Feature and Genus Category Assignment

Nate Olson

2017-04-04

Objective

Revise feature classifications to define situations that result in different performance expectation. Assign raw features and aggregated genus level features to categories.

Feature Categories

- Null - features not present in more than one PCR replicate for any sample of a biological replicate, and pipeline.
- Full - features present in at least two PCR replicates for all samples of a biological replicate, and pipeline.
- Mix - features only present in at least two PCR replicates for a mixed sample but not observed in any of the unmixed sample PCR replicates.
- Pre - present in three or more PCR replicates for unmixed pre-treatment samples, not observed in any PCR replicates of the unmixed post treatment samples, and present in at least 20 total PCR replicates.
- Post - present in three or more PCR replicates for the unmixed post-treatment samples, not observed in any PCR replicates of the unmixed pre-treatment samples, and present in at least 8 total PCR replicates.

```
## Extracting a tidy dataframe with count values from MRexpiment objects
get_count_df <- function(mrobj, agg_genus = FALSE){
      if(agg_genus){
            mrobj <- aggregateByTaxonomy(mrobj, lvl = "Rank6",
                                         norm = FALSE, log = FALSE, sl = 1)
      }

      mrobj <- cumNorm(mrobj, p = 0.75)
      mrobj %>%
            # not sure whether or not to normalize counts prior to analysis
            MRcounts(norm = TRUE, log = FALSE, sl = 1000) %>%
            as.data.frame() %>%
            rownames_to_column(var = "feature_id") %>%
            gather("id","count", -feature_id)
}


get_rep_info <- function(count_df){
      count_replicate_df <- count_df %>%
            mutate(detect = if_else(count > 0, 1, 0)) %>%
            group_by(pipe, biosample_id, titration, t_fctr, feature_id) %>%
            summarise(total_detect = sum(detect),
                      n_replicates = n(),
                      avg_non0_count = sum(count)/total_detect) %>%
            mutate(detect_prop = total_detect/n_replicates) %>%
            select(-total_detect)
      count_replicate_df %>% ungroup() %>%
      mutate(t_fctr = paste0("T",t_fctr)) %>%
      select(pipe, biosample_id, feature_id, t_fctr, detect_prop)
}


assign_cat <- function(rep_info){
      prop_summary <- rep_info %>%
            group_by(pipe, biosample_id, feature_id) %>%
            summarise(prop_max = max(detect_prop),
                      prop_min = min(detect_prop),
                      prop_sum = sum(detect_prop))

      unmix_prop <- rep_info %>%
            filter(t_fctr %in% c("T0","T20")) %>%
            spread(t_fctr, detect_prop)

      left_join(prop_summary, unmix_prop) %>%
            mutate(cat_null = if_else(prop_max < 0.5, 1, 0),
                   cat_full = if_else(prop_min >= 0.75, 1, 0),
                   cat_mix  = if_else(prop_max >= 0.5 & T0 == 0 & T20 == 0, 1, 0),
                   ## Post prop 5 - expected at least three replicates for titrations 4, 5, 10, and 15
                   ## Pre prop 3 - expected at least three replicates for titrations 1, 2, 3, and 4
                   ## titration 4, is ~94% post
                   ## titration 4, is ~94% post
                   cat_pre  = if_else(T20 >= 0.75 & T0 == 0 & prop_sum > 5, 1, 0),
                   cat_post = if_else(T0 >= 0.75 & T20 == 0 & prop_sum > 3, 1, 0),
                   cat_none = if_else(cat_null + cat_full + cat_mix + cat_pre + cat_post == 0, 1, 0))
}
```

Feature Level Category Assignments

```
count_df <- mrexp %>% map_df(get_count_df, .id = "pipe") %>%
      left_join(pData(mrexp$dada2)) %>%
      filter(biosample_id != "NTC")

rep_info <- get_rep_info(count_df)

feature_info <- assign_cat(rep_info)

feature_cat <- feature_info %>%
      select(pipe, biosample_id, feature_id,
             cat_null, cat_full, cat_mix, cat_pre, cat_post, cat_none) %>%
      gather(cat, value, -pipe, -biosample_id, -feature_id) %>%
      filter(value == 1) %>% select(-value)

feature_cat %>% saveRDS("../data/feature_categories_df.rds")
```

