Package ‘vimp’

December 9, 2022

Type Package

Title Perform Inference on Algorithm-Agnostic Variable Importance

Version 2.3.1

Description Calculate point estimates of and valid confidence intervals for nonparametric, algorithm-agnostic variable importance measures in high and low dimensions, using flexible estimators of the underlying regression functions. For more information about the methods, please see Williamson et al. (Biometrics, 2020), Williamson et al. (JASA, 2021), and Williamson and Feng (ICML, 2020).

Depends R (>= 3.1.0)

Imports SuperLearner, stats, dplyr, magrittr, ROCR, tibble, rlang, MASS, boot, data.table

Suggests knitr, rmarkdown, gam, xgboost, glmnet, ranger, polspline, quadprog, covr, testthat, ggplot2, cowplot, cvAUC, tidyselect

License MIT + file LICENSE


BugReports https://github.com/bdwilliamson/vimp/issues

RoxygenNote 7.2.1

VignetteBuilder knitr

LazyData true

NeedsCompilation no

Author Brian D. Williamson [aut, cre]
(https://orcid.org/0000-0002-7024-548X),
Jean Feng [ctb],
Noah Simon [ths] (https://orcid.org/0000-0002-8985-2474),
Marco Carone [ths] (https://orcid.org/0000-0003-2106-0953)

Maintainer Brian D. Williamson <brian.d.williamson@kp.org>

Repository CRAN

Date/Publication 2022-12-09 17:10:02 UTC
R topics documented:

average_vim .................................................. 3
bootstrap_se .................................................. 4
check_fitted_values ......................................... 5
check_inputs .................................................. 6
create_z ....................................................... 7
cv_vim ......................................................... 8
estimate ........................................................ 14
estimate.predictiveness_measure ......................... 15
estimate_eif_projection ...................................... 15
estimate_nuisances .......................................... 16
estimate_type_predictiveness .............................. 17
est_predictiveness ........................................... 18
est_predictiveness_cv ....................................... 19
extract_sampled_split_predictions ...................... 21
format.predictiveness_measure ......................... 22
format.vim .................................................... 22
get_cv_sl_folds .............................................. 23
get_full_type ................................................ 23
get_test_set ................................................... 24
make_folds ...................................................... 24
make_kfold ..................................................... 25
measure_accuracy ........................................... 25
measure_anova ............................................... 27
measure_auc ................................................. 28
measure_average_value .................................... 29
measure_cross_entropy ..................................... 31
measure_deviance .......................................... 32
measure_mse .................................................. 33
measure_r_squared ......................................... 35
merge_vim ...................................................... 36
predictiveness_measure .................................... 37
print.predictiveness_measure ............................. 39
print.vim ...................................................... 40
process_arg_lst ............................................. 40
run_sl ......................................................... 41
sample_subsets ............................................. 42
scale_est ...................................................... 43
spvim_ics ...................................................... 43
spvim_se ....................................................... 44
sp_vim ........................................................ 45
vim ............................................................. 49
vimp ............................................................ 53
vimp_accuracy ............................................... 55
vimp_anova .................................................... 60
vimp_auc ....................................................... 63
vimp_ci ........................................................ 68
**average_vim**

Average multiple independent importance estimates

**Description**

Average the output from multiple calls to `vimp_regression`, for different independent groups, into a single estimate with a corresponding standard error and confidence interval.

**Usage**

```r
average_vim(..., weights = rep(1/length(list(...)), length(list(...))))
```

**Arguments**

- `...` an arbitrary number of `vim` objects.
- `weights` how to average the vims together, and must sum to 1; defaults to 1/(number of vims) for each vim, corresponding to the arithmetic mean

**Value**

an object of class `vim` containing the (weighted) average of the individual importance estimates, as well as the appropriate standard error and confidence interval. This results in a list containing:

- `s` - a list of the column(s) to calculate variable importance for
- `SL.library` - a list of the libraries of learners passed to SuperLearner
- `full_fit` - a list of the fitted values of the chosen method fit to the full data
- `red_fit` - a list of the fitted values of the chosen method fit to the reduced data
- `est` - a vector with the corrected estimates
- `naive` - a vector with the naive estimates
- `update` - a list with the influence curve-based updates
- `mat` - a matrix with the estimated variable importance, the standard error, and the $(1 - \alpha) \times 100\%$ confidence interval
- `full_mod` - a list of the objects returned by the estimation procedure for the full data regression (if applicable)
- `red_mod` - a list of the objects returned by the estimation procedure for the reduced data regression (if applicable)
- `alpha` - the level, for confidence interval calculation
- `y` - a list of the outcomes
Examples

```r
# generate the data
p <- 2
n <- 100
x <- data.frame(replicate(p, stats::runif(n, -5, 5)))

# apply the function to the x's
smooth <- (x[,1]/5)^2*(x[,1]+7)/5 + (x[,2]/3)^2

# generate Y ~ Normal (smooth, 1)
y <- smooth + stats::rnorm(n, 0, 1)

# set up a library for SuperLearner; note simple library for speed
library("SuperLearner")
learners <- c("SL.glm", "SL.mean")

# get estimates on independent splits of the data
samp <- sample(1:n, n/2, replace = FALSE)

# using Super Learner (with a small number of folds, for illustration only)
est_2 <- vimp_regression(Y = y[samp], X = x[samp, ], indx = 2, V = 2,
run_regression = TRUE, alpha = 0.05,
SL.library = learners, cvControl = list(V = 2))
est_1 <- vimp_regression(Y = y[-samp], X = x[-samp, ], indx = 2, V = 2,
run_regression = TRUE, alpha = 0.05,
SL.library = learners, cvControl = list(V = 2))
est <- average_vim(est_1, est_2, weights = c(1/2, 1/2))
```

---

**bootstrap_se**

**Compute bootstrap-based standard error estimates for variable importance**

**Description**

Compute bootstrap-based standard error estimates for variable importance

**Usage**

```r
bootstrap_se(Y = NULL, f1 = NULL, f2 = NULL, type = "r_squared", b = 1000,
boot_interval_type = "perc", alpha = 0.05)
```
check_fitted_values

Arguments

Y  the outcome.

f1  the fitted values from a flexible estimation technique regressing Y on X. A vector of the same length as Y; if sample-splitting is desired, then the value of f1 at each position should be the result of predicting from a model trained without that observation.

f2  the fitted values from a flexible estimation technique regressing either (a) f1 or (b) Y on X withholding the columns in indx. A vector of the same length as Y; if sample-splitting is desired, then the value of f2 at each position should be the result of predicting from a model trained without that observation.

type  the type of importance to compute; defaults to r_squared, but other supported options are auc, accuracy, deviance, and anova.

b  the number of bootstrap replicates (only used if bootstrap = TRUE and sample_splitting = FALSE); defaults to 1000.

boot_interval_type  the type of bootstrap interval (one of "norm", "basic", "stud", "perc", or "bca", as in boot(boot.ci)) if requested. Defaults to "perc".

alpha  the level to compute the confidence interval at. Defaults to 0.05, corresponding to a 95% confidence interval.

Value

a bootstrap-based standard error estimate

Description

Check pre-computed fitted values for call to vim, cv_vim, or sp_vim

Usage

check_fitted_values(
  Y = NULL,
  f1 = NULL,
  f2 = NULL,
  cross_fitted_f1 = NULL,
  cross_fitted_f2 = NULL,
  sample_splitting_folds = NULL,
  cross_fitting_folds = NULL,
  cross_fitted_se = TRUE,
  V = NULL,
  ss_V = NULL,
  cv = FALSE
)
Arguments

Y the outcome
f1 estimator of the population-optimal prediction function using all covariates
f2 estimator of the population-optimal prediction function using the reduced set of covariates
cross_fitted_f1 cross-fitted estimator of the population-optimal prediction function using all covariates
cross_fitted_f2 cross-fitted estimator of the population-optimal prediction function using the reduced set of covariates
sample_splitting_folds the folds for sample-splitting (used for hypothesis testing)
cross_fitting_folds the folds for cross-fitting (used for point estimates of variable importance in cv_vim and sp_vim)
cross_fitted_se logical; should cross-fitting be used to estimate standard errors?
V the number of cross-fitting folds
ss_V the number of folds for CV (if sample_splitting is TRUE)
cv a logical flag indicating whether or not to use cross-fitting

Details

Ensure that inputs to vim, cv_vim, and sp_vim follow the correct formats.

Value

None. Called for the side effect of stopping the algorithm if any inputs are in an unexpected format.

Usage

check_inputs(Y, X, f1, f2, indx)
create_z

Arguments

Y  the outcome
X  the covariates
f1 estimator of the population-optimal prediction function using all covariates
f2 estimator of the population-optimal prediction function using the reduced set of covariates
indx the index or indices of the covariate(s) of interest

Details

Ensure that inputs to vim, cv_vim, and sp_vim follow the correct formats.

Value

None. Called for the side effect of stopping the algorithm if any inputs are in an unexpected format.

create_z

Create complete-case outcome, weights, and Z

Description

Create complete-case outcome, weights, and Z

Usage

create_z(Y, C, Z, X, ipc_weights)

Arguments

Y  the outcome
C  indicator of missing or observed
Z  the covariates observed in phase 1 and 2 data
X  all covariates
ipc_weights the weights

Value

a list, with the complete-case outcome, weights, and Z matrix
cv_vim

Nonparametric Intrinsic Variable Importance Estimates and Inference using Cross-fitting

Description

Compute estimates and confidence intervals using cross-fitting for nonparametric intrinsic variable importance based on the population-level contrast between the oracle predictiveness using the feature(s) of interest versus not.

Usage

```r
cv_vim(
Y = NULL,
X = NULL,
cross_fitted_f1 = NULL,
cross_fitted_f2 = NULL,
f1 = NULL,
f2 = NULL,
indx = 1,
V = ifelse(is.null(cross_fitting_folds), 5, length(unique(cross_fitting_folds))),
sample_splitting = TRUE,
final_point_estimate = "split",
sample_splitting_folds = NULL,
cross_fitting_folds = NULL,
stratified = FALSE,
type = "r_squared",
run_regression = TRUE,
SL.library = c("SL.glmnet", "SL.xgboost", "SL.mean"),
alpha = 0.05,
delta = 0,
scale = "identity",
na.rm = FALSE,
C = rep(1, length(Y)),
Z = NULL,
ipc_scale = "identity",
ipc_weights = rep(1, length(Y)),
ipc_est_type = "aipw",
scale_est = TRUE,
nuisance_estimators_full = NULL,
nuisance_estimators_reduced = NULL,
exposure_name = NULL,
cross_fitted_se = TRUE,
bootstrap = FALSE,
b = 1000,
boot_interval_type = "perc",
...
```

Arguments

Y
-the outcome.

X
-the covariates. If type = "average_value", then the exposure variable should be part of X, with its name provided in exposure_name.

cross_fitted_f1
-the predicted values on validation data from a flexible estimation technique regressing Y on X in the training data. Provided as either (a) a vector, where each element is the predicted value when that observation is part of the validation fold; or (b) a list of length V, where each element in the list is a set of predictions on the corresponding validation data fold. If sample-splitting is requested, then these must be estimated specially; see Details. However, the resulting vector should be the same length as Y; if using a list, then the summed length of each element across the list should be the same length as Y (i.e., each observation is included in the predictions).

cross_fitted_f2
-the predicted values on validation data from a flexible estimation technique regressing either (a) the fitted values in cross_fitted_f1, or (b) Y, on X withholding the columns in indx. Provided as either (a) a vector, where each element is the predicted value when that observation is part of the validation fold; or (b) a list of length V, where each element in the list is a set of predictions on the corresponding validation data fold. If sample-splitting is requested, then these must be estimated specially; see Details. However, the resulting vector should be the same length as Y; if using a list, then the summed length of each element across the list should be the same length as Y (i.e., each observation is included in the predictions).

f1
-the fitted values from a flexible estimation technique regressing Y on X. If sample-splitting is requested, then these must be estimated specially; see Details. If cross_fitted_se = TRUE, then this argument is not used.

f2
-the fitted values from a flexible estimation technique regressing either (a) f1 or (b) Y on X withholding the columns in indx. If sample-splitting is requested, then these must be estimated specially; see Details. If cross_fitted_se = TRUE, then this argument is not used.

indx
-the indices of the covariate(s) to calculate variable importance for; defaults to 1.

V
-the number of folds for cross-fitting, defaults to 5. If sample_splitting = TRUE, then a special type of V-fold cross-fitting is done. See Details for a more detailed explanation.

sample_splitting
-should we use sample-splitting to estimate the full and reduced predictiveness? Defaults to TRUE, since inferences made using sample_splitting = FALSE will be invalid for variables with truly zero importance.

final_point_estimate
-if sample splitting is used, should the final point estimates be based on only the sample-split folds used for inference ("split", the default), or should they
instead be based on the full dataset ("full") or the average across the point estimates from each sample split ("average")? All three options result in valid point estimates – sample-splitting is only required for valid inference.

**sample_splitting_folds**
the folds used for sample-splitting; these identify the observations that should be used to evaluate predictiveness based on the full and reduced sets of covariates, respectively. Only used if run_regression = FALSE.

**cross_fitting_folds**
the folds for cross-fitting. Only used if run_regression = FALSE.

**stratified**
if run_regression = TRUE, then should the generated folds be stratified based on the outcome (helps to ensure class balance across cross-validation folds)

**type**
the type of importance to compute; defaults to r_squared, but other supported options are auc, accuracy, deviance, and anova.

**run_regression**
if outcome Y and covariates X are passed to vimp_accuracy, and run_regression is TRUE, then Super Learner will be used; otherwise, variable importance will be computed using the inputted fitted values.

**SL.library**
a character vector of learners to pass to SuperLearner, if f1 and f2 are Y and X, respectively. Defaults to SL.glmnet, SL.xgboost, and SL.mean.

**alpha**
the level to compute the confidence interval at. Defaults to 0.05, corresponding to a 95% confidence interval.

**delta**
the value of the δ-null (i.e., testing if importance < δ); defaults to 0.

**scale**
should CIs be computed on original ("identity") or another scale? (options are "log" and "logit")

**na.rm**
should we remove NAs in the outcome and fitted values in computation? (defaults to FALSE)

**C**
the indicator of coarsening (1 denotes observed, 0 denotes unobserved).

**Z**
either (i) NULL (the default, in which case the argument C above must be all ones), or (ii) a character vector specifying the variable(s) among Y and X that are thought to play a role in the coarsening mechanism. To specify the outcome, use "Y"; to specify covariates, use a character number corresponding to the desired position in X (e.g., "1").

**ipc_scale**
what scale should the inverse probability weight correction be applied on (if any)? Defaults to "identity". (other options are "log" and "logit")

**ipc_weights**
weights for the computed influence curve (i.e., inverse probability weights for coarsened-at-random settings). Assumed to be already inverted (i.e., ipc_weights = 1 / [estimated probability weights]).

**ipc_est_type**
the type of procedure used for coarsened-at-random settings; options are "ipw" (for inverse probability weighting) or "aipw" (for augmented inverse probability weighting). Only used if C is not all equal to 1.

**scale_est**
should the point estimate be scaled to be greater than or equal to 0? Defaults to TRUE.

**nuisance_estimators_full**
(only used if type = "average_value") a list of nuisance function estimators on the observed data (may be within a specified fold, for cross-fitted estimates).
Specifically: an estimator of the optimal treatment rule; an estimator of the propensity score under the estimated optimal treatment rule; and an estimator of the outcome regression when treatment is assigned according to the estimated optimal rule.

nuisance_estimators_reduced
(only used if type = "average_value") a list of nuisance function estimators on the observed data (may be within a specified fold, for cross-fitted estimates). Specifically: an estimator of the optimal treatment rule; an estimator of the propensity score under the estimated optimal treatment rule; and an estimator of the outcome regression when treatment is assigned according to the estimated optimal rule.

exposure_name
(only used if type = "average_value") the name of the exposure of interest; binary, with 1 indicating presence of the exposure and 0 indicating absence of the exposure.

cross_fitted_se
should we use cross-fitting to estimate the standard errors (TRUE, the default) or not (FALSE)?

bootstrap
should bootstrap-based standard error estimates be computed? Defaults to FALSE (and currently may only be used if sample_splitting = FALSE).

b
the number of bootstrap replicates (only used if bootstrap = TRUE and sample_splitting = FALSE); defaults to 1000.

boot_interval_type
the type of bootstrap interval (one of "norm", "basic", "stud", "perc", or "bca", as in boot(boot.ci)) if requested. Defaults to "perc".

... other arguments to the estimation tool, see "See also".

Details

We define the population variable importance measure (VIM) for the group of features (or single feature) $s$ with respect to the predictiveness measure $V$ by

$$
\psi_{0,s} := V(f_0, P_0) - V(f_{0,s}, P_0),
$$

where $f_0$ is the population predictiveness maximizing function, $f_{0,s}$ is the population predictiveness maximizing function that is only allowed to access the features with index not in $s$, and $P_0$ is the true data-generating distribution.

Cross-fitted VIM estimates are computed differently if sample-splitting is requested versus if it is not. We recommend using sample-splitting in most cases, since only in this case will inferences be valid if the variable(s) of interest have truly zero population importance. The purpose of cross-fitting is to estimate $f_0$ and $f_{0,s}$ on independent data from estimating $P_0$; this can result in improved performance, especially when using flexible learning algorithms. The purpose of sample-splitting is to estimate $f_0$ and $f_{0,s}$ on independent data; this allows valid inference under the null hypothesis of zero importance.

Without sample-splitting, cross-fitted VIM estimates are obtained by first splitting the data into $K$ folds; then using each fold in turn as a hold-out set, constructing estimators $f_{n,k}$ and $f_{n,k,s}$ of $f_0$.
and \( f_{0,s} \), respectively on the training data and estimator \( P_{n,k} \) of \( P_0 \) using the test data; and finally, computing

\[
\psi_{n,s} := K^{(-1)} \sum_{k=1}^{K} \{ V(f_{n,k}, P_{n,k}) - V(f_{n,k,s}, P_{n,k}) \}.
\]

With sample-splitting, cross-fitted VIM estimates are obtained by first splitting the data into \( 2K \) folds. These folds are further divided into 2 groups of folds. Then, for each fold \( k \) in the first group, estimator \( f_{n,k} \) of \( f_0 \) is constructed using all data besides the \( k \)th fold in the group (i.e., \((2K-1)/(2K)\) of the data) and estimator \( P_{n,k} \) of \( P_0 \) is constructed using the held-out data (i.e., \( 1/2K \) of the data); then, computing

\[
v_{n,k} = V(f_{n,k}, P_{n,k}).
\]

Similarly, for each fold \( k \) in the second group, estimator \( f_{n,k,s} \) of \( f_{0,s} \) is constructed using all data besides the \( k \)th fold in the group (i.e., \((2K-1)/(2K)\) of the data) and estimator \( P_{n,k} \) of \( P_0 \) is constructed using the held-out data (i.e., \( 1/2K \) of the data); then, computing

\[
v_{n,k,s} = V(f_{n,k,s}, P_{n,k}).
\]

Finally,

\[
\psi_{n,s} := K^{(-1)} \sum_{k=1}^{K} \{ v_{n,k} - v_{n,k,s} \}.
\]

See the paper by Williamson, Gilbert, Simon, and Carone for more details on the mathematics behind the \texttt{cv_vim} function, and the validity of the confidence intervals.

