# How to use C-Ranker

**𝒞-Ranker is a post-database searching software for peptide identification. The goal of C-Ranker is to identify correct PSMs output from the database searching tool SEQUEST. C-Ranker is developed in Matlab and C. This version, 2.0.1, designed for Windows users without installation of Matlab.**

### What's new

- version 2.0.1: 2014.11. Add '-e' and '-n' parameter for 'cranker_read' command to allow users determine whether to employ the feature enzN, enzC and numProt;
- version 2.0.0: 2014.8. Designed for Windows users without installation of Matlab.

The following files contained in the distributed package:

1. MCRInstaller.exe The MATLAB Compiler Runtime (MCR) library with version 7.14 (R2012a).
2. cranker_read.exe Read data of PSM records.
3. cranker_solve.exe Calculate scores for each PSM.
4. cranker_write.exe Put out the results.
5. cranker_version.exe Get C-Ranker version.
6. testData.xls A demo Excel data file.
7. how to use cranker2_0_0.pdf This file.

# 1 Install MCR

MCR is required to setup to enable C-Ranker execution without an installed version of Matlab. Double click the file MCRInstaller.exe to install the MCR.

# 2 Prepare data files

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

Open the MS-DOS command window. Change the directory to the path where C-Ranker executable files located. Type the following script in the MS-DOS command window.

(1) Read data. C-Ranker loads the data in Excel file ‘testData.xls’ to a new file ‘testData.mat’ by typing

```
cranker_read  testData.xls  testData.mat
```

The Excel file testData.xls consists of the raw PSM records. The file ‘testData.mat’ will be created to store the PSM records in Matlab MAT file format. By default, C-Ranker find or create the files in current directory. If the files locate in other directory, the directory should be included in the file name. For instance, `D:\data\cranker_read.txt`, `D:\data\cranker_read.mat`.

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

```
cranker_solve testData.mat testData_score.mat
```

The trained model and calculated scores are stored in a file `testData_score.mat` (users may set other names with extension ‘.mat’). The values of scores follow in the interval $[-1, 1]$. A PSM with higher score indicates that it is more likely to be correct.

(3) Output identified PSMs. C-Ranker output identified reliable PSMs to a text file by typing

```
    cranker_write testData.mat testData_score.mat
```

If the command is successfully executed, a file named "****_result_dd-mm-yyyy.txt" will be generated, where dd-mm-yyyy indicates the current date, such as "22-Aug-2014".

# 4 Change data representation

## 4.1 Default data representation

By default 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 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;

C-Ranker deals with the features xcorr, deltacn and ions in special:

- xcorr, deltacn: As these two features play more important roles in identification, C-Ranker assigns larger weights to them in the default setting,
- ions: 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 C-Ranker to distinct PSM data and calculate the appended features.

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

- C-Ranker identifies the label of a PSM record by the following 2 alternative methods:

  Method 1. 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.

- 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

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 C-Ranker

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 C-Ranker, insert a minus ’-’ in the beginning of the attribute name. For instance, ’-deltaM’ indicates that ’deltaM’ attribute would be neglected by 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 |
| …… | …… | | | | | |

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 C-Ranker command

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 C-Ranker to file;
- cranker_version: get the version of 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
```

## 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 | 1 or 0 | whether split the samples into 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; | 1.0 |
| -c2 | | the weight for encouraging more target PSMs; | 1.0 |
| -r | | a positive scalar, the kernel parameter; | 1.0 |
| -z | | maximum number of samples of the training set; | 20000 |
| -m | | number of submodels; | 5 |
| -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** 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
```

## 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
```

### 5.4 cranker_version

**Usage**

```
cranker_version
cranker_version -date
```

**Description**

`cranker_version`: return the C-Ranker version;

`cranker_version  -date`: return the release date of the C-Ranker;

## 6 Tips of practical use

- It is recommended that users run C-Ranker on a computer with high computation capability if possible. For a dataset having about 400,000 PSM records, it may cost about 5 hours on a PC with CPU Intel Core i5-2400 3.10GHz × 4 and Memory 8GB.