Joseph Bui
Brandon Tsui
Luigi Cheng
Professor Shannon Ellis
Professor Aaron Fraenkel

# Comparison of Differential Gene Expression Analysis Tools

## Abstract

RNA-Seq (named as an abbreviation of "RNA sequencing") is a technology-based sequencing technique that uses next-generation sequencing (NGS) to reveal the presence and quantity of RNA in a biological sample at a given moment, analyzing the continuously changing cellular transcriptome. Differential expression analysis (DEA) takes the normalized read count data (number of sequence reads originated from a particular gene) and performs statistical analysis to discover quantitative changes in expression levels between experimental groups. As technologies keep progressing and improving, there are now multiple tools that can be used to carry out differential expression analysis. The purpose of our project is to take a closer look at some of these tools and compare their performance to understand which tools are optimal for DEA in different situations. Specifically, the software that we are going to focus on are: ABSSeq[1], voom.limma[2], PoissonSeq[3], DESeq2[4], NOISeq[5], ttest[6] , and edgeR[7]. We will compare their performances by looking at parameters such as Area Under the ROC Curve (AUC), False Discovery Rate (FDR), Type I error rate, Sensitivity, and Specificity. In this project, we discovered that DESeq2 & edgeR performed similarly and were the overall best tools in most metrics and datasets. However, there were some circumstances in which other tools had advantages. When there were outliers with unusually high counts, both voom.limma & ttest performed significantly better, while ABSSeq showed benefits in controlling Type I errors when the number of truly differentially expressed genes was low. Overall, there was no single tool applicable in every situation. We could only identify each tool's strengths and weaknesses and showed circumstances where some tools were better over others.

## Background

In genetic research, the understanding of transcriptomes – the set of all RNA transcripts – is crucial for researchers to gain insight into the development of diseases, conditions, or disorders

with known genetic etiologies. The goal of the transcriptomic research is to catalog all transcript species, including mRNAs, non-coding RNAs & small RNAs, and, more importantly, to quantify each transcript's changing expression levels during development and under various environmental and disease conditions. Identifying genes that are differentially expressed helps determine which biological mechanisms could affect a disease or disorder. In the past, researchers primarily used hybridization approaches, such as microarrays, to deduce and quantify these transcriptomes[9]. Hybridization involves incubating labeled cDNA to microarrays. While being relatively high throughput and inexpensive, it also has limitations such as reliance upon existing knowledge about genome sequence and limited dynamic range of detection[9]. Traditionally, researchers utilized microarrays to analyze genes. However, RNA-Sequencing has recently grown in popularity. It provides much more precise measurements and has the potential to cover a broader range of transcripts that haven't been correlated to an existing genome. Compared with microarrays, RNA-Sequencing also has an extremely low background signal and all at a lower cost.

With the emergence of RNA-sequencing data, countless software have been developed to extract information from such data. To process the RNA-sequencing data, a researcher needs to first quality check the reads produced by RNA-seq. For some instances, it is necessary to clean the RNA-seq reads from contamination from adapters during preprocessing. Next, the cleaned reads need to be aligned, mapping each read to a genome. Finally, the researcher analyzes the differentially expressed genes among all the samples between experimental conditions. For each step of this process, there are corresponding tools that can be used to help the researchers. In every genetic study using RNA-seq data, researchers must determine which tools to use and how to use them precisely. There is no single standardized pipeline for differential expression due to the diversity in types of RNA data. Here, we hope to investigate how different software perform on distinct synthetic datasets. Because DEA is the most crucial part of the pipeline for RNA-sequencing, we want to focus the core of this project on comparing gene differential expression tools. Overall, we are investigating the following tools: ABSSeq[1], voom.limma[2], PoissonSeq[3], DESeq2[4], NOISeq[5], ttest[6], and edgeR[7]. We are evaluating their performance on Area Under the ROC Curve (AUC), False Discovery Rate (FDR), Type I error rate, Sensitivity, and Specificity. This project's importance is to possibly help future researchers by providing

them information about which tools they could utilize for their RNA-seq research for the best results based on the composition of the data and which accuracy metrics needed to be controlled.

## Dataset

Since differentially expressed (DE) genes are determined only within a degree of certainty in a real-life RNA-seq dataset, we decided to test the tools using simulated post-alignment datasets. By doing so, we can control variables such as the proportion of genes differentially expressed, the number of up and down-regulated genes, the number of outliers, and the number of samples per condition. For a synthetic dataset, we can know with complete certainty which genes are truly differentially expressed. Therefore, we can calculate accuracy metrics outputted by the tools against the actual metrics. This feat would have been impossible to accomplish in an actual experiment. We chose to create several datasets with various combinations of differentially expressed genes and samples per condition to capture the different types of real-life genetic data (Table 1). Thus, we can observe if specific tools performed better when there were more or fewer differentially expressed genes.

| **Sim. study** | $G_{DE}^{up}$ | $G_{DE}^{down}$ | **\|{g; $\phi_g$ = 0}\|** | **'Single' outlier fraction** | **'Random' outlier fraction** |
|---|---|---|---|---|---|
| $B_0^0$ | 0 | 0 | 0 | 0 | 0 |
| $B_0^{1250}$ | 1,250 | 0 | 0 | 0 | 0 |
| $B_{625}^{625}$ | 625 | 625 | 0 | 0 | 0 |
| $B_0^{4000}$ | 4,000 | 0 | 0 | 0 | 0 |
| $B_{2000}^{2000}$ | 2,000 | 2,000 | 0 | 0 | 0 |
| $P_0^0$ | 0 | 0 | 6,250 | 0 | 0 |
| $P_{625}^{625}$ | 625 | 625 | 6,250 | 0 | 0 |

| | | | | | |
|---|---|---|---|---|---|
| $S_0^0$ | 0 | 0 | 0 | 10% | 0 |
| $S_{625}^{625}$ | 625 | 625 | 0 | 10% | 0 |
| $R_0^0$ | 0 | 0 | 0 | 0 | 5% |
| $R_{625}^{625}$ | 625 | 625 | 0 | 0 | 5% |

**Table 1** The table above shows the different generated The 'B' represents the baseline, 'P' represents the Poisson, 'S' represents the single outlier, and 'R' represents the random outlier. $|\{g: \Phi_g = 0\}|$ represents the number of genes whose counts were drawn from a Poisson distribution. In each simulated study, there are different numbers of differentially expressed genes between the 2 conditions which will be explained more thoroughly below.

In all simulated studies, there are 12,500 total genes and 2, 5, & 10 samples between 2 conditions, denoted by $S_1$ and $S_2$. The simulated studies (first column) have superscripts and subscripts. The superscript represents the number of upregulated DE genes ($G_{DE}^{up}$) and the subscript represents the number of downregulated DE genes ($G_{DE}^{down}$) in the second condition ($S_2$) compared to the first condition ($S_1$). The baseline, single outlier, and random outlier have counts generated from the Negative Binomial distribution, whereas Poisson is generated from the Poisson distribution. The 'single' outlier fraction is the fraction of genes in a selected single sample where the corresponding count is multiplied with a factor between 5 and 10 — 'random' outlier fraction is similar but with a randomly selected sample. Real-life data are usually very messy and often prove to be more challenging than plain synthetic datasets. Hence, we decided to add conditions such as the Poisson distribution, single and random fractions in order to emulate real-life data's unpredictability and test each tool against different circumstances.

## Methods

### Creating the Synthetic Data

We used compcodeR[10] to investigate the different tools by first creating the synthetic data using the built-in function, *generateSyntheticData*[11]. For the distinct 11 simulated datasets, we specified the parameters: *`n.vars`* = 12,500, *`samples.per.cond`* = 2, 5, or 10, *`dispersions`* = # $|\{g; \phi_g = 0\}|$ column in Figure 1. To produce the fraction of differentially expressed genes that is

upregulated in $S_2$ compared to $S_1$, *`fraction.upregulated`* = the ratios shown in Figure 1 (i.e. 0.5 for $B_{625}^{625}$). For the single outlier fraction datasets, we specified *`single.outlier.high.prob`* = 0.05 (fraction of single outlier has unusually high counts) and *`single.outlier.low.prob`* = 0.05 (fraction of single outlier has unusually low counts). As for the random outlier fraction datasets, we specified *`random.outlier.high.prob`* = 0.025 (fraction of random outliers with unusually high counts) and *`random.outlier.low.prob`* = 0.025 (fraction of random outliers with unusually low counts). For each dataset, we generated 10 different versions since each are randomly generated from different distributions in order to capture the variance in performance of each tool.

## Performing DEA

For performing tools supported by compcodeR, we used the built-in function called *`runDiffExp`* where we specify the *`result.extent`* parameter as: DESeq2, edgeR, NOISeq, voom.limma, or ttest. ABSSeq and PoissonSeq, which are not as commonly used, were not supported by compcodeR and needed to be run separately. For both tools, the count matrix was extracted from the compcodeR object for each dataset and labeled to distinguish between conditions based on the number of samples per condition. Both were then run using default parameters, and genes were labeled 1 or 0 based on a cutoff value of 0.05 for adjusted p-value.

| **Tool** | **Normalization Method** | **Statistical Method** |
|---|---|---|
| ***ABSSeq***[1] | • Qtotal<br>  ○ "qtotal assesses the influence of DE on data structure to normalize the data."[1] | • Uses absolute counts difference between 2 groups<br>• Utilizes Negative binomial distribution and moderating fold-change according to heterogeneity of dispersion across expression level |
| ***voom.limma***[2] | • Trimmed Means of M-values (TMM) | • Applies voom transformation then uses t-test<br>• Voom precision weights unlock linear model analysis tools for read counts<br>  ○ Fits linear model to expression data for each |

| | | |
|---|---|---|
| | | gene |
| ***PoissonSeq***[3] | • Novel normalization using goodness of fit statistic and maximum likelihood estimations over multiple iterations[12] | • Poisson goodness-of-fit statistic<br>• Calculates a core statistic on the basis of a Poisson log-linear model<br>• Estimates the false discovery date using a modified version of the permutation plug-in method |
| ***DESeq2***[4] | • Counts are divided by geometric mean for each gene across all samples | • Estimates the variance-mean dependence in count data using Negative Binomial Distribution<br>• Uses Wald test to perform differential gene expression |
| ***NOISeq***[5] | • Trimmed Means of M-values (TMM) | • Performs quality control of count data<br>• Normalization & filter low-counts<br>• Models noise distribution of count changes by contrasting fold-change differences & absolute expression differences for all features in all samples in same condition |
| ***ttest***[6] | • Trimmed Means of M-values (TMM) | • Uses edgeR package to perform differential expression analysis<br>• Compares 2 conditions using t-test, applied to normalized counts |
| ***edgeR***[7] | • Trimmed Means of M-values (TMM) | • Based on Negative Binomial Distributions, including empirical Bayes estimation, exact tests, generalized linear models, and quasi-likelihood tests<br>• Implements genewise exact tests for differences in the means between 2 conditions of negative-binomially distributed counts[8] |

**Table 2** The table above shows the normalization methods of the 7 tools that we are exploring with an explanation to how the tools perform differential gene expressions analysis in our project.

## Creating the metrics for comparison

After running differential analysis using each tool on every dataset, we found different results depending on each specific tool's output (e.g., ABSSeq's output included variance and fold change among others for each gene). To simplify the results, we chose to use the adjusted p-value with a cutoff value of 0.05 to predict DE genes. The compcodeR data objects also included labels for the true DE genes, which we appended to the results in order to calculate statistical metrics for comparison and analysis. Using these labels and predictions, we were able to calculate the overall accuracy (percentage of genes correctly classified) of each tool on each dataset and more complex statistics such as AUC, FDR, specificity, and sensitivity. We chose only to calculate Type I Error Rate (False Positive Rate) on the datasets with zero differentially expressed genes.

Once we calculated each metric on each dataset, we combined the results into two matrices, separating the datasets with zero differentially expressed genes ($B_0^0$, $P_0^0$, $S_0^0$, and $R_0^0$) into their own matrices due to them having different statistics. Using these matrices, we made boxplots for each dataset and statistic combination (i.e. AUC and $B_{625}^{625}$), grouping samples per condition in the x-axis and tool by color, and aggregating together the dataset version.

