MIEC-SVM: Automated Pipeline for Protein Peptide/ligand Interaction Prediction

Nan Li[a], Richard I. Ainsworth[a], Meixin Wu[a], Bo Ding[a], Wei Wang[a]

[a] Department of Chemistry and Biochemistry

University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0359, USA

*corresponding author

Wei Wang: Tel: 858-822-4240. Fax: 858-822-4236. E-mail: wei-wang@ucsd.edu

Supplementary Material

Table of Contents

Part I. Installation of MIEC-SVM Pipeline

1. **System requirements.**
    a. Unix/Linux systems. Fully tested on CentOS 6.
    b. Perl5 with perl core module support.
    c. GNU GCC 4.4 or plus.
    d. GNU Bash as system default shell.

2. **Dependency.** The MIEC-SVM pipeline requires the installation of the following external software:
    a. SCWRL4: a side chain conformation prediction program, freely available to non-profit users. SCWRL 4.0 is currently supported in the pipeline. **It must be installed before the installation of the pipeline.** Web link: http://dunbrack.fccc.edu/scwrl4/
    b. AMBER package: a software package for molecular dynamics simulations. AMBER contains a great number of tools. For the pipeline, programs tleap, sander, MMPBSA.py are needed along with ff03.r1 and gaff force fields. “tleap” and “MMPBSA.py” are freely available in the most recent release of AMBER. “sander” is freely available in AmberTools15. Amber11, Amber14 , AmberTools14, and AmberTools 15 are supported in the pipeline. **At least one version of the Amber package must be installed before the installation of the pipeline.** Web link: http://ambermd.org/
    c. LIBSVM: a software package for support vector machine training and prediction that is freely available. **Version 3.0 is included in the pipeline.** Versions from 3.0 to 3.2 are supported in the pipeline with the modification of printing decision value. Web link: https://www.csie.ntu.edu.tw/~cjlin/libsvm/ How to modify the code for decision value:
    d. R: the project for statistical computing. The MIEC-SVM training part of the pipeline needs the “glmnet” package installed in the R installation. For R version 3.0 and above is supported. For “glmnet”, version 1.9-8 and 2.0-2 are supported. **R-3.1.1 and glmnet 1.9-8 is included in the pipline.** Web link for R: http://www.r-project.org/ ; web link for glmnet: http://cran.r-project.org/web/packages/glmnet/index.html

3. **Installation of the pipeline.**

a. **Set up the environmental variable for the external software.** For SCWRL4, the installation path must be added to the PATH variable. For Amber, AMBERHOME must be added to the environment variable.

Below shows an example of modifying environment variable PATH and AMBERHOME by assuming that SCWRL4 is installed at "/soft/scwrl4" and AMBER is installed at "/soft/amber14":

```
# ~/.bash_profile
export AMBERHOME="/soft/amber14"
PATH=$PATH:/soft/scwrl4
export PATH
```

After modify the ".bash_profile" file, run the source command or re-login to update the environment variables:

```
source ~/.bash_profile
```

b. **Run installation script.** On the command line, run the following command:

```
tar xfz MIEC_pipeline_v1.1.tar.gz
cd MIEC_pipeline_v1.1
./install.pl $PWD
```

The pipeline installation script "install.pl" will check whether all external software have been properly set up. If any external software is not installed or its installation path is not properly added into the $PATH variable, the pipeline installation will stop until the problem is fixed.

Part II. Workflow of MIEC-SVM pipeline

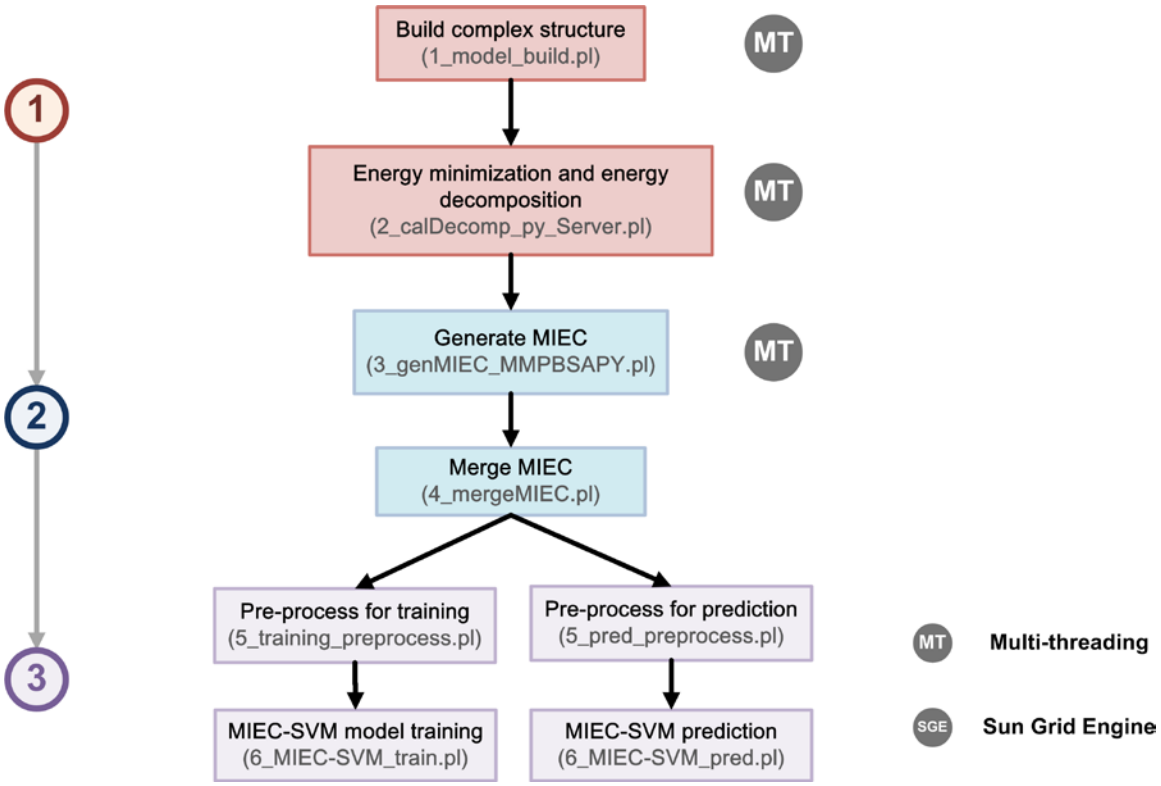

Figure 1. Flow of data for MIEC-SVM pipeline application on servers/workstations.

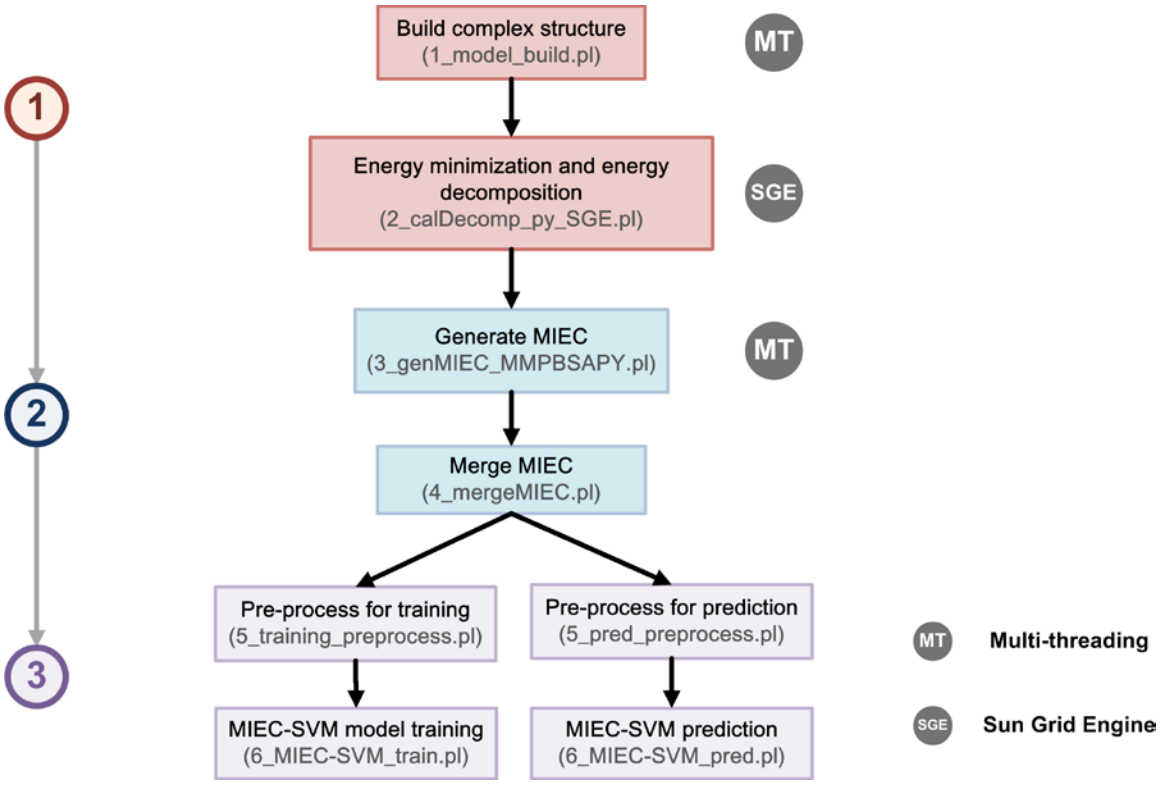

Figure 2. Flow of data for MIEC-SVM pipeline application with HPC (high performance computing) support and SGE job scheduler.

