Package 'RFPM'

May 7, 2026

Title Floating Percentile Model

Version 1.1

Date 2023-10-24

Description Floating Percentile Model with additional functions for optimizing inputs and evaluating outputs and assumptions.

Imports stats, graphics, grDevices, tidyr, dplyr, lawstat

Suggests knitr, rmarkdown

VignetteBuilder knitr

License GPL (>= 3)

Encoding UTF-8

LazyData true

Maintainer Brian Church <brianc@windwardenv.com>

RoxygenNote 7.2.3

Depends R (>= 3.5.0)

NeedsCompilation no

Author Brian Church [aut, cre] (ORCID:
<https://orcid.org/0000-0003-2220-0153>),
Claire Detering [aut]

Repository CRAN

Date/Publication 2023-10-25 17:40:02 UTC

Contents

`c.northport` *Northport Data,* Chironomus dilutus *Toxicity*

Description

Example dataset

Format

A data.frame with 147 rows and 15 variables:

Sample sediment sample ID

Al aluminum concentration, mg/kg dry weight

As arsenic concentration, mg/kg dry weight

Cu copper concentration, mg/kg dry weight

Cd cadmium concentration, mg/kg dry weight

Cr chromium concentration, mg/kg dry weight

Fe iron concentration, mg/kg dry weight

Pb lead concentration, mg/kg dry weight

Hg mercury concentration, mg/kg dry weight

Ni nickel concentration, mg/kg dry weight

Zn zinc concentration, mg/kg dry weight

Organism organism abbreviation, `CD` = *Chironomus dilutus*

Meas_Day measurement day for toxicity endpoint (post-initiation)

Endpoint toxicity endpoint

Hit logical; whether the sample was classified as toxic

Details

`c.northport` is a field-collected dataset containing sediment chemical concentrations for metals and toxicity classification data for the midge *Chironomus dilutus*. Data are from a freshwater site in Washington State (Dowling and Roland 2019).

References

Dowling B, Roland J. 2019. Establishment of site-specific SMS metals cleanup objectives for contaminated sediments - Northport waterfront and nearshore state cleanup site. Washington State Department of Ecology.

`chemSig` *Chemical Variable Selection within the Floating Percentile Model*

Description

Determine which chemicals in a dataset have significantly higher concentrations among toxic samples by comparison to non-toxic samples

Usage

```
chemSig(
  data,
  paramList,
  testType = NULL,
  alpha.var = 0.05,
  alpha.norm = 0.05,
  alpha.test = 0.05,
  alpha = NULL,
  alternative = "less",
  var.alternative = "two.sided",
  var.equal = NULL,
  warn = TRUE,
  ExcelMode = NULL
)
```

Arguments

`data`	data.frame containing, at a minimum, chemical concentrations as columns and a logical `Hit` column classifying toxicity
`paramList`	character vector of column names of chemical concentration variables in `data`
`testType`	character string; whether to run parametric or non-parametric tests (default = `NULL`). See Details for more information.
`alpha.var`	numeric value between 0 and 1; type-I error rate for testing equal variance assumption (default = `0.05`)
`alpha.norm`	numeric value between 0 and 1; type-I error rate for testing normality assumption (default = `0.05`)
`alpha.test`	numeric value between 0 and 1; type-I error rate for testing differences between Hit and No-hit datasets (default = `0.05`)
`alpha`	numeric value between 0 and 1; type-I error rate that, if supplied by the user, will be applied to all tests (default = `NULL`)

alternative	alternative hypothesis type for equality of central tendency (default = "less")
var.alternative	alternative hypothesis type for equal variance test (default = "two.sided")
var.equal	logical; whether to assume equal variance (default = NULL)
warn	logical; whether to generate a warning associated with ExcelMode (default = TRUE)
ExcelMode	logical; whether to force chemSig to run like the WA Department of Ecology's Excel-based floating percentile model calculator (default = FALSE)

Details

chemSig is called within FPM via chemSigSelect, which generates a subset of chemicals to pass into the floating percentile model algorithm. chemSig only returns a logical vector describing which parameters in paramList should be selected for benchmark development based on having significantly higher concentrations when Hit == TRUE than when Hit == FALSE. The user has the ability to manipulate several of the parameters of the selection algorithm, or they can allow chemSig to test for assumptions and use appropriate hypothesis tests based on those results. By default, chemSig will use shapiro.test to confirm normality, then either var.test if the data are normal or fligner.test if the data are non-normal to confirm equal variance. Finally, the function will use t.test if the data are normal (using the Welch method if unequal variance), wilcox.test if non-normal with equal variance, or brunner.munzel.test if non-normal with unequal variance.

The testType argument can be one of p, P, param, Param, parametric, or Parametric for parametric test types or non, Non, np, NP, nonparam, Nonparam, non-param, Non-param, nonparametric, Nonparametric, non-parametric, Non-parametric, or Non-Parametric.

The user has the option of providing a single alpha level to apply to all tests (e.g., 0.05) or to specify test-specific alpha levels via alpha.var, alpha.norm, and alpha.test. Note that FPM by default uses alpha = 0.05 for all tests.

While alternative and var.alternative can be adjusted, we strongly recommend that they not be changed from the default values. For example, changing alternative from "less" (default) to "two.sided" would result in the assumption that chemical concentrations could be significantly higher (as well as lower) when there is no toxicity than when there is, which is inappropriate. The "greater" alternative, which is never appropriate, is not an accepted input for alternative. Similarly, the assumption of equal variance relates to a "two.sided" argument, therefore changing the var.alternative to be "less" or "greater" would not be appropriate.

