Title

Visualization methods for RNA-sequencing data analysis

Authors

Lindsay Rutter[1] and Dianne Cook[2]

[1]Bioinformatics and Computational Biology Program, Iowa State University

[2]Department of Department of Econometrics and Business Statistics, Monash University

Abstract

It was initially claimed that RNA-seq produced unbiased data that did not require sophisticated normalization. However, studies have since revealed that RNA-seq data is biased and that accurate detection of differentially expressed genes is not a trivial task. In light of these findings, researchers should analyze RNA-seq data like they would any other biased multivariate data. The most effective approach to modern data analysis is to iterate between models and visuals, and to enhance the appropriateness of models based on feedback from visuals. Unfortunately, researchers do not often use models and visuals in a complimentary fashion when analyzing RNA-seq data. Here, we use real RNA-seq data to show that our visualization tools can detect normalization problems, DEG designation problems, and common errors in RNA-seq analysis. We also show that our tools can identify genes of interest that cannot be obtained by models. In this paper, we do not propose that users radically change their approach to RNA-seq analysis. Instead, we propose that users simply modify their approach to RNA-seq analysis by assessing the sensibility of their models with multivariate statistical graphics.

Keywords

Data visualization; Exploratory data analysis; Interactive graphics; RNA-sequencing.

1. Introduction

RNA-sequencing (RNA-seq) uses next-generation sequencing (NGS) to estimate the quantity of RNA in biological samples at given timepoints. In recent years, decreasing cost and increasing throughput has rendered RNA-seq an attractive alternative to transcriptome profiling. Prior to RNA-seq, gene expression studies were performed with microarray techniques, which required prior knowledge of reference sequences. RNA-seq does not have this limitation, and has enabled a new range of applications such as transcriptome de novo assembly (Robertson et al., 2010) and detection of alternative splicing processes (Anders, Reyes, and Huber, 2012; Pan et al., 2008). Coupled with its high resolution and sensitivity, RNA-seq will likely revolutionize our understanding of the intricacies of eukaryotic transcriptomes (Wang, Gerstein, and Snyder, 2009; Zhao et al., 2014).

RNA-seq data is multivariate data, and its basic form is a matrix containing mapped read counts for n rows of genes and p columns of samples. These mapped read counts provide estimations of the gene expression levels across samples. Researchers typically conduct RNA-seq studies to identify differentially expressed genes (DEGs) between treatment groups. In most popular RNA-seq analysis packages, this objective is approached with models, such as

the negative binomial model (Anders and Huber, 2010; Trapnell et al., 2013; Trapnell et al., 2012; Robinson et al., 2010) and linear regression models (Law et al., 2014).

Initially, it was widely claimed that RNA-seq produced unbiased data that did not require sophisticated normalization (Wang et al., 2009; Morin et al., 2008; Marioni et al., 2008). However, numerous studies have since revealed that RNA-seq data is replete with biases and that accurate detection of DEGs is not a negligible task. Problems that complicate the analysis of RNA-seq data include nucleotide and read-position biases (Hansen et al., 2010), biases related to gene lengths and sequencing depths (Oshlack, Robinson, and Young, 2010; Robinson and Oshlack, 2010), biases introduced during library preparation (McIntyre et al., 2011), biases pertaining to the number of replications (Schurch et al., 2016), biases derived from overlapping sense-antisense transcripts and gene isoforms (Trapnell et al., 2013), and the confounding combination of technical and biological variability (Bullard et al., 2010).

In light of these complications, researchers should analyze RNA-seq data like they would any other biased multivariate data. Simply applying models to such data is problematic because models hold assumptions that they alone cannot call into question. Fortunately, data visualization enables researchers to see patterns and problems they may not otherwise detect with traditional modeling. As a result, the most effective approach to data analysis is to iterate between models and visuals, and enhance the appropriateness of applied models based on feedback from visuals (Shneiderman, 2002). With RNA-seq data, we primarily want to compare the variability between replicates and between treatment groups. This is visually best achieved by drawing the mapped read count distributions across all genes and samples. Unfortunately, the few plotting tools offered in popular RNA-seq packages do not allow users to effectively view their data in this manner.

In this paper, we strive to remedy this problem by publishing new and effective RNA-seq plotting tools. We use real RNA-seq data to show that our tools can detect normalization problems, DEG designation problems, and common errors in the analysis pipeline. We also show that our tools can identify genes of interest that cannot otherwise be obtained by models. We emphasize that interactive graphics should be an indisposable component of modern RNA-seq analysis: Researchers should be able to quickly flip through plots of genes that appear promising or problematic, and link between plots to swiftly obtain various perspectives of their data. Here, we do not propose that users drastically change their approach to RNA-seq analysis. Instead, we propose that users simply modify their approach to RNA-seq analysis by assessing the sensibility of their models with multivariate graphical tools, namely with parallel coordinate plots, scatterplot matrices, and replicate point plots.

2. Parallel Coordinate Plots

Parallel coordinate plots are essential to visually verify the relationships between variables in multivariate data. A parallel coordinate plot draws each row (gene) as a line. Connections between samples with positive correlations are flat, and connections between samples with negative correlations are crossed. The ideal dataset has more variability between treatments than between replicates. Researchers can quickly confirm this with a parallel coordinate plot: There should be flat connections between replicates but crossed connections between treatments.

There are several packages within the Bioconductor software that provide graphics for RNA-seq data analysis (Huber et al., 2015). Two of the most common graphic techniques are

side-by-side boxplots and Multidimensional Scaling (MDS) plots (Love, Huber, and Anders, 2014; Risso et al., 2011; Robinson et al., 2010; Su et al., 2016; Ritchie et al., 2015; Marini, 2017). Unfortunately, these plots can hide problems that still exist in the data even after normalization and that could be better detected with parallel coordinate plots.

Figure 1 exemplifies this problem for two simulated datasets, one displayed on the left half and the other displayed on the right half of the figure. Each dataset contains two treatment groups (A and B) with three replicate samples. We cannot detect any notable differences between the left and right datasets from the side-by-side boxplots at the top of the figure as they both show fairly consistent five number summaries across their six samples. Likewise, we cannot detect notable differences between the datasets from the MDS plots in the middle of the figure as they both suggest that the datasets are clustered by the two treatment groups, although the first replicate from treatment A appears as an outlier in the right MDS plot.

[Figure 1 about here.]

Despite this, we immediately see from the parallel coordinate plots on the bottom of the figure that the left and right datasets have an important difference. The left dataset has consistent (level) lines between replicates and inconsistent (crossed) lines between treatment groups. This suggests that some of the genes (lines) have consistently low values for treatment group A and consistently high values for treatment group B, while some genes have the opposite phenomenon. As a result, these plotted genes are likely candidates for differential expression. In contrast, the right dataset does not possess this ideal structure and suggests that the genes may not be candidates for differential expression. We could not see this important distinction from the side-by-side boxplots and MDS plots because they simply provide summaries about the data at the sample resolution, while the parallel coordinate plot shows the sample connections for each of the 50 genes.

3. Scatterplot matrices

3.1 *Overview of scatterplot matrices*

A scatterplot matrix is another effective multivariate visualization tool that plots the mapped read count distributions across all genes and samples. Specifically, it represents each row (gene) as a point in each scatterplot. With this method, users can quickly discover unexpected patterns, recognize geometric shapes, and assess the structure and association between multiple variables in a manner that is different from most common practices.

The ideal dataset will have larger variability between treatment groups than between replicates. As Figure 2 shows, researchers can quickly confirm this with a scatterplot matrix. Within each scatterplot, most genes should fall along the x=y line (in red) as we expect only a small proportion of them to show differential expression between samples. However, a fraction of the genes should have lower variability between replicates than between treatments, and so we should expect the spread of the scatterplot points to fall more closely along the x=y relationship between replicates than between treatments. Indeed, in Figure 2, we created a scatterplot matrix for a public RNA-seq dataset that contains three replicates for two developmental stages of soybean cotyledon (S1 and S2) (Brown and Kudson, 2015). We can immediately verify that the nine scatterplots between treatment pairs (in the bottom-left corner of the matrix) have more spread around the x=y line than the six scatterplots between replicate pairs.

