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 which 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 takes the normalised read count data and performs statistical analysis to discover quantitative changes in expression levels between experimental groups. As technology progresses by each year passing, there are now a lot of technological tools available in the Internet that can perform such differential expression analysis. The purpose of our project is to take a closer look at some of these tools, and compare their performances to understand which tools are optimal to utilize. Specifically, the softwares 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 are going to compare their performances by looking at parameters such as Area Under the Curve (AOC), False Discovery Rate (FDR), Type I error rate, and Sensitivity.

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 catalogue all species of transcript, including mRNAs, non-coding RNAs & small RNAs and more importantly, to quantify the changing expression levels of each transcript during development and under various environmental and disease conditions. Identifying genes that are differentially expressed is extremely helpful in determining which biological mechanisms could have an effect on 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 labelled cDNA to microarrays, which while being relatively high throughput and inexpensive, also have limitations such as reliance upon existing knowledge about genome sequence, and limited dynamic range of detection[9]. Traditionally, researchers used to perform microarrays to analyze genes. However, RNA-Sequencing has recently grown in popularity due to their ability to provide much more precise and accurate measurements, and their potential to cover a wider range of transcripts that haven't been correlated to an existing genome, while also having extremely low background signal all at a lower cost compared with microarrays.

With the emergence of RNA-sequencing data, countless softwares 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. Finally, the researcher analyzes the differentially expressed genes among all the samples. 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 both determine which tools to use and how to precisely use them as there is no single standardized pipeline for differential expression due to the diversity in types of RNA data. Here, we hope to investigate which software is the most efficient and yield the best results by comparing specific tools on both a synthetic dataset and a real life dataset. Because quality control and cleaning of RNA-seq data can only be assessed by computational efficiency and costs, we would want to focus the core of this project on comparing gene differential expression tools. Different tools utilize distinct methods to normalize and calculate the abundance of each gene expressed in each sample, so we would like to test these tools against each other to study which would produce the best results. Overall we include for analysis the following tools: ABSSeq[1], voom.limma[2], PoissonSeq[3], DESeq2[4], NOISeq[5], ttest[6] and edgeR[7] while evaluating their performance on Area Under the Curve (AOC), False Discovery Rate (FDR), Type I error rate, and Sensitivity. The importance of this project 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 need to be controlled.

Dataset

Real life RNA-seq data are hard to interpret and genes determined to be differentially expressed are determined only within a degree of certainty in a normal experiment, we decided to test the tools using a simulated post-alignment dataset, in which we can control variables such as proportion of genes differentially expressed, number of up and down-regulated genes, outliers, and samples per condition. Because it is a synthetic dataset, we can know with full certainty which genes are truly differentially expressed and therefore we can actually calculate accuracy metrics against the truth, which we would not be able to know in a real experiment. We chose to create several datasets with various combinations of number of differentially expressed genes and samples per condition in order to capture the different types of real life genetic data. This will allow us to test, for example, if certain tools work better when there are many differentially expressed genes as opposed to few.

Sim. study	G_{DE}^{up}	G_{DE}^{down}	**|{g; ϕ_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%

Figure 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. 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 2, 5, & 10 samples between 2 conditions, denoted by S_1 and S_2. The simulated studies (left column) have superscripts which 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). 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.

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*` = 5, `*dispersions*` = # |{g; ϕ_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, `*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, `*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).

Performing Differential Expression

For performing tools that were supported by compcodeR, we used the built-in function called `*runDiffExp*` where we specify the `*result.extent*` parameter as: DESeq2, edgeR, NOISeq, voom.limma, and ttest. However, the last two tools that we wanted to investigate are not part of compcodeR, ABSSeq and PoissonSeq, both of which require different parameters that will be explained further below.

***ABSSeq*[1]**

This tool performs differential expression analysis of RNAseq data by absolute counts difference between two groups, utilizing Negative binomial distribution and moderating fold-change according to heterogeneity of dispersion across expression level.

***voom.limma*[2]**

This tool performs differential expression analysis of RNAseq data (comparing two conditions) by applying the voom transformation (from the limma package) followed by differential expression analysis with a t-test. Voom precision weights unlock linear model analysis tools for RNA-seq read counts. Then, limma fits a linear model to the expression data for each gene

***PoissonSeq*[3]**

This tool estimates the sequencing depths of experiments using a new method based on Poisson goodness-of-fit statistic, and calculates a core statistic on the basis of a Poisson log-linear model, and then estimates the false discovery date using a modified version of the permutation plug-in method.

***DESeq2*[4]**

This tool performs differential gene expression analysis by estimating the variance-mean dependence in count data using the Negative Binomial Distribution. More specifically, DESeq2 estimates the size factors using the gene counts of the samples, then estimation of dispersion, and lastly performs Negative Binomial Distribution GLM fitting and Wald significance tests. For our purpose, we created a parametric type of fitting of the dispersions to the mean intensity and utilized the Wald test to perform differential gene analysis.

NOISeq[5]

This tool performs differential gene expression analysis of RNA-seq expression data between two conditions without parametric assumptions. NOISeq models the noise distribution of count changes by contrasting fold-change differences and absolute expression differences for all the features in samples within the same condition. NOISeq has 3 main functions: performing quality control of count data, normalization and filter low-counts, and performing differential expression analysis.

ttest[6]

This tool uses the edgeR package to perform differential expression analysis of RNAseq data (comparing two conditions) using a t-test, applied to the normalized counts.

edgeR.exact[7]

This tool performs differential gene expression analysis of RNA-seq expression profiles with biological replication. This R package contains multiple statistical methodology based on the Negative Binomial Distributions, including empirical Bayes estimation, exact tests, generalized linear models, and quasi-likelihood tests. For our investigation, we implemented genewise exact tests for differences in the means between the 2 conditions of negative-binomially distributed counts[8].

Results

Exploratory Analysis of the Tools

Figure 2 The figure above shows the runtime taken by each tool to process each individual synthetic dataset.

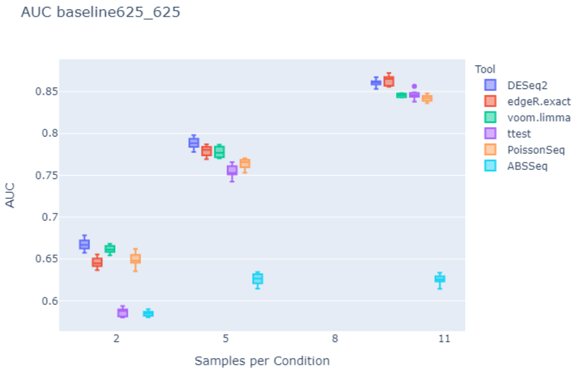
The results produced from each tool on the 11 synthetic datasets were stored in `*<synthetic_dataset_num><tool>.rds*`, where *synthetic_dataset_num* was either `*baseline0_0*`, `*poisson625_625*`, etc. and <*tool*> represented the tool performed on the respective dataset; each file returned a matrix of genes with its p-values and log fold changes. Therefore, we wanted to investigate how the tools did with a basic summary: number of genes that were identified as differentially expressed with a significance level of 0.05, meaning p-values that were less than 0.05.

