Bayesian hierarchical modeling using omics data

Bayesian statistics

Migla Miskinyte Reis 59606

MAEBD

Universidade Nova de Lisboa

May 2022

Contents

Chapter 1

Introduction

1.1 Bayesian hierarchical modeling for omics

Bayesian hierarchical modelling (BHM) is a statistical model written in multiple levels (hierarchical form) that estimates the parameters of the posterior distribution using the Bayesian method. The full BHM is composed of sub-models, correctly propagating uncertainties in each sub-model from one level to the next allowing to integrate them with the observed data and account for all the uncertainty that is present.

Emerging "omics" technologies (e.g. proteomics, transcriptomics, metabolomics, microarrays), as well as other highthroughput screening for phenotypes, generate not only large, but also sometimes very complex datasets. Thus, extraction of meaningful results remains a big challenge in the field. For example, in transcriptomics, Bayesian hierarchical modeling could help to break down different predictors (such as experimental groups, time points, sequencing technologies, even interactions between gene groups) into smaller sub-models and thus facilitate model implementation and data analysis.

1.2 Biological problem

Survival of tsetse-transmitted trypanozoon species crucially depends on maintenance and expression of their mitochondrial genome, termed kinetoplast (kDNA) [4]. Replication, segregation and expression of kDNA are extraordinary complex processes that involve an estimated 300 proteins, only a minority of which have been identified [2, 8, 9]. Moreover, the kDNA is intrinsically different from mammalian mitochondrial DNA (mtDNA), is essential for parasites survival and is a validated target for anti-trypanosomatid therapies. In this study, genetic RNA interference genome-wide target sequencing approach (RIT-seq) was used in order to find novel mtDNA maintenance factors (unpublished data, see Figure S1). Using the proposed dataset, the main aim of this project is to build and explore Bayesian hierarchical model using large omics dataset in R.

1.3 Dataset description

The total corresponding dataset consists of whole-genome data of *Trypanosoma brucei* parasite (a total of 9566 different genes) containing 3 different experimental groups (day 0 - untreated full complexity *T. brucei* genome, and placebo (Control) and treatment groups (Treated) after 5 days of experiment), each group consisting of 5 biological replicates (15 samples in total). It is important to mention that although day 0 group should have fragments mapping to whole-genome, Treated on Control groups should only have a subset of overall complexity and variance for those groups should be higher. The expectation is that Treated groups should have more reads mapping to genes of interest than other two groups (see Figure S1). Moreover, 2 samples from D0 group and 3 samples from Control group (5 samples in total) were sequenced using different sequencing technology resulting in different library size (total reads mapping to genome, see Table 2.3). Hence because last 5 samples were prepared differently, overall higher variability is expected.

OBJECTIVE: The main objective for this project is to implement a Bayesian hierarchical model for differential gene expression.

Chapter 2

Results and Discussion

2.1 Exploratory data analysis

2.1.1 Multi-dimensional scaling

It is useful to visualise the level of similarity of individual samples of a dataset using multi-dimensional scaling (MDS), first proposed by Torgerson in 1952 [10]. In such approach, the distance between each pair of samples can be interpreted as the leading log-fold change (defined as the root-mean-square of the largest 500 log2-fold changes between that pair of samples) between the samples for the genes that best distinguish that pair of samples. From MDS plot it is apparent that replicate samples from 3 different groups mostly cluster together, however two samples from control group are not well separated and cluster with treated groups (Figure 2.1).