[Figure 2 about here.]

After confirming this expected trend, users can use the scatterplot matrix to focus on subsets of genes: Outlier genes that deviate from the x=y line in replicate scatterplots might be problematic, whereas outlier genes that deviate from the x=y line in treatment scatterplots might be DEGs. In order to achieve this functionality, the plots must be rendered interactive.

Notice that each gene in our data is plotted once in each of the 15 scatterplots. With 73,320 genes in our data, more than one million points must be plotted. Rendering all points interactive would slow down the interactive capabilities of the plot. To solve this, we can tailor the geometric object of the scatterplots to be hexagon bins rather than points. This dramatically reduces the number of geometric objects to be plotted, and increases the interactivity speed.

The reader can visit https://rnaseqvisualization.shinyapps.io/scatmat to access the interactive version of Figure 2. Readers can read the "About" Tab to fully understand how to use the application. Essentially, the user can hover over a hexagon bin to see how many genes it contains. When the user clicks on a hexagon bin, the names of the genes are listed and superimposed as orange points across all scatterplots. The genes are also linked to a second plot that superimposes them as parallel coordinate lines on a side-by-side boxplot of all gene counts. This interactivity and linking allows users to quickly examine genes of interest from multiple perspectives superimposed onto the summary of all genes in the dataset.

The scatterplot matrix can also be used after DEG calls to quickly examine DEGs obtained from a given model. As shown in Figure 3, the DEGs can be superimposed as orange points onto the scatterplot matrix. We expect for DEGs to fall along the x=y line for replicates and deviate from the x=y line between treatment groups, as is confirmed in Figure 3. As a side note, we can also link these DEGs as parallel coordinate lines on a side-by-side boxplot like in Figure 3 to confirm the expected pattern of differential expression from a second viewpoint. If we do not observe what should be expected of DEGs, then the DEG calls from the model need to be scrutinized further.

[Figure 3 about here.]

3.2 *Assessing normalization with scatterplot matrices*

There is still substantial discussion about the normalization of RNA-seq data, and the scatterplot matrix can be used to understand and assess various algorithms. To exemplify this point, we will use a publicly-available RNA-seq dataset on Saccharomyces cerevisiae (yeast) grown in YP-Glucose (YPD) (Risso, 2011). The data contained four cultures from independent libraries that were sequenced using two library preparation protocols and either one or two lanes in a total of three flow-cells (see Table 1 which is from Risso, 2011). This experimental design allowed researchers to examine various levels and combinations of technical effects (library preparation and protocol and flow cell) and biological effects (culture).

[Table 1 about here.]

The four cultures (Y1, Y2, Y4, and Y7) were treated as biological replicates for which differential expression was not expected. Hence, the authors could establish a false positive rate in relation to the number of DEGs called between these groups. They then demonstrated that within-lane regression alone was insufficient in effectively removing biases. Instead,

aggressive corrections for both within-lane (GC-content and gene length) and between-lane (count distribution and sequencing depth) biases were needed to effectively reduce the false-positive rate of differential expression calls.

Figure 4A shows the scatterplot matrix of the read counts from the Y1 and Y4 treatments after within-lane normalization. As we stated earlier, we expect most genes to show similar expression between samples, except for the handful that are differentially expressed. However, it is immediately clear that the data still was not sufficiently normalized as the distribution of genes is not centered around the x=y lines. In contrast, Figure 4B shows the scatterplot matrix of the read counts from the Y1 and Y4 treatments after *both* within-lane and between-lane normalization, as was recommended by the authors due to its reduced false-positive rate. Indeed, the scatterplot matrix now follows the expected structure with most genes falling along the x=y line with thicker deviations from it between treatment groups than between replicate groups.

Additionally, we can also confirm from Figure 4B that the read counts fall closer to the x=y line between the Y4 replicates (bottom-right scatterplot) than between the Y1 replicates (top-left scatterplot). This is expected because the Y1 replicates had additional technical variability as they used two different flow cells, whereas the Y4 replicates were from the same flow cell. As such, the scatterplot matrix can also be used to quickly inspect patterns of biological and technical variability in the dataset.

[Figure 4 about here.]

3.3 *Checking for common errors with scatterplot matrices*

Irreproducibility is prevalent in high-throughput biological studies. A study in Nature Genetics surveyed eighteen published microarray expression analyses and reported that only two were exactly reproducible (Ioannidis et al., 2009). The extent of the problem has spawned a field called "forensic bioinformatics" whereby researchers attempt to reverse-engineer reported results back into the raw datasets simply to derive the methodologies used in published studies (Baggerly and Coombes, 2009).

Even though irreproducibility is merely cumbersome when it masks methods, it is unquestionably hazardous when it masks errors. With regards to personalized medicine, for example, obscured errors may cause well-intentioned researchers to present evidence for drugs that are ineffective or even harmful to patients (Baggerly and Coombes, 2009). Forensic bioinformaticians who have actively investigated common errors in high-throughput biological studies have concluded that the largeness of the data itself may hinder our ability to detect errors (Baggerly and Coombes, 2009). They also discovered that the most common errors are simple errors, such as mixing up sample labels (Baggerly and Coombes, 2009). Collectively, these findings suggest that simple errors can be difficult to detect using common practices in high-throughput studies.

Fortunately, scatterplot matrices are a simple tool to check for common errors like sample mislabeling. Figure 5 shows the resulting scatterplot matrix after we deliberately swapped the labels of the third replicate of the first treatment group (S1.3) with the first replicate of the second treatment group (S2.1) in the previously-mentioned cotyledon dataset. We can immediately see that, as expected, there are nine scatterplots with thicker distributions around the x=y line and six scatterplots with thinner distributions around the x=y line. However, we notice that a subset of these thick and thin scatterplots appear outside of

their expected locations given the expected variability between treatments versus replicates. Rearranging the columns of the two samples that appear suspicious in Figure 5 would indeed lead us back to the clean-looking scatterplot matrix we saw in Figure 2. We cannot detect this mislabeling problem as clearly and as convincingly with traditional plots, as can be verified with this dataset by comparing the boxplots and MDS plots before sample switching (left side of Figure 6) and after sample switching (right side of Figure 6).

[Figure 5 about here.]

[Figure 6 about here.]

3.4 *Finding unexpected patterns in scatterplot matrices*

Most popular RNA-seq plotting tools display summaries about the read counts, such as fold change summaries, principal component summaries, five number summaries, and dispersion summaries. In contrast to this trend, scatterplot matrices display the non-summarized read counts for all genes. This trait allows for geometric shapes and patterns relevant to the read count distribution to be readily visible in the scatterplot matrix.

An example of how geometric shapes in the scatterplot matrix can provide applicable information to researchers is shown in Figure 7. The dataset comes from an RNA-seq study conducted to identify gene expression responses in soybean leaves after exposure to iron-sufficient and iron-deficient soil conditions (Moran Lauter et al., 2014). After normalizing the data, we see the expected pattern of a scatterplot matrix in Figure 7, with more variation around the x=y line between treatments than between replicates.

However, one streak structure in the bottom right scatterplot stands out. A small subset of transcripts between replicates of the iron-sufficient group sharply deviates from the x=y line. By interacting with the plot, we determined the identification of the five transcripts that deviated the most from the expected pattern, and searched for their putative functions. We discovered that these transcripts are reportedly involved in biotic and abiotic stress responses as well as the production of superoxides to combat microbial infections.

