CMF – R Package Implementing the Continuous Molecular Fields Approach

1. Inrtoduction

The package CMF contains a set of R functions that implement the Continuous Molecular Fields (CMF) approach to building 3D-QSAR models. The reference version of the R environment for statistical computing and graphics is 3.0.1.

2. Installation and Setup of the Required Software

The latest version of the R environment can be downloaded and installed from the homepage of the R Project for Statistical Computing http://www.r-project.org/. In order to perform visualization of molecules, molecular fields and fields of regression coefficients, it is necessary to install two additional packages: rgl and misc3d. They can be installed from the CRAN homepage http://cran.r-project.org/. In order to use quantum chemical molecular fields, it is necessary to install the MOPAC12 program http://openmopac.net/MOPAC2012.html. In order to run the software implementing the CMF approach to building 3D-QSAR models, it is necessary to keep all files, including all R scripts and all data files, in the same folder (directory), which should be specified as the current working directory.

4. The Basic Workflow for Building 3D-QSAR Models Using the CMF Approach

Two major modes of building and analyzing the performance of 3D-QSAR models are provided. The first one is based on a single splitting between a training set and an external (independent) test set. *The model built on the training set is further applied to predict activity values of compounds contained in the test set.* This mode requires the use of the following four input files:

- a file (in the mol2 format) containing 3D structures of the molecules belonging to the training set (its default name is ligands-train.mol2);
- a file (in the delimited txt format) containing experimental (measured) activity (property) values of the corresponding chemical compounds belonging to the training set (its default name is activity-train.txt);
- a file (in the mol2 format) containing 3D structures of the molecules belonging to the test set (its default name is ligands-pred.mol2);
- a file (in the delimited txt format) containing experimental (measured) activity (property) values of the corresponding chemical compounds belonging to the test set (its default name is activity-pred.txt).

The second mode is based on the procedure of external n-fold cross-validation. In this case the whole set of compounds is split into the training and test sets n times, so each compound appears in a test set exactly once. *The 3D-QSAR model built on a training set is applied to the corresponding test set, and all prediction results are accumulated.* This mode requires the use of the following two input files:

- a file (in the mol2 format) containing 3D structures of the molecules belonging to the whole set (its default name is ligands-all.mol2);
- a file (in the delimited txt format) containing experimental (measured) activity (property) values of the corresponding chemical compounds belonging to the whole set (its default name is activity-all.txt)

4.1. Aligning molecules

Molecules in a dataset can be aligned using two different approaches. If the set of compounds under study is congeneric, then it can be aligned by least-square fitting (algorithm arun) to a common template substructure, which should be contained in all molecules belonging to this set. Otherwise, alignment can be performed using the seal algorithm. If the first approach is chosen, it is first necessary to obtain the template substructure to be used for performing alignment. For example, it can be extracted from some molecule by specifying a list of serial numbers of atoms. This can be accomplished using the script cmf-do-make-template.R, in which the values of the following parameters can be specified:

- mdb-fname – file name containing the structure from which the template substructure is to be extracted;
- imol – the serial number of the molecule in the file mdb-fname, from which the template substructure is to be extracted;
- atom_list – the list of serial numbers of atoms used for extracting the template substructure;
- template_fname – file name for the template substructure.

The template substructure can further be used for performing molecular alignment. This can be carried out using the following script **cmf-do-make-template**:

```
# File name of molecular databased to be aligned
mdb_fname <- "ligands-train.mol2"

# Molecules from mdb_fname to be aligned
iimol <- c(1:72)

# File name of template
templ_fname <- "ligands-template.mol2"

# File name for aligned database
mdb_a_fname <- "ligands-aligned.mol2"

# Algorithm (arun/seal)
algorithm <- "arun"

mdb <- read_mol2(mdb_fname)
mdb_tmp <- read_mol2(templ_fname)
templ <- mdb_tmp[[1]]
if (algorithm == "arun") {
  mdb_a <- align_mdb_template(mdb, templ)
#  mdb_a <- align_mdb_template(mdb, templ, iimol)
} else if (algorithm == "seal") {
  mdb_a <- align_mdb_seal(mdb, templ)
} else {
  cat("Unknown algorithm\n")
}
write_mol2(mdb_a, mdb_a_fname)
```

in which the following parameters can be specified:

- mdb_fname – file name of the molecular database to be aligned;
- iimol – List of molecules from mdb_fname to be aligned (if this parameter is dropped, the whole molecular database is to be aligned);
- templ_fname – file name of the template substructure;
- mdb_a_fname – file name of the produced aligned database;
- algorithm – alignment algorithm (in this case, arun).

4.2. Computing kernels for the training set

The first step in building 3D-QSAR models using the CMF approach is to compute kernels for all pairs of compounds from the training set and to write them to a file. The kernels can be computed for five types of molecular fields:

- q - electrostatic molecular field;
- vdw - steric molecular field;
- logp - hydrophobicity field;
- abra - hydrogen-bond acidity field;
- abrb - hydrogen-bond basicity field.

All kernels are computed for the following fixed set of attenuation parameter: 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1.0, 2.0, 3.0, 5.0, 10.0. To carry out this kernel computation procedure, one can run the computational R script **cmf-do-comp-kernels-train**:

```
# Training set file name
train_fname <- "ligands-train.mol2"

# The name of the file with kernels for training
kernels_train_fname <- "ligands-kernels-train.RData"

# Molecular fields
mfields <- c("q","vdw","logp","abra","abrb")

# Verbose computation of kernels (TRUE/FALSE, 1/0)
print_comp_kernels <- TRUE

comp_kernels_train(
  train_fname = train_fname,
  kernels_train_fname = kernels_train_fname,
  mfields = mfields,
  print_comp_kernels = print_comp_kernels
)
```

The values of the following parameters may be specified by editing their default values in the script:

- train_fname - the name of the mol2 file containing the structures from the training set (default: “ligands-train.mol2”);
- kernels_train_fname - the name of file with computed kernels (default: “ligands-kernels-train.RData”);
- mfields – the list of molecular fields for which kernels should be computed (default: c(“q”,“vdw”,“logp”,“abra”,“abrb”));
- print_comp_kernels – whether to verbose computation of kernels (TRUE/FALSE or 1/0), (default: TRUE)

