Package 'xrfcal'

October 17, 2022

Title Calibrating XRF scanners elemental counts to reference elemental concentrations

Version 0.8.0.1

Maintainer Shayan Kabiri <shayankabiri6@gmail.com>

Description Methods for converting elemental counts acquired by XRF scanners to reference elemental concentrations acquired by conventional methods such as ICP (Kabiri et al. (2022) <doi>, Weltje and Tjalingii (2008) <doi:10.1016/j.epsl.2008.07.054>).

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

Depends R (>= 2.10)

LazyData true

Imports lmodel2,
zCompositions,
agrmt,
stats,
caret,
Cubist,
randomForest,
tibble,
glmnet

Author Shayan Kabiri [cre, aut]
(<https://orcid.org/0000-0003-3231-5843>)
Nicholas M. Holden [aut]
(<https://orcid.org/0000-0002-0452-4632>),
Rory Flood [aut]
Jonathan N. Turner [aut]
(<https://orcid.org/0000-0001-6596-1038>),
Sharon M. O'Rourke [aut]
(<https://orcid.org/0000-0002-4565-1689>)

R topics documented:

`alr` *Simplex matrix to additive logratio matrix*

Description

This function takes a composition matrix X with D elements in simplex space, closes it to 1 and converts it to a matrix in additive logratio space with D-1 elements with column d as the denominator.

Usage

```
alr(X, d)
```

Arguments

`X`	A matrix of compositions in simplex space with D columns. Doesn't need to be closed.
`d`	Index of the column to be used as denominator

Value

A matrix of compositions in alr space with D-1 columns

References

Aitchison, J. (1982). The Statistical Analysis of Compositional Data. Journal of the Royal Statistical Society. Series B (Methodological), 44(2), 139-177. http://www.jstor.org/stable/2345821

Examples

```
#Transform elemetal counts to alr with calcium as denominator
Xalr <- alr(xrf$X,6)
```

`averager` *Reduce the dataset to a representative dataset*

Description

This function takes counts matrix X and concentrations matrix Y and reduce it to a smaller representative dataset by taking the average, first and third quantile of counts for each unique concentration. This is useful when a unique concentration is associated with many count instances.

Usage

```
averager(X, Y)
```

Arguments

X	A matrix of XRF counts. Zeros should be replaced beforehand.
Y	A matirx of reference concentrations.

Value

A list with two objects, reduced X, `obj[[1]]`, reduced Y, `obj[[2]]`

Examples

```
#Replace zeros in X
dl <- apply(xrf$X, 2, agrmt::minnz)
nzX <- zCompositions::multRepl(xrf$X, dl = dl, label = 0)

#Reduce dataset
xrfRed <- averager(nzX,xrf$Y)
Xred <- xrfRed[[1]]
Yred <- xrfRed[[2]]
```

---

invalr *Additive logratio matrix to simplex matrix*

Description

This function takes a composition matrix X with D-1 elements in additive logratio space converts it back to a matrix in simplex space with D elements with column d as the denominator of alr transformation.

Usage

```
invalr(X, d)
```

Arguments

X	A matrix of compositions in alr space with D-1 columns
d	Index of the column that was used as denominator for original alr transformation

Value

A matrix of compositions in simplex space with D columns

References

Aitchison, J. (1982). The Statistical Analysis of Compositional Data. Journal of the Royal Statistical Society. Series B (Methodological), 44(2), 139-177. http://www.jstor.org/stable/2345821

Examples

```
#Transform elemetal counts to alr with calcium as denominator and transform it back to simplex.
Xalr <- alr(xrf$X,6)
Xsimplex <- invalr(Xalr,6)
```

`pred` *Predict new concentrations based on a trained calibration model*

Description

This function takes a trained calibration model and a matrix of elemental counts and predicts elemental concentrations for the matrix.

Usage

```
pred(mdl, newX)
```

Arguments

`mdl`	A model object created by xrfcal::calib().
`newX`	An elemental counts matrix in simplex space to make concentration predictions on.

Details

The procedure for predicting on new data in `pred` function is as followed:

1. take in the output of calibration function `xrfcal`.
2. if there are zeroes in `newX`, replace zeroes with `multRepl` from `zCompositions`.
3. for each element to be predicted, transform newX to alr with best denominator for that element.
4. predict all elements for each best deonimnator and transform back to simplex but only get the element of interest as output.
5. aggregate all element predictions and close them to 1.
6. export predicted concentrations.

Value

A matrix of predicted elemental concentrations in simplex space.

References

Weltje, G. J., & Tjallingii, R. (2008). Calibration of XRF core scanners for quantitative geochemical logging of sediment cores: Theory and application. Earth and Planetary Science Letters, 274(3), 423-438. https://doi.org/10.1016/j.epsl.2008.07.054

Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324

Quinlan, J.R. (1993). Combining Instance-Based and Model-Based Learning. ICML.

