Introduction to XgeneR for homozygous crosses in a single condition.

Maria T. Carilli

```
devtools::load_all('../../XgeneR/')
```

```
## i Loading XgeneR
```

```
# library(XgeneR)
```

Load in RNAseq count data and metadata describing the samples. This example uses brown adipose tissue samples from male mice reared in cold conditions from Ballinger et al. (2023).

The counts must be a matrix in the format genes by sample, with row names gene names and column names sample names. The metadata must include a column with header "Allele" that for every sample in the count columns indicates if the sample is from parent strain 1 (P1), parent strain 2 (P2), allele specific expression in hybrids of parental allele 1 (H1), or allele specific expression in hybrids of parental allele 2 (H2).

```
count_path <- "/home/maria/XgeneR/inst/extdata/BATcold_ballinger_counts.csv"
metadata_path <- "/home/maria/XgeneR/inst/extdata/BATcold_ballinger_metadata.csv"


counts <- read.csv(count_path, row.names = 1)
counts <- as.matrix(counts)
metadata <- read.csv(metadata_path, row.names = 1)
```

Now, create an XgeneR fitObject, which requires at a minimum counts and metadata. If testing in only a single condition, the `fields_to_test` argument is `NULL` (default).

```
# create fitObject
fit_obj <- new("fitObject",counts = counts, metadata = metadata, fields_to_test = NULL)
```

Run edgeR to produce raw p-values and Benjamini-Hochberg false discovery rates (corrected by the number of genes per test) for the null hypotheses of "null: no cis" regulation and no trans regulation ("null: no trans").

```
# run edgeR tests
fit_obj <- fit_edgeR(fit_obj)
```

```
## Using classic mode.
```

```
## [1] 2
```

```
head(fit_obj@raw_pvals[["null: no cis"]])
```

```
## [1] 0.4140923 0.7100664 0.8274924 0.9246898 0.6863330 0.4743206
```

```
head(fit_obj@raw_pvals[["null: no trans"]])
```

```
## [1] 0.49437880 0.88236335 0.09631475 0.92984395 0.92192280 0.65086037
```

```
head(fit_obj@BH_FDRs[["null: no cis"]])
```

```
## [1] 0.1717408 0.4901480 0.6741915 0.8506526 0.4570955 0.2215085
```

```
head(fit_obj@BH_FDRs[["null: no trans"]])
```

```
## [1] 0.16860222 0.71697565 0.00485607 0.82455508 0.80548565 0.33229856
```

Create diagnostic plots.

The following creates a histogram of raw p-values and Benjamini Hochberg corrected FDRs.

```
fig_dir <- "./figures"
if (!dir.exists(fig_dir)) {
  dir.create(fig_dir, recursive = TRUE)
}
png("./figures/pvalue_histogram_one-condition.png", width=400, height = 400)
pval_plot <- plotPvalHistograms(fit_obj)
ggplot2::ggsave("./figures/pvalue_histogram_one-condition.png",plot=pval_plot,width=4,height=4)
```

The following function `getAssignmentsandPlot` returns a data frame with regulatory assignments based on an FDR threshold of `alpha` as well as a plot that produces a visualization of the log2 of parental ratios and hybrid ratios colored by assignment in untransformed and transformed coordinate systems and the proportion cis. `results$df` is a dataframe regulatory assignments for the given `combo`, while `results$plot` is the plot.

```
png("./figures/tri_plot_one-condition.png", width=400, height = 1000)
results <- getAssignmentsAndPlot(fit_obj,alpha=0.05)
ggplot2::ggsave("./figures/tri_plot_one-condition.png",plot=results$plot,width=4,height=10)
```

Finally, plot histogram of genes assigned to each category.

```
png("./figures/tri_plot.png", width=400, height = 400)
p <- plotRegulatoryHistogram (results$df,title="Regulatory assignments quantified")
```

```
## [1] "conserved" "trans"     "cisxtrans" "cis"       "cis+trans"
```

```
ggplot2::ggsave("./figures/reg_histogram_one-condition.png",plot=p,width=4,height=4)
```