(Kscore) scoring function.

Default value: Selected scoring function default files.

Any combination of the following are possible choices:

(b,b-H,b+H,a,a-H,a+H,c,b-NH3,b-NH3+H,b-H2O-H,y,y-H,y+H,x,z,y-NH3,y-H2O)

-sc b,y,a-H = Scoring peptides using the b,y and a-H series.

-sc a,c,y-H2O = Scoring peptides using the a,c and y-H2O series.

-sm

MS/MS data collection mode.

Default value: *-sm* 1.

-sm 0 = Profile mode.

-sm 1 = Centroid mode.

–v
Output RAID code current version.

D. RAId Enhanced Organism Databases Status

In the table below are some examples of enhanced databases available for download to be used with RAId. The numbers in the table have not been updated they were taken from our 2008 publication² and does not reflect the information content of the most up to date enhanced databses. Databases available for download are constantly updated with newly documented SAPs, PTMs and their associated disease when available.

Organism	DB_name	Protein	NP	NM	SP	SAPs	PTMs	DB_size (byte)
<i>Homo sapiens</i>	hsa	29284	35059	35031	15030	116073	84406	16,265,018
<i>Anopheles gambiae</i>	angam	12388	12719	12706	112	350	50	6,042,277
<i>Arabidopsis thaliana</i>	artha	29651	31740	31711	5527	5207	11977	12,318,213
<i>Bos taurus</i>	botau	23796	26504	26491	3979	3295	15810	11,188,490
<i>Caenorhabditis elegans</i>	caele	22563	23097	23097	2890	1045	7756	10,050,609
<i>Canis familiaris</i>	cafam	31705	33834	33821	528	2766	4196	18,458,474
<i>Danio rerio</i>	darer	31192	36150	36137	1552	7358	3841	14,477,794
<i>Drosophila melanogaster</i>	drmel	17232	20207	20207	2568	5611	9290	9,796,785
<i>Equus caballus</i>	eqcab	17300	17637	17624	171	485	1045	9,404,150
<i>Gallus gallus</i>	gagal	18154	18724	18681	1455	1109	6522	8,728,501
<i>Macaca mulatta</i>	mamul	32547	38141	38128	207	1370	1262	14,498,187
<i>Mus musculus</i>	mumus	28506	35503	35451	12170	27614	61684	14,363,491
<i>Oryza sativa</i>	orsat	26636	26784	26777	1205	1291	2182	10,679,924
<i>Pan troglodytes</i>	patro	41464	52130	52117	482	3721	3734	20,217,986
<i>Plasmodium falciparum</i>	plfal	5240	5267	5267	88	56	184	3,995,386
<i>Rattus norvegicus</i>	ranor	28914	39425	39389	5569	9297	33240	15,879,569
<i>Saccharomyces cerevisiae</i>	sacer	5699	5880	0	5807	5507	13220	2,927,330

Table 1. The header abbreviations in this table are explained as follows. The second column, headed by DB_name, documents the abbreviated database name for searches using standalone version of RAId. The column headed by "Protein" indicates the final number of protein clusters in the processed organismal databases. The columns headed by NP, NM, and SP summarize the break down of the total number of accession numbers included respectively from protein products, transcript products, and SwissProt protein entries. The columns headed by SAPs and PTMs indicate respectively the total number of annotated SAPs and PTMs included. The last column shows the database size in bytes.

Figure 1. - Information-preserved protein clustering example

```

consensus seq.      ...DPR.....LQRLVADN<(N08)>GSE ...
member seq.        ...DPR<{W00}>...LKRLVVDN<(N11)>GSE ...
updated consensus seq.  ...DPR<{W00}>...LQ<{K00}>RLVA<{V00}>DN<(N08,N11)>GSE...

```

Figure 1. Information-preserved protein clustering example. Once a consensus sequence is selected, members of a cluster are merged into the consensus one-by-one. This figure illustrates how the information of a member sequence is merged into the consensus sequence. Amino acid followed by two zeros indicates an annotated SAP. Every annotated PTM has a two-digit positive integer that is used to distinguish different modifications. The difference in the primary sequences between a member and the consensus introduces *cluster-induced* SAPs. In this example, the residues Q and A (in red) in the consensus are different from the residues K and V (in blue) in the member sequence. As a consequence, K becomes a cluster-induced SAP associated with Q and V becomes a cluster-induced SAP associated with A. The annotated SAP, <{W00}>, associated with residue R in the member sequence is merged into the consensus sequence, see the updated consensus sequence in the figure. Note that the annotated PTM, <(N11)>, associated with N in the member sequence is merged with a different annotated PTM, <(N08)>, at the same site of the consensus sequence. Although, the SAPs, PTMs are merged, each annotation's origin and disease associations are kept in the processed definition file allowing for faithful information retrieval at the final reporting stage of the RAId's program.

