# Package ‘eNetXplorer’

June 14, 2020

**Type** Package

**Title** Quantitative Exploration of Elastic Net Families for Generalized Linear Models

**Version** 1.1.1

**Date** 2020-06-14

**Author** Julian Candia and John S. Tsang

**Maintainer** Julian Candia <julian.candia@nih.gov>

**Depends** R (>= 2.10)

**biocViews**

**Imports** glmnet, stats, graphics, methods, grDevices, Matrix, progress, survival, survcomp, survivalROC, calibrate, RColorBrewer, gplots, expm

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**Description** Provides a quantitative toolkit to explore elastic net families and to uncover correlates contributing to prediction under a cross-validation framework. Fits linear, binomial (logistic), multinomial and Cox regression models. Candia J and Tsang JS, BMC Bioinformatics (2019) 20:189 <doi:10.1186/s12859-019-2778-5>.

**License** GPL-3

**LazyData** true

**LazyLoad** yes

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2020-06-14 19:00:03 UTC

## R topics documented:

---

`eNetXplorer-package` *explores elastic net families for generalized linear models*

---

## Description

Provides a quantitative toolkit to explore elastic net families and to uncover correlates contributing to prediction under a cross-validation framework. Fits linear, binomial (logistic), multinomial, and Cox regression models.

## Details

| | |
|---|---|
| Package: | eNetXplorer |
| Type: | Package |
| Version: | 1.1.1 |
| Date: | 2020-06-14 |
| License: | GPL-3 |

This package provides a full pipeline of analysis: `eNetXplorer` takes in `x`, `y` data to generate a family of elastic net models over a range of `alpha` values; `summary` generates a summary of results in tabular format; `plot` provides a variety of plots to visualize results; `summaryPDF` generates a report in PDF format; and `export` creates plain text output files for downstream processing.

## Author(s)

Julian Candia and John S. Tsang
Maintainer: Julian Candia <`julian.candia@nih.gov`>

### References

Candia J and Tsang JS. *eNetXplorer: an R package for the quantitative exploration of elastic net families for generalized linear models*, BMC Bioinformatics (2019) 20:189.

### See Also

`eNetXplorer`, `summary`, `plot`, `summaryPDF`, `export`

### Examples

```
data(QuickStartEx)
fit = eNetXplorer(x=QuickStartEx$predictor,y=QuickStartEx$response,
family="gaussian",n_run=20,n_perm_null=10,seed=111)
summary(fit)
plot(x=fit,plot.type="measuredVsOOB",alpha.index=4)
summaryPDF(x=fit,dest_dir=tempdir())
export(x=fit,dest_dir=tempdir())
```

---

## `breastCancerSurv` *gene signature for breast cancer survival*

---

### Description

Gene signature proposed by Desmedt et al associated with breast cancer survival. Microarray data from van't Veer et al and van de Vijver et al. Dataset adapted from Schroeder et al.

### Usage

```
data(breastCancerSurv)
```

### Format

A numerical matrix of predictors is provided with subjects as rows and genes as columns. The response is a two-column matrix with survival time (in days) and status (0=censored, 1=dead).

### Details

Desmedt et al proposed a gene signature associated with breast cancer clinical outcome that captures different biological processes: AURKA (proliferation), PLAU (tumor invasion/metastasis), STAT1 (immune response), VEGFA (angiogenesis), CASP3 (apoptosis), ESR1 (ER signaling) and ERBB2 (HER2 signaling). Microarray data was obtained from van't Veer et al and van de Vijver et al. The dataset was adapted from Schroeder et al. Missing predictor data was imputed using the `missForest` package. Subjects with missing survival data were removed.

### References

Desmedt C et al. *Biological Processes Associated with Breast Cancer Clinical Outcome Depend on the Molecular Subtypes*, Clinical Cancer Research (2008) 14(16):5158-5165.

Schroeder MS et al. *survcomp: an R/Bioconductor package for performance assessment and comparison of survival models*, Bioinformatics (2011) 27(22):3206-3208.

van de Vijver MJ et al. *A Gene Expression Signature as a Predictor of Survival in Breast Cancer*, New England Journal of Medicine (2002) 347(25):1999-2009.

van't Veer LJ et al. *Gene expression profiling predicts clinical outcome of breast cancer*, Nature (2002) 415:530-536.

## `eNetXplorer` *generates family of elastic net models for different alphas*

### Description

Elastic net uses a mixing parameter `alpha` to tune the penalty term continuously from ridge (`alpha=0`) to lasso (`alpha=1`). `eNetXplorer` generates a family of elastic net models over different values of `alpha` for the quantitative exploration of the effects of shrinkage. For each `alpha`, the regularization parameter `lambda` is chosen by optimizing a quality (objective) function based on out-of-bag cross-validation predictions. Statistical significance of each model, as well as that of individual features within a model, is assigned by comparison to a set of null models generated by random permutations of the response. `eNetXplorer` fits linear (gaussian), logistic (binomial), multinomial and Cox regression models.

### Usage

```
eNetXplorer(x, y, family=c("gaussian","binomial","multinomial","cox"),
alpha=seq(0,1,by=0.2), nlambda=100, nlambda.ext=NULL, seed=NULL, scaled=TRUE,
n_fold=5, n_run=100, n_perm_null=25, save_obj=FALSE, dest_dir=getwd(),
dest_dir_create=TRUE, dest_obj="eNet.Robj", QF.FUN=NULL, QF_label=NULL,
cor_method=c("pearson","kendall","spearman"),
binom_method=c("accuracy","precision","recall","Fscore","specificity","auc"),
multinom_method=c("avg accuracy","avg precision","avg recall","avg Fscore"),
binom_pos=NULL, fscore_beta=NULL, fold_distrib_fail.max=100,
cox_index=c("concordance","D-index"), logrank=FALSE, survAUC=FALSE,
survAUC_time=NULL, survAUC_method=c("KM","NNE"), survAUC_lambda=NULL,
survAUC_span=NULL, ...)
```

### Arguments

`x` Input numerical matrix with instances as rows and features as columns. Instance and feature labels should be provided as row and column names, respectively. Can be in sparse matrix format (inherit from class `"sparseMatrix"` as in package `Matrix`). Cannot handle missing values.

y Response variable. For family="gaussian", numerical vector. For family= "binomial", factor with two levels. For family="multinomial", factor with two or more levels. For categorical families, if a vector is supplied, it will be coerced into a factor. For family="cox", matrix with columns named "time" and "status", where the latter is a binary indicator of event (1) or right-censoring (0).

family Response type: "gaussian" (numerical), "binomial" (2-level factor), "multinomial" (factor with >=2 levels) or "cox" (survival time and censoring status).

alpha Sequence of values for the mixing parameter penalty term in the elastic net family. Default is seq(0,1,by=0.2).

nlambda Number of values for the regularization parameter lambda. Default is 100. Irrespective of nlambda, the range of lambda values is assigned by glmnet.

nlambda.ext If set to a value larger than nlambda, this will be the number of values for lambda obtained by extending the range assigned by glmnet symmetrically while keeping the lambda density uniform in log scale. Default is NULL, which will not extend the range of lambda assigned by glmnet.

seed Sets the pseudo-random number seed to enforce reproducibility. Default is NULL.

scaled Z-score transformation of individual features across all instances. Default is TRUE.

n_fold Number of cross-validation folds per run. lambda is chosen based on the maximization of a quality function on out-of-bag-instances averaged over all runs. Default is 5.

n_run Number of runs (i.e. cross-validated model iterations); for each run, instances are randomly assigned to cross-validation folds. Default is 100.

n_perm_null Number of random null-model permutations of the response per run. Default is 25.

save_obj Logical to save the eNetXplorer object. Default is FALSE.

dest_dir Destination directory. Default is the working directory.

dest_dir_create

Creates destination directory if it does not exist already. Default is TRUE.

dest_obj Name for output eNetXplorer object.

