ENAR Student Paper Submission

Visualization methods for RNA-sequencing data

Summary: This is the summary for this paper.

Key words: Data visualization; Exploratory data analysis; Interactive graphics; RNA-sequencing; Statistical graphics

This paper is being submitted for consideration as an ENAR Student Paper

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 (Grabherr et al., 2011; 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).

Gene expression 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-specific and read-position specific 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 senseantisense 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 have 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 analyze data is to iterate between models and visuals, and to enhance the appropriateness of applied models based on feedback from visuals (Unwin, 1992; Shneiderman, 2002).

When visualizing RNA-seq data, we primarily want to compare the variability between replicates and between treatment groups. This is best achieved by viewing the mapped read count distributions across all genes and samples. Unfortunately, the few visual tools available in popular RNA-seq packages do not allow users to effectively view their data in this manner.

Here, we use real RNA-seq data to demonstrate that our visualization 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 any models. In line with modern multivariate data exploration, we emphasize that interactive graphics should be an indisposable component of RNA-seq analysis. Researchers should be able to quickly flip through plots of genes that appear promising or problematic, and link between different types of plots to quickly obtain various perspectives of their data. In this paper, we are not proposing that users completely 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 parallel coordinate plots, scatterplot matrices, and replicate line 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) in the data table as a line. Connections between samples with positive correlations will be flat, and connections between samples with negative correlations will be crossed. The ideal dataset will have large variability between treatment groups but little variability between replicates. Researchers can quickly confirm this with a parallel coordinate plot: There should be flat connections between replicates but crossed connections between treatment groups.

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 of the figure 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 boxlots at the top of the figure as they both show fairly consistent five number summaries across their six samples. Likewise, we cannot detect any 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 group A appears as an outlier in the right MDS plot.

[Figure 1 about here.]

Despite this, we immediately see from the parallel coordinate plots at the bottom of the figure that the left and right datasests 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 on the sample level, while the parallel coordinate plot shows the sample connections for each of the 50 genes.

3. Scatterplot matrices

3.1 *Overview of scatterplot matrices*

Pairwise scatterplot matrices are another effective multivariate visualization tool that plot the mapped read count distributions across all genes and samples. A scatterplot matrix draws each row (gene) in the data table as a point in each scatterplot. With these plots, users can quickly discover unexpected patterns, recognize geometric shapes, and assess the structure and association between multiple variables simultaneously in a different manner than 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 genes to show differential expression between samples. However, if the data has lower variability between replicates than between treatments, then we expect the spread of the scatterplot observations 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). We can immediately verify that the nine scatterplots between treatments (in the bottom-left corner of the matrix) have more spread around the x=y line than the six scatterplots between replicates.

[Figure 2 about here.]

After confirming this expected trend, users can also use the scatterplot matrix to focus on subsets of genes: Outlier genes that deviate from the x=y line in replicate scatterplots are problematic genes, whereas outlier genes that deviate from the x=y line in treatment scatterplots are potential differentially expressed genes. In order to achieve this capability, users must be able to interact with the plot. Figure x shows a static screenshot of this capability, but the interactive version of this plot is available at x.

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 significantly slow down the interactive capabilities of the plot. To solve this, as shown in Figure x, 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.

In the interactive version of the plot, 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 gene counts from all samples. This interactivity and linking allows users to quickly examine genes of interest from multiple perspectives superimposed on the summary of the genes from the whole dataset. Readers can interact with this graphic at x. We show a static screen shot of the interative graphic being used in which a user has clicked on a hexagon in the scatterplot between the groups S1.1 and S1.2 that deviated from the x=y line. This hexagon contained two genes, which are superimposed in orange across all scatterplots. Upon viewing the parallel coordinate plot, we see that these two genes are indeed problematic for having inconsistently high read counts for the second replicate of the S1 treatment group. We can also immediately obtain the identification of these two genes as ”Glyma06g11430.1” and ”Glyma13g02510.1”.

[Figure 3 about here.]

[Figure 4 about here.]

The scatterplot matrix can also be used after differential expression analysis to quickly examine DEGs obtained from a given model. As shown in Figure 5, the DEGs can be superimposed as orange points onto the scatterplot matrix. We would 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 5. As a side note, we can also link these DEGs as parallel coordinate lines on a side-by-side boxplot like in Figure 6 to confirm with another viewpoint that we see the expected pattern for differential expression. If we do not observe what should be expected of DEGs, then the DEG calls from the model need to be scrutinized further.

[Figure 5 about here.]

[Figure 6 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). As shown in Table 1, the data contained four cultures from independent libraries that were sequenced using two different library preparation protocols and either one or two lanes in a total of three flow-cells. This experimental design allowed for researchers to examine various levels and combinations of technical effects (library preparation and protocol and flow cell) and biological effects (culture).

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 7 shows the scatterplot matrix of the Y1 and Y4 replicate read counts after within-lane normalization. It is immediately clear that the data has not been sufficiently normalized due to the deviation from the x=y lines between the treatment groups. In contrast, Figure 8 shows the scatterplot matrix of the Y1 and Y4 replicate read counts after both within-lane and between-lane normalization, as was recommended by the authors for its reduced false-positive rate. Indeed, the scatterplot matrix now follows the expected structure with most genes falling along the x=y line and thicker deviations from it between treatment groups than between replicate groups.