## Performing DEA on Real Life Data

In addition to analyzing the synthetic datasets' results, we also analyzed a real-life dataset from the last quarter. This dataset analyzed RNA sequencing on brain tissues of different brain regions to find similarities in molecular changes that might exist in three psychiatric disorders. The dataset extracted genes from post-mortem brains of patients who suffered Schizophrenia (SZ), Major Depressive Disorder (MDD), and Bipolar Disorder (BPD). In order to compare the genes of each disease to a healthy brain's genes, the dataset also extracted genes from a control group, where patients did not suffer any psychiatric disorder.

In this project, we utilized the count matrix post-cleaning and alignment obtained from last quarter to perform differential gene expression analysis with each tool mentioned in this report. To do so, we first preprocessed the dataset by pairing each psychiatric disorders' genes with the control's genes. Hence, we ended up with three pairs of datasets: SZ-Control, MDD-Control, and BPD-Control. We then used annotations of age at death and brain pH to

predict whether each gene was differentially expressed or not using each individual tool. Like we did with synthetic datasets, genes were labeled 1 or 0 based on a cutoff value of 0.05 for each tool's adjusted p-value.

## Results

### *Exploratory Analysis of the Tools*

#### Timing of the Tools

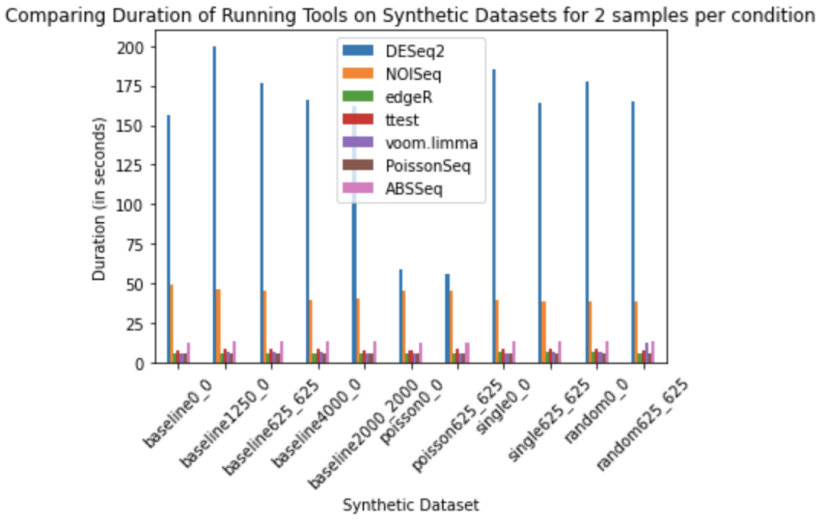



**Figure 1** The figure above shows the duration for each tool on each individual synthetic dataset with 2 samples per condition.

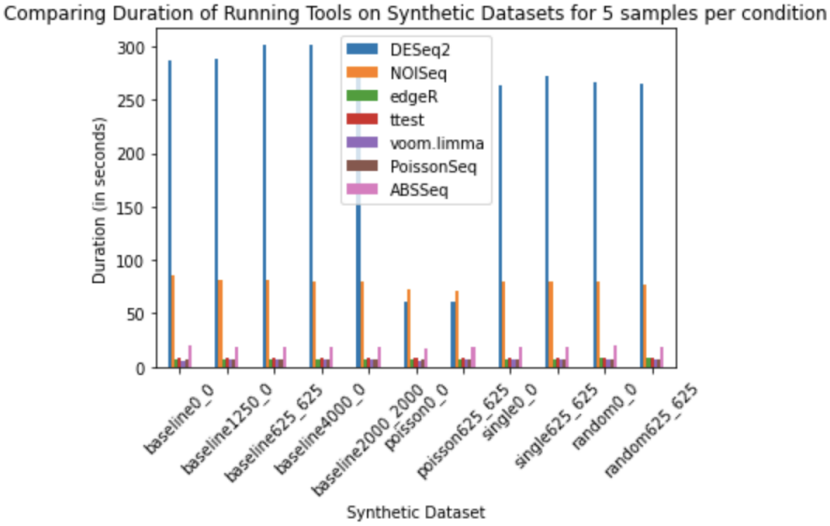



**Figure 2** The figure above shows the duration for each tool on each individual synthetic dataset with 5 samples per condition.

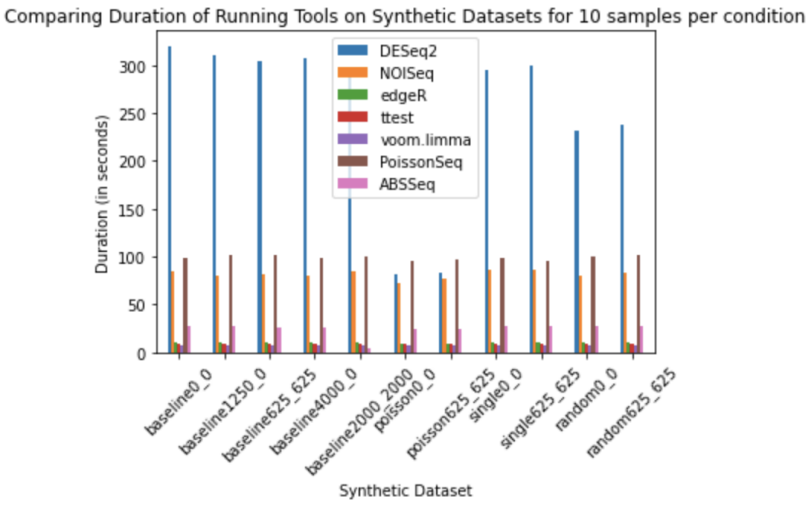
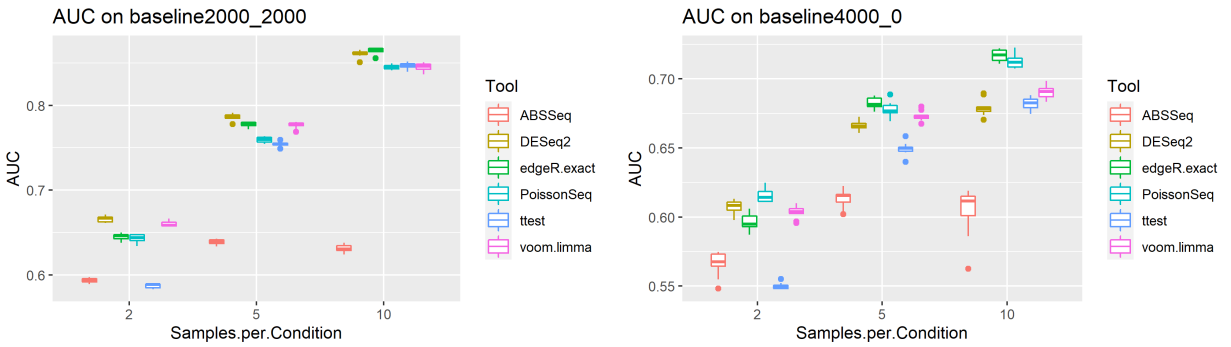



**Figure 3** The figure above shows the duration for each tool on each individual synthetic dataset with 10 samples per condition.

Based on the three graphs, it is portrayed that as the number of samples per condition increases from 2, 5, and 10, the duration of the tools also increases in all synthetic datasets. This is reasonable as it requires more time for tools to perform their analysis when there are more samples per condition and thus more overall data. More specifically, DESeq2 and NOISeq require the most time to perform their analyses which could be explained by the process of how they execute their differential gene expression analysis compared to the other tools. DESeq2 performs an internal normalization for each gene across all samples using a geometric mean and then is divided by the mean. In addition, DESeq2 uses shrinkage estimation for dispersions & fold changes, so a dispersion value is estimated for each gene which could contribute to why it takes a long period of time[15]. Similarly, NOISeq creates a noise distribution of count changes by comparing the number of reads per gene in all samples with all the same conditions. It then uses the distribution to assess whether the change in count number is most likely a noise or truly DE gene[16]. Hence, it takes a longer duration to complete than other tools in which they use standard normalization methods (i.e., Trimmed Mean of M-values) or don't use a noise distribution to predict whether a gene is DE or not. Additionally, for both tools, they filter low-counts from their DEA, which could also contribute to why they take longer to perform than the other tools that don't filter out low-counts during normalization.

## Overall Trends




**Figures 4a and 4b** The figures above shows boxplots of the values of AUC on $B_{2000}^{2000}$ and $B_{0}^{4000}$ respectively. In both, AUC rises significantly when samples per condition rises but the values in Figure 4a are both greater and less variable than in Figure 4b though the number of total genes differentially expressed are the same.

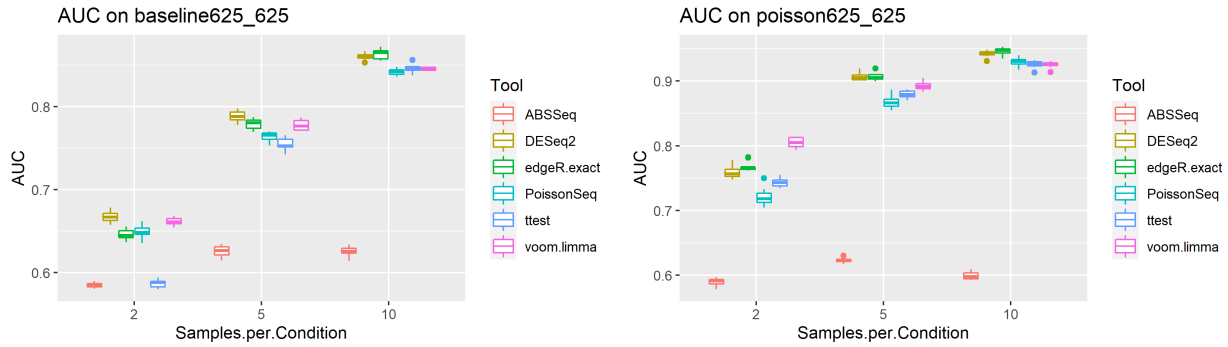
Across all datasets and tools, several trends seemed to be universal. For one, in almost every single graph outside of specific statistics on $B_{0}^{4000}$, performance increased significantly with the increase in samples per condition. In Figure 4a, we see a clear difference in every tool's average values (except ABSSeq) with around a 30% increase in AUC between 2 samples per condition and 10 in the rest of the tools. This increase is seen in the greatest magnitude when looking at sensitivity, which indicates that increasing the samples per condition increases a tool's ability to correctly classify truly DE genes. In general, these results are primarily intuitive as more samples per condition mean more data that each tool can use to identify genes.

One other trend we found was the effect of the proportion of up and down-regulated genes. In our experiment we set up our data to examine this in two different instances: between $B_{625}^{625}$ and $B_{0}^{1250}$ and between $B_{2000}^{2000}$ and $B_{0}^{4000}$. Interestingly, the results between $B_{625}^{625}$ and $B_{0}^{1250}$ were nearly identical in almost every statistic. On the other hand, there was a huge disparity across the board between $B_{2000}^{2000}$ and $B_{0}^{4000}$ datasets. In Figure 4b we not only see around a 10% decrease in AUC compared to Figure 4a, but in almost every case, the variance is greater, and the results are less consistent across dataset versions. This indicates that an uneven number of up and downregulated negatively impacts the accuracy of results when the difference between them is greater. One possible reason for this is that most of the tool normalization factors do not assume such an imbalanced distribution, leading to more false positives. Real-life RNA data is rarely

built like $B_0^{4000}$ in which ⅓ of the genes are all upregulated, so it stands to reason that the tools would have a more challenging time distinguishing between DE genes in this situation.

For the full set of graphs on each statistic and dataset, please see the appendix at the end of the report.

