Package 'MetaClean'

September 11, 2020

Type Package

Title Detection of Low-Quality Peaks in Untargeted Metabolomics Data

Version 1.0.0

Author Kelsey Chetnik

Maintainer Kelsey Chetnik <kchetnik73@gmail.com>

Description Utilizes 11 peak quality metrics and 8 diverse machine learning algorithms to build a classifier for the automatic
assessment of peak integration quality of peaks from untargeted metabolomics analyses.
The 12 peak quality metrics were adapted from those defined in the following references:
Zhang, W., & Zhao, P.X. (2014) <doi:10.1186/1471-2105-15-S11-S5>
Toghi Eshghi, S., Auger, P., & Mathews, W.R. (2018) <doi:10.1186/s12014-018-9209-x>.

biocViews S4Vectors

Imports reshape2, knitr, ggplot2, plotrix, tools, utils, klaR,
fastAdaboost, rpart, randomForest, kernlab, BiocStyle, methods,
graph, Rgraphviz, caret

Depends R (>= 3.5.0), MLmetrics, xcms

Suggests markdown

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2020-09-11 18:50:03 UTC

R topics documented:

`calculateApexMaxBoundaryRatio` *Calculate Apex-Max Boundary Ratio (of a Chromatographic Peak)*

Description

Calculates the Apex-Max Boundary Ratio of the integrated region of a chromatographic peak. The Apex-Max Boundary Ratio is found by taking the ratio of the intensity of the peak apex over the intensity of the maximum of the two boundary intensities.

Usage

```
calculateApexMaxBoundaryRatio(peakData, pts)
```

Arguments

`peakData`	A vector containing characteristic information about a chromatographic peak - including the retention time range
`pts`	A 2D matrix containing the retention time and intensity values of a chromatographic peak

Details

This function repurposed from TargetedMSQC. Toghi Eshghi, S., Auger, P., & Mathews, W. R. (2018). Quality assessment and interference detection in targeted mass spectrometry data using machine learning. Clinical Proteomics, 15. https://doi.org/10.1186/s12014-018-9209-x

Value

The apex-max boundary ratio (double)

Examples

```
# Calculate Apex Max-Boundary Ratio for a peak
data(ex_pts)
data(ex_peakData)
apexMaxBoundary <- calculateApexMaxBoundaryRatio(peakData = ex_peakData, pts = ex_pts)
```

---

calculateElutionShift *Calculate Elution Shift (of a Peak Group)*

Description

Calculate the Elution Shift of each chromatographic peak in a group of samples. For each sample, the Elution Shift is found by calculating the difference between the peak apex (max intensity) of that chromatographic peak and the median peak apex of all samples and normalizing it by the peak base (which is equal to the average difference between the two peak boundaries). The Elution Shift of the Peak Group is equal to the mean of the Elution Shift of each chromatographic peak.

Usage

```
calculateElutionShift(peakDataList, ptsList)
```

Arguments

peakDataList	A list of vectors containing characteristic information about a chromatographic peak - including the retention time range
ptsList	A list of 2D matrices containing the retention time and intensity values of a chromatographic peak

Details

This function repurposed from TargetedMSQC. Toghi Eshghi, S., Auger, P., & Mathews, W. R. (2018). Quality assessment and interference detection in targeted mass spectrometry data using machine learning. Clinical Proteomics, 15. https://doi.org/10.1186/s12014-018-9209-x

Value

The Elution Shift of a Peak Group (double)

Examples

```
# Calculate Elution Shift for each peak
data(ex_ptsList)
data(ex_peakDataList)
elutionShift <- calculateElutionShift(peakDataList = ex_peakDataList, ptsList = ex_ptsList)
```

`calculateEvaluationMeasures` *Calculate Evaluation Measures*

Description

Calculate evaluation measures using the predictions generated during cross-validation.

Usage

```
calculateEvaluationMeasures(pred, true)
```

Arguments

`pred`	factor. A vector of factors that represent predicted classes
`true`	factor. A vector of factors that represent the true classes

Value

A dataframe with the following columns: Model, CVNum, RepNum, Accuracy, PassFScore, PassRecall, PassPrecision, FailFScore, FailRecall, FailPrecision

Examples

```
# Calculate Evaluation Measures for test data
test_evalMeasures <- calculateEvaluationMeasures(pred=test_predictions_class,
pqMetrics_test$Class)
```

`calculateFWHM` *Calculate FWHM2Base (of a Chromatographic Peak)*

Description

Calculates the FWHM2Base of the integrated region of a chromatographic peak. The FWHM2Base is found by determining the peak width at half of the maximum intensity and normalizing this value by the width of the base of the peak.