4.3. Building 3D-QSAR regression model using the training set

The second step is to build a 3D-QSAR model and write it to a file. This is performed by running the script **cmf-do-krr-train**:

```
# Activity file name
act_train_fname <- "activity-train.txt"

# Activity column number
act_colnum <- 1

# Separator
sep <- ","

# Kernels file name
kernels_train_fname <- "ligands-kernels-train.RData"
```

```
# Model file name
model_fname <- "ligands-model.RData"

# Types of molecular fields
mfields <- c("q","vdw","logp","abra","abrb")

# Whether to perform grid search for attenuation factor alpha (TRUE/FALSE, 1/0).
alpha_grid_search <- TRUE

# Whether to perform grid search for Tikhonov regularization coefficient gamma (TRUE/FALSE, 1/0).
gamma_grid_search <- FALSE

# Whether to form tconic kernel combination
conic_kernel_combination <- FALSE

# Whether to optimize h (TRUE/FALSE, 1/0)
optimize_h <- FALSE

# Whether to set b=0 (TRUE/FALSE, 1/0)
set_b_0 <- FALSE

# Print intermediate results in the course of optimiization (TRUE/FALSE, 1/0)
print_interm_icv <- TRUE

# Produce intermediate scatter plots in the course of optimization  (TRUE/FALSE, 1/0)
plot_interm_icv <- TRUE

# Print final results on interna; cross-validation (TRUE/FALSE, 1/0)
print_final_icv <- TRUE

# Produce final scatter plot for internal cross-valudation (TRUE/FALSE, 1/0)
plot_final_icv <- TRUE

cmf_krr_train(
  act_train_fname = act_train_fname,
  act_colnum = act_colnum,
  sep = sep,
  kernels_train_fname = kernels_train_fname,
  model_fname = model_fname,
  mfields = mfields,
  alpha_grid_search = alpha_grid_search,
  gamma_grid_search = gamma_grid_search,
  conic_kernel_combination = conic_kernel_combination,
  optimize_h = optimize_h,
  set_b_0 = set_b_0,
  print_interm_icv = print_interm_icv,
  plot_interm_icv = plot_interm_icv,
  print_final_icv = print_final_icv,
  plot_final_icv = plot_final_icv
)
```

which implements the kernel ridge regression (KRR) algorithm. The values of the following parameters can be specified by changing their default values in this script:

- act_fname - the name of the text file containing the values of activity for all compounds from the training set (default: "activity-train.txt");
- act_colnum - activity column number in the file act_fname (default: 2);
- sep - separator of text fields in the file act_fname containing activity values (default: ",");
- kernels_fname - the name of the file containing kernels computed for the training set (default: "ligands-kernels-train.RData");
- model_fname - the name of the file to which the 3D-QSAR model is written (default: "ligands-model.RData");
- alpha_grid_search- whether to perform the grid search for hyper-parameter αf. (TRUE/FALSE) or (1/0) [This option specifies the mode of finding the optimized values of αf (attenuation factor for molecular fields) values. In the first mode, when alpha_grid_search=TRUE, all values of gamma taken from the grid (the fixed set of 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1.0, 2.0, 3.0, 5.0, 10.0) are considered, the value providing the highest Q2 is chosen. In the second mode (alpha_grid_search=FALSE) γ is included in the list of the hyper-parameters subjected to optimization using the Nelder-Mead algorithm. In this case linear interpolation of kernel values is performed.], (default: TRUE);
- gamma_grid_search - whether to perform the grid search for hyper-parameter gamma (TRUE/FALSE) or (1/0) [This option specifies the mode of finding the optimized values of γ (regularization parameter in KRR) values. In the first mode, when gamma_grid_search=TRUE, all values of gamma taken from the grid (the fixed set of 0.01, 0.05, 0.1, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0, 2.5, 3.0, 3.5, 4.0, 5.0, 10.0, 15.0, 20.0) are considered, the value providing the highest Q2 is chosen. In the second mode (gamma_grid_search=FALSE) γ is included in the list of the hyper-parameters subjected to optimization using the Nelder-Mead algorithm.], (default: FALSE);
- conic_kernel_combination - whether to form conic kernel combination (with positive mixing coefficients h) (TRUE/FALSE) or (1/0), (default: FALSE);
- optimize_h - whether to optimize the mixing coefficients h (TRUE/FALSE) or (1/0), (default: FALSE);
- mfields – the list of molecular fields for which kernels should be computed (default: c("q","vdw","logp","abra","abrb"));
- set_b_0 – whether to set the offset b=0 (this defines the difference between KRR1 and KRR2 modifications of the KRR method) (TRUE/FALSE) or (1/0), (default: FALSE);
- print_interm_icv – whether to print intermediate results in the course of optimization (TRUE/FALSE, 1/0), (default: TRUE);
- plot_interm_icv – whether to produce intermediate scatter plots in the course of optimization (TRUE/FALSE, 1/0), (default: TRUE);
- print_final_icv – whether to print final results on internal cross-validation (TRUE/FALSE, 1/0), (default: TRUE);
- plot_final_icv – whether to produce final scatter plot for internal cross-validation (TRUE/FALSE, 1/0), (default: TRUE);

4.4. The Use of 3D-QSAR Model for Predicting Activities of Test Set Compounds

Predicting activities of compounds belonging to the external test set is carried out in two steps. In the first step, one should compute kernels comparing molecular fields of all compounds belonging to the test set with molecular fields of all compounds belonging to the training set. To do that, it is necessary to run script **cmf-do-comp-kernels-pred** :

```
# Training set file name
train_fname <- "ligands-train.mol2"

# Prediction set file name
pred_fname <- "ligands-pred.mol2"

# Computed kernels file name
kernels_pred_fname <- "ligands-kernels-pred.RData"
```

```
# Molecular fields
mfields <- c("q", "vdw", "logp", "abra", "abrb")

# Verbose computation of kernels (TRUE/FALSE, 1/0)
print_comp_kernels <- TRUE

comp_kernels_pred(
  train_fname = train_fname,
  pred_fname = pred_fname,
  kernels_pred_fname = kernels_pred_fname,
  mfields = mfields,
  print_comp_kernels = print_comp_kernels
)
```

The values of the following parameters may be modified in the script: - train_fname - the name of the mol2 file containing the structures from the training set (default: "ligands-train.mol2"); - pred_fname - the name of the mol2 file containing the structures from the test set (default: "ligands-pred.mol2"); - kernels_pred_fname - file name for storing computed kernels (default: "ligands-kernels-pred.mol2"); - mfields – the list of molecular fields for which kernels should be computed (default: c("q","vdw","logp","abra","abrb"));

- print_comp_kernels – whether to verbose computation of kernels (TRUE/FALSE or 1/0), (default: TRUE).

