Package ‘batchtma’

October 12, 2022

Title Batch Effect Adjustments
Version 0.1.6
Description Different adjustment methods for batch effects in biomarker data, such as from tissue microarrays. Some methods attempt to retain differences between batches that may be due to between-batch differences in "biological" factors that influence biomarker values.
License GPL-3
Encoding UTF-8
RoxygenNote 7.1.1
biocViews
Imports broom (>= 0.7.0), dplyr (>= 1.0.0), geepack, ggplot2, limma, nnet, purrr (>= 0.3.0), quantreg, rlang (>= 0.4.0), stringr, tibble, tidyr (>= 1.1.0), magrittr
Suggests knitr, markdown, tidyverse
VignetteBuilder knitr
NeedsCompilation no
Author Konrad Stopsack [aut, cre] (<https://orcid.org/0000-0002-0722-1311>), Travis Gerke [aut] (<https://orcid.org/0000-0002-9500-8907>)
Maintainer Konrad Stopsack <stopsack@post.harvard.edu>
Repository CRAN
Date/Publication 2021-12-06 08:10:02 UTC

R topics documented:

  adjust_batch ................................................................. 2
  batchtma ................................................................. 5
  diagnose_models ......................................................... 5
  plot_batch .............................................................. 7

Index 10
adjust_batch generates biomarker levels for the variable(s) markers in the dataset data that are corrected (adjusted) for batch effects, i.e. differential measurement error between levels of batch.

**Usage**

```r
adjust_batch(
  data,
  markers,
  batch,
  method = c("simple", "standardize", "ipw", "quantreg", "quantnorm"),
  confounders = NULL,
  suffix = "_adjX",
  ipw_truncate = c(0.025, 0.975),
  quantreg_tau = c(0.25, 0.75),
  quantreg_method = "fn"
)
```

**Arguments**

- `data` : Data set
- `markers` : Variable name(s) to batch-adjust. Select multiple variables with tidy evaluation, e.g. `markers = starts_with("biomarker")`
- `batch` : Categorical variable indicating batch.
- `method` : Method for batch effect correction:
  - **simple** : Simple means per batch will be subtracted. No adjustment for confounders.
  - **standardize** : Means per batch after standardization for confounders in linear models will be subtracted. If no confounders are supplied, method = simple is equivalent and will be used.
  - **ipw** : Means per batch after inverse-probability weighting for assignment to a specific batch in multinomial models, conditional on confounders, will be subtracted. Stabilized weights are used, truncated at quantiles defined by the ipw_truncate parameters. If no confounders are supplied, method = simple is equivalent and will be used.
  - **quantreg** : Lower quantiles (default: 25th percentile) and ranges between a lower and an upper quantile (default: 75th percentile) will be unified between batches, allowing for differences in both parameters due to confounders. Set the two quantiles using the quantregTau parameters.
  - **quantnorm** : Quantile normalization between batches. No adjustment for confounders.
**confounders**  Optional: Confounders, i.e. determinants of biomarker levels that differ between batches. Only used if method = standardize, method = ipw, or method = quantreg, i.e. methods that attempt to retain some of these "true" between-batch differences. Select multiple confounders with tidy evaluation, e.g., confounders = c(age, age_squared, sex).

**suffix**  Optional: What string to append to variable names after batch adjustment. Defaults to "_adjX", with X depending on method:
- _adj2 from method = simple
- _adj3 from method = standardize
- _adj4 from method = ipw
- _adj5 from method = quantreg
- _adj6 from method = quantnorm

**ipw_truncate**  Optional and used for method = ipw only: Lower and upper quantiles for truncation of stabilized weights. Defaults to c(0.025, 0.975).

**quantreg_tau**  Optional and used for method = quantreg only: Quantiles to scale. Defaults to c(0.25, 0.75).

**quantreg_method**  Optional and used for method = quantreg only: Algorithmic method to fit quantile regression. Defaults to "fn". See parameter method of rq.

**Details**

If no true differences between batches are expected, because samples have been randomized to batches, then a method that returns adjusted values with equal means (method = simple) or with equal rank values (method = quantnorm) for all batches is appropriate.

If the distribution of determinants of biomarker values (confounders) differs between batches, then a method that retains these "true" differences between batches while adjusting for batch effects may be appropriate: method = standardize and method = ipw address means; method = quantreg addresses lower values and dynamic range separately.

Which method to choose depends on the properties of batch effects (affecting means or also variance?) and the presence and strength of confounding. For the two mean-only confounder-adjusted methods, the choice may depend on whether the confounder–batch association (method = ipw) or the confounder–biomarker association (method = standardize) can be modeled better. Generally, if batch effects are present, any adjustment method tends to perform better than no adjustment in reducing bias and increasing between-study reproducibility. See references.

All adjustment approaches except method = quantnorm are based on linear models. It is recommended that variables for markers and confounders first be transformed as necessary (e.g., log transformations or splines). Scaling or mean centering are not necessary, and adjusted values are returned on the original scale. Parameters markers, batch, and confounders support tidy evaluation.

Observations with missing values for the markers and confounders will be ignored in the estimation of adjustment parameters, as are empty batches. Batch effect-adjusted values for observations with existing marker values but missing confounders are based on adjustment parameters derived from the other observations in a batch with non-missing confounders.
Value

The data dataset with batch effect-adjusted variable(s) added at the end. Model diagnostics, using the attribute `.batchtma` of this dataset, are available via the `diagnose_models` function.