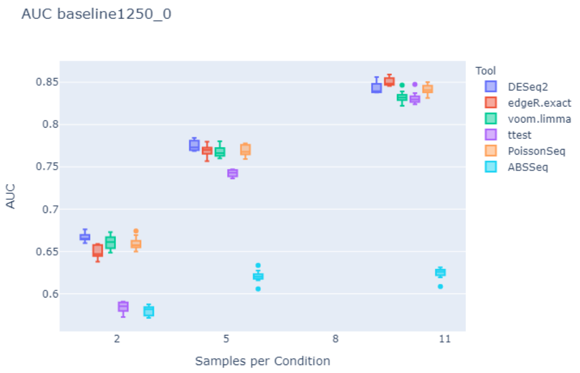
Synthetic Data	**DESeq2**	**edgeR**	**NOISeq**	**ttest**	**voom.limma**	**PoissonSeq**	**ABSSeq**
baseline0_0	762	590	1	593	640	831	0
baseline1250_0	1559	1270	3238	1229	1324	1515	0
baseline625_625	1525	1281	6563	1222	1327	1510	0
baseline4000_0	3453	2645	1186	2845	3048	2789	0

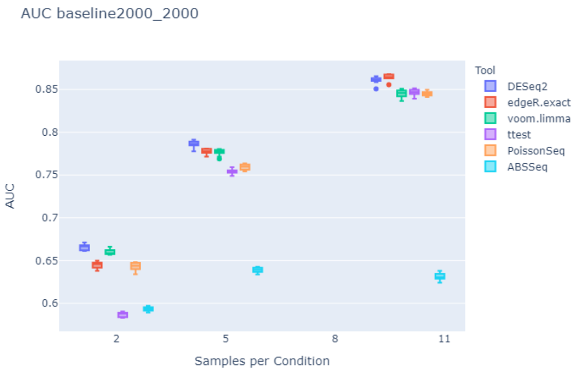
baseline2000_2000	3107	2826	3405	2614	2870	2857	0
poisson0_0	410	408	3	527	508	702	7
poisson625_625	1303	1319	3436	1331	1338	1380	327
single0_0	1041	731	1	505	574	887	0
single625_625	1785	1406	4455	1117	1263	1528	0
random0_0	1999	1275	1	436	469	899	0
random625_625	2419	1677	5906	909	1032	1269	0

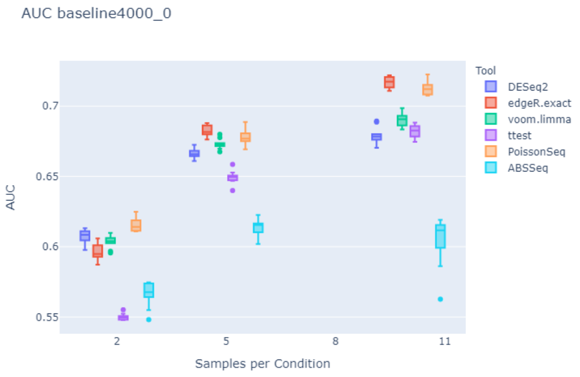
Figure 3 The figure above was produced from counting all the genes that had a p-value less than 0.05 significance level. In the "Synthetic Data" column, the different synthetic datasets are represented as the same format as **Figure 1** (from above) where the corresponding numbers represent the number of genes differentially expressed in condition 1 vs condition 2. The values under each column, representing a software, are the counts of genes that are significantly expressed (< 0.05). Overall, it can be seen that most tools do not perfectly identify the number of truly differentially expressed genes.

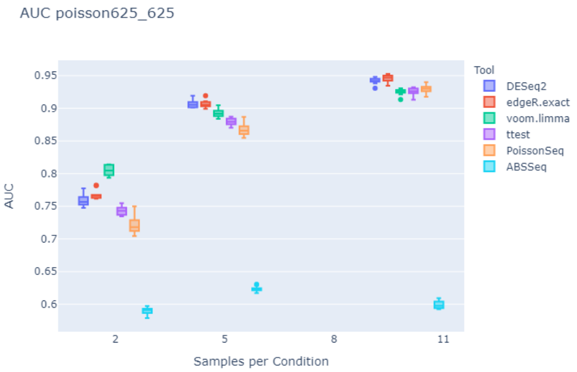
First, it can be seen that ABSSeq identified 0 genes as significantly expressed in almost all synthetic datasets except for the poisson synthetic dataset. This finding should be looked into as these results are peculiar as one would expect more genes to be considered differentially expressed in synthetic data where there are truly differentially expressed genes. Second, for NOISeq, it almost perfectly identifies either 1 or 3 genes that are differentially expressed in the synthetic datasets where 0 genes are differentially expressed. This is unique compared to the other tools, except ABSSeq, where most softwares identify a couple hundred of genes that are differentially expressed when there are 0 truly differentially expressed genes in the synthetic dataset. This is interesting or something we have to investigate further because there should not be a common pattern where all these tools (DESeq2, edgeR, ttest, voom.limma, and PoissonSeq) consider genes as differentially expressed when there are 0 truly expressed genes. For the synthetic datasets which had 1,250 truly differentially expressed genes, it seems that edgeR was overestimating the true number by about 20 and ttest was underestimating the true number by about 20 as well. Overall, this is a very rough look at the initial results as this statistic looks merely at the total counts which ignores whether the genes are even correctly labeled at all. We have some preliminary tables with such statistics as FDR and AUC that have not yet scaled to work for all datasets due to their differences and are not ready to be shared yet.

