hpgltools

atb abelew@gmail.com

2016-02-02

Hpgltools: Stupid R tricks.

The following block shows how I handle autloading requisite libraries for my code. This makes it easier for me to download/install the R requirements on a new computer, something which I have found myself needing to do more than I would have guessed.

```
## This block serves to load requisite libraries and set some options.
library("hpgltools")
## To set up an initial vignette, use the following line:
## devtools::use_vignette("hpgltools")
autoloads_all()
```

```
## Note: the specification for S3 class "AsIs" in package 'RJSONIO' seems equivalent to one from packag
```

```
## Note: the specification for S3 class "AsIs" in package 'XMLRPC' seems equivalent to one from package
```

```
##
## groupGOTerms:    GOBPTerm, GOMFTerm, GOCCTerm environments built.
```

```
opts_knit$set(progress=TRUE, verbose=TRUE, purl=FALSE, error=TRUE, stop_on_error=FALSE, fig.width=7, fi
options(java.parameters="-Xmx8g")  ## used for xlconnect -- damn 4g wasn't enough
theme_set(theme_bw(base_size=10))
set.seed(1)
```

Rendering the vignette

The following block has a few lines I use to load data, save it, and render pdf/html reports. I do this under the veritable editor, 'emacs,' with the key combination "Control-c, Control-n" for each line I want to evaluate in R, or "Control-c, Control-c" for a paragraph.

```
load("RData")
rm(list=ls())
save(list=ls(all=TRUE), file="RData")
render("hpgltools.Rmd", output_format="pdf_document")
render("hpgltools.Rmd", output_format="html_document")
```

Tasks that hpgltools helps me perform

This code was written to speed up and simplify a few specific tasks:

- Reading RNA sequencing count tables (in R/count_tables.R)
- Normalization of data (R/normalization.R)

- Graphing metrics of data to check and evaludate batch effects (R/plots.R)
- Performing contrasts of the data using voom/limma (R/misc_functions.R)
- Plotting RNA abundances by condition/batch (R/plots.R)
- Simplifying ontology/KEGG searches (R/ontology.R)

The following paragraphs will attempt to show how I use it.

Annotation information

Every RNA sequencing experiment I have played with has required a different handling of the genome's annotation. Most, but not all, have kept the data of interest in a gff file. Here is an example of how I process one of those files and make a data frame of genes as well as tooltips, which will be used for googleVis graphs later. In every experiment I have played with, I make a 'reference' directory into which I copy the current annotation data, this way I have a consistent and known version of the annotation. In the example below, this is the TriTrypDB version 8.1 of the T. cruzi genome.

```
tcruzi_annotations = import.gff3("reference/gff/clbrener_8.1_complete.gff.gz")
annotation_info = as.data.frame(tcruzi_annotations)

genes = annotation_info[annotation_info$type=="gene",]
gene_annotations = genes
rownames(genes) = genes$Name
tooltip_data = genes
tooltip_data = tooltip_data[,c(11,12)]
tooltip_data$tooltip = paste(tooltip_data$Name, tooltip_data$description, sep=": ")
tooltip_data$tooltip = gsub("\\+", " ", tooltip_data$tooltip)
rownames(tooltip_data) = tooltip_data$Name
tooltip_data = tooltip_data[-1]
tooltip_data = tooltip_data[-1]
colnames(tooltip_data) = c("name.tooltip")
head(tooltip_data)

## Here are the newer ways I do the above:
## Ahh crap though I don't have the directory reference/gff/, I only saved a rdata image for these exam
## Well, the following is an easier way of doing the above
##  new_annotations = hpgltools:::gff2df(gff="reference/gff/clbrener_8.1_complete.gff.gz")
##  tooltip_data = make_tooltips(annotations=new_annotations)  ## or gff= works too
```

Reading count tables

In Dr. El-Sayed's lab, there is a very specific naming convention for RNA sequencing experiments. Every sequencing run has an 'HPGL' (host pathogen genomics lab) identifier. All experiments have associated metadata, including the condition in the experiment, the batch, bioanalyzer reports, etc. When I play with data, I keep all this information in a csv file 'samples.csv' and the processed count-tables for the experiment in a specific directory: processed_data/. Therefore, I have a couple functions which automate the import of data into R in the hopes that no mistakes are made.

Here is an example from a recent experiment.

```
samples = read.csv("data/all_samples.csv")
```

```
## Warning in file(file, "rt"): cannot open file 'data/all_samples.csv': No
## such file or directory
```

```
## Error in file(file, "rt"): cannot open the connection
```

```
knitr::kable(head(samples))
```

---

structure(function (object) standardGeneric("samples"), generic = structure("samples", package = "Biobase"), package = "Biobase", group = list(), valueClass = character(0), signature = "object", default = \001NULL\001, skeleton = (function (object) stop("invalid call in method dispatch to 'samples' (no default method)", domain = NA))(object), class = structure("standardGeneric", package = "methods"))
Since I didn't want to copy over all my count tables, you, dear reader, will have to trust that there is a file for each entry in the above table which corresponds to the Sample.ID. These may be organized by sample name or condition. The following code shows how I create an expressionset and fill it with the count data.

```
r example_data = counts(make_exampledata(ngenes=10000, columns=24)) ## create_expt()
usually expects that there are a bunch of count tables ## from htseq in the directory:
processed_data/count_tables/ ## These may be organised in separate directories by
condition(type) ## in one directory each by sample.  By default, this assumes they
will be ## named sample_id.count.gz, but this may be changed with the suffix argument.
all_expt = create_expt("data/all_samples.csv", count_dataframe=example_data)
## Warning in file(file, "rt"): cannot open file 'data/all_samples.csv': No ## such
file or directory
## Error in file(file, "rt"): cannot open the connection
```

