Package: FastRet (via r-universe)

February 11, 2025

Title Retention Time Prediction in Liquid Chromatography

Version 1.1.4

Description A framework for predicting retention times in liquid chromatography. Users can train custom models for specific chromatography columns, predict retention times using existing models, or adjust existing models to account for altered experimental conditions. The provided functionalities can be accessed either via the R console or via a graphical user interface. Related work: Bonini et al. (2020) <doi:10.1021/acs.analchem.9b05765>.

License GPL-3

Language en-US

URL https://github.com/spang-lab/FastRet/, https://spang-lab.github.io/FastRet/

BugReports https://github.com/spang-lab/FastRet/issues

biocViews Retention, Time, Chromotography, LC-MS

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports bslib, caret, cluster, data.table, digest, DT, future, ggplot2, glmnet, htmltools, promises, rcdk, readxl, shiny (>= 1.8.1), shinybusy, shinyhelper, shinyjs, xgboost, xlsx

Suggests cli, devtools, knitr, languageserver, lintr, pkgdown, pkgbuild, pkgload, rlang, rmarkdown, servr, tibble, testthat (>= 3.0.0), toscutil, usethis, withr

LazyData true

Config/testthat/edition 3

Config/testthat/parallel true

Config/testthat/start-first train_frm-gbtree, train_frm-lasso, preprocess_data, read_rp_xlsx, fit_gbtree, getCDsFor1Molecule, plot_frm, adjust_frm, read_rpadj_xlsx

Config/pak/sysreqs make default-jdk libicu-dev libpng-dev zlib1g-dev

Repository https://spang-lab.r-universe.dev

RemoteUrl https://github.com/spang-lab/fastret

RemoteRef HEAD

RemoteSha afc5eb72d68d6f3673208ebea9ecfb29bf093f93

Contents

`adjust_frm` *Adjust an existing FastRet model for use with a new column*

Description

The goal of this function is to train a model that predicts RT_ADJ (retention time measured on a new, adjusted column) from RT (retention time measured on the original column) and to attach this "adjustmodel" to an existing FastRet model.

Usage

```
adjust_frm(
  frm = train_frm(),
  new_data = read_rpadj_xlsx(),
  predictors = 1:6,
  nfolds = 5,
  verbose = 1
)
```

Arguments

frm	An object of class frm as returned by train_frm().
new_data	Dataframe with columns "RT", "NAME", "SMILES" and optionally a set of chemical descriptors.
predictors	Numeric vector specifying which predictors to include in the model in addition to RT. Available options are: 1=RT, 2=RT^2, 3=RT^3, 4=log(RT), 5=exp(RT), 6=sqrt(RT).
nfolds	An integer representing the number of folds for cross validation.
verbose	A logical value indicating whether to print progress messages.

Value

An object of class frm, which is a list with the following elements:

- model: A list containing details about the original model.
- df: The data frame used for training the model.
- cv: A list containing the cross validation results.
- seed: The seed used for random number generation.
- version: The version of the FastRet package used to train the model.
- adj: A list containing details about the adjusted model.

Examples

```
frm <- read_rp_lasso_model_rds()
new_data <- read_rpadj_xlsx()
frmAdjusted <- adjust_frm(frm, new_data, verbose = 0)
```

---

fastret_app *The FastRet GUI*

Description

Creates the FastRet GUI

Usage

```
fastret_app(port = 8080, host = "0.0.0.0", reload = FALSE, nsw = 1)
```

Arguments

port	The port the application should listen on
host	The address the application should listen on
reload	Whether to reload the application when the source code changes
nsw	The number of subworkers each worker is allowed to start. The higher this number, the faster individual tasks like model fitting can be processed.

Value

A shiny app. This function returns a shiny app that can be run to interact with the model.

An object of class `shiny.appobj`.

Examples

```
x <- fastret_app()
if (interactive()) shiny::runApp(x)
```

`getCDs` *Get Chemical Descriptors for a list of molecules*

Description

Calculate Chemical Descriptors for a list of molecules. Molecules can appear multiple times in the list.

Usage

```
getCDs(df, verbose = 1, nw = 1)
```

Arguments

`df`	dataframe with two mandatory columns: "NAME" and "SMILES"
`verbose`	0: no output, 1: progress, 2: more progress and warnings
`nw`	number of workers for parallel processing

Value

A dataframe with the chemical descriptor values appended as columns to the input dataframe.

Examples

```
cds <- getCDs(head(RP, 3), verbose = 1, nw = 1)
```

predict.frm *Predict retention times using a FastRet Model*

Description

Predict retention times for new data using a FastRet Model (FRM).