Part III. Tutorial of pipeline examples

All examples are in the directory “example”. There are three examples and each has a full version and a lite version. If you have adequate computational resource, please use the full version; otherwise please try lite version first.

In this tutorial, we will assume that the pipeline is installed at “/soft/MIEC-SVM_pipeline_v1.1/”. Users need to change this path to the real installation path when running all the examples below.

The running time of all the pipeline programs is measured using the server with two Intel Xeon E5-2430v2 (2.5GHz) CPUs and 96G RAM. The storage is formatted in xfs format and mounted to the server through NFS via 10G-BASE Ethernet.

For the three examples:

The hCBX1 example shows how to train MIEC-SVM model from experimental binding information.

The hSUV92 example shows how to use the existing domain-peptide MIEC-SVM model to predict the interaction between the mutant and peptides.

The HIV-PR example shows how to use the existing drug resistance MIEC-SVM model to predict the drug resistance profile of a novel HIV-PR inhibitor ligand.

I. MIEC-SVM model construction for hCBX1

HP1-like chromo domain containing protein was first observed in Drosophila protein HP1. It recognizes H3K9me3 mark and is a component of heterochromatin. hCBX1 is one of the three human homologs of dHP1. A recent peptide microarray study (He, et al., 2015) shows that hCBX1 can recognize a much wider range of trimethylated lysine containing peptides. In this example, we use the interaction data between hCBX1 and 457 peptides to demonstrate the process of building the MIEC-SVM model from scratch. In this case, two hCBX1 template structures are taken from the MD trajectory of 1GUW (the complex structure between mCBX1 and H3K9me3 that has the same sequence as hCBX1). The interactions between hCBX1 and the 457 peptides are identified by the peptide microarray in the (He, et al., 2015).

The hCBX1 example will show users the basic workflow of the MIEC-SVM pipeline and the process of training an MIEC-SVM model from experimentally determined binding data.

Step1. Build complex structures

For the first step, run the following command:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/1_model_build.pl -n 4 -l
peptide_list_CBX -s snapshot_info -d CBX_domain_list -od
MTN_domain_list -om MTN_mutant_list >& log.step1
```

“1_model_build.pl” builds the complex structure for each CBX1-peptide interaction. The interactions are determined using the information from three parts: the template complex structure, the peptide sequence list, and the CBX1 domain information.

Options:

“-n 4” specifies the number of CPU cores used for the script. In this example, four cores are used.

“-l peptide_list_CBX” specifies the peptide list that contains the peptide sequence information for the complex building process. The peptide list is a plain text file with the format shown below:

Peptide Serial	Peptide Sequence	Annotation
9	RT[K_Me3]QTAR[K_Me]S	H3.3-9
10	RT[K_Me]QTAR[K_Me3]S	H3.3-9
11	RT[K_Me3]QTAR[K_Me3]S	H3.3-9
12	[RMe2a]TKQTAR[K_Me]S	H3.3-9
13	[RMe2a]TKQTAR[K_Me3]S	H3.3-9
14	R[pThr]KQTAR[K_Me]S	H3.3-9

The peptide serial number is used to name the peptide in the protein-peptide complexes. The peptide sequence uses the single-letter code for standard amino acids and square brackets for special residues. For example, “[K_Me]” stands for mono-methylated lysine, “[K_Me3]” stands for tri-methylated lysine, “[RMe2a]” stands for di-methylated arginine, and “[pThr]” stands for phosphorylated threonine. The information of special residues must be included in **“dat/mol_info/mol_list”** and **“dat/mol_info/mol_atom_list”** for “1_model_build.pl” to recognize them.

“mol_list” stores the information file of special residues, which contains both residues with PTM and small molecule ligands. The default “mol_list” is located in “dat/mol_info/mol_list” with the following format:

Code in PDB	Code in sequence	Original residue	Terminal mode	N-ter atom	C-ter atom	mol2 name	frcmod name
SPH	pSer	SER	Normal	N	C	SPH.mol2	null
SPN	pSer	SER	Nter	null	C	SPN.mol2	null
SPC	pSer	SER	Cter	N	null	SPC.mol2	null
M3L	K_Me3	LYS	Normal	N	C	M3L.mol2	M3L.frcmod
L10	L10	L10	Normal	null	null	L10.mol2	L10.frcmod

For the “mol_list” file, the first column is the three-letter code used in the PDB file. The second column is the code used in the “peptide_list” file, which is surrounded by the square bracket. The third column is the three-letter code of the un-modified residue for residues with PTM or the three-letter code for small molecule ligands. The fourth column is the terminal mode of the residue: “Nter” means the force field parameter is for the residue on the N-terminus, “Cter” means the force field parameter is for the residue on the C-terminus, and “Normal” means the force field parameter is for residues in the middle of the protein/peptide’s primary sequence. The fifth and sixth columns specify the N-ter and C-ter atom name respectively for connection information that is used in tleap. The seventh and the eighth columns are the file name of the mol2 file with the partial charge for the special residue and the file name of frcmod (force field modification) for the special residue respectively. Both files must be stored in the directory **“dat/parm/”**.

“mol_atom_list” stores the atom name table for residues with PTM. The atom name of residues with PTM is needed to run the “scwrl” program in “1_model_build.pl”, but the atom name of small molecules is not needed. The default “mol_atom_list” is located in **“dat/mol_list/mol_atom_list”** with the following format:

PTM residue name	Atom name in mol2	Atom name in ff03.r1
SPN	N	N
SPN	CA	CA
SPN	CB	CB
SPN	CG	CG
SPN	C	C
SPN	O	O
SPN	P1	null
SPN	O4	null
SPN	O5	null
SPN	O6	null
SPN	H1	H1
SPN	HA	HA
SPN	HB2	HB2
SPN	HB3	HB3
SPN	H2	H2
SPN	H3	H3

The first column is the three-letter code corresponding to “Code in PDB” in the “mol_list” file. The second and third columns are atom names in the residue with PTM and in the

un-modified residue respectively. “null” is used in the third column when there is no corresponding atom in the un-modified residue.

“-s snapshot_info” specifies the file location of the template complex structures. Below shows the snapshot_info file used in the CBX1 example:

```
MS19    1_template/CBX1-1_MS19_455.pdb
MS20    1_template/CBX1-1_MS20_455.pdb
```

The first column specifies the snapshot name which will be used to construct the file name for complex structure. The second column is the location of the complex template structure file with either the relative path or the absolute path to the file. In the CBX1 example, two snapshots from MD simulation are used for construction of the complex structures. At the final stage of MIEC construction, the mean MIEC profile will be generated from the MIEC profiles of both snapshots.

“-d CBX_domain_list” is the complex information file. It has two sub-types: the domain information file and the mutant information file. The domain information file specifies the related information for the complex template structures. The mutant information file specifies the related information for the modeled complexes. Below shows the formats of the complex information file:

```
1       CBX1-1  :1-50   :51-59  :58     NA      CBX1-1
```

The first column is the serial number. The second column is the domain name. The third and fourth columns are the residue range of the receptor and the ligand respectively. The range starts with the a colon, followed by the range of residues. If the ligand only contains one residue, the user can use a colon plus residue number to specify the ligand residue range. The fifth column 5 is the residues whose side chain conformation should not change during the virtual mutagenesis by scwrl. In this example, it is used to check if the tri-methylated lysine has the correct side chain conformation. The sixth column is the range of residues that are considered as frame residues. We will explain this column in the HIV-protease drug resistance example. The seventh column is a repeat of domain name, which will not be used here.

“-od MTN_domain_list” and “-om MTN_mutant_list” are two outputs of complex information files that contains the information for all complexes built in this step. In the complex building step, complex structures from the combination of domains and peptides will be built. The information of domains will be stored in the file specified by the “-od” option and the information of each domain-peptide complex will be stored in the file specified by the “-om” option.

Running result:

If “1_model_build.pl” finishes correctly, it will return the number of complexes successfully processed. In this example, the log file “log.step1” after a successful run contains the following information:

Full version:

```
914 complexes successfully processed
```

Lite version:

```
100 complexes successfully processed
```

All the scwrl processing files are stored in directory “2_scwrl_mutation”. All the tleap processing files are stored in directory “3_tleap_mutation”. The symbolic link files are stored in directory “4_tleap_mutation_renum”.

If any problems occur during the process, the log file will print out which complex fails and at which step the failure occurs. Such as:

```
CBX1-1_MS19_23 Scwrl failed
CBX1-1_MS20_27 tleap failed
```

The user can use this information and check the log files in either “2_scwrl_mutation” or “3_tleap_mutation” to correct any errors in their input PDB file. Before any further debug, please check first if the residue number is compatible to the pipeline. **For the correct PDB file of the template complex structure, the residue number starts from 1 on the receptor and the residue number should be consecutive till the last residue on the ligand. No gaps in the residue numbers are allowed.**

We run the above command on a server with Xeon E5-2430v2 (2.5GHz). The total running time is about 3 minutes for the full version.

Step2. Energy Minimization and Decomposition

In this step, the modeled complex structure from step 1 will be optimized by energy minimization. Then, the residue-residue energy profile of the optimized structure is calculated using MMPBSA.py in AMBER. This step requires considerable computational resources. Therefore, HPC cluster support is implemented at this step. Nevertheless, working with a standalone server or workstation is still practical when the number of complexes is small. For HPC clusters using an SGE job scheduler, users should use “2_calDecomp_py_SGE.pl”. For standalone servers, users should consider “2_calDecomp_py_Server.pl”.

On a cluster running an SGE job scheduler, run the following command for energy minimization and decomposition:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/2_calDecomp_py_SGE.pl –m
MTN_mutant_list >& log.step2
```

On standalone servers or workstations, run the following command for energy minimization and decomposition:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/2_calDecomp_py_Server.pl –m
MTN_mutant_list –n 4 >& log.step2
```

Options:

"-m MTN_mutant_list", is the mutant information file for all complex structures to be processed. The file is generated by "1_model_build.pl" in the "–om" option.

-n number of CPUs used.

Running result:

For the both versions of the 2nd step programs, the log file contains the number of complexes that have been successfully processed. The "SGE" version will also keep checking the number of jobs that have completed and print out the process into the log file.

All minimized pdb files are stored in the directory "5_minimization". All energy decomposition files are stored in the directory "6_decomposition". Intermediate files are stored in the directory "mutant_process". If any error occurs, the user can check the files in "mutant_process" to find the problem.

For each complex, the running time of minimization and energy decomposition on the test server takes 6-7 minutes for the full version.

Step3. MIEC profile generation

In this step, MIEC profiles will be built from the energy decomposition files obtained from step 2. The command for the CBX1 example is:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/3_genMIEC_MMPBSAPY.pl -n 4 -d
MTN_domain_list -ml MTN_mutant_list –pn 9 -ms
```

```
0_para_files/Chromo_MSA.fa -pl 0_para_files/chromo_pair_list_10A.list -
o MIEC_10A_MMGBSA_CBX1.out >& log.step3
```

Options:

-n number of CPUs used.

"-d MTN_domain_list" is the complex information file for all domains included in the MIEC profile generation. It links each complex to the respective multiple sequence alignment information. The file is generated by "1_model_build.pl" in the "–od" option.

"-m MTN_mutant_list" is the complex information file for all mutants included in the MIEC profile generation. It provides the lookup information for mutant name. The file is generated by "1_model_build.pl" in the"–om" option.

"-ms 0_para_files/Chromo_MSA.fa" is the multiple sequence alignment file for chromo domain. "3_genMIEC_MMPBSAPY.pl" requires a FASTA format multiple sequence alignment file. The domain names used in the multiple sequence alignment file should be consistent with the domain name in the complex information file "MTN_domain_list".

"-pl 0_para_files/chromo_pair_list_10A.list" is the pair list file for MIEC generation. The pair list file specifies the receptor-ligand residue pairs. Each row of the file corresponds to one residue pair. The first residue number is for the receptor and corresponds to the general residue number in the multiple sequence alignment, which counts in gaps and has the same number for all domains in the MSA. The second residue number is for the ligand, where the C-ter residue is assigned to 0 and the N-ter residue is assigned to the negative value of its sequence distance to the C-ter residue.

Running result:

"3_genMIEC_MMPBSAPY.pl" will output the percentage of finished complexes if everything goes smoothly. If there are any error messages in the log file, users need to check the format of the input files.

In this example, the MIEC profile is output as "MIEC_10A_MMGBSA_CBX1.out". The first column of the MIEC profile is the complex name. The second column is the binding information, 1 for binder and 0 for non-binder; if binding information is not known, 0 is used for default. The rest of the columns are energy values for each residue pair. In this example, each residue pair has four energy values, i.e. van de Waals, electrostatics, generalized Born, and surface area. There are 158 receptor-ligand pairs and 8 ligand-ligand pairs between adjacent ligand residues. Therefore, the number of MIEC

components for each complex is (158+8)*4=664. For each row, the column number is 664+2=666.

The running time is about 8 minutes on the test server for the full version.

Step4. Merge MIEC

This step is not needed for this example. Check the tutorial of HIV protease drug resistance for the details.

Step5. Pre-process for training

The MIEC profile needs to be pre-processed before the MIEC-SVM training. The pre-process in this example contains two steps: the scaling of each column and the integration of binding information into the profile. For the pre-process of the CBX1 example, run the following command:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/5_training_preprocess.pl -d
MTN_domain_list -m MIEC_10A_MMGBSA_CBX1.out -b 0_para_files/CBX1_binder
-or 0_para_files/CBX1.range.new -om MIEC_10A_MMGBSA_CBX1_scaled.out >&
log.step5
```

Options:

“-d MTN_domain_list” is the complex information file for CBX1 domain.

“-m MIEC_10A_MMGBSA_CBX1.out” is the MIEC profile generated from step 3.

“-b 0_para_files/CBX1_binder” is a list of binding complexes. It is a plain text file of all the experimentally determined binders and such information will be integrated into the MIEC profile.

“-or 0_para_files/CBX1.range.new” is the output scaling parameter file which is needed for the further step “Pre-process for prediction”. The scaling method used here linearly scales each energy component into values between -1 and 1 for each column.

“-om MIEC_10A_MMGBSA_CBX1_scaled.out” is the output MIEC profile file after pre-process.

Running result:

The running time for this step is less than 1 minute for the full version.

Step6. MIEC-SVM training

The training process contains two steps. We start with the LASSO logistic regression to select the most important features to the classification. Then, we train the SVM model with the selected feature. For the LASSO step, a list of lambda is needed for feature selection. Each lambda corresponds to one set of selected features and a feature selected MIEC profile. The SVM model is trained upon each feature selected MIEC profile. And the 3-fold cross validation is performed on the same feature selected MIEC profile to evaluate the performance of the SVM model. Users can select the best model to make future predictions.

To perform the MIEC-SVM training, run the following command:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/6_MIEC-SVM_train.pl -m
MIEC_10A_MMGBSA_CBX1_scaled.out -lm 0_para_files/lambda_list_100 -r
training.result >& log.step6
```

Options:

"-m MIEC_10A_MMGBSA_CBX1_scaled.out" is the MIEC profile after the pre-process and is generated from the previous step.

"-lm 0_para_files/lambda_list_100" is a list of lambda values for LASSO logistic regression based feature selection.

"-r training.result" is the cross validation result for all candidate SVM models. The first column is the name of the model. The remaining columns are sensitivity (SEN), specificity (SPC), accuracy+ (ACC+), accuracy- (ACC-), accuracy (ACC), MCC, ROC AUC, and PR AUC respectively. The definitions of sensitivity, specificity, accuracy+, accuracy-, accuracy, and MCC are:

SEN = TP/(TP+FN)

SPC = TN/(TN+FP)

ACC+ = TP/(TP+FP)

ACC- = TN/(TN+FN)

ACC = (TP+TN)/(TP+FP+TN+FN)

MCC = (TP×TN- FP×FN)/sqrt[(TP+FP) ×(TP+FN) ×(TN+FP) ×(TN+FN)]

Running result:

The performance of each selected feature set (SVM model) is stored in the file "training.result". All SVM models are stored in directory "13_MIEC-SVM_model". All feature subset files are stored in directory "11_MIEC_subset". All feature selected MIEC profiles are stored in directory "12_MIEC_lassoFS". Users can select the appropriate file for future MIEC-SVM prediction.

We run the above command on a server with Xeon E5-2430 and 96G RAM. The total running time is about 36 minutes for the full version.

II. MIEC-SVM model prediction for hSUV92

SUV92 is an H3K9 specific methyltransferase and can recognize the H3K9me3 mark with its chromo domain. The peptide microarray study shows that the binding specificity can be changed by altering a few key residues on its chromo domain. In this example, wild type SUV92 is mutated into two mutants and their binding specificity to the 457 peptides is predicted by the chromo domain MIEC-SVM model.