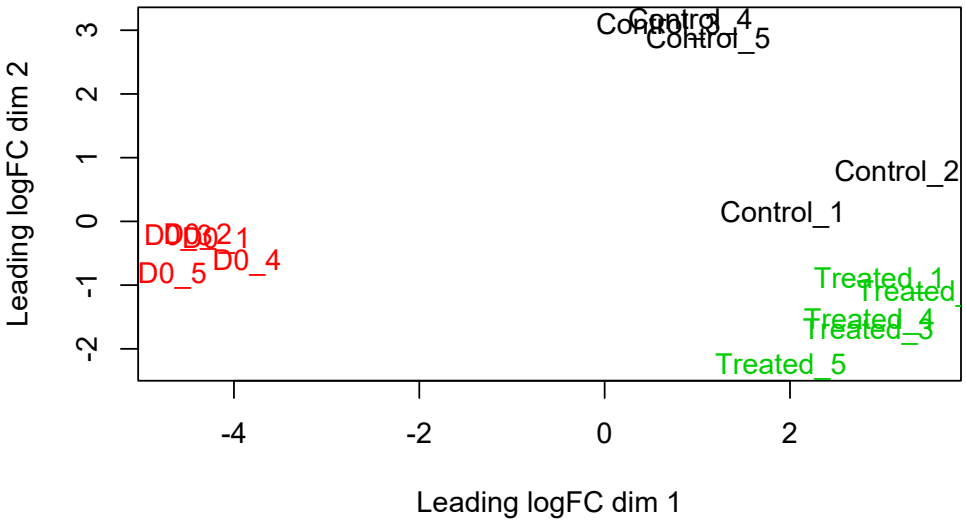


Figure 2.1 Multi-dimensional scaling plot. D0 groups (red) cluster well together, while Control (black) and Treated (green) groups are not well separated.

Table 2.1 Variation $(\times 10^7)$ between samples before log-transformation

Day 0		Treated		Control	
Sample	Var	Sample	Var	Sample	Var
1	7.55	1	57.15	1	15.61
2	2.46	2	59.72	2	60.92
3	1.43	3	35.35	3	1.28
4	7.28	4	29.45	4	1.77
5	3.28	5	24.24	5	2.25

Table 2.2 Variation $(\times 10^7)$ between samples after log-transformation

Day 0		Treated		Control	
Sample	Var	Sample	Var	Sample	Var
1	4.10	1	21.22	1	20.89
2	3.65	2	23.04	2	22.68
3	3.24	3	21.46	3	15.27
4	5.18	4	21.21	4	16.64
5	4.05	5	20.21	5	17.70

2.1.2 Variance

Looking at variance between each of the samples in the dataset before (Table 2.1) and after log-transformation (Table 2.2), it is obvious that overall variation between variables is stabilized and that D0 group has overall less variation than the other two groups. Overall, log-transformed data yield explained variances roughly similar to the standardized data (when each variable is centered and scaled to have unit variance as in correlation matrix).

2.1.3 Normality check

RIT-seq data does not follow a normal distribution (Fig. S2a and Fig. S3b) for the reasons explained beforehand, and even after log transformation, data is not normally distributed. As an example, 1 sample was chosen to illustrate this (see Fig. S2b and Fig. S3b and Anderson-Darling test chosen for large n over Shapiro-Wilks [5] : A = 22.033 and p-value<0.05).

2.1.4 Differential gene expression analysis to extract sample dataset

After exploratory data analysis, I log-transformed my data to counts per million (CPM), which decreased variability of the dataset between each treatment (Table 2.1 and Table 2.2). Looking at normal Q-Q plot before and after log-transformation (see Figure S2), it is obvious that data is not normally distributed. Thus, other distributions, such as poisson or negative binomial distribution are more appropriate to model our data.

Table 2.3 Differences in sample library sizes and TMM normalization factors

Samples	Group	Library size	TMM normalization
1	D0	20202480	2.50
2	D0	16650110	2.95
3	D0	18680762	3.43
4	D0	19436013	2.13
5	D0	24140482	2.72
1	Treated	27181542	0.63
2	Treated	24570257	0.59
3	Treated	23481497	0.54
4	Treated	27494516	0.58
5	Treated	20940650	0.47
1	Control	22406900	0.55
2	Control	38353129	0.57
3	Control	3759288	0.72
4	Control	4442737	0.76
5	Control	4778067	0.73

Differential gene expression analysis was performed using edgeR package in R [1]. Because 5 samples were sequenced using different sequencing technology, samples differ in library size. Moreover, genes differ in size, and larger genes weight more on the analysis than smaller genes and could distort our analysis. In order to normalize our data, a method called trimmed median of M values (TMM) was used [7] (see Table 2.3). TMM calculated a scaling factor for each sample based on fold changes between samples based on overall read depth while excluding the effects of genes that are highly expressed or show extreme changes in expression between samples.

Briefly, differential gene expression analysis was done by fitting data using generalized linear model (GLM) with quasi-likelihood (QL) approach to account for the uncertainty in gene-wise estimates of dispersion as proposed by the authors [6]. A total of 1573 genes were enriched (similarly to an example gene Tb927.7.1400 in Figure S1) in Treated group against D0 group with a chosen threshold of 1.5 logFC (log fold change) and a p-value less than 0.05 (grey area in Fig. S4). Similarly, 375 genes were enriched in Treated group versus Control group (not shown). In total, 117 common genes enriched genes were found in both comparisons. 117 top enriched genes were chosen as a smaller dataset for Bayesian Hierarchical modeling using Hamiltonian Monte Carlo (HMC) sampling in *Stan*.

