

Package ‘FREEtree’

July 21, 2025

Type Package

Title Tree Method for High Dimensional Longitudinal Data

Version 0.1.0

Description This tree-based method deals with high dimensional longitudinal data with correlated features through the use of a piecewise random effect model. FREE tree also exploits the network structure of the features, by first clustering them using Weighted Gene Co-expression Network Analysis ('WGCNA'). It then conducts a screening step within each cluster of features and a selecting step among the surviving features, which provides a relatively unbiased way to do feature selection. By using dominant principle components as regression variables at each leaf and the original features as splitting variables at splitting nodes, FREE tree delivers easily interpretable results while improving computational efficiency.

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

LazyData true

Imports glmertree, pre, WGCNA, MASS

RoxygenNote 7.1.0

Suggests knitr, rmarkdown, testthat (>= 2.1.0)

NeedsCompilation no

Author Yuancheng Xu [aut],
Athanasse Zafirov [cre],
Christina Ramirez [aut],
Dan Kojis [aut],
Min Tan [aut],
Mike Alvarez [aut]

Maintainer Athanasse Zafirov <zafirov@gmail.com>

Repository CRAN

Date/Publication 2020-06-25 15:00:03 UTC

Contents

data	2
FREEtree	13
FREEtree_PC	15
FREEtree_time	17
get_split_names	18
Index	20

data	<i>A dataset containing simulated feature long and wide data. The last six columns contain outcome variable, patient ID, treatment, time and time squared features.</i>
------	---

Description

A dataset containing simulated feature long and wide data. The last six columns contain outcome variable, patient ID, treatment, time and time squared features.

Usage

data

Format

A data frame with 100 rows and 406 variables:

- rand_int** control variable (not used)
- time** time trend variable (1 to 6)
- time2** squared time trend variable
- treatment** binary treatment feature
- patient** patient ID for 20 patients
- y** outcome variable
- V1** simulated feature correlated to varying degrees
- V2** simulated feature correlated to varying degrees
- V3** simulated feature correlated to varying degrees
- V4** simulated feature correlated to varying degrees
- V5** simulated feature correlated to varying degrees
- V6** simulated feature correlated to varying degrees
- V7** simulated feature correlated to varying degrees
- V8** simulated feature correlated to varying degrees
- V9** simulated feature correlated to varying degrees
- V10** simulated feature correlated to varying degrees

V48 simulated feature correlated to varying degrees
V49 simulated feature correlated to varying degrees
V50 simulated feature correlated to varying degrees
V51 simulated feature correlated to varying degrees
V52 simulated feature correlated to varying degrees
V53 simulated feature correlated to varying degrees
V54 simulated feature correlated to varying degrees
V55 simulated feature correlated to varying degrees
V56 simulated feature correlated to varying degrees
V57 simulated feature correlated to varying degrees
V58 simulated feature correlated to varying degrees
V59 simulated feature correlated to varying degrees
V60 simulated feature correlated to varying degrees
V61 simulated feature correlated to varying degrees
V62 simulated feature correlated to varying degrees
V63 simulated feature correlated to varying degrees
V64 simulated feature correlated to varying degrees
V65 simulated feature correlated to varying degrees
V66 simulated feature correlated to varying degrees
V67 simulated feature correlated to varying degrees
V68 simulated feature correlated to varying degrees
V69 simulated feature correlated to varying degrees
V70 simulated feature correlated to varying degrees
V71 simulated feature correlated to varying degrees
V72 simulated feature correlated to varying degrees
V73 simulated feature correlated to varying degrees
V74 simulated feature correlated to varying degrees
V75 simulated feature correlated to varying degrees
V76 simulated feature correlated to varying degrees
V77 simulated feature correlated to varying degrees
V78 simulated feature correlated to varying degrees
V79 simulated feature correlated to varying degrees
V80 simulated feature correlated to varying degrees
V81 simulated feature correlated to varying degrees
V82 simulated feature correlated to varying degrees
V83 simulated feature correlated to varying degrees
V84 simulated feature correlated to varying degrees