The second step is to make predictions by running the script **cmf-do-krr-pred** :

```
# Prediction for KRR models with continuous molecular fields


# Model file name
model_fname <- "ligands-model.RData"

# Kernels for training file name
kernels_train_fname <- "ligands-kernels-train.RData"

# Kernels for prediction file name
kernels_pred_fname <- "ligands-kernels-pred.RData"

# Column name or activity (if specified)
act_colnum <- 1

# Separator
sep <- ","

# Activity for prediction set (if specified)
act_pred_fname <- "activity-pred.txt"

# File with predictions
pred_fname <- "ligands-pred.RData"

# Print statistical parameters for prediction (TRUE/FALSE, 1/0)
print_pred <- TRUE

# Produce scatter plot prediction on external database (TRUE/FALSE, 1/0)
plot_pred <- TRUE
```

```
res <- cmf_krr_pred(
  model_fname = model_fname,
  kernels_train_fname = kernels_train_fname,
  kernels_pred_fname = kernels_pred_fname,
  act_colnum = act_colnum,
  sep = sep,
  act_pred_fname = act_pred_fname,
  pred_fname = pred_fname,
  print_pred = print_pred,
  plot_pred = plot_pred
)
```

The results of prediction is written to the file referenced by the variable pred_fname. The values of the following parameters may be modified in the script:

- model_fname - the name of the file containing 3D-QSAR model (default: “ligands-model.RData”);
- kernels_pred_fname - file name of the file containing kernels that compare molecular fields of molecules belonging to the training and the test sets (computed at the first step) (default: “ligands-kernels-pred.RData”);
- act_colnum - the number of the column with activity values in the file act_pred_fname (If act_colnum=0 then it is assumed that the experimental values of activities are unknown) (default: 2);
- sep - separator of text fields in the file act_pred_fname containing activity values (if act_colnum > 0) (default “,”);
- act_pred_fname - the name of the text file containing the values of activity for all compounds from the test set (if act_colnum > 0) (default: “activity-pred.txt”);
- pred_fname - file with predictions (default: “ligands-pred.RData”); print_pred – whether to print statistical parameters for prediction (TRUE/FALSE, 1/0), (default: FALSE);
- plot_pred – whether to produce scatter plot prediction on external database (TRUE/FALSE, 1/0), (default: FALSE).

4.5. Building 3D-QSAR models using external n-fold cross-validation

Before building 3D-QSAR models using external n-fold cross-validation, it is necessary to compute kernels for the whole dataset. This can be done using the script **cmf-do-comp-kernels-all**:

```
# The whole set file name
all_fname <- "ligands-all.mol2"

# The name of the files containing kernels for all molecules
kernels_all_fname <- "ligands-kernels-all.RData"

# Molecular fields
mfields <- c("q","vdw","logp","abra","abrb")

# Verbose computation of kernels (TRUE/FALSE, 1/0)
print_comp_kernels <- TRUE

comp_kernels_all(
  all_fname = all_fname,
  kernels_all_fname = kernels_all_fname,
  mfields = mfields,
  print_comp_kernels = print_comp_kernels
)
```

The following parameters in it can be specified:

- all_fname - the name of the mol2 file containing the whole set of compounds (default: “ligands-all.mol2”);
- kernels_all_fname - the name of file with computed kernels (default: “ligands-kernels-all.RData”);
- mfields – the list of molecular fields for which kernels should be computed (default: c(“q”,“vdw”,“logp”,“abra”,“abrb”));
- print_comp_kernels – whether to verbose computation of kernels (TRUE/FALSE or 1/0), (default: TRUE) After that, in order to perform external n-fold cross-validation, one can launch the script **cmf-do-krr-ecv**:

```
# The number of folds in n-folds cross-validation
nfolds <- 5

# Activity file name
act_all_fname <- "activity-all.txt"

# Activity column number
act_colnum <- 1

# Separator
sep <- ","

# Kernels file name
kernels_all_fname <- "ligands-kernels-all.RData"

# Types of molecular fields
mfields <- c("q","vdw","logp","abra","abrb")

# The name of the file with external cross-validation results
ecv_fname <- "ligands-ecv.RData"

# Whether to perform grid search for attenuation factor alpha (TRUE/FALSE, 1/0).
alpha_grid_search <- TRUE

# Whether to perform grid search for gamma (TRUE/FALSE, 1/0).
gamma_grid_search <- FALSE

# Whether to form tconic kernel combination
# (with positive mixing coefficients) (TRUE/FALSE, 1/0)
# Conic combination preserves the positive definiteness of kernel matrices
# but leads to models with worse predictive performance assessed unsing cross-validation
conic_kernel_combination <- FALSE

# Whether to optimize h (TRUE/FALSE)
optimize_h <- FALSE

# Whether to set b=0 (TRUE/FALSE, 1/0)
set_b_0 <- FALSE

# Print statistical parameters of the external n-folds cross-validation procedure (TRUE/FALSE, 1/0)
print_ecv <- TRUE

# Produce scatter plot for external coss-validation (TRUE/FALSE, 1/0)
plot_ecv <- TRUE

# Print intermediate results in the course of optimiization (TRUE/FALSE, 1/0)
print_interm_icv <- TRUE
```

```
# Produce intermediate scatter plots in the course of optimization  (TRUE/FALSE, 1/0)
plot_interm_icv <- TRUE

# Print final results on interna; cross-validation (TRUE/FALSE, 1/0)
print_final_icv <- TRUE

# Produce final scatter plot for internal cross-valudation (TRUE/FALSE, 1/0)
plot_final_icv <- TRUE

# Print statistical parameters for prediction (TRUE/FALSE, 1/0)
print_pred <- TRUE

# Produce scatter plot prediction on external database (TRUE/FALSE, 1/0)
plot_pred <- TRUE

ecv <- cmf_krr_ecv(
  nfolds = nfolds,
  act_all_fname = act_all_fname,
  act_colnum = act_colnum,
  sep = sep,
  kernels_all_fname = kernels_all_fname,
  mfields = mfields,
  ecv_fname = ecv_fname,
  alpha_grid_search = alpha_grid_search,
  gamma_grid_search = gamma_grid_search,
  conic_kernel_combination = conic_kernel_combination,
  optimize_h = optimize_h,
  set_b_0 = set_b_0,
  print_ecv = print_ecv,
  plot_ecv = plot_ecv,
  print_interm_icv = print_interm_icv,
  plot_interm_icv = plot_interm_icv,
  print_final_icv = print_final_icv,
  plot_final_icv = plot_final_icv,
  print_pred = print_pred,
  plot_pred = plot_pred
)
```

in which the following parameters can be changed:

- nfolds – the number of folds in cross-validation (default: 5);
- act_all_fname - the name of the text file containing the values of activity for all compounds from the training set (default: “activity-all.txt”);
- act_colnum - activity column number in the file act_fname (default: 2);
- sep - separator of text fields in the file act_fname containing activity values (default: “,”);
- kernels_all_fname - the name of file with computed kernels (default: “ligands-kernels-all.RData”);
- mfields – the list of molecular fields for which kernels should be computed (default: c(“q”,“vdw”,“logp”,“abra”,“abrb”));
- ecv_fname - the name of the file with external cross-validation results (default: “ligands-ecv.RData”);
- alpha_grid_search- whether to perform the grid search for hyper-parameter αf. (TRUE/FALSE) or (1/0) [This option specifies the mode of finding the optimized values of αf (attenuation factor for

molecular fields) values. In the first mode, when alpha_grid_search=TRUE, all values of gamma taken from the grid (the fixed set of 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1.0, 2.0, 3.0, 5.0, 10.0) are considered, the value providing the highest Q2 is chosen. In the second mode (alpha_grid_search=FALSE) γ is included in the list of the hyper-parameters subjected to optimization using the Nelder-Mead algorithm. In this case linear interpolation of kernel values is performed.], (default: TRUE);

- gamma_grid_search - whether to perform the grid search for hyper-parameter gamma (TRUE/FALSE) or (1/0) [This option specifies the mode of finding the optimized values of γ (regularization parameter in KRR) values. In the first mode, when gamma_grid_search=TRUE, all values of gamma taken from the grid (the fixed set of 0.01, 0.05, 0.1, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0, 2.5, 3.0, 3.5, 4.0, 5.0, 10.0, 15.0, 20.0) are considered, the value providing the highest Q2 is chosen. In the second mode (gamma_grid_search=FALSE) γ is included in the list of the hyper-parameters subjected to optimization using the Nelder-Mead algorithm], (default: FALSE)
- conic_kernel_combination - whether to form conic kernel combination (with positive mixing coefficients h) (TRUE/FALSE or 1/0), (default: FALSE);
- optimize_h - whether to optimize the mixing coefficients h (TRUE/FALSE or 1/0), (default: FALSE);
- set_b_0 - whether to set the offset b=0 (this defines the difference between KRR1 and KRR2 modifications of the KRR method) (TRUE/FALSE or 1/0), (default: FALSE);
- print_ecv - print statistical parameters of the external n-folds cross-validation procedure (TRUE/FALSE, 1/0), (default: TRUE);
- plot_ecv - produce scatter plot for external coss-validation (TRUE/FALSE, 1/0), (default: TRUE);
- print_interm_icv – whether to print intermediate results in the course of optimization (TRUE/FALSE, 1/0), (default: TRUE);
- plot_interm_icv – whether to produce intermediate scatter plots in the course of optimization (TRUE/FALSE, 1/0), (default: TRUE);
- print_final_icv – whether to print final results on internal cross-validation (TRUE/FALSE, 1/0), (default: TRUE);
- plot_final_icv – whether to produce final scatter plot for internal cross-validation (TRUE/FALSE, 1/0), (default: TRUE);
- print_pred – whether to print statistical parameters for prediction (TRUE/FALSE, 1/0), (default: FALSE);
- plot_pred – whether to produce scatter plot prediction on external database (TRUE/FALSE, 1/0), (default: FALSE). In order to perform external n-fold cross-validation with reshuffles (random renumbering of molecules), one can launch the script **cmf-do-krr-ecvr**

```
# The number of folds for n-folds cross-validation
nfolds <- 5

# The number of reshuffles
nreshuffles <- 20

# Activity file name
act_all_fname <- "activity-train.txt"

# Activity column number
act_colnum <- 1

# Separator
sep <- ","
```

```
# Kernels file name
kernels_all_fname <- "ligands-kernels-train.RData"

# Types of molecular fields
mfields <- c("q","vdw","logp","abra","abrb")

# File name for reshuffled cross-validation results
ecvr_fname <- "ligands-ecvr-train.RData"

# Whether to perform grid search for attenuation factor alpha (TRUE/FALSE, 1/0).
alpha_grid_search <- FALSE

# Whether to perform grid search for gamma (TRUE/FALSE, 1/0).
gamma_grid_search <- FALSE

# Whether to form tconic kernel combination
# (with positive mixing coefficients) (TRUE/FALSE, 1/0)
# Conic combination preserves the positive definiteness of kernel matrices
# but leads to models with worse predictive performance assessed unsing cross-validation
conic_kernel_combination <- FALSE

# Whether to optimize h (TRUE/FALSE)
optimize_h <- TRUE

# Whether to set b=0 (TRUE/FALSE, 1/0)
set_b_0 <- FALSE

# Seed for random number generator (-1 - no seed)
seed <- 0

# Print statistical parameters of the reshuffled external n-folds cross-validation procedure (TRUE/FALS
print_ecvr <- TRUE

# Produce scatter plot for reshuffled external coss-validation (TRUE/FALSE, 1/0)
plot_ecvr <- TRUE

# Print statistical parameters of the external n-folds cross-validation procedure (TRUE/FALSE, 1/0)
print_ecv <- TRUE

# Produce scatter plot for external coss-validation (TRUE/FALSE, 1/0)
plot_ecv <- TRUE

# Print intermediate results in the course of optimiization (TRUE/FALSE, 1/0)
print_interm_icv <- FALSE

# Produce intermediate scatter plots in the course of optimization  (TRUE/FALSE, 1/0)
plot_interm_icv <- FALSE

# Print final results on interna; cross-validation (TRUE/FALSE, 1/0)
print_final_icv <- FALSE

# Produce final scatter plot for internal cross-valudation (TRUE/FALSE, 1/0)
plot_final_icv <- FALSE
```

```
# Print statistical parameters for prediction (TRUE/FALSE, 1/0)
print_pred <- FALSE

# Produce scatter plot prediction on external database (TRUE/FALSE, 1/0)
plot_pred <- FALSE

ecvr <- cmf_krr_ecvr(
  nfolds = nfolds,
  nreshuffles = nreshuffles,
  act_all_fname = act_all_fname,
  act_colnum = act_colnum,
  sep = sep,
  kernels_all_fname = kernels_all_fname,
  mfields = mfields,
  ecvr_fname = ecvr_fname,
  alpha_grid_search = alpha_grid_search,
  gamma_grid_search = gamma_grid_search,
  conic_kernel_combination = conic_kernel_combination,
  optimize_h = optimize_h,
  set_b_0 = set_b_0,
  seed = seed,
  print_ecvr = print_ecvr,
  plot_ecvr = plot_ecvr,
  print_ecv = print_ecv,
  plot_ecv = plot_ecv,
  print_interm_icv = print_interm_icv,
  plot_interm_icv = plot_interm_icv,
  print_final_icv = print_final_icv,
  plot_final_icv = plot_final_icv,
  print_pred = print_pred,
  plot_pred = plot_pred
)
```

in which the following parameters can be changed:

- nfolds – the number of folds in cross-validation (default: 5);
- act_all_fname - the name of the text file containing the values of activity for all compounds from the training set (default: “activity-all.txt”);
- act_colnum - activity column number in the file act_fname (default: 2);
- sep - separator of text fields in the file act_fname containing activity values (default: “,”);
- kernels_all_fname - the name of file with computed kernels (default: “ligands-kernels-all.RData”);
- mfields – the list of molecular fields for which kernels should be computed (default: c(“q”,“vdw”,“logp”,“abra”,“abrb”));
- ecvr_fname - the name of the file with the results of external cross-validation with reshuffles (default: “ligands-ecvr.RData”);
- alpha_grid_search- whether to perform the grid search for hyper-parameter αf. (TRUE/FALSE) or (1/0) [This option specifies the mode of finding the optimized values of αf (attenuation factor for molecular fields) values. In the first mode, when alpha_grid_search=TRUE, all values of gamma taken from the grid (the fixed set of 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1.0, 2.0, 3.0, 5.0, 10.0) are considered, the value providing the highest Q2 is chosen. In the second mode (alpha_grid_search=FALSE) γ is included in the list of the hyper-parameters subjected to optimization using the Nelder-Mead algorithm. In this case linear interpolation of kernel values is performed.], (default: TRUE);
- gamma_grid_search - whether to perform the grid search for hyper-parameter gamma (TRUE/FALSE) or (1/0) [This option specifies the mode of finding the optimized values of γ (regularization parameter in KRR) values. In the first mode, when gamma_grid_search=TRUE, all values of gamma taken

from the grid (the fixed set of 0.01, 0.05, 0.1, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0, 2.5, 3.0, 3.5, 4.0, 5.0, 10.0, 15.0, 20.0) are considered, the value providing the highest Q2 is chosen. In the second mode (gamma_grid_search=FALSE) γ is included in the list of the hyper-parameters subjected to optimization using the Nelder-Mead algorithm], (default: FALSE)

- conic_kernel_combination - whether to form conic kernel combination (with positive mixing coefficients h) (TRUE/FALSE or 1/0), (default: FALSE);
- optimize_h - whether to optimize the mixing coefficients h (TRUE/FALSE or 1/0), (default: FALSE);
- set_b_0 - whether to set the offset b=0 (this defines the difference between KRR1 and KRR2 modifications of the KRR method) (TRUE/FALSE or 1/0), (default: FALSE);
- seed – seed for random number generator (seed<0 means that seed is not specified)
- print_ecv - print statistical parameters of the external n-folds cross-validation procedure (TRUE/FALSE, 1/0), (default: TRUE);
- plot_ecv - produce scatter plot for external coss-validation (TRUE/FALSE, 1/0), (default: TRUE);
- print_interm_icv – whether to print intermediate results in the course of optimization (TRUE/FALSE, 1/0), (default: TRUE);
- plot_interm_icv – whether to produce intermediate scatter plots in the course of optimization (TRUE/FALSE, 1/0), (default: TRUE);
- print_final_icv – whether to print final results on internal cross-validation (TRUE/FALSE, 1/0), (default: TRUE);
- plot_final_icv – whether to produce final scatter plot for internal cross-validation (TRUE/FALSE, 1/0), (default: TRUE);
- print_pred – whether to print statistical parameters for prediction (TRUE/FALSE, 1/0), (default: FALSE);
- plot_pred – whether to produce scatter plot prediction on external database (TRUE/FALSE, 1/0), (default: FALSE). Models built for each fold of the external cross-validation with reshuffles procedure and stored in the file with its results can be used for making multiple predictions for an external test set. This can be carried out using the script **cmf-do-krr-ecvr-pred**:

```
# File name for reshuffled external cross-validation
ecvr_fname <- "ligands-ecvr.RData"

# Kernels for training file name
kernels_train_fname <- "ligands-kernels-train.RData"

# Kernels for prediction file name
kernels_pred_fname <- "ligands-kernels-pred.RData"

# Column name or activity (if specified)
act_colnum <- 1

# Separator
sep <- ","

# Activity for prediction set (if specified)
act_pred_fname <- "activity-pred.txt"

# File with predictions
pred_fname <- "ligands-ecvr-pred.RData"

# Print statistical parameters for prediction (TRUE/FALSE, 1/0)
print_pred <- TRUE

# Produce scatter plot prediction on external database (TRUE/FALSE, 1/0)
plot_pred <- TRUE
```

```
ecvr_pred <- cmf_krr_ecvr_pred(
  ecvr_fname = ecvr_fname,
  kernels_train_fname = kernels_train_fname,
  kernels_pred_fname = kernels_pred_fname,
  act_colnum = act_colnum,
  sep = sep,
  act_pred_fname = act_pred_fname,
  pred_fname = pred_fname,
  print_pred = print_pred,
  plot_pred = plot_pred
)
```

in which the following parameters can be changed:

- ecvr_fname - the name of the file with the results of external cross-validation with reshuffles (default: "ligands-ecvr.RData");
- kernels_train_fname - the name of file with computed kernels (default: "ligands-kernels-train.RData");
- kernels_pred_fname - file name of the file containing kernels that compare molecular fields of molecules belonging to the training and the test sets (default: "ligands-kernels-pred.RData");
- act_colnum - activity column number in the file act_pred_fname; zero value means that the activity is not specified (default: 2);
- sep – separator of columns in file act_pred_fname (default: ",");
- act_pred_fname – file with activity values for the test set (if specified), (default: "activity-pred.txt").
- pred_fname - the name of the file containing the results of predictions (default: "ligands-ecvr-pred.mol2");
- print_pred – whether to print statistical parameters for prediction (TRUE/FALSE, 1/0), (default: FALSE);
- plot_pred – whether to produce scatter plot prediction on external database (TRUE/FALSE, 1/0), (default: FALSE). Histogram of predicted values for each individual compounds can be visualized using the script **cmf-do-ecvr-view-pred**:

```
# Visualisation of prediction histogram from multiple predictions m
# made by external cross-validation with reshuffling (ecvr)

# File with predictions
pred_fname <- "ligands-ecvr-pred.RData"

# Molecule to predict
imol <- 1

load(pred_fname)
hist(ecvr_pred$YPred[imol,], main="Prediction Histogram", xlab="Predicted Value", ylab="Frequency")
```

in which the following parameters can be changed:

- pred_fname - the name of the file containing the results of predictions (default: "ligands-ecvr-pred.mol2");
- imol – molecule for which predictions should be done.