Examining data

Once the data is read in, the first task is always to look at it and evaluate for batch effects and thus decide what to do about them. However, different normalization methods are appropriate in different data sets, therefore I have some functions which attempt to make this easier. For this, I will make a dummy data set using limma's makeExampleData()

"'r ## graph_metrics() performs the following: ## runs a libsize plot, non-zero genes plot, boxplot, correlation/distance heatmaps, and pca plots ## It performs a normalization of the data (log2(quantile(cpm)) by default), and does it again ## It then uses limma's removeBatchEffect() to make a stab at removing batch effect, and does it again.
An important thing to remember: the data from makeExampleData() is not very interesting, so the resulting ## plots are also not interesting... fun = graph_metrics(expt=all_expt) "'

```
## Graphing number of non-zero genes with respect to CPM by library.
## Graphing library sizes.
## Warning in hpgl_libsize(expt, title = libsize_title, ...): restarting ##
interrupted promise evaluation
## Graphing a boxplot.
## Warning in hpgl_boxplot(expt, title = boxplot_title, ...): restarting ##
interrupted promise evaluation
## Graphing a correlation heatmap.
## Warning in hpgl_heatmap(data, colors = colors, design = design, method = ## method,
: restarting interrupted promise evaluation
## Graphing a standard median correlation.
## Warning in hpgl_smc(expt, method = cormethod, title = smc_title, ...): ##
restarting interrupted promise evaluation
## Graphing a distance heatmap.
## Warning in hpgl_heatmap(data, colors = colors, design = design, method = ## method,
: restarting interrupted promise evaluation
## Graphing a standard median distance.
## Warning in hpgl_smd(expt, method = distmethod, title = smd_title, ...): ##
restarting interrupted promise evaluation
## Graphing a PCA plot.
## Warning in hpgl_pca(expt, title = pca_title, ...): restarting interrupted ##
promise evaluation
```

```
## Plotting a density plot.
## Warning in hpgl_density(expt, title = dens_title): restarting interrupted ##
promise evaluation
## Error in pca$plot: $ operator is invalid for atomic vectors
r fun
## Error in eval(expr, envir, enclos): object 'fun' not found
```
“‘r ## The following are some examples of other ways to make use of these plots:
##fun_boxplot = hpgl_boxplot(df=fun) ##print(fun_boxplot) ##log_boxplot =
hpgl_boxplot(df=fun, scale=“log”) ##print(log_boxplot) ##hpgl_corheat(df=fun, colors=hpgl_colors)
##hpgl_disheat(df=fun, colors=hpgl_colors) ##hpgl_smc(df=fun, colors=hpgl_colors)
##hpgl_libsize(df=fun) ##hpgl_qq_all(df=fun) “‘
Normalizing data
RNAseq data must be normalized. Here is one easy method:
```
r ## normalize_expt will do this on the expt class, replace the expressionset therein,
and ## make a backup of the data inside the expt class. norm_expt =
normalize_expt(all_expt)
## Error in normalize_expt(all_expt): object 'all_expt' not found
r head(exprs(norm_expt$expressionset))
```

```
##             GSM1368273 GSM1368274 GSM1368275 GSM1368276 GSM1368277 ## SPAC212.11
-1.1408282 -0.8259441  0.8035750  0.4766813  0.2390415 ## SPAC212.09c  0.2934339
1.8797086  1.7851730  1.5340145  2.1853216 ## SPNCRNA.70  -4.6503833 -4.6503726
-4.6503815 -4.6503099 -4.6503780 ## SPAC212.12  -3.7759142 -4.6503726 -4.6503815
-4.6503099 -4.6503780 ## SPAC212.04c  0.9659613 -0.5465020  0.5827055  0.9545521
-0.5051534 ## SPAC212.03  -4.6503833 -4.6503726 -4.6503815 -4.6503099 -4.6503780 ##
GSM1368278 GSM1368279  GSM1368280  GSM1368281 GSM1368282 ## SPAC212.11   2.5543907
-0.6304741 -0.08898623  0.01890437 -1.5074161 ## SPAC212.09c  2.4002729  2.5885617
3.18531929  2.82760866  2.6416919 ## SPNCRNA.70  -4.4981773 -4.6503737 -4.65038026
-2.57237190 -4.6503740 ## SPAC212.12  -0.3854772 -1.1692470 -4.65038026 -4.65037441
-2.4279816 ## SPAC212.04c  0.2365323  1.0707255  1.97654554  2.36271558  0.6993372 ##
SPAC212.03  -4.4981773 -4.6503737 -4.65038026 -4.65037441 -4.6503740 ##
GSM1368283 GSM1368284 GSM1368285 GSM1368286  GSM1368287 ## SPAC212.11  -1.1836837
-0.7434938 -2.2055988  -4.650382 -0.06541247 ## SPAC212.09c  2.1626853  2.4505699
2.0098104   1.817223  2.71740378 ## SPNCRNA.70  -4.6503833 -4.6503844 -4.6503837
-4.650382 -4.65037497 ## SPAC212.12  -4.6503833 -4.6503844 -3.6108553  -4.650382
-4.65037497 ## SPAC212.04c  0.6675317  0.8939361  0.8870505   0.110430  1.11191478 ##
SPAC212.03  -4.6503833 -4.6503844 -4.6503837  -4.650382 -4.65037497 ##
GSM1368288 GSM1368289 GSM1368290 GSM1368291 GSM1368292 ## SPAC212.11  -0.3608774
-1.190949   0.232261 -1.4760109 -0.9346109 ## SPAC212.09c  1.7149663   2.595437
2.004652  0.5762979  1.3633281 ## SPNCRNA.70  -4.6503840  -4.650380  -4.650382
-4.6503818 -4.6503823 ## SPAC212.12  -4.6503840  -4.650380  -2.797939 -4.6503818
-2.6503823 ## SPAC212.04c  0.9362473   1.031444   0.956126  1.0267505  1.4276202 ##
SPAC212.03  -4.6503840  -4.650380  -4.650382 -4.6503818 -3.4279899 ##
GSM1368293 GSM1368294 GSM1368295 GSM1368296 GSM1368297 ## SPAC212.11   -1.312512
0.4024945   0.110430  0.3988949  -2.250451 ## SPAC212.09c   1.890499  1.8379111
1.783319  1.9765772   2.015744 ## SPNCRNA.70   -4.650382 -4.6503754  -4.650382
-4.6503486  -4.650381 ## SPAC212.12   -4.650382 -4.6503754  -4.650382 -4.6503486
-4.650381 ## SPAC212.04c   2.157688  2.4192982   1.251825  1.9182409   3.225283 ##
SPAC212.03   -4.650382 -4.6503754  -4.650382 -4.6503486  -4.650381 ##
GSM1368298 GSM1368299 GSM1368300 GSM1368301 GSM1368302 ## SPAC212.11   -1.065419
-0.5006395  -4.650383  -1.775908  -3.462756 ## SPAC212.09c   2.081315  2.0873302
1.293434   2.506690   2.618484 ## SPNCRNA.70   -4.650381 -4.6503866  -4.650383
-4.650377  -4.650383 ## SPAC212.12   -2.462754 -4.6503866  -3.534906  -2.025886
-4.650383 ## SPAC212.04c   3.092396  2.9866407   2.238360   2.421384   1.810869 ##
SPAC212.03   -3.235344 -4.6503866  -4.650383  -4.650377  -4.650383 ##
GSM1368303 GSM1368304 GSM1368305 GSM1368306 GSM1368307 ## SPAC212.11   -2.843033
-4.650381 -0.9623242  -1.781387  -0.544183 ## SPAC212.09c   1.819027   2.135344
1.6625028   1.994277   1.172631 ## SPNCRNA.70   -4.650388  -4.650381 -4.6503802
-4.650381  -4.650382 ## SPAC212.12   -4.650388  -3.650381 -4.6503802  -4.650381
-4.650382 ## SPAC212.04c   1.614836   2.327535  2.1148599   2.100047   1.632449 ##
SPAC212.03   -3.820313  -4.650381 -4.6503802  -4.650381  -4.650382 ##
GSM1368308 ## SPAC212.11  -0.2967356 ## SPAC212.09c  2.5134169 ## SPNCRNA.70
-4.6503725 ## SPAC212.12  -2.5348953 ## SPAC212.04c  1.8172330 ## SPAC212.03
-4.6503725
r ## size factor, tmm, rle, upperQuartile all require a design matrix. norm_boxplot =
hpgl_boxplot(expt=norm_expt)
## Error in hpgl_boxplot(expt = norm_expt): argument "data" is missing, with no
default
r print(norm_boxplot)
## Error in print(norm_boxplot): error in evaluating the argument 'x' in selecting a
method for function 'print': Error: object 'norm_boxplot' not found
r norm_disheat = hpgl_disheat(expt=norm_expt)
```

```
## Error in hpgl_heatmap(data, colors = colors, design = design, method = method, :
argument "data" is missing, with no default
r print(norm_disheat)
## Error in print(norm_disheat): error in evaluating the argument 'x' in selecting a
method for function 'print': Error: object 'norm_disheat' not found
```