Step1. Build complex structures

For the first step, run the following command:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/1_model_build.pl -n 4 –m mutant_info -l
peptide_list_CBX -s snapshot_info -d CBX_domain_list -od
MTN_domain_list -om MTN_mutant_list >& log.step1
```

The 1_model_build.pl will mutate SUV92 according to the file "mutant_info" and then build the complex structure for each SUV92 mutant and peptide combination. The input information contains four parts: the template complex structure, the mutant information, the peptide sequence list, and the CBX1 domain information.

Options:

"-n 4" specifies that four CPU cores will be used for the script;

"-m mutant_info" specifies the mutant information. The format is shown below:

```
MTN1   LEU3 LEU40 PHE43
```

The first column is the mutant name. The rest of columns show the mutation position and mutated residue.

“-l peptide_list” specifies the information of the peptide being considered in the complex model process. See tutorial for CBX1 for details of the peptide list.

“-s snapshot_info” specifies the file location of the template complex structures. See tutorial for CBX1 for details of the snapshot information file.

“-d CBX_domain_list” is the mutant information file that specifies domain information for all domains. See tutorial for CBX1 for details of the mutant information file.

“-od MTN_domain_list” and “-om MTN_mutant_list” are two output mutant information files that contain information for all complexes built in this step. In the complex building step, complex structures from the combination of protein receptors and peptides will be built. The information for protein receptors will be stored in the file specified by the “-od” option and the information for each receptor-peptide complex will be stored in the file specified by the “-om” option.

Running result:

If “1_model_build.pl” finishes correctly, it will return the number of complexes successfully processed. In this example, the log file “log.step1” contains the following:

Full version:

```
1828 complexes successfully processed
```

Lite version:

```
50 complexes successfully processed
```

All the scwrl processing files are stored in directory “2_scwrl_mutation”. All the tleap processing files are stored in directory “3_tleap_mutation”. The symbolic link files are stored in directory “4_tleap_mutation_renum”.

If any problem occurs during the process, the log file will print out which complex failed and at which step the failure happened. Such as:

```
SUV92-1-MTN1_MS20_23 Scwrl failed
SUV92-1-MTN1_MS20_27 tleap failed
```

The user can use this information and check the log files in either “2_scwrl_mutation” or “3_tleap_mutation” to correct any errors in their input PDB file. Before any further debug, please check first if the residue number is compatible to the pipeline. For the compatible

template complex structure PDB file, the residue number starts from 1 on the receptor and the residue number is consecutive till the last residue on the ligand. No gap in the residue numbers are allowed.

We run the above command on a server with Xeon E5-2430v2 (2.5GHz). The total running time is about 2 minutes for the full version.

Step2. Energy Minimization and Decomposition

In this step, the modeled complex structure from step 1 will be optimized by energy minimization. Then, the residue-residue energy profile of the optimized structure is calculated using MMPBSA.py in AMBER. This step requires considerable computational resources. HPC cluster support is preferred for this step if the number of complex structures is large. Working with a standalone server or workstation is only practical when the number of complexes is small. For HPC clusters using the SGE job scheduler, users should use "2_calDecomp_py_SGE.pl". For standalone server, users should consider "2_calDecomp_py_Server.pl".

On a cluster running an SGE job scheduler, run the following command for energy minimization and decomposition:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/2_calDecomp_py_SGE.pl –m
MTN_mutant_list >& log.step2
```

On standalone servers or workstations, run the following command for energy minimization and decomposition:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/2_calDecomp_py_Server.pl –m
MTN_mutant_list –n 4 >& log.step2
```

Options:

"-m MTN_mutant_list", is the mutant information file for all complex structures to be processed. The file is generated by "1_model_build.pl" in its "–om" option.

-n number of CPUs used.

Running result:

For the both versions of the script, the log file contains the number of complexes that have been successfully processed.

All minimized pdb files are stored in the directory “5_minimization”. All energy decomposition files are stored in the directory “6_decomposition”. Intermediate files are stored in the directory “mutant_process”. If any error occurs, the user can check the files in “mutant_process” to figure out the problem.

For each complex, the time of minimization and energy decomposition on a server of Xeon E5-2430v2 (2.5GHz) is 6-7 minutes.

Step3. MIEC profile generation

In this step, MIEC profiles will be built from the energy decomposition files obtained from the last step. The command line for the SUV92 example is:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/pipAux_genMSA_byMutant.pl
0_para_files/Chromo_MSA.fa SUV92-1 mutant_info 0_para_files/SUV92MTN.fa

/soft/MIEC-SVM_pipeline_v1.1/bin/3_genMIEC_MMPBSAPY.pl -n 4 -d
MTN_domain_list -ml MTN_mutant_list –pn 9 -cl
0_para_files/BS_Polar.list -ms 0_para_files/SUV92MTN.fa -pl
0_para_files/chromo_pair_list_10A.list -o
MIEC_10A_MMGBSA_SUV92MTN.out >& log.step3
```

The first command generates the multiple sequence alignment file for the SUV92 mutants. The second command generates the MIEC profile for the SUV92 mutants.

Options:

-n number of CPUs used.

“-d MTN_domain_list” is the information file for all domains included in the MIEC generation. It provides the lookup information for the multiple sequence alignment file. The file is generated by “1_model_build.pl” in its “–od” option.

“-m MTN_mutant_list” is the information file for all mutants included in the MIEC generation. It provides the lookup information for mutant name. The file is generated by “1_model_build.pl” in its “–om” option.

“-ms 0_para_files/Chromo_MSA.fa” is the multiple sequence alignment file for chromo domain. “3_genMIEC_MMPBSAPY.pl” requires a FASTA format multiple sequence alignment file. The domain names used in the multiple sequence alignment file need to be consistent with the domain name in the information file “MTN_domain_list”.

"-pl 0_para_files/chromo_pair_list_10A.list" is the pair list file for MIEC generation. The pair list file specifies the receptor-ligand residue pairs. Each row of the file corresponds to one residue pair. The first residue number is for the receptor and corresponds to the general residue number in the multiple sequence alignment, which counts in gaps and has the same number for all domains in the MSA. The second residue number is for the ligand, where the C-ter residue is assigned to 0 and N-ter residue is assigned to the negative value of its sequence distance to the C-ter residue.

"-cl 0_para_files/BS_Polar.list" is the MIEC component list file for MIEC generation. The file specifies the combination of MIEC component in generation of MIEC profile. The file has the following format:

```
BVDW
SVDW
BELE BGB
SELE SGB
BGBSUR
SGBSUR
```

The file specifies that the "BELE" and "BGB" will combine into one component as well as "SELE" and "SGE". The keywords for MIEC components include: TVDW, TELE, TGB, TGBSUR, BVDW, BELE, BGB, BGBSUR, SVDW, SELE, SGB, and SGBSUR. The "TXXX" components are the total binding energy contribution between the two residues. The "BXXX" components are the backbone related binding energy contributions between the two residues, each residue pair has two "BXXX" values which are the binding energy between the backbone of residue 1 and all atoms of residue 2 and vice versa. The "SXXX" components are the side-chain related binding energy contributions between the two residues, each residue pair has two "SXXX" values which are the binding energy between the side-chain of residue 1 and all atoms of residue 2 and vice versa. Users can specify any combination of these components to generate the respective MIEC profile.

Running result:

"3_genMIEC_MMPBSAPY.pl" will output an empty log file if everything goes smoothly. If there are any error messages in the log file, users should check the format of their input files.

In this example, the MIEC profile is output as "MIEC_10A_MMGBSA_SUV92MTN.out". The first column of the MIEC profile is the complex name. The second column is the binding information, 1 for binder and 0 for non-binder; if binding information is not known, 0 is used for default. The rest of the

columns are energy values for each residue pair. In this example, each residue pair has twelve energy values (each component has two values), i.e. backbone van de Waals (BVDW), sidechain van de Waals (SVDW), backbone polar (BELE+BGB), sidechain polar (SELE+SGB), backbone surface area (BGBSUR), and sidechain surface area (SGBSUR). There are 158 receptor-ligand pairs and 8 ligand-ligand pairs between adjacent ligand residues. Therefore, the number of MIEC components for each complex is (158+8)*12=1992. For each row, the column number is 1992+2=1994.

We run the above command on a server with Xeon E5-2430 and 96G RAM. The total running time is about 8 minutes for the full version.

Step4. Merge MIEC

This step is not needed for this example. Check the HIV protease drug resistance tutorial for details.

Step5. Pre-process for prediction

The MIEC profile needs to be pre-processed before the MIEC-SVM prediction. The pre-process in this example contains two steps: scaling each column into -1 to 1 and applying the pre-defined feature selection set. For the pre-process of the SUV92 example, run the following command:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/5_pred_preprocess.pl -d
MTN_domain_list -m MIEC_10A_MMGBSA_SUV92MTN.out -r SUV92MTN.range -s
0_para_files/MIEC_model.subEZ –om
MIEC_10A_MMGBSA_SUV92MTN_scaled.out >& log.step5
```

Options:

“-d MTN_domain_list” is the information file for the CBX1 domain.

“-m MIEC_10A_MMGBSA_SUV92MTN.out” is the MIEC profile generated from the last step.

“-r SUV92MTN.range” is the output scaling information file for SUV92 mutants.

“-s 0_para_files/MIEC_model.subEZ” is the pre-defined selected feature set which will be applied to “MIEC_10A_MMGBSA_SUV92MTN.out”. This file should be consistent with the MIEC-SVM model that is used in prediction.

"-om MIEC_10A_MMGBSA_CBX1_scaled.out" is the output scaled MIEC profile file.

Running result:

We run the above command on a server with Xeon E5-2430 and 96G RAM. The total running time is less than 1 minute for the full version.

Step6. MIEC-SVM prediction

To perform the MIEC-SVM prediction, run the following command:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/6_MIEC-SVM_pred.pl -ml
0_para_files/MIEC-SVM.model -mt MIEC_10A_MMGBSA_SUV92MTN_scaled.out -r
pred_result_SUV92 >& log.step6
```

Options:

"-ml 0_para_files/MIEC-SVM.model" is the pre-trained chromo domain MIEC-SVM model for the prediction of the SUV92 mutants.

"-mt MIEC_10A_MMGBSA_SUV92MTN_scaled.out" is the MIEC profile after the pre-process step, generated from the previous step.