Usage

```
## S3 method for class 'frm'
predict(object = train_frm(), df = object$df, adjust = NULL, verbose = 0, ...)
```

Arguments

object	An object of class frm as returned by train_frm().
df	A data.frame with the same columns as the training data.
adjust	If object was adjusted using adjust_frm(), it will contain a property object$adj. If adjust is TRUE, object$adj will be used to adjust predictions obtained from object$model. If FALSE object$adj will be ignored. If NULL, object$model will be used, if available.
verbose	A logical value indicating whether to print progress messages.
...	Not used. Required to match the generic signature of predict().

Value

A numeric vector with the predicted retention times.

See Also

train_frm(), adjust_frm()

Examples

```
frm <- read_rp_lasso_model_rds()
newdata <- head(RP)
yhat <- predict(frm, newdata)
```

preprocess_data *Preprocess data*

Description

Preprocess data so they can be used as input for `train_frm()`.

Usage

```
preprocess_data(
  data,
  degree_polynomial = 1,
  interaction_terms = FALSE,
  verbose = 1,
  nw = 1
)
```

Arguments

`data`	dataframe with columns RT, NAME, SMILES
`degree_polynomial`	defines how many polynomials get added (if 3 quadratic and cubic terms get added)
`interaction_terms`	if TRUE all interaction terms get added to data set
`verbose`	0 == no output, 1 == show progress, 2 == show progress and warnings
`nw`	number of workers to use for parallel processing

Value

A dataframe with the preprocessed data

Examples

```
data <- head(RP, 3) # Only use first three rows to speed up example runtime
pre <- preprocess_data(data, verbose = 0)
```

read_retip_hilic_data *Download and read the HILIC dataset from Retip the package*

Description

Downloads and reads the HILIC dataset from the Retip package. The dataset is downloaded from `https://github.com/oloBion/Retip/raw/master/data/HILIC.RData`, saved to a temporary file and then read and returned.

Usage

```
read_retip_hilic_data(verbose = 1)
```

Arguments

`verbose`	Verbosity level. 1 == print progress messages, 0 == no progress messages.

Value

df A data frame containing the HILIC dataset.

References

Retip: Retention Time Prediction for Compound Annotation in Untargeted Metabolomics Paolo Bonini, Tobias Kind, Hiroshi Tsugawa, Dinesh Kumar Barupal, and Oliver Fiehn Analytical Chemistry 2020 92 (11), 7515-7522 DOI: 10.1021/acs.analchem.9b05765

Examples

```
df <- read_retip_hilic_data(verbose = 0)
```

read_rpadj_xlsx *Hypothetical retention times (RT) measured on a reverse phase (RP) column*

Description

Subset of the data from `read_rp_xlsx()` with some slight modifications to simulate changes in temperature and/or flowrate.

Usage

```
read_rpadj_xlsx()
```

Value

A dataframe with 25 rows (metabolites) and 3 columns `RT`, `SMILES` and `NAME`.

Examples

```
x <- read_rpadj_xlsx()
```

`read_rp_lasso_model_rds` *LASSO Model trained on RP dataset*

Description

Read a LASSO model trained on the RP dataset using `train_frm()`.

Usage

```
read_rp_lasso_model_rds()
```

Value

A `frm` object.

Examples

```
frm <- read_rp_lasso_model_rds()
```

`read_rp_xlsx` *Read retention times (RT) measured on a reverse phase (RP) column*

Description

Read retention time data from a reverse phase liquid chromatography measured with a temperature of 35 degree and a flowrate of 0.3ml/min. The data also exists as dataframe in the package. To use it directly in R just enter `RP`.

Usage

```
read_rp_xlsx()
```

Value

A dataframe of 442 metabolites with columns `RT`, `SMILES` and `NAME`.

Source

Measured by functional genomics lab at the University of Regensburg.

See Also

RP

Examples

```
x <- read_rp_xlsx()
all.equal(x, RP)
```

RP *Retention Times (RT) Measured on a Reverse Phase (RP) Column*

Description

Retention time data from a reverse phase liquid chromatography measured with a temperature of 35 degree and a flowrate of 0.3ml/min. The same data is available as an xlsx file in the package. To read it into R use `read_rp_xlsx()`.

Usage

```
RP
```

Format

A dataframe of 442 metabolites with the following columns:

RT Retention time

SMILES SMILES notation of the metabolite

NAME Name of the metabolite

Source

Measured by functional genomics lab at the University of Regensburg.

See Also

read_rp_xlsx

`selective_measuring` *Selective Measuring*

Description

The function `adjust_frm()` is used to modify existing FastRet models based on changes in chromatographic conditions. It requires a set of molecules with measured retention times on both the original and new column. This function selects a sensible subset of molecules from the original dataset for re-measurement. The selection process includes:

1. Generating chemical descriptors from the SMILES strings of the molecules. These are the features used by `train_frm()` and `adjust_frm()`.
2. Standardizing chemical descriptors to have zero mean and unit variance.
3. Training a Ridge Regression model with the standardized chemical descriptors as features and the retention times as the target variable.
4. Scaling the chemical descriptors by coefficients of the Ridge Regression model.
5. Applying PAM clustering on the entire dataset, which includes the scaled chemical descriptors and the retention times.
6. Returning the clustering results, which include the cluster assignments, the medoid indicators, and the raw data.

