How to use $\mathcal{C}$-Ranker software package

$\mathcal{C}$-Ranker is a post-database searching software for peptide identification. The software package implements solvers for both C-Ranker model and cost-sensitive Ranker (CS-Ranker) model. The goal of the package is to identify correct PSMs output from the database searching SEQUEST. Users can find that it is easily to employ $\mathcal{C}$-Ranker to identify reliable PSMs from other database searching engines, such as MASCOT, and X!TANDEM. It is developed mainly in Matlab and naturally supports Matlab platform.

What's new

- version 4.2.0: 2018.2 Improve the efficiency of the online solver, OLCS-Ranker, especially on large-scale datsets. It keeps certain amount of PSMs from entering into the training process.
- version 4.0.0: 2017.8 Provide an online solver, OLCS-Ranker, that evidently accelerate the training process.
- version 2.1.0: 2014.11.21 Provided Matlab interfaces.
- version 2.0.1: 2014.11.19 Added '-e' and '-n' parameter for 'cranker_read' command to allow users to choose whether to employ the feature enzN, enzC and numProt;
- version 2.0.0: 2014.8.

The following folders and files contained in the distributed package for Matlab platform:

1. .\ demo a folder consisting of a demo m-file to use $\mathcal{C}$-Ranker and a demo data file;
2. .\ interface a folder containing the interfaces of $\mathcal{C}$-Ranker (m files);
3. .\ lib_m a folder containing the Matlab codes of $\mathcal{C}$-Ranker (m files);
4. .\ cranker_setup.m an m file to set up $\mathcal{C}$-Ranker;

1 Platforms and paths

- **On Matlab platform**

  For operating systems that Matlab has installed, users can run $\mathcal{C}$-Ranker by calling the Matlab interfaces directly. Turn the Matlab directory of the $\mathcal{C}$-Ranker path which contains the $\mathcal{C}$-Ranker Matlab codes and folders and call

  ```
  cranker_setup
  ```

  in Matlab command window to setup $\mathcal{C}$-Ranker paths and compile C files.

2 Prepare data files

$\mathcal{C}$-Ranker supports text files and Excel files. The input format of the data file looks like as follows,

spectrum	peptide	protein	ions	xcorr	deltacn	sprank	hit_mass
B_GCN5_jun01.0901.1	F.AGVGA.M	YAL009W	4/8	1.108	1	1	373.4099
B_GCN5_jun01.0904.1	F.IAGM.S	Reverse_Q0045	3/6	0.605	1	13	390.4986
.....							

- The first row in the data file ("spectrum, peptide, ...") indicates the attribute names of the PSM data. The order of the attributes in data representation doesn't matter, but the names of the attributes must be correct.
- In text data file, the values are separated by tab ('\t') space by default.

3 Run C-ranker

3.1 On Matlab platform

Open Matlab platform and type the following commands in Matlab command window.

(1) Read data. $\mathcal{C}$-Ranker loads the data in Excel file 'testData.xls' to a new file 'testData.mat' by typing

```
cranker_read('testData.xls','testData.mat');
```

(2) Calculate scores of PSMs. $\mathcal{C}$-Ranker trains a classification model and calculates the score for each PSM record by typing

```
cranker_solve('testData.mat','testData_score.mat');
```

(3) Output identified PSMs. $\mathcal{C}$-Ranker output identified reliable PSMs to a text file by typing

```
cranker_write('testData.mat','testData_score.mat');
```

Refer Section **??** for the descriptions of these input parameters.

4 Change data representation

4.1 Default data representation

By default $\mathcal{C}$-Ranker uses 9 attributes for representing a PSM data point, of which 5 comes from original sequest output file:

xcorr, deltacn, sprank, ions, hit_mass;

the other 4 are calculated by $\mathcal{C}$-Ranker:

enzN, enzC, numProt, xcorrR

with the meanings:

- enzN: 1 or 0, whether the peptide preceded by a tryptic site;
- enzC: 1 or 0, whether the peptide has a tryptic C-terminus;
- numProt: number of times the matched protein matches other PSMs;
- xcorrR: deltacn/xcorr;

$\mathcal{C}$-Ranker deals with the features xcorr, deltacn and ions in special:

- xcorr, deltacn: As these two features play more important roles in identification, $\mathcal{C}$-Ranker assigns larger weights to them in the default setting,
- ions: $\mathcal{C}$-Ranker supports the ratio format for representing ions, e.g., "4/8".