Author(s)

Konrad H. Stopsack

References


See Also

https://stopsack.github.io/batchtma/

Examples

```r
# Data frame with two batches
# Batch 2 has higher values of biomarker and confounder
data <- data.frame(
  tma = rep(1:2, times = 10),
  biomarker = rep(1:2, times = 10) +
    runif(max = 5, n = 20),
  confounder = rep(0:1, times = 10) +
    runif(max = 10, n = 20)
)

# Adjust for batch effects
# Using simple means, ignoring the confounder:
adjust_batch(
  data = df,
  markers = biomarker,
  batch = tma,
  method = simple
)
# Returns data set with new variable "biomarker_adj2"

# Use quantile regression, include the confounder,
# change suffix of returned variable:
adjust_batch(
    data = df,
    markers = biomarker,
    batch = tma,
    method = quantreg,
    confounders = confounder,
    suffix = "_batchadjusted"
)
# Returns data set with new variable "biomarker_batchadjusted"

diagnose_models: Model diagnostics after batch adjustment

Description
The goal of the batchma is to provide functions for batch effect-adjusting biomarker data. It implements different methods that address batch effects while retaining differences between batches that may be due to “true” underlying differences in factors that drive biomarker values (confounders).

Functions
adjust_batch: Adjust for batch effects
diagnose_models: Model diagnostics after batch adjustment
plot_batch: Plot biomarkers by batch

References

See Also
https://stopsack.github.io/batchma/

diagnose_models: Model diagnostics after batch adjustment

Description
After adjust_batch has performed adjustment for batch effects, diagnose_models provides an overview of parameters and adjustment models. Information is only available about the most recent run of adjust_batch on a dataset.
Usage

diagnose_models(data)

Arguments

data  Batch-adjusted dataset (in which adjust_batch has stored information on batch adjustment in the attribute .batchtma)

Value

List:
- adjust_method Method used for batch adjustment (see adjust_batch).
- markers Variables of biomarkers for adjustment
- suffix Suffix appended to variable names
- batchvar Variable indicating batch
- confounders Confounders, i.e. determinants of biomarker levels that differ between batches. Returned only if used by the model.
- adjust_parameters Tibble of parameters used to obtain adjust biomarker levels. Parameters differ between methods:
  - simple, standardize, and ipw: Estimated adjustment parameters are a tibble with one batchmean per marker and .batchvar.
  - quantreg returns a tibble with numerous values per marker and .batchvar: unadjusted (un_...) and adjusted (ad_...) estimates of the lower (..._lo) and upper quantile (..._hi) and interquantile range (..._iq), plus the lower (all_lo) and upper quantiles (all_hi) across all batches.
  - quantnorm does not explicitly estimate parameters.
- model_fits List of model fit objects, one per biomarker. Models differ between methods:
  - standardize: Linear regression model for the biomarker with .batchvar and confounders as predictors, from which marginal predictions of batch means for each batch are obtained.
  - ipw: Logistic (2 batches) or multinomial models for assignment to a specific batch with .batchvar as the response and confounders as the predictors, used to generate stabilized inverse-probability weights that are then used in a linear regression model to estimate marginally standardized batch means.
  - quantreg: Quantile regression with the marker as the response variable and .batchvar and confounders as predictors.
  - simple and quantnorm do not fit any regression models.

Examples

# Data frame with two batches
# Batch 2 has higher values of biomarker and confounder
df <- data.frame(
  tma = rep(1:2, times = 10),
  biomarker = rep(1:2, times = 10) +
  confounder = rep(1:2, times = 10) +
  # Rest of the data frame
To provide a simple visualization of potential batch effects, \texttt{plot_batch} generates a Tukey box plot overlaid by a jittered dot plot, inspired by the Stata plugin \texttt{stripplot}.

Boxes span from the 1st to the 3rd quartile; thick lines indicate medians; whiskers span up to 1.5 times the interquartile range; and asterisks indicate means.

\textbf{Usage}

\begin{verbatim}
plot_batch(
data, marker, batch, color = NULL, maxlevels = 15, title = NULL, ...
)
\end{verbatim}

\textbf{Arguments}

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>Dataset.</td>
</tr>
</tbody>
</table>

\textbf{Description}

To provide a simple visualization of potential batch effects, \texttt{plot_batch} generates a Tukey box plot overlaid by a jittered dot plot, inspired by the Stata plugin \texttt{stripplot}.

Boxes span from the 1st to the 3rd quartile; thick lines indicate medians; whiskers span up to 1.5 times the interquartile range; and asterisks indicate means.
marker  Variable indicating the biomarker.
batch  Variable indicating the batch.
color  Optional: third variable to use for symbol color and shape. For example, color can be used to show differences in a confounder.
maxlevels  Optional: Maximum number of levels for color parameter to accept as a discrete variable, rather than a continuous variable. Defaults to 15.
title  Optional: character string that specifies plot title
...  Optional: Passed on to ggplot.

Value
ggplot2 object, which can be further modified using standard ggplot2 functions. See examples.

References

See Also
More powerful visualizations of batch effects exist in the BatchQC package:

Examples
# Define example data
df <- data.frame(
  tma = rep(1:2, times = 10),
  biomarker = rep(1:2, times = 10) + 
    runif(max = 5, n = 20),
  confounder = rep(0:1, times = 10) + 
    runif(max = 10, n = 20)
)

# Visualize batch effects:
plot_batch(
  data = df,
  marker = biomarker,
  batch = tma,
  color = confounder
)

# Label y-axis, changing graph like other ggplots:
plot_batch(
  data = df,
  marker = biomarker,
plot_batch

batch = tma,
    color = confounder
) +
ggplot2::labs(y = "Biomarker (variable 'noisy')")
Index

adjust_batch, 2, 5, 6
batchtma, 5
diagnose_models, 4, 5, 5
ggplot, 8
log, 3
plot_batch, 5, 7
rq, 3
splines, 3