ProtocolCapture - Benchmarking Ligand-Based Virtual High-Throughput Screening with the PubChem Database

ProtocolCapture for Manuscript: Benchmarking Ligand-Based Virtual High-Throughput Screening with the PubChem Database

Authors

Mariusz Butkiewicz, Edward W. Lowe Jr., Ralf Mueller, Jeffrey L. Mendenhall, Pedro L. Teixeira, C. David Weaver, and Jens Meiler

Overview

This is the protocol capture for the manuscript : **Benchmarking Ligand-Based Virtual High-Throughput Screening with the PubChem Database**.

Environment

Use these commands to setup your environment to replicate this protocol capture.

If you are replicating this protocol capture, first copy the folder elsewhere and update paths accordingly

```
/bin/tcsh
cd
/blue/meilerlab/projects/QSARPubChemBenchmark2012/2012-11-27-Benchmarking_Ligand-Based_Virtual_High-Throu
protocol_capture_cshrc
```

Protocol Capture

Documentation

- The protocol capture was conducted on Linux CentOS 5! bcl version: 2.5.0; bcl svn revision: 4313;
- download the protocol capture directory structure into a directory of you choice (root)
- Note: every step assumes your root directory is the top level directory of the protocol capture directory!
- The following steps capture the procedure for one of the benchmark data sets (SAID 1798). All other benchmark data sets can be processed in the same way.
- Your licensed BCL executable should go into the directory /bin/bcl/ and be renamed to bcl.exe ! Please make sure you licensed all necessary applications for the BCL::ChemInfoFramework!
- Please make sure you have CORINA installed!

Step	Text	Commands	Comments

Data set generation download AIDs	3. Experimental Section 3.1. Determination of confirmatory high-throughput screening data sets for diverse protein targets Publicly available libraries of small organic molecules from a diverse set of HTS experiments were obtained from PubChem. The following listing of PubChem assays identifies the PubChem summary id (SAID) of the primary protein target and describes the determination of active compounds from confirmatory screens given by PubChem assay ids (AID) (see manuscript). The inactive compounds are taken from the corresponding primary assay. For every SAID a section in the manuscript is explaining which AIDs are involved. The inactive compounds are taken from the main primary for each target. The association of SAIDs and corresponding primary screen AID is given in a later section (Data set generation) below.	• cd /bin/dataset_generation/ * ./aid_download.sh YOUR_AID_NUMBER	• The script aid_download.sh will download the molecules associated with a specific PubChem AID. The result is a .sdf.gz file with all relevant molecules and a .csv file containing the biological data for every compound. • YOUR_AID_NUMBER is the PubChem AID of interest
Data set generation process PubChem AIDS	The raw data given by a .sdf.gz file with all relevant molecules and a .csv file containing the biological data for every compound has to be post-processed to split the actives from the inactive compounds.	• cd /bin/dataset_generation/ * ./molecule_pipeline.sh YOUR_AID_NUMBER.csv YOUR_AID_NUMBER.sdf.gz * #the generate .bin files have to be copied to the /bin/cross_validation_pipeline/data directory * mv ./YOUR_AID_NUMBER_actives.bin ../cross_validation_pipeline/data • mv ./YOUR_AID_NUMBER_inactives.bin ../cross_validation_pipeline/data	• YOUR_AID_NUMBER is the PubChem AID of interest • The script molecule_pipeline.sh will clean up all molecules by removing duplicates, generating 3D coordinates with CORINA, randomize the order and seperate actives from inactives. • If active compounds have no biological value (EC50/IC50) assigned, then a given value is set. • binary files (.bin) containing small molecule descriptors and the associated biological data will be generated for machine learning training • .bin files have to by present in the /bin/cross_validation_pipeline/data directory

Descriptor selection preparation	BCL::ChemInfo is a tailored method that streamlines data processing such as data set generation and cross-validation. The framework hosts a range of small molecule descriptors, descriptor selection strategies, and ML technologies.	• # it is recommended to copy and rename the directory /bin/cross_validation_pipeline to represent the data set designation! The protocol capture will ommit this step! • cd /bin/cross_validation_pipeline • # make sure that bcl.exe symlink is valid in bcl/ • # edit the file include.sh and set all variables accordingly • # adjust the variable dataid to follow the pattern 'aidYOUR_AID_NUMBER', an example is given in the file for aid891. • # variable dataset_size should approximate the number of compounds used in both .bin files to request the right amount of memory from the pbs scheduler • # the section TRAINING OBJ, CUTOFF AND PARITY should set the variable cutoff and parity properly and only one of the objective function string should be unlocked! The training objective function will be applied in every iteration step during training. • # the section ITERATE specifies the chosen machine learning technique. only one set of the variables learning, iterate, and training_chunk_composition • # the section FINAL OBJ FUNCTION specifies the objective function consisting of variables obj_final and obj_final_prefix applied only once at the end of ML training • # all remaining variables should be set accordingly to their description in include.sh	• the set of scripts in /bin/cross_validation_pipeline makes it possible to perform descriptor selection (IG,FS, SFFS) and consensus predictions. • the include script is the main configuration file