In the interest of transparency, we return most of the calculations within the \texttt{vim} object. This results in a list including:

- \texttt{s} the column(s) to calculate variable importance for
- \texttt{SL.library} the library of learners passed to \texttt{SuperLearner}
- \texttt{full_fit} the fitted values of the chosen method fit to the full data (a list, for train and test data)
- \texttt{red_fit} the fitted values of the chosen method fit to the reduced data (a list, for train and test data)
- \texttt{est} the estimated variable importance
- \texttt{naive} the naive estimator of variable importance
- \texttt{eif} the estimated efficient influence function
- \texttt{eif_full} the estimated efficient influence function for the full regression
- \texttt{eif_reduced} the estimated efficient influence function for the reduced regression
- \texttt{se} the standard error for the estimated variable importance
- \texttt{ci} the \( (1 - \alpha) \times 100\% \) confidence interval for the variable importance estimate
- \texttt{test} a decision to either reject (\texttt{TRUE}) or not reject (\texttt{FALSE}) the null hypothesis, based on a conservative test
- \texttt{p_value} a p-value based on the same test as \texttt{test}
- \texttt{full_mod} the object returned by the estimation procedure for the full data regression (if applicable)
- \texttt{red_mod} the object returned by the estimation procedure for the reduced data regression (if applicable)
\textbf{cv_vim} \\

\textbf{alpha} the level, for confidence interval calculation \\
\textbf{sample_splitting_folds} the folds used for hypothesis testing \\
\textbf{cross_fitting_folds} the folds used for cross-fitting \\
\textbf{y} the outcome \\
\textbf{ipc_weights} the weights \\
\textbf{mat} a tibble with the estimate, SE, CI, hypothesis testing decision, and p-value \\

\textbf{Value} \\
An object of class \texttt{vim}. See Details for more information. \\

\textbf{See Also} \\
\texttt{SuperLearner} for specific usage of the \texttt{SuperLearner} function and package. \\

\textbf{Examples} \\
n <- 100 
p <- 2 
# generate the data 
x <- data.frame(replicate(p, stats::runif(n, -5, 5))) 

# apply the function to the x's 
smooth <- (x[,1]/5)^2*(x[,1]+7)/5 + (x[,2]/3)^2 

# generate Y \sim \text{Normal} (smooth, 1) 
y <- as.matrix(smooth + stats::rnorm(n, 0, 1)) 

# set up a library for SuperLearner; note simple library for speed 
library("SuperLearner") 
learners <- c("SL.glm") 

# ----------------------------------------- 
# using Super Learner (with a small number of folds, for illustration only) 
# ----------------------------------------- 
set.seed(4747) 
est <- cv_vim(Y = y, X = x, indx = 2, V = 2, 
type = "r_squared", run_regression = TRUE, 
SL.library = learners, cvControl = list(V = 2), alpha = 0.05) 

# ------------------------------------------ 
# doing things by hand, and plugging them in 
# (with a small number of folds, for illustration only) 
# ------------------------------------------ 
# set up the folds 
indx <- 2 
V <- 2 
Y <- matrix(y) 
set.seed(4747) 
# Note that the CV.SuperLearner should be run with an outer layer
# of 2*V folds (for V-fold cross-fitted importance)
full_cv_fit <- suppressWarnings(SuperLearner::CV.SuperLearner(
  Y = Y, X = x, SL.library = learners, cvControl = list(V = 2 * V),
  innerCvControl = list(list(V = V))
))
full_cv_preds <- full_cv_fit$SL.predict
# use the same cross-fitting folds for reduced
reduced_cv_fit <- suppressWarnings(SuperLearner::CV.SuperLearner(
  Y = Y, X = x[, -indx, drop = FALSE], SL.library = learners,
  cvControl = SuperLearner::SuperLearner.CV.control(
    V = 2 * V, validRows = full_cv_fit$folds
  ),
  innerCvControl = list(list(V = V))
))
reduced_cv_preds <- reduced_cv_fit$SL.predict
# for hypothesis testing
cross_fitting_folds <- get_cv_sl_folds(full_cv_fit$folds)
set.seed(1234)
sample_splitting_folds <- make_folds(unique(cross_fitting_folds), V = 2)
set.seed(5678)
est <- cv_vim(Y = y, cross_fitted_f1 = full_cv_preds,
  cross_fitted_f2 = reduced_cv_preds, indx = 2, delta = 0, V = V, type = "r_squared",
  cross_fitting_folds = cross_fitting_folds,
  sample_splitting_folds = sample_splitting_folds,
  run_regression = FALSE, alpha = 0.05, na.rm = TRUE)

---

**estimate**

*Estimate a Predictiveness Measure*

**Description**

Generic function for estimating a predictiveness measure (e.g., R-squared or classification accuracy).

**Usage**

`estimate(x, ...)`

**Arguments**

- `x`  
  An R object. Currently, there are methods for `predictiveness_measure` objects only.

- `...`  
  further arguments passed to or from other methods.
estimate.predictiveness_measure

Obtain a Point Estimate and Efficient Influence Function Estimate for a Given Predictiveness Measure

Description

Obtain a Point Estimate and Efficient Influence Function Estimate for a Given Predictiveness Measure

Usage

## S3 method for class 'predictiveness_measure'
estimate(x, ...)

Arguments

x
an object of class "predictiveness_measure"

... other arguments to type-specific predictiveness measures (currently unused)

Value

A list with the point estimate, naive point estimate (for ANOVA only), estimated EIF, and the predictions for coarsened data EIF (for coarsened data settings only)

estimate_eif_projection

Estimate projection of EIF on fully-observed variables

Description

Estimate projection of EIF on fully-observed variables

Usage

estimate_eif_projection(
  obs_grad = NULL,
  C = NULL,
  Z = NULL,
  ipc_fit_type = NULL,
  ipc_eif_preds = NULL,
  ...
)

**Arguments**

- `obs_grad` the estimated (observed) EIF  
- `C` the indicator of coarsening (1 denotes observed, 0 denotes unobserved).  
- `Z` either NULL (if no coarsening) or a matrix-like object containing the fully observed data.  
- `ipc_fit_type` if "external", then use `ipc_eif_preds`; if "SL", fit a SuperLearner to determine the IPC correction to the efficient influence function.  
- `ipc_eif_preds` if `ipc_fit_type = "external"`, the fitted values from a regression of the full-data EIF on the fully observed covariates/outcome; otherwise, not used.  
- `...` other arguments to SuperLearner, if `ipc_fit_type = "SL"`.  

**Value**

the projection of the EIF onto the fully-observed variables

---

**estimate_nuisances**  
Estimate nuisance functions for average value-based VIMs

**Description**

Estimate nuisance functions for average value-based VIMs

**Usage**

```r
estimate_nuisances(
    fit,
    X,
    exposure_name,
    V = 1,
    SL.library,
    sample_splitting,
    sample_splitting_folds,
    verbose,
    weights,
    cross_fitted_se,
    split = 1,
    ...
)
```

**Arguments**

- `fit` the fitted nuisance function estimator  
- `X` the covariates. If `type = "average_value"`, then the exposure variable should be part of `X`, with its name provided in `exposure_name`. 
**estimation_type_predictiveness**

**Description**

Estimate the specified type of predictiveness

**Usage**

```r
estimate_type_predictiveness(arg_lst, type)
```

**Arguments**

- `arg_lst` a list of arguments; from, e.g., `predictiveness_measure`
- `type` the type of predictiveness, e.g., "r_squared"
est_predictiveness

Estimate a nonparametric predictiveness functional

Description

Compute nonparametric estimates of the chosen measure of predictiveness.

Usage

est_predictiveness(
  fitted_values,
  y,
  a = NULL,
  full_y = NULL,
  type = "r_squared",
  C = rep(1, length(y)),
  Z = NULL,
  ipc_weights = rep(1, length(C)),
  ipc_fit_type = "external",
  ipc_eif_preds = rep(1, length(C)),
  ipc_est_type = "aipw",
  scale = "identity",
  na.rm = FALSE,
  nuisance_estimators = NULL,
  ...
)

Arguments

fitted_values fitted values from a regression function using the observed data.
y the observed outcome.
a the observed treatment assignment (may be within a specified fold, for cross-fitted estimates). Only used if type = "average_value".
full_y the observed outcome (from the entire dataset, for cross-fitted estimates).
type which parameter are you estimating (defaults to r_squared, for R-squared-based variable importance)?
C the indicator of coarsening (1 denotes observed, 0 denotes unobserved).
Z either NULL (if no coarsening) or a matrix-like object containing the fully observed data.
ipc_weights weights for inverse probability of coarsening (e.g., inverse weights from a two-phase sample) weighted estimation. Assumed to be already inverted (i.e., ipc_weights = 1/[estimated probability weights]).
ipc_fit_type if "external", then use ipc_eif_preds; if "SL", fit a SuperLearner to determine the correction to the efficient influence function.
**est_predictiveness_cv**

Estimate a nonparametric predictiveness functional using cross-fitting

**Description**

Compute nonparametric estimates of the chosen measure of predictiveness.

**Usage**

```r
est_predictiveness_cv(
  fitted_values,
  y,
  full_y = NULL,
  folds,
  type = "r_squared",
  C = rep(1, length(y)),
  Z = NULL,
  folds_Z = folds,
  ipc_weights = rep(1, length(C)),
)```

**Details**

See the paper by Williamson, Gilbert, Simon, and Carone for more details on the mathematics behind this function and the definition of the parameter of interest.

**Value**

A list, with: the estimated predictiveness; the estimated efficient influence function; and the predictions of the EIF based on inverse probability of censoring.


```r
ipc_fit_type = "external",
ipc_eif_preds = rep(1, length(C)),
ipc_est_type = "aipw",
scale = "identity",
na.rm = FALSE,
...)
```

**Arguments**

- `fitted_values` fitted values from a regression function using the observed data; a list of length `V`, where each object is a set of predictions on the validation data, or a vector of the same length as `y`.
- `y` the observed outcome.
- `full_y` the observed outcome (from the entire dataset, for cross-fitted estimates).
- `folds` the cross-validation folds for the observed data.
- `type` which parameter are you estimating (defaults to `r_squared`, for R-squared-based variable importance)?
- `C` the indicator of coarsening (1 denotes observed, 0 denotes unobserved).
- `Z` either `NULL` (if no coarsening) or a matrix-like object containing the fully observed data.
- `folds_Z` either the cross-validation folds for the observed data (no coarsening) or a vector of folds for the fully observed data `Z`.
- `ipc_weights` weights for inverse probability of coarsening (e.g., inverse weights from a two-phase sample) weighted estimation. Assumed to be already inverted (i.e., `ipc_weights = 1 / [estimated probability weights]`).
- `ipc_fit_type` if "external", then use `ipc_eif_preds`; if "SL", fit a SuperLearner to determine the correction to the efficient influence function.
- `ipc_eif_preds` if `ipc_fit_type = "external"`, the fitted values from a regression of the full-data EIF on the fully observed covariates/outcome; otherwise, not used.
- `ipc_est_type` IPC correction, either "ipw" (for classical inverse probability weighting) or "aipw" (for augmented inverse probability weighting; the default).
- `scale` if doing an IPC correction, then the scale that the correction should be computed on (e.g., "identity"; or "logit" to logit-transform, apply the correction, and back-transform).
- `na.rm` logical; should NA's be removed in computation? (defaults to `FALSE`)
- `...` other arguments to SuperLearner, if `ipc_fit_type = "SL"`.

**Details**

See the paper by Williamson, Gilbert, Simon, and Carone for more details on the mathematics behind this function and the definition of the parameter of interest. If sample-splitting is also requested (recommended, since in this case inferences will be valid even if the variable has zero true importance), then the prediction functions are trained as if $2K$-fold cross-validation were run, but are evaluated on only $K$ sets (independent between the full and reduced nuisance regression).
extract_sampled_split_predictions

Value

The estimated measure of predictiveness.

Description

Use the cross-validated Super Learner and a set of specified sample-splitting folds to extract cross-fitted predictions on separate splits of the data. This is primarily for use in cases where you have already fit a CV.SuperLearner and want to use the fitted values to compute variable importance without having to re-fit. The number of folds used in the CV.SuperLearner must be even.

Usage

extract_sampled_split_predictions(
  cvsl_obj = NULL,
  sample_splitting = TRUE,
  sample_splitting_folds = NULL,
  full = TRUE,
  preds = NULL,
  cross_fitting_folds = NULL,
  vector = TRUE
)

Arguments

cvsl_obj An object of class "CV.SuperLearner"; must be entered unless preds is specified.
sample_splitting logical; should we use sample-splitting or not? Defaults to TRUE.
sample_splitting_folds A vector of folds to use for sample splitting
full logical; is this the fit to all covariates (TRUE) or not (FALSE)?
preds a vector of predictions; must be entered unless cvsl_obj is specified.
cross_fitting_folds a vector of folds that were used in cross-fitting.
vector logical; should we return a vector (where each element is the prediction when the corresponding row is in the validation fold) or a list?

Value

The predictions on validation data in each split-sample fold.
See Also

CV.SuperLearner for usage of the CV.SuperLearner function.

---

format.predictiveness_measure

Format a predictiveness_measure object

Description

Nicely formats the output from a predictiveness_measure object for printing.

Usage

```r
## S3 method for class 'predictiveness_measure'
format(x, ...)
```

Arguments

- `x` the predictiveness_measure object of interest.
- `...` other options, see the generic format function.

---

format.vim

Format a vim object

Description

Nicely formats the output from a vim object for printing.

Usage

```r
## S3 method for class 'vim'
format(x, ...)
```

Arguments

- `x` the vim object of interest.
- `...` other options, see the generic format function.
get_cv_sl_folds

Get a numeric vector with cross-validation fold IDs from CV.SuperLearner

Usage

get_cv_sl_folds(cv_sl_folds)

Arguments

cv_sl_folds The folds from a call to CV.SuperLearner; a list.

Value

A numeric vector with the fold IDs.

get_full_type

Obtain the type of VIM to estimate using partial matching

Description

Obtain the type of VIM to estimate using partial matching

Usage

get_full_type(type)

Arguments

type the partial string indicating the type of VIM

Value

the full string indicating the type of VIM
get_test_set  

**Description**

Return test-set only data

**Usage**

```r
get_test_set(arg_lst, k)
```

**Arguments**

- `arg_lst`: a list of estimates, data, etc.
- `k`: the index of interest

**Value**

the test-set only data

make_folds  

**Description**

Create Folds for Cross-Fitting

**Usage**

```r
make_folds(y, V = 2, stratified = FALSE, C = NULL, probs = rep(1/V, V))
```

**Arguments**

- `y`: the outcome
- `V`: the number of folds
- `stratified`: should the folds be stratified based on the outcome?
- `C`: a vector indicating whether or not the observation is fully observed; 1 denotes yes, 0 denotes no
- `probs`: vector of proportions for each fold number

**Value**

a vector of folds
Description

Turn folds from 2K-fold cross-fitting into individual K-fold folds

Usage

```r
make_kfold(
  cross_fitting_folds,
  sample_splitting_folds = rep(1, length(unique(cross_fitting_folds)));
  C = rep(1, length(cross_fitting_folds))
)
```

Arguments

cross_fitting_folds
  the vector of cross-fitting folds

sample_splitting_folds
  the sample splitting folds

C
  vector of whether or not we measured the observation in phase 2

Value

the two sets of testing folds for K-fold cross-fitting

---

Description

Estimate the classification accuracy

Usage

```r
measure_accuracy(
  fitted_values,
  y,
  full_y = NULL,
  C = rep(1, length(y)),
  Z = NULL,
  ipc_weights = rep(1, length(y)),
  ipc_fit_type = "external",
  ipc_eif_preds = rep(1, length(y)),
)```
measure_accuracy

ipc_est_type = "aipw",
scale = "logit",
na.rm = FALSE,
nuisance_estimators = NULL,
a = NULL,
...
)

Arguments

fitted_values fitter values from a regression function using the observed data (may be within a specified fold, for cross-fitted estimates).

y the observed outcome (may be within a specified fold, for cross-fitted estimates).

full_y the observed outcome (not used, defaults to NULL).

C the indicator of coarsening (1 denotes observed, 0 denotes unobserved).

Z either NULL (if no coarsening) or a matrix-like object containing the fully observed data.

ipc_weights weights for inverse probability of coarsening (IPC) (e.g., inverse weights from a two-phase sample) weighted estimation. Assumed to be already inverted. (i.e., ipc_weights = 1 / [estimated probability weights]).

ipc_fit_type if "external", then use ipc_eif_preds; if "SL", fit a SuperLearner to determine the IPC correction to the efficient influence function.

ipc_eif_preds if ipc_fit_type = "external", the fitted values from a regression of the full-data EIF on the fully observed covariates/outcome; otherwise, not used.

ipc_est_type IPC correction, either "ipw" (for classical inverse probability weighting) or "aipw" (for augmented inverse probability weighting; the default).

scale if doing an IPC correction, then the scale that the correction should be computed on (e.g., "identity"; or "logit" to logit-transform, apply the correction, and back-transform).

na.rm logical; should NAs be removed in computation? (defaults to FALSE)

nuisance_estimators not used; for compatibility with measure_average_value.

a not used; for compatibility with measure_average_value.

... other arguments to SuperLearner, if ipc_fit_type = "SL".

Value

A named list of: (1) the estimated classification accuracy of the fitted regression function; (2) the estimated influence function; and (3) the IPC EIF predictions.
**measure_anova**

Estimate ANOVA decomposition-based variable importance.

**Description**

Estimate ANOVA decomposition-based variable importance.

**Usage**

```r
measure_anova(
  full,
  reduced,
  y,
  full_y = NULL,
  C = rep(1, length(y)),
  Z = NULL,
  ipc_weights = rep(1, length(y)),
  ipc_fit_type = "external",
  ipc_eif_preds = rep(1, length(y)),
  ipc_est_type = "aipw",
  scale = "logit",
  na.rm = FALSE,
  nuisance_estimators = NULL,
  a = NULL,
  ...
)
```

**Arguments**

- **full**
  - fitted values from a regression function of the observed outcome on the full set of covariates.
- **reduced**
  - fitted values from a regression on the reduced set of observed covariates.
- **y**
  - the observed outcome (may be within a specified fold, for cross-fitted estimates).
- **full_y**
  - the observed outcome (not used, defaults to NULL).
- **C**
  - the indicator of coarsening (1 denotes observed, 0 denotes unobserved).
- **Z**
  - either NULL (if no coarsening) or a matrix-like object containing the fully observed data.
- **ipc_weights**
  - weights for inverse probability of coarsening (IPC) (e.g., inverse weights from a two-phase sample) weighted estimation. Assumed to be already inverted. (i.e., ipc_weights = 1/[estimated probability weights]).
- **ipc_fit_type**
  - if "external", then use ipc_eif_preds; if "SL", fit a SuperLearner to determine the IPC correction to the efficient influence function.
- **ipc_eif_preds**
  - if ipc_fit_type = "external", the fitted values from a regression of the full-data EIF on the fully observed covariates/outcome; otherwise, not used.
ipc_est_type  
IPC correction, either "ipw" (for classical inverse probability weighting) or "aipw" (for augmented inverse probability weighting; the default).

scale  
if doing an IPC correction, then the scale that the correction should be computed on (e.g., "identity"; or "logit" to logit-transform, apply the correction, and back-transform).

na.rm  
logical; should NAs be removed in computation? (defaults to FALSE)

nuisance_estimators  
not used; for compatibility with measure_average_value.

a  
not used; for compatibility with measure_average_value.

...  
other arguments to SuperLearner, if ipc_fit_type = "SL".

Value

A named list of: (1) the estimated ANOVA (based on a one-step correction) of the fitted regression functions; (2) the estimated influence function; (3) the naive ANOVA estimate; and (4) the IPC EIF predictions.