Discussion with the authors of the study revealed that a lab biologist documented accidentally tearing a leaf on one of these replicates. Hence, these transcripts that markedly deviate from the expected pattern in an otherwise well-controlled experiment might represent those that changed expression in relation to this incident. Even though the main motivation of the study had been to investigate the molecular underpinnings of iron metabolism, through our exploratory data analysis, we can derive a post-hoc hypothesis about what genes tentatively respond to leaf cutting. Of course, this would only serve as a hypothesis generator; conventional genetic studies and additional evidence would be needed to confirm any possible role these genes have on this biological activity. Regardless, we would not have observed this interesting structure or derived this post-hoc hypothesis from any models.

[Figure 7 about here.]

4. Replicate point plots

We demonstrated how to view differentially expressed genes onto the Cartesian coordinates of the scatterplot matrix in Figure 3. Unfortunately, this figure becomes limited when we investigate treatment groups that contain a large number of replicates because we then have too many small scatterplots for it to remain an effective visualization tool. Moreover,

researchers could benefit from additional plotting tools that allow them to quickly verify individual differentially expressed genes returned from a model. As a result, we are developing a plot that allows users to visualize *one* differentially expressed gene of interest onto the Cartesian coordinates of *one* scatterplot matrix.

The "replicate line plot" was developed by a group of researchers who demonstrated it could detect model scaling problems in microarray data (Cook et al., 2007). Unfortunately, this plot is only applicable on datasets where treatment groups contain exactly two replicates. The plot we now introduce is an extension of the "replicate line plot" that can be applied to datasets with two or more replicates. We call this new plot a "replicate point plot".

In the replicate point plot, each gene is plotted once for each possible combination of sample pairs between treatment groups. For example, there are nine ways to pair a sample from one treatment group with a sample from the other treatment group in the cotyledon dataset (S1.1 and S2.1, S1.1 and S2.2, S1.1 and S2.3, S1.2 and S2.1, S1.2 and S2.2, S1.2 and S2.3, and S1.3 and S2.1, S1.3 and S2.2, and S1.3 and S2.3). Hence, a given gene from this dataset is plotted as nine points in the replicate point plot. With 73,320 genes in this dataset, we would need to plot 659,880 points. This would reduce the speed of interactive functionality as well as cause overplotting problems. As a result, we again use hexagon bins to summarize this massive information (Figure 8 shows four example replicate point plots).

Once the background of hexagons has been drawn to give us a sense of the distribution of all treatment pair combinations for all genes, the user can superimpose the nine points of one gene of interest (colored orange in Figure 8). Subplots A and B of Figure 8 show the replicate point plots for two example genes from the aforementioned cotyledon dataset, and subplots C and D of Figure 8 show the replicate point plots for two example genes from the cotyledon dataset after samples S1.3 and S2.1 have been swapped. Note that, in the examples in Figure 8, the read counts of treatment pair combinations sometimes overlap, resulting in the appearance of less than nine orange points in a given replicate point plot.

[Figure 8 about here.]

The interactive replicate point plot for the cotyledon data (Figure 8A and B) is available at https://rnaseqvisualization.shinyapps.io/repPoint, and the interactive replicate point plot for the cotyledon data after the samples were swapped (Figure 8C and D) is available at https://rnaseqvisualization.shinyapps.io/repPointSwitch. As can be verified in the interactive version of the replicate point plot, users are provided several input fields that tailor the plot functionality. For instance, the user may wish to quickly scroll through differentially expressed genes one by one in the order of lowest to highest FDR values. Please read the "About" tab in the interactive links for more information.

Figure 8A and B show the static results of applying such settings to the cotyledon data. Each of these plots shows an example gene drawn from the ten genes with the lowest FDR values. For the case of Figure 8A, the nine points of the gene are superimposed in a manner we would expect from a differently expressed gene: They are located far from the $x=y$ line (difference between treatments) and are close to each other (similarity between replicates). The points for the gene represented in Figure 8B are also far from the $x=y$ line (difference between treatments) and are so close in proximity that they almost entirely overlay each other (precise similarity between replicates).

In contrast, Figure 8C and D show the static results of applying such settings to the cotyledon data after swapping the S1.3 and S2.1 samples. Each of these plots shows an

example gene drawn from the ten genes with the lowest FDR values. We expect this swap to cause the data to now produce unreliable results. Indeed, we can see this problem in Figure 8C and D, as they contain orange points that are not far from the x=y line and are not close to each other. This indicates that the genes with the lowest FDR values do not show the pattern we expect of differentially expressed genes. In this case, the results of the model match those from our visual exploratory analysis: As can be seen in the interactive graphic, none of the FDR values reached significance for the swapped cotyledon data.

Interestingly, we reach this same conclusion using parallel coordinate plots. In Figure 9A, we took the 100 genes with the lowest FDR values from the cotyledon data and ran them through a hierarchical clustering algorithm with six clusters. Rather than plotting all 100 parallel coordinates onto a common background boxplot, we can reduce overplotting issues and see patterns more easily by separating with clustering. Nonetheless, we see that the six clusters all represent one of two expected patterns for differentially expressed genes: Each cluster either shows consistent high read counts for treatment S1 and consistent low read counts for treatment S2, or vice versa.

In contrast, Figure 9B shows the result of taking the 100 genes with the lowest FDR values from the *swapped* cotyledon data and running them through a hierarchical clustering algorithm with six clusters. Here, we see that the expression patterns do not fall into the two previously-mentioned expected patterns of differentially expressed genes. Instead, the difference between the treatment groups is suspiciously small and the difference between the replicates is suspiciously large. In addition, we may suspect that samples S1.3 and S2.1 were swapped: For example, the cluster with the largest number of genes (n=43) shows a pattern that would look more like differential gene expression if these two samples were reversed.

[Figure 9 about here.]

5. Discussion

We have used real data to demonstrate that scatterplot matrices, parallel coordinate plots, and replicate point plots can help users check for normalization problems, catch common errors in the analysis pipeline, and confirm that the variation between replicates and treatments is as expected. We also show that these visual tools are useful for researchers to quickly investigate the list of differentially expressed genes that come out of a model and ensure which ones make sense. Similarly, in the case that no differentially expressed genes were returned, users can quickly investigate the list of genes with the lowest FDR values to ensure that the negative diagnosis of the model seems accurate. In addition, we demonstrated that our simple visualization tools can allow researchers to discover genes of interest that could not be obtained with models.

At this point, we aim to continue speeding up the interactive graphics, creating better color legends, extending the customisability of the graphics, and combining them into an easy-to-use R package for users. We plan to develop a clear and intuitive vignette for users to ensure them that rather than suggesting that they radically change their approach to RNA-seq analysis, we simply suggest that they incorporate these visual tools during their usual analysis pipeline to quickly check that the model assumptions and results are reasonable. We plan to include several real RNA-seq datasets in our R package and then demonstrate the syntax and value of our methods on these datasets in the vignette. Before publishing our R

package, we will gather more examples using real RNA-seq data to show that these graphics can discover problems with or confirm proper use of popular RNA-seq analysis models.

References

Anders, S. and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biology* **11,** R106.

Anders, S., Reyes, A., and Huber, W. (2012). Detecting differential usage of exons from RNA-seq data. *Genome Research* **22,** 20082017.

Baggerly, K.A. and Coombes, K.R. (2009). Deriving chemosensitivity from cell lines: Forensic bioinformatics and reproducible research in high-throughput biology. *The Annals of Applied Statistics* **3,** 13091334.

Brown, A.V. and Hudson, K.A. (2015). Developmental profiling of gene expression in soybean trifoliate leaves and cotyledons. *BMC Plant Biology* **15,** 169.

Bullard, J. H., Purdom, E., Hansen, K. D., and Dudoit, S. (2010). Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics* **11,** 94.

Cook, D., Hofmann, H., Lee, E.-K., Yang, H., Nikolau, B., and Wurtele, E. (2007). Exploring gene expression data, using plots. *Journal of Data Science* **5,** 151-182.

Hansen, K.D., Brenner, S.E., and Dudoit, S. (2010). Biases in Illumina transcriptome sequencing caused by random hexamer priming. *Nucleic Acids Research* **38,** e131.