Category Sanity Check

```
cat_check <- feature_cat %>%
      group_by(pipe, biosample_id, feature_id) %>%
```

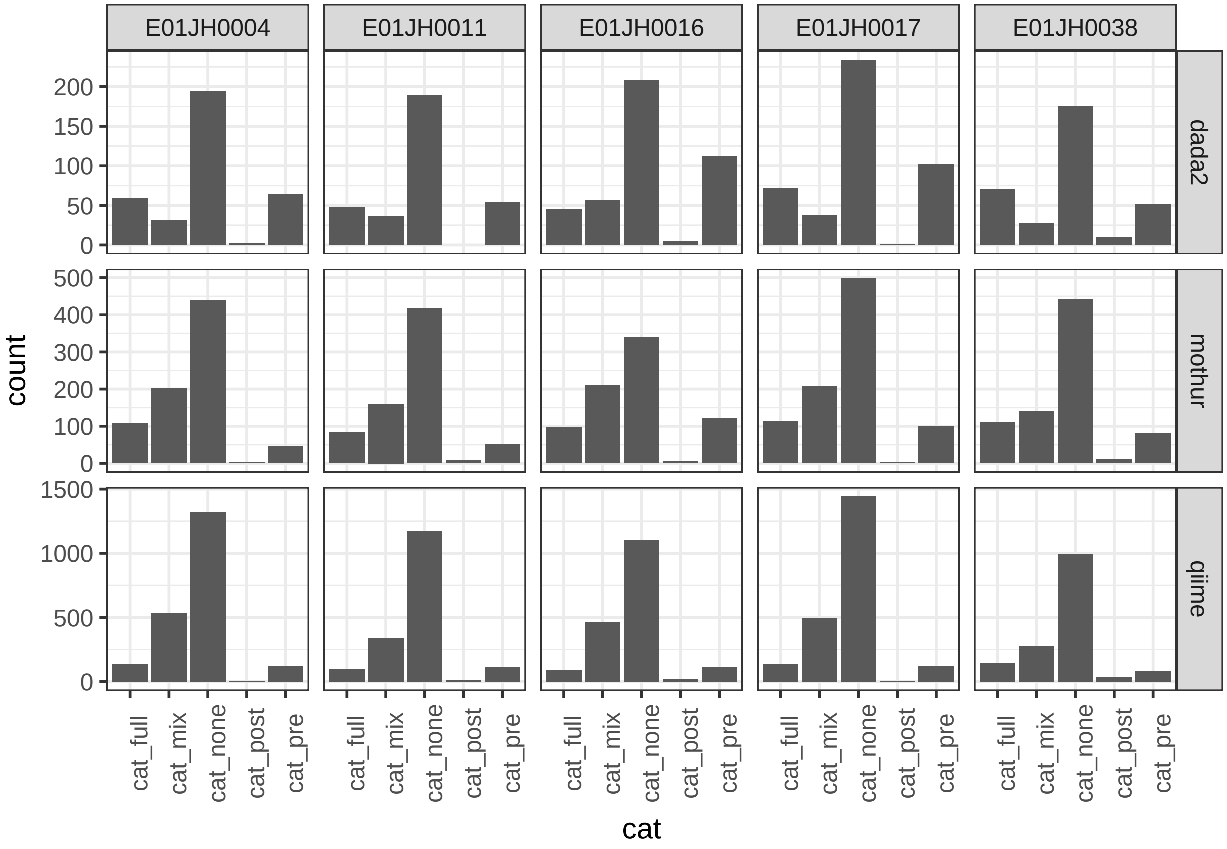
```r
      summarise(n_cat = n())
cat_check %>% filter(n_cat != 1)
```

```
## Source: local data frame [0 x 4]
## Groups: pipe, biosample_id [0]
##
## # ... with 4 variables: pipe <chr>, biosample_id <chr>, feature_id <chr>,
## #   n_cat <int>
```

```r
# cat_check <- feature_categories %>%
#       select(pipe, biosample_id, feature_id,
#              cat_null, cat_full, cat_mix, cat_pre, cat_post, cat_none) %>%
#       gather(cat, value, -pipe, -biosample_id, -feature_id) %>%
#       group_by(pipe, biosample_id, feature_id) %>%
#       mutate(n_cat = sum(value)) %>%
#       filter(n_cat != 1, value != 0)
# cat_check %>% arrange(feature_id)
```