---

**measure_auc**

*Estimate area under the receiver operating characteristic curve (AUC)*

**Description**

Compute nonparametric estimate of AUC.

**Usage**

```r
measure_auc(
  fitted_values,
  y,
  full_y = NULL,
  C = rep(1, length(y)),
  Z = NULL,
  ipc_weights = rep(1, length(y)),
  ipc_fit_type = "external",
  ipc_eif_preds = rep(1, length(y)),
  ipc_est_type = "aipw",
  scale = "logit",
  na.rm = FALSE,
  nuisance_estimators = NULL,
  a = NULL,
  ...
)
```
Arguments

- **fitted_values**: fitted values from a regression function using the observed data (may be within a specified fold, for cross-fitted estimates).
- **y**: the observed outcome (may be within a specified fold, for cross-fitted estimates).
- **full_y**: the observed outcome (not used, defaults to NULL).
- **C**: the indicator of coarsening (1 denotes observed, 0 denotes unobserved).
- **Z**: either NULL (if no coarsening) or a matrix-like object containing the fully observed data.
- **ipc_weights**: weights for inverse probability of coarsening (IPC) (e.g., inverse weights from a two-phase sample) weighted estimation. Assumed to be already inverted. (i.e., ipc_weights = 1 / [estimated probability weights]).
- **ipc_fit_type**: if "external", then use ipc_eif_preds; if "SL", fit a SuperLearner to determine the IPC correction to the efficient influence function.
- **ipc_eif_preds**: if ipc_fit_type = "external", the fitted values from a regression of the full-data EIF on the fully observed covariates/outcome; otherwise, not used.
- **ipc_est_type**: IPC correction, either "ipw" (for classical inverse probability weighting) or "aipw" (for augmented inverse probability weighting; the default).
- **scale**: if doing an IPC correction, then the scale that the correction should be computed on (e.g., "identity"; or "logit" to logit-transform, apply the correction, and back-transform).
- **na.rm**: logical; should NAs be removed in computation? (defaults to FALSE)
- **nuisance_estimators**: not used; for compatibility with measure_average_value.
- **a**: not used; for compatibility with measure_average_value.
- **...**: other arguments to SuperLearner, if ipc_fit_type = "SL".

Value

A named list of: (1) the estimated AUC of the fitted regression function; (2) the estimated influence function; and (3) the IPC EIF predictions.

---

**measure_average_value**  Estimate the average value under the optimal treatment rule

**Description**

Compute nonparametric estimate of the average value under the optimal treatment rule.
measure_average_value

Usage

measure_average_value(
  nuisance_estimators,
  y, 
  a, 
  full_y = NULL, 
  C = rep(1, length(y)), 
  Z = NULL, 
  ipc_weights = rep(1, length(y)), 
  ipc_fit_type = "external", 
  ipc_eif_preds = rep(1, length(y)), 
  ipc_est_type = "aipw", 
  scale = "identity", 
  na.rm = FALSE, 
  ...
)

Arguments

nuisance_estimators
  a list of nuisance function estimators on the observed data (may be within a specified fold, for cross-fitted estimates). Specifically: an estimator of the optimal treatment rule; an estimator of the propensity score under the estimated optimal treatment rule; and an estimator of the outcome regression when treatment is assigned according to the estimated optimal rule.

y
  the observed outcome (may be within a specified fold, for cross-fitted estimates).

a
  the observed treatment assignment (may be within a specified fold, for cross-fitted estimates).

full_y
  the observed outcome (not used, defaults to NULL).

C
  the indicator of coarsening (1 denotes observed, 0 denotes unobserved).

Z
  either NULL (if no coarsening) or a matrix-like object containing the fully observed data.

ipc_weights
  weights for inverse probability of coarsening (IPC) (e.g., inverse weights from a two-phase sample) weighted estimation. Assumed to be already inverted. (i.e., ipc_weights = 1/[estimated probability weights]).

ipc_fit_type
  if "external", then use ipc_eif_preds; if "SL", fit a SuperLearner to determine the IPC correction to the efficient influence function.

ipc_eif_preds
  if ipc_fit_type = "external", the fitted values from a regression of the full-data EIF on the fully observed covariates/outcome; otherwise, not used.

ipc_est_type
  IPC correction, either "ipw" (for classical inverse probability weighting) or "aipw" (for augmented inverse probability weighting; the default).

scale
  if doing an IPC correction, then the scale that the correction should be computed on (e.g., "identity"; or "logit" to logit-transform, apply the correction, and back-transform).

na.rm
  logical; should NAs be removed in computation? (defaults to FALSE)

... other arguments to SuperLearner, if ipc_fit_type = "SL".
**measure_cross_entropy**

**Value**

A named list of: (1) the estimated classification accuracy of the fitted regression function; (2) the estimated influence function; and (3) the IPC EIF predictions.

---

**Description**

Compute nonparametric estimate of cross-entropy.

**Usage**

```r
measure_cross_entropy(
  fitted_values,
  y,
  full_y = NULL,
  C = rep(1, length(y)),
  Z = NULL,
  ipc_weights = rep(1, length(y)),
  ipc_fit_type = "external",
  ipc_eif_preds = rep(1, length(y)),
  ipc_est_type = "aipw",
  scale = "identity",
  na.rm = FALSE,
  nuisance_estimators = NULL,
  a = NULL,
  ...
)
```

**Arguments**

- `fitted_values`: fitted values from a regression function using the observed data (may be within a specified fold, for cross-fitted estimates).
- `y`: the observed outcome (may be within a specified fold, for cross-fitted estimates).
- `full_y`: the indicator of coarsening (1 denotes observed, 0 denotes unobserved).
- `C`: either NULL (if no coarsening) or a matrix-like object containing the fully observed data.
- `ipc_weights`: weights for inverse probability of coarsening (IPC) (e.g., inverse weights from a two-phase sample) weighted estimation. Assumed to be already inverted. (i.e., `ipc_weights = 1 / [estimated probability weights]).
- `ipc_fit_type`: if "external", then use `ipc_eif_preds`; if "SL", fit a SuperLearner to determine the IPC correction to the efficient influence function.
measure_deviance

ipc_eif_preds if ipc_fit_type = "external", the fitted values from a regression of the full-data EIF on the fully observed covariates/outcome; otherwise, not used.

ipc_est_type IPC correction, either "ipw" (for classical inverse probability weighting) or "aipw" (for augmented inverse probability weighting; the default).

scale if doing an IPC correction, then the scale that the correction should be computed on (e.g., "identity"; or "logit" to logit-transform, apply the correction, and back-transform).

na.rm logical; should NAs be removed in computation? (defaults to FALSE)

nuisance_estimators not used; for compatibility with measure_average_value.

a not used; for compatibility with measure_average_value.

... other arguments to SuperLearner, if ipc_fit_type = "SL".

Value

A named list of: (1) the estimated cross-entropy of the fitted regression function; (2) the estimated influence function; and (3) the IPC EIF predictions.

---

measure_deviance Estimate the deviance

Description

Compute nonparametric estimate of deviance.

Usage

```r
measure_deviance(
  fitted_values,
  y,
  full_y = NULL,
  C = rep(1, length(y)),
  Z = NULL,
  ipc_weights = rep(1, length(y)),
  ipc_fit_type = "external",
  ipc_eif_preds = rep(1, length(y)),
  ipc_est_type = "aipw",
  scale = "logit",
  na.rm = FALSE,
  nuisance_estimators = NULL,
  a = NULL,
  ... )
```
Arguments

- `fitted_values` - fitted values from a regression function using the observed data (may be within a specified fold, for cross-fitted estimates).
- `y` - the observed outcome (may be within a specified fold, for cross-fitted estimates).
- `full_y` - the observed outcome (not used, defaults to NULL).
- `C` - the indicator of coarsening (1 denotes observed, 0 denotes unobserved).
- `Z` - either NULL (if no coarsening) or a matrix-like object containing the fully observed data.
- `ipc_weights` - weights for inverse probability of coarsening (IPC) (e.g., inverse weights from a two-phase sample) weighted estimation. Assumed to be already inverted. (i.e., `ipc_weights = 1 / [estimated probability weights]).
- `ipc_fit_type` - if "external", then use `ipc_eif_preds`; if "SL", fit a SuperLearner to determine the IPC correction to the efficient influence function.
- `ipc_eif_preds` - if `ipc_fit_type = "external"`, the fitted values from a regression of the full-data EIF on the fully observed covariates/outcome; otherwise, not used.
- `ipc_est_type` - IPC correction, either "ipw" (for classical inverse probability weighting) or "aipw" (for augmented inverse probability weighting; the default).
- `scale` - if doing an IPC correction, then the scale that the correction should be computed on (e.g., "identity"; or "logit" to logit-transform, apply the correction, and back-transform).
- `na.rm` - logical; should NAs be removed in computation? (defaults to FALSE)
- `nuisance_estimators` - not used; for compatibility with `measure_average_value`.
- `a` - not used; for compatibility with `measure_average_value`.
- `...` - other arguments to SuperLearner, if `ipc_fit_type = "SL"`.

Value

A named list of: (1) the estimated deviance of the fitted regression function; (2) the estimated influence function; and (3) the IPC EIF predictions.

---

### measure_mse

<table>
<thead>
<tr>
<th>Estimate mean squared error</th>
</tr>
</thead>
</table>

Description

Compute nonparametric estimate of mean squared error.
Usage

measure_mse(
  fitted_values,
  y,
  full_y = NULL,
  C = rep(1, length(y)),
  Z = NULL,
  ipc_weights = rep(1, length(y)),
  ipc_fit_type = "external",
  ipc_eif_preds = rep(1, length(y)),
  ipc_est_type = "aipw",
  scale = "identity",
  na.rm = FALSE,
  nuisance_estimators = NULL,
  a = NULL,
  ...
)

Arguments

fitted_values  fitted values from a regression function using the observed data (may be within a specified fold, for cross-fitted estimates).
y  the observed outcome (may be within a specified fold, for cross-fitted estimates).
full_y  the observed outcome (not used, defaults to NULL).
C  the indicator of coarsening (1 denotes observed, 0 denotes unobserved).
Z  either NULL (if no coarsening) or a matrix-like object containing the fully observed data.
ipc_weights  weights for inverse probability of coarsening (IPC) (e.g., inverse weights from a two-phase sample) weighted estimation. Assumed to be already inverted. (i.e., ipc_weights = 1/[estimated probability weights]).
ipc_fit_type  if "external", then use ipc_eif_preds; if "SL", fit a SuperLearner to determine the IPC correction to the efficient influence function.
ipc_eif_preds  if ipc_fit_type = "external", the fitted values from a regression of the full-data EIF on the fully observed covariates/outcome; otherwise, not used.
ipc_est_type  IPC correction, either "ipw" (for classical inverse probability weighting) or "aipw" (for augmented inverse probability weighting; the default).
scale  if doing an IPC correction, then the scale that the correction should be computed on (e.g., "identity"; or "logit" to logit-transform, apply the correction, and back-transform).
na.rm  logical; should NAs be removed in computation? (defaults to FALSE)
nuisance_estimators  not used; for compatibility with measure_average_value.
a  not used; for compatibility with measure_average_value.
...  other arguments to SuperLearner, if ipc_fit_type = "SL".
**Value**

A named list of: (1) the estimated mean squared error of the fitted regression function; (2) the estimated influence function; and (3) the IPC EIF predictions.

---

**Description**

Estimate R-squared

**Usage**

```r
measure_r_squared(
  fitted_values,
  y,
  full_y = NULL,
  C = rep(1, length(y)),
  Z = NULL,
  ipc_weights = rep(1, length(y)),
  ipc_fit_type = "external",
  ipc_eif_preds = rep(1, length(y)),
  ipc_est_type = "aipw",
  scale = "logit",
  na.rm = FALSE,
  nuisance_estimators = NULL,
  a = NULL,
  ...
)
```

**Arguments**

- **fitted_values**: fitted values from a regression function using the observed data (may be within a specified fold, for cross-fitted estimates).
- **y**: the observed outcome (may be within a specified fold, for cross-fitted estimates).
- **full_y**: the indicator of coarsening (1 denotes observed, 0 denotes unobserved).
- **C**: the indicator of coarsening (1 denotes observed, 0 denotes unobserved).
- **Z**: either NULL (if no coarsening) or a matrix-like object containing the fully observed data.
- **ipc_weights**: weights for inverse probability of coarsening (IPC) (e.g., inverse weights from a two-phase sample) weighted estimation. Assumed to be already inverted. (i.e., \( \text{ipc_weights} = 1 / \{\text{estimated probability weights}\} \)).
- **ipc_fit_type**: if "external", then use ipc_eif_preds; if "SL", fit a SuperLearner to determine the IPC correction to the efficient influence function.
merge_vim

### Description
Take the output from multiple different calls to vimp_regression and merge into a single vim object; mostly used for plotting results.

### Usage
```r
merge_vim(...)```

### Arguments
```r
... an arbitrary number of vim objects, separated by commas.
```

### Value
an object of class vim containing all of the output from the individual vim objects. This results in a list containing:
- `s` - a list of the column(s) to calculate variable importance for
- `SL.library` - a list of the libraries of learners passed to SuperLearner
- `full_fit` - a list of the fitted values of the chosen method fit to the full data
- `red_fit` - a list of the fitted values of the chosen method fit to the reduced data
- `est` - a vector with the corrected estimates
- `naive` - a vector with the naive estimates
Construct a Predictiveness Measure

Description

Construct a Predictiveness Measure

Examples

```r
# generate the data
# generate X
p <- 2
n <- 100
x <- data.frame(replicate(p, stats::runif(n, -5, 5)))

# apply the function to the x's
smooth <- (x[,1]/5)^2*(x[,1]+7)/5 + (x[,2]/3)^2

# generate Y ~ Normal (smooth, 1)
y <- smooth + stats::rnorm(n, 0, 1)

# set up a library for SuperLearner; note simple library for speed
library("SuperLearner")
learners <- c("SL.glm", "SL.mean")

# using Super Learner (with a small number of folds, for illustration only)
est_2 <- vimp_regression(Y = y, X = x, indx = 2, V = 2,
                         run_regression = TRUE, alpha = 0.05,
                         SL.library = learners, cvControl = list(V = 2))
est_1 <- vimp_regression(Y = y, X = x, indx = 1, V = 2,
                         run_regression = TRUE, alpha = 0.05,
                         SL.library = learners, cvControl = list(V = 2))
est <- merge_vim(est_1, est_2)
```
predictiveness_measure

Usage

predictiveness_measure(
  type = character(),
  y = numeric(),
  a = numeric(),
  fitted_values = numeric(),
  cross_fitting_folds = rep(1, length(fitted_values)),
  full_y = NULL,
  nuisance_estimators = list(),
  C = rep(1, length(y)),
  Z = NULL,
  folds_Z = cross_fitting_folds,
  ipc_weights = rep(1, length(y)),
  ipc_fit_type = "SL",
  ipc_eif_preds = numeric(),
  ipc_est_type = "aipw",
  scale = "identity",
  na.rm = TRUE,
  ...
)

Arguments

type  the measure of interest (e.g., "accuracy", "auc", "r_squared")
y     the outcome of interest
a     the exposure of interest (only used if type = "average_value")
fitted_values fitted values from a regression function using the observed data (may be within
                 a specified fold, for cross-fitted estimates).
cross_fitting_folds folds for cross-fitting, if used to obtain the fitted values. If not used, a vector of
                      ones.
full_y the observed outcome (not used, defaults to NULL).
nuisance_estimators a list of nuisance function estimators on the observed data (may be within
                      a specified fold, for cross-fitted estimates). For the average value measure: an es-
                      timator of the optimal treatment rule (f_n); an estimator of the propensity score
                      under the estimated optimal treatment rule (g_n); and an estimator of the out-
                      come regression when treatment is assigned according to the estimated optimal
                      rule (q_n).
C      the indicator of coarsening (1 denotes observed, 0 denotes unobserved).
Z      either NULL (if no coarsening) or a matrix-like object containing the fully ob-
       served data.
folds_Z either the cross-validation folds for the observed data (no coarsening) or a vector
           of folds for the fully observed data Z.
print.predictiveness_measure

ipc_weights weights for inverse probability of coarsening (IPC) (e.g., inverse weights from a two-phase sample) weighted estimation. Assumed to be already inverted. (i.e., ipc_weights = 1 / [estimated probability weights]).

ipc_fit_type if "external", then use ipc_eif_preds; if "SL", fit a SuperLearner to determine the IPC correction to the efficient influence function.

ipc_eif_preds if ipc_fit_type = "external", the fitted values from a regression of the full-data EIF on the fully observed covariates/outcome; otherwise, not used.

ipc_est_type IPC correction, either "ipw" (for classical inverse probability weighting) or "aipw" (for augmented inverse probability weighting; the default).

scale if doing an IPC correction, then the scale that the correction should be computed on (e.g., "identity"; or "logit" to logit-transform, apply the correction, and back-transform).

na.rm logical; should NAs be removed in computation? (defaults to FALSE)

... other arguments to SuperLearner, if ipc_fit_type = "SL".

Value

An object of class "predictiveness_measure", with the following attributes:

print.predictiveness_measure

Print predictiveness_measure objects

Description

Prints out a table of the point estimate and standard error for a predictiveness_measure object.

Usage

## S3 method for class 'predictiveness_measure'
print(x, ...)

Arguments

x the predictiveness_measure object of interest.

... other options, see the generic print function.
print.vim  

Description

Prints out the table of estimates, confidence intervals, and standard errors for a `vim` object.

Usage

```r
## S3 method for class 'vim'
print(x, ...)
```

Arguments

- `x`  
  the `vim` object of interest.
- `...`  
  other options, see the generic `print` function.

process_arg_lst  

Description

Process argument list for Super Learner estimation of the EIF

Usage

```r
process_arg_lst(arg_lst)
```

Arguments

- `arg_lst`  
  the list of arguments for Super Learner

Value

a list of modified arguments for EIF estimation
Run a Super Learner for the provided subset of features

Usage

run_sl(
  Y = NULL,
  X = NULL,
  V = 5,
  SL.library = "SL.glm",
  univariate_SL.library = NULL,
  s = 1,
  cv_folds = NULL,
  sample_splitting = TRUE,
  ss_folds = NULL,
  split = 1,
  verbose = FALSE,
  progress_bar = NULL,
  indx = 1,
  weights = rep(1, nrow(X)),
  cross_fitted_se = TRUE,
  full = NULL,
  vector = TRUE,
  ...
)

Arguments

Y the outcome
X the covariates
V the number of folds
SL.library the library of candidate learners
univariate_SL.library the library of candidate learners for single-covariate regressions
s the subset of interest
cv_folds the CV folds
sample_splitting logical; should we use sample-splitting for predictiveness estimation?
ss_folds the sample-splitting folds; only used if sample_splitting = TRUE
split the split to use for sample-splitting; only used if sample_splitting = TRUE
sample_subsets

Create necessary objects for SPVIMs

Description

Creates the Z and W matrices and a list of sampled subsets, S, for SPVIM estimation.

Usage

sample_subsets(p, gamma, n)

Arguments

p
the number of covariates

gamma
the fraction of the sample size to sample (e.g., gamma = 1 means sample n subsets)

n
the sample size

Value

a list, with elements Z (the matrix encoding presence/absence of each feature in the uniquely sampled subsets), S (the list of unique sampled subsets), W (the matrix of weights), and z_counts (the number of times each subset was sampled)

Examples

p <- 10
gamma <- 1
n <- 100
set.seed(100)
subset_lst <- sample_subsets(p, gamma, n)
**scale_est**  
*Return an estimator on a different scale*

**Description**  
Return an estimator on a different scale

**Usage**  