Aitchison, J. (1982). The Statistical Analysis of Compositional Data. Journal of the Royal Statistical Society. Series B (Methodological), 44(2), 139-177. http://www.jstor.org/stable/2345821

Examples

```
#Remove count columns with more than 20 percent zeroes

zeroes <- apply(xrf$X,2,function(x){return(length(which(x==0))/length(x))})
ind <- which(zeroes>0.2)
if (length(ind)>0)
  X <- xrf$X[,-ind]

#Replace zeros in X
dl <- apply(X, 2, agrmt::minnz)
nzX <- zCompositions::multRepl(X, dl = dl, label = 0)

#Reduce dataset
xrfRed <- averager(nzX,xrf$Y)
Xred <- xrfRed[[1]]
Yred <- xrfRed[[2]]

#Split dataset into train and test sets
set.seed(123)
ind <- sample.int(ceiling(dim(Xred)[1]/3))
testX <- Xred[ind,]
testY <- Yred[ind,]
trainX <- Xred[-ind,]
trainY <- Yred[-ind,]

#Build a calibration model based on training data
mdl <- xrfcal(trainX,trainY, method = "RF", reduce=FALSE)

#Predict for testing data
Yhat <- pred(mdl,testX)

#Calculate R2 and RMSE for test set
res <- caret::postResample(Yhat$K,testY$K)

#Plot predicted potassium concentrations vs actual concentrations for test set
plot(y=Yhat$K,x=testY$K,ylim = c(0,max(cbind(Yhat$K,testY$K))),
   xlim = c(0,max(cbind(Yhat$K,testY$K))),ylab="Calibrated Concentrations for K (mass fraction)",
    xlab ="Actual Concentrations for K (mass fraction) ")
lines(-1:1,-1:1)
text(0.2*max(cbind(Yhat$K,testY$K)),0.8*max(cbind(Yhat$K,testY$K)),
    paste("R2 =",signif(res[2],2),"\n", "RMSE =",signif(res[1],2)))
```

---

xrf *Example XRF counts and reference concentration data*

Description

An example of XRF counts and reference concentrations to be used. `xrf$X` is the elemental counts measured by XRF core scanner and `xrf$Y` are reference elemental concentrations measured by ICP.

Usage

```
xrf
```

Format

An object of class `list` of length 2.

References

ADD REFERENCE LATER

`xrfcal` *Calibrate XRF counts to element concentrations*

Description

This function takes an XRF counts matrix and reference concentrations matrix and returns a model object with R2s and RMSEs for each element with various denominators for alr and best R2, RMSE, and denominator for each element. Four methods are available to use, with logratio calibration equation (LRCE) using a single input for regression for each element, and elastic net (ENET), cubist (Cubist) and random forest (RF), taking multiple inputs. Random forest and cubist are more computationally expensive to train but generally more powerful.

Usage

```
xrfcal(
  X,
  Y,
  method = "RF",
  dl = 0,
  oalr = FALSE,
  reduce = TRUE,
  thresh = 0.2,
  mtry = ncol(X) - 1,
  ntree = 20
)
```

Arguments

`X`	Elemental counts matrix in simplex space. Doesn't have to be closed.
`Y`	Reference concentrations matrix in simplex space. Doesn't have to be closed.
`method`	Regression method. Either "LRCE", "ENET", "Cubist" or "RF".
`dl`	Detection limits to use for zero replacement algorithm. A vector with D elements. If not provided will be estimated.
`oalr`	A boolean parameter. If true the model results will be in alr space.
`reduce`	A boolean parameter. If true X and Y will be reduced to a representative dataset.
`thresh`	Threshold of fraction of zeroes in an element's counts to drop that element for model training
`mtry`	The number of variables for random forest to sample randomly at each split.
`ntree`	Number of estimators for random forest.

Details

In order to linearlize and reduce diversion from normal distribution and to make methods applicable to linear geometry, also applicable to compositional geometry as well, it is argued based on work of Aitchison (1982) that elemental counts and concentrations should be transformed to an additive logratio space. In the current package in addition to simple linear regression between each two elements in additive logratio space (LRCE), multiple regressions with cubist, random forest and multiple linear regression are also implemented. In addition to this for each element all possible denominators for alr transformation are considered and the best denominator is chosen for prediction purposes.

The procedure for developing the model is as followed:

1. drop elements with xrf zero counts below the threshold.
2. if "LRCE" is chosen as regression method, only use the mutual elements in X and Y. Otherwise, use all.
3. if there are zeroes in X, use `multRepl` from `zCompositions` to replace zeroes. If detection limits are not provided estimate them as the minimum counts detected by XRF scanner for each element.
4. if `reduce` is `TRUE`, reduce dataset to a representative dataset by taking mean, first and third quantile. of counts for each unique Y and aggregate them to a new dataset.
5. split data in three folds.
6. for each fold, transform training and test set to alr space for each element as denominator.
7. for each element as denominator train the model, make predictions on test sets, transform back to simplex, and record average RMSE and R2 over three folds.
8. export results for investigation and parameters for the prediction function `pred`.

Value

A model object with attributes R2, RMSE, BestR2, BestRMSE and BestDenom. Rest of attributes are used for prediction function.

References

Weltje, G. J., & Tjallingii, R. (2008). Calibration of XRF core scanners for quantitative geochemical logging of sediment cores: Theory and application. Earth and Planetary Science Letters, 274(3), 423-438. https://doi.org/10.1016/j.epsl.2008.07.054

Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324

Quinlan, J.R. (1993). Combining Instance-Based and Model-Based Learning. ICML.

Aitchison, J. (1982). The Statistical Analysis of Compositional Data. Journal of the Royal Statistical Society. Series B (Methodological), 44(2), 139-177. http://www.jstor.org/stable/2345821

Examples

```
#Using the example data in the package build a calibration model
mdl <- xrfcal(xrf$X,xrf$Y, method = "LRCE")

#Plot R2s for different denominators for potassium
barplot(mdl$R2[which(rownames(mdl$R2)=="K"),],names.arg = colnames(mdl$R2),
    xlab="Denominator",ylab=paste("Crossvalidated R² for potassium"))

#Print out best R2 and best denominator for each element
print(rbind(mdl$BestR2,mdl$BestDenom))
```

Index