Figure 2. - Structure of Enhanced Database.

```

...RTLVLGCKLG SAGGTD<{H00}>YGLR QFAEGSTEKL ..... [
...IEYISYFWVI GN<(N08,N09,N10,N11,N12)>QSSMWFAT SLSIFYFLKI ANFSNYIFLW LKSRTNMVLP
FMIVFLLISS LLNFAYIAKI LNDYKT<{M00}>KN<(N08,N09,N10,N11,N12)>DT VWDLNMYKSE ... [

```

Figure 2. Consensus protein sequences NP_775259 (first line, residues 480 – 510 shown) and NP_076410 (second and third lines, residues 81 – 170 shown) are used as examples to demonstrate the structure of our sequence file, part of the enhanced database. A “[” character is always inserted after the last amino acid of each protein to serve as a separator. Annotated SAPs and PTMs associated with an amino acid are included in a pair of angular brackets following that amino acid. SAPs are further enclosed by a pair of curly brackets while PTMs are further enclosed by a pair of round brackets. Amino acid followed by two zeros indicates an annotated SAP. Every annotated PTM has a two-digit positive integer that is used to distinguish different modifications.

Figure 3. - Illustration of Database Compression.

(A)
 ...LEVRQGTQLPLVR $\{\{W00\}\}$ DRSPM $\{\{V00\}\}$ (M01)CTWLILGSKEQTVTIR ...

(B)
 ...LEVRQGTQLPLVRDRSPMCTWLILGSKEQTVTIR
 JQGTQLPLVRSRSPVCTWLILGSKJQGTQLPLVRSRSP_mCTWLILGSKJQGTQLPLVWSRSPMCTWLILGSK
 JQGTQLPLVWSRSPVCTWLILGSKJQGTQLPLVWSRSP_mCTWLILGSK

Figure 3. In this example, the sequence has two nearby variable sites with residues R and M colored in red. Residue R may be replaced by a residue W due to a possible SAP; while residue M may be replaced by a residue V or an acetylated methionine (M01, in our notation) due to respectively a possible SAP or PTM. This information is encoded in our sequence file as shown in part (A). To encode the same information, method proposed in reference⁹ would have up to five additional highly similar peptides separated by a letter “J” appended to the end of the primary sequence, see part (B). Here a lower case m is used to denote the acetylated methionine. Another key difference in the two methods shown above is on the limit of allowed number of enzymatic miscleavages. In our method, there is no limit on the number of allowed miscleavages, while in other approaches, the number of miscleavages is usually set to below a certain threshold. As an example, in our method, the variant peptides SPVCTWLILGSKEQTVTIR and SP_mCTWLILGSKEQTVTIR are already included in (A). But in the approach of reference⁹, in order to allow consideration of this variant peptide, one either needs to additionally append this peptide or to have much longer flanking peptides than shown in (B).

E. Database Formatting

If users want to use a different database they can do so by first formatting the database. The database to be format has to be a file in **FASTA format** and the database can be easily format by using `-fp` option.

The FASTA file used has to have the following format.

Fasta file example:

> |*key*|Id.Seq1(sequence identifier)| sequence description.

MLLATLLLLLLGGALAHDPRIIFPNHACEDPPAVLLEVQGTLQRPLVRDSRTSPANCTWLILGSKEQTVT

, where the allowed values for *key* are: gi, sp, tr, ref, pdb.

Example:

```
./RAId -fp /path/input_database_filename /path/output_database_filename
```

Formatting the database will produce four files:

output_database_filename.def, **output_database_filename.inf**, **output_db_filename.prs**, **output_db_filename.seq**.

F. User Enhanced Database Formatting

RAId also permits users to create their own enhanced database. To generate a user enhanced database the user need to create two files: a FASTA file containing the protein sequences of interest and a second file containing the user expertise/knowledge of these proteins PTMs, SAPs and diseases.

The FASTA file used by RAId has to have the following format.