ExcelMode assumes testType = "parametric", var.equal = TRUE, alternative = "less", and alpha = 0.1. In actuality, the Excel-based tool uses a one-way ANOVA test to compare two levels of Hit, which is equivalent to a t-test so long as alpha is adjusted to 0.1. Thus, testType, alternative, var.alternative, and var.equal are overridden when ExcelMode = TRUE. This argument was included for those interested in using 'RFPM' as an alternative to the Excel-based calculator tool to obtain identical benchmark results. Note that the Excel FPM tool also includes other features which may complicate comparability such as outlier analysis. Outlier analysis is not conducted by RFPM.

Value

named logical vector

Examples

```
paramList = c("Cd", "Cu", "Fe", "Mn", "Ni", "Pb", "Zn")
chemSig(h.tristate, paramList, "nonparametric")
chemSig(h.tristate, paramList, "parametric")
```

`chemSigSelect` *Chemical Data Selection within the Floating Percentile Model*

Description

Generate a dataset of concentrationso for chemicals with significantly higher concentrations among toxic samples by comparison to non-toxic samples

Usage

```
chemSigSelect(data, paramList, plot = FALSE, ...)
```

Arguments

`data`	data.frame containing, at a minimum, chemical concentrations as columns and a logical `Hit` column classifying toxicity
`paramList`	character vector naming columns of `data` containing concentrations
`plot`	logical value indicating whether or not to generate figures comparing concentrations between subsets where `Hit == TRUE` and `Hit == FALSE` (default = `FALSE`)
`...`	additional arguments passed to `chemSig`

Details

`chemSigSelect` is used within `FPM` to quickly run `chemSig` and export concentrations from `data` for the significant chemicals to include in the floating percentile model algorithm. Results are exported in list with two data.frames, separated into chemicals that were significant (`"sig"`) and non-significant (`"nonsig"`) For information on `data` and `paramList`, see details of `?FPM`.

If `plot = TRUE`, then a series of plots will be exported comparing the Hit/No-hit data subsets for all chemicals in `paramList`. Plots with blue lines are generated from significant chemicals, and plots with green lines are generated from non-significant chemicals. The user currently has only limited control over graphical parameters of plots, and attempts to change graphical parameters are likely to cause errors.

Value

list of two data.frames (`"sig"` and `"nonsig"`), `chemSigSelect` class object

See Also

plot.chemSigSelect, FPM, chemSig

Examples

```
paramList = c("Cd", "Cu", "Fe", "Mn", "Ni", "Pb", "Zn")
chemSigSelect(h.tristate, paramList)$sig[1:6,]
chemSigSelect(h.tristate, paramList, testType = "p")$sig[1:6,]
```

---

chemVI *Chemical Variable Importance for Floating Percentile Model Benchmarks*

Description

Generate statistics describing the relative importance of chemicals among benchmarks generated by FPM

Usage

```
chemVI(data, paramList, ...)
```

Arguments

data	data.frame containing, at a minimum, chemical concentrations as columns and a logical Hit column classifying toxicity
paramList	character vector naming columns of data containing concentrations
...	additional arguments passed to chemSig, chemSigSelect, and FPM

Details

The purpose of chemVI is to inform the user about the relative influence of each chemical over the sediment quality benchmarks generated by FPM. Three statistics are generated: chemDensity, MADP, dOR, dFM, and dMCC. The chemDensity statistic (which is also generated by FPM) describes how little a particular chemical's value increased within the floating percentile model algorithm. Low chemDensity (close to 0) means that the value was able to increase substantially within the algorithm without triggering one or more of the criteria for stopping the algorithm (see ?FPM), whereas high chemDensity (close to 1) indicates the final benchmark for that chemical did not float (increase) much before being locked in. In other words, low chemDensity might be interpreted as relatively low importance. We caution against using this metric in isolation, as it is the more difficult to interpret of the three. The MADP statistic (or mean absolute difference percent) is calculated by sequentially dropping each chemical from consideration, recalculating the benchmarks for the remaining chemicals, and then determining how much each benchmark changed (as a percent of the original value). Thus, the MADP is a measure of a chemical's influence over other benchmarks. The dOR statistic is the difference between the overall reliability of benchmarks with all chemicals versus without each chemical. dFM and dMCC are similar to the dOR statistic, but for the Fowlkes-Mallows Index and Matthew's Correlation Coefficient. In any case, larger positive values indicate a greater impact of a chemical on the overall predictive performance of floating percentile model benchmarks. Small values (close to 0) indicate low influence. Larger negative values indicate that the chemical actually adversely impacts toxicity predictions. If there are chemicals with negative values, consider reevaluting the data without the associated chemical or using optimFPM or cvFPM to optimize the overall reliability prior to running FPM and chemVI.

Value

data.frame with 2 columns

See Also

chemSig, chemSigSelect, optimFPM, cvFPM, FPM

Examples

```
paramList = c("Cd", "Cu", "Fe", "Mn", "Ni", "Pb", "Zn")
chemVI(h.tristate, paramList, testType = "np")
chemVI(h.tristate, paramList, testType = "p")
```

---

colorGradient *Color Gradient Generator*

Description

Generate a gradient of hexadecimal colors from a numeric vector

Usage

```
colorGradient(x, colors = heat.colors(10), colsteps = 100, na.rm = TRUE, ...)
```

Arguments

x	numeric vector
colors	values recognizable by R as colors - text, hexadecimal, numbers, etc. (default = grDevices::heat.colors(10)).
colsteps	numeric value, number of unique colors to include in gradient
na.rm	logical value (default = TRUE) indicating whether to exclude NA values in x
...	additional arguments passed to grDevices::colorRamp

Details

colorGradient generates a color gradient based on a numeric input x. The original version (named 'color.gradient') was written by David Hoop in a 2016 stack.overflow response to question (online). The function was modified slightly for adaptability of inputs. Note that the full color gradient is used if possible, which can exaggerate small differences in x. This function is applied by optimFPM when generating color-based optimization matrix graphics (i.e., when optimizing both alpha and FN_crit).