Summary Figures

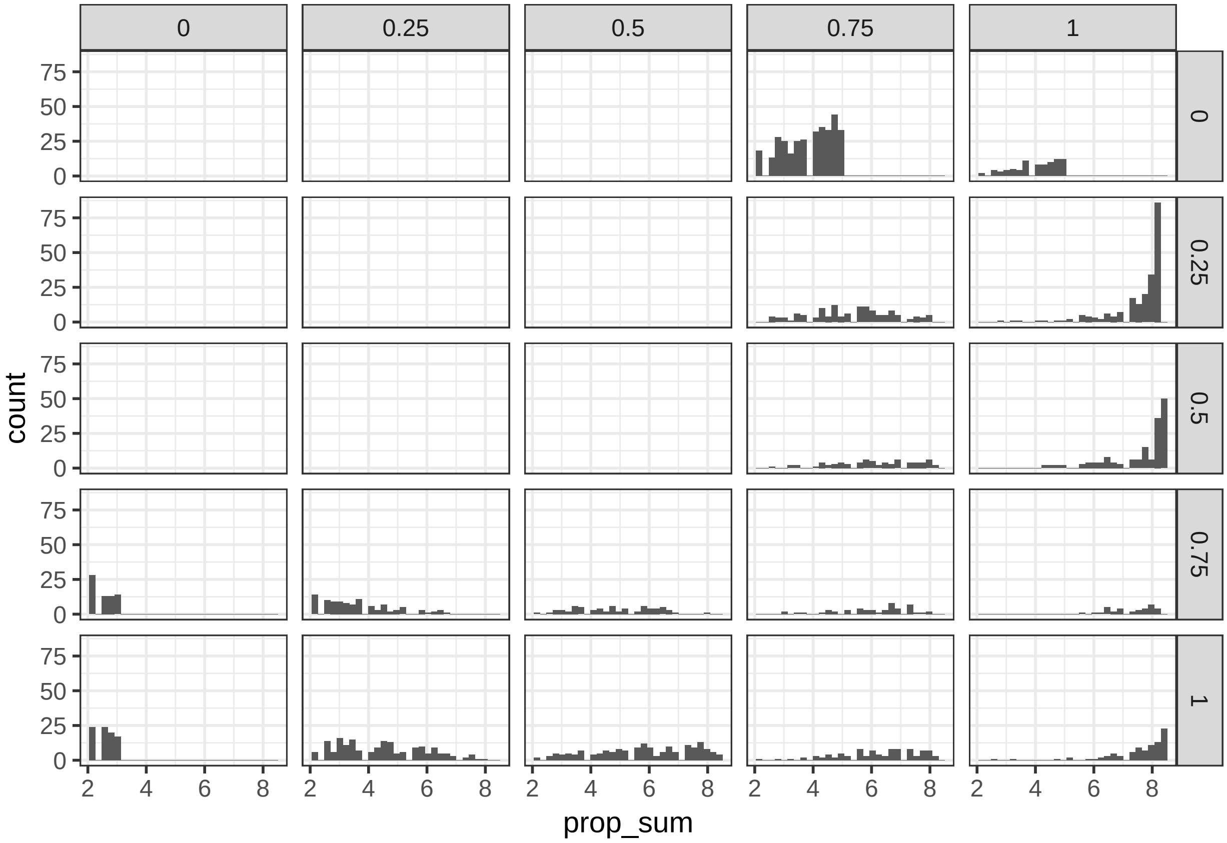
```r
feature_cat %>% filter(cat != "cat_null") %>%
      ggplot() + geom_bar(aes(x = cat)) +
      facet_grid(pipe ~ biosample_id, scales = "free_y") +
      theme_bw() + theme(axis.text.x = element_text(angle = 90))
```



While there are a large number of unclassified features, few are potentially informative. Ones that stand out are features detected in 3 of 4 T0 (pre-treatment features), and observed between 8 and 20 PCR replicates (2

< prop_sum < 5).

```
feature_info %>% filter(cat_none == 1, prop_sum > 2, T0 > 0.5 | T20 > 0.5) %>%
      ggplot() + geom_histogram(aes( x= prop_sum)) + facet_grid(T0 ~ T20) + theme_bw()
```