Moreover, we can also confirm that the read counts fall closer to the x=y line between the Y4 replicates than between the Y1 replicates. This is expected because the Y1 replicates had additional technical variability from the use of two different flow cells (Table 1). As such, the scatterplot matrix can also be used to quickly confirm expected patterns of variability in the dataset.

[Figure 7 about here.]

[Figure 8 about here.]

[Table 1 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. In 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 quick and easy tool to check for common errors like sample mislabeling. Figure 9 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 there are nine scatterplots with thicker distributions around the x=y line and six scatterplots with thinner distributions around the x=y line. We know there are nine between-treatment and six between-replicate combinations between all pairwise combinations of the six samples. Hence, Figure 9 contains the correct number of scatterplots showing thick and thin distributions around the x=y line, only some scatterplots appear to be out of order. Rearranging the two samples that appear suspicious in the scatterplot matrix would indeed lead us back to the clean-looking scatterplot matrix we saw in Figure 2.

[Figure 9 about here.]

Unfortunately, this common type of error cannot be as readily detected in the plotting tools available in popular RNA-seq packages. For instance, Figure x shows the MDS plot and boxplot of the dataset before and after the sample switch. (!!!!! ADD MORE 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 unsummarized 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 10. 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. A more detailed explanation can be found in (Moran Lauter et al., 2014). After normalizing the data, we see the expected pattern of a scatterplot matrix in Figure 10, with more variation around the x=y line between treatments than between replicates.

However, one streak structure in the botton 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 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 10 about here.]

How include replicate line plot? (Just show images from cotyledon data) (or try SVA data with artificially high number of false positives)

4. Discussion

Put your final comments here.

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 chemosenstivity from cell lines: Forensic bioinformatics and reproducible research in high-throughput biology. *The Annals of Applied Statistics* **3,** 13091334.

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.

Grabherr, M. G., Haas, B. J., Yassour, M., Levin, J. Z., Thompson, D. A., Amit, I., et al. (2011). Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature Biotechnology* **29,** 644652.

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.

Unwin, A. (1992). How interactive graphics will revolutionize statistical practice. *The Statistician* **41,** 365-369

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.

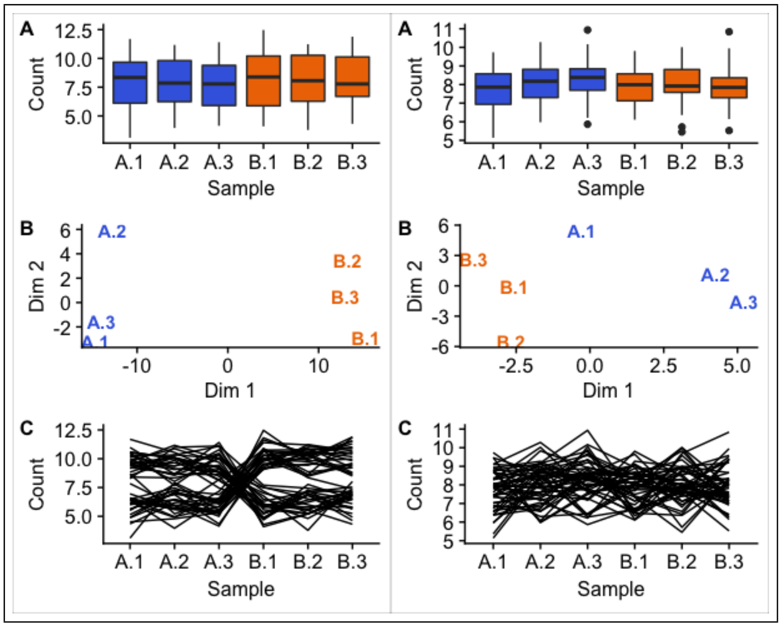


Figure 1. Caption.