Value

character vector

See Also

optimFPM, heat.colors, colorRamp

Examples

```
x <- rnorm(n = 100)
cols <- colorGradient(x, c("red", "white", "blue"))
plot(x, col = cols)
```

---

cvFPM *Cross-Validation Optimization of the Floating Percentile Model*

Description

Use leave-one-out (LOO) or k-folds cross-validation methods to calculate parameter inputs that optimize benchmark performance while attempting to account for out-of-sample uncertainty

Usage

```
cvFPM(
  data,
  paramList,
  FN_crit = seq(0.1, 0.9, by = 0.05),
  alpha.test = seq(0.05, 0.5, by = 0.05),
  k = 5,
  seed = 1,
  plot = TRUE,
  which = c(1, 2, 3, 4),
  colors = heat.colors(10),
  colsteps = 100,
  ...
)
```

Arguments

data	data.frame containing, at a minimum, chemical concentrations as columns and a logical Hit column classifying toxicity
paramList	character vector of column names of chemical concentration variables in data
FN_crit	numeric vector of values between 0 and 1 indicating false negative threshold for floating percentile model benchmark selection (default = seq(0.1, 0.9, 0.05))
alpha.test	numeric vector of values between 0 and 1 indicating type-I error rate for chemical selection (default = seq(0.05, 0.5, by = 0.05))
k	integer with length = 1 and value > 1 indicating how many folds to include in k-folds type cross-validation method (default = 5)

seed	integer with length = 1 indicating the random seed to set for assigning k classes for k-folds cross-validation (default = 1)
plot	whether to plot the output of cvFPM (default = TRUE)
which	numeric or character indicating which type of plot to generate (see Details; default = c(1, 2, 3, 4))
colors	values recognizible as colors to be passed to colorRampPalette (via colorGradient) to generate a palette for plotting (default = heat.colors(10))
colsteps	integer; number of discrete steps to interpolate colors in colorGradient (default = 100)
...	additional arguments passed to chemSig and FPM

Details

cvFPM allows users to "tune" the FN_crit and alpha.test arguments used by FPM (via chemSig). This is achieved by splitting the empirical dataset into "test" and "training" subsets, calculating benchmarks for the training set, and then evaluating the benchmarks' ability to predict Hits in the out-of-sample test set. The output of cvFPM is similar to optimFPM: optimal FN_crit and alpha.test inputs based on several classification metrics (see ?optimFPM for more details). The key difference between cvFPM and optimFPM is that cvFPM attempts to account for out-of-sample uncertainty, whereas optimFPM is specific (and potentially overly specific) to the full FPM dataset. Because the primary use of FPM SQBs will be to predict toxicity in sediment samples where toxicity is not measured, the FPM should be parameterized in a way that best accounts for out-of-sample uncertainty. In other words, while FPM generates classification metrics like "overall reliability" for SQBs, they are unlikely to achieve the expected reliability when applied to new samples. This is an inherent limitation of SQBs, which the FPM cannot fully address but that cvFPM considers.

Two cross-validation methods are available, controlled through the k argument. If the user specifies k = NULL or k = nrow(data), then leave-one-out (LOO) is used. LOO is computationally intensive but is better suited to small datasets. Other values of k (e.g., the default k = 5) will result in applying a k-folds cross-validation method, which uses larger test subsets (and smaller training sets), evaluates fewer scenarios, and greatly improves runtime for large datasets. The seed argument can be used to establish a consistent result; if is.null(seed), the result will vary based on randomization. Allowing for randomization may be desireable to understand between-run variability in cvFPM output caused by re-sampling of training/test sets.

By setting plot = TRUE (the default), the outcome of cross-validation can be visualized over the range of FN_crit values considered. Visualizing the results can inform the user about variability in the cross-validation process, ranges of potentially reasonable FN_crit values, etc. Graphical output depends on whether many FN_crit and/or many alpha.test are evaluated, with line plots or heat plots alternately generated.

IMPORTANT: cvFPM is not in itself optimized for runtime - running cvFPM can take a long time

The which argument can be used to specify which of the metric-specific plots should be generated when plot = TRUE. Inputs to which are, by default, c(1, 2, 3, 4).

Value

data.frame of metric output, base R graphical output

See Also

chemSig, FPM, optimFPM

Examples

```
paramList = c("Cd", "Cu", "Fe", "Mn", "Ni", "Pb", "Zn")
par(mfrow = c(2,2))
cvFPM(h.tristate, paramList, seq(0.1, 0.9, 0.1), 0.05)
```

FPM *Floating Percentile Model*

Description

Generate sediment quality benchmarks using the floating percentile model algorithm

Usage

```
FPM(
  data,
  paramList,
  FN_crit = 0.2,
  paramFixed = NULL,
  paramOverride = FALSE,
  increment = 10,
  precision = 0.1,
  empirical = TRUE,
  defIter = 5,
  seed = 1,
  densInfo = FALSE,
  lockInfo = FALSE,
  hitInfo = FALSE,
  ...
)
```

Arguments

`data`	data.frame containing, at a minimum, chemical concentrations as columns and a logical `Hit` column classifying toxicity
`paramList`	character vector of column names of chemical concentration variables in `data`
`FN_crit`	numeric vector of values between 0 and 1 indicating false negative threshold(s) for benchmark selection (default = `0.2`)
`paramFixed`	character vector of column names of chemical concentration variables to retain, bypassing testing for specific chemicals (default = `NULL`). See Details.
`paramOverride`	logical; whether to retain every chemical variable in `paramList` (default = `FALSE`). See Details.

increment	numeric value greater than 1; number of increments to evaluate (default = 10). See Details.
precision	numeric value between 0 and 1 (default = 0.1)
empirical	logical; whether to return the highest empirical value meeting acceptable conditions of the FPM algorithm (default = TRUE)
defIter	numeric value greater than 0; default number of iterations to use in the case of negative or zero values in data (default = 5)
seed	random seed to set for reproducible results; only for handling edge cases of ranking ties (default = 1)
densInfo	logical; whether to return the "density" statistic defining how much FPM criteria changed within the algorithm (default = FALSE)
lockInfo	logical; whether to return the reason for and order in which benchmarks were "locked" within the model algorithm (default = FALSE). See Details.
hitInfo	logical; whether to return the predicted Hit results as part of the output (default = FALSE)
...	additional argument passed to chemSigSelect and chemSig

Details

FPM is the main function provided in 'RFPM', which was developed firstly as a redevelopment of the Washington Department of Ecology's Excel-based floating percentile model tool (Avocet 2003; Ecology 2011), and secondly as a means to evaluate uncertainties and sensitivities associated with the model. FPM generates sediment quality benchmarks for chemicals with significantly higher concentrations among Hit samples (meaning they were determined to be categorically toxic).