Usage

```
selective_measuring(raw_data, k_cluster = 25, verbose = 1)
```

Arguments

`raw_data`	The raw data to be processed. Must be a dataframe with columns NAME, RT and SMILES.
`k_cluster`	The number of clusters for PAM clustering.
`verbose`	The level of verbosity.

Value

A list containing the following elements:

- `clustering`: a data frame with raw data, cluster assignments, and medoid indicators
- `clobj`: the PAM clustering object
- `coefs`: the coefficients from the Ridge Regression model
- `model`: the Ridge Regression model
- `df`: the preprocessed data
- `dfz`: the standardized features
- `dfzb`: the features scaled by coefficients of the Ridge Regression model

Examples

```
x <- selective_measuring(RP[1:50, ], k = 5, verbose = 0)
# For the sake of a short runtime, only the first 50 rows of the RP dataset
# were used in this example. In practice, you should always use the entire
# dataset to find the optimal subset for re-measurement.
```

`start_gui` *Start the FastRet GUI*

Description

Starts the FastRet GUI

Usage

```
start_gui(port = 8080, host = "0.0.0.0", reload = FALSE, nw = 2, nsw = 1)
```

Arguments

`port`	The port the application should listen on
`host`	The address the application should listen on
`reload`	Whether to reload the application when the source code changes
`nw`	The number of worker processes started. The first worker always listens for user input from the GUI. The other workers are used for handling long running tasks like model fitting or clustering. If `nw` is 1, the same process is used for both tasks, which means that the GUI will become unresponsive during long running tasks.
`nsw`	The number of subworkers each worker is allowed to start. The higher this number, the faster individual tasks like model fitting can be processed. A value of 1 means that all subprocesses will run sequentially.

Details

If you set `nw = 3` and `nsw = 4`, you should have at least 16 cores, one core for the shiny main process. Three cores for the three worker processes and 12 cores (3 * 4) for the subworkers. For the default case, `nworkers = 2` and `nsw = 1`, you only need 3 cores, as `nsw = 1` means that all subprocesses will run sequentially.

Value

A shiny app. This function returns a shiny app that can be run to interact with the model.

Examples

```
if (interactive()) start_gui()
```

`train_frm` *Train a new FastRet model (FRM) for retention time prediction*

Description

Trains a new model from molecule SMILES to predict retention times (RT) using the specified method.

Usage

```
train_frm(
  df,
  method = "lasso",
  verbose = 1,
  nfolds = 5,
  nw = 1,
  degree_polynomial = 1,
  interaction_terms = FALSE,
  rm_near_zero_var = TRUE,
  rm_na = TRUE,
  rm_ns = FALSE,
  seed = NULL
)
```

Arguments

`df`	A dataframe with columns "NAME", "RT", "SMILES" and optionally a set of chemical descriptors. If no chemical descriptors are provided, they are calculated using the function `preprocess_data()`.
`method`	A string representing the prediction algorithm. Either "lasso", "ridge" or "gbtree".
`verbose`	A logical value indicating whether to print progress messages.
`nfolds`	An integer representing the number of folds for cross validation.
`nw`	An integer representing the number of workers for parallel processing.
`degree_polynomial`	An integer representing the degree of the polynomial. Polynomials up to the specified degree are included in the model.
`interaction_terms`	A logical value indicating whether to include interaction terms in the model.
`rm_near_zero_var`	A logical value indicating whether to remove near zero variance predictors. Setting this to TRUE can cause the CV results to be overoptimistic, as the variance filtering is done on the whole dataset, i.e. information from the test folds is used for feature selection.

rm_na — A logical value indicating whether to remove NA values. Setting this to TRUE can cause the CV results to be overoptimistic, as the variance filtering is done on the whole dataset, i.e. information from the test folds is used for feature selection.

rm_ns — A logical value indicating whether to remove chemical descriptors that were considered as not suitable for linear regression based on previous analysis of an independent dataset.

seed — An integer value to set the seed for random number generation to allow for reproducible results.

Details

Setting rm_near_zero_var and/or rm_na to TRUE can cause the CV results to be overoptimistic, as the predictor filtering is done on the whole dataset, i.e. information from the test folds is used for feature selection.

Value

A trained FastRet model.

Examples

```
system.time(m <- train_frm(RP[1:80, ], method = "lasso", nfolds = 2, nw = 1, verbose = 0))
# For the sake of a short runtime, only the first 80 rows of the RP dataset
# are used in this example. In practice, you should always use the entire
# training dataset for model training.
```

Index