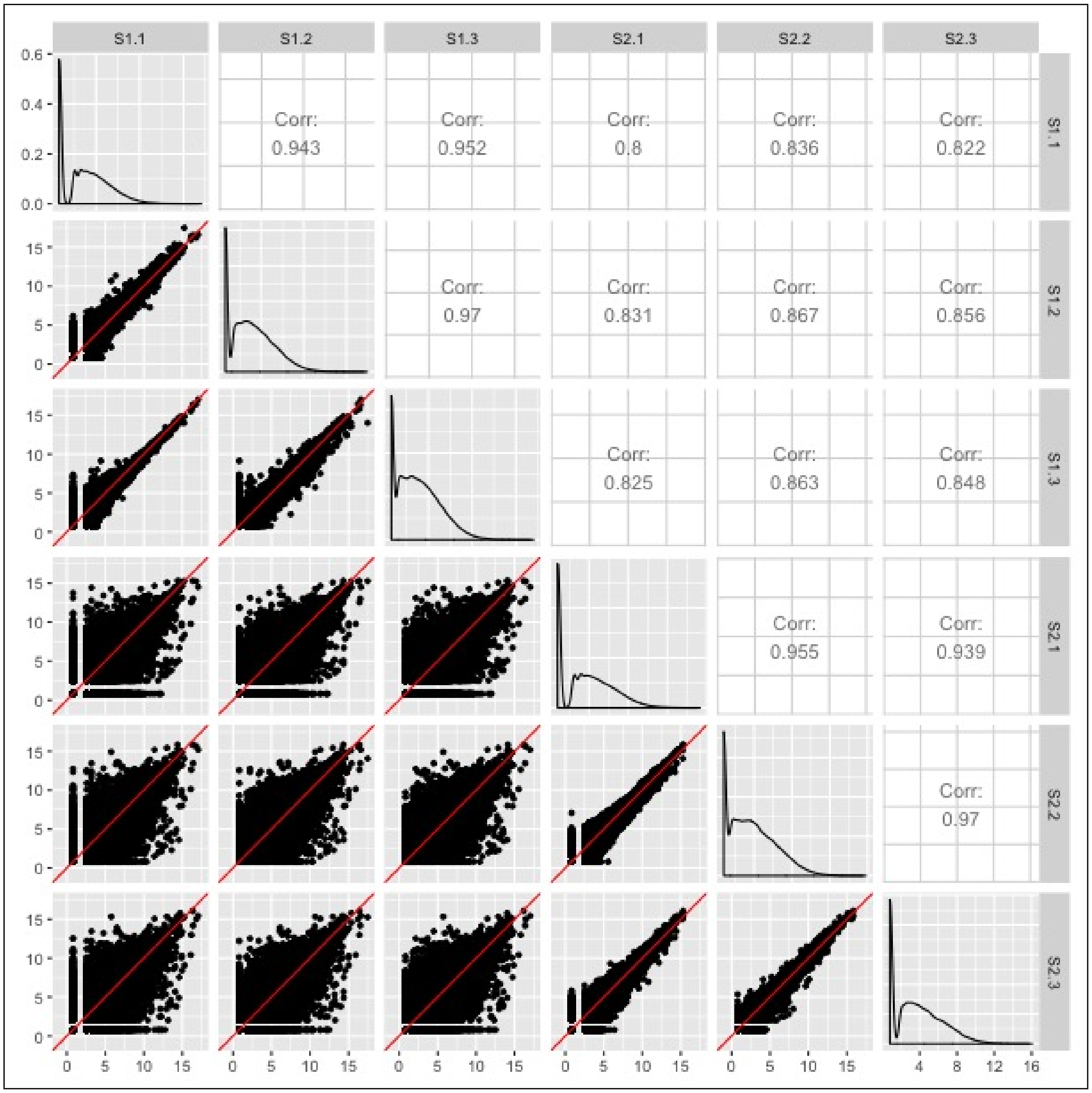


Figure 2. Caption.