4.6. Producing Scatter Plots

The script **cmf-do-krr-train**:

```
# KRR with continuous molecular fields and optimization of  gamma and h.
# alpha is optimized by grid search
# Any combination of molecular fields is supported

# Activity file name
act_train_fname <- "activity-train.txt"

# Activity column number
act_colnum <- 1

# Separator
sep <- ","

# Kernels file name
kernels_train_fname <- "ligands-kernels-train.RData"

# Model file name
model_fname <- "ligands-model.RData"

# Types of molecular fields
mfields <- c("q","vdw","logp","abra","abrb")

# Whether to perform grid search for attenuation factor alpha (TRUE/FALSE, 1/0).
alpha_grid_search <- TRUE

# Whether to perform grid search for Tikhonov regularization coefficient gamma (TRUE/FALSE, 1/0).
gamma_grid_search <- FALSE

# Whether to form tconic kernel combination
conic_kernel_combination <- FALSE

# Whether to optimize h (TRUE/FALSE, 1/0)
optimize_h <- FALSE

# Whether to set b=0 (TRUE/FALSE, 1/0)
set_b_0 <- FALSE

# Print intermediate results in the course of optimiization (TRUE/FALSE, 1/0)
print_interm_icv <- TRUE

# Produce intermediate scatter plots in the course of optimization  (TRUE/FALSE, 1/0)
plot_interm_icv <- TRUE

# Print final results on interna; cross-validation (TRUE/FALSE, 1/0)
print_final_icv <- TRUE

# Produce final scatter plot for internal cross-valudation (TRUE/FALSE, 1/0)
plot_final_icv <- TRUE

cmf_krr_train(
  act_train_fname = act_train_fname,
  act_colnum = act_colnum,
  sep = sep,
  kernels_train_fname = kernels_train_fname,
  model_fname = model_fname,
  mfields = mfields,
  alpha_grid_search = alpha_grid_search,
  gamma_grid_search = gamma_grid_search,
  conic_kernel_combination = conic_kernel_combination,
  optimize_h = optimize_h,
  set_b_0 = set_b_0,
  print_interm_icv = print_interm_icv,
  plot_interm_icv = plot_interm_icv,
  print_final_icv = print_final_icv,
  plot_final_icv = plot_final_icv
)
```

and **cmf-do-krr-pred**:

```
# Model file name
model_fname <- "ligands-model.RData"

# Kernels for training file name
kernels_train_fname <- "ligands-kernels-train.RData"

# Kernels for prediction file name
kernels_pred_fname <- "ligands-kernels-pred.RData"

# Column name or activity (if specified)
act_colnum <- 1

# Separator
sep <- ","

# Activity for prediction set (if specified)
act_pred_fname <- "activity-pred.txt"

# File with predictions
pred_fname <- "ligands-pred.RData"

# Print statistical parameters for prediction (TRUE/FALSE, 1/0)
print_pred <- TRUE

# Produce scatter plot prediction on external database (TRUE/FALSE, 1/0)
plot_pred <- TRUE

res <- cmf_krr_pred(
  model_fname = model_fname,
  kernels_train_fname = kernels_train_fname,
  kernels_pred_fname = kernels_pred_fname,
  act_colnum = act_colnum,
  sep = sep,
  act_pred_fname = act_pred_fname,
  pred_fname = pred_fname,
  print_pred = print_pred,
  plot_pred = plot_pred
)
```

not only build 3D-QSAR models and use them to make predictions, but also produce the corresponding scatter plots.

Meanwhile, if it is necessary to reproduce these scatter plots, two computational R scripts can be used for this purpose:

The script **cmf-do-plot-cv**:

```
# Label for axis X
xlab <- "Predicted"

# Label for axis Y
ylab <- "Experimental"

# Margin for points in all directions
margin <- 0.5

load(fmodel)

cinf_plotxy(
  model$y_pred_cv,
  model$y_exp,
  xlab = xlab,
  ylab = ylab,
  margin = margin,
  main = main
)
```

produces the scatter plot Predicted/Expetimental for the results of internal cross-validation. The values of the following parameters may be modified in the script:

- fmodel - the name of the file containing model (default: "ligands-model.RData");
- main – the main title (default: "Scatter Plot for Cross-Validation (Internal)");
- xlab - text label for axis X (default: "Predicted");
- ylab - text label for axis Y (default: "Experimental");
- margin - margin for points in all directions (default: 0.5).

and

The script **cmf-do-plot-pred** produces scatter plot Predicted/Experimental for the results of predictions performed on external test set. The values of the following parameters may be modified in the script:

- fpred - the name of the file with predictions (default: "ligands-pred.RData");
- main – the main title (default: "Scatter Plot for Prediction");
- xlab - text label for axis X (default: "Predicted");
- ylab - text label for axis Y (default: "Experimental");
- margin - margin for points in all directions (default: 0.5)

```
# Script for drawing scatter plot for predicting test set


# File with predictions
fpred <- "ligands-pred.RData"

# Main title
main <- "Scatter Plot for Prediction"

# Label for axis X
xlab <- "Predicted"

# Label for axis Y
```

```
ylab <- "Experimental"

# Margin for points in all directions
margin <- 0.5

load(fpred)
cinf_plotxy(
  pred$y_pred,
  pred$y_exp,
  xlab=xlab,
  ylab=ylab,
  margin=margin,
  main=main
)
```