FPM is an algorithmic approach to setting sediment quality benchmarks using sediment chemistry data and toxicity test results. Toxicity is treated as a binary classification - either a Hit == TRUE or Hit == FALSE (meaning toxic or non-toxic) by some user-defined definition. The most important input to FPM apart from the empirical data is FN_crit, which determines an upper limit for false negative errors associated with floating percentile model benchmarks. The default FN_crit recommended by the Department of Ecology is 0.2; though intended to be protective, the value of 0.2 is arbitrary. We recommend that the user run the optimFPM and/or cvFPM functions to find the FN_crit value(s) that optimize benchmark performance within an acceptable error range for the site. optimFPM can also help users optimize the alpha parameter (see ?chemSig), which is also somewhat arbitrarily set at a conventional default of 0.05.

There are two arguments that have defaults in FPM that the user may desire to change in certain circumstances, but that we generally recommend not changing without good reason. These are paramFixed and paramOverride, which override the chemical selection process, resulting in potentially non-toxic chemicals being assigned benchmarks. The paramFixed argument, which only forces named chemicals into the model algorithm, is looser than paramOverride, which forces all chemicals in paramList into the model algorithm. See ?chemSig for more information regarding default parameters used within FPM. Even if chemical names are supplied to paramFixed, FPM will still use hypothesis testing methods to consider all other chemicals for inclusion.

increment determines (inversely) how large or small values should be that are added to percentile values in the model algorithm. A larger increment results in smaller incremental additions and vice-versa. The WA Department of Ecology recommends a default of increment = 10. This is a

reasonable value, and we recommend not decreasing increment below 10. Increasing increment will increase computation time, and may or may not result in more accurate benchmarks. So, we recommend not increasing increment much higher than 10.

precision determines how many iterative loops will be attempted within the model algorithm when trying to increase each benchmark. If increasing the benchmark would increase the false negative rate above FN_crit, the benchmark would then be decreased, the increment size is divided by increment, and then the smaller incremental addition is used to increase the benchmark. This process repeats for a fixed number of iterations, which is related to precision. If the benchmark cannot be increased after the fixed number of iterations, the benchmark is locked in place. The default value for precision is 0.1, but the value could be lower, if desired. Lowering the value will increase computation time and may or may not result in more accurate benchmarks. In general, we recommend reducing precision rather than increasing increment in order to potentially enhance the precision of benchmark calculations.

empirical by default returns empirical concentrations from data that meets the conditions of the FPM. The user can set this argument FALSE if an exact FPM calculation is desired. The exact calculation will still meet the FPM requirements.

The hitInfo argument allows the user to export the Hit predictions (FPM_Hit) for data based on the calculated FPM criteria as well as the associated FN/FP/TP/TN class.

The lockInfo argument allows the user to export information about what caused the model algorithm to lock for each chemical. Output options are: "FN" for exceeding the false negative limit (i.e., FN_crit), "FP" if the number of false positives was reduced to zero, "Max" if the empirical maximum concentration was exceeded, or Mix if more than one of the first three options occurred.

The following classification statistics are reported alongside the generated benchmarks: TP, FN, TN, and FP - the numbers of true positive, false negative, true negative, and false positive predictions pFN and pFP - proportions of false predictions (false No-hit and false Hit, respectively) sens - sensitivity; the probability of detecting a Hit spec - specificity; the probability of detecting a No-hit OR - overall reliability; the probability of making a correct prediction (Hit or No-hit) FM - Fowlkes-Mallows Index; geometric mean of sensitivity and the positive predictive rate MCC - Matthew's Correlation Coefficient; metric analogous to Pearson's coefficient, but instead defining correspondance between categorical predictions and reality (rather than for continuous data).

The second output of FPM is a metric called chemDensity. This is a measure of how much the percentile "floated" in the algorithm from the starting position up to the chemical's value at which it was locked in place. Values of chemDensity closer to 1 floated less and vice-versa. By floating less, this indicates that even small changes in the chemical concentration resulted in one of the acceptance criteria failing (as discussed above with regard to lockInfo). When comparing the chemDensity among chemicals, those with lower values might be viewed as having less of an influence on toxicity predictions and vice-versa. For those interested in understanding the relative importance of chemicals among benchmarks, we recommend using chemVI and considering the MADP and dOR outputs.

Value

list of 2 or 4 objects (depending on lockInfo):

1. Benchmarks and toxicity classification error statistics;
2. order in which benchmarks were locked in place;
3. reason for benchmarks being locked in place; and

4. `chemDensity` statistic

References

Avocet. 2003. Development of freshwater sediment quality values for use in Washington State. Phase II report: Development and recommendation of SQVs for freshwater sediments in Washington State. Publication No. 03-09-088. Prepared for Washington Department of Ecology. Avocet Consulting, Kenmore, WA. Ecology. 2011. Development of benthic SQVs for freshwater sediments in Washington, Oregon, and Idaho. Publication no. 11-09-054. Toxics Cleanup Program, Washington State Department of Ecology, Olympia, WA.