2.2 Stan model description

In order to model gene expression variability for the same genes across samples, the negative-binomial (NB) distribution is often used in RNA-seq data analysis and thus have been incorporated into existing packages, such as edgeR and DESeq2. The main reasoning for choosing NB distribution is because individual gene expression (or reads per that gene) vary across samples. This highly

increases variability among the counts obtained for each gene. So although, Poisson distribution would be a good choice for modeling count data in general, because of larger number of counts and very small probability of event occurring, it is less suitable for gene expression analysis, because it assumes that mean is equal to variance. To illustrate this, with our dataset, I have computed a vector of mean values and plotted it against the vector of variance values for each gene (see Figure 2.2).

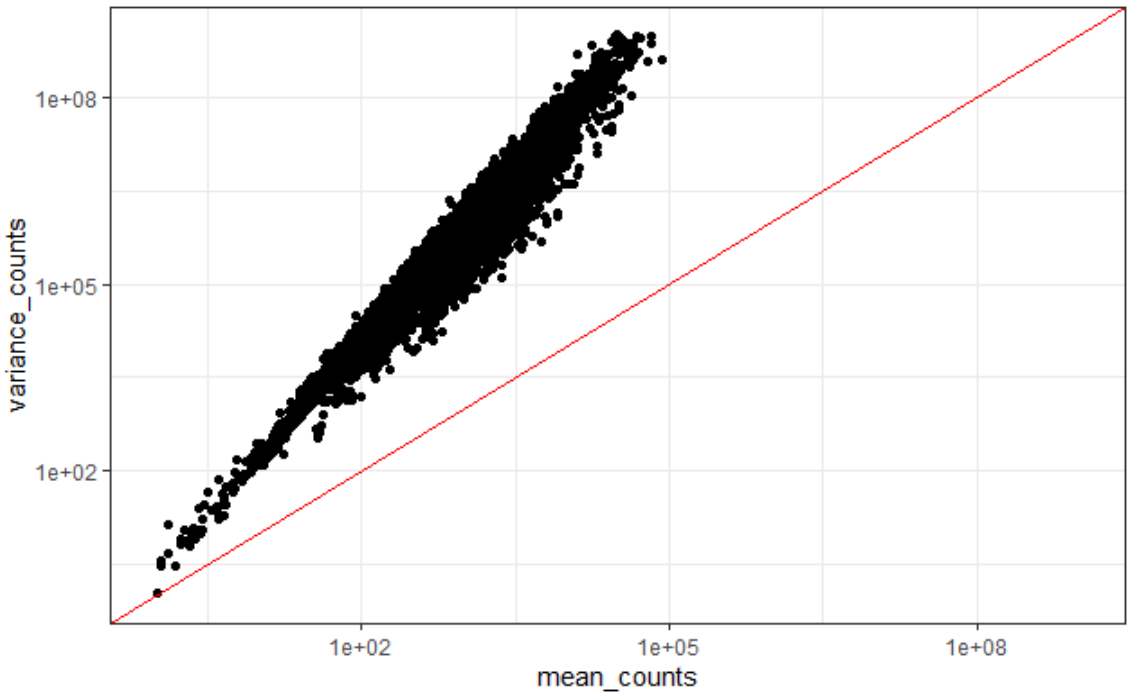


Figure 2.2 Poisson distribution would be appropriate for modeling counts data if μ=σ.

Because the mean is not equal to variance (data does not fall on red line in Figure 2.2), variance is greater for genes with higher mean, our data is not suitable for the Poisson distribution. However, if we greatly increase a sample size and add many more biological replicates, which is often impossible due to experimental design or sequencing costs, we could use the Poisson. Thus, for modeling few replicates, where mean is smaller than variance, NB is more appropriate model. When variable y follows a negative binomial distribution and depends on a set of k explanatory variables X, a simple NB model could be described as:

$$y_i \sim NB(\mu_i, \phi) \tag{2.1}$$

$$E(y_i) = \mu_i \tag{2.2}$$

$$Var(y_i) = \mu_i + \mu_i^2/\phi \tag{2.3}$$

$$log(\mu_i) = \beta_0 + \beta_1 * X1_i + \ldots + \beta_k * Xk_i \tag{2.4}$$

To model our dataset, I have chosen a recently published Bayesian hierarchical model for diferentially expressed counts [3], which uses a more complex hierarchical NB model for gene expression $Counts_{ij}$:

$$Counts_{ij} \sim NB(exp(\alpha_{ij} + \beta_{ij} \cdot Design_{kj} + LogNormFactors_j + LogEffLength), \Phi) \quad (2.5)$$

$$\Phi \sim Uniform(0, 2000000000) \quad (2.6)$$

$$LogNormFactors \sim Normal(0, 0.05) \quad (2.7)$$

In this model, the logarithm of the expected counts for each gene depends on an average gene expression value α, the expression change between different groups β, binary design matrix (e.g. design <- c(0, 0, 0, 0, 0, 1, 1, 1, 1,1)), the normalization factors of each sample (*LogNormFactors*) and on each gene effective length (*LogEffLength*). In this model Φ is dispersion parameter (variability).