## Area under the ROC Curve (AUC)



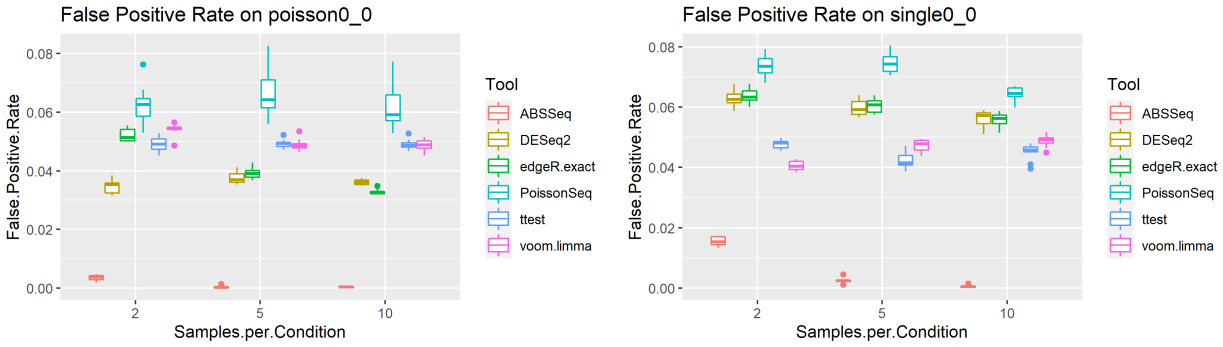

**Figures 5a and 5b** The figures above shows boxplots of the values of AUC on $B_{625}^{625}$ and $P_{625}^{625}$ respectively.

_____The ROC curve is a popular graphical tool for comparing the performance of multiple classifiers. The name “ROC" is an acronym for receiver operating characteristics, which comes from communications theory. The overall performance of a classifier is given by the area under the ROC curve (AUC). So, the larger the AUC, the better the classifier. We could observe that DESeq2, edgeR, voom.limma, ttest and PoissonSeq have fairly similar AUC results, having a difference that maxes out at 0.1 depending on the dataset being analyzed. ABSSeq performed the worst across all datasets, resulting in AUC lower than 0.65 even with the datasets with 10 samples per condition. This makes ABSSeq an inferior tool to use if AUC is an important metric to take into account while working for a project, as it will only produce poor or worthless AUC results.

Across the tools that performed well, edgeR excelled in datasets with the most samples per condition excluding those in $S_{625}^{625}$ and $R_{625}^{625}$. In these two datasets, DESeq2 and voom.limma performed far better. However, when there were fewer samples per condition, DESeq2 produces the best AUC results for most datasets. It is worth noting that PoissonSeq performed worse than the other tools, even in the $P_{625}^{625}$ dataset. This is surprising, since PoissonSeq performs differential

gene expression analysis using a Poisson log-linear model. Also, the AUC of all the tools performs less well when performed on the single and random outlier dataset.

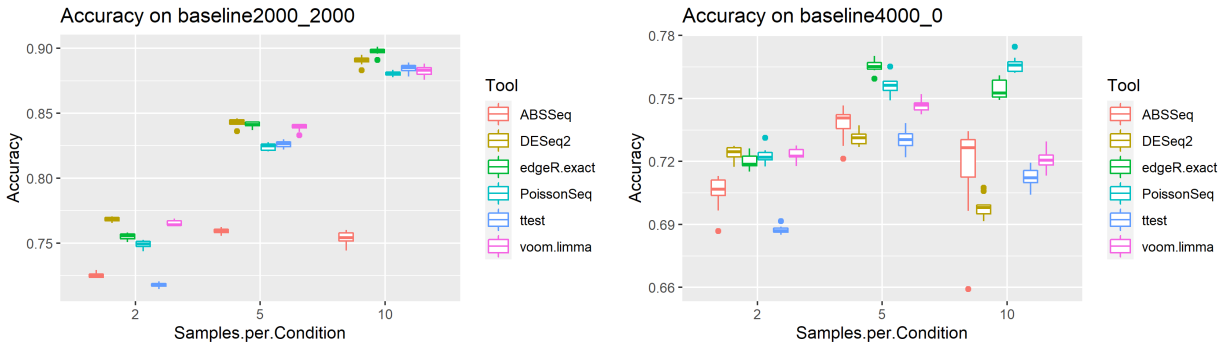
## Type I Error Rate




**Figures 6a and 6b** The figures above shows boxplots of the values of Type I Error Rate on $P_0^0$ and $S_0^0$ respectively which both show a clear difference in rank between ABSSeq and the other tools.

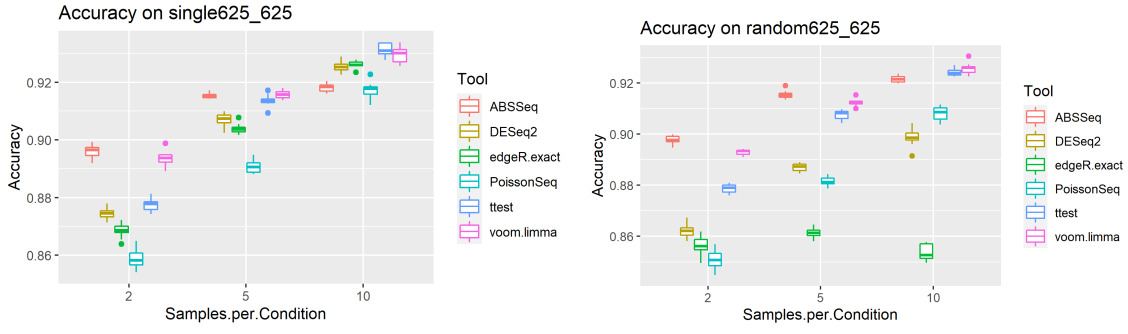
Type I Error Rate, or False Positive Rate, calculates the proportion of genes incorrectly identified as differentially expressed by dividing the number of false positives by the total number of non-differentially expressed genes. This metric was calculated on the datasets with zero truly differentially expressed genes that were meant to emulate circumstances where differentially expressed genes were expected to be rare. Thus, false positives would need to be controlled. ABSSeq performs the best in this metric in particular, as it has very low Type I Error rates compared to the rest of the tools. This could indicate that ABSSeq is more useful when there are fewer differentially expressed genes, since it has a higher threshold for classification. Most tools average below 10% false positive rate, which is an acceptable threshold, yet there were still some note trends. Like sensitivity, samples per condition did not have as drastic of an effect on performance as it did with other tools. In some cases, the data with larger samples per condition had higher average false positive rates. Also, PoissonSeq was the most unstable amongst the tools, as it performed with high variance in different datasets and had the highest rank across almost every dataset. On the contrary, voom.limma and ttest had low variances and performed better on the datasets with outliers ($S_0^0$ and $R_0^0$) compared to DESeq2and EdgeR, which ranked better only on $P_0^0$.

## Accuracy




**Figures 7a** and **7b** The figures above shows boxplots of the values of Accuracy on $B_{2000}^{2000}$ and $B_{0}^{4000}$ respectively.

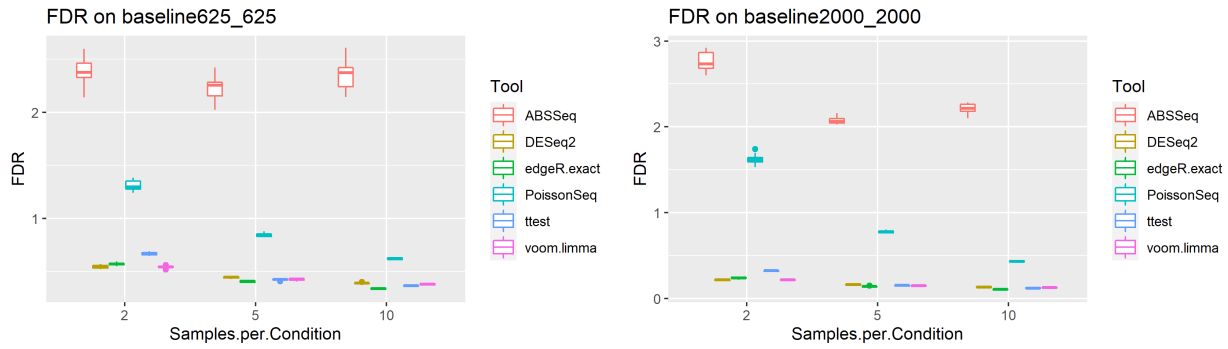
In general, DESeq2 and edgeR would perform similarly, in terms of accuracy, in all datasets (including the Poisson dataset), except one differentiating point is how DESeq2 would not perform nearly as well as edgeR when there are more genes upregulated in condition 2 compared to condition 1. As shown in the left graph above (**Figure 7a**), DESeq2's accuracy would fall short of edgeR about ~0.02, proving that edgeR is a better statistic than DESeq2 when accounting for accuracy of predicting truly DE genes, but DESeq2 would also be a viable tool. However, in the right graph above (**Figure 7b**), DESeq2's accuracy cannot compare to edgeR as it performs worse than almost all tools when there is an uneven distribution of genes upregulated in condition 2 compared to condition 1. This discrepancy could be due to the fact that DESeq2's normalization method uses the geometric mean of the genes of the samples in which the tool does not take into account how a sample could have more upregulated/downregulated genes than the other condition.




**Figures 8a and 8b** The figures above shows boxplots of the values of Accuracy on $S_{625}^{625}$ and $R_{625}^{625}$ respectively.

It is worth noting that while ABSSeq did not perform well on the baseline & Poisson generated datasets, ABSSeq performed better than almost all tools when handling datasets with a single outlier or random outlier with unusually high counts. As shown in the left graph above (**Figure 8a**), ABSSeq would perform the best when there were 2 or 5 samples per condition, but would produce subpar results when there are 10 samples per condition. Based on the other metrics, the previous trend would show that as the number of samples per condition increases, the more the metric would increase. In this case, however, ABSSeq is lower-ranking than the other tools with the most number of samples per condition, reinforcing the idea that this tool is substandard for DEA, as it can only perform better than other tools when there are only 2 or 5 samples per condition. Further, in the graph on the right (**Figure 8b**), ABSSeq performs the best when there are 2 or 5 samples per condition, but is outperformed by both ttest and voom.limma when there are random outliers. So, voom.limma or ttest should be utilized in the real life setting instead of ABSSeq because outliers are usually random in a real-life dataset. Researchers typically work with datasets that contain greater than 10 samples per condition for more precise results, so these findings prove that ABSSeq is only correct when there are a few samples per condition which are not realistic.

## False Discovery Rate




**Figures 9a and 9b** The figures above shows boxplots of the values of False Discovery Rate on $B_{625}^{625}$ and $B_{2000}^{2000}$ respectively.


**Figures 10a and 10b** The figures above shows boxplots of the values of False Discovery Rate on $S_{625}^{625}$ and $R_{625}^{625}$ respectively.

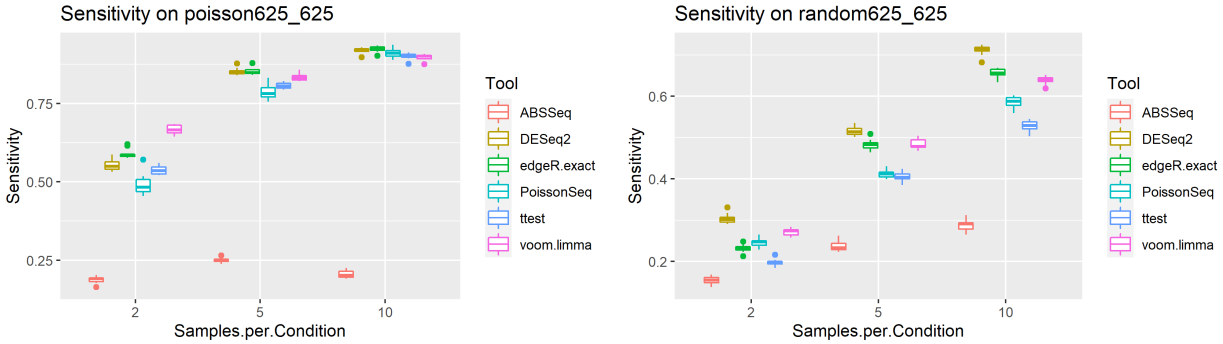
In all graphs, DESeq2, edgeR, voom.limma, and ttest, all rank around 0.5 for false discovery rate with a small variance, which means that these tools are good at predicting whether a gene is differentially expressed. When the differentially expressed genes were regulated in different directions, increasing the number of DE genes from 1,250 to 4000 (i.e., $B_{625}^{625} \rightarrow B_{2000}^{2000}$), FDR would be controlled and decreased (**Figures 9a & 9b**). On the other hand, for instances where DE genes were regulated in the same directions (i.e., $B_{0}^{1250} \rightarrow B_{0}^{4000}$) had no influence on the control for FDR; it did not increase or decrease. However, there is a similar trend occurring here where DESeq2 and edgeR will perform better than voom.limma & ttest except in cases where there is an outlier (**Figures 10a & 10b**). This strengthens the fact that voom.limma & ttest are better tools when there are outliers with abnormally high counts.