Voom/limma etc

There are a couple ways to call limma using the expt class. In some cases, it might be useful to pull out a subset of the data and only compare the samples of specific conditions/batches/etc.

```
r ## el_subset means to pull out only those samples which represent 'Early Log'
growth. el_subset = expt_subset(norm_expt, "stage=='EL'") ## Conversely, one may pull
samples which are early log and also wild type elwt_subset = expt_subset(norm_expt,
"stage=='EL'&type=='WT'") ## These subsets may be characterized with the plots as
above ## Here is a qq plot as an example. elwt_qqs = hpgl_qq_all(expt=elwt_subset)
## Error in hpgl_qq_all(expt = elwt_subset): unused argument (expt = elwt_subset)
r ## Simple comparison will take the first condition as control and the second ## as
experimental, if we look at el_subset, we will see that means conditions ## 'a' and
'b'.  Thus performing simple_comparison will look for differentially ## expressed
genes between them. head(el_subset$design)
##  [1] strain      minute      replicate   id          condition ##  [6] batch
sample.id   colors      stage       type ## [11] counts      intercounts ## <0 rows>
(or 0-length row.names)
r ab_comparison = simple_comparison(el_subset)
## Error in `contrasts<-`(`*tmp*`, value = contr.funs[1 + isOF[nn]]): contrasts can be
applied only to factors with 2 or more levels
r ## A summary of the data will show the data provided: ## The following plots and
pieces of data show the output provided by simple_comparison() ## This function isn't
really intended to be used, but provides a reference point for performing other
analyses. summary(ab_comparison)
## Error in summary(ab_comparison): error in evaluating the argument 'object' in
selecting a method for function 'summary': Error: object 'ab_comparison' not found
r print(ab_comparison$amean_histogram)  ## A histogram of the per-gene mean values
## Error in print(ab_comparison$amean_histogram): error in evaluating the argument 'x'
in selecting a method for function 'print': Error: object 'ab_comparison' not found
r print(ab_comparison$coef_amean_cor)   ## The correlation of the means (should not be
significant)
## Error in print(ab_comparison$coef_amean_cor): error in evaluating the argument 'x'
in selecting a method for function 'print': Error: object 'ab_comparison' not found
r print(ab_comparison$coefficient_scatter) ## A scatter plot of condition b with
respect to a
## Error in print(ab_comparison$coefficient_scatter): error in evaluating the argument
'x' in selecting a method for function 'print': Error: object 'ab_comparison' not
found
r print(ab_comparison$coefficient_x) ## A histogram of the gene abundances of a
## Error in print(ab_comparison$coefficient_x): error in evaluating the argument 'x'
in selecting a method for function 'print': Error: object 'ab_comparison' not found
r print(ab_comparison$coefficient_y) ## A histogram of the gene abundances of b
## Error in print(ab_comparison$coefficient_y): error in evaluating the argument 'x'
in selecting a method for function 'print': Error: object 'ab_comparison' not found
r print(ab_comparison$coefficient_both) ## A histogram of the gene abundances of a and
b
## Error in print(ab_comparison$coefficient_both): error in evaluating the argument
'x' in selecting a method for function 'print': Error: object 'ab_comparison' not
found
```

```
r ## Note to self, I keep meaning to change the colors of that to match the others
print(ab_comparison$coefficient_lm) ## The description of the line which describes the
relationship
## Error in print(ab_comparison$coefficient_lm): error in evaluating the argument 'x'
in selecting a method for function 'print': Error: object 'ab_comparison' not found
r ## of all of the genes in a to those in b print(ab_comparison$coefficient_lmsummary)
## A summary of the robust linear model in coefficient_lm
## Error in print(ab_comparison$coefficient_lmsummary): error in evaluating the
argument 'x' in selecting a method for function 'print': Error: object 'ab_comparison'
not found
r ## This has some neat things like the R-squared value and the parameters used to
arrive at the linear model. ## ab_comparison$coefficient_weights ## a list of weights
by gene, bigger weights mean closer to the linear model. ## ab_comparison$comparisons
## the raw output from limma print(ab_comparison$contrasts)  ## The output from
limma's makeContrasts()
## Error in print(ab_comparison$contrasts): error in evaluating the argument 'x' in
selecting a method for function 'print': Error: object 'ab_comparison' not found
r print(ab_comparison$contrast_histogram)  ## A histogram of the values of b-a for
each gene
## Error in print(ab_comparison$contrast_histogram): error in evaluating the argument
'x' in selecting a method for function 'print': Error: object 'ab_comparison' not