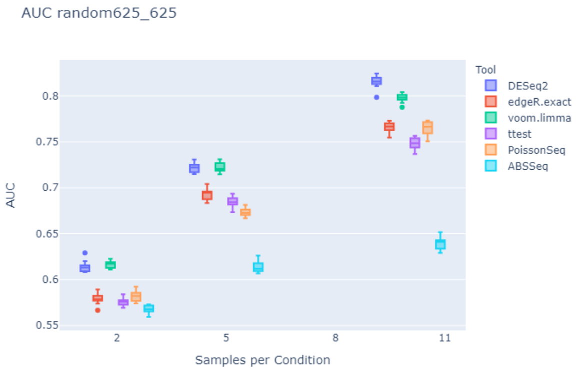

AUC baseline625_625

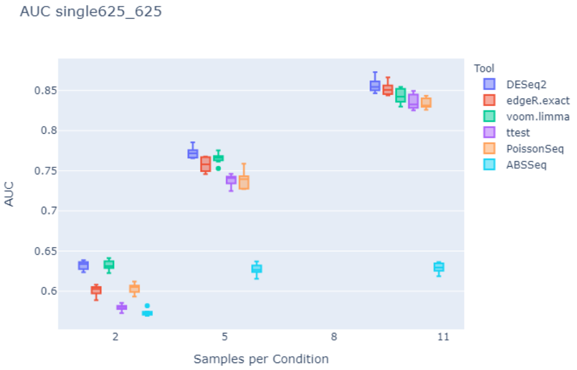

AUC baseline1250_0


AUC baseline2000_2000


AUC baseline4000_0

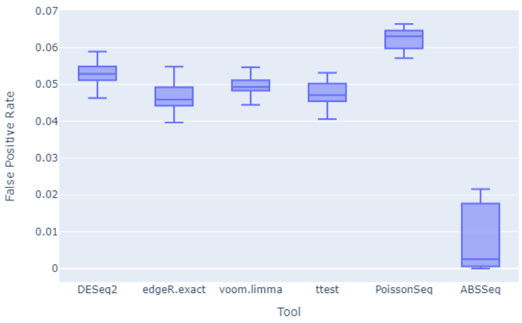




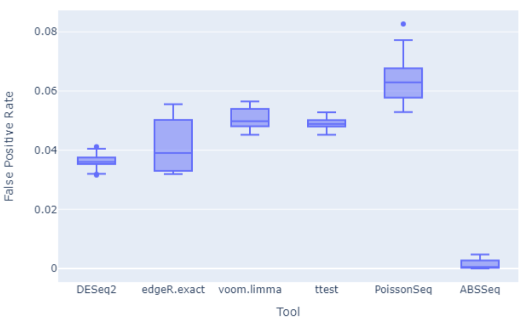


Generally, with the increasing of samples, the performances of all the tools increase for the AUC metric. This is intuitively predictable, as we get better results if we work with more data. We could observe that DESeq2, edgeR.exact, voom.limma, ttest and PoissonSeq have fairly similar AUC results, having a difference that maxes out at 0.1 depending of the dataset being analyzed. ABSeq performed the worst across all datasets, resulting in AUC lower than 0.65 even with the most samples per condition datasets. This makes ABSeq a bad 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.

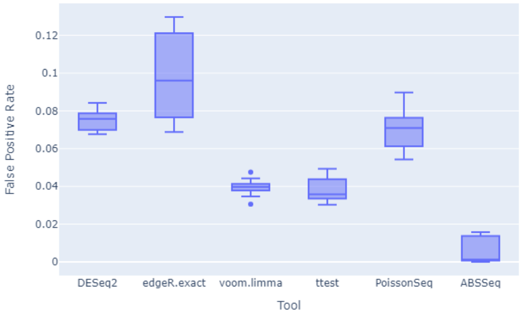
Across the tools that performed well, edgeR.exact excelled with the most samples per condition datasets (excluding single and random outlier datasets). However, when there were less samples per condition, DESeq2 produces the best AUC results for most of the datasets. It is worth noting that PoissonSeq performed worse than the other tools in the poisson distributed dataset, which is surprising, since PoissonSeq performs differential gene expression analysis using a Poisson log-linear model . Also, the AUC of all the tools perform less well when performed on the single and random outlier dataset.



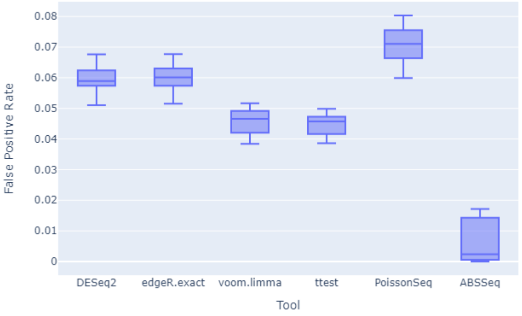
Type I Error Rate baseline0_0



Type I Error Rate poisson0_0



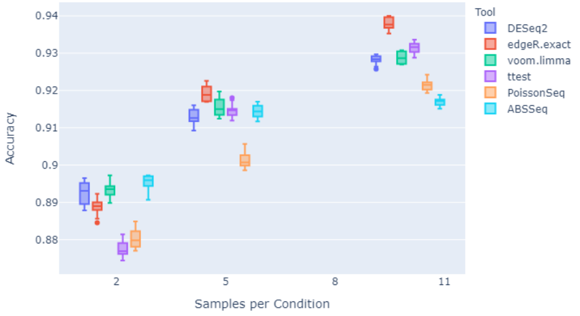
Type I Error Rate random0_0



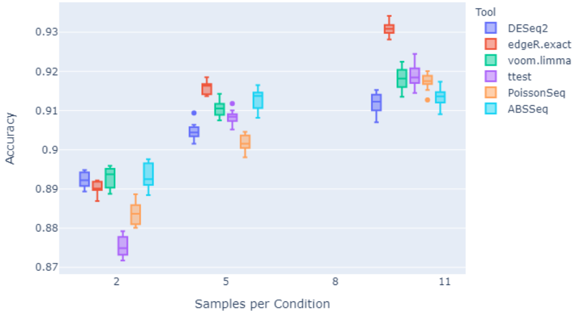
Type I Error Rate single0_0

Type I Error represents the percentage of genes mistakenly classified as differentially expressed. It can be shown that ABSeq performs the best in this metric in particular, as it has very low Type I Error rates compared to the rest of the tools. Most tools average below 10% false discovery rate, which is an acceptable threshold. Except for that, there is not a general pattern that can be used to characterize all the tools across the different datasets. Amongst the tools, edgeR.exact was the most unstable, as it performed with high variance in different datasets. On the contrary, voom.limma and ttest had low variances, and as a more stable Type I Error rate result.