Descriptor selection	**Selection of an optimized descriptor set guides QSAR model training** To reduce the total number of inputs to ML algorithms, it is advantageous to remove obsolete descriptors in order to minimize the number of degrees of freedom that need to be determined. Further, noise is reduced while the ratio of data points versus degrees of freedom increases. The determination of an optimal set of descriptors for each data set was evaluated by various selection methods such as Information gain [82], F ? Score [83], and Sequential Forward Feature selection [84].	• cd /bin/cross_validation_pipeline • # get all options for descriptor selection by IG (information gain) and FS (fscore) ./submit_descriptor_reduction.sh • # to launch descriptor selection FS execute: ./submit_descriptor_reduction.sh fscore 2 local 10 600 • # to launch descriptor selection IG execute: ./submit_descriptor_reduction.sh infogain 2 local 10 600 • # to launch descriptor selection by SFFS (sequential feature forward selection) you need to have access to a pbs scheduler queue! The type (SFFS) was set in the include.sh script! ./submit_descriptor_selection.sh start • # Once the descriptor selection has stopped you can retrieve results : cd results/ # for IG and FS: ./results_descriptor_reduction.sh #for SFFS ./results_descriptor_selection.sh	• After choosing the descriptor selection method of choice start your descriptor selection run.

Cross-validation	**Cross-validation ascertains robustness of QSAR models** The active and inactive data sets are divided into ten equal-sized partitions. The first partition is specified as the independent data set which is constant during cross-validation. Of the remaining nine partitions a second partition is selected as the monitoring data set. The remaining eight subsets constitute the training data set. A different monitoring data set is chosen systematically for each iteration of the cross-validation. In a set of ten data partitions each of those ten partitions can be assigned as independent data set leaving nine possibilities of assigning one remaining data partition as the monitoring data set. This results in 10 x 9 = 90 possible model training configurations. All final models trained using the optimized descriptor sets in this study are 10 x 9 -fold cross-validated. This procedure still ensures that every molecule in the data set was part of an independent data partition at least once during cross-validation. Data sets for ANNs and SVMs were balanced by oversampling actives, while decision trees and Kohonen networks required no oversampling. To reduce the computational burden, all descriptor selection schemes use a 5 x 1 = 5 fold cross-validation set up, where the monitoring data partition is systematically incremented but only one independent data set configuration is evaluated.	cd /bin/cross_validation_pipeline # to determine the best performing descriptor set # if you ran descriptor selection by IG or FS retrieve your final descriptor set with: ./get_best_descriptors_by_reduction.sh fscore OR ./get_best_descriptors_by_reduction.sh infogain # if you ran descriptor selection by SFFS retrieve your final descriptor set with: ./get_best_descriptors.sh # to start cross-validation with the best performing descriptor set launch: ./submit_cross_validation.sh # to determine the results of the cross-validation run, execute: cd results ./result_cross_validation.sh roc \ufeff# the results should contain a gnuplot script that can be executed to obtain a graphical representation (.png) gnuplot cv_result.gz.gnuplot	Once the final descriptor set is determined a full 10x9 cross-validation can be applied to determine the objective function of the final cross-validated model. The cross-validated models based on the best performing descriptor set will be stored in the MySQL database. After retrieving the results from the final cross-validation run
Consensus Prediction		• cd /bin/cross_validation_pipeline* # to determine a consensus prediction, copy all available cv_results.gz to a separate directory (eg. /bin/consensus_prediction) and rename each filename to contain the machine learning technique. ./bcl.exe ComputejuryStatistics -input `ls /bin/consensus_prediction` -potency_cutoff 4.0 -table_name table.txt	The application ComputejuryStatistics takes all raw experimental/predicted values of available cross-validation runs and compute the consensus between all possible combinations or raw experimental/predicted value files.

Data set generation

Below are all summary AIDs (SAID) listed which collect all primary, confirmation and counter screens. The active compounds are available

Summary AID (SAID)	Primary Screen (AID)
435008	434989
1798	626
435034	628
1843	1672
2258	2239
463087	449739
488997	488975
2689	2661
485290	485290