1 Supplementary Figures

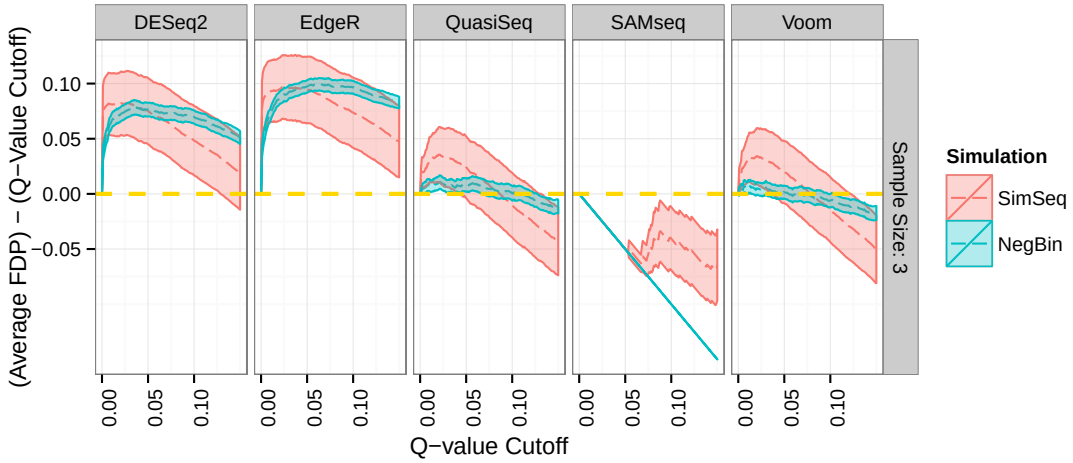

Figure 1: Plot of average false discovery proportion (FDP) minus q-value cutoff for simulations without Cook's distance filtering for simulations based on the Bottomly dataset with a sample size of three. The dashed golden line at 0 represents an average FDP that is exactly equal to its q-value cutoff so that a method that achieves this parity is neither liberal nor conservative with respect to false discovery rate control. The solid lines indicate approximate 95% pointwise confidence intervals for mean FDP minus q-value cutoff. *SAMseq* results for negative binomial simulations fall on the line $y = -x$ because all genes have q-values above 0.15 for all simulation runs; thus, no genes can be declared DE using *SAMseq*.

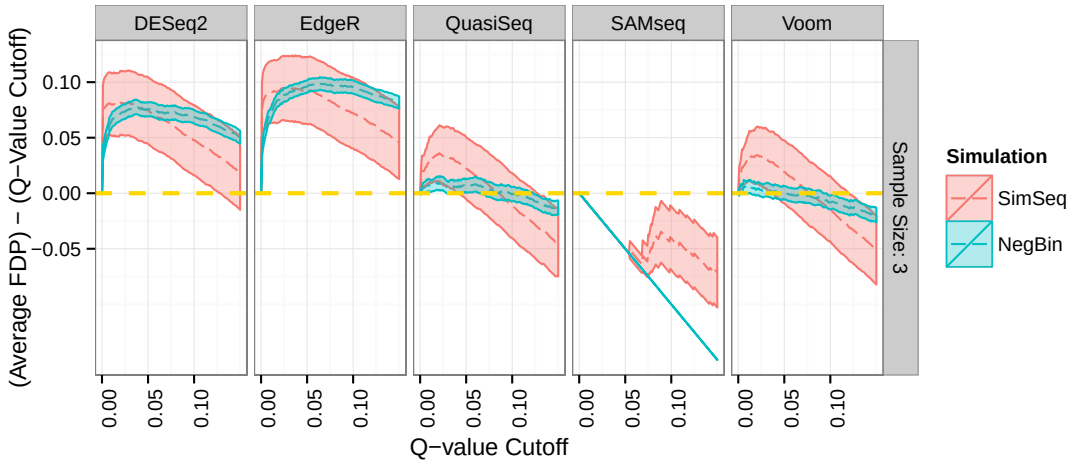

Figure 2: Plot of average false discovery proportion (FDP) minus q-value cutoff for simulations with Cook's distance filtering for simulations based on the Bottomly dataset with a sample size of three. The dashed golden line at 0 represents an average FDP that is exactly equal to its q-value cutoff so that a method that achieves this parity is neither liberal nor conservative with respect to false discovery rate control. The solid lines indicate approximate 95% pointwise confidence intervals for mean FDP minus q-value cutoff. *SAMseq* results for negative binomial simulations fall on the line $y = -x$ because all genes have q-values above 0.15 for all simulation runs; thus, no genes can be declared DE using *SAMseq*.