QF.FUN User-defined quality (objective) function as maximization criterion to select lambda based on response vs out-of-bag predictions (see example below). If not set, family-specific default quality functions are used, as follows: for family="gaussian", default is correlation; for family="binomial", it is accuracy; for family="multinomial", it is average accuracy; for family="cox", it is the concordance index (default) or D-index (set by cox_index).

QF_label Label for user-defined quality function, if QF.FUN is provided.

cor_method For family="gaussian", correlation method to be used in the default quality function cor.test. Default is "pearson".

binom_method For family="binomial", method to be used in the quality function. Default is "accuracy".

multinom_method
For family="multinomial", method to be used in the quality function. Default is "avg accuracy".

binom_pos
For family="binomial" and quality function methods other than the default ("accuracy"), this is the class to be considered positive. Default is the first level of the response factor.

fscore_beta
For family="binomial" and quality function method "Fscore", or for family= "multinomial" and quality function method "avg Fscore", this is the beta factor to balance precision and recall. Default is 1.

fold_distrib_fail.max
For categorical models, maximum number of failed attempts per run to have all classes represented in each in-bag fold. If this number is exceeded, the execution is halted; try again with larger n_fold, by removing/reasigning classes of small size, and/or with larger fold_distrib_fail.max. Default is 100.

cox_index
For family="cox", index method to be used in the default quality function. Default is "concordance", alternative choice is "D-index".

logrank
For family="cox", logical to generate cross-validated log-rank test p-values of low- vs high-risk groups, defined by the median of out-of-bag predicted risk. Default is FALSE.

survAUC
For family="cox", logical to calculate area-under-curve (AUC) from cross-validated time-dependent ROC curves based on out-of-bag predicted risk. Default is FALSE.

survAUC_time
For family="cox" (if survAUC=T), numerical vector with timepoints of interest; time must be in the same units as the response variable y.

survAUC_method
For family="cox" (if survAUC=T), Kaplan-Meier ("KM") or Nearest-Neighbor Estimation ("NNE") method. Default is "KM".

survAUC_lambda
For family="cox" (if survAUC=T), smoothing parameter for the "NNE" method. Default is NULL.

survAUC_span
For family="cox" (if survAUC=T), span for the "NNE" method. Default is NULL.

...
Accepts parameters from glmnet.control(...) to allow changes of factory default parameters in glmnet. If not explicitly set, it will use factory defaults.

## Details

For each alpha, a set of nlambda values is obtained using the full data; if provided, nlambda.ext allows to extend the range of lambda values symmetrically while keeping its density uniform in log scale. Using these values of lambda, elastic net cross-validation models are generated for n_run random assignments of instances among n_fold folds; the best lambda is determined by the maximization of a quality (objective) function that compares out-of-bag predictions against the response. A variety of quality functions are implemented for each response type, namely: for gaussian models, correlation (different correlation methods available); for binomial models, accuracy, precision, recall, F-score, specificity, area-under-curve; for multinomial models, average accuracy, precision, recall, F-score; for Cox regression models, concordance, D-index (Schroeder et al). Some of these choices require additional parameters: binomial measures that are not invariant under class permutation (see Sokolova & Lapalme) require to specify which class is to be considered positive; F-score requires to specify the value of the beta factor to balance precision and recall (F-score equals precision for beta=0 and tends to recall in the large beta limit). Besides these built-in options, user-defined quality functions can be provided via `QF.FUN`. For each run, using the same assignment of instances into folds, `n_perm_null` null models are generated by shuffling the response. By using the quality function to compare the out-of-bag performance of the model to that of the null models, an empirical significance p-value is assigned to the model. Similar procedures allow to obtain p-values for individual features based on absolute coefficient magnitude and on the frequency of non-zero coefficients. A family of elastic net models is thus generated for multiple values of `alpha` spanning the range from ridge (`alpha=0`) to lasso (`alpha=1`). This function returns an `eNetXplorer` object on which summary, plotting and export functions in this package can be applied for further analysis. For details about the underlying elastic net models (Friedman et al; Zhou & Hastie), refer to the `glmnet` package and references therein. For more details about `eNetXplorer`, see Candia & Tsang.

For Cox regression models, setting `logrank=T` generates cross-validated log-rank test p-values of low- vs high-risk groups, which are defined by the median of out-of-bag predicted risk (Simon et al). Moreover, setting `survAUC=T` and providing a numerical vector `survAUC_time` with timepoints of interest generates the AUC from cross-validated time-dependent ROC curves based on out-of-bag predicted risk (Simon et al) using the `survivalROC` package, which implements Kaplan-Meier and Nearest-Neighbor Estimation methods (Heagerty et al).

## Value

An object with S3 class `"eNetXplorer"`.

| | |
|---|---|
| `predictor` | Predictor matrix used for regression (in sparse matrix format). |
| `response` | Response variable used for regression. |
| `family` | Input parameter. |
| `alpha` | Input parameter. |
| `nlambda` | Input parameter. |
| `nlambda.ext` | Input parameter. |
| `seed` | Input parameter. |
| `scaled` | Input parameter. |
| `n_fold` | Input parameter. |
| `n_run` | Input parameter. |
| `n_perm_null` | Input parameter. |
| `QF_label` | Input parameter. |
| `cor_method` | Input parameter. |
| `binom_method` | Input parameter. |
| `multinom_method` | Input parameter. |
| `binom_pos` | Input parameter. |
| `fscore_beta` | Input parameter. |
| `fold_distrib_fail.max` | Input parameter. |

`cox_index` Input parameter.

`logrank` Input parameter.

`survAUC` Input parameter.

`survAUC_time` Input parameter.

`survAUC_method` Input parameter.

`survAUC_lambda` Input parameter.

`survAUC_span` Input parameter.

`instance` Instance labels.

`feature` Feature labels.

`glmnet_params` `glmnet` parameters used for regression.

`best_lambda` `lambda` values chosen by cross-validation.