See Also

optimFPM, cvFPM, chemSig, chemSigSelect, chemVI

Examples

```
paramList = c("Cd", "Cu", "Fe", "Mn", "Ni", "Pb", "Zn")
FPM(h.tristate, paramList, ExcelMode = TRUE, warn = FALSE)
FPM(h.tristate, paramList, c(0.1, 0.2, 0.3))
```

`h.northport` *Northport Data,* Hyalella azteca *Toxicity*

Description

Example dataset

Format

A data.frame with 147 rows and 15 variables:

Sample sediment sample ID

Al aluminum concentration, mg/kg dry weight

As arsenic concentration, mg/kg dry weight

Cu copper concentration, mg/kg dry weight

Cd cadmium concentration, mg/kg dry weight

Cr chromium concentration, mg/kg dry weight

Fe iron concentration, mg/kg dry weight

Pb lead concentration, mg/kg dry weight

Hg mercury concentration, mg/kg dry weight

Ni nickel concentration, mg/kg dry weight

Zn zinc concentration, mg/kg dry weight

Organism organism abbreviation, `HA` = *Hyalella azteca*

Meas_Day measurement day for toxicity endpoint (post-initiation)

Endpoint toxicity endpoint

Hit logical; whether the sample was classified as toxic

Details

`h.northport` is a field-collected dataset containing sediment chemical concentrations for metals and toxicity classification data for the amphipod *Hyalella azteca*. Data are from a freshwater site in Washington State (Dowling and Roland 2019).

References

Dowling B, Roland J. 2019. Establishment of site-specific SMS metals cleanup objectives for contaminated sediments - Northport waterfront and nearshore state cleanup site. Washington State Department of Ecology.

`h.tristate` *Tristate Data,* Hyalella azteca *Toxicity*

Description

Example dataset

Format

A data frame with 43 rows and 13 variables:

Sample.ID sediment sample ID

Sample.type sediment sample type

Organism organism abbreviation, `HA` = *Hyalella azteca*

Day measurement day for toxicity endpoint (post-initiation)

Endpoint toxicity endpoint

Hit logical; whether we classified the sample as toxic (based on effect >25%)

Cd cadmium concentration, mg/kg dry weight

Cu copper concentration, mg/kg dry weight

Fe iron concentration, mg/kg dry weight

Mn manganese concentration, mg/kg dry weight

Ni nickel concentration, mg/kg dry weight

Pb lead concentration, mg/kg dry weight

Zn zinc concentration, mg/kg dry weight

Details

`h.trisate` is a field-collected dataset containing sediment chemical concentrations for metals and toxicity classification data for the amphipod *Hyalella azteca*. Data are from freshwater sites in Kansas, Missouri, and Oklahoma (Ingersoll et al 2008). The Hit variable was assigned by the authors of this package.

References

Ingersoll CG, MacDonald DD, Besser JM, Brumbaugh WG, Ivey CD, Kemble NE, Kunz JL, May TW, Wang N, Smorong DE. 2008. Sediment chemistry, toxicity, and bioaccumulation data report for the US Environmental Protection Agency - Department of the Interior sampling of metal-contaminated sediment in the Tri-state Mining District in Missouri, Oklahoma, and Kansas. US Geological Survey and MacDonald Environmental Sciences, Ltd., Columbia, MO.

`highNoise` *Simulated Floating Percentile Model Data, High Random Noise in Toxicity Prediction*

Description

Example dataset

Format

data.frame, containing 147 rows and 11 variables:

Al aluminum concentration, mg/kg dry weight

As arsenic concentration, mg/kg dry weight

Cu copper concentration, mg/kg dry weight

Cd cadmium concentration, mg/kg dry weight

Cr chromium concentration, mg/kg dry weight

Fe iron concentration, mg/kg dry weight

Pb lead concentration, mg/kg dry weight

Hg mercury concentration, mg/kg dry weight

Ni nickel concentration, mg/kg dry weight

Zn zinc concentration, mg/kg dry weight

Hit logical; whether the sample was classified as toxic

Details

`highNoise` provides a sample of simulated data that were developed to analyze variability and sensitivity of `FPM`. Simulated datasets (n=1000) were originally developed for the analyses; `highNoise` is a single realization of the simulation. Simulations were generated using the covariance matrix of `h.northport` sediment chemical concentrations and the `rmvnorm` function from the 'splus2R' package (Constantine and Hesterberg 2021). `Hit` values were generated with `toxCRM` using simulated concentrations of `Cr`, `Cu`, `Fe`, and `Zn` as inputs. The inputs to `toxCRM` were based on empirical toxicity ranges (minimum and maximum effect levels) as well as arbitrarily based on probable effect concentration (PEC) and threshold effect concentration (TEC) values from MacDonald et al (2000). Inflection points were set equal to PECs, and steepnesses were set as the ratio of the PEC to the TEC. Fe inputs were arbitrary chosen to reflect its low toxicity relative to Cr, Cu, and Zn (e.g., high inflection point and relatively low steepness). Random normal noise was added to the

highNoise datasets by inputting eSD values into toxCRM equal to 20% of the range of chemical concentrations (chemical-specific). The median toxicity output (for the 4 chemicals) was compared to a threshold of 75% with Hit == TRUE assigned to values below that level and Hit == FALSE assigned to values >=75%.

References

Constantine W, Hesterberg T. 2021. splus2R: Supplemental S-PLUS Functionality in R. Version 1.3-3 (online). Updated January 30. Available from: https://cran.r-project.org/web/packages/splus2R/index.html.
MacDonald DD, Ingersoll CG, Berger TA. 2000. Development and evaluation of consensus-based sediment quality guidelines for freshwater ecosystems. Arch Environ Contam Toxicol 39(5):20-31.