found
r head(ab_comparison$downsignificant)  ## The list of genes which are significantly
down in b vs a
## Error in head(ab_comparison$downsignificant): error in evaluating the argument 'x'
in selecting a method for function 'head': Error: object 'ab_comparison' not found
r dim(ab_comparison$downsignificant)
## Error in eval(expr, envir, enclos): object 'ab_comparison' not found
r ## ab_comparison$fit ## the result from lmFit() print(ab_comparison$ma_plot)  ## An
ma plot of b vs a
## Error in print(ab_comparison$ma_plot): error in evaluating the argument 'x' in
selecting a method for function 'print': Error: object 'ab_comparison' not found
r print(ab_comparison$pvalue_histogram) ## A histogram of the p-values, one would hope
to see a spike in the low numbers
## Error in print(ab_comparison$pvalue_histogram): error in evaluating the argument
'x' in selecting a method for function 'print': Error: object 'ab_comparison' not
found
r head(ab_comparison$table) ## The full contrast table
## Error in head(ab_comparison$table): error in evaluating the argument 'x' in
selecting a method for function 'head': Error: object 'ab_comparison' not found
r head(ab_comparison$upsignificant)  ## The list of genes which are significantly up
in b vs a
## Error in head(ab_comparison$upsignificant): error in evaluating the argument 'x' in
selecting a method for function 'head': Error: object 'ab_comparison' not found
r dim(ab_comparison$upsignificant)
## Error in eval(expr, envir, enclos): object 'ab_comparison' not found
r print(ab_comparison$volcano_plot) ## A Volcano plot of b vs a
## Error in print(ab_comparison$volcano_plot): error in evaluating the argument 'x' in
selecting a method for function 'print': Error: object 'ab_comparison' not found
r ## ab_comparison$voom_data  ## The output from voom() print(ab_comparison$voom_plot)
## A ggplot2 version of the mean/variance trend provided by voom()
## Error in print(ab_comparison$voom_plot): error in evaluating the argument 'x' in
selecting a method for function 'print': Error: object 'ab_comparison' not found
```

```
r ## The data structure ab_comparison$comparisons contains the output from eBayes()
which comprises the last ## limma step... funkytown =
write_limma(data=ab_comparison$comparisons, excel=FALSE, csv=FALSE)
## Error in write_limma(data = ab_comparison$comparisons, excel = FALSE, : object
'ab_comparison' not found
r ## Lets make up some gene lengths gene_lengths = funkytown[[1]]
## Error in eval(expr, envir, enclos): object 'funkytown' not found
r gene_lengths$width = sample(nrow(gene_lengths))
## Error in sample(nrow(gene_lengths)): error in evaluating the argument 'x' in
selecting a method for function 'sample': Error in nrow(gene_lengths) : ##   error in
evaluating the argument 'x' in selecting a method for function 'nrow': Error: object
'gene_lengths' not found
r gene_lengths$ID = rownames(gene_lengths)
## Error in rownames(gene_lengths): error in evaluating the argument 'x' in selecting
a method for function 'rownames': Error: object 'gene_lengths' not found
r gene_lengths = gene_lengths[,c("ID","width")]
## Error in eval(expr, envir, enclos): object 'gene_lengths' not found
r ## And some GO categories goids=funkytown[[1]]
## Error in eval(expr, envir, enclos): object 'funkytown' not found
r all_go_categories = AnnotationDbi::keys(GO.db) goids$GO = sample(all_go_categories,
nrow(gene_lengths))
## Error in nrow(gene_lengths): error in evaluating the argument 'x' in selecting a
method for function 'nrow': Error: object 'gene_lengths' not found
r goids$ID = rownames(goids)
## Error in rownames(goids): error in evaluating the argument 'x' in selecting a
method for function 'rownames': Error: object 'goids' not found
r goids = goids[,c("ID","GO")]
## Error in eval(expr, envir, enclos): object 'goids' not found
r ontology_fun = limma_ontology(funkytown, gene_lengths=gene_lengths, goids=goids,
n=100, overwrite=TRUE)
## Error in eval(expr, envir, enclos): could not find function "limma_ontology"
r testme = head(funkytown[[1]], n=40)
## Error in head(funkytown[[1]], n = 40): error in evaluating the argument 'x' in
selecting a method for function 'head': Error: object 'funkytown' not found
r tt = simple_clusterprofiler(testme, goids=goids, gff=goids)
## Warning in readChar(con, 5L, useBytes = TRUE): cannot open compressed file ##
'geneTable.rda', probable reason 'No such file or directory'
## Error in check_clusterprofiler(gff, goids): object 'goids' not found
r ttt = cluster_trees(testme, tt)
## Error in cluster_trees(testme, tt): object 'tt' not found
r tttt = simple_topgo(testme)
## Error in make_id2gomap(goid_map = goid_map, goids_df = goids_df, overwrite =
overwrite): There is neither a id2go file nor a data frame of goids.
```