- V85 simulated feature correlated to varying degrees
- V86 simulated feature correlated to varying degrees
- V87 simulated feature correlated to varying degrees
- V88 simulated feature correlated to varying degrees
- V89 simulated feature correlated to varying degrees
- V90 simulated feature correlated to varying degrees
- V91 simulated feature correlated to varying degrees
- V92 simulated feature correlated to varying degrees
- V93 simulated feature correlated to varying degrees
- V94 simulated feature correlated to varying degrees
- V95 simulated feature correlated to varying degrees
- V96 simulated feature correlated to varying degrees
- V97 simulated feature correlated to varying degrees
- V98 simulated feature correlated to varying degrees
- V99 simulated feature correlated to varying degrees
- V100 simulated feature correlated to varying degrees
- V101 simulated feature correlated to varying degrees
- V102 simulated feature correlated to varying degrees
- V103 simulated feature correlated to varying degrees
- V104 simulated feature correlated to varying degrees
- V105 simulated feature correlated to varying degrees
- V106 simulated feature correlated to varying degrees
- V107 simulated feature correlated to varying degrees
- V108 simulated feature correlated to varying degrees
- V109 simulated feature correlated to varying degrees
- V110 simulated feature correlated to varying degrees
- V111 simulated feature correlated to varying degrees
- V112 simulated feature correlated to varying degrees
- V113 simulated feature correlated to varying degrees
- V114 simulated feature correlated to varying degrees
- V115 simulated feature correlated to varying degrees
- V116 simulated feature correlated to varying degrees
- V117 simulated feature correlated to varying degrees
- V118 simulated feature correlated to varying degrees
- V119 simulated feature correlated to varying degrees
- V120 simulated feature correlated to varying degrees
- V121 simulated feature correlated to varying degrees

- V159 simulated feature correlated to varying degrees
- V160 simulated feature correlated to varying degrees
- V161 simulated feature correlated to varying degrees
- V162 simulated feature correlated to varying degrees
- V163 simulated feature correlated to varying degrees
- V164 simulated feature correlated to varying degrees
- V165 simulated feature correlated to varying degrees
- V166 simulated feature correlated to varying degrees
- V167 simulated feature correlated to varying degrees
- V168 simulated feature correlated to varying degrees
- V169 simulated feature correlated to varying degrees
- V170 simulated feature correlated to varying degrees
- V171 simulated feature correlated to varying degrees
- V172 simulated feature correlated to varying degrees
- V173 simulated feature correlated to varying degrees
- V174 simulated feature correlated to varying degrees
- V175 simulated feature correlated to varying degrees
- V176 simulated feature correlated to varying degrees
- V177 simulated feature correlated to varying degrees
- V178 simulated feature correlated to varying degrees
- V179 simulated feature correlated to varying degrees
- V180 simulated feature correlated to varying degrees
- V181 simulated feature correlated to varying degrees
- V182 simulated feature correlated to varying degrees
- V183 simulated feature correlated to varying degrees
- V184 simulated feature correlated to varying degrees
- V185 simulated feature correlated to varying degrees
- V186 simulated feature correlated to varying degrees
- V187 simulated feature correlated to varying degrees
- V188 simulated feature correlated to varying degrees
- V189 simulated feature correlated to varying degrees
- V190 simulated feature correlated to varying degrees
- V191 simulated feature correlated to varying degrees
- V192 simulated feature correlated to varying degrees
- V193 simulated feature correlated to varying degrees
- V194 simulated feature correlated to varying degrees
- V195 simulated feature correlated to varying degrees