"-r pred_result_SUV92" is the prediction result. The format of the prediction result is:

```
1       1       SUV92-1-MTN1_81 0       0       -0.950513
1       1       SUV92-1-MTN1_82 0       0       -0.086495
1       1       SUV92-1-MTN1_83 0       0       -0.250127
1       1       SUV92-1-MTN1_84 0       1       0.069993
1       1       SUV92-1-MTN1_85 0       1       0.273403
1       1       SUV92-1-MTN1_86 0       1       0.241967
```

The first two columns are for internal use. The third column is the complex name. The fourth column is the binder (1)/non-binder (0) flag in the MIEC profile. The fifth column is the predicted interaction type: binder (1) or non-binder (0). The sixth column is the decision value from SVM, positive for binder and negative for non-binder.

Running result:

All predictions are in the file "pred_result_SUV92".

We run the above command on a server with Xeon E5-2430 and 96G RAM. The total running time is less than 1 minute for the full version.

III. HIV Protease Drug Resistance Prediction

HIV protease is one of the primary targets for HAART (Highly Active Antiretroviral Therapy). However, drug resistance becomes the major hurdle for the continued effectiveness of inhibitors to HIV protease and other HIV target proteins. In this example, the complex of wild type HIV protease and a potential inhibitor ligand is mutated into 2382 previously identified HIV protease mutants. Then the drug resistance profile is predicted using the HIV protease drug resistance MIEC-SVM model to evaluate the drug resistance profile of the ligand.

Step1. Build complex structures

For the first step, run the following command:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/1_model_build.pl -n 4 -m mutant_info -s
snapshot_info -d PR_domain_list -od MTN_domain_list -om
MTN_mutant_list >& log.step1
```

The 1_model_build.pl will mutate the HIV protease according to the file “mutant_info” and then build the complex structure of each HIV protease mutant and ligand. The input information contains three parts: the template complex structure, the mutant information, and the HIV protease domain information.

Options:

“-n 4” specifies the four CPU cores will be used for the script;

“-m mutant_info” specifies the mutant information. See the SUV92 mutant tutorial for details of the mutant information file.

“-l peptide_list” specifies the information of peptide considered in the complex model process. See the CBX1 tutorial for details of the peptide list.

“-s snapshot_info” specifies the file location of template complex structures. See the CBX1 tutorial for details of the snapshot information file.

“-d PR_domain_list” is the mutant information file that specifies domain information for HIV protease. See the CBX1 tutorial for details of the mutant information file.

"-od MTN_domain_list" and "-om MTN_mutant_list" are the two outputs of mutant information files that contain information for all complexes built in this step. In the complex building step, complex structures from the combination of protein receptors and peptides will be built. The information for protein receptors will be stored in the file specified by the "-od" option and the information of each receptor-peptide complex will be stored in the file specified by the "-om" option.

Running result:

If "1_model_build.pl" finishes correctly, it will return the number of complexes successfully processed. In this example, the log file "log.step1" contains the following:

Full version:

```
2383 complexes successfully processed
```

Lite version:

```
40 complexes successfully processed
```

All the scwrl processing files are stored in directory "2_scwrl_mutation". All the tleap processing files are stored in directory "3_tleap_mutation". The symbolic link files are stored in directory "4_tleap_mutation_renum".

If any problem occurs during the process, the log file will print out which complex failed and at which step the failure happened. Such as:

```
PR-MTN1_L10 Scwrl failed
PR-MTN3_L10 tleap failed
```

The user can use this information and check the log files in either "2_scwrl_mutation" or "3_tleap_mutation" to correct any errors in their input PDB files. Before any further debug, please check if the residue number is compatible with the pipeline. For the compatible template complex structure PDB file, the residue number starts from 1 on the receptor and the residue number is consecutive until the last residue on the ligand. No gaps in the residue numbers are allowed.

We run the above command on a server with Xeon E5-2430v2 (2.5GHz). The total running time is about 10 minutes for the full version.

Step2.1 Energy Minimization and Decomposition

In this step, the modeled complex structure from step 1 will be optimized by energy minimization. Then, the residue-residue energy profile of the optimized structure is

calculated using MMPBSA.py in AMBER. This step requires considerable computational resources. HPC cluster support is preferred for this step if the number of complex structures is large. Working with a standalone server or workstation is only practical when the number of complexes is small. For HPC clusters using an SGE job scheduler, users should use “2_calDecomp_py_SGE.pl”. For standalone servers, users should consider “2_calDecomp_py_Server.pl”.

On a cluster running an SGE job scheduler, run the following command for energy minimization and decomposition:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/2_calDecomp_py_SGE.pl –m
MTN_mutant_list >& log.step2
```

On standalone servers or workstations, run the following command for energy minimization and decomposition:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/2_calDecomp_py_Server.pl –m
MTN_mutant_list –n 4 >& log.step2
```

Options:

“-m MTN_mutant_list”, is the mutant information file for all complexes structures to be processed. The file is generated by “1_model_build.pl” in its “–om” option.

-n number of CPUs used.

Running result:

For both versions of the script, the log file contains the number of complexes that have been successfully processed.

All minimized pdb files are stored in the directory “5_minimization”. All energy decomposition files are stored in the directory “6_decomposition”. Intermediate files are stored in the directory “mutant_process”. If any error occurs, the user can check the files in “mutant_process” to figure out the problem.

For each complex, the time of minimization and energy decomposition on a server of Xeon E5-2430v2 (2.5GHz) is 6-7 minutes for the full version.

Step2.2 Polar and non-polar surface area calculation

In the HIV protease example, the MIEC profile contains the two components not from MM/GBSA: the surface area from polar and non-polar atoms. The polar and non-polar surface area uses the sasa_phi program in the external program directory. To calculate the polar and non-polar surface area, run the following command:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/2_calDecomp_py_Server.pl -m
MTN_mutant_list -n 4 -t 1 >& log.step2.sasa
```

Options:

This script must be run after all minimized pdb files are obtained (after finishing Step2.1).

"-t 1", flag for energy decomposition, 0 for MM/GBSA, 1 for sasa.

Running result:

The sasa result is located in directory "6_decomposition" with suffix "_sasa.out".

For all 2383 complexes, the time for the sasa calculation takes about 14 minutes for the full version.

Step3.1. MIEC profile generation (MM/GBSA)

In this step, MIEC profiles will be built from the energy decomposition files obtained from the last step. The command for the HIV-PR example is:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/pipAux_genMSA_byMutant.pl
0_para_files/HIV-PR_MSA.fa PR mutant_info 0_para_files/ HIV-PR-
MTN_MSA.fa

/soft/MIEC-SVM_pipeline_v1.1/bin/3_genMIEC_MMPBSAPY.pl -n 4 -d
MTN_domain_list -ml MTN_mutant_list –ms 0_para_files/HIV-PR-MTN_MSA.fa
-pl 0_para_files/HIV-PR_pair_list -cl 0_para_files/HIV-
PR_component_list -o MIEC_PR_MTN.out >& log.step3
```

The first command generates the multiple sequence alignment file for the HIV protease mutants. The second command generates the MIEC profile for the HIV protease mutants.

Options:

-n number of CPUs used.

"-d MTN_domain_list" is the information file for all domains included in the MIEC generation. It provides the lookup information for the multiple sequence alignment file. The file is generated by "1_model_build.pl" in its "–od" option.

"-m MTN_mutant_list" is the information file for all mutants included in the MIEC generation. It provides the lookup information for mutant name. The file is generated by "1_model_build.pl" in its "–om" option.

"-ms 0_para_files/HIV-PR-MTN_MSA.fa" is the multiple sequence alignment file for HIV protease mutants. "3_genMIEC_MMPBSAPY.pl" requires a FASTA format multiple sequence alignment file. The domain names used in the multiple sequence alignment file need to be consistent with the domain name in the information file "MTN_domain_list".

"-pl 0_para_files/HIV-PR_pair_list" is the pair list file for MIEC generation. The pair list file specifies the receptor-ligand residue pairs. Each row of the file corresponds to one residue pair. The first residue number is for the receptor and corresponds to the general residue number in the multiple sequence alignment, which counts in gaps and has the same number for all domains in the MSA. The second residue number is for the ligand, where the C-ter residue is assigned to 0 and N-ter residue is assigned to the negative value of its sequence distance to the C-ter residue. In the HIV protease example, the combination of symmetric residues of protease is implemented through specifying multiple pairs in one row.

"-cl 0_para_files/HIV-PR_component_list" is the MIEC component list file for MIEC generation. The file specifies the combination of MIEC components in generation of MIEC profile.

Running result:

"3_genMIEC_MMPBSAPY.pl" will output an empty log file if everything goes smoothly. If there are any error messages in the log file, users need to check the format of their input files.

We run the above command on a server with Xeon E5-2430 and 96G RAM. The total running time is about 43 minutes for the full version.

Step3.2. MIEC profile generation (sasa)

In this step, MIEC profiles will be built from the energy decomposition files obtained from the last step. The command for the HIV-PR example is:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/pipAux_genMIEC_sasa_HIV.pl -n 4 -d
MTN_domain_list -ml MTN_mutant_list –ms 0_para_files/HIV-PR-MTN_MSA.fa
-pl 0_para_files/HIV-PR_pair_list -o MIEC_PR_MTN_SASA.out >&
log.step3.SASA
```

The first command generates the multiple sequence alignment file for the HIV protease mutants. The second command generates the MIEC profile for the SUV92 mutants.