Fasta file example:

```
> |key|Id.Seq1(sequence identifier)| sequence description.
MLLATLLLLLLGGALAHDPRIIFPNHACEDPPAVLLEVQGTQRPLVRDSRTSPANCTWLILGSKEQTVT
IRFQKLHLACGSERLTLRSPLQPLISLCEAPPSPLQLPGGNVTITYSYAGARAPMGQGFLSYSQDWLMC
LQEEFQCLNHRCVSAVQRCDGVDACGDGSDDEAGCSSDPFPGLTPRPVPSLPCNVTLEDFYGVFSSPGYT
...
> |key|Id.Seq2(sequence identifier)| sequence description.
MTDFFFTHIIFPNHACEDPPAVLLEVQGTQRPLVRDSRTSPANCTWLILGSKEQTVT
GSERLTLRSPLQPLISLCEAPPSPLQLPGGNVTITYSYAGARAPMGQGFLSYSQDWLMC
AVQRCDGVDACGDGSDDEAGCSSDPFPGLTPRPVPSLPCNVTLEDFYGVFSSPGYT
...
```

, where the allowed values for *key* are: gi, sp, tr, ref, pdb.

Knowledge file example:

```
> |key|Id.Seq1
48  SAP  R   W      deadly cancer
56  PTM  N   N08,N09,N10,N11,N12
111 PTM  N   N08,N09,N10,N11,N12
139 SAP  M   V      diabetes
193 SAP  N   L,I,V
193 PTM  N   N08
299 PTM  N   N08,N09,N10,N11,N12
365 SAP  A   T      color blind
434 SAP  S   C,T,V,P insulin dependent diabetes
558 SAP  R   H,P,W
```

The **knowledge file** structure is explained below.

```
> |key|seq-identifier
```

First column field is the residue position.

Second column field signifies a SAP or PTM.

Third column field is the original residue present in the sequence.

Fourth column field is either a list of possible SAPs (L,I,V) or a list of possible PTMs (N08,N09,N10,N11,N12)

Fifth column field is the disease name if any at the given position.

Once the user has created the two files as described above the user can generate a knowledge database that RAId can process by executing the UserDb.pl script as shown below.

Example:

```
./UserDb.pl fasta_file_name knowledge_file_name output_format_database_name
```

The output_format_database_name is the database that can be processed by RAId.

G. Post-Translation Modifications (PTMs) File

RAId_PTM_file is the file that contains information related to amino acid residues and their corresponding post-translational modifications. The user can add any new post-translational modification to this file as long as one keeps with the same annotation structure shown below.

Line Code	Description
ID	Chemical Name of Amino Acid/PTM
AC	Residue Key
TG	Target Unmodified Amino Acid
RW	Unmodified Amino Acid Molecular Weight
MW	Modified Amino Acid Molecular Weight
PA	Location of the Modification in the Amino Acid Residue
PP	Position of the Amino Residue in the Peptide
CF	Chemical Modification to the Amino Acid Residue
MM	Monoisotopic Mass Difference $MM=MW-RW$
KY	Other Common Names Used to Identify the Same Molecule
LT	Other Terms Found In Literature not Necessary Correct Names

Some examples of the addition of new residues to the **RAId_PTM_file** file.

ID	Cholesterol glycine ester
AC	G01
TG	Glycine
RW	57.021465
MW	425.365766
PA	Amino acid backbone.
PP	C-terminal.
CF	C27 H44
MM	368.344301
KY	Lipoprotein.
LT	None

ID	N-palmitoyl cysteine
AC	C06
TG	Cysteine
RW	103.009186
MW	341.238852
PA	Amino acid backbone.
PP	N-terminal.
CF	C16 H30 O1
MM	238.229666
KY	Lipoprotein; Palmitate; Palmitoylation.
LT	Polmitoylation

II. RAID COMMAND LINE EXECUTION EXAMPLES

A. RAId_DbS Command Line Examples

Example 1: RAId in database search mode with some search options *-xxx*.

Command line:

```
>./RAId -ex 1 -ez 1 -nmcs 3 -nc 10 -dt 0.05 -pt 0.8 -ng 1.007825 -cg 17.002739 -evc
10 -mc C32 -ssr b,y,c,z -rap NONE -ras NONE -rnp S06,T10 -db /path/database.name -ip
/path/msms_filename -op /path/ -of output_file_name.
```

The example above would execute RAId in database search mode:

RAId_DbS (parametric distribution based on the central limit theorem) *-ex 1*.

Trypsin as the enzyme *-ez 1*.

Number of allowed missed cleavage sites 3 *-nmcs 3*.

Number of logical cores *-nc 10*.

Molecular error tolerance of daughter ion 0.05 Da. *-dt 0.05*.

Molecular error tolerance of parent ion 0.8 Da. *-pt 0.05*.

N-terminal group hydrogen *-ng 1.0078*.

C-terminal group free acid *-cg 17.0027*.

Maximum *E-value* allowed for reported peptide *-evc 10*.

Cysteine modification *-mc C32*.