Usage

```
calculateFWHM(peakData, pts)
```

Arguments

peakData	A vector containing characteristic information about a chromatographic peak - including the retention time range
pts	A 2D matrix containing the retention time and intensity values of a chromatographic peak

Details

This function repurposed from TargetedMSQC. Toghi Eshghi, S., Auger, P., & Mathews, W. R. (2018). Quality assessment and interference detection in targeted mass spectrometry data using machine learning. Clinical Proteomics, 15. https://doi.org/10.1186/s12014-018-9209-x

Value

The FWHM2Base value (double)

Examples

```
# Calculate FWHM2Base for a peak
data(ex_pts)
data(ex_peakData)
fwhm <- calculateFWHM(peakData=ex_peakData, pts=ex_pts)
```

---

calculateGaussianSimilarity

Calculate Gaussian Similarity (of a Chromatographic Peak)

Description

Calculates the Gaussian Similarity of the integrated region of a chromatographic peak. The Gaussian Similarity is found by calculating the dot product of the standard normalized intensity values of a chromatographic peak and the standard normalized intensity values of a Gaussian curve fitted to the intensities of the original curve.

Usage

```
calculateGaussianSimilarity(peakData, pts)
```

Arguments

peakData	A vector containing characteristic information about a chromatographic peak - including the retention time range
pts	A 2D matrix containing the retention time and intensity values of a chromatographic peak

Details

This function repurposed from Zhang et al. For details, see Zhang, W., & Zhao, P. X. (2014). Quality evaluation of extracted ion chromatograms and chromatographic peaks in liquid chromatography/mass spectrometry-based metabolomics data. BMC Bioinformatics, 15(Suppl 11), S5. https://doi.org/10.1186/1471-2105-15-S11-S5

Value

The Gaussian Similarity value (double)

Examples

```
# Calculate Gaussian Similarity for a peak
data(ex_pts)
data(ex_peakData)
gaussianSimilarity <- calculateGaussianSimilarity(peakData = ex_peakData, pts = ex_pts)
```

---

calculateJaggedness *Calculate Jaggedness (of a Chromatographic Peak)*

Description

Calculates the Jaggedness of the integrated region of a chromatographic peak. The Jaggedness is found by determining the fraction of time points the intensity of the peak changes direction - excluding the peak apex and any intensity changes below a flatness factor.

Usage

```
calculateJaggedness(peakData, pts, flatness.factor = 0.05)
```

Arguments

peakData	A vector containing characteristic information about a chromatographic peak - including the retention time range
pts	A 2D matrix containing the retention time and intensity values of a chromatographic peak
flatness.factor	A numeric value between 0 and 1 that allows the user to adjust the sensitivity of the function to noise. This function calculates the difference between each adjacent pair of points; any value found to be less than flatness.factor * maximum intensity is set to 0.

Details

This function repurposed from TargetedMSQC. Toghi Eshghi, S., Auger, P., & Mathews, W. R. (2018). Quality assessment and interference detection in targeted mass spectrometry data using machine learning. Clinical Proteomics, 15. https://doi.org/10.1186/s12014-018-9209-x

Value

The jaggedness of a chromatographic peak (double)

Examples

```
# Calculate Jaggedness for a peak
data(ex_pts)
data(ex_peakData)
jaggedness <- calculateJaggedness(peakData = ex_peakData, pts = ex_pts)
```

`calculateModality` *Calculate Modality (of a Chromatographic Peak)*

Description

Calculates the Modality of the integrated region of a chromatographic peak. The Modaily is found by taking the ratio of the magnitude of the largest drop in intensity (exluding the apex) and the maximum intensity of the peak.

Usage

```
calculateModality(peakData, pts, flatness.factor = 0.05)
```

Arguments

`peakData`	A vector containing characteristic information about a chromatographic peak - including the retention time range
`pts`	A 2D matrix containing the retention time and intensity values of a chromatographic peak
`flatness.factor`	A numeric value between 0 and 1 that allows the user to adjust the sensitivity of the function to noise. This function calculates the difference between each adjacent pair of points; any value found to be less than flatness.factor * maximum intensity is set to 0.