V196 simulated feature correlated to varying degrees
V197 simulated feature correlated to varying degrees
V198 simulated feature correlated to varying degrees
V199 simulated feature correlated to varying degrees
V200 simulated feature correlated to varying degrees
V201 simulated feature correlated to varying degrees
V202 simulated feature correlated to varying degrees
V203 simulated feature correlated to varying degrees
V204 simulated feature correlated to varying degrees
V205 simulated feature correlated to varying degrees
V206 simulated feature correlated to varying degrees
V207 simulated feature correlated to varying degrees
V208 simulated feature correlated to varying degrees
V209 simulated feature correlated to varying degrees
V210 simulated feature correlated to varying degrees
V211 simulated feature correlated to varying degrees
V212 simulated feature correlated to varying degrees
V213 simulated feature correlated to varying degrees
V214 simulated feature correlated to varying degrees
V215 simulated feature correlated to varying degrees
V216 simulated feature correlated to varying degrees
V217 simulated feature correlated to varying degrees
V218 simulated feature correlated to varying degrees
V219 simulated feature correlated to varying degrees
V220 simulated feature correlated to varying degrees
V221 simulated feature correlated to varying degrees
V222 simulated feature correlated to varying degrees
V223 simulated feature correlated to varying degrees
V224 simulated feature correlated to varying degrees
V225 simulated feature correlated to varying degrees
V226 simulated feature correlated to varying degrees
V227 simulated feature correlated to varying degrees
V228 simulated feature correlated to varying degrees
V229 simulated feature correlated to varying degrees
V230 simulated feature correlated to varying degrees
V231 simulated feature correlated to varying degrees
V232 simulated feature correlated to varying degrees

V270 simulated feature correlated to varying degrees
V271 simulated feature correlated to varying degrees
V272 simulated feature correlated to varying degrees
V273 simulated feature correlated to varying degrees
V274 simulated feature correlated to varying degrees
V275 simulated feature correlated to varying degrees
V276 simulated feature correlated to varying degrees
V277 simulated feature correlated to varying degrees
V278 simulated feature correlated to varying degrees
V279 simulated feature correlated to varying degrees
V280 simulated feature correlated to varying degrees
V281 simulated feature correlated to varying degrees
V282 simulated feature correlated to varying degrees
V283 simulated feature correlated to varying degrees
V284 simulated feature correlated to varying degrees
V285 simulated feature correlated to varying degrees
V286 simulated feature correlated to varying degrees
V287 simulated feature correlated to varying degrees
V288 simulated feature correlated to varying degrees
V289 simulated feature correlated to varying degrees
V290 simulated feature correlated to varying degrees
V291 simulated feature correlated to varying degrees
V292 simulated feature correlated to varying degrees
V293 simulated feature correlated to varying degrees
V294 simulated feature correlated to varying degrees
V295 simulated feature correlated to varying degrees
V296 simulated feature correlated to varying degrees
V297 simulated feature correlated to varying degrees
V298 simulated feature correlated to varying degrees
V299 simulated feature correlated to varying degrees
V300 simulated feature correlated to varying degrees
V301 simulated feature correlated to varying degrees
V302 simulated feature correlated to varying degrees
V303 simulated feature correlated to varying degrees
V304 simulated feature correlated to varying degrees
V305 simulated feature correlated to varying degrees
V306 simulated feature correlated to varying degrees

- V307 simulated feature correlated to varying degrees
- V308 simulated feature correlated to varying degrees
- V309 simulated feature correlated to varying degrees
- V310 simulated feature correlated to varying degrees
- V311 simulated feature correlated to varying degrees
- V312 simulated feature correlated to varying degrees
- V313 simulated feature correlated to varying degrees
- V314 simulated feature correlated to varying degrees
- V315 simulated feature correlated to varying degrees
- V316 simulated feature correlated to varying degrees
- V317 simulated feature correlated to varying degrees
- V318 simulated feature correlated to varying degrees
- V319 simulated feature correlated to varying degrees
- V320 simulated feature correlated to varying degrees
- V321 simulated feature correlated to varying degrees
- V322 simulated feature correlated to varying degrees
- V323 simulated feature correlated to varying degrees
- V324 simulated feature correlated to varying degrees
- V325 simulated feature correlated to varying degrees
- V326 simulated feature correlated to varying degrees
- V327 simulated feature correlated to varying degrees
- V328 simulated feature correlated to varying degrees
- V329 simulated feature correlated to varying degrees
- V330 simulated feature correlated to varying degrees
- V331 simulated feature correlated to varying degrees
- V332 simulated feature correlated to varying degrees
- V333 simulated feature correlated to varying degrees
- V334 simulated feature correlated to varying degrees
- V335 simulated feature correlated to varying degrees
- V336 simulated feature correlated to varying degrees
- V337 simulated feature correlated to varying degrees
- V338 simulated feature correlated to varying degrees
- V339 simulated feature correlated to varying degrees
- V340 simulated feature correlated to varying degrees
- V341 simulated feature correlated to varying degrees
- V342 simulated feature correlated to varying degrees
- V343 simulated feature correlated to varying degrees