`model_QF_est` Quality function values obtained by cross-validation.

`QF_model_vs_null_pval`
P-value from model vs null comparison to assess statistical significance.

`lambda_values` List of `lambda` values used for each `alpha`.

`lambda_QF_est` List of quality function values obtained for each `alpha`.

`predicted_values`
List of out-of-bag predicted values for each `alpha`; rows are instances and columns are median/mad predictions (for linear and Cox regression) or class predictions (for binomial and multinomial regression).

`feature_coef_wmean`
Mean of feature coefficients (over runs) weighted by non-zero frequency (over folds) in sparse matrix format, with features as rows and `alpha` values as columns. For multinomial regression, it is a list of matrices (one matrix for each class).

`feature_coef_wsd`
Standard deviation of feature coefficients (over runs) weighted by non-zero frequency (over folds) in sparse matrix format, with features as rows and `alpha` values as columns. For multinomial regression, it is a list of matrices (one matrix for each class).

`feature_freq_mean`
Mean of non-zero frequency in sparse matrix format, with features as rows and `alpha` values as columns. For multinomial regression, it is a list of matrices (one matrix for each class).

`feature_freq_sd`
Standard deviation of non-zero frequency in sparse matrix format, with features as rows and `alpha` values as columns. For multinomial regression, it is a list of matrices (one matrix for each class).

`null_feature_coef_wmean`
Analogous to `feature_coef_wmean` for null model permutations.

`null_feature_coef_wsd`
Analogous to `feature_coef_wsd` for null model permutations.

`null_feature_freq_mean`
Analogous to `feature_freq_mean` for null model permutations.

`null_feature_freq_sd`
: Analogous to `feature_freq_sd` for null model permutations.

`feature_coef_model_vs_null_pval`
: P-value from model vs null comparison to assess statistical significance of mean non-zero feature coefficients in sparse matrix format, with features as rows and `alpha` values as columns. For multinomial regression, it is a list of matrices (one matrix for each class).

`feature_freq_model_vs_null_pval`
: P-value from model vs null comparison to assess statistical significance of mean non-zero feature frequencies in sparse matrix format, with features as rows and `alpha` values as columns. For multinomial regression, it is a list of matrices (one matrix for each class).

`logrank_pval`
: For Cox regression (if `logrank=T`), cross-validated log-rank test p-value of low- vs high-risk groups, defined by the median of out-of-bag predicted risk.

`AUC_mean`
: For Cox regression (if `survAUC=T`), mean AUC from cross-validated time-dependent ROC curves based on out-of-bag predicted risk, with timepoints (given by `survAUC_time`) as rows and `alpha` values as columns.

`AUC_sd`
: For Cox regression (if `survAUC=T`), standard deviation of AUC.

`AUC_perc05`
: For Cox regression (if `survAUC=T`), 5th percentile of AUC.

`AUC_perc50`
: For Cox regression (if `survAUC=T`), 50th percentile (median) of AUC.

`AUC_perc95`
: For Cox regression (if `survAUC=T`), 95th percentile of AUC.

`AUC_pval`
: For Cox regression (if `survAUC=T`), p-value of AUC from model vs null comparison to assess statistical significance.

## Author(s)

Julian Candia and John S. Tsang
Maintainer: Julian Candia <julian.candia@nih.gov>

## References

Candia J and Tsang JS. *eNetXplorer: an R package for the quantitative exploration of elastic net families for generalized linear models*, BMC Bioinformatics (2019) 20:189.

Friedman J, Hastie T and Tibshirani R. *Regularization paths for generalized linear models via coordinate descent*, Journal of Statistical Software (2010) 33:1-22.

Heagerty PJ, Lumley T and Pepe MS. *Time-dependent ROC Curves for Censored Survival Data and a Diagnostic Marker*, Biometrics (2000) 56, 337-344.

Schroeder MS, Culhane AC, Quackenbush J, Haibe-Kains B. *survcomp: an R/Bioconductor package for performance assessment and comparison of survival models*, Bioinformatics (2011) 27:3206-8.

Simon RM, Subramanian J, Li M-C and Menezes S. *Using cross-validation to evaluate predictive accuracy of survival risk classifiers based on high-dimensional data*, Briefings in Bioinformatics (2011) 12:203-14.

Sokolova M and Lapalme G. *A systematic analysis of performance measures for classification tasks*, Information Processing and Management (2009) 45, 427-437.

Zou H and Hastie T. *Regularization and variable selection via the elastic net*, Journal of the Royal Statistical Society Series B (2005) 67:301-20.

## See Also

summary, plot, summaryPDF, export

## Examples

```
# Linear models (synthetic dataset comprised of 20 features and 75 instances):
data(QuickStartEx)
fit = eNetXplorer(x=QuickStartEx$predictor, y=QuickStartEx$response,
family="gaussian", n_run=20, n_perm_null=10, seed=111)


# Custom QF provided (negative mean squared error)
data(QuickStartEx)
customQF = function(predicted,response){
     -mean((predicted-response)**2)
}
fit = eNetXplorer(x=QuickStartEx$predictor, y=QuickStartEx$response,
family="gaussian", n_run=20, n_perm_null=10, seed=111, QF.FUN=customQF, QF_label="MSE")


# Linear models to predict numerical day-70 H1N1 serum titers based on
# day-7 cell population frequencies:
data(H1N1_Flow)
fit = eNetXplorer(x=H1N1_Flow$predictor_day7, y=H1N1_Flow$response_numer[rownames(
H1N1_Flow$predictor_day7)], family="gaussian", n_run=25, n_perm_null=15, seed=111)


# Binomial models to predict acute myeloid (AML) vs acute lymphoblastic (ALL)
# leukemias:
data(Leukemia_miR)
fit = eNetXplorer(x=Leuk_miR_filt$predictor, y=Leuk_miR_filt$response_binomial,
family="binomial", n_run=25, n_perm_null=15, seed=111)


# Multinomial models to predict acute myeloid (AML), acute B-cell lymphoblastic
# (B-ALL) and acute T-cell lymphoblastic (T-ALL) leukemias:
data(Leukemia_miR)
fit = eNetXplorer(x=Leuk_miR_filt$predictor, y=Leuk_miR_filt$response_multinomial,
family="multinomial", n_run=25, n_perm_null=15, seed=111)


# Binomial models to predict B-ALL vs T-ALL:
data(Leukemia_miR)
fit = eNetXplorer(x=Leuk_miR_filt$predictor[Leuk_miR_filt$response_multinomial!="AML",],
y=Leuk_miR_filt$response_multinomial[Leuk_miR_filt$response_multinomial!="AML"],
family="binomial", n_run=25, n_perm_null=15, seed=111)
```

```
# Cox regression models to predict survival based on 7-gene signature:
data(breastCancerSurv)
fit = eNetXplorer(x=breastCancerSurv$predictor, y=breastCancerSurv$response, family="cox",
n_run=25, n_perm_null=15, seed=111)
```

## export *generates plain text files from eNetXplorer object*

### Description

This function enables the extraction of three different levels of data (input, summary, and detailed output results) from an `eNetXplorer` object. Plain text data files are produced with tab- or comma-separated-value formats.