A cell-means model using all conditions and batches

"'r ## acb stands for "kept_conditions_batches" which takes too long to ## type when setting up the contrasts. acb = paste0(kept_qcpml2$conditions, kept_q$cpml2batches) kept_data = exprs(kept_qcpml2$expressionset) table(acb) ## The invocation of table() keptows me to count up the contribution of ## each condition/batch combination to the whole data set.

Doing this (as I understand it) means I do nothave to worry about ## balanced samples so much, but must be more careful to understand ## the relative contribution of each sample type to the entire data ## set.

complete_model = model.matrix(~0 + acb) complete_fit = lmFit(kept_data, complete_model) complete_voom = hpgl_voom(kept_data, complete_model) complete_voom$plot complete_model ## This is an example of what happens when I have heterogenous numbers of samples ## on each side of a contrast, so that a normal design matrix of conditions + batches ## would not work, so instead I add up the contributions of each batch (capital letters) ## and average them out, then use the resulting terms in the various contrasts below. epi_cl14 = "acbcl14_epiF" epi_clbr = "acbclbr_epiE" tryp_cl14 = "(acbcl14_trypB + acbcl14_trypD + acbcl14_trypG) / 3" tryp_clbr = "acbclbr_trypG" a60_cl14 = "(acbcl14_a60A * 2/3) + (acbcl14_a60B * 1/3)" a60_clbr = "acbclbr_a60A" a96_cl14 = "acbcl14_a96C" a96_clbr = "acbclbr_a96C" epi_cl14clbr = paste0("(",epi_cl14,")", " - ","(",epi_clbr,")") tryp_cl14clbr = paste0("(",tryp_cl14,")"," - ","(",tryp_clbr,")") a60_cl14clbr = paste0("(",a60_cl14,")"," - ","(",a60_clbr,")") a96_cl14clbr = paste0("(",a96_cl14,")"," - ","(",a96_clbr,")") epitryp_cl14 = paste0("(",tryp_cl14,")"," - ","(",epi_cl14,")") epitryp_clbr = paste0("(",tryp_clbr,")"," - ","(",epi_clbr,")") epia60_cl14 = paste0("(",a60_cl14,")"," - ","(",epi_cl14,")") epia60_clbr = paste0("(",a60_clbr,")"," - ","(",epi_clbr,")") a60a96_cl14 = paste0("(",a96_cl14,")"," - ","(",a60_cl14,")") a60a96_clbr = paste0("(",a96_clbr,")"," - ","(",a60_clbr,")") a60tryp_cl14 = paste0("(",tryp_cl14,")"," - ","(",a60_cl14,")") a60tryp_clbr = paste0("(",tryp_clbr,")"," - ","(",a60_clbr,")") ## The following contrast is messed up in some as of yet unknown way. epitryp_cl14clbr = paste0("(",epitryp_cl14,")"," - ","(",epitryp_clbr,")") ## So I will add some more contrasts using data which doesn't get screwed up epia60_cl14clbr = paste0("(",epia60_cl14,")"," - ","(",epia60_clbr,")") a60tryp_cl14clbr = paste0("(",a60tryp_cl14,")"," - ","(",a60tryp_clbr,")") a60a96_cl14clbr = paste0("(",a60a96_cl14,")"," - ","(",a60a96_clbr,")") complete_contrasts_v2 = makeContrasts(epi_cl14=epi_cl14, epi_clbr=epi_clbr, tryp_cl14=tryp_cl14, tryp_clbr=tryp_clbr, a60_cl14=a60_cl14, a60_clbr=a60_clbr, a96_cl14=a96_cl14, a96_clbr=a96_clbr, epi_cl14clbr=epi_cl14clbr, tryp_cl14clbr=tryp_cl14clbr, a60_cl14clbr=a60_cl14clbr, a96_cl14clbr=a96_cl14clbr, epitryp_cl14=epitryp_cl14, epitryp_clbr=epitryp_clbr, epia60_cl14=epia60_cl14, epia60_clbr=epia60_clbr, a60a96_cl14=a60a96_cl14, a60a96_clbr=a60a96_clbr, a60tryp_cl14=a60tryp_cl14, a60tryp_clbr=a60tryp_clbr, epitryp_cl14clbr=epitryp_cl14clbr, epia60_cl14clbr=epia60_cl14clbr, a60tryp_cl14clbr=a60tryp_cl14clbr, a60a96_cl14clbr=a60a96_cl14clbr, levels=complete_voom$design) ## This colnames() is annoyingly necessary to avoid really obnoxious contrast names. colnames(complete_contrasts_v2) = c("epi_cl14","epi_clbr","tryp_cl14","tryp_clbr","a60_cl14","a60_clbr","a96_cl14","a96_clbr","epi_cl14clbr","tryp_cl14cll kept_fits = contrasts.fit(complete_fit, complete_contrasts_v2) kept_comparisons = eBayes(kept_fits) "'