Huber, W., Carey, V.J., Gentleman, R,, Anders, S,, Carlson, M., and Carvalho, B.S. (2015). Orchestrating high-throughput genomic analysis with Bioconductor. *Nature Methods* **12,** 115-121.

Ioannidis, J.P., Allison, D.B., Ball, C.A., Coulibaly, I., Cui, X., Culhane, A.C., Falchi, M., Furlanello, C., Game, L., Jurman, G., Mangion, J., Mehta, T., Nitzberg, M., Page, G.P., Petretto, E. and van Noort, V. (2009). Repeatability of published microarray gene expression analyses. *Nature Genetics* **41,** 149155.

Law, C.W., Chen, Y., Shi, W., and Smyth, G.K. (2014). voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biology* **15,** R29.

Love, M.I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology* **15,** 550.

Marini, F. (2017). pcaExplorer: Interactive visualization of RNA-seq data using a principal components approach. GitHub, Inc. https://github.com/federicomarini/pcaExplorer (accessed October 7, 2017).

Marioni, J.C., Mason, C.E., Mane, S.M., Stephens, M., and Gilad, Y. (2008). RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research* **18,** 15091517.

McIntyre, L.M., Lopiano, K.K., Morse, A.M., Amin, V., Oberg, A.L., Young, L.J., et al. (2011). RNAseq: Technical variability and sampling. *BMC Genomics* **12,** 293.

Moran Lauter, A.N., Peiffer, G.A., Yin, T., Whitham, S.A., Cook, D., and Shoemaker, R.C. (2014). Identification of candidate genes involved in early iron deficiency chlorosis signaling in soybean (glycine max) roots and leaves. *BMC Genomics* **15,** 125.

Morin, R., Bainbridge, M., Fejes, A., Hirst, M., Krzywinski, M., Pugh, T., et al. (2008). Profiling the HeLa S3 transcriptome using randomly primed cDNA and massively parallel short-read sequencing. *Biotechniques* **45,** 8194.

Oshlack, A., Robinson, M.D., and Young, M.D. (2010). From RNA-seq reads to differential expression results. *Genome Biology* **11,** 220.

Pan, Q., Shai, O., Lee, L. J., Frey, B. J., and Blencowe, B. J. (2008). Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. *Nature Genetics* **40,** 14131415.

Risso, D., Schwartz, K., Sherlock, G., Dudoit, S. (2011). GC-Content normalization for RNA-Seq data. *BMC Bioinformatics* **12,** 480.

Ritchie, M.E., Phipson, B., Wu, D., Hu, Y., Law, C.W., Shi, W., and Smyth, G.K. (2015). Limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research* **43,** e47.

Robertson, G., Schein, J., Chiu, R., Corbett, R., Field, M., Jackman, S. D., et al. (2010). De novo assembly and analysis of RNA-seq data. *Nature Methods* **7,** 909912.

Robinson, M.D. and Oshlack, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology* **11,** R25.

Robinson, M.D., McCarthy, D.J., and Smyth, G.K. (2010). edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26,** 139140.

Shneiderman, B. (2002). Inventing discovery tools: Combining information visualization with data mining. *Information Visualization* **1,** 5-12.

Schurch, N.J., Schofield, P., Gierliski, M., Cole, C., Sherstnev, A., Singh, V., et al. (2016). How many biological replicates are needed in an RNA-seq experiment and which differential expression tool should you use? *RNA* **22,** 839851.

Shu, S., Ritchie, M.E., Law, C., and Lee, S. (2016). Glimma: Interactive HTML graphics. GitHub, Inc. https://github.com/Shians/Glimma/ (accessed October 7, 2017).

Trapnell, C., Hendrickson, D.G., Sauvageau, M., Goff, L., Rinn, J.L., and Pachter, L. (2013). Differential analysis of gene regulation at transcript resolution with RNA-seq. *Nature Biotechnology* **31,** 4653.

Trapnell, C., Roberts, A., Goff, L., Pertea, G., Kim, D., and Kelley D.R. (2012). Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature Protocols* **7,** 562578.

Wang, Z., Gerstein, M., and Snyder, M. (2009). RNA-Seq: A revolutionary tool for transcriptomics. *Nature Reviews Genetics* **10,** 5763.

Zhao, S., Fung-Leung, W.-P., Bittner, A., Ngo, K., and Liu, X. (2014). Comparison of RNA-Seq and microarray in transcriptome profiling of activated T cells. *PLoS ONE* **9,** e78644.

Zhao, S., Fung-Leung, W.-P., Bittner, A., Ngo, K., and Liu, X. (2014). Comparison of RNA-Seq and microarray in transcriptome profiling of activated T cells. *PLoS ONE* **9,** e78644.

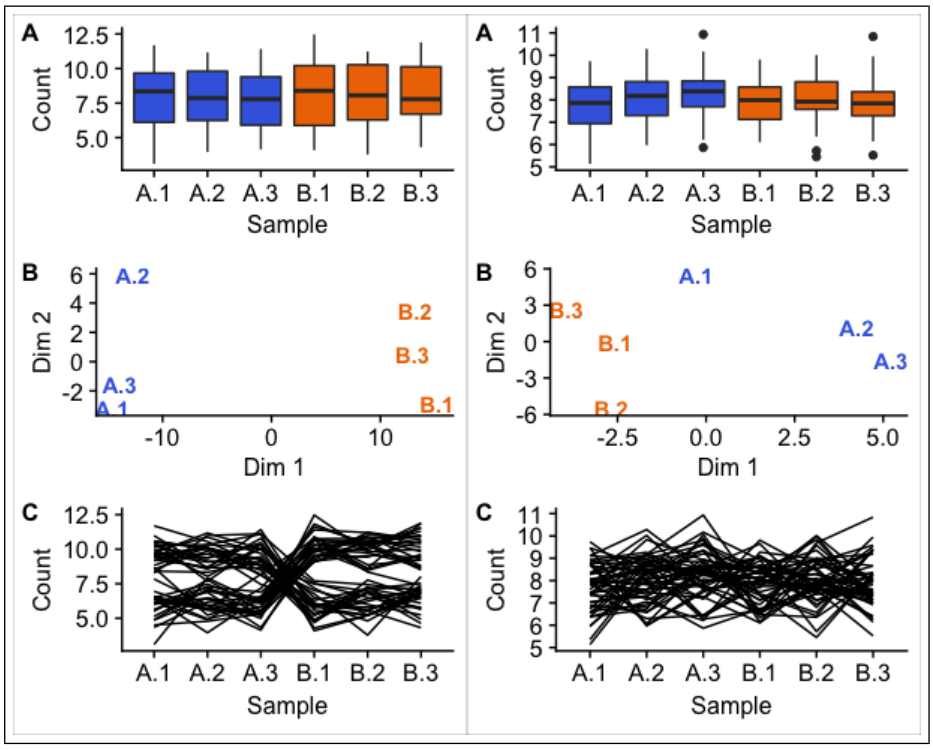

Figure 1: One simulated dataset is shown on the left half and another simulated dataset is shown on the right half of the figure. We do not see crucial distinctions between the left and right datasets when we compare their boxplots (A subplots) and MDS plots (B subplots). However, their parallel coordinate plots (C subplots) show a critical difference between their structures. Namely, the left dataset is composed of genes with small replicate variation and large treatment group variation (suggesting DEGs), while the right dataset is composed of genes with similar variation between replicates and treatment groups (not suggesting DEGs).