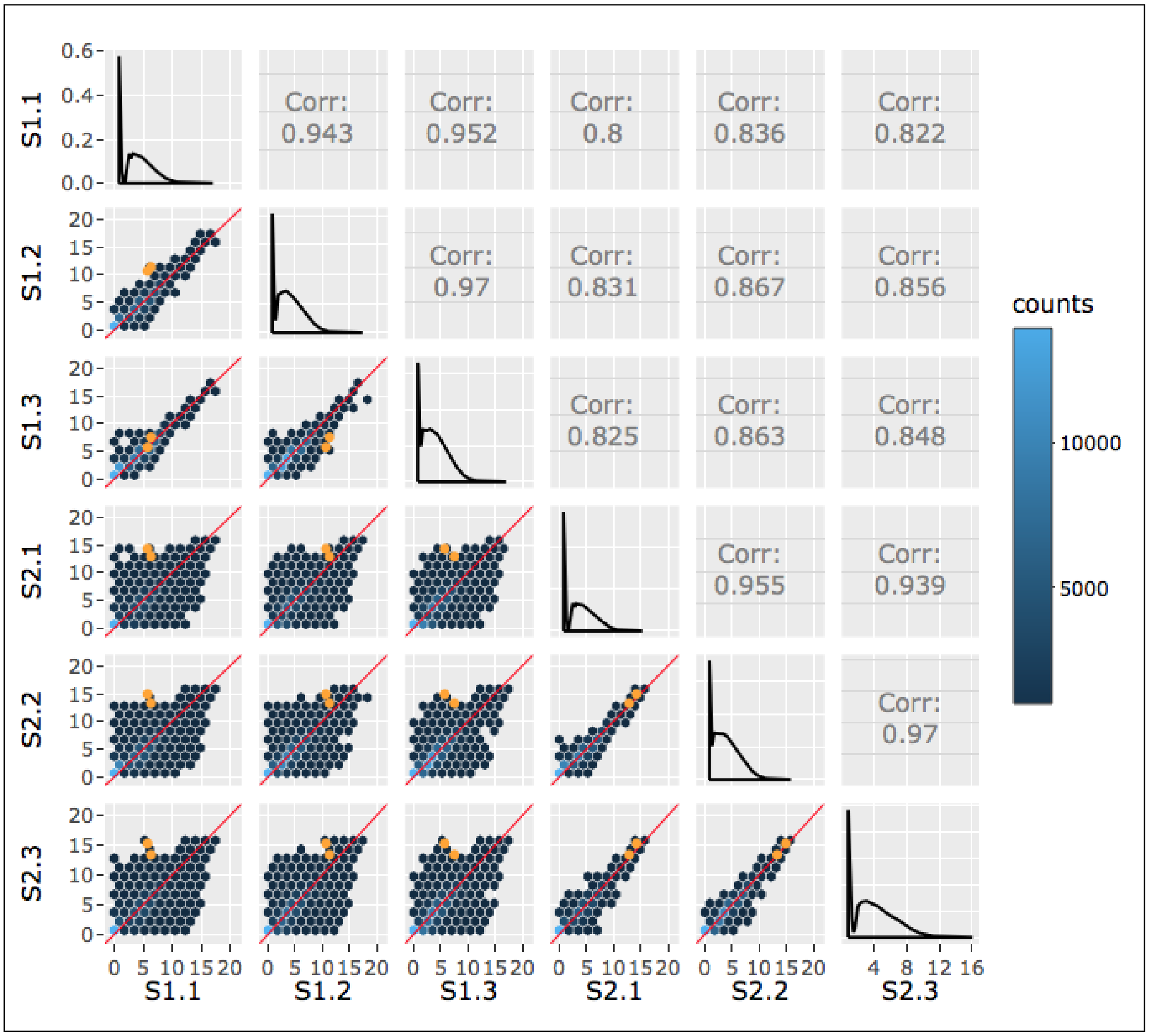
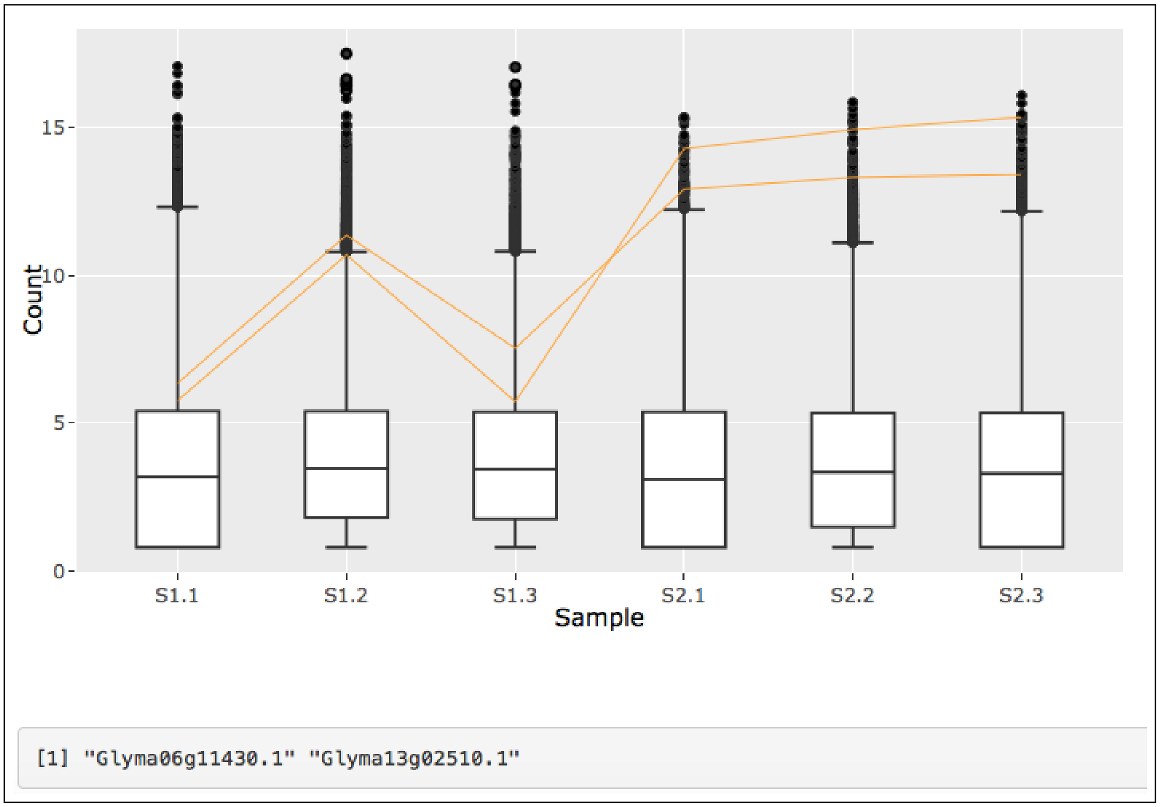


Figure 3. Caption.



```
[1] "Glyma06g11430.1" "Glyma13g02510.1"
```

Figure 4. Caption.