The average gene expression value α is drawn from a normal distribution, with hyperparameters μ_α and σ_α:

$$\alpha_{ij} \sim Normal(\alpha_\mu, \alpha_\sigma) \quad (2.8)$$

Change in expression β is drawn from a normal again, assuming mean to be equal to zero and with hyperparameter σ_β:

$$\beta ij \sim Normal(0, \beta_\sigma) \quad (2.9)$$

Hyperparamters μ_α and σ_α related to the average expression of all genes and its standard deviation, were sampled from normal distributions:

$$\mu_\alpha \sim Normal(\frac{n}{T}, 1.17) \quad (2.10)$$

$$\sigma_\alpha \sim Normal(0, 2) \quad (2.11)$$

$$\sigma_\beta \sim Normal(0, 1), \quad (2.12)$$

where n is all gene expression counts and T is the gene coverage (transcriptome length). Model was implemented using *Stan* framework using HMC sampling (code provided in **Appendix B**)

2.3 Model diagnostics

After performing HMC using proposed model in *Stan*, we can visualise main estimated hyperparameters and standard variation associated with them (Figure 2.3).

To check visually if chains have converged, traceplot method is often used to plot time series of the posterior draws (Figure 2.4, grey shaded area indicates warm-up iterations). Indeed, traceplots do not indicate any convergence problems.

Another important diagnostic, known as the potential scale reduction statistic $\hat{R}$ or as the Gelman-Rubin convergence diagnostic:

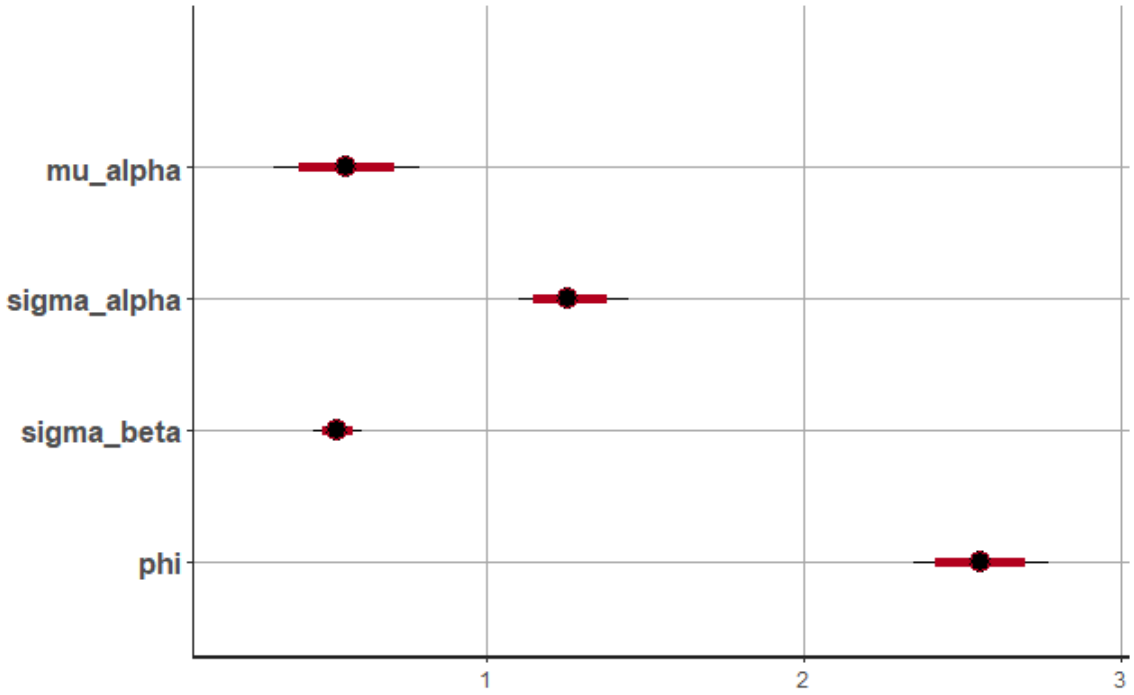


Figure 2.3 Model hyperparameters

$$\widehat{Var}(\theta) = (1 - \frac{1}{\mathtt{n_{iter}}}) \underbrace{\left(\frac{1}{\mathtt{n_{chains}}(\mathtt{n_{iter}} - 1)} \sum_{j=1}^{\mathtt{n_{chains}}} \sum_{i=1}^{\mathtt{n_{iter}}} (\theta_{ij} - \bar{\theta}_j)^2 \right)}_{\text{Within chain var}} + \tag{2.13}$$

