How to use OLCS-Ranker software package

OLCS-Ranker is a post-database searching software for peptide identification. The software package implements solvers for cost-sensitive Ranker (CS-Ranker) model. The goal of the package is to identify correct PSMs output from the database searching engines based on target-decoy strategy. OLCS-Ranker do not depend on any particular searching engine. It could be applied to deal with the PSM records output by the search engine of Sequest, X!Tandem, Mascot, Tide, Comet, etc., once the sample data are provided in the required format. OLCS-Ranker is developed mainly in Matlab and naturally supports Matlab platform. Moreover, the complied Exe files are also provided, thus OLCS-Ranker commands could also run in Windows OS after installing Matlab Runtime Complier (MRC, a free software). Alternatively, we have also provided a web-based GUI for users of OLCS-Ranker. A user can visit http://161.6.5.181:8000/olcs-ranker/. In case that the server is under maintenance, please contact zhonghang.xia@wku.edu.

What's new

- version 4.4.5: 2020.2.14
  fix bugs of reading raw data
  - compatible with raw data with empty values in string features;
  - compatible with raw data when there is no digest information;
- version 4.4.2: 2019.12.11
  - add cross-validation to determine the parameter values in `olcs-solve()`
  - provide a set of python interfaces based on matlab engine;
  - adjust the definiation of FDR `COEF_FDR = 2.0;  % FDR = COEF_FDR * FP / (TP + FP)`
  - fix the bug of `crossValidate()`: `Line 381. Share variables between nested functions`.
  - `readData()` ** be compatible with illegal specified feature names
    ** remove ('MultipleDelimsAsOne',1) setting for `textscan()`
  - `config()`: reset the default value `save_result_text = 1;`
- version 4.4.1: 2018.11.13 test on Alibaba Cloud
- version 4.3: 2018.9.5
  - add cross-validation to select values of cdecoy, ctarget and kernel parameter
  - revise identifyPar() to support non-string input parameters
  - revise `kernelMatrix_OnceaLine()` to support linear kernel

  - fix a debug of `split_train_test.m` when save `X_feature`;
  - `completeArg()` add a mode of `mode_compulsory`
  - `olcs_solve()`: fix a bug of `arg.flag_split_train_test`
  - adjust the usage of the parameter of `verbose`

- 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 3.0.0: 2015.2.2
  - Provide Interfaces for multiple PCs/workers.

    Users can split the PSM records to multiple data files, and train sub-models independently on different PCs/workers. It reduces the training time and is applicable for large datasets.
  - Provides Matlab platform users to define the combination method of submodel scores.
- 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. Designed for Windows users without installation of Matlab.
- version 1.0: 2012.12 The first version of CRanker.

The following files contained in the distributed package for Windows OS:

1. olcs_read.exe Read data of PSM records.
2. olcs_solve.exe Calculate scores for each PSM.
3. olcs_write.exe Put out the results.
4. olcs_version.exe Get OLCS-Ranker version.
5. testData.txt A demo text data file.
6. Readme.txt a file describing the details of how to installing Matlab Complier Runtime (MCR).

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 OLCS-Ranker and a demo data file;
2. .\ interface a folder containing the interfaces of OLCS-Ranker (m files);

3. .\ lib_m a folder containing the Matlab codes of OLCS-Ranker (m files);

4. .\ olcs_setup.m an m file to set up OLCS-Ranker paths;

5. .\ config.m an m file for setting the parameter values of OLCS-Ranker.

1 Platforms and paths

- **On Matlab platform**

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

  ```
  olcs_setup
  ```

  in Matlab command window to setup OLCS-Ranker paths and compile C files.

- **On Windows OS without Matlab platform**

  For Windows users, it need to install specified version of MATLAB Compiler Runtime (MCR, a free software) first. MCR is required to setup to enable OLCS-Ranker execution without an installed version of Matlab. Double click the file MCRInstaller.exe to install the MCR.

  For detail of downloading, installing MCR, users may refer to the Readme file.

2 Prepare data files

OLCS-Ranker supports text files and Excel files.

The text data input file is a simple Tab-delimited format where each row contains a per PSM:

$$\text{attribute}\,1\ \langle\text{tab}\rangle\ \text{attribute}\,2\ \langle\text{tab}\rangle\ \cdots\ \langle\text{tab}\rangle\ \text{attribute}\,N$$

An input format of the text data file looks like as follows,

spectrum	⟨tab⟩	peptide	⟨tab⟩	protein	⟨tab⟩	ions	⟨tab⟩	xcorr	⟨tab⟩	deltacn
B_GCN5_1	⟨tab⟩	F.AGVGA.M	⟨tab⟩	YAL09W	⟨tab⟩	0.5	⟨tab⟩	1.108	⟨tab⟩	1
B_GCN5_2	⟨tab⟩	F.IAGM.S	⟨tab⟩	Reverse_Q045	⟨tab⟩	0.8	⟨tab⟩	0.605	⟨tab⟩	1
⋯	⟨tab⟩	⋯	⟨tab⟩	⋯	⟨tab⟩	⋯	⟨tab⟩	⋯	⟨tab⟩	⋯

Requirement of the PSM data file format

1. The provided PSM data file should be tab-delimited text file or an Excel file.
2. The first row in the data file ("spectrum, peptide, protein ...") indicates the attribute names of the PSM data.
3. The order of the attributes in data representation doesn't matter, but the names of some special attributes should be correct.