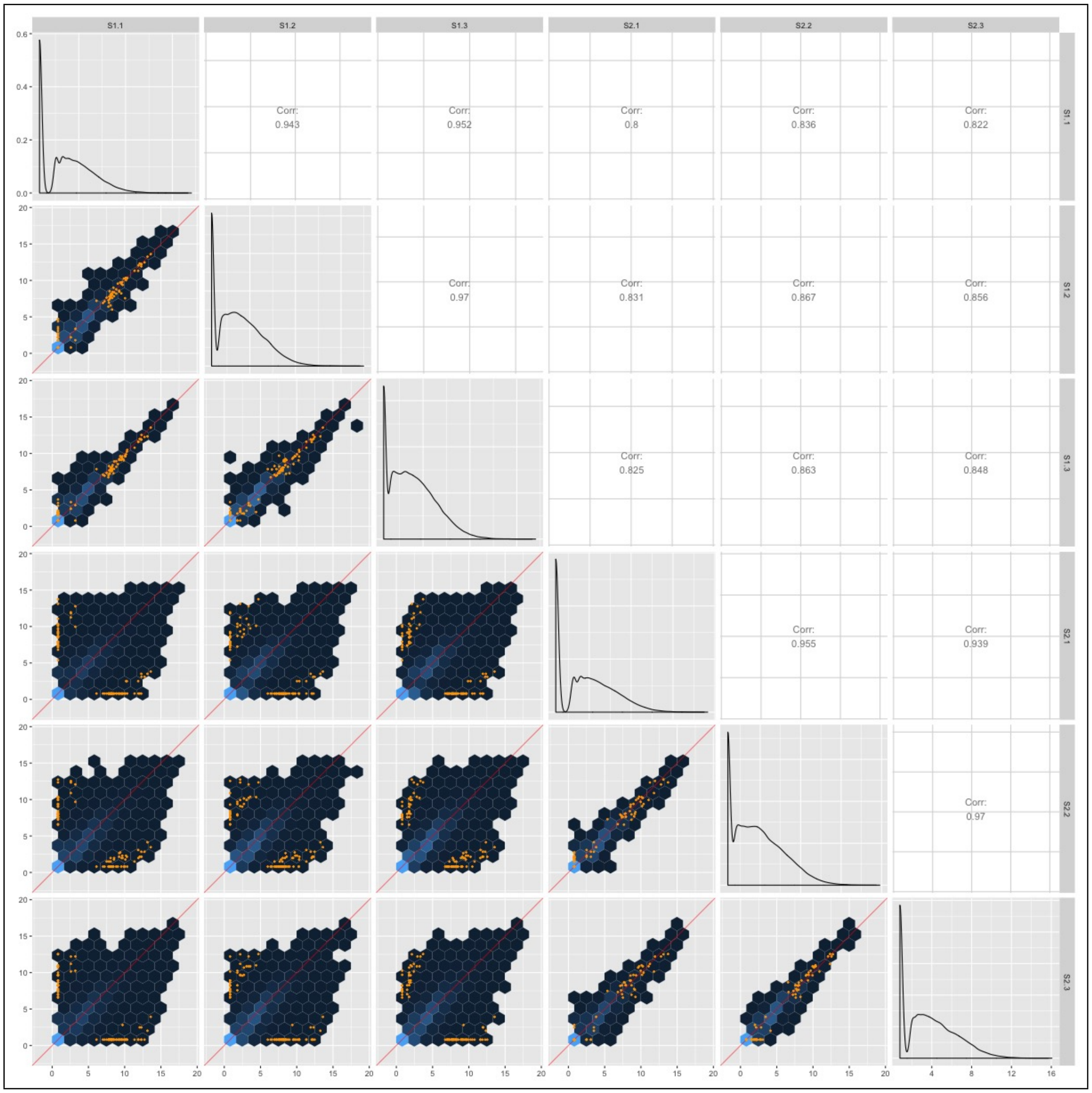


Figure 5. Caption.