$$\frac{1}{\mathtt{n_{iter}}} \underbrace{\left(\frac{\mathtt{n_{iter}}}{\mathtt{n_{chains}} - 1} \sum_{j=1}^{\mathtt{n_{chains}}} (\bar{\theta}_j - \bar{\bar{\theta}})^2 \right)}_{\text{Between chain var}} \tag{2.14}$$

$\hat{R}$ combines information on the variation within and between chains, thus assessing whether each chain converged to a stationary target distribution and whether all chains converged to the same target distributions at the same time. As expected for good convergence in our model, $\hat{R}$ is close to 1. Values below 1.1 are considered to be acceptable (see Figure 2.5).

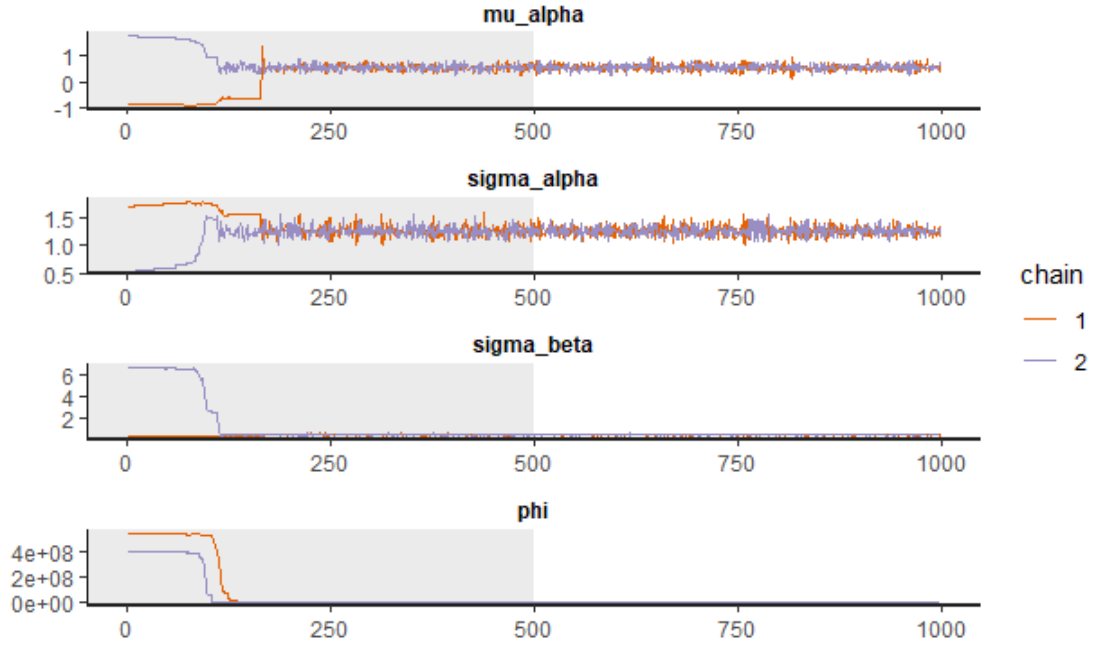


Figure 2.4 Traceplot diagnostic

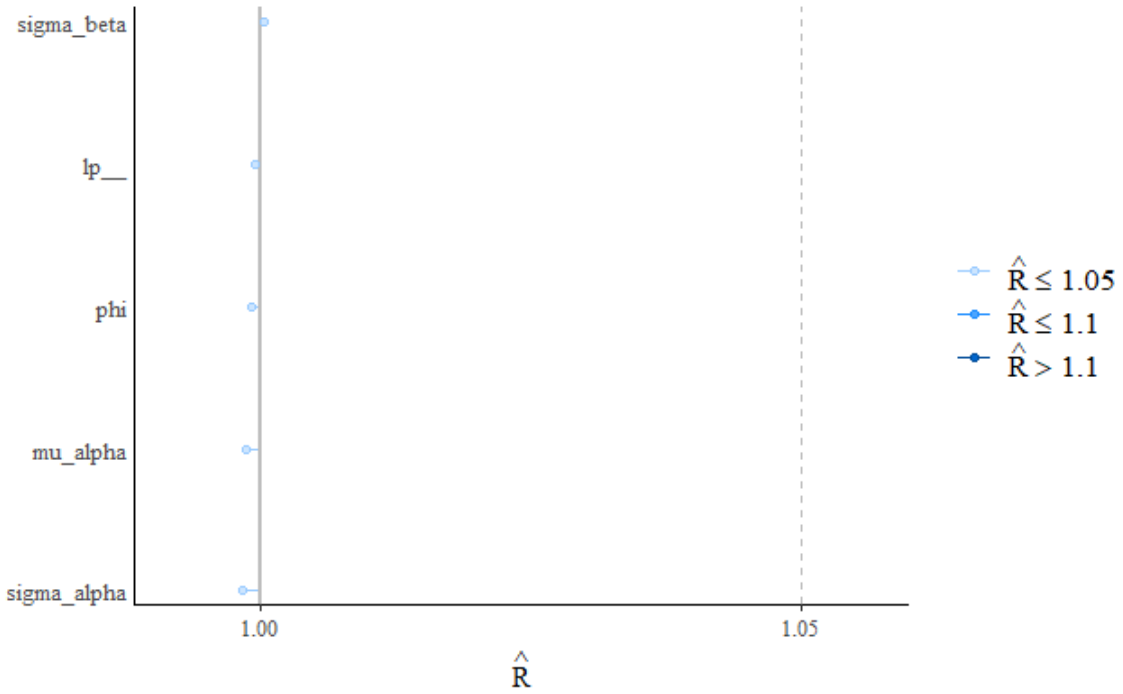


Figure 2.5 $\hat{R}$ diagnostics