Genus Level Category Assignments

```
count_df <- mrexp %>% map_df(get_count_df,agg_genus = TRUE, .id = "pipe") %>%
      left_join(pData(mrexp$dada2)) %>%
      filter(biosample_id != "NTC")

rep_info <- get_rep_info(count_df)

feature_info <- assign_cat(rep_info)

feature_cat <- feature_info %>%
      select(pipe, biosample_id, feature_id,
             cat_null, cat_full, cat_mix, cat_pre, cat_post, cat_none) %>%
      gather(cat, value, -pipe, -biosample_id, -feature_id) %>%
      filter(value == 1) %>% select(-value)

feature_cat %>% saveRDS("../data/genus_categories_df.rds")
```

Category Sanity Check

```
cat_check <- feature_cat %>%
      group_by(pipe, biosample_id, feature_id) %>%
      summarise(n_cat = n())
cat_check %>% filter(n_cat != 1)
```

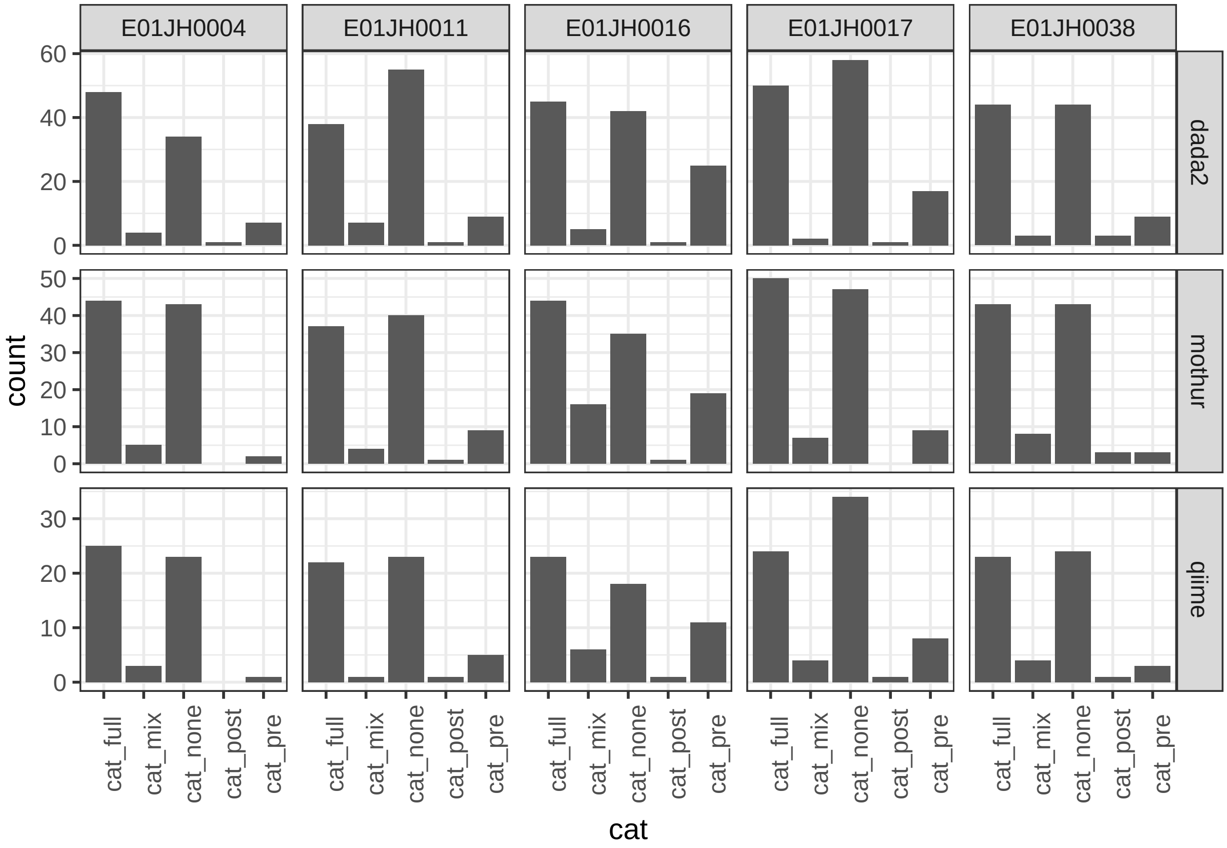
```
## Source: local data frame [0 x 4]
## Groups: pipe, biosample_id [0]
##
## # ... with 4 variables: pipe <chr>, biosample_id <chr>, feature_id <chr>,
## #   n_cat <int>
```

```
# cat_check <- feature_categories %>%
#       select(pipe, biosample_id, feature_id,
#              cat_null, cat_full, cat_mix, cat_pre, cat_post, cat_none) %>%
#       gather(cat, value, -pipe, -biosample_id, -feature_id) %>%
#       group_by(pipe, biosample_id, feature_id) %>%
#       mutate(n_cat = sum(value)) %>%
#       filter(n_cat != 1, value != 0)
# cat_check %>% arrange(feature_id)
```