2 *SimSeq* Example Source Code on the KIRC dataset from The Cancer Genome Atlas

The following section provides example source code for using the *SimSeq* package from the Comprehensive R Archive Network (`http://cran.r-project.org/`). The source RNA-seq dataset used is the Kidney Renal Clear Cell Carcinoma (KIRC) dataset from The Cancer Genome Atlas project (The Cancer Genome Atlas Research Network, 2013) available for download at `https://tcga-data.nci.nih.gov/tcga/`. Example 1 simulates an RNA-seq dataset with 1000 DE genes and 4000 EE genes. Example 2 provides preprocessing steps to save run time for simulation studies with repeated simulations. Example 3 shows how the user can choose the exact list of genes to be EE and DE in a simulated RNA-seq dataset. Example 4 shows how the weights vector can be modified so that genes with log base 2 fold change less than a prespecified threshold have zero probability of being selected to be DE in the simulated RNA-seq dataset. Example 5 simulates an RNA-seq dataset with a three treatment group design.

We describe the three treatment group design of Example 5 in more detail. Because we are subsampling from a source RNA-seq dataset with two treatment groups, only pairwise differences in the three simulated treatment groups are possible. First, we randomly select without replacement the set of all DE genes according to a vector of probability sampling weights and randomly select the remaining EE genes without replacement according to equal probability weights. We then randomly partition the selected DE genes into three DE groups: $\mathcal{G}^{(1)}$, $\mathcal{G}^{(2)}$, and $\mathcal{G}^{(3)}$. We then let $\mathcal{G}^{(1)}$ be the set of genes to be DE in simulated treatment group 1 and EE between simulated treatment groups 2 and 3, let $\mathcal{G}^{(2)}$ be the set of genes to be DE in simulated treatment group 2 and EE between simulated treatment groups 1 and 3, and let $\mathcal{G}^{(3)}$ be the set of genes to be DE in simulated treatment group 3 and EE between simulated treatment groups 1 and 2. We let $\mathcal{G}_0$ be the set of EE genes selected. We then run the *SimSeq* algorithm twice. In the first run of the *SimSeq* algorithm, we let the set of DE genes simulated be $\mathcal{G}^{(2)} \cup \mathcal{G}_0$ and the set of EE genes be $\mathcal{G}^{(1)} \cup \mathcal{G}^{(3)}$ and then discard the first simulated treatment group of the simulated RNA-seq dataset. In the second run of the *SimSeq* algorithm, we switch the order of the treatment groups of the source RNA-seq dataset and let the set of DE genes simulated be $\mathcal{G}^{(2)} \cup \mathcal{G}^{(3)}$ and the set of EE genes be $\mathcal{G}^{(1)} \cup \mathcal{G}^{(0)}$. The two simulated datasets are then combined together to form one RNA-seq dataset with a three treatment group design.

```
### Example 1: Simulate Matrix with 1000 DE genes and 4000 EE genes
require(SimSeq)
require(fdrtool)

data(kidney)
counts <- kidney$counts # Matrix of read counts from KIRC dataset
replic <- kidney$replic # Replic vector indicating paired columns
treatment <- kidney$treatment # Treatment vector indicating Non-Tumor or Tumor columns

nf <- apply(counts, 2, quantile, 0.75)

data.sim <- SimData(counts = counts, replic = replic, treatment = treatment,
                    sort.method = "paired", k.ind = 5, n.genes = 5000, n.diff = 1000,
                    norm.factors = nf)

### Example 2: Calculate weights vector beforehand to save run time in
### repeated simulations
sort.list <- SortData(counts = counts, treatment = treatment, replic = replic,
                      sort.method = "paired", norm.factors = nf)
counts <- sort.list$counts
```

```
replic <- sort.list$replic
treatment <- sort.list$treatment
nf <- sort.list$norm.factors