The effective sample size $\texttt{N}_{\texttt{eff}}$ is another estimate of the number of independent draws from the posterior distribution of the estimate of interest. A small effective sample size indicates high autocorrelation within chains, which in turn indicates that chains explore the posterior density very slowly and inefficiently. It is considered that a ratio of $\texttt{N}_{\texttt{eff}}$/N <0.001 is considered good, i.e.,

we want a minimum of one effective sample per 1000 post-warmup iterations of our chains (see Figure 2.6).

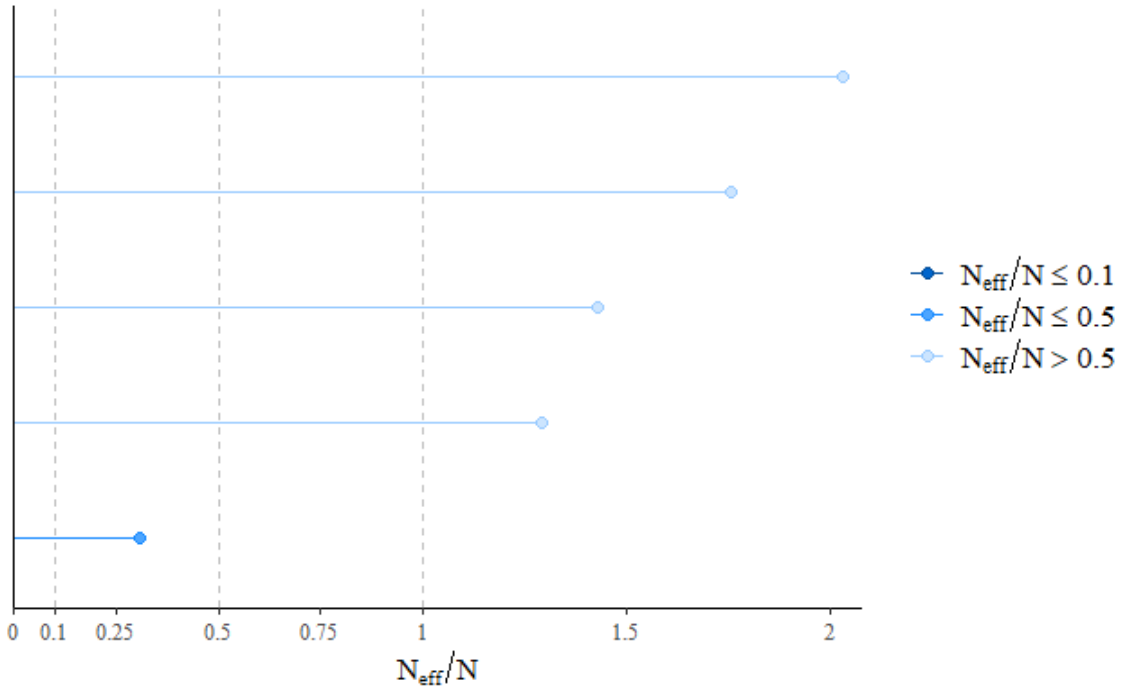


Figure 2.6 N_{eff}/N diagnostics

Autocorrelation analysis also did not indicate any potential problems in the model. We expect a correlation in the first lag and not in the next lags, which we do not see in our model after lag 5 (Figure 2.7).

