

Introduction to BioChemical Library (BCL) command syntax

BCL-Rosetta Drug Design Workshop

January 2022

BCL application group options are available by running the executable with no additional options

```
tungsten: /home/brownbp1/BCL_Workshop_2022% bcl.exe
```

```
BCL Help
```

```
BCL v4.2.0, r0, compiled on Sun Jan  9 03:10:46 2022
```

```
Usage: bcl.exe <application_group>:<application> [Other parameters] [Flags]  
[@filenames]
```

```
<application_group>:<application> name of the bcl application to be executed, Choose  
from the following:
```

```
<group>:          Description
```

```
<group>:Help      prints all <application>'s and descriptions for the group
```

```
...
```

```
...
```

The applications in each application group can be further explored by requesting help for an individual application group

```
tungsten: /home/brownp1/BCL_Workshop_2022% bcl.exe model:Help
```

Members of Group model:

model:	Group of applications that primarily use or create machine learning models. Member applications:
ComputeStatistics	Evaluate quality measures of qsar model predictions or just experimental/predicted values and present results in table and as gnuplot graphics.
PredictionMerge	Merge cross-validation/prediction matrices horizontally or simply append.
Test	Test any machine learning model, including ANNs, SVMs, and many more.
TestANNWithDropoutResampling	Test an ANN using dropout at test-time to compute the distribution of values for each output.
Train	Train any machine learning model, including ANNs, SVMs, and many more.

The applications can be further explored by passing help options to the application

```
tungsten: /home/brownbp1/BCL_Workshop_2022% bcl.exe model:Train --help
```

```
model:Train Help
```

```
BCL v4.2.0, r0, compiled on Sun Jan 9 03:10:46 2022
```

```
Usage: bcl.exe model:Train [OPTIONAL PARAMETERS] -training <> -monitoring <> -  
independent <> [OPTIONAL FLAGS] [@FILENAMES]
```

The applications can be further explored by passing help options to the application

```
tungsten: /home/brownbp1/BCL_Workshop_2022% bcl.exe model:Train --help
```

PARAMETERS: arguments that immediately follow the application name

<iterator> data label of a specific iterate used as training algorithm in `opti::Approximator`, default: "LinearRegression", choose any implementation of `bcl::model::ApproximatorBase`:

Customizable Implementations

- * `ApplicabilityDomainKohonen` : A kohonen-map based implementation to detect whether a point is within the applicability domain of a model. Inspired by Assessment of applicability domain for multivariate counter-propagation artificial neural network predictive models by minimum Euclidean distance space analysis: A case study; "`ApplicabilityDomainKohonen(help)`" shows internal options
- ...
- * `MultiOutputSVM` : trains a support vector machine using sequential-minimal-optimization (see http://en.wikipedia.org/wiki/Sequential_Minimal_Optimization); "`MultiOutputSVM(help)`" shows internal options
- ...
- * `NeuralNetwork` : trains a neural network (see http://en.wikipedia.org/wiki/Artificial_neural_network); "`NeuralNetwork(help)`" shows internal options
- ...

The applications can be further explored by passing help options to the application

```
tungsten: /home/brownbp1/BCL_Workshop_2022% bcl.exe model:Train --help
```

BCL FLAGS: affect general BCL functionality, but may not be relevant for all applications

```
-message_level : adjust the MessageLevel
  <level> minimum level of messages that will be printed, default: "Standard", Allowed values: {Error, Silent, Critical,
Standard, Verbose, Debug}
  <verbosity> set to Detail to print the source file and line of origination for each message, default: "Summary", Allowed
values: {Summary, Detail}
-logger : change the logger this executable uses
  default: "Default", Choose from the following: { Default, File}
  <identifier> define a logger identifier - for file, it is the filename to be opened, optional
-help : output user help for the bcl or an application
-readme : output readme information for the application
...
<enum_file> file that is similar to a written Enums derived class
-random_seed : adjust the random seed; if flag is used, system time is used as seed, if additional parameter is passed, the
given number will be used, otherwise default will be used
  <seed> random seed for the random number generator, default: "5489", range: [0,18446744073709551615]
...
-scheduler : choice of scheduler and number of cpus
  <scheduler> type of scheduler, default: "Serial", Choose from the following: { Serial, PThread}
  <number_cpus> number of cpus for a multi job scheduler, default: "1", range: [1,1000]
```