```r  
scale_est(obs_est = NULL, grad = NULL, scale = "identity")  
```

**Arguments**

- `obs_est`: the observed VIM estimate
- `grad`: the estimated efficient influence function
- `scale`: the scale to compute on

**Details**

It may be of interest to return an estimate (or confidence interval) on a different scale than originally measured. For example, computing a confidence interval (CI) for a VIM value that lies in (0,1) on the logit scale ensures that the CI also lies in (0,1).

**Value**

the scaled estimate

---

**spvim_ics**  
*Influence function estimates for SPVIMs*

**Description**

Compute the influence functions for the contribution from sampling observations and subsets.

**Usage**

```r  
spvim_ics(Z, z_counts, W, v, psi, G, c_n, ics, measure)  
```
Arguments
- **Z**: the matrix of presence/absence of each feature (columns) in each sampled subset (rows)
- **z_counts**: the number of times each unique subset was sampled
- **W**: the matrix of weights
- **v**: the estimated predictiveness measures
- **psi**: the estimated SPVIM values
- **G**: the constraint matrix
- **c_n**: the constraint values
- **ics**: a list of influence function values for each predictiveness measure
- **measure**: the type of measure (e.g., "r_squared" or "auc")

Details
The processes for sampling observations and sampling subsets are independent. Thus, we can compute the influence function separately for each sampling process. For further details, see the paper by Williamson and Feng (2020).

Value
a named list of length 2; `contrib_v` is the contribution from estimating V, while `contrib_s` is the contribution from sampling subsets.

---

**spvim_se**

*Standard error estimate for SPVIM values*

Description
Compute standard error estimates based on the estimated influence function for a SPVIM value of interest.

Usage
`spvim_se(ics, idx = 1, gamma = 1, na.rm = FALSE)`

Arguments
- **ics**: the influence function estimates based on the contributions from sampling observations and sampling subsets: a list of length two resulting from a call to `spvim_ics`
- **idx**: the index of interest
- **gamma**: the proportion of the sample size used when sampling subsets
- **na_rm**: remove NAs?
Details

Since the processes for sampling observations and subsets are independent, the variance for a given SPVIM estimator is simply the sum of the variances based on sampling observations and on sampling subsets.

Value

The standard error estimate for the desired SPVIM value

See Also

spvim_ics for how the influence functions are estimated.

sp_vim

Shapley Population Variable Importance Measure (SPVIM) Estimates and Inference

Description

Compute estimates and confidence intervals for the SPVIMs, using cross-fitting.

Usage

sp_vim(
  Y = NULL,
  X = NULL,
  V = 5,
  type = "r_squared",
  SL.library = c("SL.glmnet", "SL.xgboost", "SL.mean"),
  univariate_SL.library = NULL,
  gamma = 1,
  alpha = 0.05,
  delta = 0,
  na.rm = FALSE,
  stratified = FALSE,
  verbose = FALSE,
  sample_splitting = TRUE,
  final_point_estimate = "split",
  C = rep(1, length(Y)),
  Z = NULL,
  ipc_scale = "identity",
  ipc_weights = rep(1, length(Y)),
  ipc_est_type = "aipw",
  scale = "identity",
  scale_est = TRUE,
  cross_fitted_se = TRUE,
  ...
)
Arguments

Y  the outcome.
X  the covariates. If type = "average_value", then the exposure variable should be part of X, with its name provided in exposure_name.
V  the number of folds for cross-fitting, defaults to 5. If sample_splitting = TRUE, then a special type of V-fold cross-fitting is done. See Details for a more detailed explanation.
type the type of importance to compute; defaults to r_squared, but other supported options are auc, accuracy, deviance, and anova.
SL.library a character vector of learners to pass to SuperLearner, if f1 and f2 are Y and X, respectively. Defaults to SL.glmnet, SL.xgboost, and SL.mean.
univariate_SL.library (optional) a character vector of learners to pass to SuperLearner for estimating univariate regression functions. Defaults to SL.polymars
gamma the fraction of the sample size to use when sampling subsets (e.g., gamma = 1 samples the same number of subsets as the sample size)
alpha the level to compute the confidence interval at. Defaults to 0.05, corresponding to a 95% confidence interval.
delta the value of the δ-null (i.e., testing if importance < δ); defaults to 0.
na.rm should we remove NAs in the outcome and fitted values in computation? (defaults to FALSE)
stratified if run_regression = TRUE, then should the generated folds be stratified based on the outcome (helps to ensure class balance across cross-validation folds)
verbose should sp_vim and SuperLearner print out progress? (defaults to FALSE)
sample_splitting should we use sample-splitting to estimate the full and reduced predictiveness? Defaults to TRUE, since inferences made using sample_splitting = FALSE will be invalid for variables with truly zero importance.
final_point_estimate if sample splitting is used, should the final point estimates be based on only the sample-split folds used for inference ("split", the default), or should they instead be based on the full dataset ("full") or the average across the point estimates from each sample split ("average")? All three options result in valid point estimates – sample-splitting is only required for valid inference.
C the indicator of coarsening (1 denotes observed, 0 denotes unobserved).
Z either (i) NULL (the default, in which case the argument C above must be all ones), or (ii) a character vector specifying the variable(s) among Y and X that are thought to play a role in the coarsening mechanism. To specify the outcome, use "Y"; to specify covariates, use a character number corresponding to the desired position in X (e.g., "1").
ipc_scale what scale should the inverse probability weight correction be applied on (if any)? Defaults to "identity". (other options are "log" and "logit")
sp_vim 47

ipc_weights weights for the computed influence curve (i.e., inverse probability weights for coarsened-at-random settings). Assumed to be already inverted (i.e., ipc_weights = 1 / [estimated probability weights]).

ipc_est_type the type of procedure used for coarsened-at-random settings; options are "ipw" (for inverse probability weighting) or "aipw" (for augmented inverse probability weighting). Only used if C is not all equal to 1.

scale should CIs be computed on original ("identity") or another scale? (options are "log" and "logit")

scale_est should the point estimate be scaled to be greater than or equal to 0? Defaults to TRUE.

cross_fitted_se should we use cross-fitting to estimate the standard errors (TRUE, the default) or not (FALSE)?

... other arguments to the estimation tool, see "See also".

Details

We define the SPVIM as the weighted average of the population difference in predictiveness over all subsets of features not containing feature j.

This is equivalent to finding the solution to a population weighted least squares problem. This key fact allows us to estimate the SPVIM using weighted least squares, where we first sample subsets from the power set of all possible features using the Shapley sampling distribution; then use cross-fitting to obtain estimators of the predictiveness of each sampled subset; and finally, solve the least squares problem given in Williamson and Feng (2020).

See the paper by Williamson and Feng (2020) for more details on the mathematics behind this function, and the validity of the confidence intervals.

In the interest of transparency, we return most of the calculations within the vim object. This results in a list containing:

SL.library the library of learners passed to SuperLearner

v the estimated predictiveness measure for each sampled subset

fit_lst the fitted values on the entire dataset from the chosen method for each sampled subset

preds_lst the cross-fitted predicted values from the chosen method for each sampled subset

est the estimated SPVIM value for each feature

ics the influence functions for each sampled subset

var_v_contribs the contributions to the variance from estimating predictiveness

var_s_contribs the contributions to the variance from sampling subsets

ic_lst a list of the SPVIM influence function contributions

se the standard errors for the estimated variable importance

ci the \((1 - \alpha) \times 100\%\) confidence intervals based on the variable importance estimates

p_value p-values for the null hypothesis test of zero importance for each variable

test_statistic the test statistic for each null hypothesis test of zero importance
test  a hypothesis testing decision for each null hypothesis test (for each variable having zero importance)

gamma  the fraction of the sample size used when sampling subsets

alpha  the level, for confidence interval calculation

delta  the delta value used for hypothesis testing

y  the outcome

ipc_weights  the weights

scale  the scale on which CIs were computed

mat  - a tibble with the estimates, SEs, CIs, hypothesis testing decisions, and p-values

Value

An object of class vim. See Details for more information.

See Also

SuperLearner for specific usage of the SuperLearner function and package.

Examples

n <- 100
p <- 2
# generate the data
x <- data.frame(replicate(p, stats::runif(n, -5, 5)))

# apply the function to the x's
smooth <- (x[,1]/5)^2*(x[,1]+7)/5 + (x[,2]/3)^2

# generate Y ~ Normal (smooth, 1)
y <- as.matrix(smooth + stats::rnorm(n, 0, 1))

# set up a library for SuperLearner; note simple library for speed
library("SuperLearner")
learners <- c("SL.glm")

# -----------------------------
# using Super Learner (with a small number of CV folds, 
# for illustration only) 
# -----------------------------
set.seed(4747)
est <- sp_vim(Y = y, X = x, V = 2, type = "r_squared", 
SL.library = learners, alpha = 0.05)
Nonparametric Intrinsic Variable Importance Estimates and Inference

Description

Compute estimates of and confidence intervals for nonparametric intrinsic variable importance based on the population-level contrast between the oracle predictiveness using the feature(s) of interest versus not.

Usage

```r
vim(
  Y = NULL,
  X = NULL,
  f1 = NULL,
  f2 = NULL,
  indx = 1,
  type = "r_squared",
  run_regression = TRUE,
  SL.library = c("SL.glmnet", "SL.xgboost", "SL.mean"),
  alpha = 0.05,
  delta = 0,
  scale = "identity",
  na.rm = FALSE,
  sample_splitting = TRUE,
  sample_splitting_folds = NULL,
  final_point_estimate = "split",
  stratified = FALSE,
  C = rep(1, length(Y)),
  Z = NULL,
  ipc_scale = "identity",
  ipc_weights = rep(1, length(Y)),
  ipc_est_type = "aipw",
  scale_est = TRUE,
  nuisance_estimators_full = NULL,
  nuisance_estimators_reduced = NULL,
  exposure_name = NULL,
  bootstrap = FALSE,
  b = 1000,
  boot_interval_type = "perc",
  ...
)
```

Arguments

- `Y` the outcome.
X the covariates. If type = "average_value", then the exposure variable should be part of \(X\), with its name provided in exposure_name.

f1 the fitted values from a flexible estimation technique regressing \(Y\) on \(X\). A vector of the same length as \(Y\); if sample-splitting is desired, then the value of \(f1\) at each position should be the result of predicting from a model trained without that observation.

f2 the fitted values from a flexible estimation technique regressing either (a) \(f1\) or (b) \(Y\) on \(X\) withholding the columns in indx. A vector of the same length as \(Y\); if sample-splitting is desired, then the value of \(f2\) at each position should be the result of predicting from a model trained without that observation.

indx the indices of the covariate(s) to calculate variable importance for; defaults to 1.

type the type of importance to compute; defaults to r_squared, but other supported options are auc, accuracy, deviance, and anova.

run_regression if outcome \(Y\) and covariates \(X\) are passed to vimp_accuracy, and run_regression is TRUE, then Super Learner will be used; otherwise, variable importance will be computed using the inputted fitted values.

SL.library a character vector of learners to pass to SuperLearner, if \(f1\) and \(f2\) are \(Y\) and \(X\), respectively. Defaults to SL.glmnet, SL.xgboost, and SL.mean.

alpha the level to compute the confidence interval at. Defaults to 0.05, corresponding to a 95% confidence interval.

delta the value of the \(\delta\)-null (i.e., testing if importance < \(\delta\)); defaults to 0.

scale should CIs be computed on original ("identity") or another scale? (options are "log" and "logit")

na.rm should we remove NAs in the outcome and fitted values in computation? (defaults to FALSE)

sample_splitting should we use sample-splitting to estimate the full and reduced predictiveness? Defaults to TRUE, since inferences made using sample_splitting = FALSE will be invalid for variables with truly zero importance.

sample_splitting_folds the folds used for sample-splitting; these identify the observations that should be used to evaluate predictiveness based on the full and reduced sets of covariates, respectively. Only used if run_regression = FALSE.

final_point_estimate if sample splitting is used, should the final point estimates be based on only the sample-split folds used for inference ("split", the default), or should they instead be based on the full dataset ("full") or the average across the point estimates from each sample split ("average")? All three options result in valid point estimates – sample-splitting is only required for valid inference.

stratified if run_regression = TRUE, then should the generated folds be stratified based on the outcome (helps to ensure class balance across cross-validation folds)

C the indicator of coarsening (1 denotes observed, 0 denotes unobserved).

Z either (i) NULL (the default, in which case the argument C above must be all ones), or (ii) a character vector specifying the variable(s) among \(Y\) and \(X\) that are
thought to play a role in the coarsening mechanism. To specify the outcome, use "Y"; to specify covariates, use a character number corresponding to the desired position in X (e.g., "1").

**ipc.scale** what scale should the inverse probability weight correction be applied on (if any)? Defaults to "identity". (other options are "log" and "logit")

**ipc.weights** weights for the computed influence curve (i.e., inverse probability weights for coarsened-at-random settings). Assumed to be already inverted (i.e., ipc.weights = 1 / [estimated probability weights]).

**ipc.est.type** the type of procedure used for coarsened-at-random settings; options are "ipw" (for inverse probability weighting) or "aipw" (for augmented inverse probability weighting). Only used if C is not all equal to 1.

**scale_est** should the point estimate be scaled to be greater than or equal to 0? Defaults to TRUE.

**nuisance_estimators_full** (only used if type = "average_value") a list of nuisance function estimators on the observed data (may be within a specified fold, for cross-fitted estimates). Specifically: an estimator of the optimal treatment rule; an estimator of the propensity score under the estimated optimal treatment rule; and an estimator of the outcome regression when treatment is assigned according to the estimated optimal rule.

**nuisance_estimators_reduced** (only used if type = "average_value") a list of nuisance function estimators on the observed data (may be within a specified fold, for cross-fitted estimates). Specifically: an estimator of the optimal treatment rule; an estimator of the propensity score under the estimated optimal treatment rule; and an estimator of the outcome regression when treatment is assigned according to the estimated optimal rule.

**exposure_name** (only used if type = "average_value") the name of the exposure of interest; binary, with 1 indicating presence of the exposure and 0 indicating absence of the exposure.

**bootstrap** should bootstrap-based standard error estimates be computed? Defaults to FALSE (and currently may only be used if sample_splitting = FALSE).

**b** the number of bootstrap replicates (only used if bootstrap = TRUE and sample_splitting = FALSE); defaults to 1000.

**boot.interval_type** the type of bootstrap interval (one of "norm", "basic", "stud", "perc", or "bca", as in boot(boot.ci)) if requested. Defaults to "perc".

... other arguments to the estimation tool, see "See also".

**Details**

We define the population variable importance measure (VIM) for the group of features (or single feature) $s$ with respect to the predictiveness measure $V$ by

$$
\psi_{0,s} := V(f_0, P_0) - V(f_{0,s}, P_0),
$$
where $f_0$ is the population predictiveness maximizing function, $f_{0,s}$ is the population predictiveness maximizing function that is only allowed to access the features with index not in $s$, and $P_0$ is the true data-generating distribution. VIM estimates are obtained by obtaining estimators $f_n$ and $f_{n,s}$ of $f_0$ and $f_{0,s}$, respectively; obtaining an estimator $P_n$ of $P_0$; and finally, setting $\psi_{n,s} := V(f_n,P_n) - V(f_{n,s},P_n)$.

In the interest of transparency, we return most of the calculations within the vim object. This results in a list including:

- `s` the column(s) to calculate variable importance for
- `SL.library` the library of learners passed to `SuperLearner`
- `type` the type of risk-based variable importance measured
- `full_fit` the fitted values of the chosen method fit to the full data
- `red_fit` the fitted values of the chosen method fit to the reduced data
- `est` the estimated variable importance
- `naive` the naive estimator of variable importance (only used if `type = "anova"`)
- `eif` the estimated efficient influence function
- `eif_full` the estimated efficient influence function for the full regression
- `eif_reduced` the estimated efficient influence function for the reduced regression
- `se` the standard error for the estimated variable importance
- `ci` the $(1 - \alpha) \times 100\%$ confidence interval for the variable importance estimate
- `test` a decision to either reject (TRUE) or not reject (FALSE) the null hypothesis, based on a conservative test
- `p_value` a p-value based on the same test as `test`
- `full_mod` the object returned by the estimation procedure for the full data regression (if applicable)
- `red_mod` the object returned by the estimation procedure for the reduced data regression (if applicable)
- `alpha` the level, for confidence interval calculation
- `sample_splitting_folds` the folds used for sample-splitting (used for hypothesis testing)
- `y` the outcome
- `ipc_weights` the weights
- `mat` a tibble with the estimate, SE, CI, hypothesis testing decision, and p-value

**Value**

An object of classes `vim` and the type of risk-based measure. See Details for more information.

**See Also**

`SuperLearner` for specific usage of the `SuperLearner` function and package.
Examples

```r
# generate the data
# generate X
p <- 2
n <- 100
x <- data.frame(replicate(p, stats::runif(n, -1, 1)))

# apply the function to the x's
f <- function(x) 0.5 + 0.3*x[1] + 0.2*x[2]
smooth <- apply(x, 1, function(z) f(z))

# generate Y ~ Bernoulli (smooth)
y <- matrix(rbinom(n, size = 1, prob = smooth))

# set up a library for SuperLearner; note simple library for speed
library("SuperLearner")
learners <- c("SL.glm")

# using Y and X; use class-balanced folds
est_1 <- vim(y, x, indx = 2, type = "accuracy",
alpha = 0.05, run_regression = TRUE,
SL.library = learners, cvControl = list(V = 2),
stratified = TRUE)