See Also

FPM, toxCRM, h.northport, perfect, lowNoise

---

lowNoise *Simulated Floating Percentile Model Data, Low Random Noise in Toxicity Prediction*

Description

Example dataset

Format

data.frame, containing 147 rows and 11 variables:

Al aluminum concentration, mg/kg dry weight

As arsenic concentration, mg/kg dry weight

Cu copper concentration, mg/kg dry weight

Cd cadmium concentration, mg/kg dry weight

Cr chromium concentration, mg/kg dry weight

Fe iron concentration, mg/kg dry weight

Pb lead concentration, mg/kg dry weight

Hg mercury concentration, mg/kg dry weight

Ni nickel concentration, mg/kg dry weight

Zn zinc concentration, mg/kg dry weight

Hit logical; whether the sample was classified as toxic

Details

lowNoise provides a sample of simulated data that were developed to analyze variability and sensitivity of FPM. Simulated datasets (n=1000) were originally developed for the analyses; lowNoise is a single realization of the simulation. Simulations were generated using the covariance matrix of h.northport sediment chemical concentrations and the rmvnorm function from the 'splus2R' package (Constantine and Hesterberg 2021). The Hit values were generated with toxCRM using simulated concentrations of Cr, Cu, Fe, and Zn as inputs. The inputs to toxCRM were based on empirical toxicity ranges (minimum and maximum effect levels) as well as arbitrarily based on probable effect concentration (PEC) and threshold effect concentration (TEC) values from MacDonald et al (2000). Inflection points were set equal to PECs, and steepnesses were set as the ratio of the PEC to the TEC. Fe inputs were arbitrary chosen to reflect its low toxicity relative to Cr, Cu, and Zn (e.g., high inflection point and relatively low steepness). Random normal noise was added to the lowNoise datasets by inputting eSD values into toxCRM equal to 10% of the range of chemical concentrations (chemical-specific). The median toxicity output (for the 4 chemicals) was compared to a threshold of 75% with Hit == TRUE assigned to values below that level and Hit == FALSE assigned to values >=75%.

References

Constantine W, Hesterberg T. 2021. splus2R: Supplemental S-PLUS Functionality in R. Version 1.3-3 (online). Updated January 30. Available from: https://cran.r-project.org/web/packages/splus2R/index.html
MacDonald DD, Ingersoll CG, Berger TA. 2000. Development and evaluation of consensus-based sediment quality guidelines for freshwater ecosystems. Arch Environ Contam Toxicol 39(5):20-31.

See Also

FPM, toxCRM, h.northport, perfect, highNoise

---

optimFPM *Optimization of Floating Percentile Model Parameters*

Description

Calculate parameter inputs that optimize benchmark performance

Usage

```
optimFPM(
  data,
  paramList,
  FN_crit = seq(0.1, 0.9, by = 0.05),
  alpha.test = seq(0.05, 0.5, by = 0.05),
  which = c(1, 2, 3, 4),
  simplify = TRUE,
  plot = TRUE,
  colors = heat.colors(10),
  colsteps = 100,
```

```
  ...
)
```

Arguments

data	data.frame containing, at a minimum, chemical concentrations as columns and a logical Hit column classifying toxicity
paramList	character vector of column names of chemical concentration variables in data
FN_crit	numeric vector over which to optimize false negative thresholds (default = seq(0.1, 0.9, by = 0.05))
alpha.test	numeric vector of type-I error rate values over which to optimize (default = seq(0.05, 0.5, by = 0.05))
which	numeric or character indicating which type of plot to generate (see Details; default = c(1, 2))
simplify	logical; whether to generate simplified output (default = TRUE)
plot	logical; whether to generate a plot to visualize the opimization results
colors	values recognizible as colors to be passed to colorRampPalette (via colorGradient) to generate a palette for plotting (default = heat.colors(10))
colsteps	integer; number of discrete steps to interpolate colors in colorGradient (default = 100)
...	additional argument passed to FPM, chemSig, chemSigSelect, and colorGradient

Details

optimFPM was designed to help optimize the predictive capacity of the benchmarks generated by FPM. The default input parameters to FPM (i.e., FN_crit = 0.2 and alpha.test = 0.05) are arbitrary, and optimization can help to objectively establish more accurate benchmarks. Graphical output from optimFPM can also help users to understand the relationship(s) between benchmark accuracy/error, FN_crit, and alpha.test.

Default inputs for FN_crit and alpha.test were selected to represent a reasonable range of values to test. Testing over both ranges will result in a two-way optimization, which can be computationally intensive. Alternatively, optimFPM can be run for one parameter at a time by specifying a single value for FN_crit or alpha.test. Note that inputting single values for both FN_crit and alpha.test will generate unhelpful results.

Several metrics are used for optimization:

1. Ratio of sensitivity/specificity ("sensSpecRatio"), calculated as the minimum of the two metrics divided by the maximum of the two. Therefore, this value will always be between 0 and 1, representing the balance between correct Hit==TRUE and Hit==FALSE predictions.
2. Overall reliability ("OR") (i.e., probability of correctly predicting Hit values)
3. Fowlkes-Mallows Index ("FM") - an average of metrics focusing on predicting Hit==TRUE
4. Matthew's Correlation Coefficient ("MCC") - a measure of the correspondence between the data and predictions analogous to a Pearson's correlation coefficient (but for binary data)

Graphical output will differ depending on whether or not a single value is input for `FN_crit` or `alpha.test`. Providing a single value for one of the two arguments will generate a line graph, whereas providing longer vectors (i.e., length > 1) of inputs for both arguments will generate dot matrix plots using `colors` to generate a color palette and `colsteps` to define the granularity of the color gradient with the palette. The order of `colors` will be plotted from more optimal to less optimal; for example, the default of `heat.colors(10)` will show optimal colors as red and less optimal colors as yellower. By default, multiple plots will be generated, however the `which` argument can control which plots are generated. Inputs to `which` are, by default, `c(1, 2, 3, 4)` for the metrics noted above, and flexible character inputs also can be used to a degree. Black squares indicate the optimal argument inputs; these values are also printed to the console and can be assigned to an object.