Options:

-n number of CPUs used.

“-d MTN_domain_list” is the information file for all domains included in MIEC generation. It provides the lookup information for the multiple sequence alignment file. The file is generated by “1_model_build.pl” in its “–od” option.

“-ml MTN_mutant_list” is the information file for all mutants included in the MIEC generation. It provides the lookup information for mutant names. The file is generated by “1_model_build.pl” in its “–om” option.

“-ms 0_para_files/HIV-PR-MTN_MSA.fa” is the multiple sequence alignment file for HIV protease mutants. “3_genMIEC_MMPBSAPY.pl” requires a FASTA format multiple sequence alignment file. The domain names used in the multiple sequence alignment file need to be consistent with the domain name in the information file “MTN_domain_list”.

“-pl 0_para_files/HIV-PR_pair_list” is the pair list file for MIEC generation. The pair list file specifies the receptor-ligand residue pairs. Each row of the file corresponds to one residue pair. The first residue number is for the receptor and corresponds to the general residue number in the multiple sequence alignment, which counts in gaps and has the same number for all domains in the MSA. The second residue number is for the ligand, where the C-ter residue is assigned to 0 and N-ter residue is assigned to the negative value of its sequence distance from the C-ter residue. In the HIV protease example, the combination of symmetric residues of protease is implemented through specifying multiple pairs in one row.

Running result:

“pipAux_genMIEC_sasa_HIV.pl” will output an empty log file if everything goes smoothly. If there are any error messages in the log file, users need to check the format of their input files.

We run the above command on a server with Xeon E5-2430 and 96G RAM. The total running time is about 8 minutes for the full version.

Step4. Merge MIEC

In this example, the MIEC profile from MM/GBSA and the MIEC profile from sasa need to be combined into a single MIEC profile. To combine the two MIEC profiles, use the following command:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/4_mergeMIEC.pl -m MTN_mutant_list -pl
0_para_files/HIV-PR_pair_list -mcb MIEC_PR_MTN.out -mca
MIEC_PR_MTN_SASA.out -o MIEC_PR-MTN_combined.out >& log.step4
```

Options:

"-m MTN_mutant_list" is the information file for all mutants included in the MIEC generation. It provides the lookup information for the mutant name. The file is generated by "1_model_build.pl" in its "–om" option.

"-pl 0_para/HIV-PR_pair_list" is the pair list file for MIEC generation.

"-mca MIEC_PR_MTN_SASA.out" is the first MIEC profile to combine.

"-mcb MIEC_PR_MTN.out" is the second MIEC profile to combine.

"-o MIEC_PR-MTN_combined.out" is the combined MIEC profile output.

Running result:

"4_mergeMIEC.pl" will generate an empty log file if there are no errors during the run time. The output file "MIEC_PR-MTN_combined.out" is the file needed for the next step.

We run the above command on a server with Xeon E5-2430 and 96G RAM. The total running time is less than 1 minute.

Step5. Pre-process for prediction

The MIEC profile needs to be pre-processed before the MIEC-SVM prediction. The pre-process in this example contains two steps: scale each column into -1 to 1and apply the

pre-defined feature selection set. For the pre-process of the HIV-PR example, run the following command:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/5_pred_preprocess.pl -d
MTN_domain_list -t 1 -m MIEC_PR-MTN_combined.out -r 0_para_files/HIV-
PR.range -s 0_para_files/HIV-PR.subEZ -o MIEC_PR-
MTN_combined_scaled.out
```

Options:

"-d MTN_domain_list" is the information file for the HIV protease mutants.

"-m MIEC_PR-MTN_combined.out" is the MIEC profile generated from the last step.

"-r 0_para_files/HIV-PR.range" is the output scaling information file for the HIV protease mutants.

"-s 0_para_files/HIV-PR.subEZ" is the pre-defined selected feature set which will be applied to "MIEC_PR-MTN_combined.out". This file should be consistent with the MIEC-SVM model that is used in prediction.

"-o MIEC_PR-MTN_combined_scaled.out" is the output scaled MIEC profile file.

Running result:

We run the above command on a server with Xeon E5-2430 and 96G RAM. The total running time is less than 1 minute for the full version.

Step6. MIEC-SVM prediction

The training process contains two steps. We start with the LASSO logistic regression to select the most important features for accurate classification. Then, we train the SVM model with the selected features. For the LASSO step, a list of lambda is needed for feature selection. Each lambda corresponds to a set of selected features. For each set of selected features, a 3-fold cross validation is performed using SVM to evaluate the performance of the model as well as the set of selected features. Finally, the user can select the best model from all the candidate models.

To perform the MIEC-SVM training, run the following command:

```
/soft/MIEC-SVM_pipeline_v1.1/bin/6_MIEC-SVM_pred.pl -ml
0_para_files/MIEC-SVM_PR-LPV.model -mt MIEC_PR-MTN_combined_scaled.out
-r pred_result >& log.step6
```

Options:

"-ml 0_para/MIEC-SVM_PR-LPV.model" is the pre-trained chromo domain MIEC-SVM model for the prediction of the HIV protease mutants.

"-mt MIEC_PR-MTN_combined_scaled.out" is the MIEC profile after the pre-process step, generated from the last step.

"-r pred_result" is the prediction result. The number in the fifth column: 1-resistance, 0-non-resistance.

Running result:

We run the above command on a server with Xeon E5-2430 and 96G RAM. The total running time is less than 1 minute for the full version.

Part IV. MIEC-SVM training guide

The following information is needed to train an MIEC-SVM model:

1. **Complex template structures of the protein domains**. "-s" option in 1_model_build.pl.
2. **Domain information, receptor and ligand residue number range**. "-d" option in 1_model_build.pl.
3. **Interaction information, a list of binders and non-binders.** "–b" option in 5_training_preprocess.pl.
4. **Multiple sequence alignment of the protein domains.** "-ms" option in 3_genMIEC_MMPBSAPY.pl.
5. **Receptor-ligand interaction residue pairs.** "-pl" option in 3_genMIEC_MMPBSAPY.pl.
6. **Peptide sequences for the protein-peptide interactions.** "-l" option in 1_model_build.pl.
7. **Partial charge and force field modification file for any non-standard amino acid residue or ligand**. Files in directory "dat/parms/" and information in file "dat/mol_info/mol_list" and "dat/mol_info/mol_atom_list".

Ideally, a reliable MIEC-SVM model needs a large amount of interaction information, an even sampling of domain and peptide/ligand sequence/structure space, and a realistic ratio of binders to non-binders.

Empirically, we require that the total number of interactions should be at least two times of the number of features. We never train an MIEC-SVM model with less than 400 interactions. The ratio between binders and non-binders depends on the domains of interest. Our experience shows that the binder/non-binder ratio can range from 1:20 (SH3 domain with Class II peptides) to 1:2 (HIV protease drug resistance).

An even sequence/structural distribution for both the domain and the peptide/ligand is critical to the success of MIEC-SVM model training. It is always bad practice to include a large group of proteins/peptides with high sequence similarity and a small group of proteins/peptides with low sequence similarity to the large group. Therefore, the generalization of the MIEC-SVM should be handled with extra care. The normal 3-fold or 5-fold cross validation cannot fully evaluate the generalization ability of the model due to the potential similarity between the training set and test set in each round of the cross validation. We recommend that users to perform a "leave one group out" test as a critical assessment of the model's generalization ability, such as the LODO (leave-one-domain-out) test in the domain-peptide interaction studies or the LOLO (leave-one-ligand-out) test in the drug resistance studies.

Part V. Descriptions of the pre-trained MIEC-SVM models provided

All pre-trained model is located in "dat/svm_models" with necessary support file such as the pair list, feature subset list, and scaling information file. Users can use these models to make predictions on the interactions of their interest. For version 1.1 of the pipeline, we provide the MIEC-SVM model for SH3 (class II peptide), PDZ, Chromo, and HIV-PR drug resistance models.