Fragmented series used to score peptide *-ssr b, y, c, z*.

Information from annotated post-translation modifications off *-rap NONE*.

Information from annotated single amino acid polymorphisms off *-ras NONE*.

All amino acid residues of serine and tyrosine are considered as modified residues *-rnp S06,T10*.

Protein database path location *-db /path/database_name*.

Input MS/MS spectrum file path location *-ip /path/msms_filename*.

Search result output path location *-op /path/*.

Output file name *-of output_file_name*.

Example 2: RAId in database search mode with some search options *-xxx*.

Command line:

```
>./RAId -ex 1 -ez 1 -nc 4 -dt 0.01 -pt 0.01 -ng 1.007825 -cg 17.002739 -evc 1 -mc C32
-ssr b,y -rap NONE -ras S,T -rnp S06,T10 -db /path/database_name -ip /path/msms_filename -op
/path/ -of output_file_name.
```

The example above would execute RAId in database search mode:

RAId.DbS (parametric distribution based on the central limit theorem) *-ex 1*.

Trypsin as the enzyme *-ez 1*.

Number of logical cores *-nc 4*.

Molecular error tolerance of daughter ion 0.01 Da. *-dt 0.01*.

Molecular error tolerance of parent ion 0.01 Da. *-pt 0.01*.

N-terminal group hydrogen *-ng 1.0078*.

C-terminal group free acid *-cg 17.0027*.

Maximum *E-value* allowed for reported peptide *-evc 1*.

Cysteine modification *-mc C32*.

Fragmented series used to score peptide *-ssr b,y*.

Information from annotated post-translation modifications not used *-rap NONE*.

Annotated single amino acid polymorphisms that mutates into serine and tyrosine are used *-ras S,T*.

All amino acid residues of serine and tyrosine are considered as modified *-rnp S06,T10*.

Protein database path location *-db /path/database_name*.

Input MS/MS spectrum file path location *-ip /path/msms_filename*.

Search result output path location *-op /path/*.

Output file name *-of output_file_name*.

Example 3: RAId in database search mode with some search options *-xxx*.

Command line:

```
>./RAId -ex 4 -dsv 2 -ez 1 -nc 4 -dt 0.05 -pt 0.8 -ng 1.007825 -cg 17.002739 -evc
10 -mc C32 -ssr b,y,c,z -rap NONE -ras S,T -rnp S06,T10 -db /path/database_name -ip
/path/msms_filename -op /path/ -of output_file_name.
```

The example above would execute RAId in database search mode:

RAId_DbS(statistics computed by extreme-value-theory): *-ex 4*.

Scoring function Hyperscore *-dsv 2*.

Trypsin as the enzyme *-ez 1*.

Number of logical cores *-nc 4*.

Molecular error tolerance of daughter ion 0.05 Da. *-dt 0.05*.

Molecular error tolerance of parent ion 0.8 Da. *-pt 0.05*.

N-terminal group hydrogen *-ng 1.0078*.

C-terminal group free acid *-cg 17.0027*.

Maximum *E-value* allowed for reported peptide *-evc 10*.

Cysteine modification *-mc C32*.

Fragmented series used to score peptide *-ssr b,y,c,z*.

Information from annotated post-translation modifications off *-rap NONE*.

Annotated single amino acid polymorphisms that mutates into serine and tyrosine are used *-ras S,T*.

All amino acid residues of serine and tyrosine are considered as modified residues *-rnp S06,T10*.

Protein database path location *-db /path/database_name*.

Input MS/MS spectrum file path location *-ip /path/msms_filename*.

Search result output path location *-op /path/*.

Output file name *-of output_file_name*.

B. RAId_aPS Command Line Examples

Example 1: Computing the total number of possible peptides within a given molecular weight.

Command line:

```
>./RAId -ex 0 -ez 1 -dt 0.01 -pt 0.01 -ng 1.007825 -cg 17.002739 -mc C32 -ip  
/path/msms_filename -op /path/ -of output_file_name.
```

RAId_aPS executing mode *-ex 0*.

Trypsin as the enzyme *-ez 1*.

Molecular error tolerance of daughter ion 0.01 Da. *-dt 0.01*.

Molecular error tolerance of parent ion 0.01 Da. *-pt 0.01*.

N-terminal group hydrogen *-ng 1.0078*.

C-terminal group free acid *-cg 17.0027*.

Cysteine modification *-mc C32*.

Input MS/MS spectrum file path location *-ip /path/msms_filename*.

Search result output path location *-op /path/*.

Output file name *-of output_file_name*.

Example 2: Generating the score distribution for all possible peptides.