4.7. Drawing Iso-surfaces for the Fields of Regression Coefficients and their Overlap with Molecular Fields

The fields of regression coefficients can be visualized with the help of iso-surfaces (colored according to the sign of coefficients and the type of molecular field). The procedure for constructing such iso-surfaces includes two stages. In the first stage the values of regression coefficients are calculated on all nodes in an imaginary grid using the R computational script **cmf-do-gen-grid**:

```
# Generation of grid for continuous molecular co-fields

# Training set file name
train_fname <- "ligands-train.mol2"

# Computed kernels file name
kernels_fname <- "ligands-kernels-train.RData"

# Model file name
model_fname <- "ligands-model.RData"

# Grid with regression coefficients file name
grid_fname <- "ligands-grid-krr.RData"

cmf_gen_grid(
  train_fname = train_fname,
  kernels_fname = kernels_fname,
  model_fname = model_fname,
  grid_fname = grid_fname
)
```

All results are written to file. The values of the following parameters may be modified in the script:

- train_fname - the name of the mol2 file containing the structures from the training set (default: “ligands-train.mol2”);
- kernels_fname – file with computed kernels (default: “ligands-kernels-train.RData”);
- model_fname - the name of the file with model (default: “ligands-model.RData”);
- grid_fname - the name of the file containing regression coefficients computed at all grid nodes (default: “ligands-grid-krr.RData”). At the second stage, iso-surfaces for the fields of regression coefficients (co-fields) are rendered in 3D. For the sake of interpretation, a molecule from the training set is also visualized in the same picture. In order to draw it, one should launch the script:

```
# View molecule with co-field


# Molecular database file name
mdb_fname <- "ligands-train.mol2"

# Molecule to visualize
imol <- 1

# Field type
ft <- "vdw"

# Grid with model coefficients file name
grid_fname <- "ligands-grid-krr.RData"

# Isosurface level
rlevel <- 0.5

# Alpha (non-transperancy) level
alpha_g <- 0.7

cmf_view_mol_field_cofield(
  mdb_fname = mdb_fname,
  imol = imol,
  ft = ft,
  grid_fname = grid_fname,
  rlevel = rlevel,
  alpha_g = alpha_g,
  draw_field = FALSE,
  draw_cofield = TRUE
)
```

The values of the following parameters may be modified in the script:

- mdb_fname - the name of the mol2 file containing the structures from molecular database (default: “ligands-train.mol2”);
- imol - a molecule to visualize (default: 1);
- ft - the field type (q/vdw/logp/abra/abrb), (default: “logp”);
- grid_fname - the name of the file containing model (regression) coefficients computed at all grid nodes (default: “ligands-grid-krr.RData”);
- rlevel – isosurface level (default: 0.5);
- alpha - alpha (non-transperancy) level for surfaces in 3D graphics (default: 0.7). One can also visualize overlapping fields of regression coefficients (i.e. co-fields) with the corresponding molecular fields. This can be done by running script **cmf-do-view-mol-field-cofield**:

```
# View overlapping fields of regression coefficients and molecular fields

# Molecular database file name
mdb_fname <- "ligands-train.mol2"

# Molecule to visualize
imol <- 1

# Field type
ft <- "vdw"
```

```
# Grid with regression coefficients file name
grid_fname <- "ligands-grid-krr.RData"

# Take alpha from model? (TRUE/FALSE)
alpha_from_model <- TRUE

# Model file name
model_fname <- "ligands-model.RData"

# Alpha value (if not taken from model)
alpha <- 1.0

# Isosurface level
rlevel <- 0.5

# Alpha (non-transperancy) level
alpha_g <- 0.5

# Whether to draw overlap between field and co-field
draw_overlap <- TRUE

cmf_view_mol_field_cofield(
  mdb_fname = mdb_fname,
  imol = imol,
  ft = ft,
  grid_fname = grid_fname,
  alpha_from_model = alpha_from_model,
  model_fname = model_fname,
  alpha = alpha,
  rlevel = rlevel,
  alpha_g = alpha_g,
  draw_field = TRUE,
  draw_cofield = TRUE,
  draw_overlap = draw_overlap
)
```

One can change the default values of the following parameters in this script:

- mdb_fname - the name of the mol2 file containing the structures from molecular database (default: “ligands-train.mol2”);
- imol - a molecule to visualize (default: 1);
- ft - the field type (q/vdw/logp/abra/abrb), (default: “vdw”);
- grid_fname - the name of the file containing model (regression) coefficients computed at all - - grid nodes (default: “ligands-grid-krr.RData”);
- alpha_from_model – whether to take the attenuation parameter αf from the stored model (TRUE/FALSE or 1/0), (default: TRUE);
- model_fname – model file name (from which the values of the attenuation parameter αf should be taken whenever alpha_from_model=TRUE), (default: “ligands-model.RData”);
- alpha – the value of the attenuation parameter αf (default: 1.0);
- rlevel – isosurface level (for both the fields of regression coefficients and molecular fields), (default: 0.5);
- alpha_g – non-transperancy level for surfaces (default: 0.5);
- draw_overlap – whether to visualize the overlap separately with additional iso-surfaces (TRUE/FALSE or 1/0) (default: FALSE).

Molecular fields without fields of regression coefficients can be visualized using the below computational R script:

```
# View overlapping fields of regression coefficients and molecular fields

# Molecular database file name
mdb_fname <- "ligands-train.mol2"

# Molecule to visualize
imol <- 1

# Field type
ft <- "vdw"

# Take alpha from model? (TRUE/FALSE)
alpha_from_model <- TRUE

# Model file name
model_fname <- "ligands-model.RData"

# Alpha value (if not taken from model)
alpha <- 1.0

# Isosurface level
rlevel <- 0.5

# Alpha (non-transperancy) level
alpha_g <- 0.5

cmf_view_mol_field_cofield(
  mdb_fname = mdb_fname,
  imol = imol,
  ft = ft,
  grid_fname = grid_fname,
  alpha_from_model = alpha_from_model,
  model_fname = model_fname,
  alpha = alpha,
  rlevel = rlevel,
  alpha_g = alpha_g,
  draw_field = TRUE,
  draw_cofield = FALSE
)
```

Default values of the following parameters can be changed in it:

- mdb_fname - the name of the mol2 file containing the structures from molecular database (default: “ligands-train.mol2”);
- imol - a molecule to visualize (default: 1);
- ft - the field type (q/vdw/logp/abra/abrb), (default: “vdw”);
- alpha_from_model – whether to take the attenuation parameter αf from the stored model (TRUE/FALSE or 1/0), (default: TRUE);
- model_fname – model file name (from which the values of the attenuation parameter αf should be taken whenever alpha_from_model=TRUE), (default: “ligands-model.RData”);

- alpha – the value of the attenuation parameter αf (default: 1.0);
- rlevel – isosurface level (for both the fields of regression coefficients and molecular fields), (default: 0.5);
- alpha_g – non-transperancy level for surfaces (default: 0.5.