- V381 simulated feature correlated to varying degrees
- V382 simulated feature correlated to varying degrees
- V383 simulated feature correlated to varying degrees
- V384 simulated feature correlated to varying degrees
- V385 simulated feature correlated to varying degrees
- V386 simulated feature correlated to varying degrees
- V387 simulated feature correlated to varying degrees
- V388 simulated feature correlated to varying degrees
- V389 simulated feature correlated to varying degrees
- V390 simulated feature correlated to varying degrees
- V391 simulated feature correlated to varying degrees
- V392 simulated feature correlated to varying degrees
- V393 simulated feature correlated to varying degrees
- V394 simulated feature correlated to varying degrees
- V395 simulated feature correlated to varying degrees
- V396 simulated feature correlated to varying degrees
- V397 simulated feature correlated to varying degrees
- V398 simulated feature correlated to varying degrees
- V399 simulated feature correlated to varying degrees
- V400 simulated feature correlated to varying degrees

FREEtree

Initial FREEtree call which then calls actual FREEtree methods depending on parameters being passed through.

Description

Initial FREEtree call which then calls actual FREEtree methods depending on parameters being passed through.

Usage

```
FREEtree(  
  data,  
  fixed_regress = NULL,  
  fixed_split = NULL,  
  var_select = NULL,  
  power = 6,  
  minModuleSize = 1,  
  cluster,  
  maxdepth_factor_screen = 0.04,
```

```

maxdepth_factor_select = 0.5,
Fuzzy = TRUE,
minsize_multiplier = 5,
alpha_screen = 0.2,
alpha_select = 0.2,
alpha_predict = 0.05
)

```

Arguments

<code>data</code>	data to train or test FREEtree on.
<code>fixed_regress</code>	user specified char vector of regressors that will never be screened out; if <code>fixed_regress = NULL</code> , method uses PC as regressor at screening step.
<code>fixed_split</code>	user specified char vector of features to be used in splitting with certainty.
<code>var_select</code>	a char vector containing features to be selected. These features will be clustered by WGCNA and the chosen ones will be used in regression and splitting.
<code>power</code>	soft thresholding power parameter of WGCNA.
<code>minModuleSize</code>	WGCNA's minimum module size parameter.
<code>cluster</code>	the variable name of each cluster (in terms of random effect) using glmer's implementation.
<code>maxdepth_factor_screen</code>	when selecting features from one module, the <code>maxdepth</code> of the <code>glmertree</code> is set to ceiling function of <code>maxdepth_factor_screen*(features in that module)</code> . Default is 0.04.
<code>maxdepth_factor_select</code>	Given screened features (from each modules, if <code>Fuzzy=FALSE</code> , that is the selected non-grey features from each non-grey modules), we want to select again from those screened features. The <code>maxdepth</code> of that <code>glmertree</code> is set to be ceiling of <code>maxdepth_factor_select*(#screened features)</code> . Default is 0.6. for the <code>maxdepth</code> of the prediction tree (final tree), <code>maxdepth</code> is set to the length of the <code>split_var</code> (fixed+chosen ones).
<code>Fuzzy</code>	boolean to indicate desire to screen like Fuzzy Forest if <code>Fuzzy = TRUE</code> ; if <code>Fuzzy= FALSE</code> , first screen within non-grey modules and then select the final non-grey features within the selected ones from each non-grey module; Use this final non-grey features as regressors (plus <code>fixed_regress</code>) and use grey features as <code>split_var</code> to select grey features. Then use final non-grey features and selected grey features together in splitting and regression variables, to do the final prediction. <code>Fuzzy=FALSE</code> is used if there are so many non-grey features and you want to protect grey features.
<code>minsize_multiplier</code>	At the final prediction tree, the <code>minsize</code> = <code>minsize_multiplier</code> times the length of final regressors. The default is 5. Note that we only set <code>minsize</code> for the final prediction tree instead of trees at the feature selection step since during feature selection, we don't have to be so careful. Note that when tuning the parameters, larger alpha and smaller <code>minsize_multiplier</code> will result in deeper tree and therefore may cause overfitting problem. It is recommended to decrease alpha and decrease <code>minsize_multiplier</code> at the same time.