In all synthetic datasets, ABSSeq has a high false discovery rate which means that it doesn't perform nearly as well as the other tools in terms of predicting whether a gene is truly differentially expressed. Furthermore, almost all tools portray low variance in all graphs, except for ABSSeq, which demonstrates how ABSSeq may not be performing analysis correctly most of the time because of how spread out the data points are from the mean. In addition to the graphs portraying the other metrics, it seems as if ABSSeq performs the worst compared to the other tools. This is reasonable as ABSSeq is a new RNA-Seq analysis tool that has recently emerged[14].

Similar to ABSSeq, PoissonSeq doesn't perform as well as the other tools (DESeq2, edgeR, voom.limma, & ttest) as it received False Discovery Rates higher than 0.5, but still performed better than ABSSeq. However, there is a unique trend here where the FDR would

decrease as the number of samples increased for PoissonSeq. This makes sense as PoissonSeq uses a log-linear model to calculate a score statistic for differential gene expression, so the more samples the model has, the better the tool would perform. Consequently, it doesn't perform as well on the Poisson distributed synthetic dataset, which is surprising because PoissonSeq performs differential gene expression analysis using a Poisson log-linear model.

## Sensitivity and Specificity




**Figures 11a and 11b** The figures above shows boxplots of the values of Sensitivity on $P_{625}^{625}$ and $R_{625}^{625}$ respectively.

Figures 11a and 11b look at sensitivity, which measures the tools' ability to correctly identify the truly differentially expressed genes. Sensitivity is calculated by dividing the number of genes correctly identified as DE by the total number of predictions of DE genes. As with most other statistics, the best performing tool overall was DESeq2, with edgeR close behind. Again, we saw better sensitivity performance with greater samples per condition, and in general, the difference between the tools was significantly smaller in the datasets with 10 samples per condition. In all datasets, all of the tools (except ABSSeq) had results typically being within 0.05 of each other. However, this was not the case for the $R_{625}^{625}$ dataset, which had more significant disparities between tools ranks. Unlike in our other graphs, voom.limma and ttest did not show a significant relative increase in performance in the datasets with outliers. However, we did see a stronger performance with PoissonSeq on the $P_{625}^{625}$ dataset compared to the other datasets in which it ranked on the lower half of the tools. Overall, there were no trends seen in the sensitivity graphs that were not already depicted in previous graphs, which gauge total performance and reliability, such as AUC.


**Figures 12a and 12b** The figures above shows boxplots of the values of Specificity on $B_{625}^{625}$ and $B_{0}^{4000}$ respectively.

Specificity is a measure of how well each tool correctly identifies non-differentially expressed genes. Specificity is calculated by dividing the number of truly non-differentially expressed genes by the total number of predicted non-differentially expressed genes. Unsurprisingly, ABSSeq was by far the best tool in this metric. As we have seen previously, ABSSeq typically classified genes as non-differentially expressed at a much higher rate than other tools. In almost every dataset, the average specificity from ABSSeq was between 0.97 and 1, especially with greater samples per condition. Outside of ABSSeq, each tool's performance was very comparable in most cases, only different by an average of 0.005 between them. Also, PoissonSeq was noticeably worse compared to the rest of the tools. Interestingly, unlike some of the other statistics, outside of ABSSeq and, to a lesser extent, edgeR, there was no increase in performance with an increase in samples per condition. This means that increasing the number of samples won't significantly affect specificity in most cases. In fact, for the $B_{0}^{4000}$ dataset, the performance actually worsened with the increase of number of samples, which could be due to normalization methods as mentioned earlier.

Again we saw a noticeable increase in performance of voom.limma and ttest in the datasets with outliers compared to DESeq2, edgeR, and PoissonSeq. However, much like the accuracy graphs, DESeq2 and edgeR had higher ranks in specificity for the $P_{625}^{625}$ datasets.

Overall, it seems like DESeq2 and edgeR are the clear choices to use when sensitivity is significant to control. Still, in cases where researchers need to be sure that genes are not differentially expressed, ABSSeq may be a viable option. Even in the data where most genes

were differentially expressed, ABSSeq controlled specificity the best and seems to be the most useful when the true number of differentially expressed genes is lower.

## Real Life Data

Using the RNA-Seq data on the three psychiatric disorders (schizophrenia, bipolar disorder, and major depressive disorder) from last quarter, we also performed these tools on the gene matrix count we generated the previous quarter.

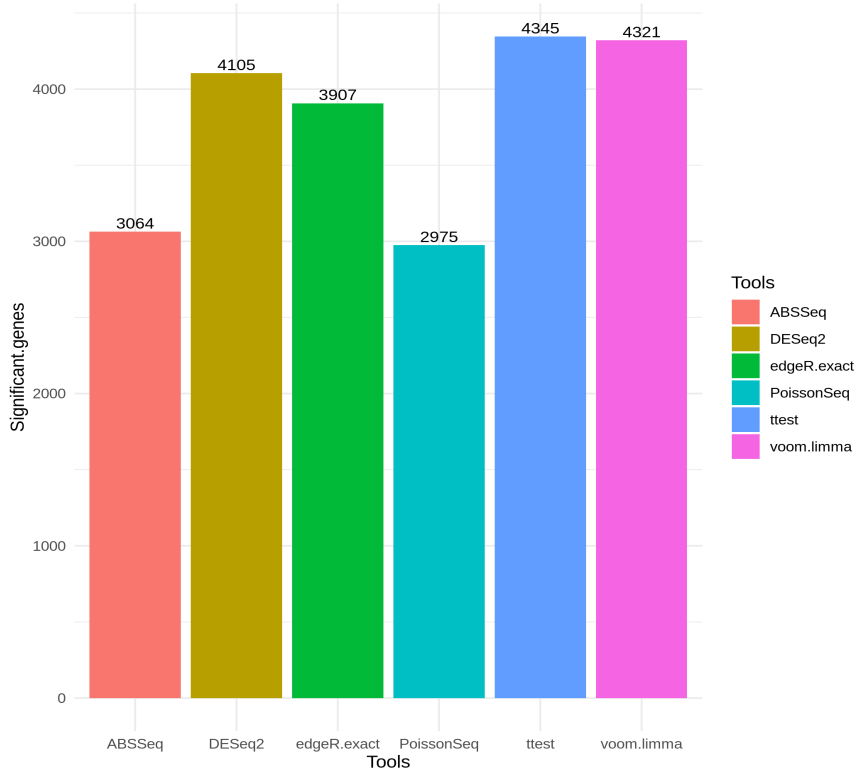
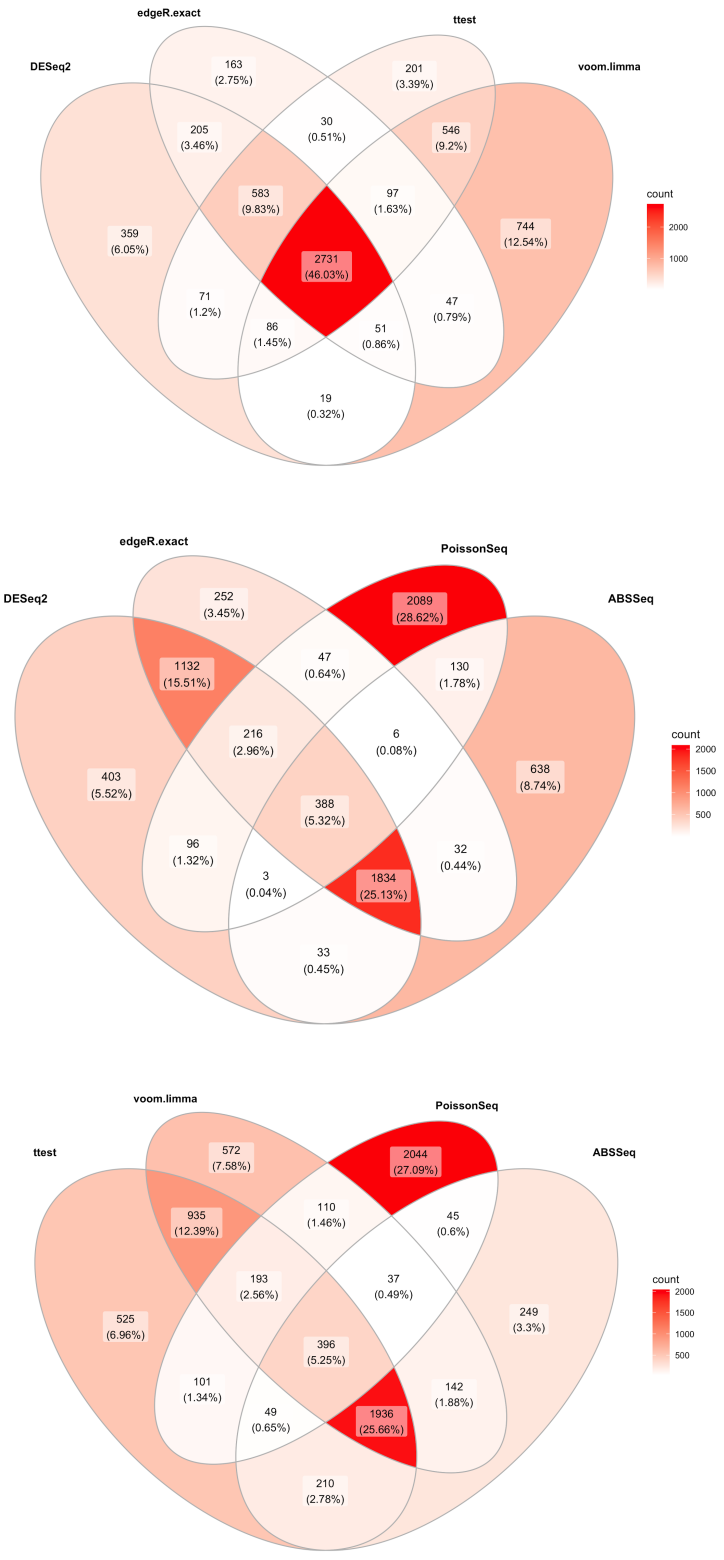



**Figure 13** The figure above shows boxplots of the number of genes of Schizophrenia patients that each tool yielded as differentially expressed.




**Figures 14a, 14b, 14c.** The figure above shows venn diagrams that represent the overlap among the set of DE genes found by different methods in the Schizophrenia section.

First, we compared the number of DE genes found by each tool (**Figure 13**). Depending on the psychiatric disease, the actual number differed, but we could see a general trend of ttest and voom.limma being the highest, while ABSSeq had the fewest among all diseases. It can also be noted that DESeq2, edgeR, ttest and voom.limma had very similar results on all diseases, so it is hard to make conclusions about which genes performed better than others since we don't know the true DE genes. The results produced here correlate to the results found in the simulated datasets where DESeq2 and edgeR would perform similarly, whereas voom.limma and ttest would also function similarly. Although voom.limma & ttest identified more DE genes here, our findings with the simulated datasets show that DESeq2 and edgeR should be more precise in their statistical analysis of finding truly DE genes.

Then, we studied the overlap of genes classified as differentially expressed among the tools (**Figure 14**). We have noticed that tools that used negative binomial distribution (i.e., DESeq2, edgeR, ttest, and voom.limma) shared the most similar results (**Figure 14a**) in which they identified the same 2,731 genes DE. For most of the datasets, more than half of the gene results yielded by such tools were classified as DE. On the other hand, we could see a large discrepancy between tools that did not use negative binomial distribution. In particular, PoissonSeq and ABSSeq tend to classify genes to be DE that most other tools refuted. As we mentioned earlier, we do not know the actual parameters of the real-life dataset. Hence, we are unable to guarantee which genes are truly DE, and can only note the similarities and differences between the results of our tools.