4.8. Analyzing Contributions to Predicted Activity Values

Contributions of different types of molecular fields and different training examples to predicted activity (property) values of compounds belonging to the test set can be calculated using the script **cmd-do- anal-contrib-pred**. The results of the analysis can be read from the variable a. The default values of the following parameters can be changes:

- model_fname – model file name (default: “ligands-model.RData”);
- kernels_pred_fname - file name of the file containing kernels that compare molecular fields of molecules belonging to the training and the test sets (default: “ligands-kernels-pred.RData”);
- act_colnum - activity column number in the file act_pred_fname; zero value means that the activity is not specified (default: 1);
- sep – separator of columns in file act_pred_fname (default: “,”);
- act_pred_fname – file with activity values for the test set (if specified), (default: “activity-pred.txt”). Contributions of different types of molecular fields and different training examples to predicted activity (property) values of compounds belonging to the training set can be calculated using the script **cmf-do-anal-contrib-train**:

```
# Prediction with analysis for the test set

# Model file name
model_fname <- "ligands-model.RData"

# Kernels file name
kernels_pred_fname <- "ligands-kernels-pred.RData"

# Column name or activity (if specified)
act_colnum <- 2

# Separator
sep <- ","

# Activity for prediction set (if specified)
act_pred_fname <- "activity-pred.txt"

anal <- cmf_pred_anal(
  model_fname = model_fname,
  kernels_pred_fname = kernels_pred_fname,
  act_colnum = act_colnum,
  sep = sep,
  act_pred_fname = act_pred_fname,
  is_train = FALSE
)
```

The results of the analysis can be read from the variable anal. The default values of the following parameters can be changes:

- model_fname – model file name (default: “ligands-model.RData”);
- kernels_fname_train - file name of the file containing kernels for - the training set (default: “ligands-kernels-train.RData”);
- act_colnum - activity column number in the file act_train_fname (default: 2);

- sep – separator of columns in file act_train_fname (default: “,”);
- act_train_fname – file with activity values for the training set (default: “activity-train.txt”).

Contributions of individual atoms to predicted activity (property) values of compounds belonging to the test set can be calculated using the script **cmf-do-anal-contrib-atoms-pred**:

```
# Analysis of atomic contributions for the test set


# Training set file name
train_fname <- "ligands-train.mol2"

# Prediction set file name
pred_fname <- "ligands-pred.mol2"

# Molecule from file pred_fname to be analyzed
imol <- 1

# Model file name
model_fname <- "ligands-model.RData"

cmf_pred_anal_atoms(
  train_fname = train_fname,
  pred_fname = pred_fname,
  imol = imol,
  model_fname = model_fname
)
```

The default values of the following parameters can be changes:

- train_fname – training set file name (default: “ligands-train.mol2”);
- pred_fname – test set file name (default: “ligands-pred.mol2”);
- imol – molecule from file pred_fname to be analyzed (default: 1);
- model_fname – model file name (default: “ligands-model.RData”). Contributions of individual atoms to predicted activity (property) values of compounds belonging to the training set can be calculated using the script **cmf-do-anal-contrib-atoms-train**:

```
# Analysis of atomic contributions in molecules from the training set


# Training set file name
train_fname <- "ligands-train.mol2"

# Molecule from file train_fname to be analyzed
imol <- 1

# Model file name
model_fname <- "ligands-model.RData"

cmf_pred_anal_atoms(
  train_fname = train_fname,
  pred_fname = train_fname,
  imol = imol,
  model_fname = model_fname
)
```

The default values of the following parameters can be changes:

- train_fname – training set file name (default: “ligands-train.mol2”);
- imol – molecule from file train_fname to be analyzed (default: 1);
- model_fname – model file name (default: “ligands-model.RData”).

4.9. Exploring Applicability Domain of 3D-QSAR Models

The package contains several tools for exploring applicability domains of 3D-QSAR model. If the standard deviation of prediction in external cross-validation with reshuffles is chosen as a distance to model parameter and mean prediction is used for computing prediction error, one can produce “coverage – mean error” and “distance to model - error” plots. The first one can be produced by running the script **cmf-do-ecvr-plot-coverage-mean-error**:

```
# Applicability domain from results of external cross-validation with reshuffles


# File name for results of external cross-validation with reshuffles
ecvr_fname <- "ligands-ecvr-train.RData"

# Coverage - mean error plot type ("in", "out", "diff")
# "in" - mean error is calculated inside applicability domain
# "out" - mean error is calculated outside applicability domain
# "diff" - difference of mean errors outside and inside applicability domain
cmep_type <- "in"

# Whether to draw smoothing curve (TRUE/FALSE, 1/0)
smoothing <- TRUE

cmf_ecvr_plot_coverage_mean_error(
  ecvr_fname = ecvr_fname,
  cmep_type = cmep_type,
  smoothing = smoothing
)
```

, in which the following default values of parameters can be changed:

- ecvr_fname - file name for results of external cross-validation with reshuffles (default: ligands-ecvr.RData);
- cmep_type – the type of the “coverage – mean error” plot (“in” - mean error is calculated inside applicability domain; “out” - mean error is calculated outside applicability domain; “diff” - difference of mean errors outside and inside applicability domain) (default: “in”);
- smoothing - whether to draw smoothing curve (TRUE/FALSE, 1/0) (default: TRUE). The “distance to model - error” plot can be produced by launching the script **cmf-do-ecvr-plot-d2m-error**:

```
# Producing "distance to model - error" plot for applicability domain studies


# File name for resukts of external cross-validation with reshuffles
ecvr_fname <- "ligands-ecvr.RData"

# Whether to calculate mean error values for parts of the points (TRUE/FALSE, 1/0)
mean_parts <- TRUE

# The number of parts for averaging
nparts <- 3
```

```
# Color for drawing mean errors of parts
color_parts <- "red"

cmf_ecvr_plot_d2m_error(
  ecvr_fname=ecvr_fname,
  mean_parts=mean_parts,
  nparts=nparts,
  color_parts=color_parts
)
```

, in which the following default values of parameters can be changed:

- ecvr_fname - file name for results of external cross-validation with reshuffles (default: ligands-ecvr.RData);
- mean_parts - Whether to calculate mean error values for parts of the points (TRUE/FALSE, 1/0) (default: TRUE);
- nparts - the number of parts for averaging (default: 3);
- color_parts - color for drawing mean errors of parts (default: "red").