A Nested Parallel Experiment Demonstrates Differences in Intensity-Dependence Between RNA-Seq and Microarrays

David G. Robinson[1], Jean Wang[1], and John D. Storey[1,2 †]

1. Lewis-Sigler Institute for Integrative Genomics, Princeton University, Princeton, NJ 08544

2. Department of Molecular Biology, Princeton University, Princeton, NJ 08544

† To whom correspondence should be addressed: jstorey@princeton.edu

```
> library(factorial)
> data(experimentDesign)
> data(rnaseqMatrix)
> data(microarrayMatrix)
> data(microarrayRedMatrix)
> data(microarrayGreenMatrix)
> data(dubious)

> # remove 0 count genes and genes classified as "dubious"
> RS.filtered <- rnaseqMatrix[!(rowSums(rnaseqMatrix) == 0 |
+                                          rownames(rnaseqMatrix) %in% dubious), ]
> sharedGenes <- intersect(rownames(microarrayMatrix), rownames(RS.filtered))
> RS.filtered <- RS.filtered[sharedGenes, ]
> MA.filtered <- microarrayMatrix[sharedGenes, ]
> MA.red.filtered <- microarrayRedMatrix[sharedGenes, ]
> MA.green.filtered <- microarrayGreenMatrix[sharedGenes, ]

> library(limma)
> library(dplyr)
> #RS.pooled <- RS.subsets[["Full"]]
> #R.pooled <- pool_matrix(microarrayRedMatrix, experimentDesign, TRUE, average=TRUE)
> #G.pooled <- pool_matrix(microarrayGreenMatrix, experimentDesign, TRUE, average=TRUE)
> #RG <- new("RGList", list(R=R.pooled, G=G.pooled))
> # TODO: what is difference?
> RG <- new("RGList", list(R=MA.red.filtered, G=MA.green.filtered))
> matrices <- list(RS=RS.filtered, MA=RG)
> subdesign.m = with(experimentDesign, cbind(Full=1,
+                                            model.matrix(~ 0 + preparation),
```

```
+                                           model.matrix(~ 0 + lane),
+                                           model.matrix(~ 0 + preparation:lane))) == 1
> # construct one row for each subset
> subsets <- data.frame(sub = colnames(subdesign.m), stringsAsFactors = FALSE)
> subsets$bool <- lapply(subsets$sub, function(n) subdesign.m[, n])
> subsets <- data.frame(technology=c("MA", "RS"), stringsAsFactors = FALSE) %>%
+     group_by(technology) %>% do(subsets)
> # wish there were a non-lapply way to do this, not smart enough:
> subsets$matrix <- lapply(1:nrow(subsets), function(i)
+     matrices[[subsets$technology[i]]][, subsets$bool[[i]]])
> subsets$pooled <- lapply(1:nrow(subsets), function(i)
+     pool_matrix(matrices[[subsets$technology[i]]], experimentDesign,
+                 subsets$bool[[i]]))
> subsets <- data.frame(normalize.method=c("scale"), stringsAsFactors = FALSE) %>%
+     group_by(normalize.method) %>% do(subsets)
> subsets$normalized <- lapply(1:nrow(subsets), function(i)
+     factorial::normalize(subsets$pooled[[i]], subsets$normalize.method[i]))

> pooled.design <- data.frame(condition = c("E", "E", "G", "G"))
> mm <- model.matrix(~ condition, pooled.design)
> subsets$fit <- lapply(subsets$normalized, function(n) lmFit(n, mm))
> subsets$eb <- lapply(subsets$fit, eBayes)

> # add intensity
> raws = list(RS=RS.filtered, MA=MA.red.filtered)
> subsets$intensities <- lapply(1:nrow(subsets), function(i)
+     rowMeans(raws[[subsets$technology[i]]][, subsets$bool[[i]]]))
> library(biobroom)
> tidied <- subsets %>% group_by(technology, sub, normalize.method) %>%
+     do(cbind(tidy(.$eb[[1]]) %>% filter(term != "(Intercept)"), intensityraw = .$intensit
+     dplyr::select(-term)
> # add intensity and significance rank
> tidied <- tidied %>% dplyr::mutate(intensity = rank(intensityraw) / n(),
+                                    significance = rank(p.value) / n())

> library(qvalue)
> tidied <- tidied %>% mutate(q.value=qvalue(p.value)$qvalues)
> summarized.sig <- tidied %>% summarize(significant=sum(q.value < .05), pi0=qvalue(p.value

> # merging the tidyr way: gather, unite, spread
> library(tidyr)
> merged <- tidied %>% gather(metric, value, -technology, -gene, -normalize.method, -sub) %
```