## Discussion

Generally, across all graphs, it can be observed that as sample size increases, so does the accuracy, which makes sense as the more samples we have per condition, the more data we have to conclude from. Additionally, we observed that DESeq2 and edgeR perform roughly the same in cases where there is a ratio upregulated/down-regulated in both samples (i.e., $B_{625}^{625}$, $B_{2000}^{2000}$, etc.). This makes sense, since both tools are very similar in that they both assume that no genes are differentially expressed. However, for cases where more genes are upregulated in one condition than the other (i.e., $B_{0}^{4000}$), edgeR would typically perform better. As mentioned prior, DESeq uses a "geometric" normalization strategy. In contrast, edgeR uses a weighted mean of

log ratios-based method, which means both normalize using size/normalization factors. There is a possibility that edgeR's normalization method performs better when there is an uneven distribution of upregulated/downregulated genes.

Moreover, it seems that in all graphs where there are no outliers with randomized counts, DESeq2 and edgeR perform better than the other tools, but for cases where there are outliers (random & single), voom.limma and ttest would perform better than these two tools. However, it was interesting to see that edgeR would perform worse than DESeq2 in cases where there are outliers because edgeR is known to be a software that can handle random outliers. In all cases, it looks like voom.limma and ttest would perform the same, which could be explained by the fact that they use the ttest to perform differential gene expression, but voom.limma performs better since it doesn't use the same process of DEA as ttest. Voom.limma uses voom transformation from the limma package, then a linear model for all gene expression. This is more thorough in contrast with ttest's, which just performs DEA after normalization.

According to the findings for all metrics, PoissonSeq did not perform well on the Poisson simulated dataset, which is surprising because we assumed that the Poisson log-linear model for the tool would predict the true DE genes from counts drawn from a Poisson distribution. However, we believe that PoissonSeq may not have performed nearly as well as we thought because we generated 12,500 genes, but only 6,250 were drawn from the Poisson distribution. So, we concluded that PoissonSeq's Poisson log-linear model's method would only perform well on the Poisson distribution only if almost all or most of the counts were drawn from the Poisson distribution. Overall it is not surprising that PoissonSeq did not do very well as it is a relatively old tool that is not regularly used and utilizes unconventional normalization methods. Therefore, the methods used by edgeR & DESeq2 may be better for DEA regardless of the distribution of the counts.

As mentioned prior, ABSSeq performed relatively poorly across all metrics even though it seems like a reliable tool that uses absolute count differences between 2 conditions. It is interesting to note that this method is a brand new approach by researchers as DEA is usually performed based on a distribution. In its poor performance, it could be that it used qtotal for normalization instead of Trimmed Means of M-values like most of the other tools did. PoissonSeq also did relatively poorly, and it too used a different normalization method unique to that tool. It could be that normalization has a more significant effect on the results, especially

since ABSSeq was similar to DESeq2 and edgeR in statistical methods as they all use a negative binomial distribution. Given that this tool is new, there should be more thorough research on ABSSeq on real-life or distinct synthetic datasets to see if it will continue to perform poorly or if this investigation is just a fluke. Although this tool doesn't utilize methods that have been the norm, it is still important to investigate how this tool performs as it has potential in DEA.

In all metrics shown in Appendix, NOISeq was not included as an investigation because the outputs of NOISeq on the synthetic data did not include p-values of each gene. We cannot make conclusions about significant differences. As a result, we could not make inferences about how NOISeq performs for DEA, but this could be something further studied by performing separate, independent research for this tool.

Another limitation to our study is that we used simulated data in which we cannot control for covariates of our samples, which many of our tools use to implement their models for performing DEA. For instance, DESeq2 allows users to build full models or reduced models for DEA. Still, our simulated datasets do not specify the gender or age of our samples, so we cannot use covariates as functions of our model, which could have produced different results. This is one major problem of our research as real-life datasets usually have information about the samples used for analysis, so our results reflect what is produced by the percent of aligned reads and do not consider the possible confounding variables.

## Conclusion

As expected, no one software performed better than the others in all of the metrics we measured, further proving that researchers need to be particular when choosing what tool to use in a given study. As more general and all-encompassing statistics, the results on AUC and accuracy show that DESeq2 and edgeR are solid choices for most studies. However, if runtime is a major factor, as can be the case, as RNA extraction becomes cheaper and more available, edgeR may be the better choice since, on average, it took many factors less time (~2-3 minutes) than DESeq2. Other tools such as voom.limma and ttest performed exceptionally well on datasets generated with the Poisson distribution instead of the negative binomial distribution, which is used to simulate variation in the input data. This could indicate that it may be better to use these tools when the data is known to have similar variability. Finally, ABSSeq, which was by far the worst in almost all other circumstances, proved to be the best in terms of False Positive

Rate in the datasets with zero differentially expressed genes. This shows that ABSSeq could be more valuable when the expected number of differentially expressed genes in a study is low since it is more selective when differentiating between genes, leading to lower false positives.

Across all experiments, one trend that we noticed was that an increase in samples per condition drastically increased every tool's performance in every statistic. While in a given experiment, it is not always feasible or practical to increase the number of samples, we can still prove how much more accurate results can be when the number of samples is increased. Specificity appeared to be the most affected by samples per condition, with the average specificity across tools almost doubling between 2 and 10 samples. There was still a noticeable difference in other statistics, often increasing by close to 50% with a sample size of 10 compared to 2.

Overall while our research found similar results to previous studies in the past[13], there is still much more that can be explored in this area especially as it continues to grow and be improved upon. In our research, we only used default parameters for most of the tools to perform differential expression analysis as a starting point. For example, DESeq2, we utilized a Wald test to perform statistical analysis. For further investigation, more research should be performed on how the tools would perform if the parameters were tuned differently from what was discussed here since some tools may have parameters that are documented to perform better in particular situations that we simulated in our synthetic data. For instance, next time, we can delve deeper into how the results would alter if we utilized a Likelihood Ratio Test instead. In addition, our findings were primarily observational as it is difficult to quantify and understand why exactly different software produce varying results based on the complexity of the statistical methods used across tools. The cleanliness of the synthetic data also needs to be considered when thinking about the results because real-life RNA sequencing data is much messier in practice, with many steps being required to even get to differential expression, each with its variability that can affect results. While we used the random and single outlier functions to attempt to simulate this idea, it may not necessarily translate the same way in practice. It would be preferable to test against even more types of diverse real-life datasets to validate our results. Finally, there are, of course, dozens of other software that could be tested against. Still, we believe that the tools we chose represented an excellent mixture of those commonly used in other studies and some that take

different statistical approaches to observe how their use affects the results and helps guide decisions in the future on which tools to use.

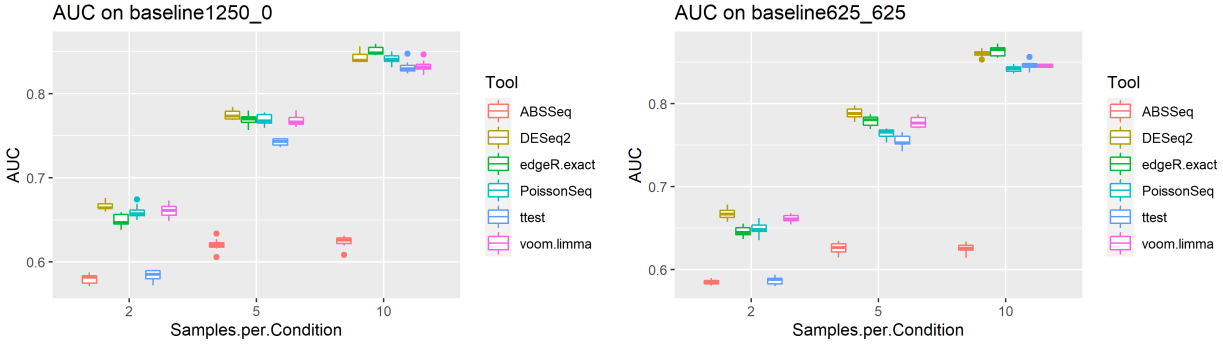
Works Cited


1. "ABSSeq." Bioconductor. Accessed March 12, 2021. https://bioconductor.org/packages/release/bioc/html/ABSSeq.html.
2. limma source: R/voom.R. Accessed March 12, 2021. https://rdrr.io/bioc/limma/src/R/voom.R.
3. "Package PoissonSeq." CRAN. Comprehensive R Archive Network (CRAN). Accessed March 12, 2021. https://cran.r-project.org/web/packages/PoissonSeq/index.html.
4. Michael I. Love, Simon Anders. Analyzing RNA-seq data with DESeq2, February 19, 2021. http://www.bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html.
5. "NOISeq." Bioconductor. Accessed March 12, 2021. https://bioconductor.org/packages/release/bioc/html/NOISeq.html.
6. "EdgeR." Bioconductor. Accessed March 12, 2021. https://bioconductor.org/packages/release/bioc/html/edgeR.html.
7. Yunshun Chen, Aaron TL Lun. exactTest: Exact Tests for Differences between Two Groups of... in edgeR: Empirical Analysis of Digital Gene Expression Data in R, January 16, 2021. https://rdrr.io/bioc/edgeR/man/exactTest.html.
8. Wang, Zhong, Mark Gerstein, and Michael Snyder. "RNA-Seq: a Revolutionary Tool for Transcriptomics." Nature News. Nature Publishing Group. Accessed March 12, 2021. https://www.nature.com/articles/nrg2484.
9. "CompcodeR." Bioconductor. Accessed March 12, 2021. https://bioconductor.org/packages/release/bioc/html/compcodeR.html.
10. "CompcodeR." function | R Documentation. Accessed March 12, 2021. https://www.rdocumentation.org/packages/compcodeR/versions/1.8.2/topics/generateSyntheticData.
11. Li, Jun, Daniela M. Witten, Iain M. Johnstone, and Robert Tibshirani. "Normalization, Testing, and False Discovery Rate Estimation for RNA-Sequencing Data." OUP

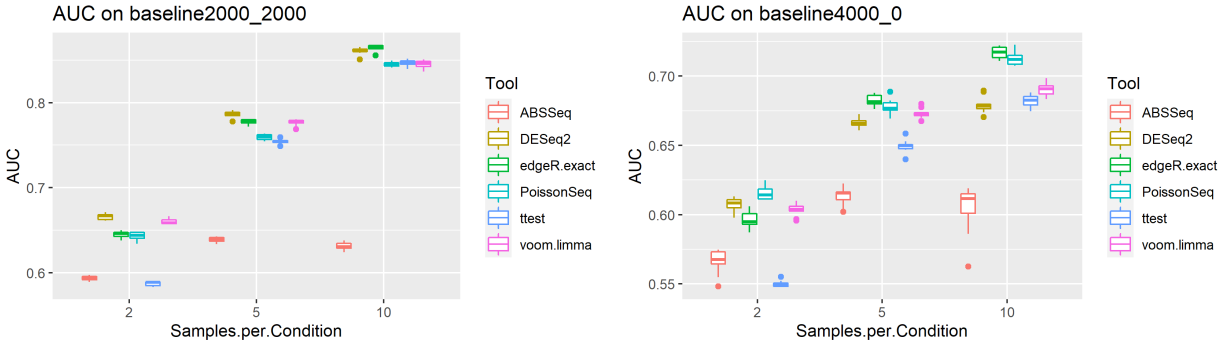
Academic. Oxford University Press, October 14, 2012. https://academic.oup.com/biostatistics/article/13/3/523/248016.

12. A. Mortazavi, BA. Williams, C. Wang G. Chen, MD. Robinson A. Oshlack, D. Koppstein A. Agarwal, Y. Hey JR. Bradford, E. Purdom JH. Bullard, W. Huber S. Anders, et al. "A Comparison of Methods for Differential Expression Analysis of RNA-Seq Data." BMC Bioinformatics. BioMed Central, January 1, 1970. https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-14-91.
13. S. Anders, W. Huber, KA. Kelly TJ. Hardcastle, DJ. McCarthy MD. Robinson, DM. Witten J. Li, L. Chen S. Srivastava, CA. Meyer J. Feng, M. Delorenzi C. Soneson, et al. "ABSSeq: a New RNA-Seq Analysis Method Based on Modelling Absolute Expression Differences." BMC Genomics. BioMed Central, January 1, 1970. https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-016-2848-2.
14. Chipster. Accessed March 12, 2021. https://chipster.csc.fi/manual/deseq2.html.
15. Tarazona, Sonia, Fernando García, Alberto Ferrer, Joaquín Dopazo, and Ana Conesa. "NOIseq: a RNA-Seq Differential Expression Method Robust for Sequencing Depth Biases." EMBnet.journal. Accessed March 12, 2021. http://journal.embnet.org/index.php/embnetjournal/article/view/265#:~:text=NOISeq%20is%20a%20non%2Dparametric,samples%20within%20the%20same%20condition.