probs <- CalcPvalWilcox(counts, treatment, sort.method = "paired",
                        sorted = TRUE, norm.factors = nf, exact = FALSE)
weights <- 1 - fdrtool(probs, statistic = "pvalue", plot = FALSE, verbose = FALSE)$lfdr

data.sim <- SimData(counts = counts, replic = replic, treatment = treatment,
                    sort.method = "paired", k.ind = 5, n.genes = 5000, n.diff = 1000,
                    weights = weights, norm.factors = nf)

### Example 3: Specify which genes you want to use in the simulation

# Randomly sample genes or feed in the exact genes you wish to use
genes.diff <- sample(1:nrow(counts), size = 1000, prob = weights)
genes <- c(sample(1:nrow(counts)[-genes.diff], 4000), genes.diff)

data.sim <- SimData(counts = counts, replic = replic, treatment = treatment,
                    sort.method = "paired", k.ind = 5, genes.select = genes,
                    genes.diff = genes.diff, weights = weights, norm.factors = nf)

### Example 4: Simulate matrix with DE genes having log base 2 fold change greater than 1

# add one to counts matrix to avoid infinities when taking logs
tumor.mean <- rowMeans(log2((counts[, treatment == "Tumor"] + 1) %*%
  diag(1/nf[treatment == "Tumor"])))
nontumor.mean <- rowMeans(log2((counts[, treatment == "Non-Tumor"] + 1) %*%
  diag(1/nf[treatment == "Non-Tumor"])))

lfc <- tumor.mean - nontumor.mean
weights.zero <- abs(lfc) < 1
weights[weights.zero] <- 0

data.sim <- SimData(counts = counts, replic = replic, treatment = treatment,
                    sort.method = "paired", k.ind = 5, n.genes = 5000, n.diff = 1000,
                    weights = weights, norm.factors = nf)

### Example 5: Simulate three treatment groups:
### 3 Different types of Differential Expression Allowed
### First Group Diff, Second and Third group Equal
### Second Group Diff, First and Third group Equal
### Third Group Diff, First and Second group Equal

k <- 5 # Sample Size in Each treatment group

### Sample DE genes beforehand
N <- nrow(counts)
genes.de <- sample(1:N, size = 1000, prob = weights) # Sample all DE genes
DE1 <- genes.de[1:333] # Sample DE genes with first trt diff
DE2 <- genes.de[334:666] # Sample DE genes with sec trt diff
DE3 <- genes.de[667:1000] # Sample DE genes with third trt diff
EE <- sample( (1:N)[-genes.de], size = 4000) #Sample EE genes
```

```
genes.tot <- c(EE, genes.de)
genes.de1 <- union(DE2, EE) #Assign DE genes for first sim
genes.de2 <- union(DE2, DE3) #Assign DE genes for second sim

data.sim1 <- SimData(counts = counts, replic = replic, treatment = treatment,
                     sort.method = "paired", k.ind = k, genes.select = genes.tot,
                     genes.diff = genes.de1, weights = weights, norm.factors = nf)

#remove pairs of columns used in first simulation
cols.rm <- c(data.sim1$col[1:(2*k)], data.sim1$col[1:(2*k)] + 1)
counts.new <- counts[, -cols.rm]
nf.new <- nf[-cols.rm]
replic.new <- replic[-cols.rm]
treatment.new <- treatment[-cols.rm]

### Set switch.trt = TRUE for second sim
data.sim2 <- SimData(counts = counts.new, replic = replic.new, treatment = treatment.new,
                     sort.method = "paired", k.ind = k, genes.select = genes.tot,
                     genes.diff = genes.de2, weights = weights, norm.factors = nf.new,
                     switch.trt = TRUE)

### Remove first k.ind entries from first sim and combine two count matrices
counts.sim <- cbind(data.sim1$counts[, -(1:k)],  data.sim2$counts)

### treatment group levels for simulated matrix
trt.grp <- rep(NA, 5000)
trt.grp[is.element(data.sim1$genes.subset, DE1)] <- "DE_First_Trt"
trt.grp[is.element(data.sim1$genes.subset, DE2)] <- "DE_Second_Trt"
trt.grp[is.element(data.sim1$genes.subset, DE3)] <- "DE_Third_Trt"
trt.grp[is.element(data.sim1$genes.subset, EE)] <- "EE"
```