1. **SH3 domain Class II peptide MIEC-SVM model**

   The MIEC-SVM model for SH3 class II peptide interactions follows most of the protocols from our published SH3 paper (Hou, et al., 2012). The interaction data is taken from the paper. In total, 16 SH3 domains were modeled against class II peptides, resulting in 599 binding interactions (binders) and 11980 non-binding interactions (non-binders). The only differences from our published protocol are that we used "scwrl4" instead of "scap" for the virtual mutagenesis and we used AMBER14 instead of AMBER10. The LODO (**L**eave **O**ne **D**omain **O**ut) test result is shown below:

	SEN	SPC	ACC+	ACC-	MCC	AUC_{ROC}
Amph_human	0.441	0.999	0.938	0.973	0.633	0.975
Asp2_human	0.308	0.988	0.571	0.966	0.399	0.913
Bbc1_yeast	0.826	0.933	0.38	0.991	0.53	0.969
Boi1_yeast	0.167	0.988	0.4	0.96	0.235	0.914
Boi2_yeast	0.316	0.963	0.3	0.966	0.272	0.908
Crk_human	0.523	0.97	0.469	0.976	0.469	0.963
Grb2_mouse	0.431	0.961	0.354	0.971	0.357	0.859
Lsb1_yeast	0.692	0.938	0.36	0.984	0.466	0.942
Lsb3_yeast	0.586	0.982	0.625	0.979	0.586	0.955
Lsb4_yeast	0.468	0.987	0.643	0.974	0.529	0.943
Pig1_human	0.909	0.959	0.526	0.995	0.673	0.973
Pin3_yeast	1	1	1	1	1	1
Rvs167_yeast	0.658	0.951	0.403	0.982	0.485	0.952
Sh3g2_human	0.074	0.989	0.25	0.955	0.114	0.646
Src8_mouse	0.228	0.986	0.448	0.962	0.296	0.909
Src_human	0.818	0.932	0.375	0.99	0.523	0.967
Average	**0.528**	**0.970**	**0.503**	**0.977**	**0.473**	**0.924**

   We provide the MIEC-SVM model for SH3 and class II peptides. The class II peptides have the motif "PXXPX[R/K]". The peptide typically has ten amino acids, such as "GPRRPRRSLP". The peptide number starts at -9 for the N-terminus and ends at 0 for the C-terminus. **Therefore, when running "3_genMIEC_MMPBSAPY.pl", option "-pn 10" should be specified.**

2. **PDZ domain MIEC-SVM model**

   The model provided here follows the protocol of our published PDZ paper (Li, et al., 2011). The interaction data was taken from Stiffler et al's microarray data. 11

mouse PDZ proteins with available complex structures were used to train the model. There are 86 binders (binding interactions) and 2301 non-binders (non-binding interactions). All complex structures are processed using the programs in the pipeline. MIEC profiles are re-generated using the AMBER14 instead of AMBER9 described in the paper. Since the number of binders is small and the ratio between binders and non-binders is high (1:27), we resample the training set using all binders and six times the number of non-binders. The LODO (Leave One Domain Out) test shows satisfactory prediction performance.

	SEN	SPC	ACC+	ACC-	MCC	AUC_{ROC}
1NF3	1	0.917	0.739	1	0.818	0.988
1V6B	0.5	0.947	0.749	0.919	0.539	0.897
1VJ6	0.446	0.788	0.263	0.895	0.192	0.727
1WG6	0.742	0.746	0.364	0.947	0.386	0.846
1WHD	0.488	0.748	0.245	0.898	0.183	0.61
2CSJ	0.313	0.855	0.252	0.883	0.15	0.78
2D90	0.916	0.775	0.414	0.983	0.523	0.915
2EDZ	0.97	0.759	0.409	0.994	0.542	0.947
2I04	0.448	0.912	0.529	0.909	0.392	0.678
2PDZ	0.233	0.961	0.581	0.883	0.293	0.871
3DIW	0.891	0.891	0.597	0.981	0.669	0.945
Average	**0.632**	**0.845**	**0.467**	**0.936**	**0.426**	**0.837**

We provide the three MIEC-SVM models from the best three resampling data subset. **Please specify "–pn 5" in "3_genMIEC_MMPBSAPY.pl" when generating the MIEC profile for the model.**

3. HIV-PR drug resistance MIEC-SVM model

For the HIV protease drug resistance MIEC-SVM model, we provide the models using the same protocol as provided in our published paper (Ding, et al., 2013). In total, there are seven models, each of which is trained using the drug resistance data from one of the seven inhibitor ligands. The voting result of the seven models is given in the final prediction result. For the details of prediction performance, please refer to the paper (Ding, et al., 2013). For details of constructing the MIEC profile for HIV-PR systems, please refer to section 3 of the tutorial.

4. Chromo domain MIEC-SVM model

In our latest study, we identify the interactions between 22 human chromo domains and 457 methylated lysine containing peptides. We then use the interaction data of 12 chromo domains, which are able to obtain structural models, to train an MIEC-SVM model using the pipeline. For chromo domain interaction data, there are 1305 binders and 4179 non-binders. All decompositions are

processed by AMBER14. We use all the data to perform the LODO test and then train the MIEC-SVM model for chromo domain. Below is the LODO result:

	SEN	SPC	ACC+	ACC-	MCC	AUC_{ROC}
CBX1	0.779	0.915	0.616	0.96	0.632	0.937
CBX2	0.892	0.522	0.265	0.962	0.306	0.798
CBX3	0.75	0.952	0.6	0.975	0.636	0.944
CBX5	0.736	0.779	0.639	0.847	0.5	0.832
CBX6	0.418	0.876	0.583	0.784	0.329	0.746
CBX7	0.421	0.975	0.706	0.922	0.499	0.894
CBX8	0.301	0.899	0.733	0.583	0.252	0.722
CDYL1	0.721	0.769	0.353	0.94	0.378	0.829
CDYL2	0.609	0.899	0.796	0.781	0.541	0.837
MPP8	0.316	0.989	0.882	0.846	0.471	0.861
SUV91	0.183	0.98	0.75	0.781	0.293	0.791
SUV92	0.381	0.956	0.698	0.851	0.43	0.829
Average	**0.542**	**0.876**	**0.635**	**0.853**	**0.439**	**0.835**

We use the MIEC-SVM model to predict the interactions between the SUV92 mutant (V3L/K40L/L43F) and 457 peptides. The AUC of ROC is 0.842 as shown in the manuscript. **Please specify "–pn 9" in "3_genMIEC_MMPBSAPY.pl" when generating the MIEC profile for the model.**

Part VI. Supported non-standard residues/ligands

The pipeline now supports the following non-standard residues by default.

1. **Phosphorylated serine**: [pSer] in peptide sequence; PDB residue name, SPH as normal residue, SPN as N-terminal residue, and SPC as C-terminal residue.
2. **Phosphorylated threonine**: [pThr] in peptide sequence; PDB residue name, TPH as normal residue, TPN as N-terminal residue, and TPC as C-terminal residue.
3. **Mono-methylated lysine**: [K_Me] in peptide sequence; PDB residue name, M1L as normal residue, M1N as N-terminal residue, and M1C as C-terminal residue.
4. **Di-methylated lysine**: [K_Me2] in peptide sequence; PDB residue name, only normal residue is provided as M2L.
5. **Tri-methylated lysine**: [K_Me3] in peptide sequence; PDB residue name, only normal and N-terminus residue is provided as M3L and M3N respectively.
6. **Symmetric di-methylated arginine**: [RMe2s] in peptide sequence; PDB residue name, only normal residue is provided as RMS.
7. **Asymmetric di-methylated arginine**: [RMe2a] in peptide sequence; PDB residue name, only normal and N-terminal residue is provided as RMA and RAN.

For customized non-standard residues, users need to get the partial charge and force field modifications to ff03.r1 and gaff. Then users need to include the residue information in the “dat/mol_info/mol_list” and “dat/mol_info/mol_atom_list” files and put the partial charge and force field modification file in the directory “dat/parm”.

Part VII. MIEC-SVM overview

MIEC-SVM is a computational method to predict protein-peptide and protein-ligand interactions. The MIEC (Molecular Interaction Energy Component) profile is used to characterize the interactions through the energetic pattern of residue pairs between the protein and the peptide/ligand. SVM (support vector machine) is a machine learning method that can make a binary classification on the protein-peptide/protein-ligand interactions, based on the MIEC profiles.

The MIEC-SVM pipeline provides an integrated and user-friendly workflow for the construction and application of the MIEC-SVM model. It consists of three sections: model building, MIEC construction, and model training/prediction.

The model building section performs virtual mutagenesis on the user provided template structures to build structures for the following application scenarios: proteins and selected peptides, proteins with mutations and selected peptides, and proteins with mutations and a ligand. Currently, it cannot build a structure between a protein and a ligand from scratch since such a problem can be solved by docking methods, such as Dock, Autodock, and Glide. Energy minimization and decomposition are processed by the AMBER package. The AMBER ff03 force field provides force field parameters and partial charges for all standard residues and the AMBER gaff force field defines atom types and force field parameters for small molecules (ligands). The pipeline provides a lightweight database MolDB so that the user can store the force field information for non-standard residues and ligands of interest, thus enabling the pipeline to pass the information to the tleap program (AMBER package) for the correct topology build. To include non-standard residues and ligands in the pipeline, users need to provide mol2 files for the partial charge and atom type definitions for each atom in the molecule. Furthermore, a force field modification file must be supplied, including all the supplementary force field parameters to the gaff force field for the molecules.

In the MIEC construction section, there are two major functions: construction of the MIEC by providing a component list and a residue pair list and secondly, the combination of any two MIEC profiles into a single MIEC. Normally, the pipeline reads in the energy decomposition results and integrates them into the MIEC profile. The standard MIEC uses MM/GBSA and the profile for each protein complex is constructed as such (P_1-VDW, P_1-ELE, P_1-GB, P_1-SA, P_2-VDW, …, P_n-GB, P_n-SA, Q_1-VDW,Q_1-ELE, …, Q_m-GB, Q_m-SA), where P_i is the i^{th} residue pair between receptor and ligand, and Q_j is the j^{th} residue pair between the adjacent ligand residues. VDW, ELE, GB, and SA are the four energy components of MM/GBSA: van de Waals (VDW), electrostatics

(ELE), generalized Born (GB), and surface area (SA). The pipeline provides flexibility for users to construct their own MIEC profile, such as a combination of any energy components, a combination of any residue pairs, or both. Users can also build a customized MIEC using an energy function other than MM/GBSA with the pipeline.