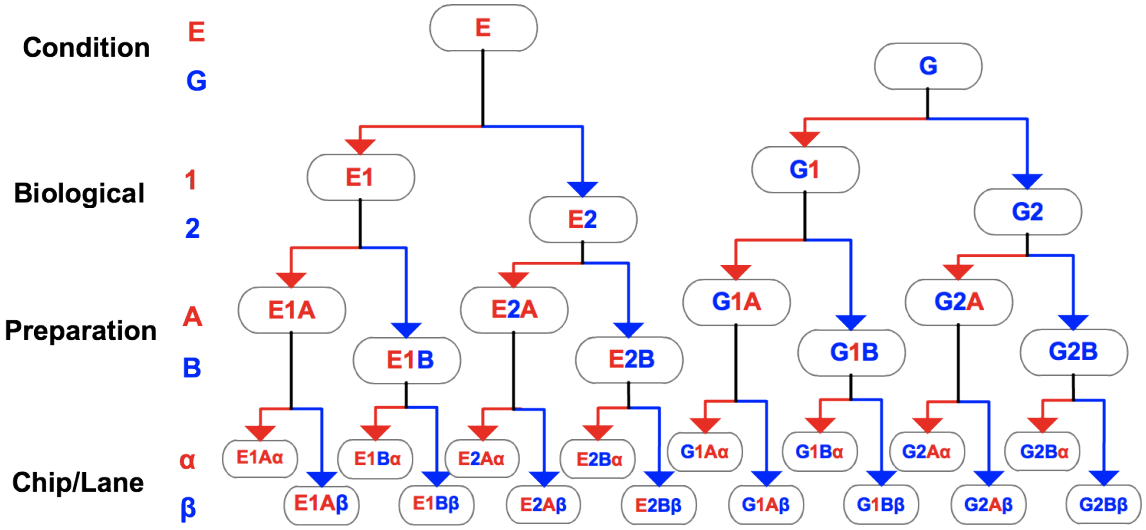

Figure 1: Schematic of the factorial experiment. The Condition and Biological steps were performed only once, while the Preparation and Chip/Lane steps were technology-specific and performed once on RNA-Seq and once on microarrays.

```
+     unite(techmet, technology, metric, sep=".") %>%
+     spread(techmet, value) %>% na.omit()
> # add in column for "minimum intensity rank"
> merged <- merged %>% mutate(intensity = pmin(MA.intensity, RS.intensity))

> # separate out the full data and combine with pathway genes
> data(pathway)
> fulldata <- merged %>% filter(sub == "Full") %>%
+     merge(pathway, by.x="gene", by.y="ORF", all=TRUE) %>%
+     dplyr::select(-sub) %>%
+     arrange(Process) %>% tbl_df()
> # process Process column to include number of elements in each
> fulldata$Process <- add_counts(factor(ifelse(is.na(fulldata$Process),
+                                              "All Other Genes",
+                                              as.character(fulldata$Process))))


> # calculating R2s also happens in a tidy framework, but only for normalization
> # methods and technologies (not subsets as happen for DE)
>
> normalize.methods <- c("scale", "quantile", "cyclicloess", "TMM", "RLE")
> geneR2s <- expand.grid(technology = c("RS", "MA"),
```

```
> library(ggplot2)
> # hardcode scale: items are alphabetical
> color_scale <- scale_color_manual(values=c("black", "blue", "lightblue", "yellow", "green
> comparefig <- ggplot(fulldata, aes(RS.estimate, MA.estimate, color = Process, alpha = int
+     geom_point() + geom_abline(col="red") +
+     xlab("RNA-Seq Log2(G/E)") + ylab("Microarray Log2(G/E)") + color_scale
> comparefig
```