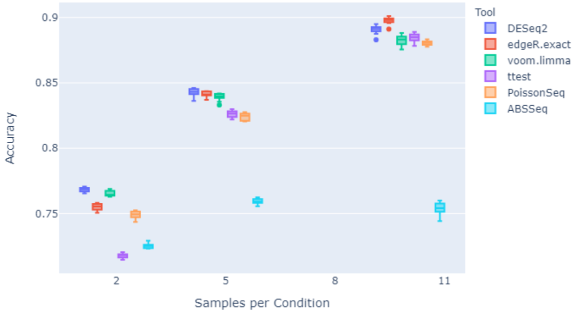
Accuracy baseline625_625



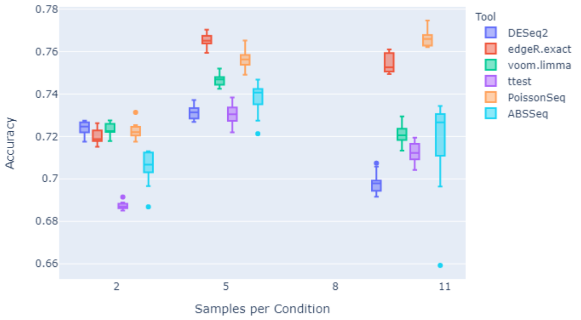
Accuracy baseline1250_0

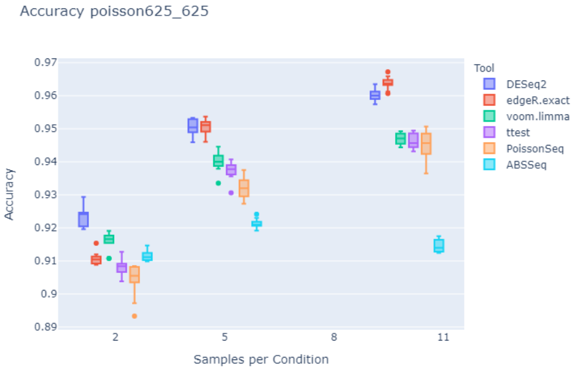


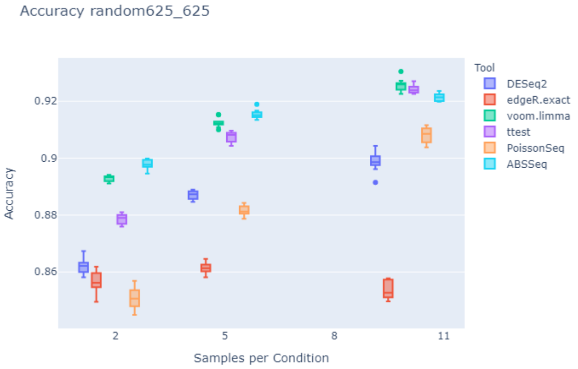
Accuracy baseline2000_2000

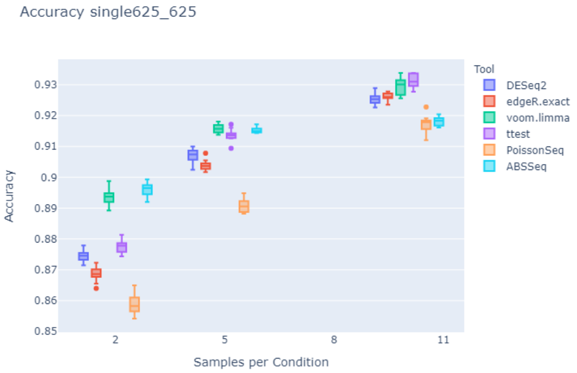


Accuracy baseline4000_0





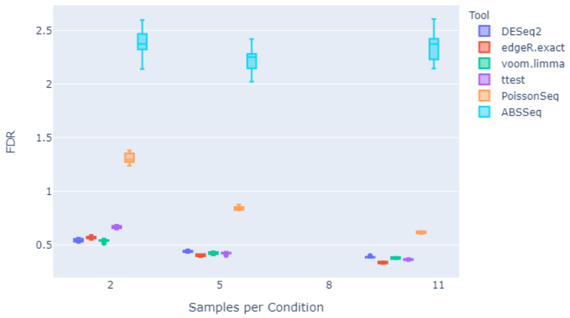




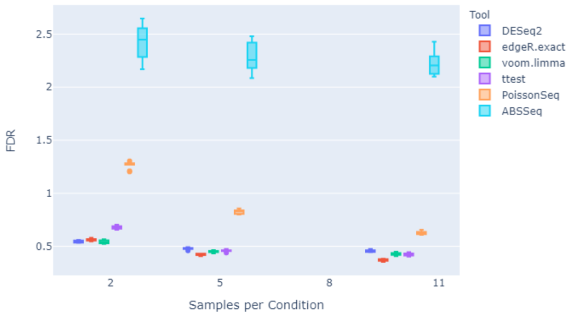
Generally, across all graphs, it can be observed that as sample size increases, so does the accuracy which makes sense as the more samples we have per condition, the more data we have to draw conclusions from. Additionally, we observed that DESeq2 and edgeR perform roughly the same in cases where there is a ratio upregulated/downregulated in both samples (i.e. baseline625_625, baseline2000_2000, etc.) which makes sense because both tools are very similar in the way that they both assume that no genes are differentially expressed. However, for cases where there are more genes upregulated in one condition than the other (i.e. baseline4000_0), edgeR would typically perform better. As mentioned prior, DESeq uses a “geometric” normalization strategy whereas edgeR uses a weighted mean of log ratios-based method which means both normalize using the calculation of size / normalization factors. It seems as if 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 normalizes the data, but ttest does not.

