Introduction to *metadeconfoundR*

Till Birkner*[1]*, Sofia K. Forslund

[1] tillb@online.de

Edited: November 06, 2019; Compiled: November 21, 2019

Contents

1 Introduction

metadeconfoundR was developed to conduct confounder-aware biomarker analysis of large scale multi-omics datasets in parallel. This analysis is a two step process.

First, significant associations between single omics features (like gut microbial OTUs) and metadata (like drug administration) are identified (fig1, left). Based on the data type of the respective metadata, either `wilcox.test()` (for binary), `cor.test()` (for continuous numerical) or `kruskal.test()` (for neither numerical nor binary) is used. All three tests are rank-based to minimize assumptions about data distribution. In addition to collecting p-values for all computed tests, effect size is measured if possible. In case of binary data, Cliff's Delta is computed by an algorithm based on the `orddom::orddom()` function. For continuous data the "estimate" component from `cor.test()` is used. Since there is no effect size for categorical data with more than 2 levels, no value is reported here. It is recommended to introduce binary pseudo-variables for each level of the categorical metadata to circumvent this drawback.

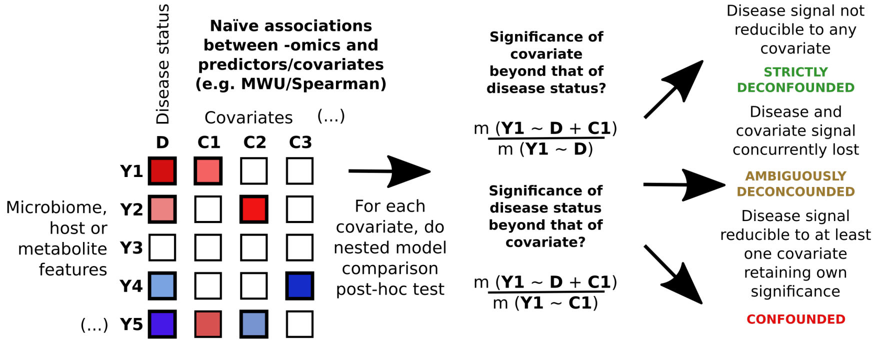
Disease status
Naïve associations between -omics and predictors/covariates (e.g. MWU/Spearman)
Covariates (...)
D C1 C2 C3
Y1
Y2
Y3
Y4
Y5
Microbiome, host or metabolite features
(...)
For each covariate, do nested model comparison post-hoc test
Significance of covariate beyond that of disease status?
m (Y1 ~ D + C1) / m (Y1 ~ D)
Significance of disease status beyond that of covariate?
m (Y1 ~ D + C1) / m (Y1 ~ C1)
Disease signal not reducible to any covariate
STRICTLY DECONFOUNDED
Disease and covariate signal concurrently lost
AMBIGUOUSLY DECONCOUNDED
Disease signal reducible to at least one covariate retaining own significance
CONFOUNDED

Figure 1: Overview of main statistical approach used to determine confounding status
(left) Each omics feature (Y) is independently tested for association to any of the predictors/covariates (D/C); Used test depends on data type of covariate. (center) For each identified feature (Y1) ↔ covariate (D/C) pair a set of linear models (either including an additional covariate or not) is fitted and a likelihood ratio is computed. (right) Based on likelihood ratio a status for the feature (Y1) ↔ covariate (D/C) pair is reported. Figure kindly provided by Sofia Forslund.

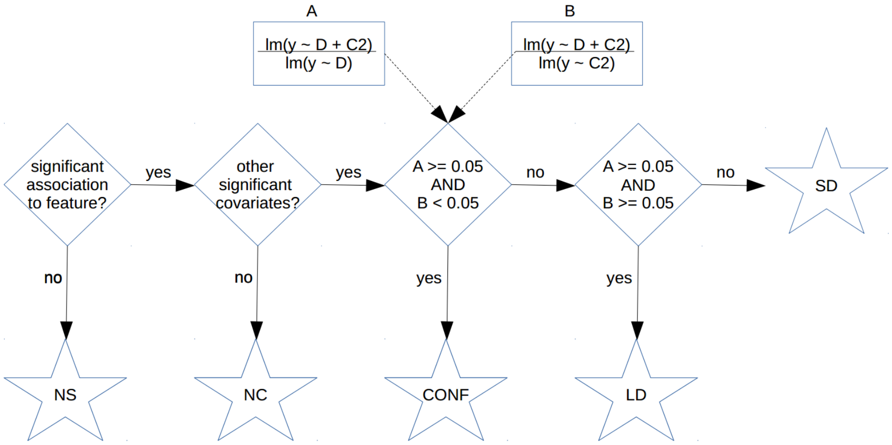
A
lm(y ~ D + C2) / lm(y ~ D)
B
lm(y ~ D + C2) / lm(y ~ C2)
significant association to feature?
yes
no
other significant covariates?
yes
no
A >= 0.05 AND B < 0.05
no
yes
A >= 0.05 AND B >= 0.05
no
yes
SD
NS
NC
CONF
LD