Summary Figures

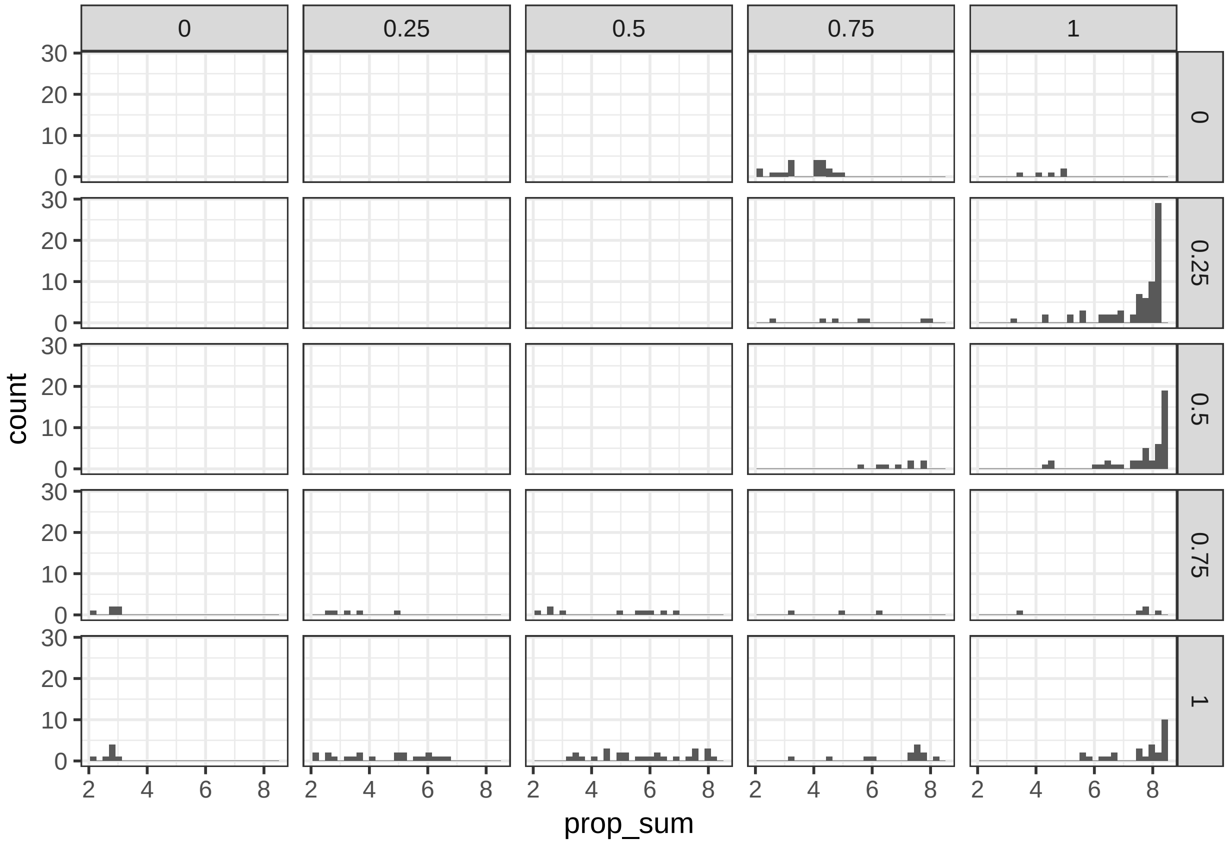
Larger proportion of full category features and fewer mix specific features when aggregating to the genus level compared to unaggregated features.

```
feature_cat %>% filter(cat != "cat_null") %>%
      ggplot() + geom_bar(aes(x = cat)) +
      facet_grid(pipe ~ biosample_id, scales = "free_y") +
      theme_bw() + theme(axis.text.x = element_text(angle = 90))
```