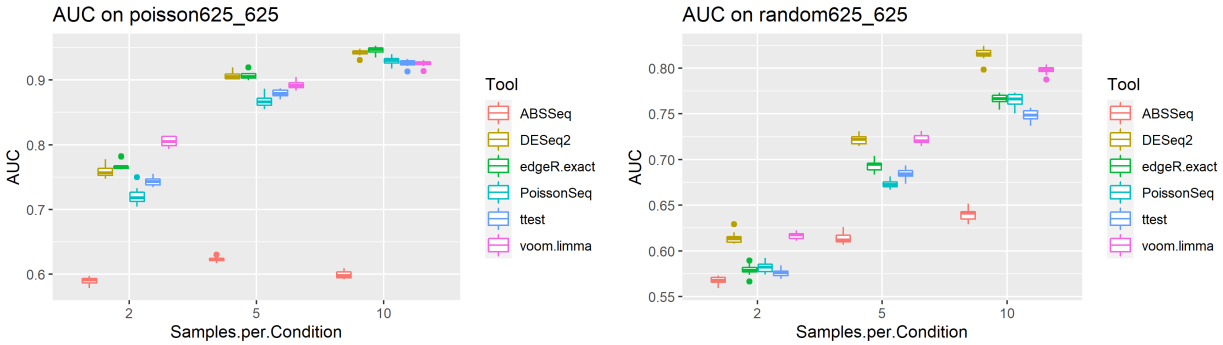
# *Appendix*


AUC on baseline1250_0
AUC
Samples.per.Condition
Tool
ABSSeq
DESeq2
edgeR.exact
PoissonSeq
ttest
voom.limma

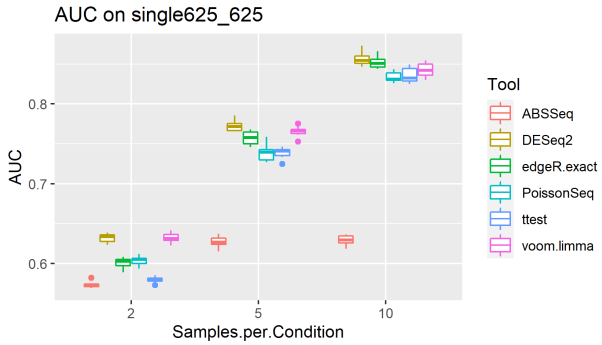

AUC on baseline625_625
AUC
Samples.per.Condition
Tool
ABSSeq
DESeq2
edgeR.exact
PoissonSeq
ttest
voom.limma


AUC on baseline2000_2000
AUC
Samples.per.Condition
Tool
ABSSeq
DESeq2
edgeR.exact
PoissonSeq
ttest
voom.limma

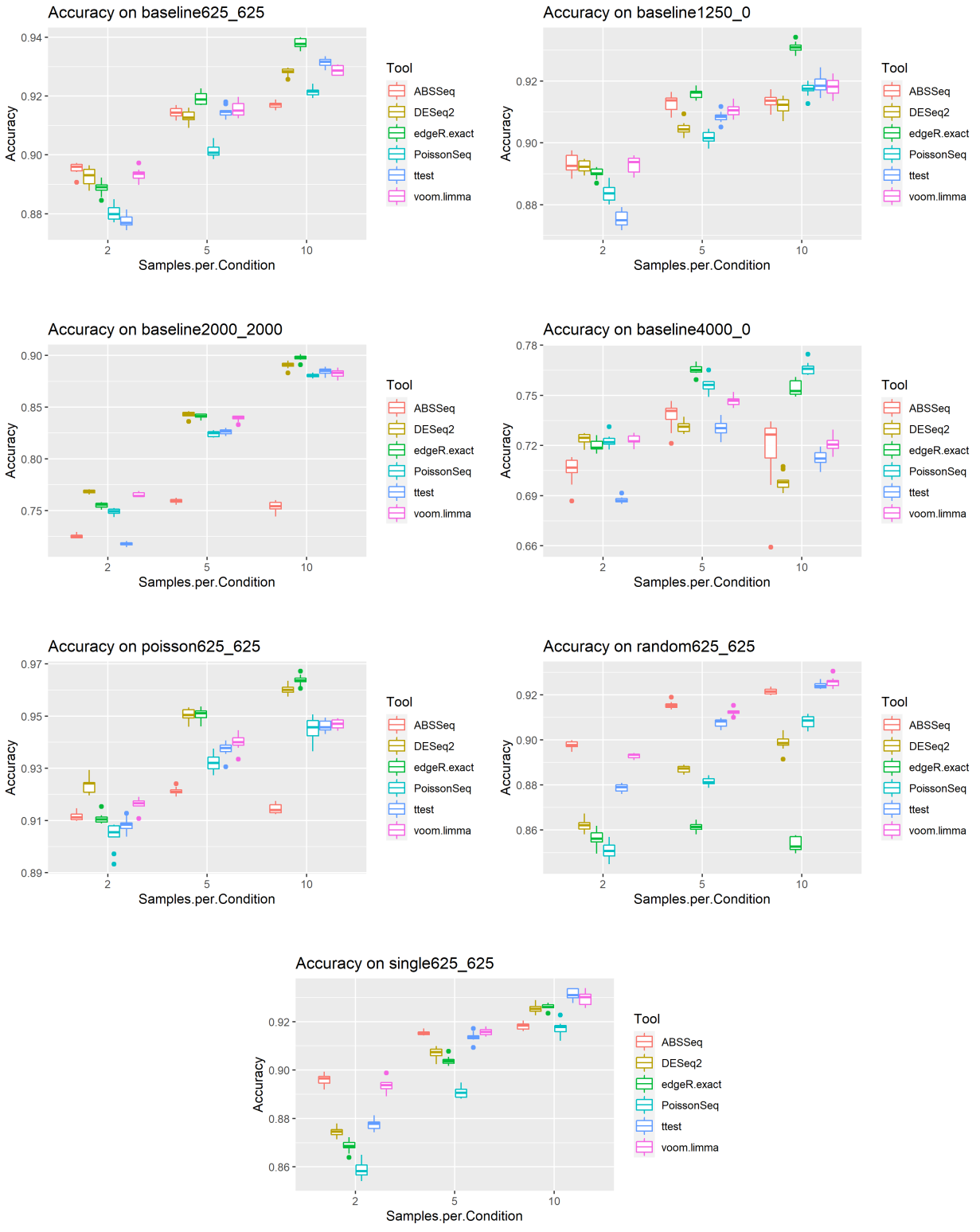

AUC on baseline4000_0
AUC
Samples.per.Condition
Tool
ABSSeq
DESeq2
edgeR.exact
PoissonSeq
ttest
voom.limma


AUC on poisson625_625
AUC
Samples.per.Condition
Tool
ABSSeq
DESeq2
edgeR.exact
PoissonSeq
ttest
voom.limma

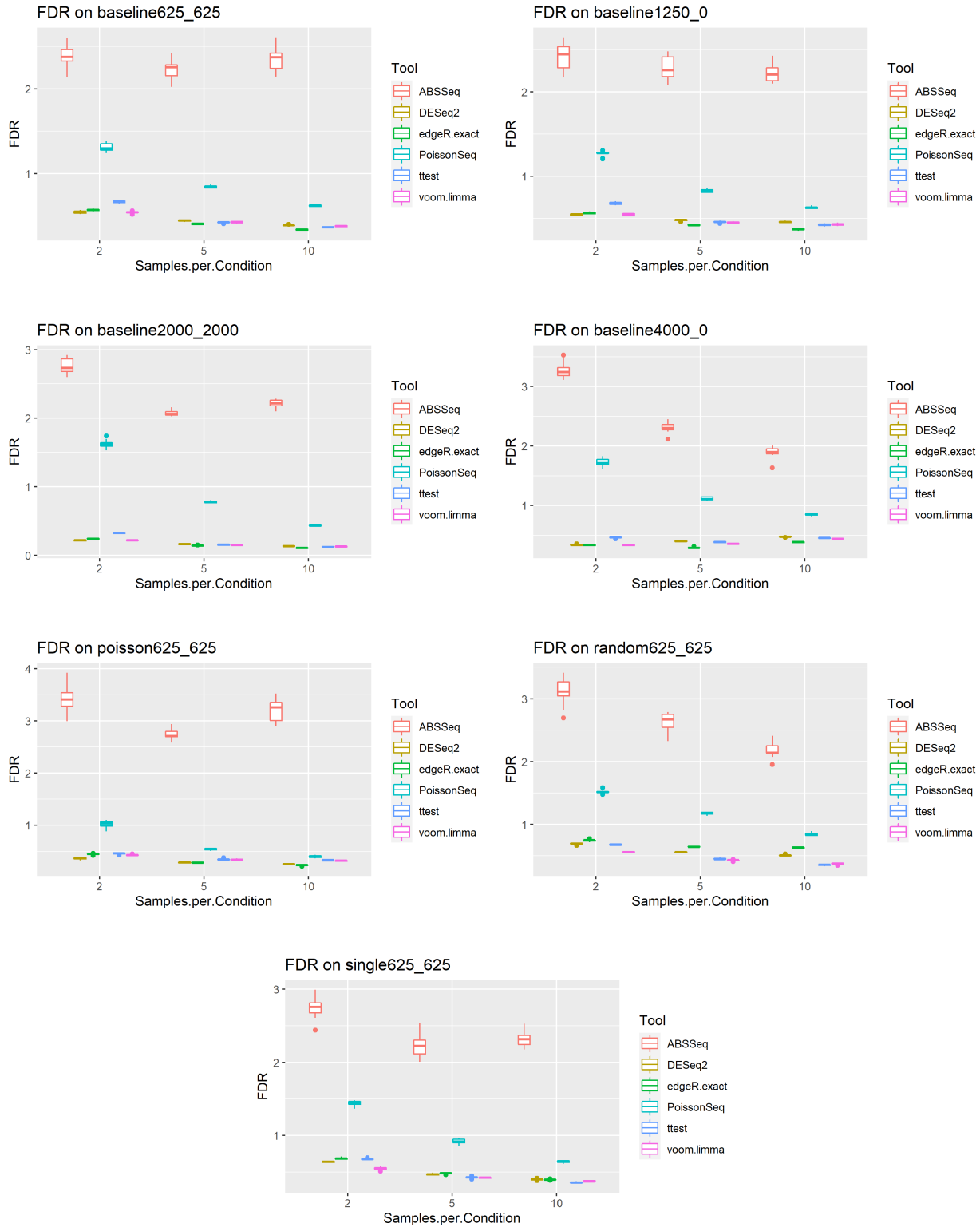

AUC on random625_625
AUC
Samples.per.Condition
Tool
ABSSeq
DESeq2
edgeR.exact
PoissonSeq
ttest
voom.limma


AUC on single625_625
AUC
Samples.per.Condition
Tool
ABSSeq
DESeq2
edgeR.exact
PoissonSeq
ttest
voom.limma