Figure 2: Comparison between the log-fold change calculated from RNA-Seq or from the microarray, showing a 0.843340795314998 Pearson correlation. The transparency of the points corresponds to the quantile of the intensity in microarray or RNA-Seq (whichever is lower). 30 genes from biologically relevant pathways are highlighted in color.

```
+                         normalize.method = normalize.methods,
+                         stringsAsFactors = FALSE)
> geneR2s <- geneR2s %>% group_by(technology, normalize.method) %>%
+     do(matrix = normalize(matrices[[.$technology[[1]]]],
+                           normalize.method=.$normalize.method[[1]]))

|================================================================================================
|================================================================================================
|================================================================================================

> geneR2s <- geneR2s[!sapply(geneR2s$matrix, is.null), ]
> geneR2s <- geneR2s %>% group_by(technology, normalize.method) %>%
+     do(construct_R2s(as.matrix(.$matrix[[1]]), rowMeans(raws[[.$technology[1]]])))

> geneR2s <- geneR2s %>% group_by(technology, normalize.method, Level) %>%
+     dplyr::mutate(Quantile = rank(Intensity) / n()) %>%
+     filter(!is.na(value), !is.infinite(value))
> geneR2s_scale <- geneR2s %>% group_by() %>% filter(normalize.method == "scale")


> library(tidyr)
> tidied.1lane.1prep <- tidied %>% filter(!grepl(":", sub) & sub != "Full") %>%
+     separate(sub, c("level", "replicate"), -2) %>%
+     mutate(replicate = ifelse(replicate %in% c("1", "A"), "rep1", "rep2"))
> # lane to lane and prep to prep comparisons
> tidied.coefs <- tidied.1lane.1prep %>%
+     dplyr::select(technology, level, gene, replicate, estimate) %>%
+     spread(replicate, estimate) %>%
```

```
> g <- ggplot(geneR2s_scale, aes(Quantile, value, color=Level, lty=technology)) +
+     geom_smooth(method="loess", se=FALSE) +
+     xlab("Quantile of microarray intensity or RNA-Seq read depth") +
+     ylab("Smoothed % of variation explained")
> g
```

Figure 3: Percent of variance explained by each nested level of the experiment as computed by ANOVA's adjusted R^2, smoothed using LOESS across the intensities.

```
> gnorm <- ggplot(geneR2s, aes(Quantile, value, color=Level, lty=normalize.method)) +
+     geom_smooth(method="loess", se=FALSE) +
+     facet_wrap(~ technology) +
+     xlab("Quantile of microarray intensity or RNA-Seq read depth") +
+     ylab("Smoothed % of variation explained")
> gnorm
```

Figure 4: Percent of variance explained by each nested level of the experiment as computed by ANOVA's adjusted R^2, smoothed using LOESS across the intensities.

```
+     mutate(absdiff = abs(rep2 - rep1))
> # use the intensity from the full data
> fullintensity <- tidied %>% filter(sub == "Full") %>% group_by() %>%
+     dplyr::select(gene, technology, intensity)
> tidied.coefs <- tidied.coefs %>% inner_join(fullintensity)


> absdiff.fig <- ggplot(tidied.coefs, aes(intensity, absdiff, color=technology)) +
+     geom_smooth(method="loess") + facet_wrap(~ level)

> library(TTR)
> window.size <- 500
> tidied.coefs <- tidied.coefs %>% arrange(technology, level, intensity) %>%
+     group_by(technology, level) %>%
+     mutate(correlation=runCor(rep1, rep2, n=250))
> tidied.coefs %>% filter(level == "lane") %>%
+     ggplot(aes(intensity, correlation, color=technology)) + geom_line()
```

Figure 5: Correlation between the 2 lanes of RNA-Seq or two chips of microarrays, within a 500 gene window rolling over intensity.

```
> MA_smear <- ggplot(fulldata, aes(MA.intensityraw, MA.estimate, color = Process)) + geom_p
> RS_smear <- ggplot(fulldata, aes(MA.intensityraw, MA.estimate, color = Process)) + geom_p
> MA_smear
```

Figure 6: Smear plots of the microarray and RNA-Seq experiments.