# using pre-computed fitted values
set.seed(4747)
V <- 2
full_fit <- SuperLearner::CV.SuperLearner(Y = y, X = x,
SL.library = learners,
cvControl = list(V = 2),
innerCvControl = list(list(V = V)))
full_fitted <- SuperLearner::predict.SuperLearner(full_fit)$pred
# fit the data with only X1
reduced_fit <- SuperLearner::CV.SuperLearner(Y = full_fitted,
X = x[, -2, drop = FALSE],
SL.library = learners,
cvControl = list(V = 2, validRows = full_fit$folds),
innerCvControl = list(list(V = V)))
reduced_fitted <- SuperLearner::predict.SuperLearner(reduced_fit)$pred

est_2 <- vim(Y = y, f1 = full_fitted, f2 = reduced_fitted,
indx = 2, run_regression = FALSE, alpha = 0.05,
stratified = TRUE, type = "accuracy",
sample_splitting_folds = get_cv_sl_folds(full_fit$folds))
```
Description

A unified framework for valid statistical inference on algorithm-agnostic measures of intrinsic variable importance. You provide the data, a method for estimating the conditional mean of the outcome given the covariates, choose a variable importance measure, and specify variable(s) of interest; 'vimp' takes care of the rest.

Author(s)


Methodology authors:

• Brian D. Williamson
• Jean Feng
• Peter B. Gilbert
• Noah R. Simon
• Marco Carone

See Also

Manuscripts:

• doi: 10.1111/biom.13392 (R-squared-based variable importance)
• doi: 10.1111/biom.13389 (Rejoinder to discussion on R-squared-based variable importance article)
• http://proceedings.mlr.press/v119/williamson20a.html (general Shapley-based variable importance)
• doi: 10.1080/01621459.2021.2003200 (general variable importance)

Other useful links:

• https://bdwilliamson.github.io/vimp/
• https://github.com/bdwilliamson/vimp
• Report bugs at https://github.com/bdwilliamson/vimp/issues

Imports

The packages that we import either make the internal code nice (dplyr, magrittr, tibble, rlang, MASS, data.table), are directly relevant to estimating the conditional mean (SuperLearner) or predictiveness measures (ROCR), or are necessary for hypothesis testing (stats) or confidence intervals (boot, only for bootstrap intervals).

We suggest several other packages: xgboost, ranger, gam, glmnet, polspline, and quadprog allow a flexible library of candidate learners in the Super Learner; ggplot2 and cowplot help with plotting variable importance estimates; testthat and covr help with unit tests; and knitr, rmarkdown, and tidyselect help with the vignettes and examples.
vimp_accuracy

Nonparametric Intrinsic Variable Importance Estimates: Classification accuracy

Description

Compute estimates of and confidence intervals for nonparametric difference in classification accuracy-based intrinsic variable importance. This is a wrapper function for cv_vim, with type = "accuracy".

Usage

vimp_accuracy(
  Y = NULL,
  X = NULL,
  cross_fitted_f1 = NULL,
  cross_fitted_f2 = NULL,
  f1 = NULL,
  f2 = NULL,
  indx = 1,
  V = 10,
  run_regression = TRUE,
  SL.library = c("SL.glmnet", "SL.xgboost", "SL.mean"),
  alpha = 0.05,
  delta = 0,
  na.rm = FALSE,
  final_point_estimate = "split",
  cross_fitting_folds = NULL,
  sample_splitting_folds = NULL,
  stratified = TRUE,
  C = rep(1, length(Y)),
  Z = NULL,
  ipc_weights = rep(1, length(Y)),
  scale = "logit",
  ipc_est_type = "aipw",
  scale_est = TRUE,
  cross_fitted_se = TRUE,
  ...
)

Arguments

Y the outcome.
X the covariates. If type = "average_value", then the exposure variable should be part of X, with its name provided in exposure_name.
cross_fitted_f1 the predicted values on validation data from a flexible estimation technique regressing Y on X in the training data. Provided as either (a) a vector, where each
element is the predicted value when that observation is part of the validation fold; or (b) a list of length \( V \), where each element in the list is a set of predictions on the corresponding validation data fold. If sample-splitting is requested, then these must be estimated specially; see Details. However, the resulting vector should be the same length as \( Y \); if using a list, then the summed length of each element across the list should be the same length as \( Y \) (i.e., each observation is included in the predictions).

\text{cross\_fitted\_f2}

the predicted values on validation data from a flexible estimation technique regressing either (a) the fitted values in \text{cross\_fitted\_f1}, or (b) \( Y \), on \( X \) withholding the columns in \text{indx}. Provided as either (a) a vector, where each element is the predicted value when that observation is part of the validation fold; or (b) a list of length \( V \), where each element in the list is a set of predictions on the corresponding validation data fold. If sample-splitting is requested, then these must be estimated specially; see Details. However, the resulting vector should be the same length as \( Y \); if using a list, then the summed length of each element across the list should be the same length as \( Y \) (i.e., each observation is included in the predictions).

\text{f1}

the fitted values from a flexible estimation technique regressing \( Y \) on \( X \). If sample-splitting is requested, then these must be estimated specially; see Details. If \text{cross\_fitted\_se} = \text{TRUE}, then this argument is not used.

\text{f2}

the fitted values from a flexible estimation technique regressing either (a) \text{f1} or (b) \( Y \) on \( X \) withholding the columns in \text{indx}. If sample-splitting is requested, then these must be estimated specially; see Details. If \text{cross\_fitted\_se} = \text{TRUE}, then this argument is not used.

\text{indx}

the indices of the covariate(s) to calculate variable importance for; defaults to 1.

\( V \)

the number of folds for cross-fitting, defaults to 5. If \text{sample\_splitting} = \text{TRUE}, then a special type of \( V \)-fold cross-fitting is done. See Details for a more detailed explanation.

\text{run\_regression}

if outcome \( Y \) and covariates \( X \) are passed to \text{vimp\_accuracy}, and \text{run\_regression} is \text{TRUE}, then Super Learner will be used; otherwise, variable importance will be computed using the inputted fitted values.

\text{SL.library}

a character vector of learners to pass to SuperLearner, if \text{f1} and \text{f2} are \( Y \) and \( X \), respectively. Defaults to \text{SL.glmnet}, \text{SL.xgboost}, and \text{SL.mean}.

\text{alpha}

the level to compute the confidence interval at. Defaults to 0.05, corresponding to a 95% confidence interval.

\text{delta}

the value of the \( \delta \)-null (i.e., testing if importance < \( \delta \)); defaults to 0.

\text{na.rm}

should we remove NAs in the outcome and fitted values in computation? (defaults to \text{FALSE})

\text{final\_point\_estimate}

if sample splitting is used, should the final point estimates be based on only the sample-split folds used for inference ("split", the default), or should they instead be based on the full dataset ("full") or the average across the point estimates from each sample split ("average")? All three options result in valid point estimates – sample-splitting is only required for valid inference.
cross_fitting_folds  
the folds for cross-fitting. Only used if run_regression = FALSE.

sample_splitting_folds  
the folds used for sample-splitting; these identify the observations that should be used to evaluate predictiveness based on the full and reduced sets of covariates, respectively. Only used if run_regression = FALSE.

stratified  
if run_regression = TRUE, then should the generated folds be stratified based on the outcome (helps to ensure class balance across cross-validation folds)

C  
the indicator of coarsening (1 denotes observed, 0 denotes unobserved).

Z  
either (i) NULL (the default, in which case the argument C above must be all ones), or (ii) a character vector specifying the variable(s) among Y and X that are thought to play a role in the coarsening mechanism. To specify the outcome, use "Y"; to specify covariates, use a character number corresponding to the desired position in X (e.g., "1").

ipc_weights  
weights for the computed influence curve (i.e., inverse probability weights for coarsened-at-random settings). Assumed to be already inverted (i.e., ipc_weights = 1 / [estimated probability weights]).

scale  
should CIs be computed on original ("identity") or another scale? (options are "log" and "logit")

ipc_est_type  
the type of procedure used for coarsened-at-random settings; options are "ipw" (for inverse probability weighting) or "aipw" (for augmented inverse probability weighting). Only used if C is not all equal to 1.

scale_est  
should the point estimate be scaled to be greater than or equal to 0? Defaults to TRUE.

cross_fitted_se  
should we use cross-fitting to estimate the standard errors (TRUE, the default) or not (FALSE)?

...  
other arguments to the estimation tool, see "See also".

Details

We define the population variable importance measure (VIM) for the group of features (or single feature) s with respect to the predictiveness measure V by

$$\psi_{0,s} := V(f_0, P_0) - V(f_{0,s}, P_0),$$

where $f_0$ is the population predictiveness maximizing function, $f_{0,s}$ is the population predictiveness maximizing function that is only allowed to access the features with index not in s, and $P_0$ is the true data-generating distribution.

Cross-fitted VIM estimates are computed differently if sample-splitting is requested versus if it is not. We recommend using sample-splitting in most cases, since only in this case will inferences be valid if the variable(s) of interest have truly zero population importance. The purpose of cross-fitting is to estimate $f_0$ and $f_{0,s}$ on independent data from estimating $P_0$; this can result in improved performance, especially when using flexible learning algorithms. The purpose of sample-splitting is to estimate $f_0$ and $f_{0,s}$ on independent data; this allows valid inference under the null hypothesis of zero importance.
Without sample-splitting, cross-fitted VIM estimates are obtained by first splitting the data into $K$ folds; then using each fold in turn as a hold-out set, constructing estimators $f_{n,k}$ and $f_{n,k,s}$ of $f_0$ and $f_{0,s}$, respectively on the training data and estimator $P_{n,k}$ of $P_0$ using the test data; and finally, computing

$$\psi_{n,s} := K^{(-1)} \sum_{k=1}^{K} \{ V(f_{n,k}, P_{n,k}) - V(f_{n,k,s}, P_{n,k}) \}.$$  

With sample-splitting, cross-fitted VIM estimates are obtained by first splitting the data into $2K$ folds. These folds are further divided into 2 groups of folds. Then, for each fold $k$ in the first group, estimator $f_{n,k}$ of $f_0$ is constructed using all data besides the $k$th fold in the group (i.e., $(2K - 1)/(2K)$ of the data) and estimator $P_{n,k}$ of $P_0$ is constructed using the held-out data (i.e., $1/2K$ of the data); then, computing

$$v_{n,k} = V(f_{n,k}, P_{n,k}).$$

Similarly, for each fold $k$ in the second group, estimator $f_{n,k,s}$ of $f_{0,s}$ is constructed using all data besides the $k$th fold in the group (i.e., $(2K - 1)/(2K)$ of the data) and estimator $P_{n,k}$ of $P_0$ is constructed using the held-out data (i.e., $1/2K$ of the data); then, computing

$$v_{n,k,s} = V(f_{n,k,s}, P_{n,k}).$$

Finally,

$$\psi_{n,s} := K^{(-1)} \sum_{k=1}^{K} \{ v_{n,k} - v_{n,k,s} \}.$$  

See the paper by Williamson, Gilbert, Simon, and Carone for more details on the mathematics behind the \texttt{cv_vim} function, and the validity of the confidence intervals.

In the interest of transparency, we return most of the calculations within the \texttt{vim} object. This results in a list including:

- \texttt{s}  the column(s) to calculate variable importance for
- \texttt{SL.library}  the library of learners passed to \texttt{SuperLearner}
- \texttt{full_fit}  the fitted values of the chosen method fit to the full data (a list, for train and test data)
- \texttt{red_fit}  the fitted values of the chosen method fit to the reduced data (a list, for train and test data)
- \texttt{est}  the estimated variable importance
- \texttt{naive}  the naive estimator of variable importance
- \texttt{EIF}  the estimated efficient influence function
- \texttt{EIF_full}  the estimated efficient influence function for the full regression
- \texttt{EIF_reduced}  the estimated efficient influence function for the reduced regression
- \texttt{se}  the standard error for the estimated variable importance
- \texttt{ci}  the $(1 - \alpha) \times 100\%$ confidence interval for the variable importance estimate
- \texttt{test}  a decision to either reject (TRUE) or not reject (FALSE) the null hypothesis, based on a conservative test
- \texttt{p_value}  a p-value based on the same test as \texttt{test}
- \texttt{full_mod}  the object returned by the estimation procedure for the full data regression (if applicable)
red_mod the object returned by the estimation procedure for the reduced data regression (if applicable)

alpha the level, for confidence interval calculation

sample_splitting_folds the folds used for hypothesis testing

cross_fitting_folds the folds used for cross-fitting

y the outcome

ipc_weights the weights

mat a tibble with the estimate, SE, CI, hypothesis testing decision, and p-value

Value

An object of classes vim and vim_accuracy. See Details for more information.

See Also

SuperLearner for specific usage of the SuperLearner function and package.

Examples

# generate the data
# generate X
p <- 2
n <- 100
x <- data.frame(replicate(p, stats::runif(n, -1, 1)))

# apply the function to the x's
f <- function(x) 0.5 + 0.3*x[1] + 0.2*x[2]
smooth <- apply(x, 1, function(z) f(z))

# generate Y ~ Normal (smooth, 1)
y <- matrix(rbinom(n, size = 1, prob = smooth))

# set up a library for SuperLearner; note simple library for speed
library("SuperLearner")
learners <- c("SL.glm", "SL.mean")

# estimate (with a small number of folds, for illustration only)
est <- vimp_accuracy(y, x, indx = 2,
alpha = 0.05, run_regression = TRUE,
SL.library = learners, V = 2, cvControl = list(V = 2))
Description

Compute estimates of and confidence intervals for nonparametric ANOVA-based intrinsic variable importance. This is a wrapper function for cv_vim, with type = "anova". This type has limited functionality compared to other types; in particular, null hypothesis tests are not possible using type = "anova". If you want to do null hypothesis testing on an equivalent population parameter, use vimp_rsquared instead.

Usage

vimp_anova(
  Y = NULL,
  X = NULL,
  cross_fitted_f1 = NULL,
  cross_fitted_f2 = NULL,
  indx = 1,
  V = 10,
  run_regression = TRUE,
  SL.library = c("SL.glmnet", "SL.xgboost", "SL.mean"),
  alpha = 0.05,
  delta = 0,
  na.rm = FALSE,
  cross_fitting_folds = NULL,
  stratified = FALSE,
  C = rep(1, length(Y)),
  Z = NULL,
  ipc_weights = rep(1, length(Y)),
  scale = "logit",
  ipc_est_type = "aipw",
  scale_est = TRUE,
  cross_fitted_se = TRUE,
  ...
)

Arguments

Y the outcome.

X the covariates. If type = "average_value", then the exposure variable should be part of X, with its name provided in exposure_name.

cross_fitted_f1 the predicted values on validation data from a flexible estimation technique regressing Y on X in the training data. Provided as either (a) a vector, where each element is the predicted value when that observation is part of the validation
fold; or (b) a list of length \( V \), where each element in the list is a set of predictions on the corresponding validation data fold. If sample-splitting is requested, then these must be estimated specially; see Details. However, the resulting vector should be the same length as \( Y \); if using a list, then the summed length of each element across the list should be the same length as \( Y \) (i.e., each observation is included in the predictions).

\texttt{cross\_fitted\_f2}

the predicted values on validation data from a flexible estimation technique regressing either (a) the fitted values in \texttt{cross\_fitted\_f1}, or (b) \( Y \), on \( X \) withholding the columns in \( \text{indx} \). Provided as either (a) a vector, where each element is the predicted value when that observation is part of the validation fold; or (b) a list of length \( V \), where each element in the list is a set of predictions on the corresponding validation data fold. If sample-splitting is requested, then these must be estimated specially; see Details. However, the resulting vector should be the same length as \( Y \); if using a list, then the summed length of each element across the list should be the same length as \( Y \) (i.e., each observation is included in the predictions).

\texttt{indx}

the indices of the covariate(s) to calculate variable importance for; defaults to 1.

\texttt{V}

the number of folds for cross-fitting, defaults to 5. If \texttt{sample\_splitting = TRUE}, then a special type of \( V \)-fold cross-fitting is done. See Details for a more detailed explanation.

\texttt{run\_regression}

if outcome \( Y \) and covariates \( X \) are passed to \texttt{vimp\_accuracy}, and \texttt{run\_regression} is \texttt{TRUE}, then Super Learner will be used; otherwise, variable importance will be computed using the inputted fitted values.

\texttt{SL.library}

a character vector of learners to pass to \texttt{SuperLearner}, if \texttt{f1} and \texttt{f2} are \( Y \) and \( X \), respectively. Defaults to \texttt{SL.glmnet}, \texttt{SL.xgboost}, and \texttt{SL.mean}.

\texttt{alpha}

the level to compute the confidence interval at. Defaults to 0.05, corresponding to a 95\% confidence interval.

\texttt{delta}

the value of the \( \delta \)-null (i.e., testing if importance < \( \delta \)); defaults to 0.

\texttt{na.rm}

should we remove NAs in the outcome and fitted values in computation? (defaults to \texttt{FALSE})

\texttt{cross\_fitting\_folds}

the folds for cross-fitting. Only used if \texttt{run\_regression = FALSE}.

\texttt{stratified}

if \texttt{run\_regression = TRUE}, then should the generated folds be stratified based on the outcome (helps to ensure class balance across cross-validation folds)

\texttt{C}

the indicator of coarsening (1 denotes observed, 0 denotes unobserved).

\texttt{Z}

either (i) \texttt{NULL} (the default, in which case the argument \( C \) above must be all ones), or (ii) a character vector specifying the variable(s) among \( Y \) and \( X \) that are thought to play a role in the coarsening mechanism. To specify the outcome, use "\( Y \)"; to specify covariates, use a character number corresponding to the desired position in \( X \) (e.g., "1").

\texttt{ipc\_weights}

weights for the computed influence curve (i.e., inverse probability weights for coarsened-at-random settings). Assumed to be already inverted (i.e., \texttt{ipc\_weights = 1 / [estimated probability weights]}).
scale should CIs be computed on original ("identity") or another scale? (options are "log" and "logit")

ipc_est_type the type of procedure used for coarsened-at-random settings; options are "ipw" (for inverse probability weighting) or "aipw" (for augmented inverse probability weighting). Only used if C is not all equal to 1.

scale_est should the point estimate be scaled to be greater than or equal to 0? Defaults to TRUE.

cross_fitted_se should we use cross-fitting to estimate the standard errors (TRUE, the default) or not (FALSE)?

... other arguments to the estimation tool, see "See also".

Details

We define the population ANOVA parameter for the group of features (or single feature) \( s \) by

\[
\psi_{0,s} := E_0\{f_0(X) - f_{0,s}(X)\}^2/\text{var}_0(Y),
\]

where \( f_0 \) is the population conditional mean using all features, \( f_{0,s} \) is the population conditional mean using the features with index not in \( s \), and \( E_0 \) and \( \text{var}_0 \) denote expectation and variance under the true data-generating distribution, respectively.

Cross-fitted ANOVA estimates are computed by first splitting the data into \( K \) folds; then using each fold in turn as a hold-out set, constructing estimators \( f_{n,k} \) and \( f_{n,k,s} \) of \( f_0 \) and \( f_{0,s} \), respectively on the training data and estimator \( E_{n,k} \) of \( E_0 \) using the test data; and finally, computing

\[
\psi_{n,s} := K^{-1} \sum_{k=1}^{K} E_{n,k}\{f_{n,k}(X) - f_{n,k,s}(X)\}^2/\text{var}_n(Y),
\]

where \( \text{var}_n \) is the empirical variance. See the paper by Williamson, Gilbert, Simon, and Carone for more details on the mathematics behind this function.

Value

An object of classes \texttt{vim} and \texttt{vim_anova}. See Details for more information.

See Also

\texttt{SuperLearner} for specific usage of the \texttt{SuperLearner} function and package.

Examples

# generate the data
# generate X
p <- 2
n <- 100
x <- data.frame(replicate(p, stats::runif(n, -5, 5)))

# apply the function to the x's
smooth <- (x[,1]/5)^2*(x[,1]+7)/5 + (x[,2]/3)^2
# generate Y ~ Normal (smooth, 1)
y <- smooth + stats::rnorm(n, 0, 1)

# set up a library for SuperLearner; note simple library for speed
library("SuperLearner")
learners <- c("SL.glm", "SL.mean")

# estimate (with a small number of folds, for illustration only)
est <- vimp_anova(y, x, indx = 2,
                 alpha = 0.05, run_regression = TRUE,
                 SL.library = learners, V = 2, cvControl = list(V = 2))

---

vimp_auc  Nonparametric Intrinsic Variable Importance Estimates: AUC

Description

Compute estimates of and confidence intervals for nonparametric difference in $AUC$-based intrinsic variable importance. This is a wrapper function for cv_vim, with type = "auc".

Usage

vimp_auc(
  Y = NULL,
  X = NULL,
  cross_fitted_f1 = NULL,
  cross_fitted_f2 = NULL,
  f1 = NULL,
  f2 = NULL,
  indx = 1,
  V = 10,
  run_regression = TRUE,
  SL.library = c("SL.glmnet", "SL.xgboost", "SL.mean"),
  alpha = 0.05,
  delta = 0,
  na.rm = FALSE,
  final_point_estimate = "split",
  cross_fitting_folds = NULL,
  sample_splitting_folds = NULL,
  stratified = TRUE,
  C = rep(1, length(Y)),
  Z = NULL,
  ipc_weights = rep(1, length(Y)),
  scale = "logit",
  ipc_est_type = "aipw",
  scale_est = TRUE,
cross_fitted_se = TRUE,
...
}

Arguments

Y
the outcome.

X
the covariates. If type = "average_value", then the exposure variable should be part of X, with its name provided in exposure_name.

cross_fitted_f1
the predicted values on validation data from a flexible estimation technique regressing Y on X in the training data. Provided as either (a) a vector, where each element is the predicted value when that observation is part of the validation fold; or (b) a list of length V, where each element in the list is a set of predictions on the corresponding validation data fold. If sample-splitting is requested, then these must be estimated specially; see Details. However, the resulting vector should be the same length as Y; if using a list, then the summed length of each element across the list should be the same length as Y (i.e., each observation is included in the predictions).

cross_fitted_f2
the predicted values on validation data from a flexible estimation technique regressing either (a) the fitted values in cross_fitted_f1, or (b) Y, on X withholding the columns in indx. Provided as either (a) a vector, where each element is the predicted value when that observation is part of the validation fold; or (b) a list of length V, where each element in the list is a set of predictions on the corresponding validation data fold. If sample-splitting is requested, then these must be estimated specially; see Details. However, the resulting vector should be the same length as Y; if using a list, then the summed length of each element across the list should be the same length as Y (i.e., each observation is included in the predictions).

f1
the fitted values from a flexible estimation technique regressing Y on X. If sample-splitting is requested, then these must be estimated specially; see Details. If cross_fitted_se = TRUE, then this argument is not used.

f2
the fitted values from a flexible estimation technique regressing either (a) f1 or (b) Y on X withholding the columns in indx. If sample-splitting is requested, then these must be estimated specially; see Details. If cross_fitted_se = TRUE, then this argument is not used.

indx
the indices of the covariate(s) to calculate variable importance for; defaults to 1.

V
the number of folds for cross-fitting. defaults to 5. If sample_splitting = TRUE, then a special type of V-fold cross-fitting is done. See Details for a more detailed explanation.

run_regression
if outcome Y and covariates X are passed to vimp_accuracy, and run_regression is TRUE, then Super Learner will be used; otherwise, variable importance will be computed using the inputted fitted values.

SL.library
a character vector of learners to pass to SuperLearner, if f1 and f2 are Y and X, respectively. Defaults to SL.glmnet, SL.xgboost, and SL.mean.
alpha

the level to compute the confidence interval at. Defaults to 0.05, corresponding to a 95% confidence interval.

delta

the value of the δ-null (i.e., testing if importance < δ); defaults to 0.

na.rm

should we remove NAs in the outcome and fitted values in computation? (defaults to FALSE)

final_point_estimate

if sample splitting is used, should the final point estimates be based on only the sample-split folds used for inference ("split", the default), or should they instead be based on the full dataset ("full") or the average across the point estimates from each sample split ("average")? All three options result in valid point estimates – sample-splitting is only required for valid inference.

cross_fitting_folds

the folds for cross-fitting. Only used if run_regression = FALSE.

cross_fitting_folds

the folds used for cross-splitting; these identify the observations that should be used to evaluate predictiveness based on the full and reduced sets of covariates, respectively. Only used if run_regression = FALSE.

stratified

if run_regression = TRUE, then should the generated folds be stratified based on the outcome (helps to ensure class balance across cross-validation folds)

C

the indicator of coarsening (1 denotes observed, 0 denotes unobserved).

Z

either (i) NULL (the default, in which case the argument C above must be all ones), or (ii) a character vector specifying the variable(s) among Y and X that are thought to play a role in the coarsening mechanism. To specify the outcome, use "Y"; to specify covariates, use a character number corresponding to the desired position in X (e.g., "1").

ipc_weights

weights for the computed influence curve (i.e., inverse probability weights for coarsened-at-random settings). Assumed to be already inverted (i.e., ipc_weights = 1 / [estimated probability weights]).

scale

should CIs be computed on original ("identity") or another scale? (options are "log" and "logit")

ipc_est_type

the type of procedure used for coarsened-at-random settings; options are "ipw" (for inverse probability weighting) or "aipw" (for augmented inverse probability weighting). Only used if C is not all equal to 1.

scale_est

should the point estimate be scaled to be greater than or equal to 0? Defaults to TRUE.

cross_fitted_se

should we use cross-fitting to estimate the standard errors (TRUE, the default) or not (FALSE)?

... other arguments to the estimation tool, see "See also".

Details

We define the population variable importance measure (VIM) for the group of features (or single feature) \( s \) with respect to the predictiveness measure \( V \) by

\[
\psi_{0,s} := V(f_0, P_0) - V(f_{0,s}, P_0),
\]
where $f_0$ is the population predictiveness maximizing function, $f_{0,s}$ is the population predictiveness maximizing function that is only allowed to access the features with index not in $s$, and $P_0$ is the true data-generating distribution.

Cross-fitted VIM estimates are computed differently if sample-splitting is requested versus if it is not. We recommend using sample-splitting in most cases, since only in this case will inferences be valid if the variable(s) of interest have truly zero population importance. The purpose of cross-fitting is to estimate $f_0$ and $f_{0,s}$ on independent data from estimating $P_0$; this can result in improved performance, especially when using flexible learning algorithms. The purpose of sample-splitting is to estimate $f_0$ and $f_{0,s}$ on independent data; this allows valid inference under the null hypothesis of zero importance.

Without sample-splitting, cross-fitted VIM estimates are obtained by first splitting the data into $K$ folds; then using each fold in turn as a hold-out set, constructing estimators $f_{n,k}$ and $f_{n,k,s}$ of $f_0$ and $f_{0,s}$, respectively on the training data and estimator $P_{n,k}$ of $P_0$ using the test data; and finally, computing

$$\psi_{n,s} := K^{(-1)} \sum_{k=1}^{K} \{ V(f_{n,k}, P_{n,k}) - V(f_{n,k,s}, P_{n,k}) \}.$$  

With sample-splitting, cross-fitted VIM estimates are obtained by first splitting the data into $2K$ folds. These folds are further divided into 2 groups of folds. Then, for each fold $k$ in the first group, estimator $f_{n,k}$ of $f_0$ is constructed using all data besides the kth fold in the group (i.e., $(2K - 1)/(2K)$ of the data) and estimator $P_{n,k}$ of $P_0$ is constructed using the hold-out data (i.e., $1/2K$ of the data); then, computing

$$v_{n,k} = V(f_{n,k}, P_{n,k}).$$

Similarly, for each fold $k$ in the second group, estimator $f_{n,k,s}$ of $f_{0,s}$ is constructed using all data besides the kth fold in the group (i.e., $(2K - 1)/(2K)$ of the data) and estimator $P_{n,k}$ of $P_0$ is constructed using the hold-out data (i.e., $1/2K$ of the data); then, computing

$$v_{n,k,s} = V(f_{n,k,s}, P_{n,k}).$$

Finally,

$$\psi_{n,s} := K^{(-1)} \sum_{k=1}^{K} \{ v_{n,k} - v_{n,k,s} \}.$$  

See the paper by Williamson, Gilbert, Simon, and Carone for more details on the mathematics behind the cv_vim function, and the validity of the confidence intervals.

In the interest of transparency, we return most of the calculations within the vim object. This results in a list including:

- **s** the column(s) to calculate variable importance for
- **SL.library** the library of learners passed to SuperLearner
- **full_fit** the fitted values of the chosen method fit to the full data (a list, for train and test data)
- **red_fit** the fitted values of the chosen method fit to the reduced data (a list, for train and test data)
- **est** the estimated variable importance
- **naive** the naive estimator of variable importance
**eif**  the estimated efficient influence function

**eif_full**  the estimated efficient influence function for the full regression

**eif_reduced**  the estimated efficient influence function for the reduced regression

**se**  the standard error for the estimated variable importance

**ci**  the \((1 - \alpha) \times 100\%\) confidence interval for the variable importance estimate

**test**  a decision to either reject (TRUE) or not reject (FALSE) the null hypothesis, based on a conservative test

**p_value**  a p-value based on the same test as **test**

**full_mod**  the object returned by the estimation procedure for the full data regression (if applicable)

**red_mod**  the object returned by the estimation procedure for the reduced data regression (if applicable)

**alpha**  the level, for confidence interval calculation

**sample_splitting_folds**  the folds used for hypothesis testing

**cross_fitting_folds**  the folds used for cross-fitting

**y**  the outcome

**ipc_weights**  the weights

**mat**  a tibble with the estimate, SE, CI, hypothesis testing decision, and p-value

**Value**

An object of classes `vim` and `vim_auc`. See Details for more information.

**See Also**

`SuperLearner` for specific usage of the `SuperLearner` function and package, and `performance` for specific usage of the ROCR package.

**Examples**