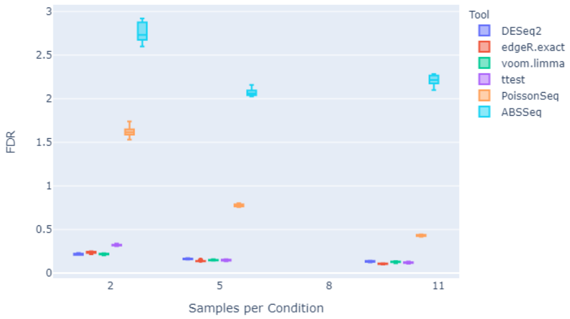
FDR baseline625_625



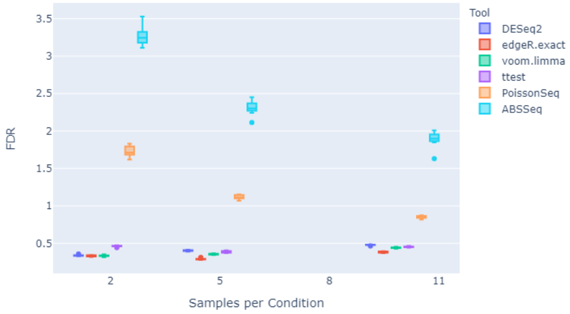
FDR baseline1250_0

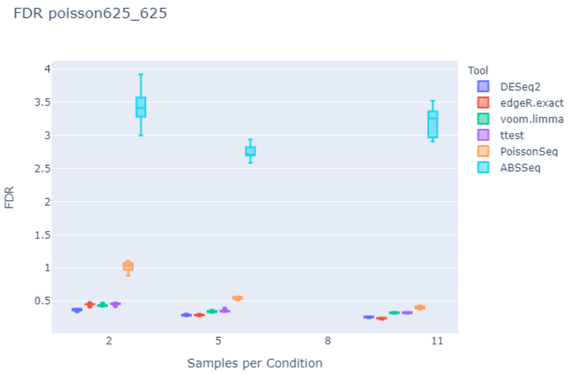


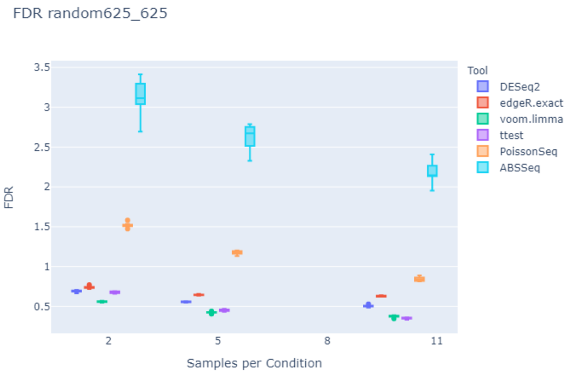
FDR baseline2000_2000

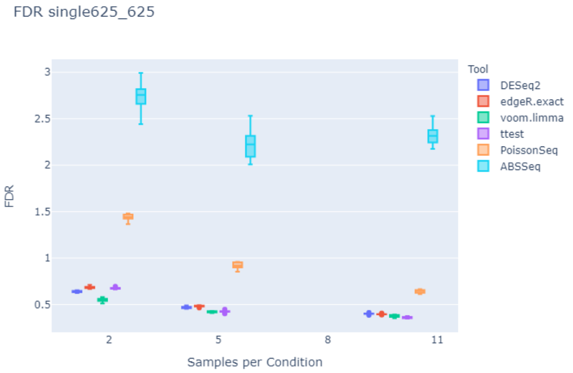


FDR baseline4000_0





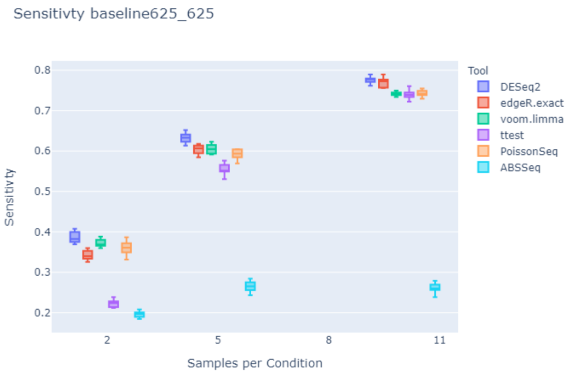


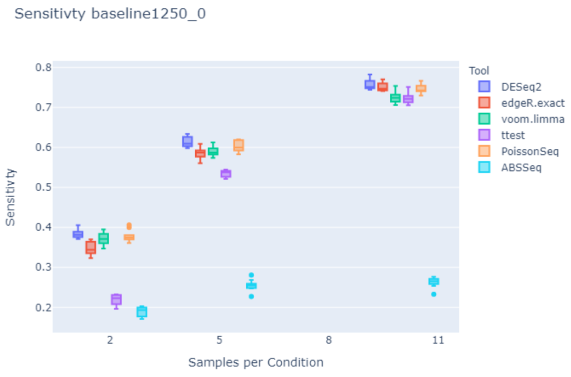


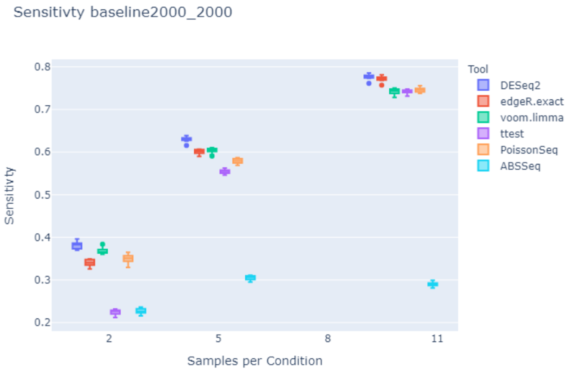
In all graphs, DESeq2, edgeR, voom.limma, and ttest, all rank around 0.5 for false discovery rate with small variance which means that these tools are good at predicting whether a gene is differentially expressed. It is also portrayed that when the differentially expressed genes were regulated in different directions, increasing the number of DE genes from 1,250 to 4000 (i.e. baseline625_625 → baseline2000_2000), FDR would be controlled and decreased. On the other hand, for instances where DE genes were regulated in the same directions (i.e. baseline1250_0 → baseline4000_0) 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 (random & single). This strengthens the fact that voom.limma & ttest are better tools when there are outliers with abnormally high counts.

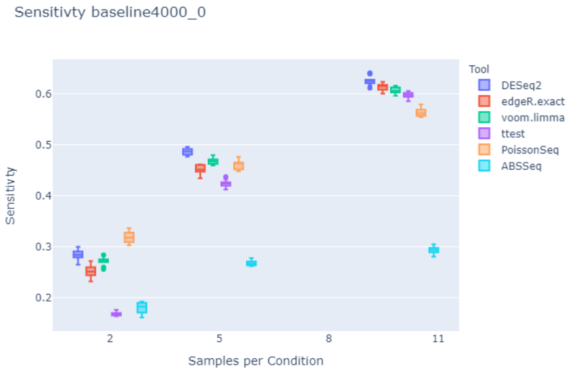
In all synthetic datasets, ABSSeq has a high false discovery rate which means that it doesn't perform nearly as well as the other tools in terms of predicting whether a gene is truly differentially expressed. Furthermore, almost all tools portray low variance in all graphs, except for ABSSeq which demonstrates 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 worse compared to the other tools which is reasonable as it is a new RNA-Seq analysis tool which has recently emerged.

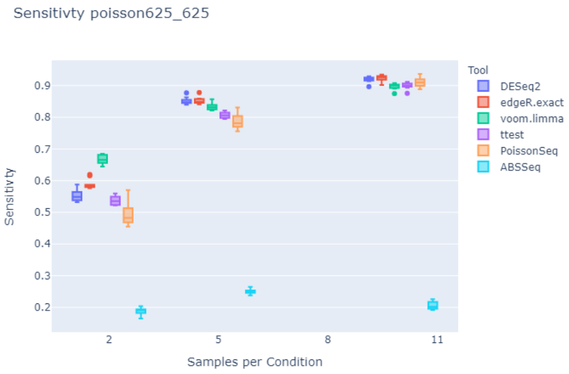
Similar to ABSSeq, PoissonSeq doesn’t perform as well as the other tools (DESeq2, edgeR, voom.limma, & ttest) as it received False Discovery Rates higher than 0.5, but still performed better than ABSSeq. However, there is a unique trend here where the FDR would decrease as the number of samples increase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