1 Supplemental Figures

```
> library(GSEAMA)
> library(org.Sc.sgd.db)
> mm <- GOMembershipMatrix(org.Sc.sgdGO, min.size = 5, max.size = 250)

> merged.metrics <- fulldata %>%
+     dplyr::select(gene, MA.p.value, RS.p.value, MA.estimate, RS.estimate) %>%
+     gather(techmet, value, -gene) %>%
+     separate(techmet, c("technology", "ignore", "metric"), c(2, 3)) %>%
+     dplyr::select(-ignore)
> test_association <- function(metric, genes) {
+     # if it's p-values, use an alternative hypothesis of "less"
+         alternative <- ifelse(all(metric >= 0 & metric <= 1), "less", "two.sided")
+         # using "less" may have strange side effects- looking into it
+         # alternative = "two.sided"
+         TestAssociation(mm, genes, metric, method="wilcoxon", alternative=alternative)
+ }
> GO_tab = toTable(org.Sc.sgdGO)
> gseama_objs <- merged.metrics %>% group_by(technology, metric) %>%
+     do(gseama = test_association(.$value, .$gene))
> wilcox_sets <- gseama_objs %>% group_by(technology, metric) %>%
+     do(.$gseama[[1]]@colData) %>%
+     mutate(Ontology = GO_tab$Ontology[match(ID, GO_tab$go_id)])
> wilcox_genes <- gseama_objs %>% group_by(technology, metric) %>%
+     do(.$gseama[[1]]@geneData)
> # get top sets using each method (is grouped by tech + metric)
> top_sets <- wilcox_sets %>% filter(rank(pvalue) <= 15) %>%
+     arrange(technology, metric, pvalue)
> # arrange p-values for each column
> set_pvalues <- wilcox_sets %>% unite(techmet, technology:metric, sep=".") %>%
+     spread(techmet, pvalue)

> library(ggplot2)
> g_legend <- function(a.gplot) {
```

```
+   tmp <- ggplot_gtable(ggplot_build(a.gplot))
+     leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
+         legend <- tmp$grobs[[leg]]
+         return(legend)
+ }

> library(proto)
> # I wish I could keep the data to plot combined and tidy.
> # However, that loses the ordering, so I have to construct separately
>
> rot <- theme(axis.text.x=element_text(color="black", angle = 90, vjust = 0.5, size=6))
> plotSets = function(sets) {
+     # this function is mostly a hack, kept from original manuscript
+     microarray.dat = CompareY(gseama_objs$gseama[[1]], sets)$data
+     microarray.dat$Technology = "Microarray"
+     RNA.Seq.dat = CompareY(gseama_objs$gseama[[3]], sets)$data
+     RNA.Seq.dat$Technology = "RNA-Seq"
+
+     # fix set names to wrap
+     #microarray.dat$Set = sapply(microarray.dat$Set, function(s) paste(strwrap(s, width=2
+     #RNA.Seq.dat$Set = sapply(RNA.Seq.dat$Set, function(s) paste(strwrap(s, width=20), co
+
+     # get rid of extreme cases in Overall from low depths, which distract
+     microarray.dat = microarray.dat[abs(microarray.dat$y) < 4 | microarray.dat$Set != "Ov
+     RNA.Seq.dat = RNA.Seq.dat[abs(RNA.Seq.dat$y) < 4 | RNA.Seq.dat$Set != "Overall", ]
+
+     (ggplot(microarray.dat, aes(Term, y, fill=Technology)) +
+             geom_sided_violin(aes(side=0)) +
+             geom_sided_violin(aes(side=1), data=RNA.Seq.dat) +
+             rot + coord_flip())
+ }
> num_genes = 12
> ordered_sets = set_pvalues %>% arrange(pmax(MA.estimate, RS.estimate))
> top_IDs <- function(ontology) head((ordered_sets %>% filter(Ontology == ontology))$ID, nu
> BP.p = plotSets(top_IDs("BP"))
> CC.p = plotSets(top_IDs("CC"))
> MF.p = plotSets(top_IDs("MF"))
```

```
> print(BP.p + theme(legend.position="top", axis.text.y=element_text(size=14, face="bold"))
```

Figure 7: The 12 sets that were found most significantly differentially expressed in microarrays and RNA-Seq, within the Biological Process ontology.

```
> print(CC.p + theme(legend.position="top", axis.text.y=element_text(size=14, face="bold"))
```

Figure 8: The 12 sets that were found most significantly differentially expressed in microarrays and RNA-Seq, within the Cellular Compartment ontology.

```
> print(MF.p + theme(legend.position="top", axis.text.y=element_text(size=14, face="bold"))
```

Figure 9: The 12 sets that were found most significantly differentially expressed in microarrays and RNA-Seq, within the Molecular Function ontology.