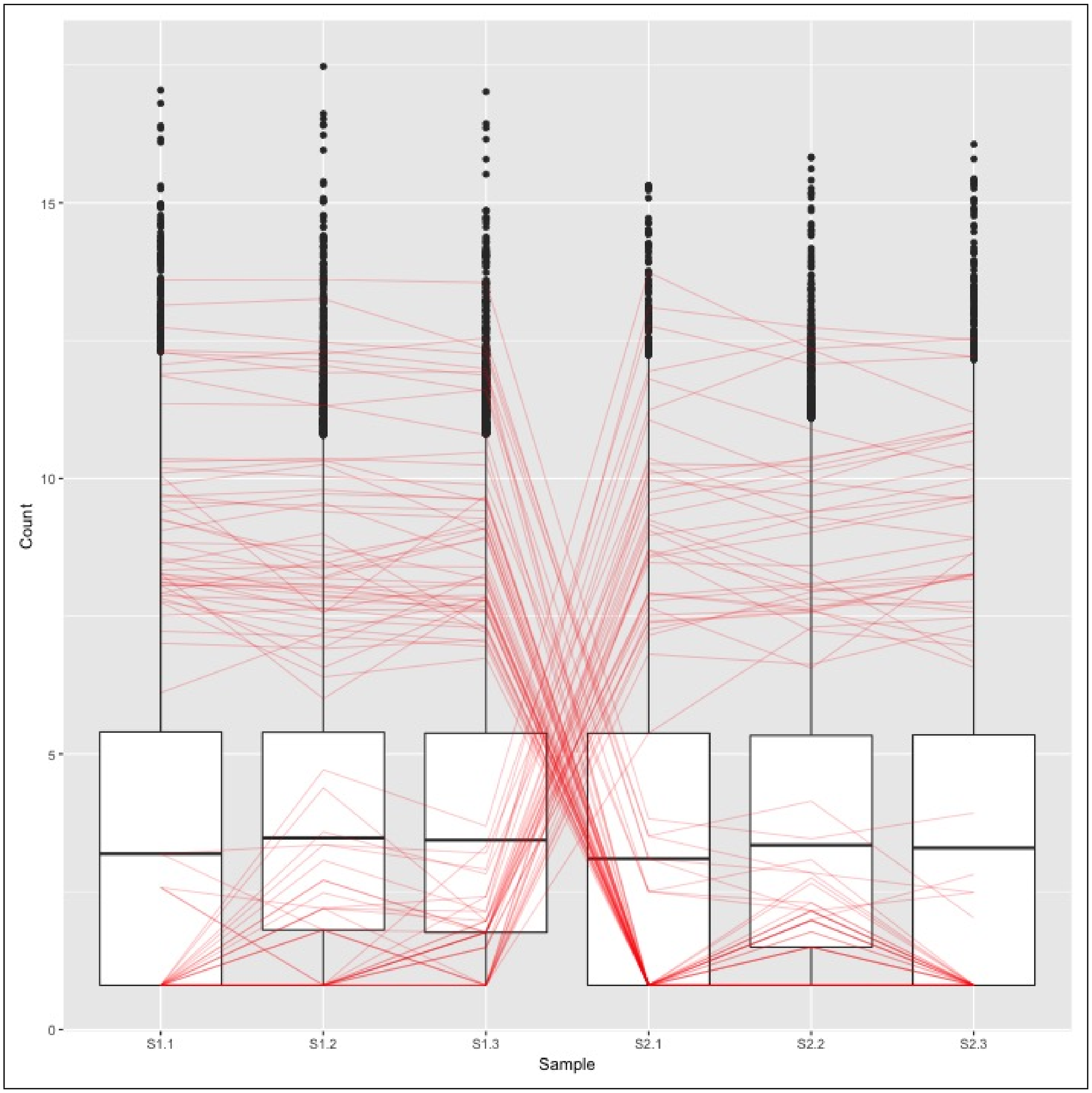


Figure 6. Caption.

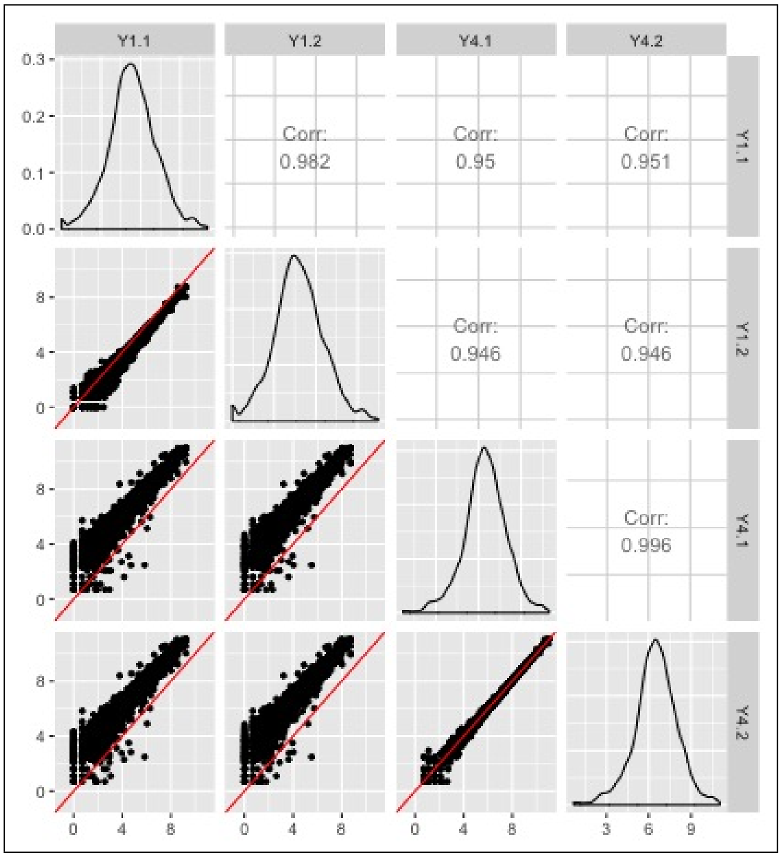


Figure 7. Caption.