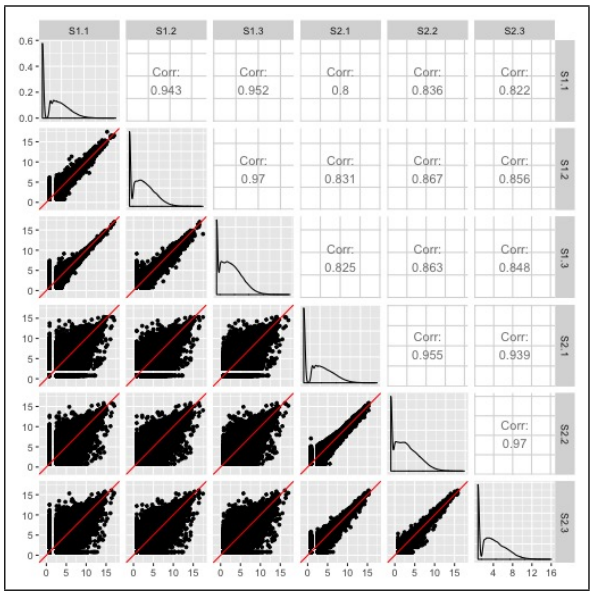

Figure 2: Example of the expected structure of an RNA-seq dataset. Within a given scatterplot, most genes (points) should fall along the x=y line. We should see genes deviate more strongly from the x=y line in treatment scatterplots than in replicate scatterplots.

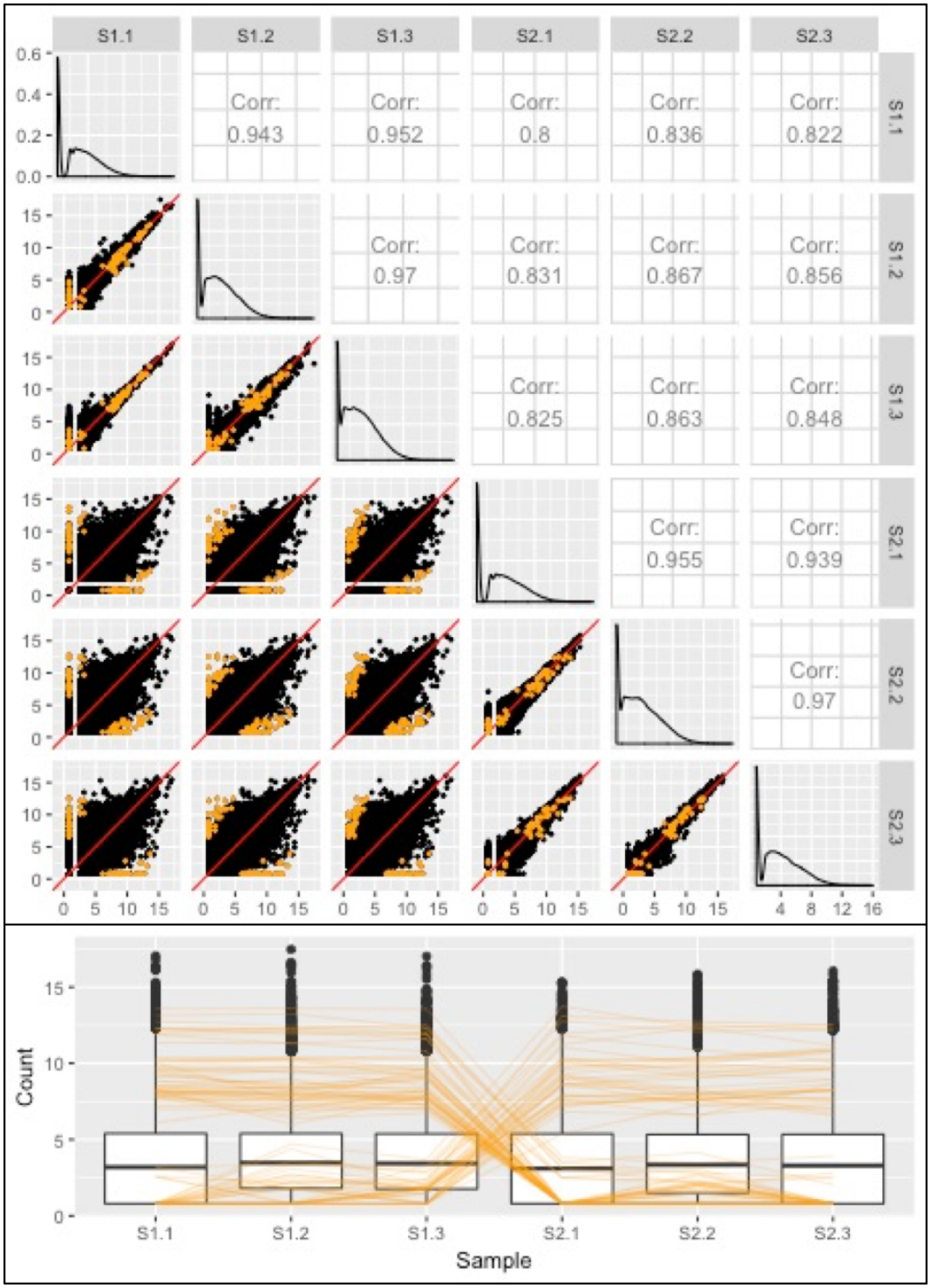

Figure 3: Example of the expected structure of DEG calls (in orange) from an RNA-seq dataset. In the scatterplot matrix, DEGs should fall along the $x{=}y$ line for replicates and deviate from it for treatments. In the parallel coordinate plot, DEGs should show levelness between replicates and crosses between treatments. These two plotting types can be linked to quickly provide users multiple perspectives of their DEG calls.