### Usage

```
export(x, dest_dir=getwd(), dest_dir_create=TRUE, delim=c("tsv","csv"),
input.data=TRUE, summary.data=TRUE, output.data=TRUE)
```

### Arguments

`x` `eNetXplorer` object.

`dest_dir` Destination directory. Default is the working directory.

`dest_dir_create` Creates destination directory if it does not exist already. Default is `TRUE`.

`delim` Delimiter for the generated files, either tab-separated (`"tsv"`) or comma-separated (`"csv"`) values. Default is `"tsv"`.

`input.data` Logical variable indicating whether to generate files containing input data (i.e. data fed into the models and model arguments). Default is `TRUE`.

`summary.data` Logical variable indicating whether to generate a file with summary results from the models. Default is `TRUE`.

`output.data` Logical variable indicating whether to generate files with detailed results from the models. Default is `TRUE`.

### Author(s)

Julian Candia and John S. Tsang
Maintainer: Julian Candia <`julian.candia@nih.gov`>

### See Also

`eNetXplorer`

## Examples

```
data(QuickStartEx)
fit = eNetXplorer(x=QuickStartEx$predictor,y=QuickStartEx$response,
family="gaussian",n_run=20,n_perm_null=10,seed=111)
export(x=fit,dest_dir=tempdir())
```

---

# H1N1_Flow *longitudinal cell population frequencies and titer response upon H1N1 vaccination*

---

## Description

Data from a cohort of healthy subjects vaccinated against influenza virus H1N1. Cell population frequencies from deep-phenotyping flow cytometry were determined longitudinally pre- (days -7, 0) and post-vaccination (days 1, 7, 70). The response is the adjusted maximum fold change (adjMFC) of serum titers at day 70 relative to baseline, as defined in Tsang et al.

## Usage

```
data(H1N1_Flow)
```

## Format

For each timepoint (days -7, 0, 1, 7, 70), a numerical matrix of predictors is provided with subjects as rows and cell populations as columns. Two versions of the serum titer response are given: `response_numer` as a numerical vector and `response_class` as a categorical vector discretized into low (0), intermediate (1) and high (2) response classes. A metadata file with cell population annotations is also provided.

## Details

Cell populations were manually gated and expressed as percent of parent. Samples and cell populations were filtered independently for each timepoint. Samples filter: excluded if median of viable cells fraction across all 5 tubes was <0.7. Cell population filter: excluded if >80% of samples had <20 cells. Data adjustment: data were log10-transformed and pooled across all timepoints, then adjusted for age, gender and ethnicity effects. For more details, see Tsang et al.

## References

Tsang JS et al. *Global Analyses of Human Immune Variation Reveal Baseline Predictors of Post-vaccination Responses*, Cell (2014) 157: 499-513.

## `Leukemia_miR` *microRNA expression of acute leukemia phenotypes*

### Description

Data of human microRNA (miR) expression of 847 miRs from 80 acute myeloid (AML) and acute lymphoblastic (ALL) leukemia cell lines, 60 primary (patient) samples, and 50 normal control samples sorted by cell type (CD34+ HSPC, Granulocytes, Monocytes, T-cells and B-cells). Acute lymphoblastic leukemia samples are further classified by B-cell (B-ALL) and T-cell (T-ALL) subphenotypes.

### Usage

```
data(Leukemia_miR)
```

### Details

Two dataset versions are provided: the full dataset `Leuk_miR_full` (190 samples x 847 miRs) and the filtered dataset `Leuk_miR_filt` (140 samples x 370 miRs). Data available at GEO under Accession Number GSE51908.

### References

Tan YS et al. *Regulation of RAB5C is important for the growth inhibitory effects of MiR-509 in human precursor-B acute lymphoblastic leukemia*, PLoS One (2014) 9:e111777.

Candia J et al. *Uncovering low-dimensional, miR-based signatures of acute myeloid and lymphoblastic leukemias with a machine-learning-driven network approach*, Converg Sci Phys Oncol (2015) 1:025002.

## `Leuk_miR_filt` *microRNA expression of acute leukemia phenotypes (filtered dataset)*

### Description

Data of human microRNA (miR) expression of 370 miRs from 80 acute myeloid (AML) and acute lymphoblastic (ALL) leukemia cell lines and 60 primary (patient) samples. Acute lymphoblastic leukemia samples are further classified by B-cell (B-ALL) and T-cell (T-ALL) subphenotypes.

### Format

The filtered dataset `Leuk_miR_filt` consists of a numerical matrix of predictors (with samples as rows and miRs as columns) and two categorical response vectors provided for binomial (AML, ALL) and multinomial (AML, B-ALL, T-ALL) classification.

### Details

The filtered dataset Leuk_miR_filt is a subset of the full dataset Leuk_miR_full, which includes only miRs with median expression >1.2 across all samples. Only leukemia samples (cell lines and primary) were kept. Data available at GEO under Accession Number GSE51908.

### References

Tan YS et al. *Regulation of RAB5C is important for the growth inhibitory effects of MiR-509 in human precursor-B acute lymphoblastic leukemia*, PLoS One (2014) 9:e111777.

Candia J et al. *Uncovering low-dimensional, miR-based signatures of acute myeloid and lymphoblastic leukemias with a machine-learning-driven network approach*, Converg Sci Phys Oncol (2015) 1:025002.

### Examples

```
# Multinomial models to predict acute myeloid (AML), acute B-cell lymphoblastic
# (B-ALL) and acute T-cell lymphoblastic (T-ALL) leukemias:
data(Leukemia_miR)
fit = eNetXplorer(x=Leuk_miR_filt$predictor, y=Leuk_miR_filt$response_multinomial,
family="multinomial", n_run=25, n_perm_null=15, seed=111)
```

## Leuk_miR_full *microRNA expression of acute leukemia phenotypes (full dataset)*

### Description

Data of human microRNA (miR) expression of 847 miRs from 80 acute myeloid (AML) and acute lymphoblastic (ALL) leukemia cell lines, 60 primary (patient) samples, and 50 normal control samples sorted by cell type (CD34+ HSPC, Granulocytes, Monocytes, T-cells and B-cells). Acute lymphoblastic leukemia samples are further classified by B-cell (B-ALL) and T-cell (T-ALL) subphenotypes.

### Format

The full dataset Leuk_miR_full consists of a numerical matrix of expression (with samples as rows and miRs as columns) and two data frames with sample and miR metadata.

### Details

Data available at GEO under Accession Number GSE51908.

## References

Tan YS et al. *Regulation of RAB5C is important for the growth inhibitory effects of MiR-509 in human precursor-B acute lymphoblastic leukemia*, PLoS One (2014) 9:e111777.