```r
# generate the data
# generate X
p <- 2
n <- 100
x <- data.frame(replicate(p, stats::runif(n, -1, 1)))

# apply the function to the x's
f <- function(x) 0.5 + 0.3*x[1] + 0.2*x[2]
smooth <- apply(x, 1, function(z) f(z))

# generate Y ~ Normal (smooth, 1)
y <- matrix(rbinom(n, size = 1, prob = smooth))

# set up a library for SuperLearner; note simple library for speed
library("SuperLearner")
learners <- c("SL.glm", "SL.mean")
```
# estimate (with a small number of folds, for illustration only)
est <- vimp_auc(y, x, indx = 2,
    alpha = 0.05, run_regression = TRUE,
    SL.library = learners, V = 2, cvControl = list(V = 2))

---

vimp_ci

Confidence intervals for variable importance

Description

Compute confidence intervals for the true variable importance parameter.

Usage

vimp_ci(est, se, scale = "identity", level = 0.95, truncate = TRUE)

Arguments

- `est`: estimate of variable importance, e.g., from a call to `vimp_point_est`.
- `se`: estimate of the standard error of `est`, e.g., from a call to `vimp_se`.
- `scale`: scale to compute interval estimate on (defaults to "identity": compute Wald-type CI).
- `level`: confidence interval type (defaults to 0.95).
- `truncate`: truncate CIs to have lower limit at (or above) zero?

Details

See the paper by Williamson, Gilbert, Simon, and Carone for more details on the mathematics behind this function and the definition of the parameter of interest.

Value

The Wald-based confidence interval for the true importance of the given group of left-out covariates.
vimp_deviance

Nonparametric Intrinsic Variable Importance Estimates: Deviance

Description

Compute estimates of and confidence intervals for nonparametric deviance-based intrinsic variable importance. This is a wrapper function for cv_vim, with type = "deviance".

Usage

vimp_deviance(
  Y = NULL,
  X = NULL,
  cross_fitted_f1 = NULL,
  cross_fitted_f2 = NULL,
  f1 = NULL,
  f2 = NULL,
  indx = 1,
  V = 10,
  run_regression = TRUE,
  SL.library = c("SL.glmnet", "SL.xgboost", "SL.mean"),
  alpha = 0.05,
  delta = 0,
  na.rm = FALSE,
  final_point_estimate = "split",
  cross_fitting_folds = NULL,
  sample_splitting_folds = NULL,
  stratified = TRUE,
  C = rep(1, length(Y)),
  Z = NULL,
  ipc_weights = rep(1, length(Y)),
  scale = "logit",
  ipc_est_type = "aipw",
  scale_est = TRUE,
  cross_fitted_se = TRUE,
  ...
)

Arguments

Y the outcome.

X the covariates. If type = "average_value", then the exposure variable should be part of X, with its name provided in exposure_name.

cross_fitted_f1 the predicted values on validation data from a flexible estimation technique regressing Y on X in the training data. Provided as either (a) a vector, where each
element is the predicted value when that observation is part of the validation fold; or (b) a list of length V, where each element in the list is a set of predictions on the corresponding validation data fold. If sample-splitting is requested, then these must be estimated specially; see Details. However, the resulting vector should be the same length as Y; if using a list, then the summed length of each element across the list should be the same length as Y (i.e., each observation is included in the predictions).

cross_fitted_f2
the predicted values on validation data from a flexible estimation technique regressing either (a) the fitted values in cross_fitted_f1, or (b) Y, on X witholding the columns in indx. Provided as either (a) a vector, where each element is the predicted value when that observation is part of the validation fold; or (b) a list of length V, where each element in the list is a set of predictions on the corresponding validation data fold. If sample-splitting is requested, then these must be estimated specially; see Details. However, the resulting vector should be the same length as Y; if using a list, then the summed length of each element across the list should be the same length as Y (i.e., each observation is included in the predictions).

f1
the fitted values from a flexible estimation technique regressing Y on X. If sample-splitting is requested, then these must be estimated specially; see Details. If cross_fitted_se = TRUE, then this argument is not used.

f2
the fitted values from a flexible estimation technique regressing either (a) f1 or (b) Y on X witholding the columns in indx. If sample-splitting is requested, then these must be estimated specially; see Details. If cross_fitted_se = TRUE, then this argument is not used.

indx
the indices of the covariate(s) to calculate variable importance for; defaults to 1.

V
the number of folds for cross-fitting, defaults to 5. If sample_splitting = TRUE, then a special type of V-fold cross-fitting is done. See Details for a more detailed explanation.

run_regression
if outcome Y and covariates X are passed to vimp_accuracy, and run_regression is TRUE, then Super Learner will be used; otherwise, variable importance will be computed using the inputted fitted values.

SL.library
a character vector of learners to pass to SuperLearner, if f1 and f2 are Y and X, respectively. Defaults to SL.glmnet, SL.xgboost, and SL.mean.

alpha
the level to compute the confidence interval at. Defaults to 0.05, corresponding to a 95% confidence interval.

delta
the value of the δ-null (i.e., testing if importance < δ); defaults to 0.

na.rm
should we remove NAs in the outcome and fitted values in computation? (defaults to FALSE)

final_point_estimate
if sample splitting is used, should the final point estimates be based on only the sample-split folds used for inference ("split", the default), or should they instead be based on the full dataset ("full") or the average across the point estimates from each sample split ("average")? All three options result in valid point estimates – sample-splitting is only required for valid inference.
cross_fitting_folds
the folds for cross-fitting. Only used if run_regression = FALSE.

sample_splitting_folds
the folds used for sample-splitting; these identify the observations that should be
used to evaluate predictiveness based on the full and reduced sets of covariates,
respectively. Only used if run_regression = FALSE.

stratified
if run_regression = TRUE, then should the generated folds be stratified based on
the outcome (helps to ensure class balance across cross-validation folds)

C
the indicator of coarsening (1 denotes observed, 0 denotes unobserved).

Z
either (i) NULL (the default, in which case the argument C above must be all
ones), or (ii) a character vector specifying the variable(s) among Y and X that are
thought to play a role in the coarsening mechanism. To specify the outcome, use
"Y"; to specify covariates, use a character number corresponding to the desired
position in X (e.g., "1").

ipc_weights
weights for the computed influence curve (i.e., inverse probability weights for
coarsened-at-random settings). Assumed to be already inverted (i.e., ipc_weights
= 1 / [estimated probability weights]).

scale
should CIs be computed on original ("identity") or another scale? (options are
"log" and "logit")

ipc_est_type
the type of procedure used for coarsened-at-random settings; options are "ipw"
(for inverse probability weighting) or "aipw" (for augmented inverse probability
weighting). Only used if C is not all equal to 1.

scale_est
should the point estimate be scaled to be greater than or equal to 0? Defaults to
TRUE.

cross_fitted_se
should we use cross-fitting to estimate the standard errors (TRUE, the default) or
not (FALSE)?

... other arguments to the estimation tool, see "See also".

Details
We define the population variable importance measure (VIM) for the group of features (or single
feature) s with respect to the predictiveness measure V by

\[
\psi_{0,s} := V(f_0, P_0) - V(f_{0,s}, P_0),
\]

where \( f_0 \) is the population predictiveness maximizing function, \( f_{0,s} \) is the population predictiveness
maximizing function that is only allowed to access the features with index not in s, and \( P_0 \) is the
true data-generating distribution.

Cross-fitted VIM estimates are computed differently if sample-splitting is requested versus if it is
not. We recommend using sample-splitting in most cases, since only in this case will inferences
be valid if the variable(s) of interest have truly zero population importance. The purpose of cross-
fitting is to estimate \( f_0 \) and \( f_{0,s} \) on independent data from estimating \( P_0 \); this can result in improved
performance, especially when using flexible learning algorithms. The purpose of sample-splitting
is to estimate \( f_0 \) and \( f_{0,s} \) on independent data; this allows valid inference under the null hypothesis
of zero importance.
Without sample-splitting, cross-fitted VIM estimates are obtained by first splitting the data into $K$ folds; then using each fold in turn as a hold-out set, constructing estimators $f_{n,k}$ and $f_{n,k,s}$ of $f_0$ and $f_{0,s}$, respectively on the training data and estimator $P_{n,k}$ of $P_0$ using the test data; and finally, computing

$$\psi_{n,s} := K^{(-1)} \sum_{k=1}^{K} \{ V(f_{n,k}, P_{n,k}) - V(f_{n,k,s}, P_{n,k}) \}.$$ 

With sample-splitting, cross-fitted VIM estimates are obtained by first splitting the data into $2K$ folds. These folds are further divided into 2 groups of folds. Then, for each fold $k$ in the first group, estimator $f_{n,k}$ of $f_0$ is constructed using all data besides the $k$th fold in the group (i.e., $(2K - 1)/(2K)$ of the data) and estimator $P_{n,k}$ of $P_0$ is constructed using the held-out data (i.e., $1/2K$ of the data); then, computing

$$v_{n,k} = V(f_{n,k}, P_{n,k}).$$

Similarly, for each fold $k$ in the second group, estimator $f_{n,k,s}$ of $f_{0,s}$ is constructed using all data besides the $k$th fold in the group (i.e., $(2K - 1)/(2K)$ of the data) and estimator $P_{n,k}$ of $P_0$ is constructed using the held-out data (i.e., $1/2K$ of the data); then, computing

$$v_{n,k,s} = V(f_{n,k,s}, P_{n,k}).$$

Finally,

$$\psi_{n,s} := K^{(-1)} \sum_{k=1}^{K} \{ v_{n,k} - v_{n,k,s} \}.$$ 

See the paper by Williamson, Gilbert, Simon, and Carone for more details on the mathematics behind the cv_vim function, and the validity of the confidence intervals.

In the interest of transparency, we return most of the calculations within the vim object. This results in a list including:

- s the column(s) to calculate variable importance for
- SL.library the library of learners passed to SuperLearner
- full_fit the fitted values of the chosen method fit to the full data (a list, for train and test data)
- red_fit the fitted values of the chosen method fit to the reduced data (a list, for train and test data)
- est the estimated variable importance
- naive the naive estimator of variable importance
- eif the estimated efficient influence function
- eif_full the estimated efficient influence function for the full regression
- eif_reduced the estimated efficient influence function for the reduced regression
- se the standard error for the estimated variable importance
- ci the $(1 - \alpha) \times 100\%$ confidence interval for the variable importance estimate
- test a decision to either reject (TRUE) or not reject (FALSE) the null hypothesis, based on a conservative test
- p_value a p-value based on the same test as test
- full_mod the object returned by the estimation procedure for the full data regression (if applicable)
red_mod  the object returned by the estimation procedure for the reduced data regression (if applicable)
alpha  the level, for confidence interval calculation
sample_splitting_folds  the folds used for hypothesis testing
cross_fitting_folds  the folds used for cross-fitting
y  the outcome
ipc_weights  the weights
mat  a tibble with the estimate, SE, CI, hypothesis testing decision, and p-value

Value
An object of classes `vim` and `vim_deviance`. See Details for more information.

See Also
`SuperLearner` for specific usage of the `SuperLearner` function and package.

Examples
# generate the data
# generate X
p <- 2
n <- 100
x <- data.frame(replicate(p, stats::runif(n, -1, 1)))

# apply the function to the x's
f <- function(x) 0.5 + 0.3*x[1] + 0.2*x[2]
smooth <- apply(x, 1, function(z) f(z))

# generate Y ~ Normal (smooth, 1)
y <- matrix(stats::rbinom(n, size = 1, prob = smooth))

# set up a library for SuperLearner; note simple library for speed
library("SuperLearner")
learners <- c("SL.glm", "SL.mean")

# estimate (with a small number of folds, for illustration only)
est <- vimp_deviance(y, x, indx = 2,
  alpha = 0.05, run_regression = TRUE,
  SL.library = learners, V = 2, cvControl = list(V = 2))
vimp_hypothesis_test

Perform a hypothesis test against the null hypothesis of $\delta$ importance

**Description**

Perform a hypothesis test against the null hypothesis of zero importance by: (i) for a user-specified level $\alpha$, compute a $(1 - \alpha) \times 100\%$ confidence interval around the predictiveness for both the full and reduced regression functions (these must be estimated on independent splits of the data); (ii) if the intervals do not overlap, reject the null hypothesis.

**Usage**