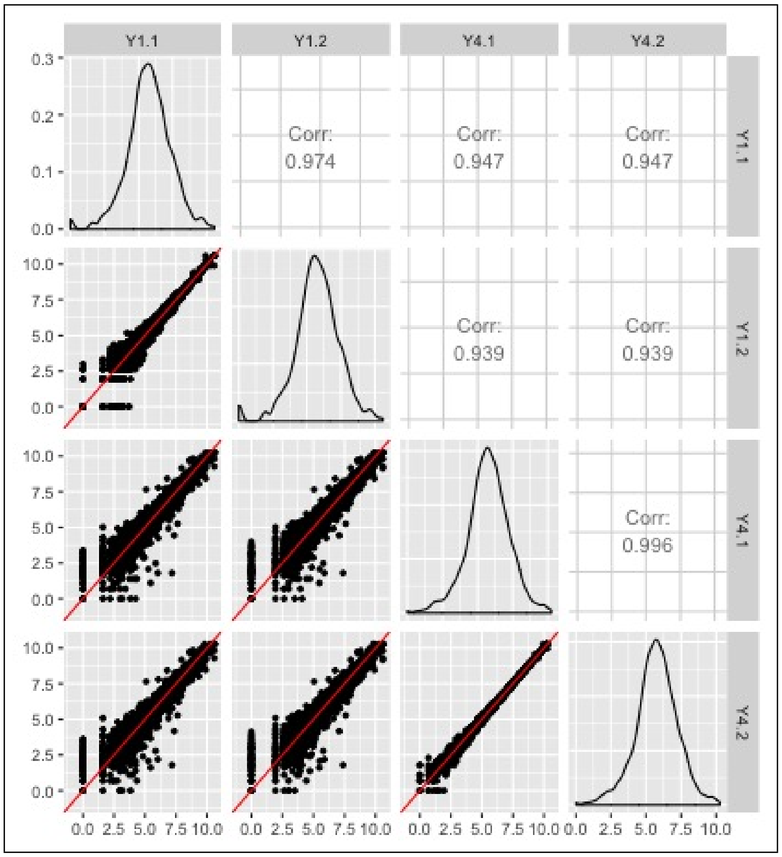


Figure 8. Caption.