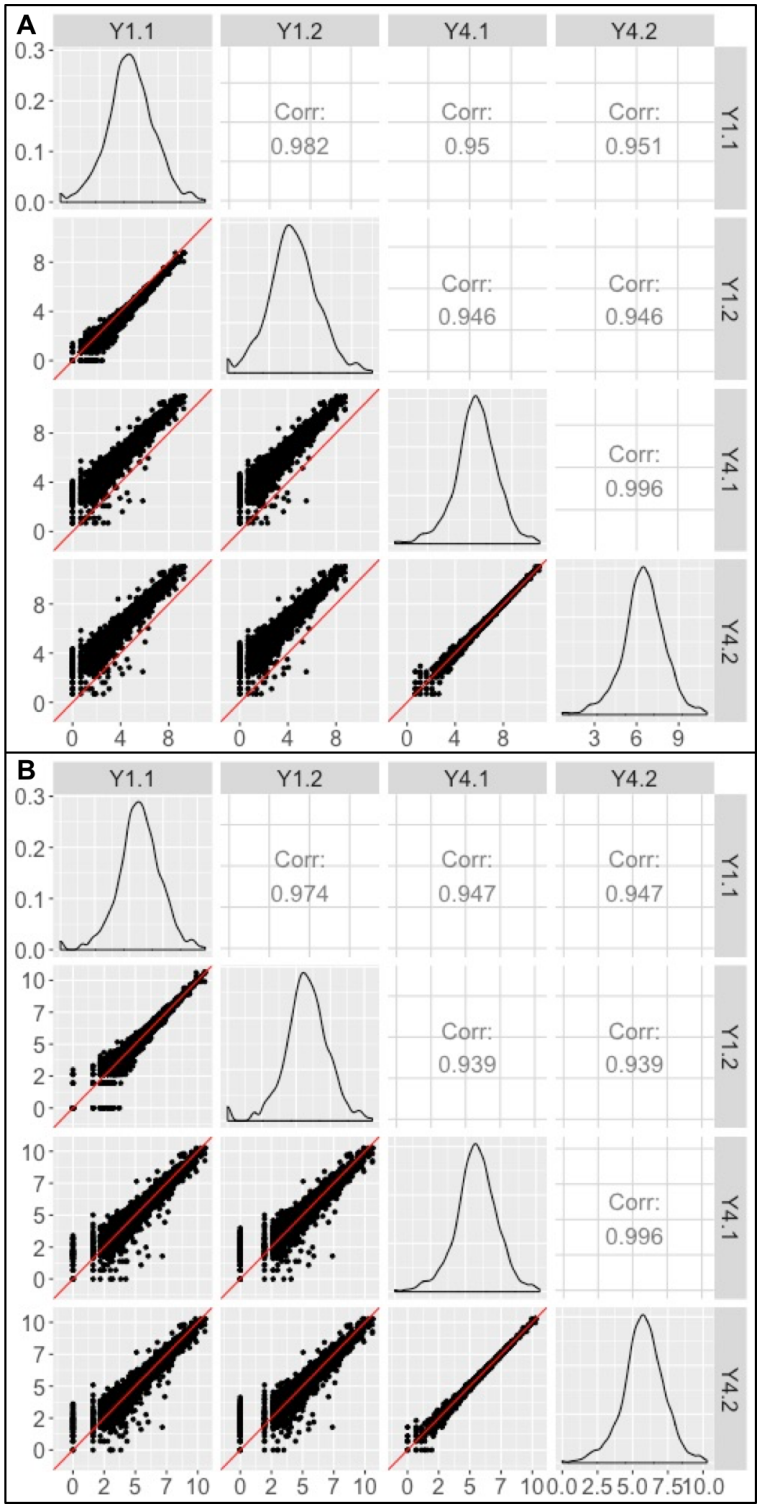

Figure 4: The collective deviation of genes from the x=y line instantly reveals that the RNA-seq dataset was not thoroughly normalized using within-lane normalization (subplot A). However, within-lane normalization followed by between-lane normalization sufficiently normalized the data (subplot B). The authors who developed these normalization methods showed that the later approach generated a lower false-positive DEG call rate in this dataset.

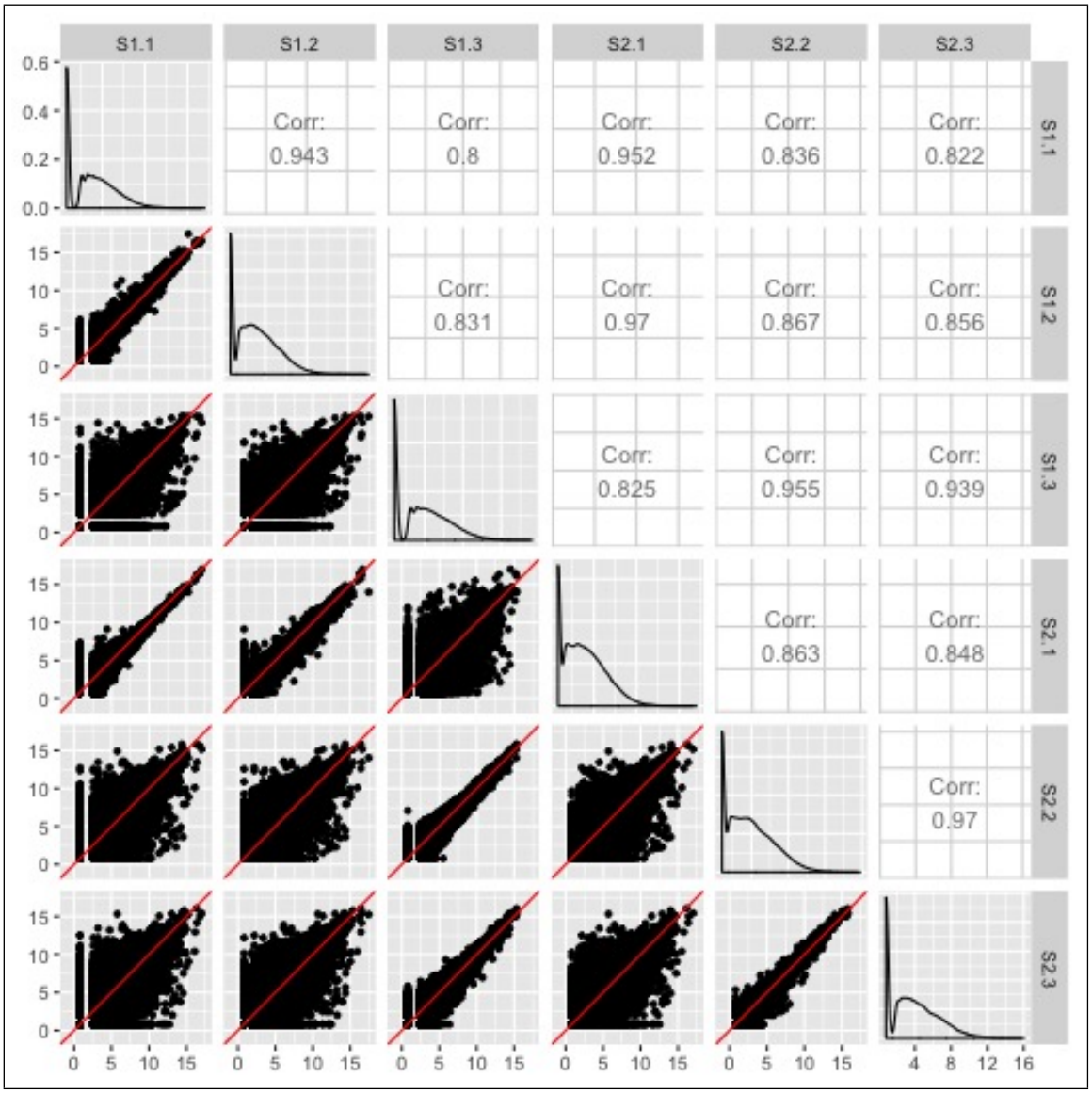

Figure 5: As expected, this scatterplot matrix contains nine scatterplots with thicker distributions (should be treatment pairs) and six scatterplots with thinner distributions (should be replicate pairs). However, two samples appear to cause a subset of scatterplots to unexpectedly show thicker distributions between replicate pairs and thinner distributions between treatment pairs. If we switch the labels of these two suspicious samples (S1.3 and S2.1), the scatterplot matrix then displays the anticipated structure we saw in Figure 2. At this point, we have evidence that these two samples may have been mislabeled, and we may wish to confirm this suspicion and correct it before continuing with the analysis.