Details

This function repurposed from TargetedMSQC. Toghi Eshghi, S., Auger, P., & Mathews, W. R. (2018). Quality assessment and interference detection in targeted mass spectrometry data using machine learning. Clinical Proteomics, 15. https://doi.org/10.1186/s12014-018-9209-x

Value

The modality of the peak (double)

Examples

```
# Calculate Modality for a peak
data(ex_pts)
data(ex_peakData)
modality <- calculateModality(peakData = ex_peakData, pts = ex_pts)
```

`calculateRetentionTimeConsistency` *Calculate Retention Time Consistency (of a Peak Group)*

Description

Calculates the Retention Time Consistency of each chromatographic peak in a group of samples. For each sample, the Retention Time Consistency is found by calculating the difference between the time at the center of the sample peak and the mean time of all peak centers normalized by the mean time of all the peak centers.

Usage

```
calculateRetentionTimeConsistency(peakDataList, ptsList)
```

Arguments

`peakDataList`	A list of vectors containing characteristic information about a chromatographic peak - including the retention time range
`ptsList`	A list of 2D matrices containing the retention time and intensity values of a chromatographic peak

Details

This function repurposed from TargetedMSQC. Toghi Eshghi, S., Auger, P., & Mathews, W. R. (2018). Quality assessment and interference detection in targeted mass spectrometry data using machine learning. Clinical Proteomics, 15. https://doi.org/10.1186/s12014-018-9209-x

Value

The Retention Time Consistency of a Peak Group (double)

Examples

```
# Calculate Retention Time Consistency for each peak
data(ex_ptsList)
data(ex_peakDataList)
rtc <- calculateRetentionTimeConsistency(peakDataList = ex_peakDataList, ptsList = ex_ptsList)
```

`calculateSharpness` *Calculate Sharpness (of a Chromatographic Peak)*

Description

Calculate Sharpness of the integrated region of a chromatographic peak. The Sharpness is found by determining the sum of the difference between the intensities of each adjacent pair of points on the peak normalized by the intensity of the peak boundaries.

Usage

```
calculateSharpness(peakData, pts)
```

Arguments

`peakData`	A vector containing characteristic information about a chromatographic peak - including the retention time range
`pts`	A 2D matrix containing the retention time and intensity values of a chromatographic peak

Details

This function repurposed from Zhang et al. For details, see Zhang, W., & Zhao, P. X. (2014). Quality evaluation of extracted ion chromatograms and chromatographic peaks in liquid chromatography/mass spectrometry-based metabolomics data. BMC Bioinformatics, 15(Suppl 11), S5. https://doi.org/10.1186/1471-2105-15-S11-S5

Value

The Sharpness value (double)

Examples

```
# Calculate Sharpness for a peak
data(ex_pts)
data(ex_peakData)
sharpness <- calculateSharpness(peakData = ex_peakData, pts = ex_pts)
```

`calculateSymmetry` *Calculate Symmetry (of a Chromatographic Peak)*

Description

Calculates the Symmetry of the integrated region of a chromatographic peak. The Symmetry is found by calcuating the correlation between the left and right halves of the peak.

Usage

```
calculateSymmetry(peakData, pts)
```

Arguments

`peakData`	A vector containing characteristic information about a chromatographic peak - including the retention time range
`pts`	A 2D matrix containing the retention time and intensity values of a chromatographic peak

Details

This function repurposed from TargetedMSQC. Toghi Eshghi, S., Auger, P., & Mathews, W. R. (2018). Quality assessment and interference detection in targeted mass spectrometry data using machine learning. Clinical Proteomics, 15. https://doi.org/10.1186/s12014-018-9209-x

Value

The Symmetry of the peak (double)

Examples

```
# Calculate Symmetry for a peak
data(ex_pts)
data(ex_peakData)
symmetry <- calculateSymmetry(peakData = ex_peakData, pts = ex_pts)
```

`calculateTPASR` *Calcualte Triangle Peak Area Similarity Ratio (TPASR) (of a Chromatographic Peak)*

Description

Calculates the Triangle Peak Area Similarity Ratio (TPASR) of the integrated region of a chromatographic peak. The TPASR is found by calculating the ratio of the difference between the area of a triangle formed by the apex and the two peak boundaries and the integrated area of the peak over the area of the triangle.