Figure 2: Status labeling process in more detail
For each feature ↔ covariate combination these steps are done. A and B are the linear model likelihood ratios. Should there be more than one "other significant covariate" (C2), linear model likelihood ratio comparison has to be repeated for every single one of them. Whenever CONF is reached, the name of C2 is returned as label. Only when SD is reached for all C2, this will be returned as label. (y = feature, D = current covariate, C2 = other significant covariates, NS = not significant, NC = no covariates (i.e. trivially deconfounded), CONF = confounded, LD = laxly deconfounded, SD = strictly deconfounded)

In the second step, all hits are checked for confounding effects (fig 1, center and right) and a status is reported for each feature ↔ metadata combination (fig 2). A “hit" here is defined as a feature ↔ metadata association with small enough fdr-corrected p-value and big enough effect size. Thresholds for both parameters can be set via `QCutoff` and `DCutoff` when starting the analysis. Since confounding of signal can only happen with more than

one different metadata associated to a certain feature, all features with only one significant metadata are trivially deconfounded and get status “No Covariates (NC)".
The actual confounder detection is done by a set of two likelihood ratio tests of nested linear models: For each feature with multiple covariates

2 Quick start

Install via tar.gz file: `install.packages("metadeconfoundR_0.1.1.tar.gz", repos = NULL)`

3 Use cases

```
library(metadeconfoundR)

data(reduced_feature)
data(metaMatMetformin)

example_output <- MetaDeconfound(featureMat = reduced_feature,
        metaMat = metaMatMetformin,
        nnodes = 1)

knitr::kable(example_output$Ds[1:3, 1:5], "latex", digits = 2, booktabs = TRUE)
```

	Status	Dataset	Metformin	continuous_dummy	altered_dummy
MS0001	-0.14	Inf	-0.16	0.01	0.08
MS0006	0.20	Inf	0.13	-0.02	0.13
MS0007	0.24	Inf	0.30	0.02	-0.02

Table 1: Summary of status assignments for each metadata variable

Status	Dataset	Metformin	continuous_dummy	altered_dummy
altered_dummy: 1	NC:11	Dataset: 1	NS:50	Dataset:18
Dataset : 1	NS: 2	LD : 2	NA	NS :29
LD : 3	SD:37	NS :32	NA	SD : 3
Metformin : 1	NA	SD :12	NA	NA
NC : 1	NA	Status : 3	NA	NA
NS :26	NA	NA	NA	NA
SD :17	NA	NA	NA	NA

Table 2: Compostition of metadata “Dataset"

CHN	MHD	SWE
256	352	145

Cliff’s Delta for categorical variables with more than two levels can’t be computed and gets set to `Inf`. By splitting up these variables into binary pseudo-variables will circumvent this. As can be seen in table 1, status assignment is not affected by missing Cliff’s Delta.

```
example_output2 <- MetaDeconfound(featureMat = reduced_feature,
        metaMat = metaMatMetformin,
        nnodes = 1)


knitr::kable(example_output2$Ds[1:3, 1:7], "latex", digits = 2, booktabs = TRUE)
```

	Status	Metformin	continuous_dummy	altered_dummy	CHN	MHD	SWE
MS0001	-0.14	-0.16	0.01	0.08	-0.43	0.19	0.35
MS0006	0.20	0.13	-0.02	0.13	0.12	-0.49	0.67
MS0007	0.24	0.30	0.02	-0.02	0.55	-0.40	-0.18

4 sessionInfo()

```
toLatex(sessionInfo())
```

- R version 3.6.1 (2019-07-05), x86_64-pc-linux-gnu
- Locale: LC_CTYPE=en_US.UTF-8, LC_NUMERIC=C, LC_TIME=de_DE.UTF-8, LC_COLLATE=en_US.UTF-8, LC_MONETARY=de_DE.UTF-8, LC_MESSAGES=en_US.UTF-8, LC_PAPER=de_DE.UTF-8, LC_NAME=C, LC_ADDRESS=C, LC_TELEPHONE=C, LC_MEASUREMENT=de_DE.UTF-8, LC_IDENTIFICATION=C
- Running under: Ubuntu 18.04.3 LTS
- Matrix products: default
- BLAS: /usr/lib/x86_64-linux-gnu/openblas/libblas.so.3
- LAPACK: /usr/lib/x86_64-linux-gnu/libopenblasp-r0.2.20.so
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Loaded via a namespace (and not attached): BiocManager 1.30.8, BiocStyle 2.12.0, compiler 3.6.1, digest 0.6.21, evaluate 0.14, htmltools 0.4.0, knitr 1.25, magrittr 1.5, Rcpp 1.0.2, rlang 0.4.0, rmarkdown 1.16, stringi 1.4.3, stringr 1.4.0, tools 3.6.1, xfun 0.10, yaml 2.2.0