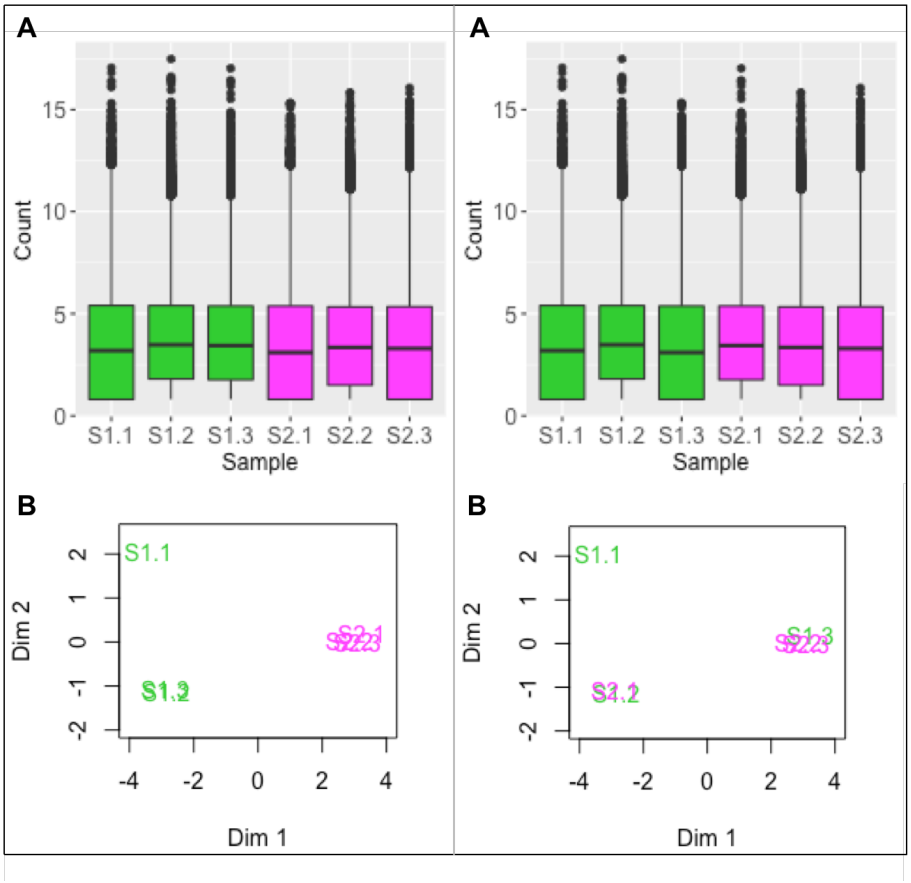

Figure 6: Boxplots and MDS plots are popular plotting tools for RNA-seq analysis. This figure shows these traditional plots applied to the cotyledon data before sample switching (left half) and after sample switching (right half). We cannot suspect from the right boxplot that samples S1.3 and S2.1 have been swapped (subplots A). This is because all six samples have similar five number summaries. For the MDS plots, we do see a cleaner separation of the two treatment groups across the first dimension in the left plot than in the right plot (subplots B). However, taking into account the second dimension, both MDS plots contain three clusters, with sample S1.1 appearing in its own cluster. Without seeing one distinct cluster for each of the two treatment groups, it is difficult to suspect that samples S1.3 and S2.1 have been swapped in the right MDS plot (subplots B). This is because we are not informed about variation at the gene level with the MDS plots like we were with the scatterplot matrix (Figure 5).

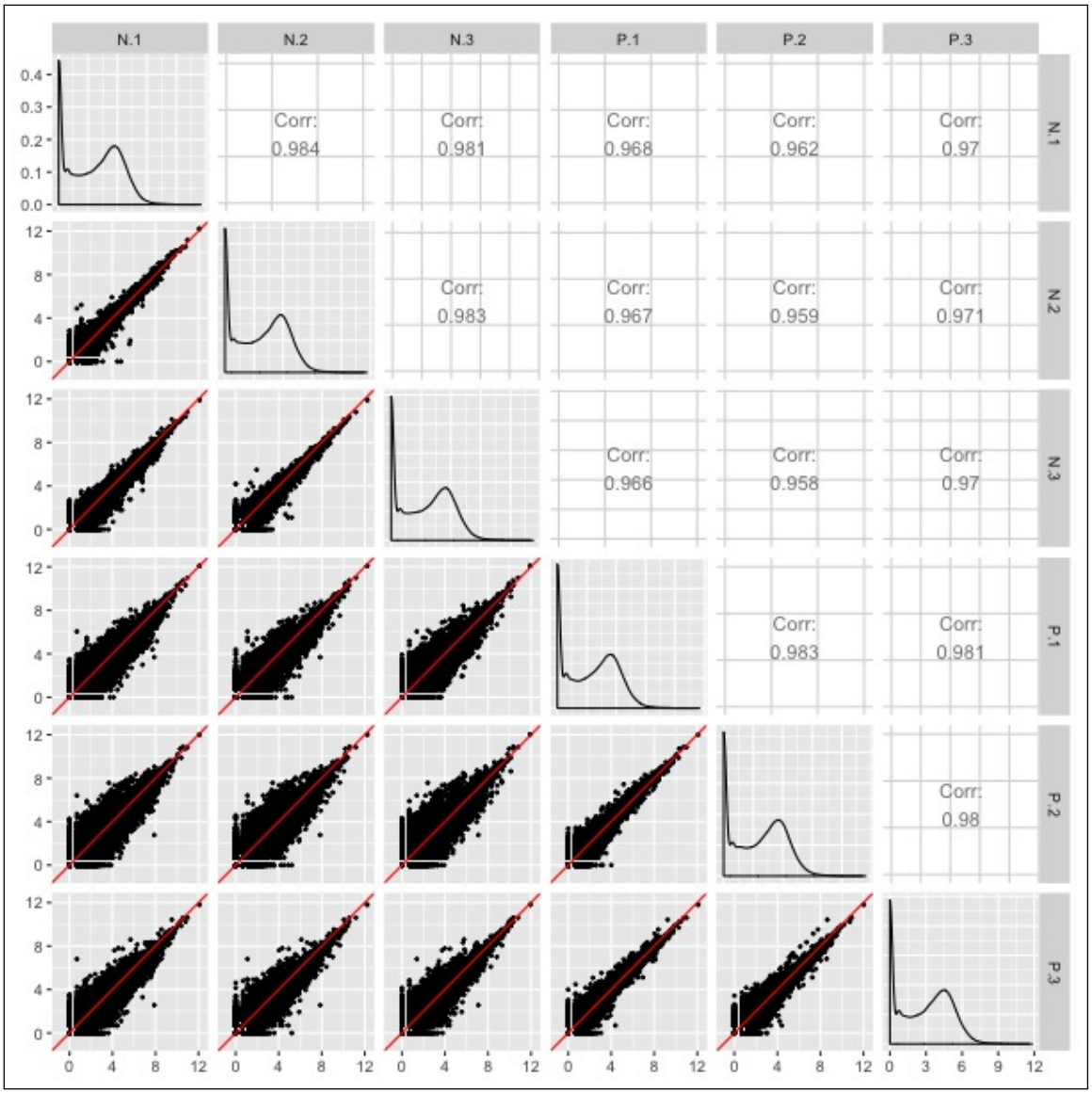

Figure 7: Scatterplot matrix of RNA-seq read counts from soybean leaves after exposure to iron-sufficient (treatment group P) and iron-deficient (treatment group N) soil conditions. We observe the expected structure of treatment pairs showing larger variability around the $x=y$ line than replicate pairs. However, we notice a pronounced streak structure in the bottom-right scatterplot that compares two replicate samples from the iron-sufficient group. The genes in the streak structure have large read counts that deviate in a parallel fashion from the $x=y$ line. Through contacting the authors of this dataset, we discovered that a leaf on one of these samples was inadvertently torn and then documented as such during the experiment. Hence, the genes within this streak structure might represent those that responded to this leaf-tearing event, an observation discovered through the scatterplot matrix that could solidify into a post-hoc hypothesis.