Candia J et al. *Uncovering low-dimensional, miR-based signatures of acute myeloid and lymphoblastic leukemias with a machine-learning-driven network approach*, Converg Sci Phys Oncol (2015) 1:025002.

## Examples

```
# Multinomial models to predict acute myeloid (AML), acute B-cell lymphoblastic
# (B-ALL) and acute T-cell lymphoblastic (T-ALL) leukemias:
data(Leukemia_miR)
predictor = Leuk_miR_full$expression_matrix
rownames(predictor) = Leuk_miR_full$sample_metadata$sample
colnames(predictor) = Leuk_miR_full$miRNA_short
response = Leuk_miR_full$sample_metadata$sample_class
fit = eNetXplorer(x=predictor, y=response,
family="multinomial", n_run=25, n_perm_null=15, seed=111)
```

---

# `plot` *generates plots from eNetXplorer object*

---

## Description

This function is a wrapper for a variety of plots, namely:

`summary`: model performance across `alpha` (to assess the relative performance among different member models in the elastic net family, as well as in relation to permutation null models);

`lambdaVsQF`: given `alpha`, quality function across `lambda` (to examine the selection of the optimal penalty parameter);

`measuredVsOOB`: (for gaussian and categorical models) given `alpha`, response vs out-of-bag predictions across instances (to assess individual instances, examine outliers, etc);

`contingency`: (for categorical models) given `alpha`, response vs out-of-bag predictions across classes;

`featureCaterpillar`: given `alpha`, caterpillar plot of feature statistics compared to permutation null models (with statistical significance annotations for individual features);

`featureHeatmap`: heatmap of feature statistics across `alpha` (including statistical significance annotations for individual features);

`KaplanMeier`: (for Cox regression models) given `alpha`, Kaplan-Meier plot of survival probability as a function of time (where the cohort is partitioned in two or more groups based on predicted risk); and

`survROC`: (for Cox regression models) given `alpha`, time-dependent ROC plot(s) based on predicted risk at the specified timepoints of interest.

## Usage

```
## S3 method for class 'eNetXplorer'
plot(x, plot.type=c("summary","lambdaVsQF","measuredVsOOB","contingency",
"featureCaterpillar","featureHeatmap","KaplanMeier","survROC"), alpha.index,
stat=c("freq","coef"), ...)
```

## Arguments

x
: eNetXplorer object.

plot.type
: Type of plot to be produced. Available plots are "summary", "lambdaVsQF", "measuredVsOOB" (gaussian and categorical models only), "contingency" (categorical models only), "featureCaterpillar", "featureHeatmap", "KaplanMeier" (Cox models only) and "survROC" (Cox models only).

alpha.index
: Integer indices to select alpha values. Default is 1:length(alpha)

stat
: Feature statistic: "freq" for nonzero frequency, "coef" for mean nonzero coefficient. Used for plot types "featureHeatmap" and "featureCaterpillar", ignored otherwise.

...
: Additional plotting parameters.

## Author(s)

Julian Candia and John S. Tsang
Maintainer: Julian Candia <julian.candia@nih.gov>

## See Also

eNetXplorer, plotSummary, plotLambdaVsQF, plotMeasuredVsOOB, plotContingency, plotFeatureCaterpillar, plotFeatureHeatmap, plotKaplanMeier, plotSurvROC.

## Examples

```
data(QuickStartEx)
fit = eNetXplorer(x=QuickStartEx$predictor, y=QuickStartEx$response,
family="gaussian", n_run=20, n_perm_null=10, seed=111)
plot(x=fit,plot.type="summary")
plot(x=fit,plot.type="lambdaVsQF",alpha.index=2)
plot(x=fit,plot.type="measuredVsOOB",alpha.index=c(1,3,5))
plot(x=fit,plot.type="featureCaterpillar",stat="coef")
plot(x=fit,plot.type="featureHeatmap",stat="freq")
```

## plotContingency *generates plot of response vs out-of-bag predictions across classes*

### Description

For categorical models, this function generates a graphical representation of the true vs predicted contingency matrix across classes for a given alpha.

### Usage

```
plotContingency(x, alpha.index=NULL, xlab="class (true)", ylab="class (predicted)",
cex.lab=0.95, main=NULL, col.main="black", cex.main=0.85, cex.axis=1,
symbol.size.inches=0.5, bg.color="steelblue2", fg.color=NULL, margin=0.2,
frequency.label=TRUE, frequency.label.cex=1, frequency.label.offset=0, ...)
```

### Arguments

| | |
|---|---|
| x | eNetXplorer object. |
| alpha.index | Integer indices to select alpha values. Default is 1:length(alpha) |
| xlab | Custom x-axis label. |
| ylab | Custom y-axis label. |
| cex.lab | Axis label size. |
| main | Custom title. |
| col.main | Title color. |
| cex.main | Title size. |
| cex.axis | Axis size. |
| symbol.size.inches | Symbol size. |
| bg.color | Symbol color. |
| fg.color | Color of symbol background. |
| margin | Margin size to accomodate symbols. |
| frequency.label | Logical to display class frequency labels. Default is TRUE. |
| frequency.label.cex | Size of class frequency labels. |
| frequency.label.offset | Offset of class frequency labels. |
| ... | Additional plotting parameters. |

### Author(s)

Julian Candia and John S. Tsang
Maintainer: Julian Candia <julian.candia@nih.gov>

## See Also

eNetXplorer, plot

## Examples

```
data(QuickStartEx)
binarized=rep("low",length(QuickStartEx$response))
binarized[QuickStartEx$response>median(QuickStartEx$response)]="high"
fit = eNetXplorer(x=QuickStartEx$predictor,y=binarized,family="binomial",n_run=20,
n_perm_null=10,seed=111)
plot(x=fit,plot.type="contingency")
plotContingency(x=fit,alpha.index=6)
```

---

# plotFeatureCaterpillar

*generates caterpillar plot of feature statistics*

---

## Description

Given alpha, this function generates a caterpillar plot of feature statistics compared to permutation null models, which includes statistical significance annotations for individual features. By default, features are selected (and ordered top-down) by statistical significance; options are provided to customize feature selection and display.

## Usage

```
plotFeatureCaterpillar(x, alpha.index=NULL, stat=c("freq","coef"),
feature.all=FALSE, feature.pval.thres=NULL, feature.set=NULL, feature.top.n=25,
signif.code=TRUE, xlab=NULL, ylab=NULL, main=NULL, col.main="black",
cex.main=0.85, line=1.5, subtitle=NULL, col.subtitle="darkgray",
line.subtitle=0.5, cex.subtitle=0.55, cexRow=NULL, cex.lab=0.95, legend=TRUE, ...)
```

## Arguments

| | |
|---|---|
| x | eNetXplorer object. |
| alpha.index | Integer indices to select alpha values. Default is 1:length(alpha) |
| stat | Feature statistic: "freq" for nonzero frequency, "coef" for mean nonzero coefficient. |
| feature.all | (Feature selection argument 1) Logical to show all features. Default is FALSE. |
| feature.pval.thres | (Feature selection argument 2) P-value threshold to select features. Default is NULL. |
| feature.set | (Feature selection argument 3) Character vector of feature names to display. Default is NULL. |

`feature.top.n` (Feature selection argument 4) Number of top features (ordered by p-value) to display. Default is 25.