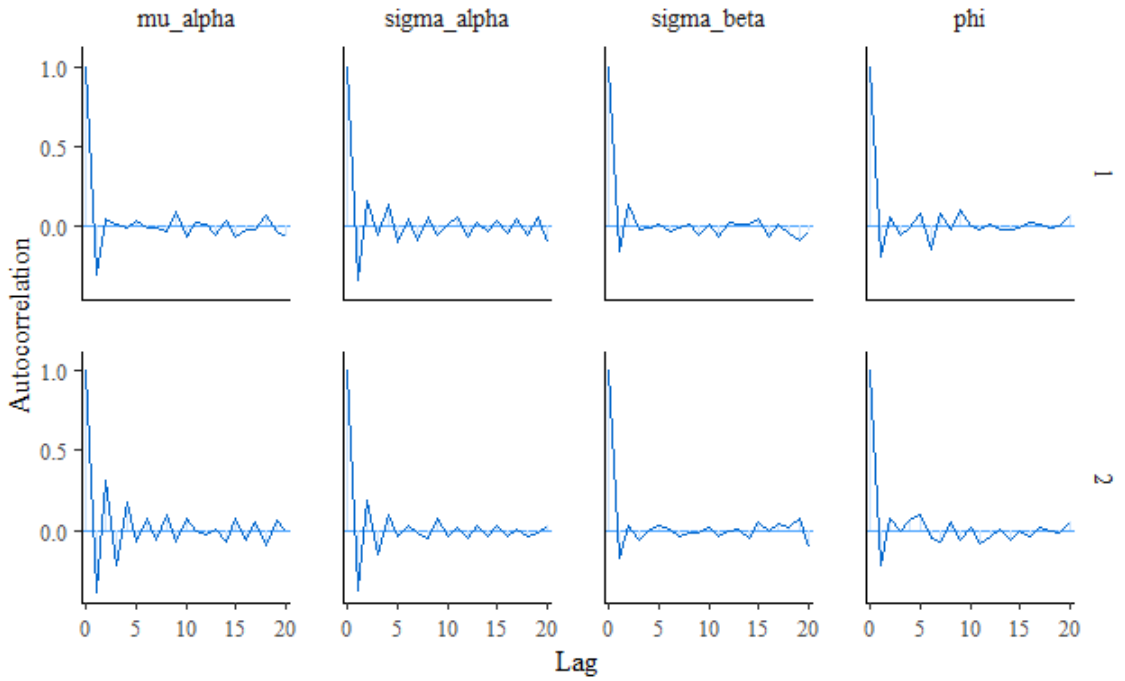


Figure 2.7 Autocorrelation

2.4 Posterior predictive check

After model run and convergence check, it is important to do posterior predictive checks. The idea of posterior predictive checks is to compare our observed data to simulated data from the model. If our model is a good fit, we should be able to use it to generate a dataset that looks similar to the observed data. For this, I have simulated data from the posterior distribution (see expression-rep in Stan model code in **Appendix B**). Interestingly, average expression for genes that are expressed at lower values is very similar between real and simulated data (Figure 2.8), but when we look at genes with very high expression, real data counts are always higher. Hence, even modelling our data to NB distribution, where variance is expected to be higher than mean, especially for highly expressed genes (Figure 2.2), it is not possible to accurately estimate expression of some of the genes.