alpha_screen alpha used in screening step.
 alpha_select alpha used in selection step.
 alpha_predict alpha used in prediction step.

Value

a glmertree object (trained tree).

Examples

```
#locate example data file
dataf <- system.file("data/data.RData", package="FREEtree")
mytree = FREEtree(data,fixed_regress=c("time","time2"), fixed_split=c("treatment"),
  var_select=paste("V",1:200,sep=""), minModuleSize = 5,
  cluster="patient", Fuzzy=TRUE, maxdepth_factor_select = 0.5,
  maxdepth_factor_screen = 0.04, minsize_multiplier = 5,
  alpha_screen = 0.2, alpha_select=0.2,alpha_predict=0.05)
```

FREEtree_PC

Version of FREEtree called when fixed_regress is NULL, uses principal components (PC) as regressors for non-grey modules.

Description

Version of FREEtree called when fixed_regress is NULL, uses principal components (PC) as regressors for non-grey modules.

Usage

```
FREEtree_PC(
  data,
  fixed_split,
  var_select,
  power,
  minModuleSize,
  cluster,
  maxdepth_factor_screen,
  maxdepth_factor_select,
  Fuzzy,
  minsize_multiplier,
  alpha_screen,
  alpha_select,
  alpha_predict
)
```

Arguments

<code>data</code>	data to train or test FREEtree on.
<code>fixed_split</code>	user specified char vector of features to be used in splitting with certainty.
<code>var_select</code>	a char vector containing features to be selected. These features will be clustered by WGCNA and the chosen ones will be used in regression and splitting.
<code>power</code>	soft thresholding power parameter of WGCNA.
<code>minModuleSize</code>	WGCNA's minimum module size parameter.
<code>cluster</code>	the variable name of each cluster (in terms of random effect) using glmer's implementation.
<code>maxdepth_factor_screen</code>	when selecting features from one module, the maxdepth of the glmertree is set to ceiling function of $\text{maxdepth_factor_screen} * (\text{features in that module})$. Default is 0.04.
<code>maxdepth_factor_select</code>	Given screened features (from each modules, if Fuzzy=FALSE, that is the selected non-grey features from each non-grey modules), we want to select again from those screened features. The maxdepth of that glmertree is set to be ceiling of $\text{maxdepth_factor_select} * (\#\text{screened features})$. Default is 0.6. for the maxdepth of the prediction tree (final tree), maxdepth is set to the length of the <code>split_var</code> (fixed+chosen ones).
<code>Fuzzy</code>	boolean to indicate desire to screen like Fuzzy Forest if Fuzzy = TRUE; if Fuzzy= FALSE, first screen within non-grey modules and then select the final non-grey features within the selected ones from each non-grey module; Use this final non-grey features as regressors (plus <code>fixed_regress</code>) and use grey features as <code>split_var</code> to select grey features. Then use final non-grey features and selected grey features together in splitting and regression variables, to do the final prediction. Fuzzy=FALSE is used if there are so many non-grey features and you want to protect grey features.
<code>minsize_multiplier</code>	At the final prediction tree, the <code>minsize</code> = <code>minsize_multiplier</code> times the length of final regressors. The default is 5. Note that we only set <code>minsize</code> for the final prediction tree instead of trees at the feature selection step since during feature selection, we don't have to be so careful. Note that when tuning the parameters, larger alpha and smaller <code>minsize_multiplier</code> will result in deeper tree and therefore may cause overfitting problem. It is recommended to decrease alpha and decrease <code>minsize_multiplier</code> at the same time.
<code>alpha_screen</code>	alpha used in screening step.
<code>alpha_select</code>	alpha used in selection step.
<code>alpha_predict</code>	alpha used in prediction step.