Table 1: Characteristics of some special attributes of a PSM record

attribute name	characteristic	spell of attribute name
protein	required if there's no attribute `label`	`protein`
peptide	useful, but not required	`peptide`
xcorr	useful, but not required	`xcorr`
deltacn	useful, but not required	`deltacn`
label	1 or -1, indicating matched to target or decoy protein, required if there's no attribute `protein`	`label`

OLCS-Ranker identifies the attributes by the attribute names. Either `protein` or `label` attribute should be provided

- The `protein` attribute should be named "protein".
- The `label` attribute should be named "label";
- The `peptide`, `xcorr`, `deltacn` attributes are not required, but we suggest users name them correctly (if any), since OLCS-Ranker deal with these attributes specially.

4. OLCS-Ranker identifies the label of a PSM record by the following two alternative methods:

   Method 1 OLCS-Ranker identifies the label of a PSM by the value of attribute "protein" (label =1 if "protein"= "a protein name", -1 if "protein"="Reverse_protein name"). A decoy protein has protein attribute values leading by "Reverse" in default.

   For instance,

`protein`	`label`	description
Reverse_YAL09W	-1	a decoy PSM
YAL009W	1	a target PSM

Method 2 Add an attribute explicitly with name `label` in the data file. The label values consist of 1 and -1, with 1 indicating a PSM matched to a target protein, and -1 a PSM matched to a decoy protein.

5. Each attribute name can contain only letters, numbers, or the underscore character (_).

6. If the raw PSM data file is Excel files with extension `.xls` or `.xlsx`, the records should be on the default sheet named "`sheet1`".

3 Run C-ranker

3.1 On Matlab platform

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

(1) Read data. OLCS-Ranker loads the data in text file 'testData.txt' to a new file 'testData.mat' by typing

```
olcs_read('testData.txt','testData.mat');
```

The file `testData.txt` 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, OLCS-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\olcs_read.txt`, `D:\data\olcs_read.mat`.

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

```
olcs_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. OLCS-Ranker output identified reliable PSMs to a text file by typing

```
olcs_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.

3.2 On Windows

Open the MS-DOS command window. Change the directory to the path where OLCS-Ranker executable files located. Type the following script in the MS-DOS command window. After installing Matlab Complier Runtime (MCR), OLCS-Ranker could run with the command `olcs_read`, `olcs_solve` and `olcs_write`.

(1) Read data. OLCS-Ranker loads the data in text file 'testData.txt' to a new file 'testData.mat' by typing

```
olcs_read  testData.txt  testData.mat
```

The text file testData.txt 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, OLCS-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\olcs_read.mat`.

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

```
olcs_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. OLCS-Ranker output identified reliable PSMs to a text file by typing

```
olcs_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.

4 Change data representation

4.1 Default data representation

OLCS-Ranker calculated the following 3 attributes

enzN, enzC, numProt

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;

if the attributes `peptide`, `protein` are provided.

Particularly,

- OLCS-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.

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

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

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 attributes

OLCS-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 "attributeM" and "attributeN", just to add two columns "attributeM" and "attributeN" in the data file. Then the data file may look like

spectrum	peptide	protein	xcorr	.....	attributeM	attributeN
B_GCN5_jun01.0901.1	F.AGVGA.M	YAL09W	1.108	.....	1.20	0.919
B_GCN5_jun01.0904.1	F.IAGM.S	Reverse_Q045	0.605	.....	0.53	0.498
.....	.....					

4.3 Remove attributes from OLCS-Ranker

OLCS-Ranker reads all the numeric attributes consists in the data file. If an attribute in the data file does not want to be employed by OLCS-Ranker, insert a minus '-' character in the beginning of the attribute name. For instance, '-attributesM' indicates that 'attributesM' attribute would be neglected by OLCS-Ranker. And the data file may look like as follows.

spectrum	peptide	protein	xcorr	.....	-attributesM	attributesN
B_GCN5_jun01.0901.1	F.AGVGA.M	YAL09W	1.108	.....	1.20	0.919
B_GCN5_jun01.0904.1	F.IAGM.S	Reverse_Q045	0.605	.....	0.53	0.498
.....	.....					

OLCS-Ranker generated 3 features in default: enzN, enzC, numProt . 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.txt to testData.mat and do not generate the values of the feature enzN, enzC and numProt.

```
olcs_read('-e','0','-n','0','testData.txt','testData.mat');
```

5 Parameters of the OLCS-Ranker command

OLCS-Ranker provides the following commands.

- olcs_read: read data from a text or Excel file;
- olcs_solve: identify correct target PSMs;
- olcs_write: output the results of OLCS-Ranker to file;
- olcs_version: get the version of OLCS-Ranker.

Their parameters are illustrated as follows.

5.1 olcs_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.

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

on Matlab platform and

```
olcs_read -w 2 -v 0  testData.xls testData.mat
```

on Windows MS/DOS command window.

5.2 olcs_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
-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

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

on Matlab platform and

```
olcs_solve -f 0  testData.mat testData_score.mat
```

on Windows MS/DOS command window.

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

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

matScoreFile = 'testData_score.mat';

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

on Matlab platform and

```
olcs_solve -f 2 myTrainFile  myTestFile.mat testData_score.mat
```

on Windows MS/DOS command window.

5.3 olcs_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

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

on Matlab platform and

```
olcs_write -fdr 0.08  testData.mat testData_score.mat
```

on Windows MS/DOS command window.

5.4 olcs_version

Usage

```
olcs_version
olcs_version('-date')
```

Description

`olcs_version`: return the OLCS-Ranker version;

`olcs_version('-date')` (Matlab) or `olcs_version -date` (Windows MS/DOS): return the release date of the OLCS-Ranker.