In the MIEC-SVM training and prediction section, separate pipeline branches are implemented for training and prediction respectively. Each branch contains two steps: the pre-processing step and the MIEC-SVM training/prediction step. In the pre-processing step, averaged MIEC profiles are generated from multi-template MIECs and the MIEC profiles are scaled for SVM prediction. Three scaling methods are implemented in the pipeline: receptor-based scaling, reference-based scaling, and range-based scaling. The receptor-based method is normally used for protein-peptide specificity prediction. It linearly scales each MIEC component into -1 and 1 using the MIEC profiles from the same receptor/protein. The reference-based scaling method is used for drug resistance prediction, where the MIEC profiles are constructed from complexes of mutants and the ligand. The method first subtracts the wild type MIEC profile from each mutant MIEC profile. Then, each MIEC component is scaled into -1 and 1, similar to that in the reference-based method. The range-based scaling method applies the previously generated scaling range file to the input MIEC profile.

For the MIEC-SVM model training section, users need to provide a file of lambda values for the LASSO logistic regression based feature selection process. The lambda value is the scaling factor for the regularization term in the target function of LASSO. Empirically, the larger the lambda is, the fewer features will be selected and vice versa. The pipeline includes a list of 100 lambda values, which are the default values from the “glmnet” package. For each lambda value, an MIEC-SVM model is trained and a cross validation is performed to evaluate the prediction performance of the model.

Part VIII. Key Concepts

Pipeline parameter file

The pipeline parameter file is a user provided file that defines the values of the pre-defined keywords that provide information as run-time variables to the pipeline. There is a further parameter file located as “conf/default.conf” that will be loaded for miscellaneous information every time the pipeline script is run. Please note that duplicated keyword entries could cause problems for the pipeline.

Complex information list

This file stores information for complexes processed in the pipeline. Each line describes the information of a complex with seven values: serial number, complex name, receptor residue range, ligand residue range, residues with fixed side chain, residues with frame and domain names. The complex name uses the following format: [protein name]-[mutant name]_[conformation name]_[peptide/ligand name], where both “protein name” and “peptide/ligand name” are required. Examples of complex names can be found in the example cases provided with the pipeline.

Conformation list

The conformation list provides template pdb files for model building. Each line describes the location of the template file and the conformation name. It supports both single conformation mode and multiple conformation mode. In single conformation mode, the conformation name is provided as “NA”. In multiple conformation mode, the conformation name and the corresponding template pdb file location are provided in each line. The MIEC profiles from the multiple conformations will be averaged to generate a single MIEC profile for respective interaction. For the template pdb file, the filename should follow the complex name format described in the complex information list section. Another requirement of the template pdb file is that the residue numbers must be consecutive and start from 1. The receptor part is labeled with the smaller numbers and the ligand part is labeled with the larger numbers. Users can also refer to the example cases for the residue numbering of the template pdb file.

MolDB

MolDB stores information for all non-standard amino acids or ligands. It contains two parts: the molecule list and atom list. The molecule list contains information for each molecule. It has 8 columns: residue name in PDB, residue name in sequence, standard amino acid name, connectivity of the residue, N-terminus atom, C-terminus atom, mol2 file name and frcmod file name. The atom list contains information of atoms in the non-standard amino acids and is used in the side chain conformation building process. There are three columns: residue name in PDB, atom name of non-standard amino acid, and

corresponding atom name of standard amino acid. There are three entries in the parameter file related to MolDB: “dir_parm” is the location to store mol2 and frcmod files; “mol_list” is the molecule list file; “mol_atom_list” is the atom list.

Parms for AMBER

AMBER needs partial charges, atom types and force field parameters for non- standard residues and ligands for inclusion in the topology file. Tripos Mol2 file is used here for partial charge and atom type specification. The force field modification file provides force field parameters for bonds, bond-angles, dihedral, and non-bonded interactions. All mol2 files and frcmod files need to be stored in the location specified by “dir_parm” in the parameter file.

Ligand information list

The ligand information file stores information relating to the peptide sequence in the protein-peptide interaction case. Non-standard residues can be provided using square bracket annotation, such as “R[K_me3]S” which corresponds to ARG, tri-methylated lysine, and SER”. Each line has three items: ligand serial number, peptide sequence and annotation. There is one entry in the parameter file for the ligand information list: “ligand_list” is the file for the ligand information list.

Ligand subset list

The ligand subset list contains the subset of ligands in the ligand information list that are used in model building, the format of which is a plain text file of ligand serial numbers.

Mutation information list

This list contains mutation site and residue information for the protein mutants. The first column is the mutant name. The rest of the columns contain mutation information indicated by the three-letter residue name to be mutated into and the residue position.

MSA for MIEC

In the domain-peptide case, proteins of one protein family need to be aligned to match the residue number in different proteins. MIEC construction will require the MSA residue number to define the receptor-ligand residue pair.

MIEC component list

The component list contains the energy terms used for MIEC-SVM construction. For MM/GBSA energy terms, the following key words are accepted: TVDW, TELE, TGB, TGBSUR, BVDW, BELE, BGB, BGBSUR, SVDW, SELE, SGB and SGBSUR. VDW, ELE, GB, and GBSUR abbreviations for van de Waals, electrostatics, generalized Born,

and surface area energy terms in the MM/GBSA. The prefix “T”, “B”, and “S” abbreviate for total energy, backbone energy, and side chain energy. Backbone and side chain energy terms always come in pairs in MM/GBSA decomposition. For example, the BVDW between residues A and B has two values: the van de Waals energy value between backbone atoms of A and all atoms of B and the value between backbone atoms of B and all atoms of A. Using the backbone and side chain split term gives a better representation of the interactions between the two residues.

Users can use the component list to combine any number terms into one MIEC component. For example, if the user specifies “TELE TGB” as one line in the component list, the polar contribution (ELE+GB) will be used as one MIEC component.

For non-MM/GBSA energy terms, the column number is used instead of the MM/GBSA energy terms. The non-MM/GBSA energy file has the following format: the first line is the name of each column; the first and second columns are residue numbers, the rest of columns are energy terms, noted as 1, 2 … in the component list.

MIEC residue pair list

The residue pair list contains the receptor-ligand residue pairs that are used to construct the MIEC profile. Receptor uses the residue number in the multiple sequence alignment starting from 1 at the N-terminus. Ligand uses non-positive residue numbers starting from 0 at the C-terminus. For example, a ligand peptide of 5 amino acids will have residue numbers from -4 (N-terminus) to 0 (C-terminus).

In a similar fashion to the component list, users can combine any residue pairs into one general residue pair in the MIEC profile.

MIEC subset list

This file is generated after the feature selection process where only part of the MIEC component in the full profile is kept. The file is a single column of numbers indicating the MIEC components contained in the subset. The MIEC component number starts from 1.

Scaling method for SVM

There are three scaling methods provided by the pipeline: receptor-based scaling, reference-based scaling, and range-based scaling. The receptor-based method is normally used for protein-peptide specificity prediction. It linearly scales each MIEC component into -1 and 1 using the MIEC profiles from the same receptor/protein. The reference-based scaling method is used for drug resistance prediction, where the MIEC profiles are constructed from complexes of mutants and the ligand. The method first subtracts the wild type MIEC profile from each mutant MIEC profile. Then, each MIEC component is

scaled into -1 and 1 similar to that in the reference-based method. The range-based scaling method applies the previously generated scaling range file to the input MIEC profile.

Range file

The range file contains the scale information used by previous MIEC scaling. It is needed for range-based MIEC scaling.

SVM model

The SVM model file is the previously trained MIEC-SVM model for MIEC-SVM prediction. For now, the MIEC-SVM models of chromo domain and HIV protease drug resistance are provided.

LAMBDA file

The LAMBDA file provides lambda values that will be used by LASSO logistics regression. For each LAMBDA value, a feature selected MIEC-SVM model is generated. Users can select the best MIEC-SVM model when multiple lambda values are provided.

Prediction result file

This six-column file indicates the MIEC-SVM prediction result. The first and second column is reserved for future use. The third column is the complex name. The fourth column is the experimentally determined binding type if applicable (0-non-binder, 1-binder). For complexes without binding information, a default binding type of 0 is used. The fifth column is the predicted binding type (0-non-binder, 1-binder). The sixth column is the decision value of SVM prediction.

References

Ding, B., Li, N. and Wang, W. (2013) Characterizing binding of small molecules. II. Evaluating the potency of small molecules to combat resistance based on docking structures, *J Chem Inf Model*, **53**, 1213-1222.
He, W*., et al.* (2015) Deciphering and engineering chromodomain-methyllysine peptide recognition, *In submission*.
Hou, T*., et al.* (2012) Characterization of domain-peptide interaction interface: prediction of SH3 domain-mediated protein-protein interaction network in yeast by generic structure-based models, *J Proteome Res*, **11**, 2982-2995.
Li, N*., et al.* (2011) Characterization of PDZ domain-peptide interaction interface based on energetic patterns, *Proteins*, **79**, 3208-3220.