While there are a large number of unclassified features, few are potentially informative. Ones that stand out are features detected in 4 T0 (pre-treatment features) with prop sum value close to 8.

```
feature_info %>% filter(cat_none == 1, prop_sum > 2, T0 > 0.5 | T20 > 0.5) %>%
      ggplot() + geom_histogram(aes( x= prop_sum)) + facet_grid(T0 ~ T20) + theme_bw()
```



Session information

```
s_info <- devtools::session_info()
print(s_info$platform)
```

```
##  setting  value
##  version  R version 3.3.3 (2017-03-06)
##  system   x86_64, darwin15.6.0
##  ui       unknown
##  language (EN)
##  collate  en_US.UTF-8
##  tz       America/New_York
##  date     2017-04-04
```

```
s_info$packages %>% filter(`*` == "*") %>% select(-`*`) %>%
      knitr::kable()
```

package	version	date	source
bbmle	1.0.18	2016-02-11	CRAN (R 3.3.2)
Biobase	2.34.0	2016-11-07	Bioconductor
BiocGenerics	0.20.0	2016-11-07	Bioconductor
BiocParallel	1.8.1	2016-11-07	Bioconductor
Biostrings	2.42.1	2016-12-19	Bioconductor
DESeq	1.26.0	2016-11-28	Bioconductor
DESeq2	1.15.28	2017-02-02	bioc (readonly/DESeq2@125913)

package	version	date	source
dplyr	0.5.0	2016-06-24	CRAN (R 3.3.2)
edgeR	3.16.5	2017-02-02	Bioconductor
forcats	0.2.0	2017-01-23	CRAN (R 3.3.2)
foreach	1.4.3	2015-10-13	CRAN (R 3.3.1)
GenomeInfoDb	1.10.3	2017-03-28	Bioconductor
GenomicAlignments	1.10.1	2017-03-28	Bioconductor
GenomicRanges	1.26.4	2017-03-28	Bioconductor
ggplot2	2.2.1	2016-12-30	CRAN (R 3.3.2)
glmnet	2.0-5	2016-03-17	CRAN (R 3.3.1)
IRanges	2.8.2	2017-03-28	Bioconductor
knitr	1.15.1	2016-11-22	CRAN (R 3.3.2)
lattice	0.20-34	2016-09-06	CRAN (R 3.3.3)
limma	3.30.13	2017-03-28	Bioconductor
locfit	1.5-9.1	2013-04-20	CRAN (R 3.3.1)
Matrix	1.2-8	2017-01-20	CRAN (R 3.3.3)
metagenomeSeq	1.16.0	2016-11-07	Bioconductor
modelr	0.1.0	2016-08-31	cran (@0.1.0)
permute	0.9-4	2016-09-09	CRAN (R 3.3.1)
phyloseq	1.19.1	2017-01-04	Bioconductor
ProjectTemplate	0.7	2016-08-11	CRAN (R 3.3.1)
purrr	0.2.2	2016-06-18	CRAN (R 3.3.1)
RColorBrewer	1.1-2	2014-12-07	CRAN (R 3.3.1)
readr	1.1.0	2017-03-22	CRAN (R 3.3.2)
readxl	0.1.1	2016-03-28	cran (@0.1.1)
Rqc	1.8.0	2016-11-07	Bioconductor
Rsamtools	1.26.1	2016-11-07	Bioconductor
S4Vectors	0.12.2	2017-03-28	Bioconductor
sads	0.3.1	2016-05-13	CRAN (R 3.3.2)
savR	1.12.0	2016-11-07	Bioconductor
ShortRead	1.32.1	2017-03-28	Bioconductor
stringr	1.2.0	2017-02-18	CRAN (R 3.3.2)
SummarizedExperiment	1.4.0	2016-11-07	Bioconductor
tibble	1.2	2016-08-26	CRAN (R 3.3.1)
tidyr	0.6.1	2017-01-10	CRAN (R 3.3.2)
tidyverse	1.1.1	2017-01-27	CRAN (R 3.3.2)
vegan	2.4-2	2017-01-17	CRAN (R 3.3.2)
XVector	0.14.1	2017-03-28	Bioconductor