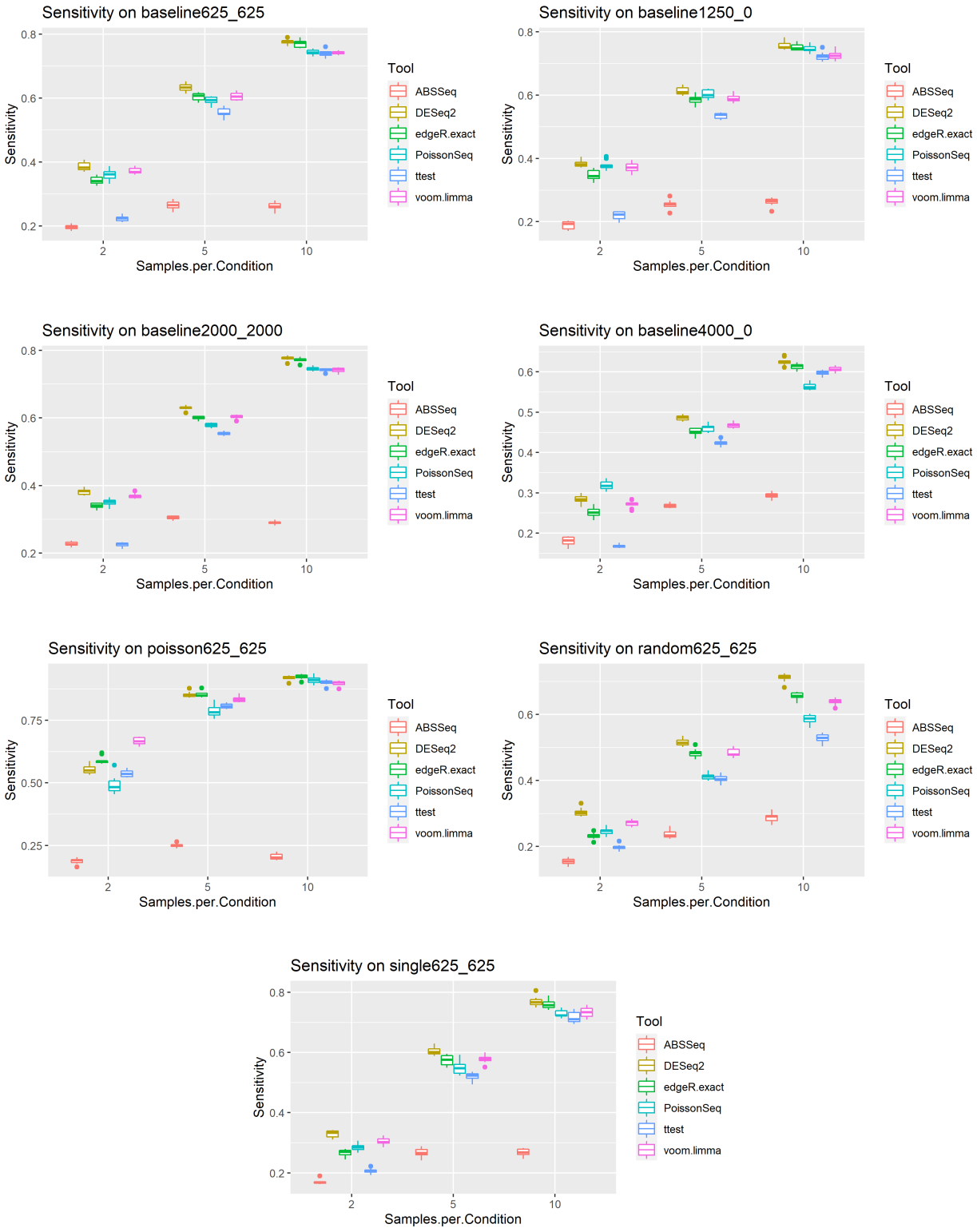
**Figures 15a, 15b, 15c, 15d, 15e, 15f, 15g.** The figure above shows the results of AUC on all the synthetic datasets .



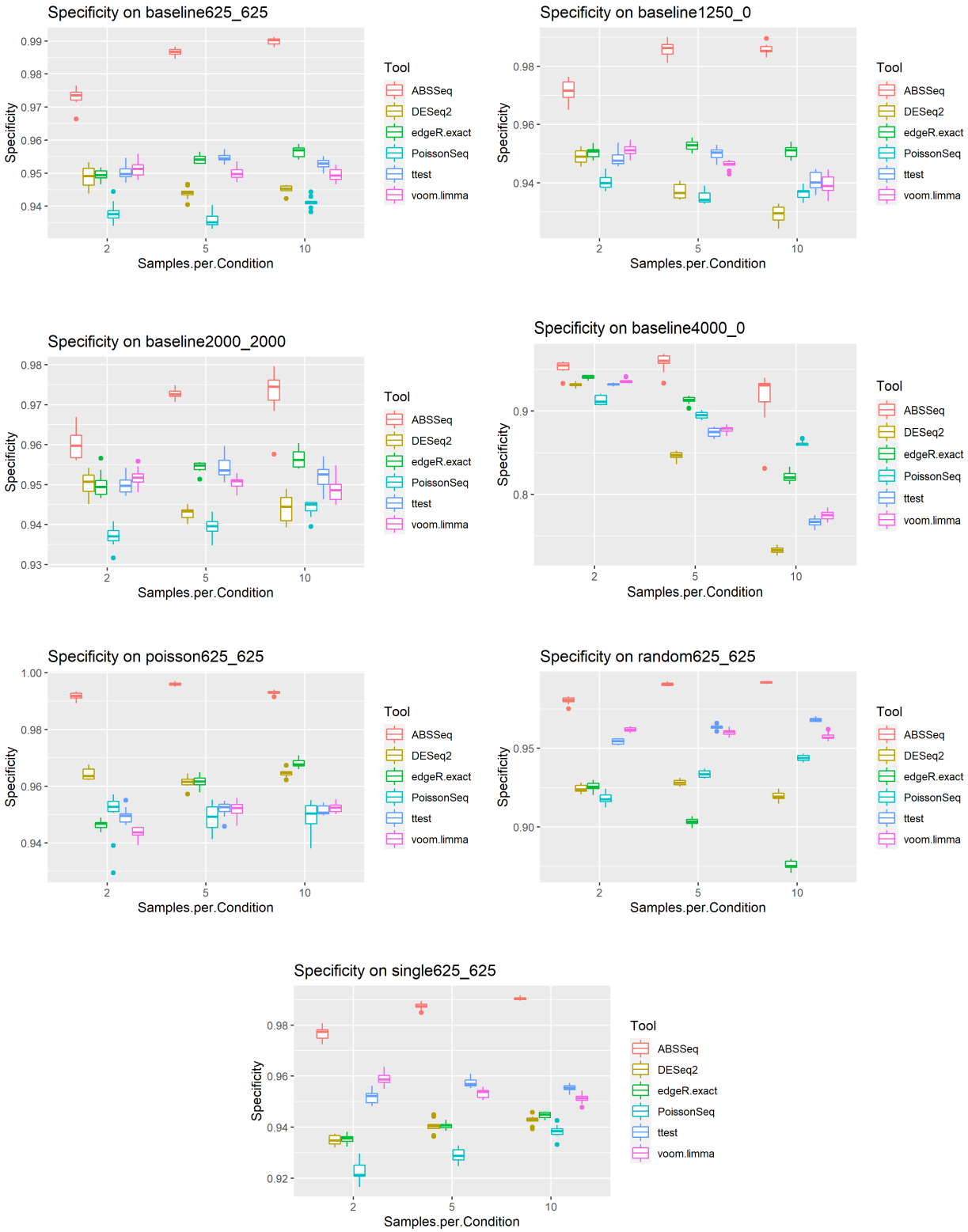

**Figures 16a, 16b, 16c, 16d, 16e, 16f, 16g.** The figure above shows the results of Accuracy on all the synthetic datasets .



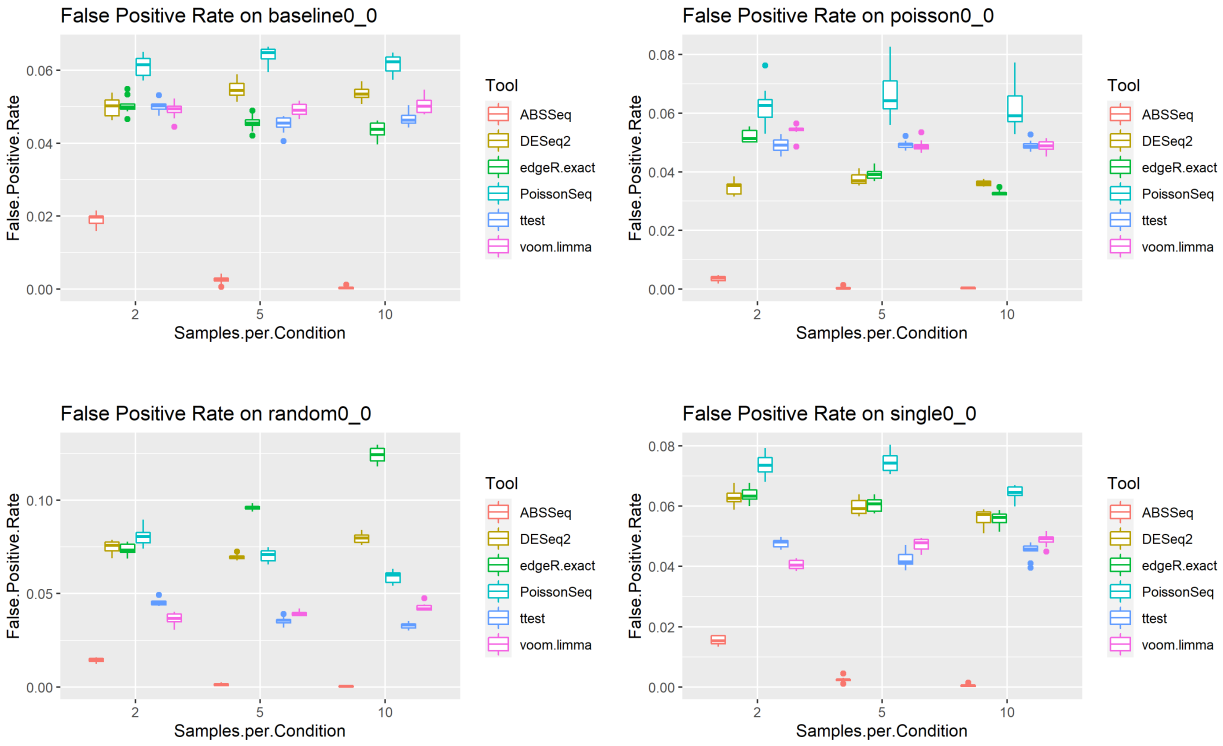

**Figures 17a, 17b, 17c, 17d, 17e, 17f, 17g.** The figure above shows the results of FDR on all the synthetic datasets .




**Figures 18a, 18b, 18c, 18d, 18e, 18f, 18g.** The figure above shows the results of Sensitivity on all the synthetic datasets .


Specificity on baseline625_625
Specificity on baseline1250_0
Specificity on baseline2000_2000
Specificity on baseline4000_0
Specificity on poisson625_625
Specificity on random625_625
Specificity on single625_625
Specificity
Samples.per.Condition
Tool
ABSSeq
DESeq2
edgeR.exact
PoissonSeq
ttest
voom.limma

**Figures 19a, 19b, 19c, 19d, 19e, 19f, 19g.** The figure above shows the results of Specificity on all the synthetic datasets .




**Figures 20a, 20b, 20c, 20d.** The figure above shows the results of False Positive Rate, or Type I Error Rate, on all the synthetic datasets .