Clean conditions, batches

On the other hand, I would like to perform arbitrary comparisons among my data even when the batches and conditions look good, so I set up my model/contrast matrices a little strangely even then:

"'r all_data = exprs(norm_expt$expressionset$)$complete_model = model.matrix(0 + all_human_expt$conditions + all_human_expt$batches) ## Shorten the column names of the model so I don't have to type so much later... tmpnames = colnames(complete_model) tmpnames = gsub("all_human_expt[[:punct:]]","", tmpnames) tmpnames = gsub("conditions","", tmpnames) colnames(complete_model) = tmpnames rm(tmpnames)

complete_voom = hpgl_voom(all_data, complete_model) complete_voom$plot complete_fit = lmFit(complete_voom, complete_model)

all_contrasts = makeContrasts(## Start with the simple coefficient groupings for each condition none4=none4, none24=none24, none48=none48, none72=none72, bead4=bead4, bead24=bead24, bead48=bead48, bead72=bead72, maj4=maj4, maj24=maj24, maj48=maj48, maj72=maj72, ama4=ama4, ama24=ama24, ama48=ama48, ama72=ama72, ## Now do a few simple comparisons ## compare beads to uninfected beadnone_4=bead4-none4, beadnone_24=bead24-none24, beadnone_48=bead48-none48, beadnone_72=bead72-none72, majnone_4=maj4-none4, majnone_24=maj24-none24, majnone_48=maj48-none48, majnone_72=maj72-none72, amanone_4=ama4-none4, amanone_24=ama24-none24, amanone_48=ama48-none48, amanone_72=ama72-none72, ## compare samples to beads majbead_4=maj4-bead4, majbead_24=maj24-bead24, majbead_48=maj48-bead48, majbead_72=maj72-bead72, amabead_4=ama4-bead4, amabead_24=ama24-bead24, amabead_48=ama48-bead48, amabead_72=ama72-bead72, ## (x-z)-(a-b) ## Use this to compare major and amazonensis amamaj_bead_4=(ama4-bead4)-(maj4-bead4), amamaj_bead_24=(ama24-bead24)-(maj24-bead24), amamaj_bead_48=(ama48-bead48)-(maj48-bead48), amamaj_bead_72=(ama72-bead72)-(maj72-bead72), ## (c-d)-(e-f) where c/d are: (amazon|major/none)/(beads/none) majbead_none_4=(maj4-none4)-(bead4-none4), majbead_none_24=(maj24-none24)-(bead24-none24), majbead_none_48=(maj48-none48)-(bead48-none48), majbead_none_72=(maj72-none72)-(bead72-none72), amabead_none_4=(ama4-none4)-(bead4-none4), amabead_none_24=(ama24-none24)-(bead24-none24), amabead_none_48=(ama48-none48)-(bead48-none48), amabead_none_72=(ama72-none72)-(bead72-none72), levels=complete_voom$design) all_fits = contrasts.fit(complete_fit, all_contrasts) all_comparisons = eBayes(all_fits) limma_list = write_limma(data=all_comparisons)
all_table = topTable(all_comparisons, adjust="fdr", n=nrow(all_data)) write.csv(all_comparisons, file="excel/all_tables.csv") ## write_limma() is a shortcut for writing out all the data structures all_comparison_tables = write_limma(all_comparisons, excel=FALSE) "'
Ontology searches
The following is an example of a simplified GO search given 20 groups of genes which are from an unannotated organism, but for which blast2GO was performed.

```
r ontology_info = read.csv(file="data/trinotate_go_trimmed.csv.gz", header=FALSE,
sep="\t")
## Warning in file(file, "rt"): cannot open file 'data/ ##
trinotate_go_trimmed.csv.gz': No such file or directory
## Error in file(file, "rt"): cannot open the connection
r ##ontology_info = read.csv(file="data/transcript_go.csv.gz", header=FALSE, sep="\t")
colnames(ontology_info) =
c("gene_id","transcript_id","group","startend","blast_go","pfam_go")
## Error in colnames(ontology_info) = c("gene_id", "transcript_id", "group", : object
'ontology_info' not found
r ## Drop any entries which don't have a putative length ontology_info =
subset(ontology_info, startend != 0)
## Error in subset(ontology_info, startend != 0): error in evaluating the argument 'x'
in selecting a method for function 'subset': Error: object 'ontology_info' not found
r ## Split the column 'startend' into two columns by the '-' sign ontology_info =
as.data.frame(transform(ontology_info, startend=reshape::colsplit(startend,
split="\\-", names=c("start","end"))))
## Error in as.data.frame(transform(ontology_info, startend =
reshape::colsplit(startend, : error in evaluating the argument 'x' in selecting a
method for function 'as.data.frame': Error in transform(ontology_info, startend =
reshape::colsplit(startend,  : ##   error in evaluating the argument '_data' in
selecting a method for function 'transform': Error: object 'ontology_info' not found
r ## Make the resulting pieces into two separate columns, start and end.
ontology_info$start = ontology_info$startend$start
```

```
## Error in eval(expr, envir, enclos): object 'ontology_info' not found
r ontology_info$end = ontology_info$startend$end
## Error in eval(expr, envir, enclos): object 'ontology_info' not found
r ## Use start and end to make length ontology_info$length = abs(ontology_info$start -
