Negative Binomial Expected Counts and EO Metric

Nate Olson

2017-04-14

Objective

Generate a data_frame with observed and predicted count values for individual PCR replicates. Expected values are calculated using unmixed PCR count table values from the same set of PCR replicates, same half of one of the replicate 96 well plates.

Negative Binomal for Weighted Count Estimates

Calculating proportion of pre and post counts using negative binomial.

- $q_{i,j,k}$ is the proportion of feature i in PCR k of sample j where a sample is defined as an individual unmixed or mixed samples for a biological replicate.
- $p_{j,k}$ is the total feature abundance for sample j, sum of all feature counts not the number of sequences generated for the sample.
- $v_{i,j,k}$ is the variance of feature i in PCR replicate j of sample k.

$$v_{i,j,k} = \frac{q_{i,j,k}(1 - q_{i,j,k})}{p_j, k}$$

- $w_{i,j,k}$ is the weight function

$$w_{i,j,k} = \frac{v_{i,j,k}^{-1}}{\sum_{k \in j} v_{i,j,k}^{-1}}$$

- $q_{i,j}$ - the weighted count estimate for feature i, k

$$q_{i,j} = \sum_{k \in j} w_{i,j,k} q_{i,j,k}$$

EO Metric

$$\frac{expected - observed}{expected + observed}$$

Functions for Processing Data

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

```
      if(css == TRUE){
            mrobj <- cumNorm(mrobj, p = 0.75)
            count_mat <- MRcounts(mrobj, norm = TRUE, log = FALSE, sl = 1000)
      }else{
            count_mat <- MRcounts(mrobj, norm = FALSE, log = FALSE, sl = 1)
      }
       count_mat %>%
            as.data.frame() %>%
            rownames_to_column(var = "feature_id") %>%
            gather("id","count", -feature_id)
}

## Calculating feature level pre and post proportions
calc_pre_post_prop <- function(count_df){
      ## Estimating $q_{i}$ for pre and post
      nb_est <- count_df %>% filter(t_fctr %in% c(0, 20)) %>%
            mutate(prop = count/total_abu,
                   prop_var = (prop * (1 - prop))/total_abu,
                   inv_var = 1/prop_var) %>%
            group_by(pipe, biosample_id, t_fctr, feature_id) %>%
            mutate(weight = inv_var / sum(inv_var)) %>%
            summarise(prop_est = sum(weight*prop))

      # Reformatting data
      nb_est %>% ungroup() %>%
            mutate(treat = if_else(t_fctr == "20", "pre","post")) %>%
            select(-t_fctr) %>%
            mutate(prop_est = if_else(is.nan(prop_est), 0, prop_est)) %>%
            spread(treat, prop_est)
}

calc_expected_prop <- function(pre_post_prop){
      titration_list <- data_frame(titration = c(1:5,10,15)) %>%
            mutate(post_prop = 2^-titration) %>%
            list() %>% rep(nrow(pre_post_prop))

      pre_post_prop %>% ungroup() %>%
            add_column(titration = titration_list) %>% unnest() %>%
            mutate(exp_prop = post * post_prop + pre * (1-post_prop)) %>%
            mutate(t_fctr = factor(titration)) %>%
            select(-post_prop)
}


calc_expected_count <- function(exp_prop_df, count_df){
      count_df %>%
            filter(t_fctr %in% c(1:5, 10, 15)) %>%
            left_join(exp_prop_df) %>%
            mutate(exp_count = total_abu * exp_prop) %>%
            filter(!(pre == 0 & post == 0 & count == 0))
}

calc_eo_metric <- function(exp_count_df){
```

```
      exp_count_df %>%
      mutate(eo_metric = (count - exp_count)/(count + exp_count))
}
```

Feature Level Expected Counts

```
count_df <- mrexp %>% map_df(get_count_df, css = FALSE, .id = "pipe") %>%
      left_join(pData(mrexp$dada2)) %>%
      filter(biosample_id != "NTC") %>%
      select(pipe, biosample_id, id, pcr_rep, feature_id, t_fctr, count)

count_df <- count_df %>% group_by(id) %>% mutate(total_abu = sum(count))


eo_exp_count_df <- count_df %>%
      calc_pre_post_prop() %>%
      calc_expected_prop() %>%
      calc_expected_count(count_df) %>%
      calc_eo_metric()

eo_exp_count_df %>% saveRDS("../data/nb_expected_eo_metric_feature_df.rds")
```

Genus Level Expected Counts

```
count_df <- mrexp %>% map_df(get_count_df, agg_genus = TRUE, css = FALSE, .id = "pipe") %>%
      left_join(pData(mrexp$dada2)) %>%
      filter(biosample_id != "NTC") %>%
      select(pipe, biosample_id, id, pcr_rep, feature_id, t_fctr, count)

count_df <- count_df %>% group_by(id) %>% mutate(total_abu = sum(count))

eo_exp_count_df <- count_df %>%
      calc_pre_post_prop() %>%
      calc_expected_prop() %>%
      calc_expected_count(count_df) %>%
      calc_eo_metric()

eo_exp_count_df %>% saveRDS("../data/nb_expected_eo_metric_genus_df.rds")
```

Session information

Git repo commit information

```
library(git2r)
repo <- repository(path = "../")
last_commit <- commits(repo)[[1]]
```

The current git commit of this file is 8c96a9ff14e2f1ef6631b77ad83cfdd9e7a61395, which is on the master branch and was made by nate-d-olson on 2017-04-14 14:16:34. The current commit message is revised text and added figure text. The repository is online at https://github.com/nate-d-olson/mgtst-pub

Platform 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     2017-04-14
```

Package Versions

```
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	1.8.2	2017-04-12	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)
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)
git2r	0.18.0	2017-01-01	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	0.20-35	2017-03-25	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

package	version	date	source
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	1.26.2	2017-04-12	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	1.3.0	2017-04-01	CRAN (R 3.3.3)
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	2.4-3	2017-04-07	CRAN (R 3.3.3)
XVector	0.14.1	2017-03-28	Bioconductor