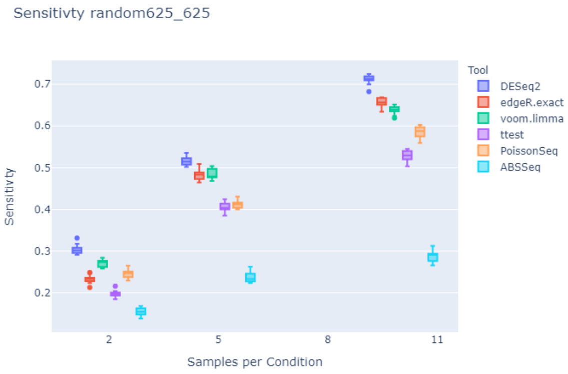


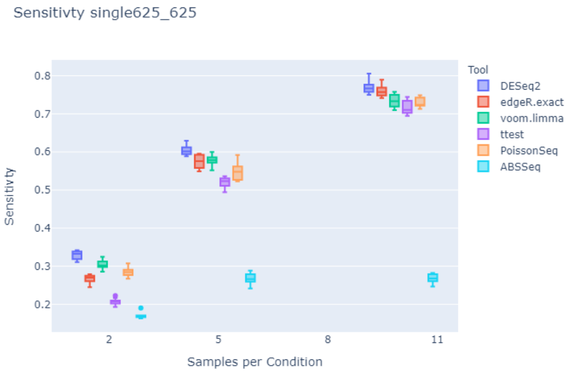


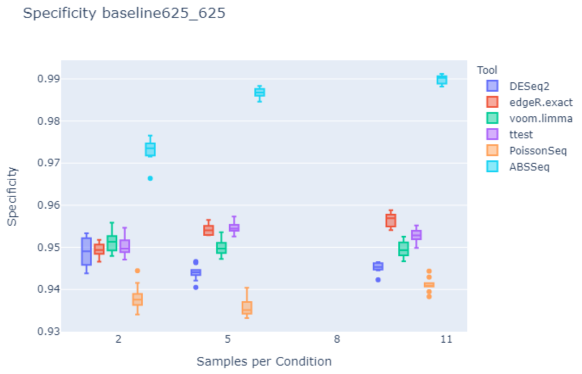


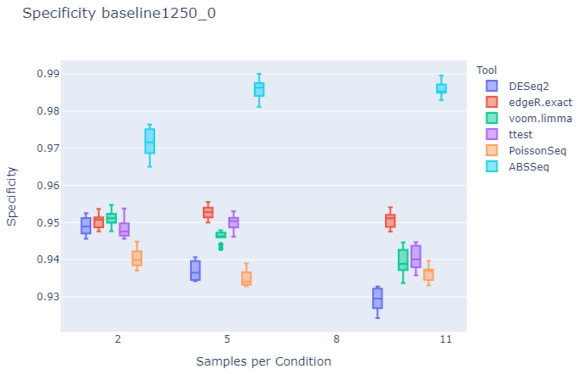


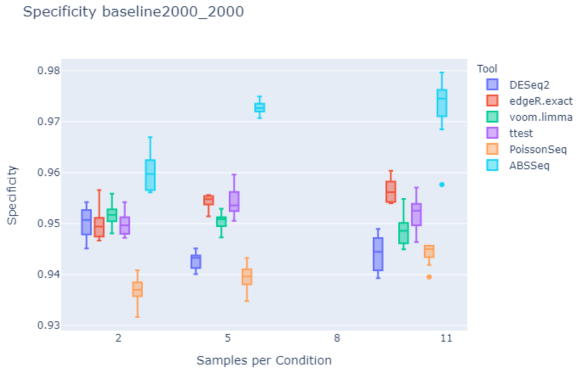

Sensitivty poisson625_625

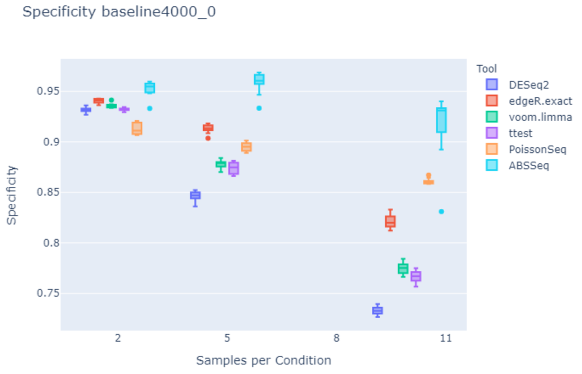

Sensitivty random625_625

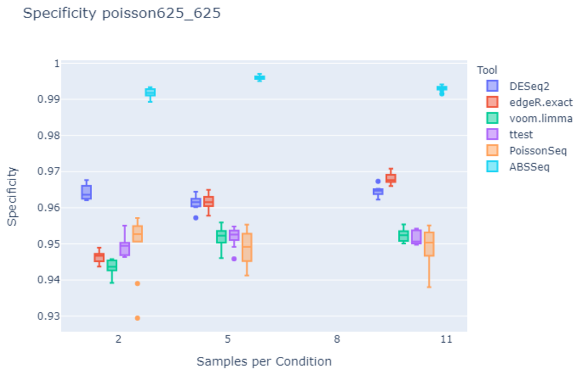

Sensitivty single625_625

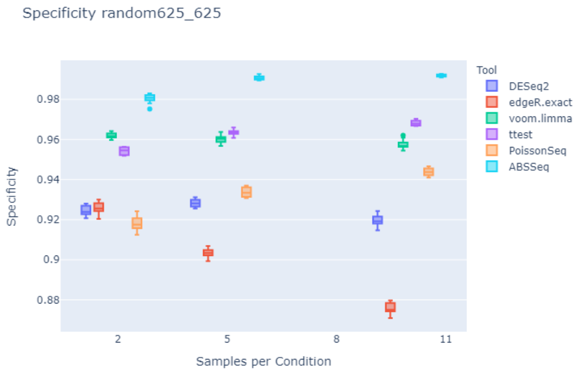

Specificity baseline625_625

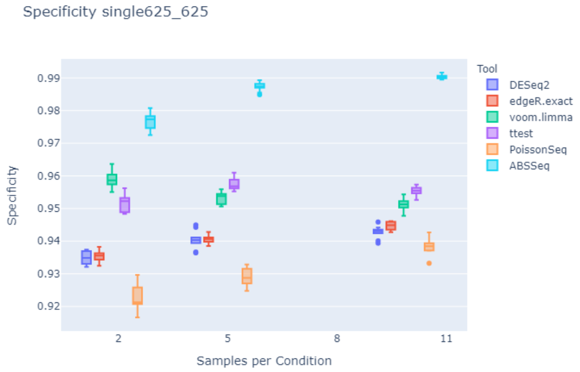

Specificity baseline1250_0


Specificity baseline2000_2000


Specificity baseline4000_0







These graphs look at sensitivity which measures the tools ability to correctly identify the truly differentially expressed genes. As with most of the other statistics, the best performing tool overall was DESeq2 with edgeR close behind. Again we saw better sensitivity performance with greater sampes per condition and in general the difference between the tools was significantly smaller in the datasets with 10 samples per condition with all of the tools outside of ABSSeq typically being within 0.05 of each other outside of the dataset with random outliers. All of the tools also performed significantly better with the poisson625_625 dataset with scores reaching above 0.9 in most tools. Unlike in our other graphs, voom.limma and ttest did not show a significant relative increase in performance in the datasets with outliers. We did however see a stronger performance with PoissonSeq on the poisson 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 better balance different statistics such as AUC.

Unsurprisingly based on the previous trends and the observation that ABSSeq typically classified genes as non-differentially expressed at a much higher rate than other tools, ABSSeq was by far the best tool in terms of specificity, as specificity is a measure of how well each tool correctly identifies non-differentially expressed genes. 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 although in general PoissonSeq was noticeably worse. Interestingly, unlike some of the other statistics, outside of ABSSeq and to a lesser extent, edgeR.exact, there was no increase in performance with an increase in samples per condition. In fact, for the baseline4000_0 dataset, the performance actually worsened with the increase which makes sense because of the smaller proportion of differentially expressed genes.

Again we saw a noticeable increase in performance of voom.limma and ttest in the datasets with outliers compared to DESeq2, edgeR.exact, and PoissonSeq although much like the accuracy graphs, DESeq2 and edgeR were better for the poisson625_625 data.

Overall it seems like DESeq2 and edgeR are the clear choices to use when sensitivity is particularly important to control but 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 the majority of genes were differentially expressed, ABSSeq controlled specificity the best but still seems to be the most useful when the true number of differentially expressed genes is lower.