ontology_info$end)
## Error in eval(expr, envir, enclos): object 'ontology_info' not found
r ## Drop the unneeded columns ontology_info =
ontology_info[,c("gene_id","transcript_id","group","start","end","length","blast_go","pfam_go")]

## Error in eval(expr, envir, enclos): object 'ontology_info' not found
r head(ontology_info)
## Error in head(ontology_info): error in evaluating the argument 'x' in selecting a
method for function 'head': Error: object 'ontology_info' not found
r ## goseq() requires mappings between ID/length and ID/GO category ## Currently I
have my toy set to assume column names, which is admittedly stupid. gene_lengths =
ontology_info[,c("transcript_id","length")]
## Error in eval(expr, envir, enclos): object 'ontology_info' not found
r colnames(gene_lengths) = c("ID","width")
## Error in colnames(gene_lengths) = c("ID", "width"): object 'gene_lengths' not found
r split_go = ontology_info[,c("transcript_id","blast_go")]
## Error in eval(expr, envir, enclos): object 'ontology_info' not found
r split_go$blast_go = as.character(split_go$blast_go)
## Error in eval(expr, envir, enclos): object 'split_go' not found
r ## The following few lines were pulled from the internet ## they serve to generate a
data structure in the format expected by goseq() ## It simply splits all space
separated GO categories into separate rows ## with the same ID
require.auto("splitstackshape") id_go = concat.split.multiple(split_go, "blast_go",
seps=" ", "long")
## This function is deprecated. Use ‘cSplit‘ instead.
## Error in is.data.table(indt): object 'split_go' not found
r id_go = as.data.frame(id_go)
## Error in as.data.frame(id_go): error in evaluating the argument 'x' in selecting a
method for function 'as.data.frame': Error: object 'id_go' not found
r colnames(id_go) = c("ID","GO")
## Error in colnames(id_go) = c("ID", "GO"): object 'id_go' not found
r go_ids = subset(id_go, GO != 0)
## Error in subset(id_go, GO != 0): error in evaluating the argument 'x' in selecting
a method for function 'subset': Error: object 'id_go' not found
r ## Pull out all entries from group 1 group_one = subset(ontology_info, group == "1")
## Error in subset(ontology_info, group == "1"): error in evaluating the argument 'x'
in selecting a method for function 'subset': Error: object 'ontology_info' not found
r group_one = group_one[,c("transcript_id","start","end")]
## Error in eval(expr, envir, enclos): object 'group_one' not found
r colnames(group_one) = c("ID","start","end")
## Error in colnames(group_one) = c("ID", "start", "end"): object 'group_one' not
found
r ## Perform the goseq() analysis group_one_go = simple_goseq(group_one,
lengths=gene_lengths, goids=go_ids)
## simple_goseq() makes some pretty hard assumptions about the data it is fed:
## It requires 2 tables, one of GOids which must have columns (gene)ID and
GO(category)
## The other table is of gene lengths with columns (gene)ID and (gene)width.
## Other columns are fine, but ignored.
```

```
## Error in simple_goseq(group_one, lengths = gene_lengths, goids = go_ids): object
'gene_lengths' not found
r group_one_go$pvalue_histogram
## Error in eval(expr, envir, enclos): object 'group_one_go' not found
r head(group_one_go$godata_interesting)
## Error in head(group_one_go$godata_interesting): error in evaluating the argument
'x' in selecting a method for function 'head': Error: object 'group_one_go' not found
r head(group_one_go$mf_subset)
## Error in head(group_one_go$mf_subset): error in evaluating the argument 'x' in
selecting a method for function 'head': Error: object 'group_one_go' not found
r group_one_go$mfp_plot
## Error in eval(expr, envir, enclos): object 'group_one_go' not found
r group_one_go$bpp_plot
## Error in eval(expr, envir, enclos): object 'group_one_go' not found
r group_one_go$ccp_plot
## Error in eval(expr, envir, enclos): object 'group_one_go' not found
r ## Print trees of the goseq() data initial_trees = goseq_trees(group_one,
group_one_go, goids_df=go_ids)
## Error in make_id2gomap(goid_map = goid_map, goids_df = goids_df, overwrite =
overwrite): object 'go_ids' not found
r initial_trees$MF
## Error in eval(expr, envir, enclos): object 'initial_trees' not found
r initial_trees$BP
## Error in eval(expr, envir, enclos): object 'initial_trees' not found
r initial_trees$CC
## Error in eval(expr, envir, enclos): object 'initial_trees' not found
```

Vignette Info

Note the various macros within the `vignette` setion of the metadata block above. These are required in order to instruct R how to build the vignette. Note that you should change the `title` field and the `\VignetteIndexEntry` to match the title of your vignette.