Value

a glmertree object (trained tree). dictionary' with keys=name of color, values=names of features of that color

FREEtree_time	<i>Version of FREEtree called when var_select and fixed_regress are specified,</i>
---------------	--

Description

Version of FREEtree called when var_select and fixed_regress are specified,

Usage

```
FREEtree_time(
  data,
  fixed_regress,
  fixed_split,
  var_select,
  power,
  minModuleSize,
  cluster,
  maxdepth_factor_screen,
  maxdepth_factor_select,
  Fuzzy,
  minsize_multiplier,
  alpha_screen,
  alpha_select,
  alpha_predict
)
```

Arguments

data	data to train or test FREEtree on.
fixed_regress	user specified char vector of regressors that will never be screened out; if fixed_regress = NULL, method uses PC as regressor at screening step.
fixed_split	user specified char vector of features to be used in splitting with certainty.
var_select	a char vector containing features to be selected. These features will be clustered by WGCNA and the chosen ones will be used in regression and splitting.
power	soft thresholding power parameter of WGCNA.
minModuleSize	minimum possible module size parameter of WGCNA.
cluster	the variable name of each cluster (in terms of random effect) using glmer's implementation.
maxdepth_factor_screen	when selecting features from one module, the maxdepth of the glmertree is set to ceiling function of maxdepth_factor_screen*(features in that module). Default is 0.04.

maxdepth_factor_select	Given screened features (from each modules, if Fuzzy=FALSE, that is the selected non-grey features from each non-grey modules), we want to select again from those screened features. The maxdepth of that glmertree is set to be ceiling of maxdepth_factor_select*(#screened features). Default is 0.6. for the maxdepth of the prediction tree (final tree), maxdepth is set to the length of the split_var (fixed+chosen ones).
Fuzzy	boolean to indicate desire to screen like Fuzzy Forest if Fuzzy = TRUE; if Fuzzy= FALSE, first screen within non-grey modules and then select the final non-grey features within the selected ones from each non-grey module; Use this final non-grey features as regressors (plus fixed_regress) and use grey features as split_var to select grey features. Then use final non-grey features and selected grey features together in splitting and regression variables, to do the final prediction. Fuzzy=FALSE is used if there are so many non-grey features and you want to protect grey features.
minsize_multiplier	At the final prediction tree, the minsize = minsize_multiplier times the length of final regressors. The default is 5. Note that we only set minsize for the final prediction tree instead of trees at the feature selection step since during feature selection, we don't have to be so careful. Note that when tuning the parameters, larger alpha and smaller minsize_multiplier will result in deeper tree and therefore may cause overfitting problem. It is recommended to decrease alpha and decrease minsize_multiplier at the same time.
alpha_screen	alpha used in screening step.
alpha_select	alpha used in selection step.
alpha_predict	alpha used in prediction step.

Value

a glmertree object (trained tree). dictionary' with keys=name of color, values=names of features of that color

get_split_names *Method for extracting names of splitting features used in a tree.*

Description

Method for extracting names of splitting features used in a tree.

Usage

```
get_split_names(tree, data)
```

Arguments

tree	a tree object.
data	train or test set.

Value

names of splitting features extracted from tree object.

Index

* **datasets**

data, [2](#)

data, [2](#)

FREEtree, [13](#)

FREEtree_PC, [15](#)

FREEtree_time, [17](#)

get_split_names, [18](#)