Value

data.frame of optimized `FN_crit` and/or `alpha.test` values

See Also

FPM, colorGradient, colorRampPalette

Examples

```
paramList = c("Cd", "Cu", "Fe", "Mn", "Ni", "Pb", "Zn")
FN_seq <- seq(0.1, 0.3, 0.05)
alpha_seq <- seq(0.05, 0.2, 0.05)
optimFPM(h.tristate, paramList, FN_seq, 0.05)
optimFPM(h.tristate, paramList, 0.2, alpha_seq)
optimFPM(h.tristate, paramList, FN_seq, alpha_seq, which=2)
```

---

perfect *Simulated Data for Floating Percentile Model, Perfect Toxicity Prediction*

Description

Example dataset

Format

data.frame containing 147 rows and 11 variables:

Al aluminum concentration, mg/kg dry weight

As arsenic concentration, mg/kg dry weight

Cu copper concentration, mg/kg dry weight

Cd cadmium concentration, mg/kg dry weight

Cr chromium concentration, mg/kg dry weight

Fe iron concentration, mg/kg dry weight

Pb lead concentration, mg/kg dry weight

Hg mercury concentration, mg/kg dry weight

Ni nickel concentration, mg/kg dry weight

Zn zinc concentration, mg/kg dry weight

Hit logical; whether the sample was classified as toxic

Details

`perfect` provides a sample of simulated data that were developed to analyze variability and sensitivity of `FPM`. Simulated datasets (n=1000) were originally developed for the analyses; `perfect` is a single example realization of the simulation. Simulations were generated using the covariance matrix of `h.northport` sediment chemical concentrations and the `rmvnorm` function from the `splus2R` package (Constantine and Hesterberg 2021). The `Hit` values were generated with `toxCRM` using simulated concentrations of `Cr`, `Cu`, `Fe`, and `Zn` as inputs. The inputs to `toxCRM` were based on empirical toxicity ranges (minimum and maximum effect levels) as well as arbitrarily based on probable effect concentration (PEC) and threshold effect concentration (TEC) values from MacDonald et al (2000). Inflection points were set equal to PECs, and steepnesses were set as the ratio of the PEC to the TEC. Fe inputs were arbitrary chosen to reflect its low toxicity relative to Cr, Cu, and Zn (e.g., high inflection point and relatively low steepness). The median toxicity output (for the 4 chemicals) was compared to a threshold of 75% with `Hit == TRUE` assigned to values below that level and `Hit == FALSE` assigned to values >=75%.

References

Constantine W, Hesterberg T. 2021. splus2R: Supplemental S-PLUS Functionality in R. Version 1.3-3 (online). Updated January 30. Available from: https://cran.r-project.org/web/packages/splus2R/index.html.
MacDonald DD, Ingersoll CG, Berger TA. 2000. Development and evaluation of consensus-based sediment quality guidelines for freshwater ecosystems. Arch Environ Contam Toxicol 39(5):20-31.

See Also

FPM, toxCRM, h.northport, lowNoise, highNoise

`plot.chemSigSelect` *Plot method for 'chemSigSelect'*

Description

The plot method for 'chemSigSelect' objects

Usage

```
## S3 method for class 'chemSigSelect'
plot(
  x,
  paramList = NULL,
```

```
  type = c("boxplot", "ecdf", "density"),
  logAxes = TRUE,
  ...
)
```

Arguments

x	chemSigSelect class object
paramList	character vector of column names of chemical concentration variables in data; values will be inherited from x if not specified
type	character vector indicating the type of plot to generate (see Details; default = c("boxplot", "ecdf", "density"))
logAxes	logical value; whether axes should be projected in log-space (default = TRUE)
...	additional arguments passed to plot. The user currently has limited capacity to change graphical parameters.

Details

plot.chemSigSelect is a class-specific plotting function that generates boxplots, empirical cumulative density functions, or kernel density plots comparing the Hit == TRUE and Hit == FALSE subsets for all chemicals in paramList. By default, all plot types are generated, but type can be changed to generate specific plot types. The available types are specified by "type = boxplot", "type = ecdf", and "type = density", though character inputs are flexible; for example, other reasonable versions of these types include "Boxplot" and "ECDF". Note that because boxplot, plot.ecdf, and plot.density functions use plotting arguments in different ways, supplying user-defined plotting arguments may cause unintended changes (or possibly errors or warnings) when plotting multiple types. For that reason, we recommend that the user select a specific plot type (using the type argument) before adjusting plot arguments (via ...) that apply to that type. Moreover, many of the plotting arguments are specified by plot.chemSigSelect and, therefore, cannot be adjusted by the user. These include but are not limited to parameters like col and lwd.

Value

base graphics

See Also

chemSigSelect, boxplot, plot.ecdf, plot.density

Examples

```
x <- chemSigSelect(data = h.tristate,
    paramList = c("Cd", "Cu", "Fe", "Mn", "Ni", "Pb", "Zn"))
plot(x, type = "boxplot")
```

predict.FPM *Predict Toxicity Using the Floating Percentile Model*

Description

Use new sediment chemistry data to generate FPM predictions

Usage

```
## S3 method for class 'FPM'
predict(object, newdata, ...)
```

Arguments

object	FPM class object, created using FPM (and extracted from the resulting list using .$FPM, where '.' is the name of the object or FPM call)
newdata	character vector of column names of chemical concentration variables in data
...	further arguments passed to or from other methods

Details

There are two things to keep in mind when using predict for 'FPM' objects. Firstly, when FPM is run, the output object is a list; one of the objects (called "FPM") is the FPM class object that can be used to predict Hits.

Secondly, unlike other default "predict" methods, predict.FPM is used strictly to predict toxicity that is not included in the original dataset used to generate fpm. To predict Hit results for the original data, set hitInfo == TRUE when running the FPM function. Note that predicting toxicity for the original dataset may be arbitrary/unnecessary, as the Hit results for the original dataset must be known.