The applications can be further explored by passing help options to the application

```
tungsten: /home/brownbp1/BCL_Workshop_2022% bcl.exe model:Train --help
```

```
APPLICATION FLAGS: syntax: -flagname [flagparameter1] ...
```

```
...
```

```
-suppress_progress_output : by default, objective function evaluations are shown as the model is training; set this flag to suppress that output
```

```
-max_minutes : maximum # of minutes to train; if # of iterations is reached first, iteration will stop then  
  <minutes> default: "1440", range: [0.5,525600]
```

```
-max_iterations : maximum number of iterations
```

```
  <iterations> default: "18446744073709551615", range: [0,18446744073709551615]
```

```
-max_unimproved_iterations : maximum number of iterations that can pass between improvement steps without stopping the training  
  <iterations> default: "18446744073709551615", range: [0,18446744073709551615]
```

```
-result_averaging_window : Window size for computing the current average result, helps smooth noisy objective functions. Example values:0 - Choose the last model; 1 - Choose the best model on the monitoring
```

```
  dataset; 2 - Choose the best model based on the last two iteration's objective functions on the monitoring dataset according to 1/3 Last round objective function + 2/3 This round objective function. Higher values consider additional previous rounds according to triangularly-weighted average
```

```
  <window_size> default: "1", range: [0,18446744073709551615]
```

```
-final_objective_function : data label for an objective function the evaluates the final model  
  default: "RMSD", choose any implementation of bcl::model::ObjectiveFunctionInterface:
```

```
...
```

Individual parameters within each application can also be probed for more information by passing the help keyword

```
tungsten: /home/brownbp1/BCL_Workshop_2022% bcl.exe model:Train \  
'ApplicabilityDomainKohonen(help) '
```

A kohonen-map based implementation to detect whether a point is within the applicability domain of a model. Inspired by Assessment of applicability domain for multivariate counter-propagation artificial neural network predictive models by minimum Euclidean distance space analysis: A case study

```
Default label : ApplicabilityDomainKohonen(\  
shuffle=0,balance=0,balance max repeats=nan,balance target ratio=1,objective  
function=bcl::model::ObjectiveFunctionInterface,map dimensions="",steps per  
update=0,length=1,radius=0,\  
neighbor kernel=Gaussian,\  
initializer=FirstVectors,\  
scaling=MinMax,cutoff=0.5,share distance metric=1)
```


Individual parameters within each application can also be probed for more information by passing the help keyword

```
tungsten: /home/brownbp1/BCL_Workshop_2022% bcl.exe model:Train \  
'ApplicabilityDomainKohonen(help) '
```

Parameters:

```
<shuffle> primarily for non-batch update; if true, shuffle the order or data points between each run through the data,  
default: "False", Any non-negative integer  
<balance> Whether to automatically balance each class (as defined by the objective function's cutoff, if applicable), default:  
"False", Any non-negative integer  
<balance max repeats> Applies only if balance=True; absolute maximum number of times that a feature can be repeated in order  
to reach the targeted ratio of positives to negatives, default: "1000000",  
Any non-negative integer  
<balance target ratio> Applies only if balance=True; target ratio between most-common and underrepresented class in the  
dataset achieved by data replication; to simulate normal balancing; this should be 1,  
but smaller values may yield more general models, default: "1", Any decimal (floating-point) value >= 0 <= 1  
<objective function> function that evaluates the model after each batch step, default: "Constant(direction=SmallerIsBetter)",  
choose any implementation of bcl::model::ObjectiveFunctionInterface:  
...  
<map dimensions> size of each dimension, grid spacing for each node will always be 1.0, Container with at least 1 Any decimal  
(floating-point) value >= 0 <= 10000  
<steps per update> # of features seen between each update of nodes (set to 0 to use the size of the training data set),  
default: "0", Any non-negative integer  
<length> # of iterations it takes for radius to decrease to 0, default: "10", Any non-negative integer  
...
```

So, when in doubt, use the help menu!

And if that does not work, message somebody on Slack, consult the upcoming tutorials, or look at a manuscript that has an associated protocol capture