Observations:

- Accuracy is heavily impacted by the amount of samples per condition. This is expected, as more samples per condition increases the accuracy of all the tools.
- edge.R performs best relative to the other tools on most of the datasets in Accuracy.
- PoissonSeq performs on dataset baseline4000_0
- edge.R performs best relative to the other tools the best on baseline datasets for all the metrics
-
- False Discovery Rates (FDR) decrease with the increase of samples per condition. ABSSeq performs the worst in FDR, with an average of 3% False Discovery Rate across all datasets. PoissonSeq is also not ideal if FDR is an important factor in the analysis. The rest of the tools perform fairly well on this metric.

Notes:

- Overall DESeq had best performance in terms of AUC
    - Exception in the baseline4000_0 dataset so it could be that it does worse with a greater proportion of differentially expressed genes

- DESeq accuracy metric was not good in certain datasets (baseline625_625, baseline1250_0, baseline 4000_0, random625_625, and single625_625)
- Overall, every tool performed significantly worse on the 4000_0 dataset in every metric
- EdgeR had good performance in accuracy metric
- PoissonSeq was weak in FDR for every dataset.
- Big variance for random0_0 edgeR for type I error graph
- For baseline0_0, it seems like most tools (DESeq, edgeR.exact, voom.limma, ttest, and PoissonSeq) predicted there were differentially expressed genes but there aren't
- DESeq2 & edgeR perform similarly for single0_0 in type I graph
- For baseline models, DESeq2 and edgeR would typically outperform voom.limma, but for the other datasets with outliers (single and random), voom.limma would typically perform better than edgeR (for accuracy, FDR, type I error)

Discussion

TODO

Conclusion

TODO

-

Works Cited

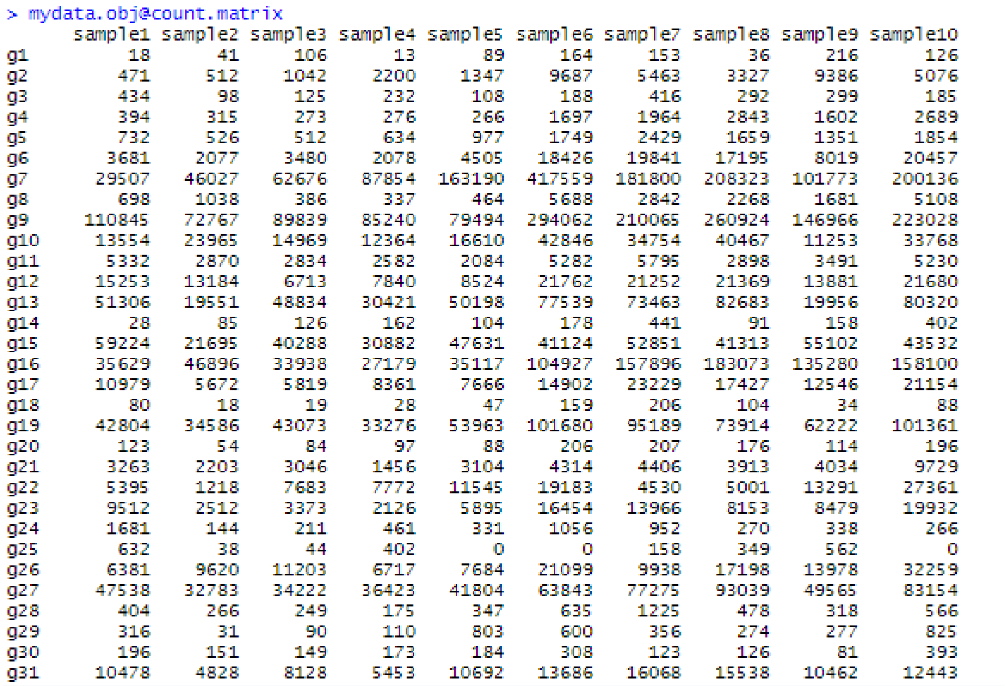
1. https://bioconductor.org/packages/release/bioc/html/ABSSeq.html
2. https://rdrr.io/bioc/limma/src/R/voom.R
3. https://cran.r-project.org/web/packages/PoissonSeq/index.html
4. http://www.bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html
5. https://bioconductor.org/packages/release/bioc/html/NOISeq.html
6. https://bioconductor.org/packages/release/bioc/html/edgeR.html
7. https://bioconductor.org/packages/release/bioc/html/edgeR.html
8. https://rdrr.io/bioc/edgeR/man/exactTest.html
9. https://www.nature.com/articles/nrg2484
10. https://bioconductor.org/packages/release/bioc/html/compcodeR.html
11. https://www.rdocumentation.org/packages/compcodeR/versions/1.8.2/topics/generateSyntheticData
12.

Appendix

Project Proposal

To understand the human genome more clearly, researchers have developed a technology called RNA-sequencing which profiles transcriptomes. Being able to understand these transcriptomes easier will allow researchers to understand the development of a disease, condition, or disorder. Due to the fact that RNA-seq data is important for that matter, there have been specific softwares developed to handle the complicated information of genes. Each software has its own purpose where some of the software are responsible for creating quality checks of the reads produced by RNA-seq, some are to clean the RNA-seq reads from contamination from adapters during preprocessing, and some are to analyze differentially expressed genes among samples. Due to the extensive amount of RNA-seq data available, there are a multitude of softwares used by researchers for this data. In every genetic study using this kind of data, researchers must both determine which tools to use and how precisely to use them as there is no single standardized pipeline for differential expression due to the diversity in types of RNA data. Therefore, in this project, we hope to investigate which software would be the most efficient and yield the best results by comparing specific tools on a synthetic dataset and some real life dataset. Because quality control and cleaning of RNA-seq data can only be assessed by computational efficiency and costs, we would want to focus the core of our project on comparing gene differential expression tools. Different softwares use distinct methods to normalize and calculate the abundance of each gene expressed in each sample, so we would like to test these tools against each other to study which would produce the best results. For instance, DESeq2 performs differential gene expression analysis based on the negative binomial distribution (http://bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html#differential-expression-analysis) whereas DGEclust performs differential gene expression analysis based on clustering (https://dvav.org/dgeclust/). Furthermore, we would like to understand the default parameters of tools more and how tweaking these parameters may affect the results. The importance of this project 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.

The output of our project will be a report explaining in detail what differential gene expression software we used, the parameters tested, and why we chose these specific software. We are going to run the synthetic and real life dataset with different tools to accomplish the same tasks. The synthetic dataset will be created using a custom library, compcodeR from the BiocManager package which has a method that allows us to create our own gene count matrices while controlling various metrics such as the initial number of genes, number of samples per “condition” and most importantly, the number of genes simulated to be differentially expressed between the two conditions. As an example, the following is a snapshot of a small dataset of 1000 genes with 5 samples per condition and 100 genes simulated to be differentially expressed:

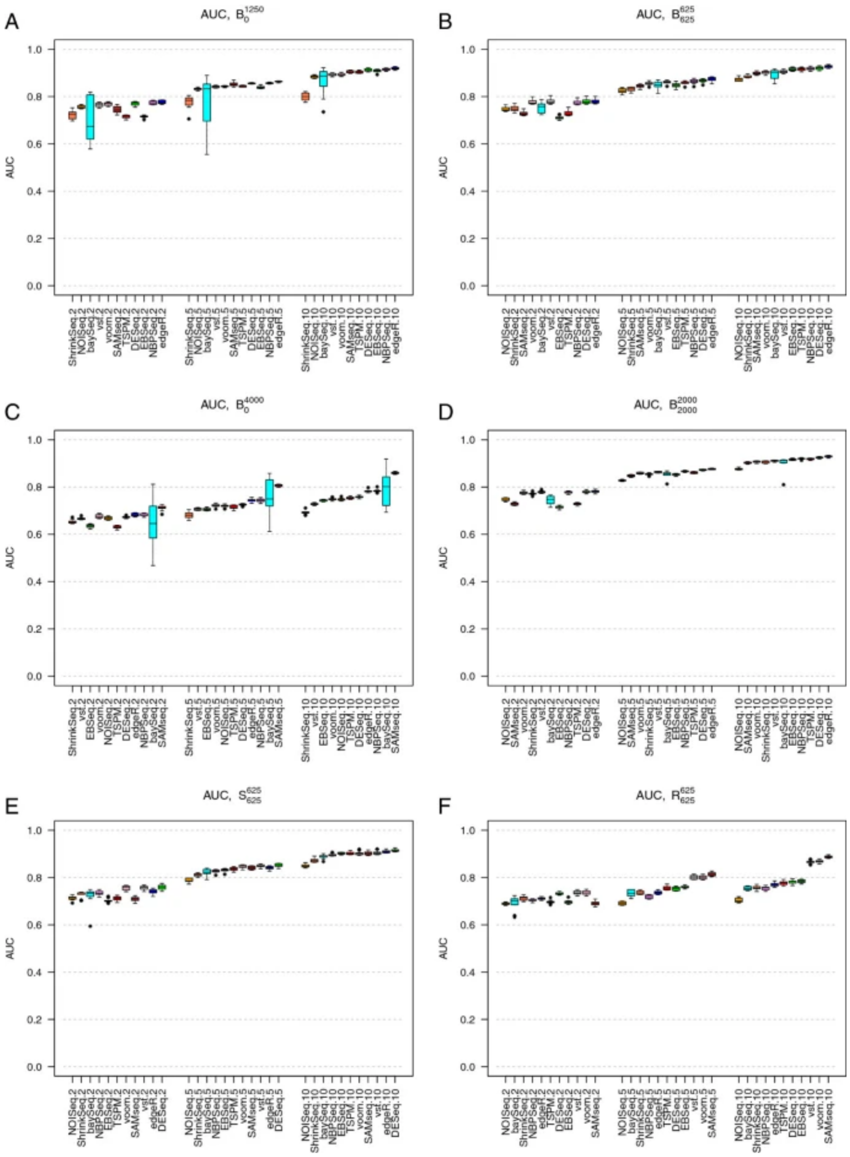


```
> mydata.obj@count.matrix
    sample1 sample2 sample3 sample4 sample5 sample6 sample7 sample8 sample9 sample10
g1       18      41     106      13      89     164     153      36     216      126
g2      471     512    1042    2200    1347    9687    5463    3327    9386     5076
g3      434      98     125     232     108     188     416     292     299      185
g4      394     315     273     276     266    1697    1964    2843    1602     2689
g5      732     526     512     634     977    1749    2429    1659    1351     1854
g6     3681    2077    3480    2078    4505   18426   19841   17195    8019    20457
g7    29507   46027   62676   87854  163190  417559  181800  208323  101773   200136
g8      698    1038     386     337     464    5688    2842    2268    1681     5108
g9   110845   72767   89839   85240   79494  294062  210065  260924  146966   223028
g10   13554   23965   14969   12364   16610   42846   34754   40467   11253    33768
g11    5332    2870    2834    2582    2084    5282    5795    2898    3491     5230
g12   15253   13184    6713    7840    8524   21762   21252   21369   13881    21680
g13   51306   19551   48834   30421   50198   77539   73463   82683   19956    80320
g14      28      85     126     162     104     178     441      91     158      402
g15   59224   21695   40288   30882   47631   41124   52851   41313   55102    43532
g16   35629   46896   33938   27179   35117  104927  157896  183073  135280   158100
g17   10979    5672    5819    8361    7666   14902   23229   17427   12546    21154
g18      80      18      19      28      47     159     206     104      34       88
g19   42804   34586   43073   33276   53963  101680   95189   73914   62222   101361
g20     123      54      84      97      88     206     207     176     114      196
g21    3263    2203    3046    1456    3104    4314    4406    3913    4034     9729
g22    5395    1218    7683    7772   11545   19183    4530    5001   13291    27361
g23    9512    2512    3373    2126    5895   16454   13966    8153    8479    19932
g24    1681     144     211     461     331    1056     952     270     338      266
g25     632      38      44     402       0       0     158     349     562        0
g26    6381    9620   11203    6717    7684   21099    9938   17198   13978    32259
g27   47538   32783   34222   36423   41804   63843   77275   93039   49565    83154
g28     404     266     249     175     347     635    1225     478     318      566
g29     316      31      90     110     803     600     356     274     277      825
g30     196     151     149     173     184     308     123     126      81      393
g31   10478    4828    8128    5453   10692   13686   16068   15538   10462    12443
```

Figure 1 The figure above shows the synthetic dataset we created with 5 samples per condition for the first 31 genes of the 1000 genes where the values are the counts produced after alignment which is needed for the next process: differential gene expression analysis. Details can be found at https://rdrr.io/bioc/compcodeR/man/generateSyntheticData.html

In order to qualify each software’s performance, we will test them each on a variety of synthetic datasets with distinct metrics in terms of the previously listed controllable statistics as well as others such as proportion of upregulated genes and effect size. Specifically, we are going to analyze the following aspects. First, we are going to address their ability to discriminate between DE and non-DE genes. For instance, if a sample were to be sampled twice the depth of of the genes in the first sample to have twice the counts compared to the second sample, and hence this is difference is not going to be caused by differential expression, hence we are referring to this phenomenon as non-DE genes. In detail, we are going to address how well these tools perform in discriminating between truly DE genes and truly non-DE genes. In addition, we are going to check their type I error rates and false discovery rate.

We are loosely following this article from 2013 (https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-14-91) which also contains a software to create synthetic RNA counts data sets (i.e. in a real experiment, the results after alignment). As an example, attached is a figure from the aforementioned paper with part of their results comparing different softwares accuracy based on AUC when testing on datasets of differing size and proportion of differentially expressed genes in the synthetic data:



another, we would expect all

Figure 2 The figure above shows area under the ROC curve (AUC) for the different software being tested on the 6 simulated datasets. The boxplots show the AUCs received across the 10 different softwares. When all the DE genes were regulated in the same direction, increasing the number of DE genes from 1,250 (**A**) to 4,000 (**C**) the performance of all methods decreased. On the other hand, when DE genes were regulated in different directions (**B** & **D**), the number of DE genes were less impactful.

This is similar to what we would like to explore but we will also consider other factors such as parameters within a specific software (i.e. DeSeq2). In addition, we plan to use the results on the synthetic data and test it against real life RNA seq data to see if different softwares or parameters also have the same effect as seen in the test data.