```r
vimp_hypothesis_test(
  predictiveness_full,
  predictiveness_reduced,
  se,
  delta = 0,
  alpha = 0.05
)
```

**Arguments**

- `predictiveness_full`: the estimated predictiveness of the regression including the covariate(s) of interest.
- `predictiveness_reduced`: the estimated predictiveness of the regression excluding the covariate(s) of interest.
- `se`: the estimated standard error of the variable importance estimator.
- `delta`: the value of the $\delta$-null (i.e., testing if importance < $\delta$); defaults to 0.
- `alpha`: the desired type I error rate (default to 0.05).

**Details**

See the paper by Williamson, Gilbert, Simon, and Carone for more details on the mathematics behind this function and the definition of the parameter of interest.

**Value**

A list, with: the hypothesis testing decision (TRUE if the null hypothesis is rejected, FALSE otherwise); the p-value from the hypothesis test; and the test statistic from the hypothesis test.
vimp_regression

Nonparametric Intrinsic Variable Importance Estimates: ANOVA

Description

Compute estimates of and confidence intervals for nonparametric ANOVA-based intrinsic variable importance. This is a wrapper function for cv_vim, with type = "anova". This function is deprecated in vimp version 2.0.0.

Usage

vimp_regression(
  Y = NULL,
  X = NULL,
  cross_fitted_f1 = NULL,
  cross_fitted_f2 = NULL,
  indx = 1,
  V = 10,
  run_regression = TRUE,
  SL.library = c("SL.glmnet", "SL.xgboost", "SL.mean"),
  alpha = 0.05,
  delta = 0,
  na.rm = FALSE,
  cross_fitting_folds = NULL,
  stratified = FALSE,
  C = rep(1, length(Y)),
  Z = NULL,
  ipc_weights = rep(1, length(Y)),
  scale = "identity",
  ipc_est_type = "aipw",
  scale_est = TRUE,
  cross_fitted_se = TRUE,
  ...
)

Arguments

Y
  the outcome.

X
  the covariates. If type = "average_value", then the exposure variable should be part of X, with its name provided in exposure_name.

cross_fitted_f1
  the predicted values on validation data from a flexible estimation technique regressing Y on X in the training data. Provided as either (a) a vector, where each element is the predicted value when that observation is part of the validation fold; or (b) a list of length V, where each element in the list is a set of predictions on the corresponding validation data fold. If sample-splitting is requested,
then these must be estimated specially; see Details. However, the resulting vector should be the same length as \( Y \); if using a list, then the summed length of each element across the list should be the same length as \( Y \) (i.e., each observation is included in the predictions).

cross_fitted_f2
the predicted values on validation data from a flexible estimation technique regressing either (a) the fitted values in cross_fitted_f1, or (b) \( Y \), on \( X \) withholding the columns in \( \text{indx} \). Provided as either (a) a vector, where each element is the predicted value when that observation is part of the validation fold; or (b) a list of length \( V \), where each element in the list is a set of predictions on the corresponding validation data fold. If sample-splitting is requested, then these must be estimated specially; see Details. However, the resulting vector should be the same length as \( Y \); if using a list, then the summed length of each element across the list should be the same length as \( Y \) (i.e., each observation is included in the predictions).

\( \text{indx} \)
the indices of the covariate(s) to calculate variable importance for; defaults to 1.

\( V \)
the number of folds for cross-fitting, defaults to 5. If \( \text{sample_splitting} = \text{TRUE} \), then a special type of \( V \)-fold cross-fitting is done. See Details for a more detailed explanation.

run_regression
if outcome \( Y \) and covariates \( X \) are passed to \( \text{vimp}\_\text{accuracy} \), and \( \text{run}\_\text{regression} \) is \text{TRUE}, then Super Learner will be used; otherwise, variable importance will be computed using the inputted fitted values.

SL.library
a character vector of learners to pass to \text{SuperLearner}, if \( f1 \) and \( f2 \) are \( Y \) and \( X \), respectively. Defaults to \text{SL.glmnet}, \text{SL.xgboost}, and \text{SL.mean}.

alpha
the level to compute the confidence interval at. Defaults to 0.05, corresponding to a 95% confidence interval.

delta
the value of the \( \delta \)-null (i.e., testing if importance < \( \delta \)); defaults to 0.

na.rm
should we remove NAs in the outcome and fitted values in computation? (defaults to \text{FALSE})

cross_fitting_folds
the folds for cross-fitting. Only used if \( \text{run}\_\text{regression} = \text{FALSE} \).

stratified
if \( \text{run}\_\text{regression} = \text{TRUE} \), then should the generated folds be stratified based on the outcome (helps to ensure class balance across cross-validation folds)

\( C \)
the indicator of coarsening (1 denotes observed, 0 denotes unobserved).

\( Z \)
either (i) \text{NULL} (the default, in which case the argument \( C \) above must be all ones), or (ii) a character vector specifying the variable(s) among \( Y \) and \( X \) that are thought to play a role in the coarsening mechanism. To specify the outcome, use "\( Y \)"; to specify covariates, use a character number corresponding to the desired position in \( X \) (e.g., "1").

ipc_weights
weights for the computed influence curve (i.e., inverse probability weights for coarsened-at-random settings). Assumed to be already inverted (i.e., ipc_weights = 1 / [estimated probability weights]).

scale
should CIs be computed on original ("identity") or another scale? (options are "log" and "logit")
ipc_est_type: the type of procedure used for coarsened-at-random settings; options are "ipw" (for inverse probability weighting) or "aipw" (for augmented inverse probability weighting). Only used if \( C \) is not all equal to 1.

scale_est: should the point estimate be scaled to be greater than or equal to 0? Defaults to TRUE.

cross_fitted_se: should we use cross-fitting to estimate the standard errors (TRUE, the default) or not (FALSE)?

Details
We define the population ANOVA parameter for the group of features (or single feature) \( s \) by
\[
\psi_{0,s} := E_0 \left\{ f_0(X) - f_{0,s}(X) \right\}^2 / \text{var}_0(Y),
\]
where \( f_0 \) is the population conditional mean using all features, \( f_{0,s} \) is the population conditional mean using the features with index not in \( s \), and \( E_0 \) and \( \text{var}_0 \) denote expectation and variance under the true data-generating distribution, respectively.

Cross-fitted ANOVA estimates are computed by first splitting the data into \( K \) folds; then using each fold in turn as a hold-out set, constructing estimators \( f_{n,k} \) and \( f_{n,k,s} \) of \( f_0 \) and \( f_{0,s} \), respectively on the training data and estimator \( E_{n,k} \) of \( E_0 \) using the test data; and finally, computing
\[
\psi_{n,s} := K^{-1} \sum_{k=1}^{K} E_{n,k} \left\{ f_{n,k}(X) - f_{n,k,s}(X) \right\}^2 / \text{var}_n(Y),
\]
where \( \text{var}_n \) is the empirical variance. See the paper by Williamson, Gilbert, Simon, and Carone for more details on the mathematics behind this function.

Value
An object of classes vim and vim_regression. See Details for more information.

See Also
SuperLearner for specific usage of the SuperLearner function and package.

Examples
# generate the data
# generate X
p <- 2
n <- 100
x <- data.frame(replicate(p, stats::runif(n, -5, 5)))

# apply the function to the x's
smooth <- (x[,1]/5)^2*(x[,1]+7)/5 + (x[,2]/3)^2

# generate Y ~ Normal (smooth, 1)
y <- smooth + stats::rnorm(n, 0, 1)
# set up a library for SuperLearner; note simple library for speed
library("SuperLearner")
learners <- c("SL.glm", "SL.mean")

# estimate (with a small number of folds, for illustration only)
est <- vimp_regression(y, x, indx = 2,
  alpha = 0.05, run_regression = TRUE,
  SL.library = learners, V = 2, cvControl = list(V = 2))

---

**vimp_rsquared**

*Nonparametric Intrinsic Variable Importance Estimates: $R^2$-based* 

**Description**

Compute estimates of and confidence intervals for nonparametric $R^2$-based intrinsic variable importance. This is a wrapper function for cv_vim, with type = "r_squared".

**Usage**

vimp_rsquared(
  Y = NULL,
  X = NULL,
  cross_fitted_f1 = NULL,
  cross_fitted_f2 = NULL,
  f1 = NULL,
  f2 = NULL,
  indx = 1,
  V = 10,
  run_regression = TRUE,
  SL.library = c("SL.glmnet", "SL.xgboost", "SL.mean"),
  alpha = 0.05,
  delta = 0,
  na.rm = FALSE,
  final_point_estimate = "split",
  cross_fitting_folds = NULL,
  sample_splitting_folds = NULL,
  stratified = FALSE,
  C = rep(1, length(Y)),
  Z = NULL,
  ipc_weights = rep(1, length(Y)),
  scale = "logit",
  ipc_est_type = "aipw",
  scale_est = TRUE,
  cross_fitted_se = TRUE,
  ...
)
Arguments

Y
the outcome.

X
the covariates. If type = "average_value", then the exposure variable should be part of X, with its name provided in exposure_name.

cross_fitted_f1
the predicted values on validation data from a flexible estimation technique regressing Y on X in the training data. Provided as either (a) a vector, where each element is the predicted value when that observation is part of the validation fold; or (b) a list of length V, where each element in the list is a set of predictions on the corresponding validation data fold. If sample-splitting is requested, then these must be estimated specially; see Details. However, the resulting vector should be the same length as Y; if using a list, then the summed length of each element across the list should be the same length as Y (i.e., each observation is included in the predictions).

cross_fitted_f2
the predicted values on validation data from a flexible estimation technique regressing either (a) the fitted values in cross_fitted_f1, or (b) Y, on X withholding the columns in indx. Provided as either (a) a vector, where each element is the predicted value when that observation is part of the validation fold; or (b) a list of length V, where each element in the list is a set of predictions on the corresponding validation data fold. If sample-splitting is requested, then these must be estimated specially; see Details. However, the resulting vector should be the same length as Y; if using a list, then the summed length of each element across the list should be the same length as Y (i.e., each observation is included in the predictions).

f1
the fitted values from a flexible estimation technique regressing Y on X. If sample-splitting is requested, then these must be estimated specially; see Details. If cross_fitted_se = TRUE, then this argument is not used.

f2
the fitted values from a flexible estimation technique regressing either (a) f1 or (b) Y on X withholding the columns in indx. If sample-splitting is requested, then these must be estimated specially; see Details. If cross_fitted_se = TRUE, then this argument is not used.

indx
the indices of the covariate(s) to calculate variable importance for; defaults to 1.

V
the number of folds for cross-fitting, defaults to 5. If sample_splitting = TRUE, then a special type of V-fold cross-fitting is done. See Details for a more detailed explanation.

run_regression
if outcome Y and covariates X are passed to vimp_accuracy, and run_regression is TRUE, then Super Learner will be used; otherwise, variable importance will be computed using the inputted fitted values.

SL.library
a character vector of learners to pass to SuperLearner, if f1 and f2 are Y and X, respectively. Defaults to SL.glmnet, SL.xgboost, and SL.mean.

alpha
the level to compute the confidence interval at. Defaults to 0.05, corresponding to a 95% confidence interval.

delta
the value of the δ-null (i.e., testing if importance < δ); defaults to 0.
na.rm should we remove NAs in the outcome and fitted values in computation? (defaults to FALSE)

final_point_estimate
if sample splitting is used, should the final point estimates be based on only the sample-split folds used for inference ("split", the default), or should they instead be based on the full dataset ("full") or the average across the point estimates from each sample split ("average")? All three options result in valid point estimates – sample-splitting is only required for valid inference.

cross_fitting_folds
the folds for cross-fitting. Only used if run_regression = FALSE.

sample_splitting_folds
the folds used for sample-splitting; these identify the observations that should be used to evaluate predictiveness based on the full and reduced sets of covariates, respectively. Only used if run_regression = FALSE.

stratified if run_regression = TRUE, then should the generated folds be stratified based on the outcome (helps to ensure class balance across cross-validation folds)

C the indicator of coarsening (1 denotes observed, 0 denotes unobserved).

Z either (i) NULL (the default, in which case the argument C above must be all ones), or (ii) a character vector specifying the variable(s) among Y and X that are thought to play a role in the coarsening mechanism. To specify the outcome, use "Y"; to specify covariates, use a character number corresponding to the desired position in X (e.g., "1").

ipc_weights weights for the computed influence curve (i.e., inverse probability weights for coarsened-at-random settings). Assumed to be already inverted (i.e., ipc_weights = 1 / [estimated probability weights]).

scale should CIs be computed on original ("identity") or another scale? (options are "log" and "logit")

ipc_est_type the type of procedure used for coarsened-at-random settings; options are "ipw" (for inverse probability weighting) or "aipw" (for augmented inverse probability weighting). Only used if C is not all equal to 1.

scale_est should the point estimate be scaled to be greater than or equal to 0? Defaults to TRUE.

cross_fitted_se should we use cross-fitting to estimate the standard errors (TRUE, the default) or not (FALSE)?

... other arguments to the estimation tool, see "See also".

Details
We define the population variable importance measure (VIM) for the group of features (or single feature) s with respect to the predictiveness measure V by

$$
\psi_{0,s} := V(f_0, P_0) - V(f_{0,s}, P_0),
$$

where $f_0$ is the population predictiveness maximizing function, $f_{0,s}$ is the population predictiveness maximizing function that is only allowed to access the features with index not in s, and $P_0$ is the true data-generating distribution.
Cross-fitted VIM estimates are computed differently if sample-splitting is requested versus if it is not. We recommend using sample-splitting in most cases, since only in this case will inferences be valid if the variable(s) of interest have truly zero population importance. The purpose of cross-fitting is to estimate $f_0$ and $f_{0,s}$ on independent data from estimating $P_0$; this can result in improved performance, especially when using flexible learning algorithms. The purpose of sample-splitting is to estimate $f_0$ and $f_{0,s}$ on independent data; this allows valid inference under the null hypothesis of zero importance.

Without sample-splitting, cross-fitted VIM estimates are obtained by first splitting the data into $K$ folds; then using each fold in turn as a hold-out set, constructing estimators $f_{n,k}$ and $f_{n,k,s}$ of $f_0$ and $f_{0,s}$, respectively on the training data and estimator $P_{n,k}$ of $P_0$ using the test data; and finally, computing

$$
\psi_{n,s} := K^{(-1)} \sum_{k=1}^{K} \{ V(f_{n,k}, P_{n,k}) - V(f_{n,k,s}, P_{n,k}) \}.
$$

With sample-splitting, cross-fitted VIM estimates are obtained by first splitting the data into $2K$ folds. These folds are further divided into 2 groups of folds. Then, for each fold $k$ in the first group, estimator $f_{n,k}$ of $f_0$ is constructed using all data besides the $k$th fold in the group (i.e., $(2K - 1)/(2K)$ of the data) and estimator $P_{n,k}$ of $P_0$ is constructed using the held-out data (i.e., $1/2K$ of the data); then, computing

$$
v_{n,k} = V(f_{n,k}, P_{n,k}).
$$

Similarly, for each fold $k$ in the second group, estimator $f_{n,k,s}$ of $f_{0,s}$ is constructed using all data besides the $k$th fold in the group (i.e., $(2K - 1)/(2K)$ of the data) and estimator $P_{n,k}$ of $P_0$ is constructed using the held-out data (i.e., $1/2K$ of the data); then, computing

$$
v_{n,k,s} = V(f_{n,k,s}, P_{n,k}).
$$

Finally,

$$
\psi_{n,s} := K^{(-1)} \sum_{k=1}^{K} \{ v_{n,k} - v_{n,k,s} \}.
$$

See the paper by Williamson, Gilbert, Simon, and Carone for more details on the mathematics behind the cv_vim function, and the validity of the confidence intervals.

In the interest of transparency, we return most of the calculations within the vim object. This results in a list including:

- **s** the column(s) to calculate variable importance for
- **SL.library** the library of learners passed to SuperLearner
- **full_fit** the fitted values of the chosen method fit to the full data (a list, for train and test data)
- **red_fit** the fitted values of the chosen method fit to the reduced data (a list, for train and test data)
- **est** the estimated variable importance
- **naive** the naive estimator of variable importance
- **eif** the estimated efficient influence function
- **eif_full** the estimated efficient influence function for the full regression
- **eif_reduced** the estimated efficient influence function for the reduced regression
se  the standard error for the estimated variable importance

**ci**  the \((1 − \alpha) \times 100\%\) confidence interval for the variable importance estimate

test  a decision to either reject (TRUE) or not reject (FALSE) the null hypothesis, based on a conservative test

**p_value**  a p-value based on the same test as test

**full_mod**  the object returned by the estimation procedure for the full data regression (if applicable)

**red_mod**  the object returned by the estimation procedure for the reduced data regression (if applicable)

**alpha**  the level, for confidence interval calculation

**sample_splitting_folds**  the folds used for hypothesis testing

**cross_fitting_folds**  the folds used for cross-fitting

**y**  the outcome

**ipc_weights**  the weights

**mat**  a tibble with the estimate, SE, CI, hypothesis testing decision, and p-value

Value

An object of classes `vim` and `vim_rsquared`. See Details for more information.

See Also

`SuperLearner` for specific usage of the `SuperLearner` function and package.

Examples

```r
# generate the data
# generate X
p <- 2
n <- 100
x <- data.frame(replicate(p, stats::runif(n, -5, 5)))

# apply the function to the x's
smooth <- (x[,1]/5)^2*(x[,1]+7)/5 + (x[,2]/3)^2

# generate Y ~ Normal (smooth, 1)
y <- smooth + stats::rnorm(n, 0, 1)

# set up a library for SuperLearner; note simple library for speed
library("SuperLearner")
learners <- c("SL.glm", "SL.mean")