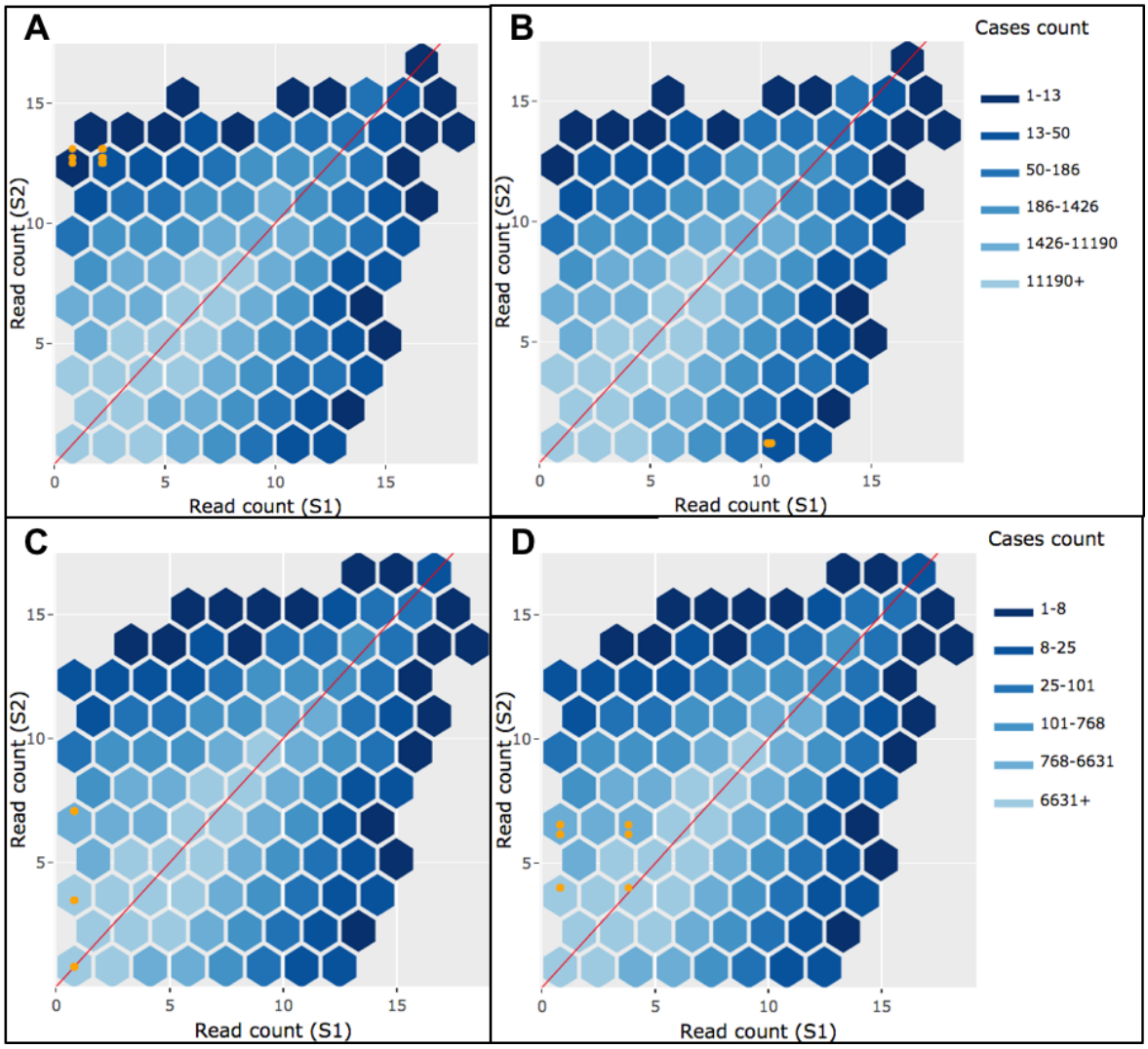

Figure 8: Subplots A and B each show a replicate point plot for a gene drawn from the ten genes with the lowest FDR values from the cotyledon data. In each case, the gene is overlaid as nine orange points (some of them overlap) showing the read count values for all nine combinations of treatment pairs. These two example genes show the pattern we expect from differentially expressed genes, having a large difference between treatments (distant from the x=y line) and small difference between replicates (orange points close to each other). In contrast, subplots C and D each show a replicate point plot for a gene drawn from the ten genes with the lowest FDR values from the cotyledon data after samples S1.3 and S2.1 have been switched. These two example genes do not show the pattern we expect from differentially expressed genes, having a small difference between treatments (close to the x=y line) and large difference between replicates (orange points far from each other). The interactive version of these plots confirms that the FDR values only reach significance for the cotyledon data before sample swapping, meaning the model and the visualizations are consistent. Hence, replicate point plots can be used to quickly sift through genes of interest and determine if the statistical outputs from the model sensibly match the visual findings.

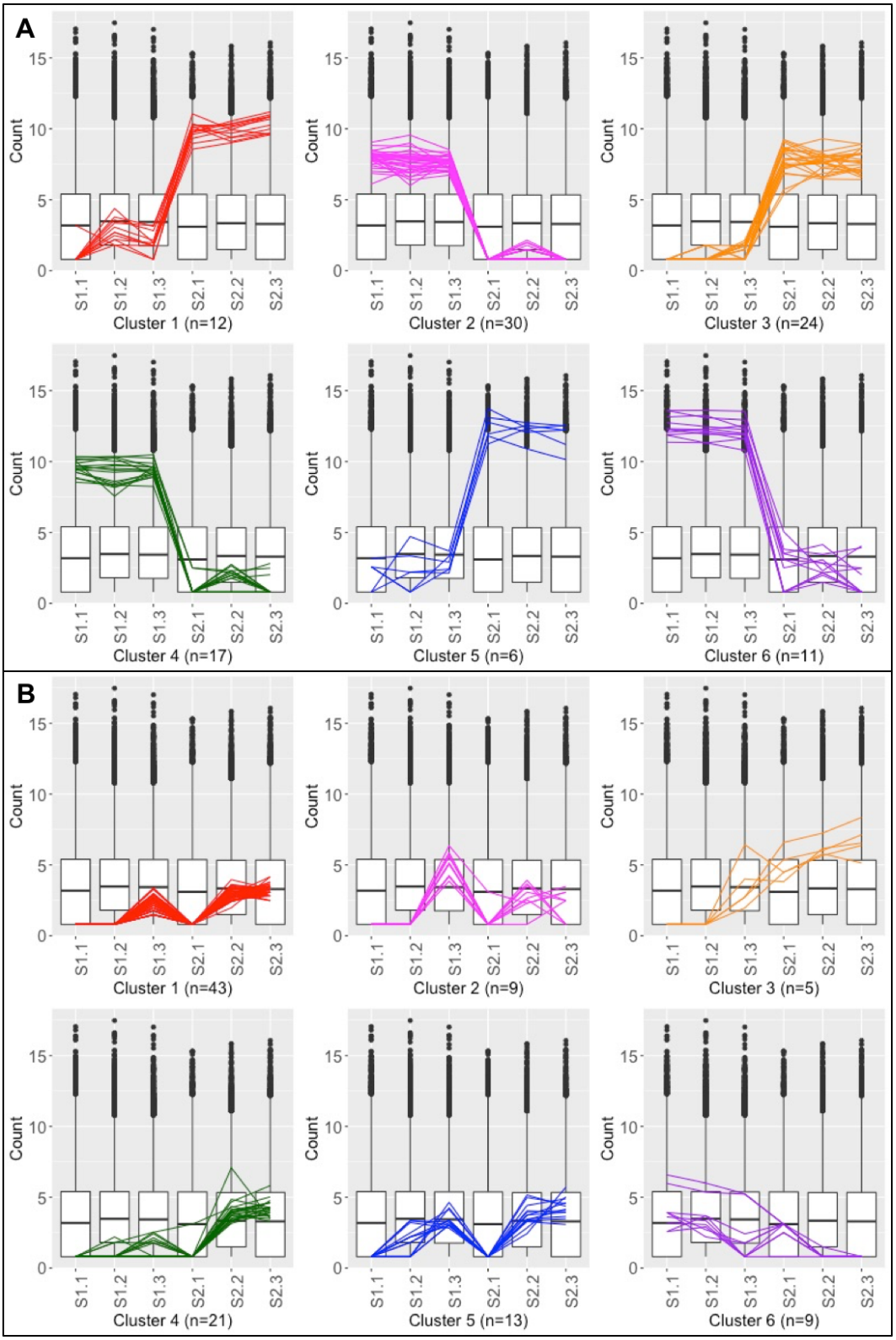

Figure 9: Hierarchical clustering of the 100 genes with the lowest FDR values from the cotyledon dataset before (subplot A) and after (subplot B) swapping samples S1.3 and S2.1. The parallel coordinates show the expected structure of differentially expressed genes before sample swapping (subplot A), but not after sample swapping (subplot B).

Culture/Library prep.	Library prep. protocol	Growth condition	Flow-cell
Y1	Protocol 1	YPD	428R1
Y1	Protocol 1	YPD	4328B
Y2	Protocol 1	YPD	428R1
Y2	Protocol 1	YPD	4328B
Y7	Protocol 1	YPD	428R1
Y7	Protocol 1	YPD	4328B
Y4	Protocol 2	YPD	61MKN
Y4	Protocol 2	YPD	61MKN

Table 1: We used this publicly-available RNA-seq dataset on Saccharomyces cerevisiae (yeast) growth because its design allows us to examine both technical effects (library preparation, flow cell, and lane) and biological effects (growth condition and culture). The authors used three growth conditions and ten cultures from independent colonies sequenced using two different library preparation protocols and either one or two lanes in a total of five flow-cells (Risso, 2011).