Command line:

```
>./RAId_DbS -ex 3 -ez 1 -daa [A00,G00,G02,G03,V00,L00,F00,Y00,W00,S00,T00,C00,N00,Q00,D00,E00,R00]  
-dt 0.05 -pt 0.8 -ng 1.007825 -cg 17.002739 -sc b,y -dsv 4 -ip /path/msms_filename -op /path/  
-of output_file_name.
```

DeNovo executing mode *-ex 3*.

Trypsin as the enzyme *-ez 1*.

Amino acids residues selected:

```
-daa [A00,G00,G02,G03,V00,L00,F00,Y00,W00,S00,T00,C00,N00,Q00,D00,E00,R00].
```

Molecular error tolerance of daughter ion 0.05 Da. *-dt 0.05*.

Molecular error tolerance of parent ion 0.8 Da. *-pt 0.05*.

N-terminal group hydrogen *-ng 1.0078*.

C-terminal group free acid *-cg 17.0027*.

Fragmented series used to score peptide *-sc b,y*.

Scoring function XCorr selected to compute score histogram *-dsv 4*.

Input MS/MS spectrum file path location *-ip /path/msms_filename*.

Search result output path location *-op /path/*.

Output file name *-of output_file_name*.

Example 3: Using RAId_aPS as a database search tool.

Command line:

```
>./RAId -ex 2 -dsv 1,2,3 -ez 1 -nc 10 -dt 0.05 -pt 0.8 -ng 1.007825 -cg 17.002739 -evc
10 -mc C32 -ssr b,y,c,z -rap NONE -ras NONE -rnp S06,T10 -db /path/database_name -ip
/path/msms_filename -op /path/ -of output_file_name.
```

The example above would execute RAId in database search mode:

Database search execution mode RAId_aPS *-ex 2*.

Scoring function used Rscore, Hyperscore and XCorr *-dsv 1,2,3*.

Trypsin as the enzyme *-ez 1*.

Number of logical cores *-nc 10*.

Molecular error tolerance of daughter ion 0.05 Da. *-dt 0.05*.

Molecular error tolerance of parent ion 0.8 Da. *-pt 0.05*.

N-terminal group hydrogen *-ng 1.0078*.

C-terminal group free acid *-cg 17.0027*.

Maximum *E-value* allowed for reported peptide *-evc 10*.

Cysteine modification *-mc C32*.

Fragmented series used to score peptide *-ssr b, y, c, z*.

Information from annotated post-translation modifications off *-rap NONE*.

Information from annotated single amino acid polymorphisms off *-ras NONE*.

All amino acid residues of serine and tyrosine are considered as modified residues *-rnp S06,T10*.

Protein database path location *-db /path/database_name*.

Input MS/MS spectrum file path location *-ip /path/msms_filename*.

Search result output path location *-op /path/*.

Output file name *-of output_file_name*.

C. RAId Command Line Protein Identification Example

Command line:

```
>./RAId -ex 5 -fl file1,file2,file3 -op /path/ -of output_file_name.
```

The example above would execute RAId in database search mode:

Performing protein identification using a list of files contained peptide identification performed by RAId mode *-ex 5*.

List of files separated by comma *-fl file1,file2,file3*.

Directory where files (file1,file2,file3) are located *-op /path/*.

Output file name *-of output_file_name*.

References

- Alves, G., A. Y. Ogurtsov, and Y. K. Yu, 2007, *Biol. Direct* **2**, 25.
- Alves, G., A. Y. Ogurtsov, and Y. K. Yu, 2008, *BMC Genomics* **9**, 505.
- Alves, G., A. Y. Ogurtsov, and Y. K. Yu, 2010, *PLoS ONE* **5**(11), e15438.
- Alves, G., W. W. Wu, G. Wang, R. F. Shen, and Y. K. Yu, 2008, *J. Proteome Res.* **7**, 3102.
- Alves, G., and Y. K. Yu, 2008, *Physica A* **387**, 6538.
- Eng, J. K., A. L. McCormack, and J. R. Yates III, 1994, *J. Amer. Soc. Mass Spectrom.* **5**, 976.
- Feny, D., and R. C. Beavis, 2003, *Anal. Chem.* **75**, 768.
- Keller, A., J. Eng, N. Zhang, X. J. Li, and R. Aebersold, 2005, *Mol. Syst. Biol.* **1**, 2005.0017.
- Schandorff, S., J. V. Olsen, J. Bunkenborg, B. Blagoev, Y. Zhang, J. S. Andersen, and M. Mann, 2007, *Nat. Methods* **4**, 465.