In addition to the 9 attributes, 3 other attributes, i.e., spectrum, protein, and peptide, are employed by $\mathcal{C}$-Ranker to distinct PSM data and calculate the appended features.

The value types of the three attributes are all strings. Particularly,

- $\mathcal{C}$-Ranker identifies the label of a PSM record by the following 2 alternative methods:

  Method 1. $\mathcal{C}$-Ranker calculates the label by the value of attribute "protein" (label =1 if "protein"= "a protein name", -1 if "protein"="Reverse_protein name"). For instance, if "protein"="Reverse_YAL009W", then the PSM is labeled -1 indicating a decoy PSM. If "protein"="YAL009W", it is labeled as 1 indicating a target PSM.

  Method 2. Add an attribute explicitly with name "label" in the data file. The label values consist of 1 and -1.

- $\mathcal{C}$-Ranker uses protein attribute (if any) to calculate the values of numProt, employs peptide attribute (if any) to calculate the values of enzN and enzC.

Users may only provide a part of the attributes listed above. But at least 1 numeric attribute and the labels should be ensured.

4.2 Add new features

$\mathcal{C}$-Ranker allows a user to add new attributes into the PSM data representation. For instance, if a user needs to add two other features named "deltaM" and "deltaN", just to add two columns "deltaM" and "deltaN" in the data file. Then the data file may look like

spectrum	peptide	protein	xcorr	.....	deltaM	deltaN
B_GCN5_jun01.0901.1	F.AGVGA.M	YAL009W	1.108	.....	1.20	0.919
B_GCN5_jun01.0904.1	F.IAGM.S	Reverse_Q0045	0.605	.....	0.53	0.498
.....	.....					

4.3 Remove attributes from $\mathcal{C}$-Ranker

$\mathcal{C}$-Ranker reads all the numeric attributes consists in the data file. If an attribute in the data file is not want be employed by $\mathcal{C}$-Ranker, insert a minus '-' in the beginning of the attribute name. For instance, '-deltaM' indicates that 'deltaM' attribute would be neglected by $\mathcal{C}$-Ranker. And the data file may look like

spectrum	peptide	protein	xcorr	.....	-deltaM	deltaN
B_GCN5_jun01.0901.1	F.AGVGA.M	YAL009W	1.108	.....	1.20	0.919
B_GCN5_jun01.0904.1	F.IAGM.S	Reverse_Q0045	0.605	.....	0.53	0.498
.....	.....					

$\mathcal{C}$-Ranker generated 4 features in default: enzN, enzC, numProt, xcorrR. Set '-e' parameter 0 for 'canker_read' command to cancel the enzN and enzC feature; Set '-n' parmaeter 0 for 'canker_read' command to cancel the numProt feature.

E.g., the following command read data from testData.xls to testData.mat and do not generate the values of the feature enzN, enzC and numProt.

```
cranker_read('-e','0','-n','0','testData.xls','testData.mat');
```

5 Parameters of the $\mathcal{C}$-Ranker command

$\mathcal{C}$-Ranker provides the following commands.

- cranker_read: read data from a text or Excel file;
- cranker_solve: identify correct target PSMs;
- cranker_write: output the results of $\mathcal{C}$-Ranker to file;
- cranker_version: get the version of $\mathcal{C}$-Ranker.

Their parameters are illustrated as follows.

5.1 cranker_read

parameter	value	description	default
-l		the delimiter of the values of different fields of text data file, effective if the file is text type;	'\b\t'
-p		a string indicating the prefix of a decoy protein. If a PSM has the protein field beginning with the given prefix, then it is labeled as -1 (false PSM), otherwise it is labeled as 1 (target PSM);	'Reverse'
-w	1, 2, ...	a positive integer, indexing the title row, e.g. titleRow = 3: then the 3rd row is title row, and the first two rows will be ignored, effective if the file is text type;	1
-e	0 or 1	0: do not employ the feature enzN and enzC; 1: employ these two features;	1
-n	0 or 1	0: do not employ the feature numProt; 1: employ this feature;	1
-v	0 or 2	0: do not print any information to command window; 2: put out progress information briefly to command window;	2

E.g. 1 Set the title row as 2nd row, and set verbose 0 not to print any information to command window.

```
cranker_read('-w',2,'-v',0,'testData.xls','testData.mat');
```

on Matlab platform.