`signif.code` Logical to display significance annotations. Default is `TRUE`.

`xlab` Label for x axis.

`ylab` Label for y axis.

`main` Custom title.

`col.main` Title color.

`cex.main` Title size.

`line` Title location.

`subtitle` Custom subtitle.

`col.subtitle` Subtitle color.

`line.subtitle` Subtitle location.

`cex.subtitle` Subtitle size.

`cexRow` Size of row labels.

`cex.lab` Axis label size.

`legend` Logical to display legend. Default is `TRUE`.

`...` Additional plotting parameters.

## Details

Feature selection criteria are hierarchical based on arguments 1 through 4 (see argument description above), with argument 1 at the top of the hierarchy. E.g. if `feature.all` is explicitly set to `TRUE`, it will take precedence over any other feature selection argument. By default, the top 25 features are displayed, ordered top-down by significance based on the given value of `alpha`.

## Author(s)

Julian Candia and John S. Tsang
Maintainer: Julian Candia <`julian.candia@nih.gov`>

## See Also

`eNetXplorer`, `plot`

## Examples

```
data(QuickStartEx)
fit = eNetXplorer(x=QuickStartEx$predictor, y=QuickStartEx$response,
family="gaussian", n_run=20, n_perm_null=10, seed=111)
plot(x=fit,plot.type="featureCaterpillar",stat="coef")
plotFeatureCaterpillar(x=fit,alpha.index=3,stat="coef",main="custom title")
```

## plotFeatureHeatmap *generates heatmap plot of feature statistics*

### Description

This function generates a heatmap plot of feature statistics across `alpha`, which includes statistical significance annotations for individual features. By default, features are selected (and ordered top-down) by statistical significance based on a given value of `alpha`; options are provided to customize feature selection and display.

### Usage

```
plotFeatureHeatmap(x, alpha.index=NULL, stat=c("freq","coef"), feature.all=FALSE,
feature.pval.thres=NULL, feature.set=NULL, feature.top.n=25, signif.code=TRUE,
xlab=NULL, ylab=NULL, main=NULL, col.main="black", cex.main=0.95, line=1,
col=NULL, breaks=NULL, scale="none", Rowv=FALSE, Colv=FALSE, na.color=NULL,
cexRow=NULL, srtRow=0, cexCol=0.75, srtCol=45, margins=c(5, 5), key=TRUE,
key.title=NA, dendogram="none", trace="none", notecol.freq="black",
notecol.coef="white", notecex=1, subtitle1=NULL, col.subtitle1="black",
line.subtitle1=-1, cex.subtitle1=0.65, subtitle2=NULL, col.subtitle2="darkgray",
line.subtitle2=-2, cex.subtitle2=0.55, ...)
```

### Arguments

`x` eNetXplorer object.

`alpha.index` Integer indices to select `alpha` values. Default is `1:length(alpha)`

`stat` Feature statistic: `"freq"` for nonzero frequency, `"coef"` for mean nonzero coefficient.

`feature.all` (Feature selection argument 1) Logical to show all features. Default is `FALSE`.

`feature.pval.thres` (Feature selection argument 2) P-value threshold to select features. Default is `NULL`.

`feature.set` (Feature selection argument 3) Character vector of feature names to display. Default is `NULL`.

`feature.top.n` (Feature selection argument 4) Number of top features (ordered by p-value) to display. Default is 25.

`signif.code` Logical to display statistical significance annotations. Default is `TRUE`.

`xlab` Label for x axis.

`ylab` Label for y axis.

`main` Custom title.

`col.main` Title color.

`cex.main` Title size.

`line` Title location.

`col` Heatmap color vector. Length must be one less than number of breaks.

`breaks` Color breaks vector. Default number of breaks is 10.

`scale` Logical to scale the data for heatmap in either the row or column direction. Default is `"none"`.

`Rowv` Logical to reorder rows by hierarchical clustering. Default is `FALSE`.

`Colv` Logical to reorder columns by hierarchical clustering. Default is `FALSE`.

`na.color` Color to use for missing values.

`cexRow` Size of row labels.

`srtRow` Angle of row labels, in degrees from horizontal.

`cexCol` Size of column labels.

`srtCol` Angle of column labels, in degrees from horizontal.

`margins` Numeric vector of length 2 containing the margins for column and row names, respectively.

`key` Logical to display key. Default is `TRUE`.

`key.title` Main title of the color key.

`dendogram` To draw dendograms. Default is `"none"`.

`trace` To display trace lines. Default is `"none"`.

`notecol.freq` Color of statistical significance annotations for feature frequency heatmaps.

`notecol.coef` Color of statistical significance annotations for feature coefficient heatmaps.

`notecex` Size of significance annotations.

`subtitle1` Custom subtitle 1.

`col.subtitle1` Color of subtitle 1.

`line.subtitle1` Position of subtitle 1.

`cex.subtitle1` Size of subtitle 1.

`subtitle2` Custom subtitle 2.

`col.subtitle2` Color of subtitle 2.

`line.subtitle2` Position of subtitle 2.

`cex.subtitle2` Size of subtitle 2.

`...` Additional plotting parameters.

## Details

Feature selection criteria are hierarchical based on arguments 1 through 4 (see argument description above), with argument 1 at the top of the hierarchy. E.g. if `feature.all` is explicitly set to `TRUE`, it will take precedence over any other feature selection argument. By default, the top 25 features are displayed, ordered top-down by significance based on the given value of `alpha`.

## Author(s)

Julian Candia and John S. Tsang
Maintainer: Julian Candia <`julian.candia@nih.gov`>

### See Also

eNetXplorer, plot

### Examples

```
data(QuickStartEx)
fit = eNetXplorer(x=QuickStartEx$predictor,y=QuickStartEx$response,
family="gaussian",n_run=20,n_perm_null=10,seed=111)
plot(x=fit,plot.type="featureHeatmap",stat="freq")
plotFeatureHeatmap(x=fit,alpha.index=3,stat="freq",main="custom title")
```

## plotKaplanMeier *generates Kaplan-Meier plot of survival probability as a function of time*

### Description

For Cox regression models, this function generates a Kaplan-Meier plot of survival probability as a function of time for a given alpha. The default behavior is to partition the cohort in two groups by the predicted risk median, but custom partitions in two or more groups (specified by a vector of predicted risk percentiles) are also possible. In the former case, provided that the eNetXplorer object was generated with the logrank=TRUE argument, the corresponding cross-validated log-rank test p-value is displayed in the default title.