Usage

```
calculateTPASR(peakData, pts)
```

Arguments

peakData	A vector containing characteristic information about a chromatographic peak - including the retention time range
pts	A 2D matrix containing the retention time and intensity values of a chromatographic peak

Details

This function repurposed from Zhang et al. For details, see Zhang, W., & Zhao, P. X. (2014). Quality evaluation of extracted ion chromatograms and chromatographic peaks in liquid chromatography/mass spectrometry-based metabolomics data. BMC Bioinformatics, 15(Suppl 11), S5. https://doi.org/10.1186/1471-2105-15-S11-S5

Value

The TPASR value (double)

Examples

```
# Calculate TPASR for a peak
data(ex_pts)
data(ex_peakData)
tpasr <- calculateTPASR(peakData = ex_peakData, pts = ex_pts)
```

---

calculateZigZagIndex *Calculate the Zig-Zag Index (of a Chromatographic Peak)*

Description

Calculates the Zig-Zag Index of the integrated region of a chromatographic peak. The Zig-Zag Index is found by calculating the sum of the slope changes between neighboring points normalized by the average intensity of the peak boundaries.

Usage

```
calculateZigZagIndex(peakData, pts)
```

Arguments

peakData	A vector containing characteristic information about a chromatographic peak - including the retention time range
pts	A 2D matrix containing the retention time and intensity values of a chromatographic peak

Details

This function repurposed from Zhang et al. For details, see Zhang, W., & Zhao, P. X. (2014). Quality evaluation of extracted ion chromatograms and chromatographic peaks in liquid chromatography/mass spectrometry-based metabolomics data. BMC Bioinformatics, 15(Suppl 11), S5. https://doi.org/10.1186/1471-2105-15-S11-S5

Value

The Zig-Zag Index value (double)

Examples

```
# Calculate ZigZag Index for a peak
data(ex_pts)
data(ex_peakData)
zigZagIndex <- calculateZigZagIndex(peakData = ex_peakData, pts = ex_pts)
```

`evalObj-class` *A custom class for storing the chromatographic peak data required by the peak metric functions for each group of samples.*

Description

A custom class for storing the chromatographic peak data required by the peak metric functions for each group of samples.

Slots

`eicPts` A list of 2D matrices containing the retention time and intensity values of each chromatographic peak

`eicPeakData` A list of vectors for each sample in the group containing characteristic information about each chromatographic peak

`eicNos` A numeric vector of the EIC numbers identifying each feature group

---

`ex_peakData` *Example peakData - value input to calculate... functions (except calculateElutionShift and calculateRetentionTimeConsistency)*

Description

An example of the input for the peakData argument for calculate... functions. It represents data from one sample for the peak of interest.

Usage

```
ex_peakData
```

Format

A list containing the following entries: `mz`, `mzmin`, `mzmax`, `rt`, `rtmin`, `rtmax`, `into`, `intb`, `maxo`, `sn`, `sample`, and `is_filled`.

`ex_peakDataList` *Example peakDataList - value input to calculteElutionShift and calculateRetentionTimeConsistency*

Description

An example of the input for the peakDataList argument for calculteElutionShift and calculateRetentionTimeConsistency. Each entry in the list is represents data for a sample for the peak of interest.

Usage

```
ex_peakDataList
```

Format

A list of lists. Each nested list contains the following entries: `mz`, `mzmin`, `mzmax`, `rt`, `rtmin`, `rtmax`, `into`, `intb`, `maxo`, `sn`, `sample`, and `is_filled`.

`ex_pts` *Example pts - value input to caculate... functions (except calculateElutionShift and calculateRetentionTimeConsistency)*

Description

An example of the input for the pts argument for calcualte... functions. It represents rt and intensity data from one sample for peak of interest.

Usage

```
ex_pts
```

Format

A two-column matrix where the first column represents `rt` and the second column represents `intensity`.

ex_ptsList *Example ptsList - value input to calculteElutionShift and calculateRetentionTimeConsistency*

Description

An example of the input for the ptsList argument for calculteElutionShift and calculateRetentionTimeConsistency. Each entry in the list is a two-column matrix consisting of rt and intensity for a sample for the peak of interest.

Usage

```
ex_ptsList
```

Format

A list of two-column matrices (one matrix per sample) where the first column represents `rt` and the second column represents `intensity`.

getBarPlots *Generate Bar Plots for the Seven Evaluation Measures*

Description

Wrapper function for generating bar plots for each classifiers for each of the seven evaluation measures.

Usage

```
getBarPlots(evalMeasuresDF, emNames = "All")
```

Arguments

`evalMeasuresDF`	A dataframe with the following columns: Model, RepNum, PosClass.FScore, PosClass.Recall, PosClass.Precision, NegClass.FScore, NegClass.Recall, NegClass.Precision, and Accuracy. The rows of the dataframe will correspond to the results of a particular model and a particular round of cross-validation.
`emNames`	A list of names of the evaluation measures to visualize. Accepts the following: PosClass.FScore, PosClass.Recall, PosClass.Precision, NegClass.FScore, NegClass.Recall, NegClass.Precision, and Accuracy. Default is "All".