# estimate (with a small number of folds, for illustration only)
est <- vimp_rsquared(y, x, indx = 2,
  alpha = 0.05, run_regression = TRUE,
  SL.library = learners, V = 2, cvControl = list(V = 2))
```

vimp_se

Estimate variable importance standard errors

Description

Compute standard error estimates for estimates of variable importance.

Usage

vimp_se(
  eif_full,
  eif_reduced,
  cross_fit = TRUE,
  sample_split = TRUE,
  na.rm = FALSE
)

Arguments

- **eif_full**: the estimated efficient influence function (EIF) based on the full set of covariates.
- **eif_reduced**: the estimated EIF based on the reduced set of covariates.
- **cross_fit**: logical; was cross-fitting used to compute the EIFs? (defaults to TRUE)
- **sample_split**: logical; was sample-splitting used? (defaults to TRUE)
- **na.rm**: logical; should NA's be removed in computation? (defaults to FALSE).

Details

See the paper by Williamson, Gilbert, Simon, and Carone for more details on the mathematics behind this function and the definition of the parameter of interest.

Value

The standard error for the estimated variable importance for the given group of left-out covariates.

---

vrc01

Neutralization sensitivity of HIV viruses to antibody VRC01

Description

A dataset containing neutralization sensitivity – measured using inhibitory concentration, the quantity of antibody necessary to neutralize a fraction of viruses in a given sample – and viral features including: amino acid sequence features (measured using HXB2 coordinates), geographic region of origin, subtype, and viral geometry. Accessed from the Los Alamos National Laboratory’s (LANL’s) Compile, Analyze, and tally Neutralizing Antibody Panels (CATNAP) database.
Usage

data("vrc01")

Format

A data frame with 611 rows and 837 variables:

- **seqname**: Viral sequence identifiers
- **subtype.is.01_AE**: Dummy variables encoding the viral subtype as 0/1. Possible subtypes are 01_AE, 02_AG, 07_BC, A1, A1C, A1D, B, C, D, O, Other.
- **subtype.is.02_AG**: Dummy variables encoding the viral subtype as 0/1. Possible subtypes are 01_AE, 02_AG, 07_BC, A1, A1C, A1D, B, C, D, O, Other.
- **subtype.is.07_BC**: Dummy variables encoding the viral subtype as 0/1. Possible subtypes are 01_AE, 02_AG, 07_BC, A1, A1C, A1D, B, C, D, O, Other.
- **subtype.is.A1**: Dummy variables encoding the viral subtype as 0/1. Possible subtypes are 01_AE, 02_AG, 07_BC, A1, A1C, A1D, B, C, D, O, Other.
- **subtype.is.A1C**: Dummy variables encoding the viral subtype as 0/1. Possible subtypes are 01_AE, 02_AG, 07_BC, A1, A1C, A1D, B, C, D, O, Other.
- **subtype.is.A1D**: Dummy variables encoding the viral subtype as 0/1. Possible subtypes are 01_AE, 02_AG, 07_BC, A1, A1C, A1D, B, C, D, O, Other.
- **subtype.is.B**: Dummy variables encoding the viral subtype as 0/1. Possible subtypes are 01_AE, 02_AG, 07_BC, A1, A1C, A1D, B, C, D, O, Other.
- **subtype.is.C**: Dummy variables encoding the viral subtype as 0/1. Possible subtypes are 01_AE, 02_AG, 07_BC, A1, A1C, A1D, B, C, D, O, Other.
- **subtype.is.D**: Dummy variables encoding the viral subtype as 0/1. Possible subtypes are 01_AE, 02_AG, 07_BC, A1, A1C, A1D, B, C, D, O, Other.
- **subtype.is.O**: Dummy variables encoding the viral subtype as 0/1. Possible subtypes are 01_AE, 02_AG, 07_BC, A1, A1C, A1D, B, C, D, O, Other.
- **subtype.is.Other**: Dummy variables encoding the viral subtype as 0/1. Possible subtypes are 01_AE, 02_AG, 07_BC, A1, A1C, A1D, B, C, D, O, Other.
- **geographic.region.of.origin.is.Asia**: Dummy variables encoding the geographic region of origin as 0/1. Regions are Asia, Europe/Americas, North Africa, and Southern Africa.
- **geographic.region.of.origin.is.Europe.Americas**: Dummy variables encoding the geographic region of origin as 0/1. Regions are Asia, Europe/Americas, North Africa, and Southern Africa.
- **geographic.region.of.origin.is.N.Africa**: Dummy variables encoding the geographic region of origin as 0/1. Regions are Asia, Europe/Americas, North Africa, and Southern Africa.
- **geographic.region.of.origin.is.S.Africa**: Dummy variables encoding the geographic region of origin as 0/1. Regions are Asia, Europe/Americas, North Africa, and Southern Africa.
- **ic50.censored**: A binary indicator of whether or not the IC-50 (the concentration at which 50% of the viral population is inhibited) is a proxy for a resistant virus.
- **ic80.censored**: A binary indicator of whether or not the IC-80 (the concentration at which 80% of the viral population is inhibited) is a proxy for a resistant virus.
**ic50.geometric.mean.imputed** Continuous IC-50. If neutralization sensitivity for the virus was assessed in multiple studies, the geometric mean was taken.

**ic80.geometric.mean.imputed** Continuous IC-90. If neutralization sensitivity for the virus was assessed in multiple studies, the geometric mean was taken.

**hxb2.46.E.1mer** Amino acid sequence features denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site. For example, hxb2.46.E.1mer records the presence of an E at HXB2-referenced site 46.

**hxb2.46.K.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.46.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.46.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.46.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.61.H.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.61.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.61.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.61.Y.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.97.E.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.97.K.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.97.N.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.97.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.124.F.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.124.P.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.125.I.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.125.L.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.127.I.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.127.V.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hxb2.130.D.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.130.E.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.130.H.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.130.I.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.130.K.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.130.L.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.130.N.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.130.Q.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.130.R.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.130.S.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.132.A.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.132.E.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.132.G.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.132.H.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.132.I.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.132.K.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.132.N.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.132.Q.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.132.R.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.132.S.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.132.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.132.V.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.132.X.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.132.Y.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.138.A.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.138.C.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.138.D.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.138.E.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.138.G.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.138.H.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.138.I.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.138.K.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.138.L.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.138.M.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.138.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.138.P.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.138.Q.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.138.R.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.138.S.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.138.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.138.V.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hxb2.138.V.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.138.gap.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.139.A.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.139.C.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.139.D.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.139.E.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.139.G.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.139.I.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.139.K.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.139.L.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.139.N.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.139.Q.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.139.R.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.139.S.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.139.T.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.139.V.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.143.A.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.143.D.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.143.E.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.143.F.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hxb2.143.G.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.143.I.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.143.K.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.143.N.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.143.P.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.143.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.143.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.143.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.143.V.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.143.Y.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.143.gap.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.144.A.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.144.D.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.144.E.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.144.G.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.144.H.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.144.I.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.144.K.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.144.L.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.144.N.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.144.P.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hxb2.144.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.144.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.144.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.144.V.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.144.Y.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.144.gap.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.150.A.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.150.D.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.150.E.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.150.G.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.150.I.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.150.K.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.150.L.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.150.M.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.150.N.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.150.P.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.150.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.150.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.150.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.150.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.150.V.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hxb2.150.Y.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.150.gap.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.156.D.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.156.I.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.156.K.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.156.N.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.156.Q.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.156.S.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.156.T.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.156.V.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.156.Y.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.179.A.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.179.I.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.179.L.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.179.M.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.179.P.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.179.Q.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.179.S.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.179.T.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.179.V.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.179.Y.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.181.I.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.181.L.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.181.M.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.181.V.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hxb2.186.A.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.186.D.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.186.E.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.186.G.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.186.K.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.186.N.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.186.P.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.186.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.186.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.186.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.186.gap.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.187.A.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.187.D.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.187.E.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.187.G.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.187.K.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.187.N.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.187.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.187.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.187.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.187.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
\texttt{hxb2.187.gap.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.190.D.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.190.E.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.190.F.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.190.G.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.190.I.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.190.K.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.190.L.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.190.M.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.190.N.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.190.Q.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.190.R.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.190.S.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.190.T.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.190.V.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.190.Y.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.197.D.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.197.K.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.197.N.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.198.A.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

\texttt{hxb2.198.S.1mer} Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.198.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.198.V.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.241.D.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.241.K.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.241.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.241.S.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.276.D.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.276.K.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.276.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.276.S.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.278.A.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.278.E.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.278.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.278.S.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.278.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.279.D.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.279.E.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.279.K.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.279.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.279.S.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.280.D.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.280.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.280.S.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.280.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.281.A.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.281.E.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.281.G.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.281.I.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.281.S.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.281.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.281.V.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.282.G.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.282.K.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.282.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.282.Q.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.282.R.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.282.Y.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.283.I.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.283.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.283.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.283.V.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.289.A.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hxb2.289.D.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.289.K.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.289.N.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.289.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.289.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.289.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.289.V.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.290.D.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.290.E.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.290.G.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.290.K.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.290.L.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.290.N.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.290.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.290.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.290.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.290.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.290.X.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.321.D.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.321.E.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.321.G.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.321.K.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.321.L.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.321.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.321.R.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.321.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.321.V.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.321.Y.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.321.gap.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.328.A.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.328.E.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.328.I.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.328.K.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.328.L.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.328.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.328.Q.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.328.R.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.339.D.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.339.E.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.339.G.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.339.H.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.339.I.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.339.K.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.339.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hxb2.339.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.339.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.339.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.339.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.339.V.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.339.Y.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.354.E.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.354.G.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.354.H.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.354.I.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.354.K.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.354.L.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.354.N.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.354.P.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.354.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.354.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.354.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.354.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.354.gap.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.355.D.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.355.G.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.355.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.355.K.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.355.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.355.Q.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.355.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.355.gap.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.362.A.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.362.D.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.362.E.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.362.G.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.362.K.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.362.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.362.Q.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.362.R.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.362.S.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.362.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.363.A.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.363.E.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.363.G.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.363.H.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.363.K.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hxb2.363.N.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.363.P.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.363.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.363.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.363.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.363.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.363.X.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.365.A.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.365.L.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.365.N.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.365.P.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.365.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.365.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.365.V.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.369.A.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.369.L.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.369.P.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.369.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.369.V.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.371.L.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.371.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.371.V.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.374.F.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.374.H.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.374.L.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.386.D.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.386.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.386.S.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.386.X.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.386.Y.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.389.A.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.389.D.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.389.E.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.389.G.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.389.H.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.389.L.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.389.K.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.389.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.389.P.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hxb2.389.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.389.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.389.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.389.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.392.D.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.392.H.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.392.K.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.392.N.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.392.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.392.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.392.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.392.Y.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.392.gap.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.394.A.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.394.D.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.394.E.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.394.I.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.394.K.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.394.N.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.394.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.394.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hxb2.394.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.394.V.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.394.gap.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.396.A.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.396.D.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.396.E.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.396.F.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.396.G.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.396.H.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.396.I.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.396.K.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.396.L.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.396.M.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.396.N.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.396.P.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.396.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.396.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.396.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.396.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.396.W.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.396.Y.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hxb2.396.gap.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.A.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.C.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.D.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.E.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.F.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.G.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.H.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.I.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.K.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.L.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.N.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.P.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.Q.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.R.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.S.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.T.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.V.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.W.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.X.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
<tr>
<td>hxb2.397.Y.1mer</td>
<td>Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.</td>
</tr>
</tbody>
</table>
hxb2.397.gap.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.A.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.D.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.E.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.F.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.G.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.H.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.I.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.K.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.L.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.M.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.N.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.P.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.Q.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.R.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.S.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.T.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.V.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.W.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.Y.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.406.gap.1mer Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.408.A.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.408.D.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.408.E.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.408.F.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.408.G.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.408.H.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.408.I.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.408.K.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.408.L.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.408.M.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.408.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.408.P.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.408.Q.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.408.R.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.408.S.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.408.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.408.V.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.408.Y.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.408.gap.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.410.D.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.410.E.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.410.G.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.410.H.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.410.I.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.410.K.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.410.L.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.410.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.410.P.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.410.R.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.410.S.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.410.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.410.V.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.410.Y.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.410.gap.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.415.E.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.415.I.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.415.K.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.415.L.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.415.M.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.415.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.415.R.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.415.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.415.V.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.415.X.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.415.Y.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.425.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.425.R.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.426.L.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.426.M.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.426.R.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.426.S.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.426.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.428.I.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.428.M.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.428.Q.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.429.A.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.429.E.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.429.G.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.429.K.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.429.Q.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.429.R.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.429.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hx**b2.430.A.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.430.G.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.430.I.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.430.T.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.430.V.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.431.E.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.431.G.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.432.K.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.432.Q.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.432.R.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.432.S.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.442.A.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.442.E.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.442.H.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.442.I.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.442.K.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.442.L.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.442.N.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.442.P.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.442.Q.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hx**b2.442.R.1**mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.442.S.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.442.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.442.V.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.448.D.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.448.E.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.448.I.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.448.K.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.448.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.448.Q.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.448.S.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.448.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.455.E.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.455.I.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.455.L.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.455.Q.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.455.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.455.V.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.456.H.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.456.I.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.456.M.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.456.R.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hxb2.456.S.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.456.W.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.456.Y.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.457.D.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.458.E.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.458.G.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.458.Q.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.458.Y.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.459.D.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.459.E.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.459.G.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.459.P.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.459.S.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.459.gap.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.460.A.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.460.D.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.460.E.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.460.G.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.460.I.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.460.K.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.460.N.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hxb2.460.P.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.460.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.460.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.460.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.460.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.460.V.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.460.gap.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.461.A.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.461.D.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.461.E.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.461.G.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.461.H.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.461.I.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.461.K.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.461.N.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.461.P.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.461.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.461.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.461.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.461.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.461.V.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hxb2.461.gap.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.462.A.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.462.D.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.462.E.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.462.G.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.462.I.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.462.K.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.462.N.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.462.P.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.462.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.462.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.462.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.462.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.462.X.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.462.gap.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.463.A.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.463.D.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.463.E.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.463.G.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.463.K.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.463.M.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hxb2.463.N.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.463.P.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.463.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.463.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.463.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.463.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.465.A.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.465.D.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.465.E.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.465.I.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.465.K.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.465.N.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.465.P.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.465.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.465.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.465.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.466.A.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.466.D.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.466.E.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.466.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.466.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hxb2.466.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.466.V.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.467.I.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.467.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.467.V.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.469.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.471.A.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.471.E.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.471.G.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.471.I.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.471.Q.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.471.S.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.471.T.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.471.V.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.474.D.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.474.E.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.474.N.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.475.I.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.475.M.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.476.K.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.476.R.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hxb2.477.D.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.477.N.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.544.L.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.544.V.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.569.S.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.569.T.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.569.X.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.589.D.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.589.N.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.655.E.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.655.I.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.655.K.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.655.N.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.655.Q.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.655.R.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.655.S.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.655.T.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.668.D.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.668.G.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.668.N.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.668.S.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.668.T.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.675.I.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.675.L.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.677.H.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.677.K.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.677.N.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.677.Q.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.677.R.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.677.S.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.680.W.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.681.Y.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.683.K.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.683.Q.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.683.R.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.688.I.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.688.V.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.702.F.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.702.I.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.702.L.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.702.V.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.29.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hxb2.49.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.59.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.88.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.130.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.132.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.133.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.134.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.135.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.136.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.137.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.138.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.139.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.140.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.141.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.142.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.143.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.144.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.145.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.146.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.147.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.148.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
hxb2.149.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.150.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.156.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.160.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.171.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.185.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.186.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.187.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.188.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.197.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.229.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.230.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.232.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.234.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.241.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.268.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.276.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.278.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.289.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.293.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.295.sequon_actual.1mer  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
**hxb2.301.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.302.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.324.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.332.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.334.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.337.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.339.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.343.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.344.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.350.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.354.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.355.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.356.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.358.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.360.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.362.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.363.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.386.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.392.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.393.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.394.sequon_actual.1mer** Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.
Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

hxb2.395.sequon_actual.1mer

hxb2.396.sequon_actual.1mer

hxb2.397.sequon_actual.1mer

hxb2.398.sequon_actual.1mer

hxb2.399.sequon_actual.1mer

hxb2.400.sequon_actual.1mer

hxb2.401.sequon_actual.1mer

hxb2.402.sequon_actual.1mer

hxb2.403.sequon_actual.1mer

hxb2.404.sequon_actual.1mer

hxb2.405.sequon_actual.1mer

hxb2.406.sequon_actual.1mer

hxb2.407.sequon_actual.1mer

hxb2.408.sequon_actual.1mer

hxb2.409.sequon_actual.1mer

hxb2.410.sequon_actual.1mer

hxb2.411.sequon_actual.1mer

hxb2.412.sequon_actual.1mer

hxb2.413.sequon_actual.1mer

hxb2.442.sequon_actual.1mer

hxb2.444.sequon_actual.1mer
**hxb2.446.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.448.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.460.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.461.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.462.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.463.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.465.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.611.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.616.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.618.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.619.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.624.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.625.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.637.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.674.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.743.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.750.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.787.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.816.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**hxb2.824.sequon_actual.1mer**  Amino acid sequence feature denoting the presence (1) or absence (0) of a residue at the given HXB2-referenced site.

**sequons.total.env**  The total number of sequons in various areas of the HIV viral envelope protein.

**sequons.total.gp120**  The total number of sequons in various areas of the HIV viral envelope protein.
sequons.total.v5 The total number of sequons in various areas of the HIV viral envelope protein.
sequons.total.loop.d The total number of sequons in various areas of the HIV viral envelope protein.
sequons.total.loop.e The total number of sequons in various areas of the HIV viral envelope protein.
sequons.total.vrc01 The total number of sequons in various areas of the HIV viral envelope protein.
sequons.total.cd4 The total number of sequons in various areas of the HIV viral envelope protein.
sequons.total.sj.fence The total number of sequons in various areas of the HIV viral envelope protein.
sequons.total.sj.trimer The total number of sequons in various areas of the HIV viral envelope protein.
cysteines.total.env The number of cysteines in various areas of the HIV viral envelope protein.
cysteines.total.gp120 The number of cysteines in various areas of the HIV viral envelope protein.
cysteines.total.v5 The number of cysteines in various areas of the HIV viral envelope protein.
cysteines.total.vrc01 The number of cysteines in various areas of the HIV viral envelope protein.
length.env The length of various areas of the HIV viral envelope protein.
length.gp120 The length of various areas of the HIV viral envelope protein.
length.v5 The length of various areas of the HIV viral envelope protein.
length.v5.outliers The length of various areas of the HIV viral envelope protein.
length.loop.e The length of various areas of the HIV viral envelope protein.
length.loop.e.outliers The length of various areas of the HIV viral envelope protein.
taylor.small.total.v5 The steric bulk of residues at critical locations.
taylor.small.total.loop.d The steric bulk of residues at critical locations.
taylor.small.total.cd4 The steric bulk of residues at critical locations.

Source
https://github.com/benkeser/vrc01/blob/master/data/fulldata.csv
Index

* datasets
  vrc01, 83
average_vim, 3
boot, 5, 11, 51
bootstrap_se, 4
check_fitted_values, 5
check_inputs, 6
create_z, 7
CV.SuperLearner, 22
cv_vim, 8
est_predictiveness, 18
est_predictiveness_cv, 19
estimate, 14
estimate.predictiveness_measure, 15
estimate.elf_projection, 15
estimate.nuisances, 16
estimate.type_predictiveness, 17
extract_sampled_split_predictions, 21
format.predictiveness_measure, 22
format.vim, 22
get_cv_sl_folds, 23
get_full_type, 23
get_test_set, 24
make_folds, 24
make_kfold, 25
measure.accuracy, 25
measure.anova, 27
measure.auc, 28
measure.average_value, 29
measure.cross_entropy, 31
measure.deviance, 32
measure.mse, 33
measure.r_squared, 35
merge_vim, 36
performance, 67
predictiveness_measure, 37
print.predictiveness_measure, 39
print.vim, 40
process_arg_lst, 40
run_sl, 41
sample_subsets, 42
scale_est, 43
sp_vim, 45
spvim.ics, 43, 45
spvim.se, 44
SuperLearner, 13, 48, 52, 59, 62, 67, 73, 77, 82
vim, 49
vimp, 53
vimp.accuracy, 55
vimp.anova, 60
vimp.auc, 63
vimp.ci, 68
vimp.deviance, 69
vimp.hypothesis_test, 74
vimp.regression, 75
vimp.r_squared, 78
vimp.r_squared, 78
vimp.r_squared, 83
vrc01, 83