Styles

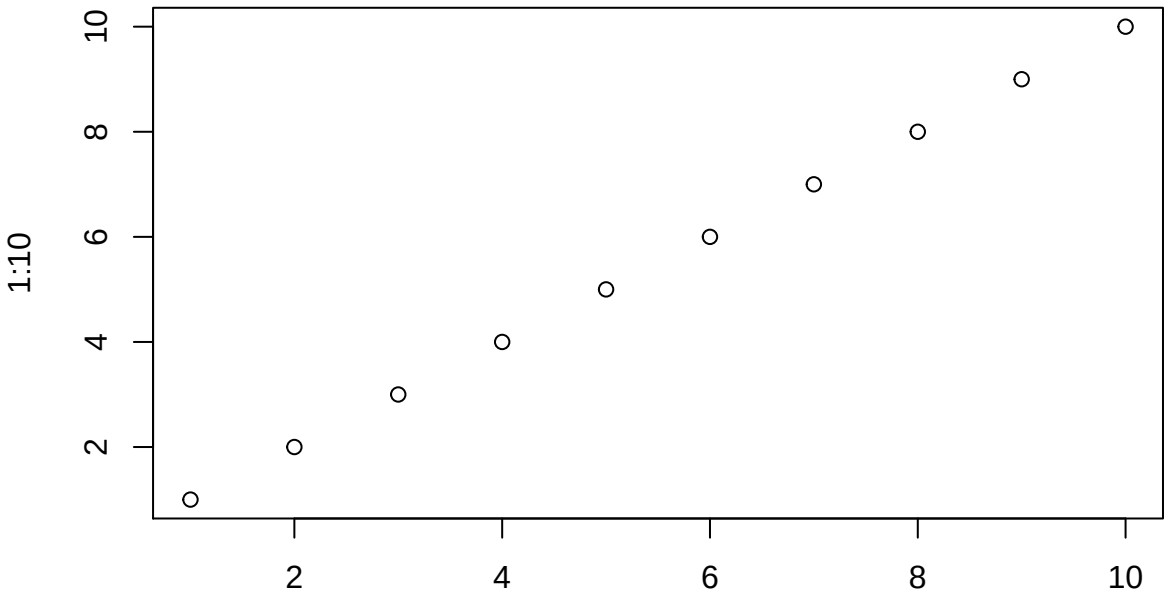
The `html_vignette` template includes a basic CSS theme. To override this theme you can specify your own CSS in the document metadata as follows:

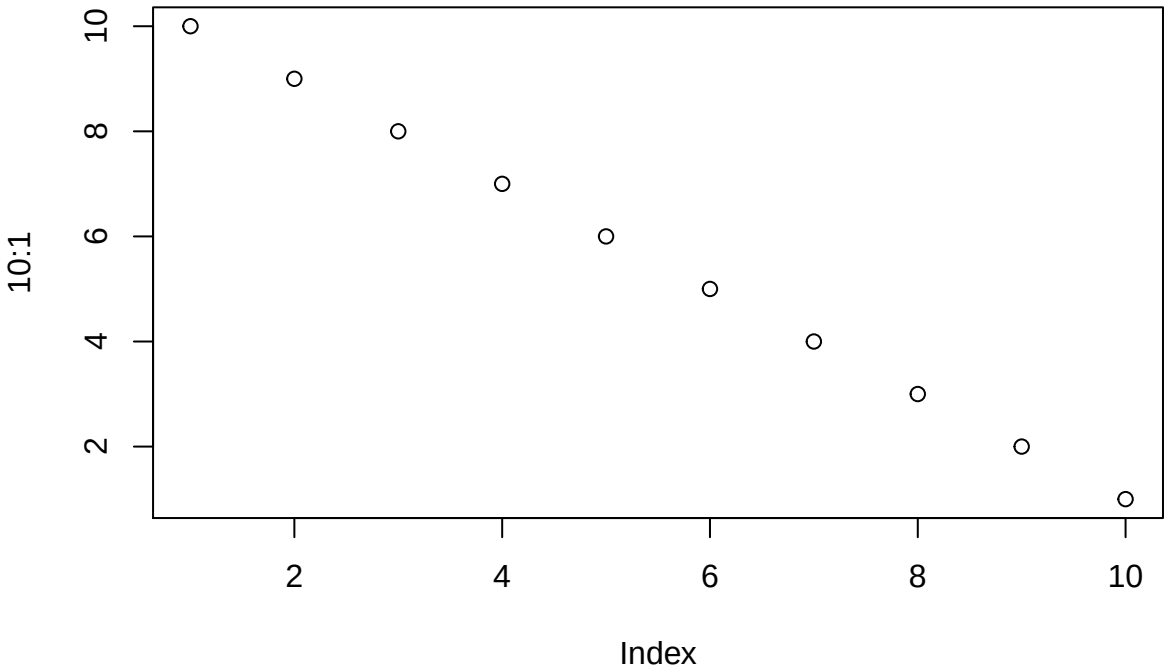
output: rmarkdown::html_vignette: css: mystyles.css

Figures

The figure sizes have been customised so that you can easily put two images side-by-side.

```
r plot(1:10) plot(10:1)
```





You can enable figure captions by `fig_caption: yes` in YAML:
output: rmarkdown::html_vignette: fig_caption: yes
Then you can use the chunk option `fig.cap = "Your figure caption."` in **knitr**.
More Examples
You can write math expressions, e.g. $Y = X\beta + \epsilon$, footnotes[1], and tables, e.g. using `knitr::kable()`.
mpg cyl disp hp drat wt qsec vs am gear carb

Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4 Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4 Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1 Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1 Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2 Valiant 18.1 6 225.0 105 2.76 3.460 20.22 1 0

[1] A footnote here.

3 1 Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4 Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2 Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2 Merc 280 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4

Also a quote using >:

> “He who gives up [code] safety for [code] speed deserves neither.” (via)