Value

A list of up to seven bar plots (one for each evaluation measure).

Examples

```
# Create a list of bar plots for each evaluation measure
makeBarPlots(evalMeasuresDF = test_evalMeasures)
```

`getEvalObj` *Extract peak data object*

Description

This function extracts, formats, and combines the chromatographic peak data from the objects returned by the getEIC() and fillPeaks() functions from the XCMS package.

Usage

```
getEvalObj(xs, fill)
```

Arguments

`xs`	An xcmsEIC object returned by the getEIC() function from the XCMS package
`fill`	An xcmsSet object with filled in peak groups

Value

An object of class evalObj

Examples

```
# call getEvalObj on test data
# \donttest{eicEval_test <- getEvalObj(xs = xs_test, fill = fill_test)}
```

`getEvaluationMeasures` *Calculate Evaluation Measures*

Description

Calculate evaluation measures using the predictions generated during cross-validation.

Usage

```
getEvaluationMeasures(models, k, repNum)
```

Arguments

models	list. A list of trained models, like that returned by trainClassifiers()
k	integer. Number of folds used in cross-validation
repNum	integer. Number of cross-validation rounds

Value

A dataframe with the following columns: Model, RepNum, Pass_FScore, Pass_Recall, Pass_Precision, Fail_FScore, Fail_Recall, Fail_Precision, Accuracy

Examples

```
# calculate all seven evaluation measures for each model and each round of cross-validation
evalMeasuresDF <- getEvaluationMeasures(models=models, k=5, repNum=10)
```

---

getPeakQualityMetrics *Calculate the 12 Peak Quality Metrics*

Description

Wrapper function for calculating the each of the 12 peak quality metrics for each feature.

Usage

```
getPeakQualityMetrics(eicEvalData, eicLabels_df, flatness.factor = 0.05)
```

Arguments

eicEvalData	An object of class evalObj containing the required chromatographic peak information
eicLabels_df	A dataframe with EICNos in the first column and Labels in the second column
flatness.factor	A numeric value between 0 and 1 that allows the user to adjust the sensitivity of the function to noise. This function calculates the difference between each adjacent pair of points; any value found to be less than flatness.factor * maximum intensity is set to 0.

Value

An Mx14 matrix where M is equal to the number of peaks. There are 14 columns in total, including one column for each of the twelve metrics, one column for EIC numbers, and one column for the class label.

Examples

```
# # calculate peak quality metrics for development dataset
pqMetrics_development <- getPeakQualityMetrics(eicEvalData = eicEval_development,
eicLabels_df = eicLabels_development)
```

`getPredicitons` *Get MetaClean Predictions*

Description

Wrapper function for retrieving predictions from a trained MetaClean classifier and a test dataset. Returns a data frame with class predictions as well as the associated probabilities for each class prediciton.

Usage

```
getPredicitons(model, testData, eicColumn)
```

Arguments

`model`	The train MetaClean model object.
`testData`	dataframe. Rows should correspond to peaks, columns should include peak quality metrics and EIC column only.
`eicColumn`	name of the EIC column

Value

a dataframe with four columns: EIC, Pred_Class, Pred_Prob_Pass, Pred_Prob_Fail

Examples

```
# train classification algorithms
best_model <- getPredictions(model = mc_model,
                                      testData = pqm_test,
                                      eicColumn = "EICNo")
```

pqm_development *Example Peak Quality Metrics Data Frame for Development Dataset.*

Description

Data frame with peaks quality metrics and labels for all of the 500 EICs in the example development dataset.

Usage

```
pqm_development
```

Format

A data frame with 13 variables (EIC Number, the 11 peak quality metrics, and Class Labels): EICNo, ApexBoundaryRatio_mean, ElutionShift_mean, FWHM2Base_mean, Jaggedness_mean, Modelaity_mean, RetentionTimeCorrelation_mean, Symmetry_mean, GaussianSimilarity_mean, Sharpness_mean, TPASR_mean, ZigZag_mean, and Class.

pqm_test *Example Peak Quality Metrics Data Frame for Test Dataset.*

Description

Data frame with peaks quality metrics and labels for all of the 500 EICs in the example test dataset.