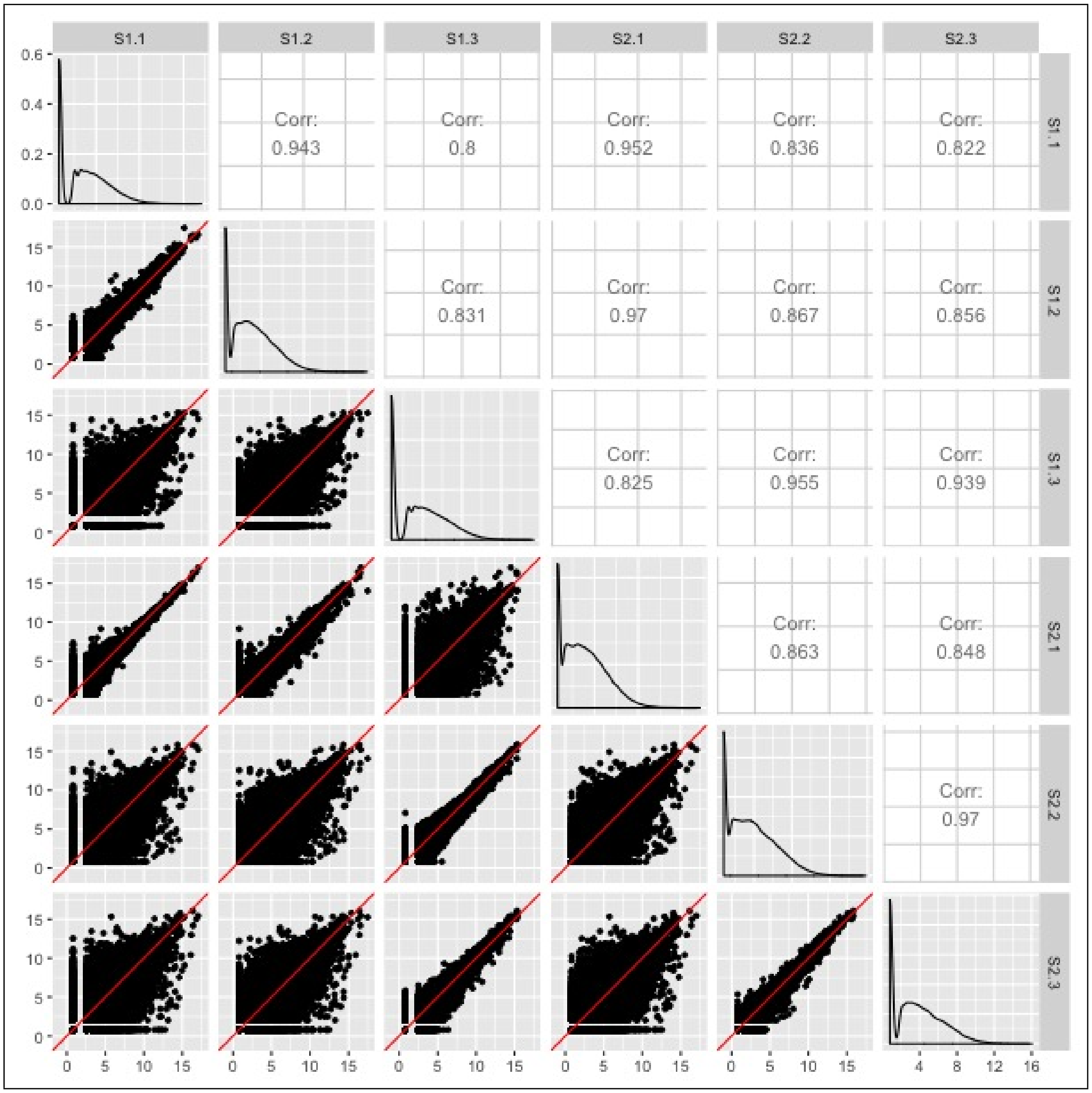


Figure 9. Caption.

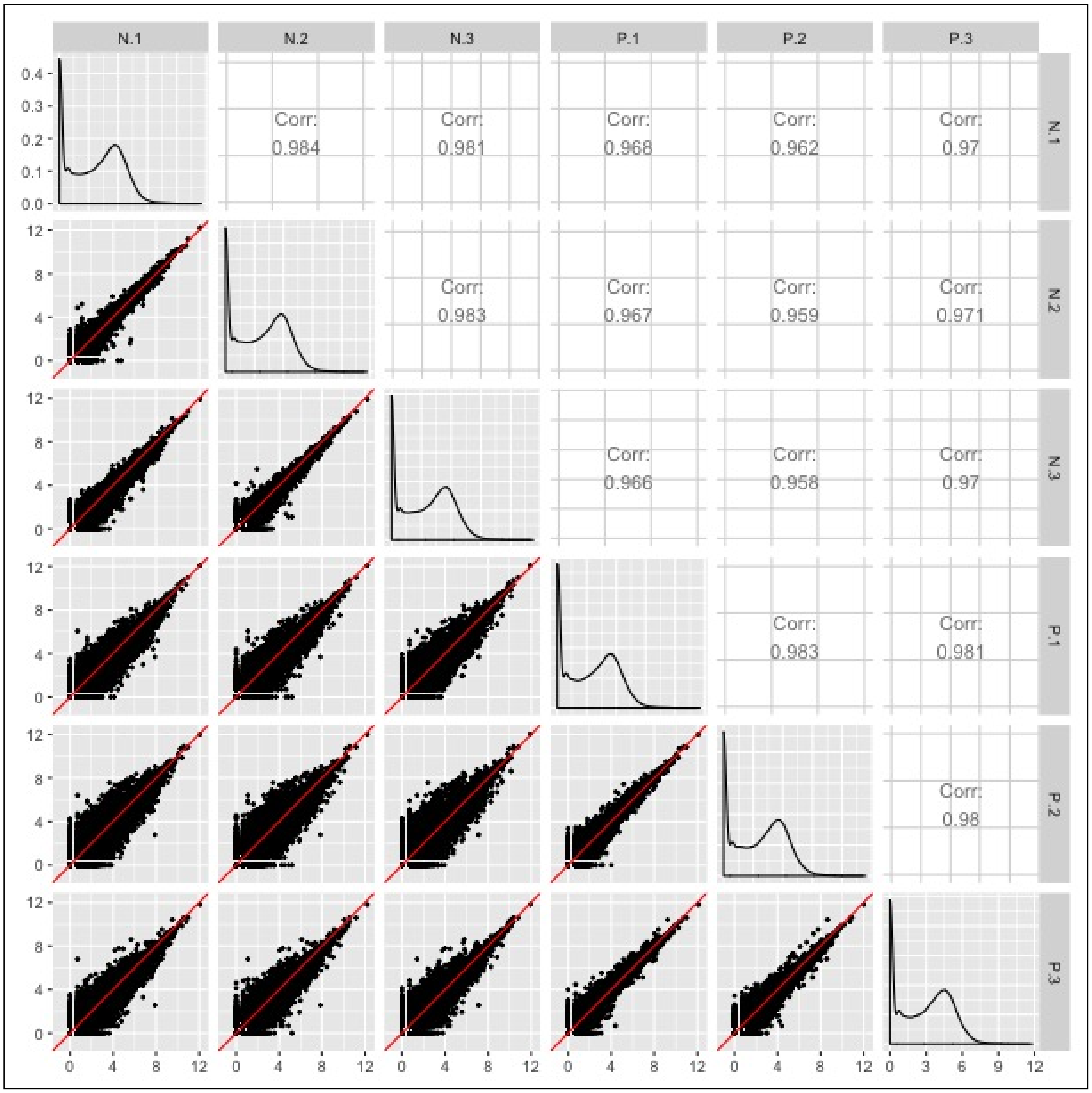


Figure 10. Caption.

Table 1
This is a simple table.

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