Note that, in order to run predict.FPM, the newdata argument must be supplied a data.frame that includes at least one sample with all of the chemical columns contained in fpm. Column headers must match exactly.

Value

logical

Examples

```
# create FPM object with chemical headings that overlap
overlap <- intersect(names(h.northport)[1:10],
            names(h.tristate)[7:13])
fpm_object = FPM(h.northport, overlap)
# run predict on the 'FPM' list item to estimate Hits
predict(object = fpm_object$FPM, newdata = h.tristate)
```

RFPM *Floating Percentile Model*

Description

Floating Percentile Model and supporting functions to inform and improve sediment benchmark development

Usage

```
RFPM()
```

Details

'RFPM' is an open-source implementation of the floating percentile model (FPM), which was originally developed by Avocet (2003) using Visual Basic for Applications and distributed to US Pacific Northwest regulatory agencies as a Microsoft Excel-based tool. 'RFPM' was developed independently by Claire Detering and Brian Church with support from John Toll and others at Windward Environmental LLC.

The purpose of the FPM is to generate aquatic toxicity-based sediment quality benchmarks for management of contaminated freshwater sediment sites. These benchmarks are intended to act as classification thresholds, meaning that an exceedance of benchmarks would imply that toxicity (as categorically defined) is likely in the sediment sample. The FPM has been used at sites in the US Pacific Northwest for many years, particularly after being published by the Washington State Department of Ecology in 2011.

The primary function in 'RFPM' is `FPM`, which runs the FPM algorithm on a data.frame object that includes concentrations of chemicals in sediment as well as a logical toxicity classification column called "Hit". Example datasets are provided. The output of `FPM` includes a set of sediment quality benchmarks for chemicals with significantly higher concentrations when `Hit == TRUE` than when `Hit == FALSE`. Plots comparing the `Hit == TRUE` and `Hit == FALSE` data can also be generated as a diagnostic tool. Supplemental functions (e.g., `optimFPM`) can help to optimize `FPM` inputs (resulting in more accurate benchmarks) or evaluate the relative importance of each chemical among the FPM benchmarks (i.e., `chemVI`).

For 'RFPM', the `FPM` algorithm has been changed from the original Avocet (2003) model. Key changes are as follows:

1. A decision tree is implemented to select statistically appropriate hypothesis tests for chemical selection. This can be overridden if the original Excel-based tool method is desired; see `?chemSig` for details. The chemical selection and FPM algorithm have been integrated into `FPM`, though chemical selection can still be run separately, if desired.
2. The iterative looping of the FPM algorithm over multiple false negative limits was not included in `FPM`. We find this functionality of the Excel-based tool to be confusing and unnecessary. Instead, we believe the results generated for different false negative limits should be independently generated rather than dependent on prior model runs. Thus, `FPM` allows for multiple false negative limits to be input in a seqeuential but independent manner, resulting in a data.frame of benchmarks with one row per false negative limit. In R terminology, `FPM` has been vectorized.

3. An optimization function called `optimFPM` was developed to optimize the overall reliability of FPM sediment quality benchmarks. Different optimization metrics are provided, and a weight of evidence should be considered when selecting inputs.
4. A function was developed to quickly calculate chemical "variable importance" statistics using `chemVI`. These statistics can inform the user about the influence of specific chemicals over the set of FPM benchmarks and, for example, whether certain benchmarks can be ignored without a significant loss of predictive ability.

Value

bibentry

References

Avocet. 2003. Development of freshwater sediment quality values for use in Washington State. Phase II report: Development and recommendation of SQVs for freshwater sediments in Washington State. Publication No. 03-09-088. Prepared for Washington Department of Ecology. Avocet Consulting, Kenmore, WA. Ecology. 2011. Development of benthic SQVs for freshwater sediments in Washington, Oregon, and Idaho. Publication no. 11-09-054. Toxics Cleanup Program, Washington State Department of Ecology, Olympia, WA.

`toxCRM` *Concentration-Response Model, 4-parameter log-logistic curve*

Description

Generate dummy toxicity data with strong concentration-response

Usage

```
toxCRM(x, max = 1, min = 0, steep, mid, eMean = 0, eSD = 0, seed = NULL)
```

Arguments

`x`	numeric vector of chemical concentrations (univariate)
`max`	numeric value, asymptotic maximum of model (default = `1`)
`min`	numeric value, asymptotic minimum of model (default = `0`)
`steep`	numeric value, steepness factor for model slope
`mid`	numeric value, inflection point concentration for CRM slope
`eMean`	numeric value, mean of random-normal error to add to dummy toxicity data (default = `0`)
`eSD`	numeric value, standard deviation of random-normal error to add to dummy toxicity data (default = `0`)
`seed`	numeric value, random seed to set for repeatable random error generating (default = `NULL`, i.e., no seed)

Details

toxCRM generates dummy toxicity data representing an idealized condition. This approach was used by the authors to simulate data and test FPM sensitivity and baseline variability. The user must specify all coefficients of the model (though default min/max values are provided) and may add some amount of random error, if desired. Random error adds noise to the dummy toxicity daata, increasing the uncertainty of toxicity predictions using floating percentile model benchmarks. The perfect, lowNoise, and highNoise datasets included in RFPM were all generated using toxCRM with increasing levels of eSD.

Value

numeric vector

See Also

perfect, lowNoise, highNoise

Examples

```
concentration = h.northport$Cu
toxVals <- toxCRM(concentration, 1, 0, 0.5,
    median(concentration), 0, 0)
plot(concentration, toxVals, log="x", main="Perfect estimate")

toxVals_withNoise <- toxCRM(x = concentration,
    1, 0, 0.5, median(concentration), 0, 0.1, seed = 1)
plot(concentration, toxVals_withNoise, log="x", main="Noisy estimate")
```

Index