Usage

```
pqm_test
```

Format

A data frame with 13 variables (EIC Number, the 11 peak quality metrics, and Class Labels): EICNo, ApexBoundaryRatio_mean, ElutionShift_mean, FWHM2Base_mean, Jaggedness_mean, Modelaity_mean, RetentionTimeCorrelation_mean, Symmetry_mean, GaussianSimilarity_mean, Sharpness_mean, TPASR_mean, ZigZag_mean, and Class.

rsdFilter *RSD Filteirng*

Description

Filters out EICs with RSD

Usage

```
rsdFilter(peakTable, eicColumn, rsdColumns, rsdThreshold = 0.3)
```

Arguments

peakTable	peak table generated by xcms group object
eicColumn	name of the EIC column
rsdColumns	names of the sample columns to be used to calcualte RSD
rsdThreshold	RSD percent threshold for filtering; default 0.3

Value

peakTable with filtered EICs

Examples

```
rsd_filtered_table <- rsdFilter(peakTable = group_table,
                                          eicColumn = eicColumn,
                                          rsdColumns = rsdColumns)
```

runCrossValidation *Run Cross-Validation for A List of Algoirthms with Peak Quality Metric Feature Sets*

Description

Wrapper function for running cross-validation on up to 8 classification algorithms using one or more of the three available metrics sets.

Usage

```
runCrossValidation(
  trainData,
  k,
  repNum,
  rand.seed = NULL,
  models = "all",
  metricSet = "M11"
)
```

Arguments

trainData	dataframe. Rows should correspond to peaks, columns should include peak quality metrics and class labels only.
k	integer. Number of folds to be used in cross-validation
repNum	integer. Number of cross-validation rounds to perform
rand.seed	integer. State in which to set the random number generator
models	character string or vector. Specifies the classification algorithms to be trained from the eight available: DecisionTree, LogisiticRegression, NaiveBayes, RandomForest, SVM_Linear, AdaBoost, NeuralNetwork, and ModelAveragedNeuralNetwork. "all" specifies the use of all models. Default is "all".
metricSet	The metric set(s) to be run with the selected model(s). Select from the following: M4, M7, and M11. Use c() to select multiple metrics. "all" specifics the use of all metrics. Default is "M11".

Value

a list of up to 8 trained models

Examples

```
# train classification algorithms
models <- trainClassifiers(trainData=pqMetrics_development, k=5, repNum=10,
 rand.seed = 453, models="DecisionTree")
```

---

summaryStats *Calculate summary statistics for evaluation measures*

Description

For repeated cross-validation, find the mean and standard error of N rounds for each model.

Usage

```
summaryStats(i, evalMeasuresDF, emNames, modelNames)
```

Arguments

i	An integer representing 1:N where N is the total number of cross-validation rounds.
evalMeasuresDF	A dataframe with the following columns: Model, RepNum, PosClass.FScore, PosClass.Recall, PosClass.Precision, NegClass.FScore, NegClass.Recall, NegClass.Precision, and Accuracy. The rows of the dataframe will correspond to the results of a particular model and a particular round of cross-validation.
emNames	A list of names of the evaluation measures to visualize. Accepts the following: PosClass.FScore, PosClass.Recall, PosClass.Precision, NegClass.FScore, NegClass.Recall, NegClass.Precision, and Accuracy. Default is "All".
modelNames	A list of the models trained.

Value

A dataframe with the following columns: Model, evalMeasure, Mean, and SE (Standard Error).

Examples

```
summaryStatsList <-  lapply(1:numModels, summaryStats,
evalMeasuresDF=evalMeasuresDF, emNames=emNames, modelNames=modelNames)
```

---

trainClassifier *Train MetaClean Classifier*

Description

Wrapper function for training one of the 8 classification algorithms using one of the three available metrics sets.

Usage

```
trainClassifier(trainData, model, metricSet, hyperparameters)
```

Arguments

trainData	dataframe. Rows should correspond to peaks, columns should include peak quality metrics and class labels only.
model	Name of the classification algorithm to be trained from the eight available: DecisionTree, LogisiticRegression, NaiveBayes, RandomForest, SVM_Linear, AdaBoost, NeuralNetwork, and ModelAveragedNeuralNetwork.
metricSet	The metric set to be run with the selected model. Select from the following: M4, M7, and M11.
hyperparameters	dataframe of the tuned hyperparameters returned by runCrossValidation()

Value

a trained MetaClean model

Examples

```
# train classification algorithms
best_model <- trainClassifier(trainData=pqMetrics_development,
                                      model="AdaBoost",
                                      metricSet="M11",
                                      hyperparameters)
```

Index