5.2 cranker_solve

parameter	value	description	default
-g	1 or 0	whether to standardize (make each attribute zero-mean and unit-variance);	1
-f	0, 1 or 2	whether split the samples into training set and test set; 0: do not split the samples into train set and test set; 1: split the samples into train set and test set; 2: employ the user-supplied training set and test set;	1
-t		a positive scaler indicating the rate of cardinality of training set to the cardinality of test set; effective only if -f is set 1;	1
-x		a positive scalar indicating the relative feature weight of xcorr and deltacn;	2.0
-c1		the weight of training error of both decoys and targets for C-Ranker model and the weight of decoys for CS-Ranker model;	4.8
-c3		the weight of training error of the targets for CS-Ranker model	2.4
-c2		the weight for encouraging more target PSMs for CRanker model	4.8
-lambda		the weight for encouraging more target PSMs for CS-Ranker model	2.4
-r		a positive scalar, the kernel parameter;	1.0
-s		solver and corresponding classification model • 'CCCP_online': employ online algorithm to solve the CS-Ranker model • 'CCCP_batch': employ the batch-CS-Ranker to solve the CS-Ranker model • 'primal_SVM: employ the built-in solver to solve the C-Ranker model. Note that `primal_SVM` solves a different model.	`CCCP_online`
-v	0 or 2	0: do not print any information to command window; 2: put out progress information briefly to command window;	2

The `CCCP_online` solver has the following parameters

parameter	value	description	default
-act		initial minimum number of the elements of the working set for starting nonconvex CCCP procedure	1000
-muSafe		threshold of the predicted function value as a safely correct prediction	0.3
-muSafeTarget		threshold of the predicted function value as a safely correct target	Inf
-tauInitial		initial value of the tolerance of violating pair	0.05
-tauMin		minimum value of the tolerance of violating pair	0.05
-fixTrainTestSet	1 or 0	1: fix the train set and test set when splitting the whole PSMs at different calls; 0: make the splitted train set and test set different at different calls;	1
-fixTrainOrder	1 or 0	1: fix the order of the training PSMs to enter into the online learning iteration at different calls; 0: make the order of the training PSMs to enter into the online learning iteration different at different calls;	1

The `CCCP_batch` solver has the following parameters

parameter	value	description	default
-z		maximum number of samples of the training set;	20000
-m		number of submodels;	5

The `primal_SVM` solver has the following parameters

parameter	value	description	default
-z		maximum number of samples of the training set;	20000
-m		number of submodels;	5
-tolFun		tolerence of built-in solver for iterated function values.	0.1

E.g. 1 A user can choose not to split the data file into training and testing files by setting

```
cranker_solve('-f',0,'testData.mat','testData_score.mat');
```

on Matlab platform.

E.g. 2 A user can choose to employ the specified training file and test file by setting

```
[trainFile,testFile] = cranker_split(dataFile);

matScoreFile = 'testData_score.mat';

cranker_solve('-f','2',trainFile,testFile,matScoreFile);
```

on Matlab platform.

5.3 cranker_write

parameter	value	description	default
-fdr		a positive scalar indicating the FDR level;	0.05
-v	0 or 2	0: do not print any information to command window; 2: put out progress information briefly to command window;	2

E.g. 1 If a user needs the TP, FP and other accuracies under FDR level 0.08, then the user may call

```
cranker_write('-fdr',0.08,'testData.mat','testData_score.mat');
```

on Matlab platform.

5.4 cranker_version

Usage

```
cranker_version
cranker_version('-date')
```

Description

`cranker_version`: return the $\mathcal{C}$-Ranker version;

`cranker_version('-date')` (Matlab): return the release date of the $\mathcal{C}$-Ranker;