### Usage

```
plotKaplanMeier(x, alpha.index=NULL, xlab="Time", ylab="Probability of Survival",
cex.lab=1, main=NULL, col.main="black", cex.main=0.95, conf.int=TRUE,
breaks_ptiles=NULL, risk.col=NULL, legend=TRUE, legend.cex=0.75, ...)
```

### Arguments

`x` eNetXplorer object (must be family="cox").

`alpha.index` Integer indices to select alpha values. Default is 1:length(alpha)

`xlab` Custom x-axis label.

`ylab` Custom y-axis label.

`cex.lab` Axis label size.

`main` Custom title.

`col.main` Title color.

`cex.main` Title size.

`conf.int` Logical to display 95% confidence intervals. Default is TRUE.

| | |
|---|---|
| `breaks_ptiles` | Vector of percentiles (in 0-1 range) to partition the cohort based on predicted risk. Default is 0.5. |
| `risk.col` | Vector of colors to display the predicted risk-based subcohorts. |
| `legend` | Logical to display legend. Default is `TRUE`. |
| `legend.cex` | Legend size. |
| `...` | Additional plotting parameters. |

### Author(s)

Julian Candia and John S. Tsang
Maintainer: Julian Candia <`julian.candia@nih.gov`>

### See Also

`eNetXplorer`, `plot`

### Examples

```
data(breastCancerSurv)
fit = eNetXplorer(x=breastCancerSurv$predictor, y=breastCancerSurv$response, family="cox",
n_run=25, n_perm_null=15, seed=111, logrank=TRUE)
plot(x=fit, plot.type="KaplanMeier")
plotKaplanMeier(x=fit, alpha.index=6, conf.int=FALSE, breaks_ptiles=c(0.333,0.667))
```

---

## `plotLambdaVsQF` *generates plot of quality function across* `lambda`

---

### Description

Given `alpha`, this function generates a plot of the quality (objective) function across `lambda`, which is useful to examine how was the "best `lambda`" value selected.

### Usage

```
plotLambdaVsQF(x, alpha.index=NULL, xlab="lambda",
ylab="QF (response vs out-of-bag predicted)", cex.lab=0.95, main=NULL,
col.main="black", cex.main=0.95, log="x", type="b", ...)
```

### Arguments

| | |
|---|---|
| `x` | `eNetXplorer` object. |
| `alpha.index` | Integer indices to select `alpha` values. Default is `1:length(alpha)` |
| `xlab` | Custom x-axis label. |
| `ylab` | Custom y-axis label. |

| | |
|---|---|
| `cex.lab` | Axis label size. |
| `main` | Custom title. |
| `col.main` | Title color. |
| `cex.main` | Title size. |
| `log` | Log scale axis. |
| `type` | Plot type. |
| `...` | Additional plotting parameters. |

## Details

By definition, the "best `lambda`" value for a given `alpha` is the one that maximizes the quality function (QF) over the range of `lambda` values considered. Therefore, QF vs `lambda` distributions with sharp, narrow, well-defined peaks provide more confidence in the selection of the optimal `lambda` value than those with less-defined peaks. Sometimes, and particularly for the ridge (`alpha`=0) solutions, QF is observed to increase or decrease monotonically with `lambda` over its entire range, causing a boundary `lambda` value to be selected; we conservatively recommend to disregard `alpha`-models generated under such circumstances. If interested in investigating further, we suggest to re-run those `alpha`-models by extending the default range of `lambda` values (via the argument `nlambda.ext`) or its density (via the argument `nlambda`). On occasion, the range of `lambda` values is effectively limited by convergence issues of the underlying `glmnet` model; in such scenario, we recommend to increment the maximum allowed number of iterations (via the argument `mxit`, which is passed on to `glmnet.control`) or to limit the complexity of the model (e.g. by filtering and reducing the number of features fed into `eNetXplorer`).

## Author(s)

Julian Candia and John S. Tsang
Maintainer: Julian Candia <`julian.candia@nih.gov`>

## See Also

`eNetXplorer`, `plot`

## Examples

```
data(QuickStartEx)
fit = eNetXplorer(x=QuickStartEx$predictor,y=QuickStartEx$response,
family="gaussian",n_run=20,n_perm_null=10,seed=111)
plot(x=fit,plot.type="lambdaVsQF")
plotLambdaVsQF(x=fit,alpha.index=c(1,3),main="custom title",col.main="red")
```

## `plotMeasuredVsOOB` *generates plot of response vs out-of-bag predictions across instances*

### Description

Given `alpha`, this function generates plots of response vs out-of-bag predictions across instances, which can be used to assess individual instances, examine outliers, etc. For linear regression models, it generates a response vs out-of-bag prediction scatterplot; it also displays the best linear fit and its 95% confidence level region. For categorical models, it generates a boxplot across classes showing the frequency of out-of-bag correct predictions.

### Usage

```
plotMeasuredVsOOB(x, alpha.index=NULL, xlab=NULL, ylab=NULL,
cex.lab=0.95, main=NULL, col.main="black", cex.main=0.85, instance.label=TRUE,
instance.label.cex=NULL, instance.label.offset=NULL,
instance.label.added.margin=NULL, col=NULL, transparency=NULL, jitter=NULL,
cex.pt=NULL, class.color=NULL, ...)
```

### Arguments

| | |
|---|---|
| `x` | `eNetXplorer` object. |
| `alpha.index` | Integer indices to select `alpha` values. Default is `1:length(alpha)` |
| `xlab` | Custom x-axis label. |
| `ylab` | Custom y-axis label. |
| `cex.lab` | Axis label size. |
| `main` | Custom title. |
| `col.main` | Title color. |
| `cex.main` | Title size. |
| `instance.label` | Logical to display instance labels. Default is `TRUE`. |
| `instance.label.cex` | Size of instance labels. |
| `instance.label.offset` | Offset of instance labels. |
| `instance.label.added.margin` | (linear regression only) Margin size to accomodate instance label display. |
| `col` | (linear regression only) Symbol color. |
| `transparency` | (categorical models only) Symbol transparency. |
| `jitter` | (categorical models only) Symbol jitter. |
| `cex.pt` | (categorical models only) Symbol size. |
| `class.color` | (categorical models only) Vector of class colors. |
| `...` | Additional plotting parameters. |

### Author(s)

Julian Candia and John S. Tsang
Maintainer: Julian Candia <julian.candia@nih.gov>

### See Also

eNetXplorer, plot

### Examples

```
data(QuickStartEx)
fit = eNetXplorer(x=QuickStartEx$predictor,y=QuickStartEx$response,
family="gaussian",n_run=20,n_perm_null=10,seed=111)
plot(x=fit,plot.type="measuredVsOOB")
plotMeasuredVsOOB(x=fit,alpha.index=2)


data(QuickStartEx)
binarized=rep("low",length(QuickStartEx$response))
binarized[QuickStartEx$response>median(QuickStartEx$response)]="high"
fit = eNetXplorer(x=QuickStartEx$predictor,y=binarized,family="binomial",n_run=20,
n_perm_null=10,seed=111)
plot(x=fit,plot.type="measuredVsOOB")
plotMeasuredVsOOB(x=fit,alpha.index=2)
```

## plotSummary *generates summary plots of model performance across alpha*

### Description

This function generates summary plots to display the performance of all models in the elastic net family. Two measures are used: 1) mean quality function of response vs out-of-bag predictions, and 2) model vs null p-values. Taken together, these plots enable visual assessments of the relative performance among different member models in the elastic net family, as well as in relation to permutation null models.