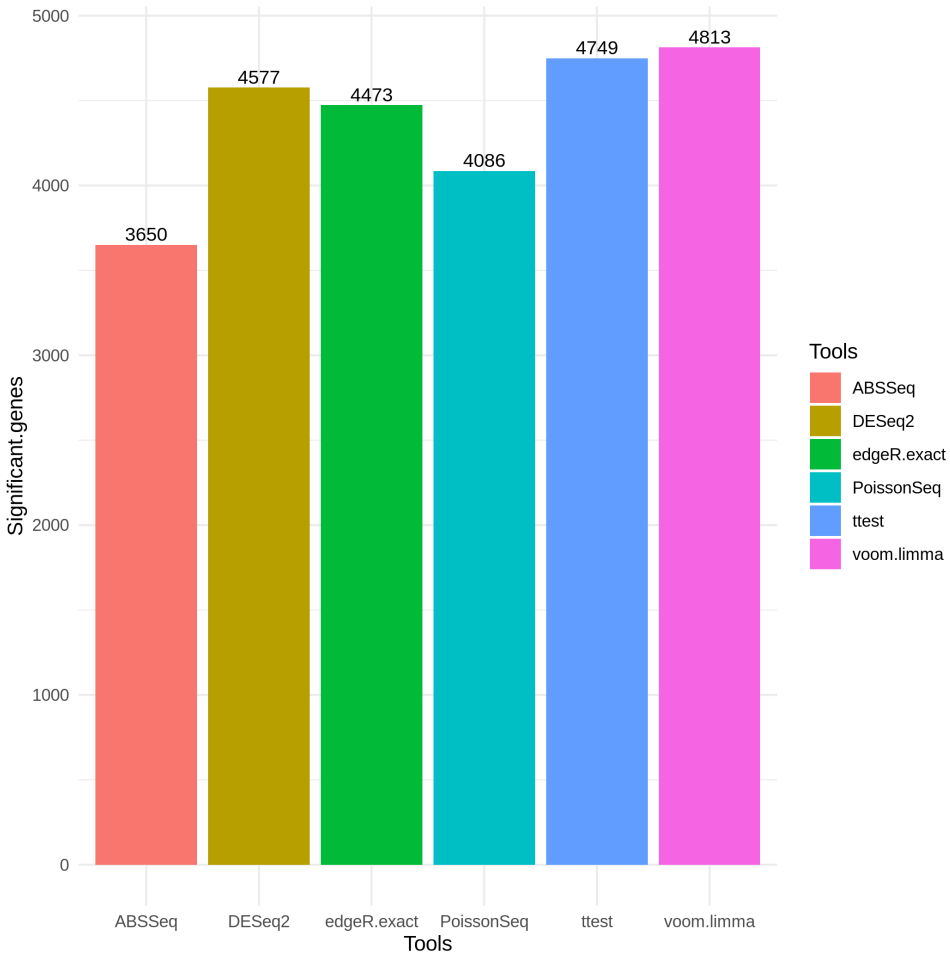
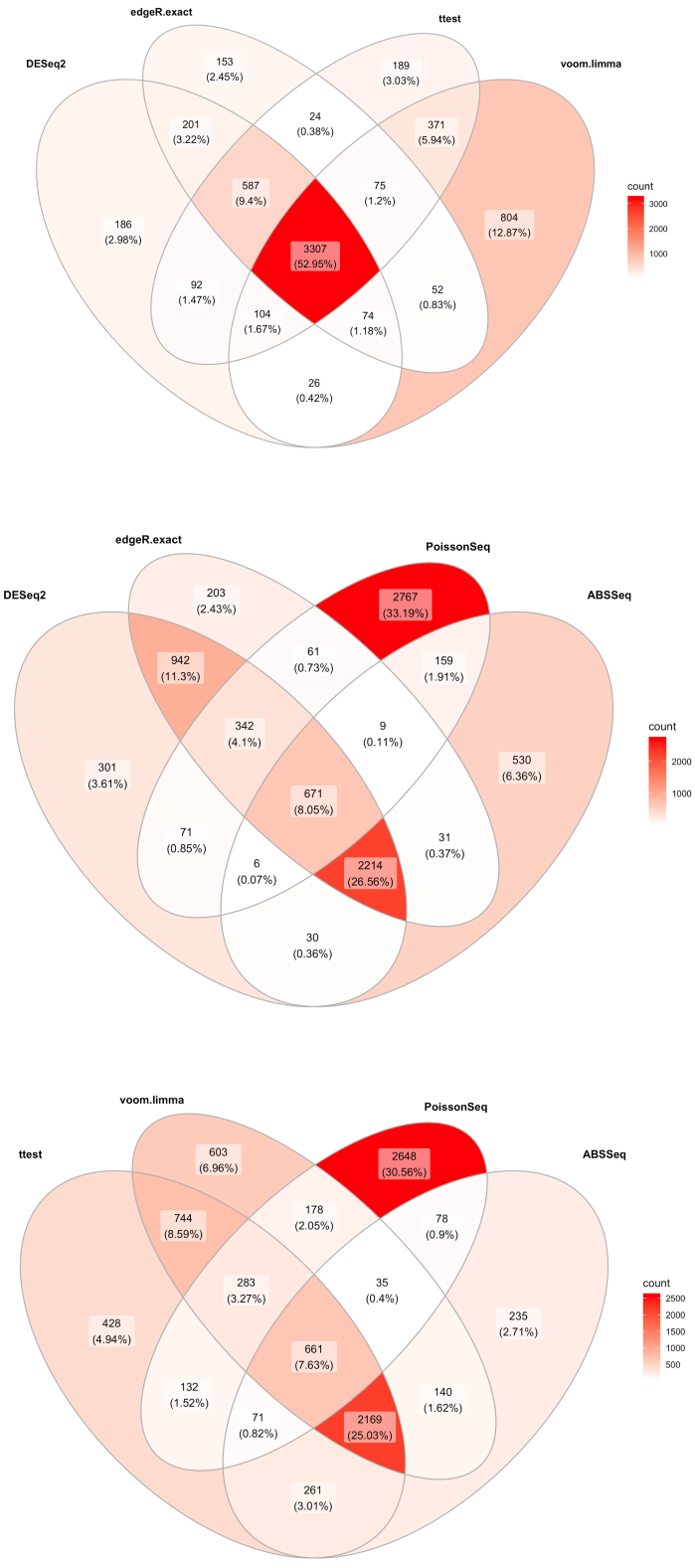



**Figure 21.** The figure above shows boxplots of the number of genes of Bipolar Disease patients that each tool yielded as differentially expressed.




**Figures 22a, 22b, 22c .** The figure above shows venn diagrams that represent the overlap among the set of DE genes found by different methods in the Bipolar Disorder section.

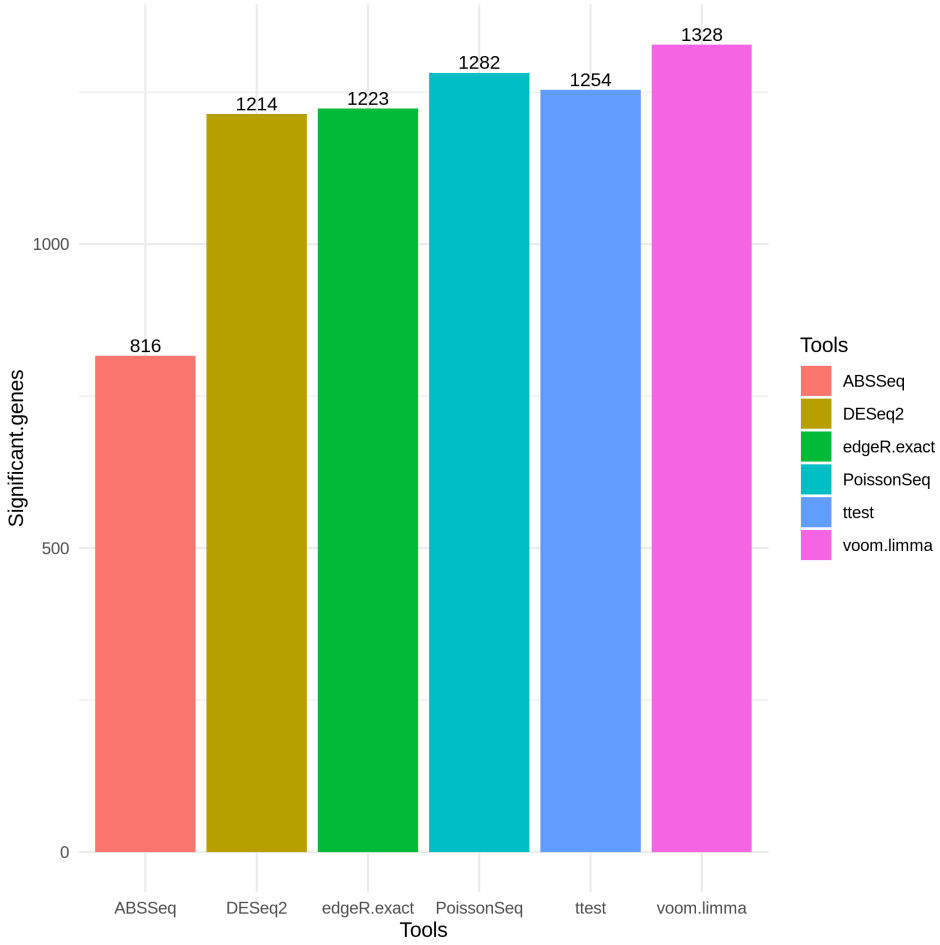
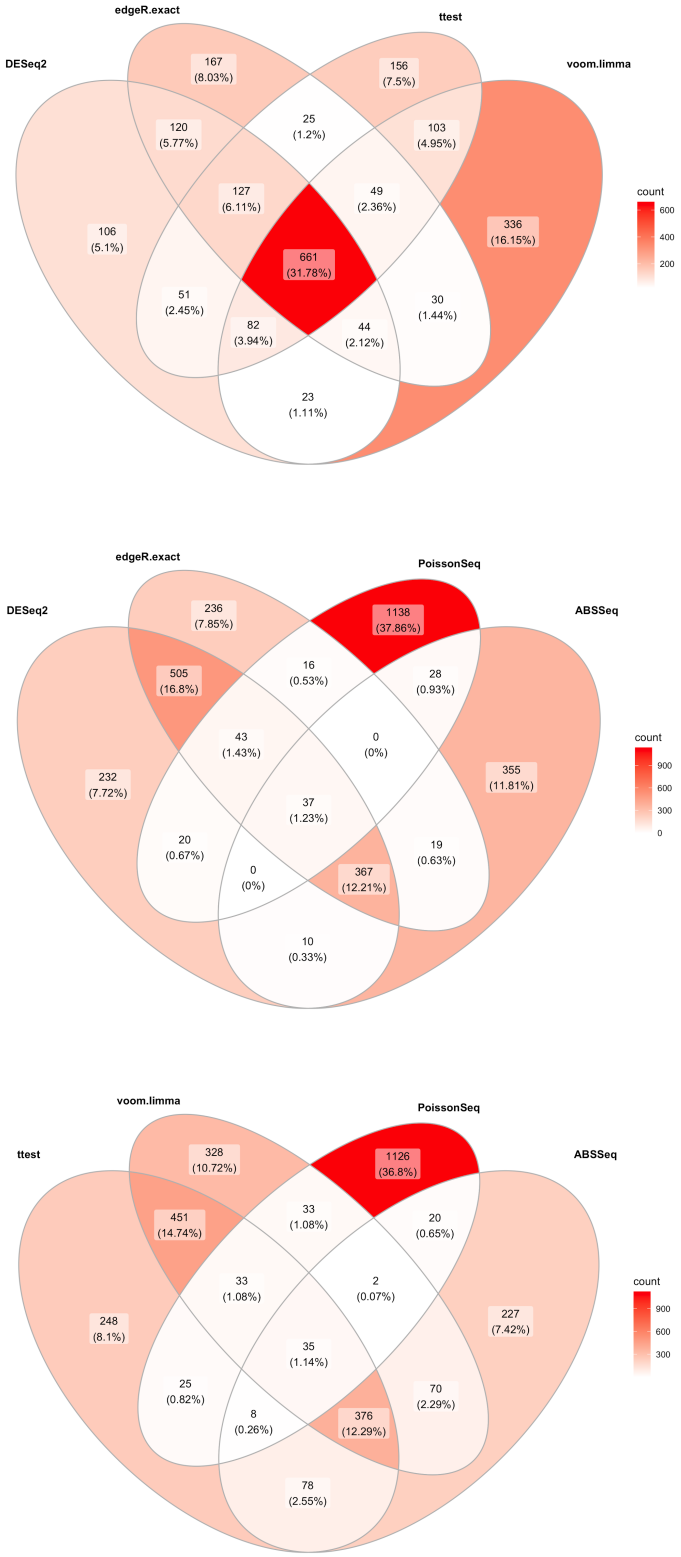



**Figure 23.** The figure above shows boxplots of the number of genes of Major Depressive Disorder patients that each tool yielded as differentially expressed.




**Figures 24a, 24b, 24c .** The figure above shows venn diagrams that represent the overlap among the set of DE genes found by different methods in the Major Depressive Disorder section.