### Usage

```
plotSummary(x, show.pval.ref=TRUE, main=NULL, col.main="black", cex.main=0.95,
line=1, ...)
```

### Arguments

| | |
|---|---|
| x | eNetXplorer object. |
| show.pval.ref | Logical to display reference lines of significance (if within range of model vs null p-values). Default is TRUE. |

| | |
|---|---|
| `main` | Custom title. |
| `col.main` | Title color. |
| `cex.main` | Title size. |
| `line` | Title position. |
| `...` | Additional plotting parameters. |

### Author(s)

Julian Candia and John S. Tsang
Maintainer: Julian Candia <`julian.candia@nih.gov`>

### See Also

`eNetXplorer`, `plot`

### Examples

```
data(QuickStartEx)
fit = eNetXplorer(x=QuickStartEx$predictor,y=QuickStartEx$response,
family="gaussian",n_run=20,n_perm_null=10,seed=111)
plot(x=fit, plot.type="summary")
plotSummary(x=fit,show.pval.ref=FALSE)
```

---

## `plotSurvROC` *generates time-dependent ROC plots from Cox predicted risks*

---

### Description

For Cox regression models, this function generates time-dependent ROC plot(s) (true positive rate vs false positive rate) for a given `alpha` at the timepoint(s) provided based on median predicted risk. Provided that the `eNetXplorer` object was generated with `survAUC=T`, the cross-validated median AUC and 95% CI are shown in the default title. For more details, see Heagerty et al and package `survivalROC`.

### Usage

```
plotSurvROC(x, alpha.index=NULL, survAUC_time, survAUC_method=c("KM","NNE"),
survAUC_lambda=NULL, survAUC_span=NULL, xlab="False positive rate (1 - Specificity)",
ylab="True positive rate (Sensitivity)", cex.lab=1, main=NULL, col.main="black",
cex.main=0.95, status0="censored", status1="event", ...)
```

## Arguments

| | |
|---|---|
| `x` | `eNetXplorer` object (must be `family="cox"`). |
| `alpha.index` | Integer indices to select alpha values. Default is `1:length(alpha)` |
| `survAUC_time` | Timepoint(s) of interest. Must be in the same time units as the survival time provided to build the `eNetXplorer` object. |
| `survAUC_method` | Kaplan-Meier (`"KM"`) or Nearest-Neighbor Estimation (`"NNE"`). Default is `"KM"`. |
| `survAUC_lambda` | Smoothing parameter for the `"NNE"` method. |
| `survAUC_span` | Span for the `"NNE"` method. |
| `xlab` | Custom x-axis label. |
| `ylab` | Custom y-axis label. |
| `cex.lab` | Axis label size. |
| `main` | Custom title. |
| `col.main` | Title color. |
| `cex.main` | Title size. |
| `status0` | Title label for censoring ("status"=0). |
| `status1` | Title label for events ("status"=1). |
| `...` | Additional parameters. |

## Author(s)

Julian Candia and John S. Tsang
Maintainer: Julian Candia <`julian.candia@nih.gov`>

## References

Heagerty PJ, Lumley T and Pepe MS. *Time-dependent ROC Curves for Censored Survival Data and a Diagnostic Marker*, Biometrics (2000) 56, 337-344.

## See Also

`eNetXplorer`, `plot`

## Examples

```
data(breastCancerSurv)
fit = eNetXplorer(x=breastCancerSurv$predictor, y=breastCancerSurv$response, family="cox",
n_run=25, n_perm_null=15, seed=111, survAUC=TRUE, survAUC_time=c(1,5)*365)
plot(x=fit, plot.type="survROC", survAUC_time=c(1,5)*365, status0="alive", status1="dead")
```

## `QuickStartEx` *synthetic dataset*

### Description

75 instances with 20 predictors and a numerical response to be used as a quick start example.

### Usage

```
data(QuickStartEx)
```

### Format

A numerical matrix of predictors is provided with instances as rows and predictors as columns. A numerical response is provided as a quick start example for linear regression models; it can be easily discretized to serve as example for binary and multinomial models as well.

## `summary` *generates list of model statistics*

### Description

This function generates a standard list of model statistics. For each `alpha`, it contains the best value of `lambda` (obtained by maximizing a quality function over out-of-bag instances), the corresponding maximum value of the quality function, and the model significance (p-value based on comparison to permutation null models).

### Usage

```
## S3 method for class 'eNetXplorer'
summary(object, ...)
```

### Arguments

| | |
|---|---|
| `object` | `eNetXplorer` object. |
| `...` | Additional parameters. |

### Value

| | |
|---|---|
| `alpha` | Vector of `alpha` values. |
| `best_lambda` | Best `lambda` obtained by maximization of the quality function. |
| `model_QF_est` | Maximum of the quality function. |
| `QF_model_vs_null_pval` | P-value from model vs null comparison to assess statistical significance. |

### Author(s)

Julian Candia and John S. Tsang
Maintainer: Julian Candia <julian.candia@nih.gov>

### See Also

eNetXplorer

### Examples

```
data(QuickStartEx)
fit = eNetXplorer(x=QuickStartEx$predictor,y=QuickStartEx$response,
family="gaussian",n_run=20,n_perm_null=10,seed=111)
summary(fit)
```

---

## summaryPDF *generates PDF report with summary of main results*

---

### Description

This function generates a PDF report that contains a plot of model performance across the alpha range, followed by plots showing detailed results for each value of alpha.

### Usage

```
summaryPDF(x, dest_dir=getwd(), dest_dir_create=TRUE, dest_file="eNetSummary.pdf")
```

### Arguments

x eNetXplorer object.

dest_dir Destination directory. Default is the working directory.

dest_dir_create
Creates destination directory if it does not exist already. Default is TRUE.

dest_file Name for output PDF file.

### Author(s)

Julian Candia and John S. Tsang
Maintainer: Julian Candia <julian.candia@nih.gov>

### See Also

eNetXplorer, plot

## Examples

```
data(QuickStartEx)
fit = eNetXplorer(x=QuickStartEx$predictor,y=QuickStartEx$response,
family="gaussian",n_run=20,n_perm_null=10,seed=111)
summaryPDF(x=fit,dest_dir=tempdir())
```

# Index