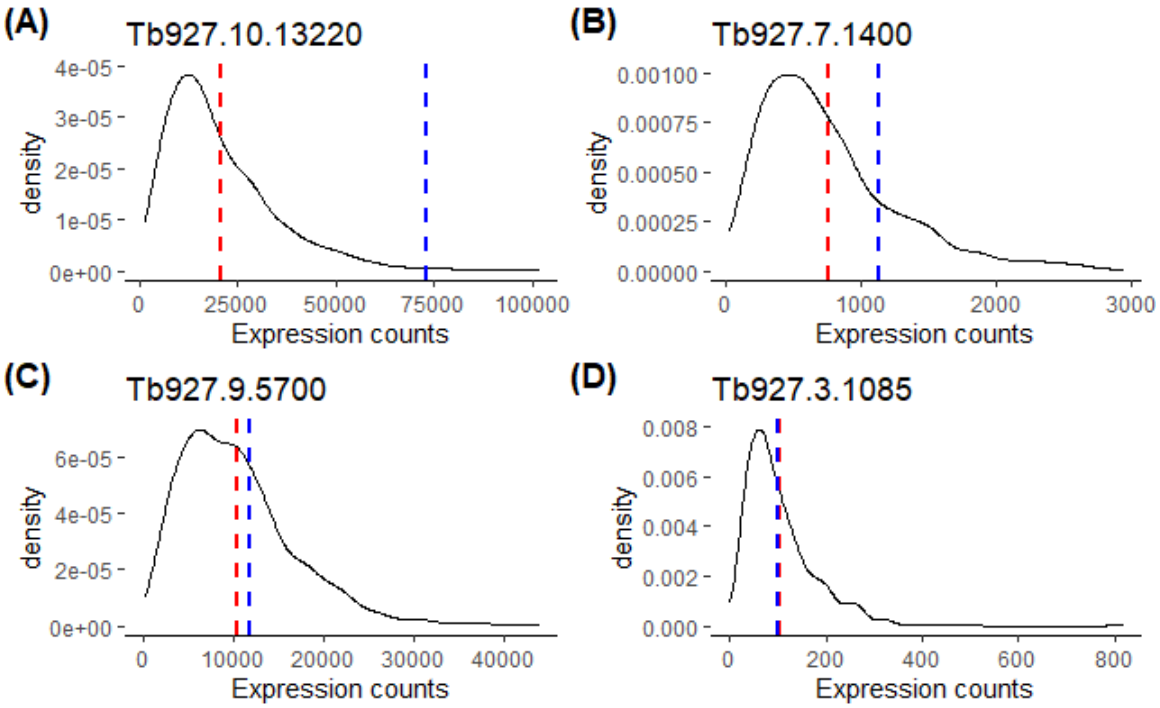


Figure 2.8 Posterior predictive checks for different candidate genes. Blue line indicates average expression for that gene in real data, red line - average for data sampled from posterior distribution.

2.5 Main conclusions

When analysing multi-dimensional datasets, such as those resulting from high-throughput sequencing experiments, one of the biggest challenges lay from the size of dataset itself encompassing thousands of measurements for different genes, treatments and replicates. Such high dimensionality not only makes it difficult to perform meaningful statistical analyses, but also to explore and visualise the data. Although I could not run a full dataset of 9565 genes (because of lack of computing power), it is apparent even with a small dataset, that our our data fits well with a proposed Bayesian

hierarchichal model when gene expression counts are not extremely large or variable between treatment groups.

Bibliography

[1] Chen, Y., Lun, A. T. L., and Smyth, G. K. (2014). *Differential Expression Analysis of Complex RNA-seq Experiments Using edgeR*, pages 51–74. Springer International Publishing, Cham.

[2] Jensen, R. E. and Englund, P. T. (2012). Network news: The replication of kinetoplast dna. *Annual Review of Microbiology*, 66(1):473–491. PMID: 22994497.

[3] Jiménez-Jiménez V, Martí-Gómez C, d. P. M. e. a. (2021). Bayesian inference of gene expression. *In: Helder I. N, editor. Bioinformatics*, 5.

[4] Morrison, L. J., Vezza, L., Rowan, T., and Hope, J. C. (2016). Animal african trypanosomiasis: Time to increase focus on clinically relevant parasite and host species. *Trends in Parasitology*, 32(8):599 – 607.

[5] Pettitt, A. N. (1977). Testing the normality of several independent samples using the anderson-darling statistic. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 26(2):156–161.

[6] Robinson, M. D., McCarthy, D. J., and Smyth, G. K. (2009). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140.

[7] Robinson MD, O. A. (2010). A scaling normalization method for differential expression analysis of rna-seq data. *Genome Biol.*, 11.

[8] Schneider, A. and Ochsenreiter, T. (2018). Failure is not an option – mitochondrial genome segregation in trypanosomes. *Journal of Cell Science*, 131(18).

[9] Shapiro, T. A. and Englund, P. T. (1995). The structure and replication of kinetoplast dna. *Annual Review of Microbiology*, 49(1):117–143. PMID: 8561456.

[10] Torgerson, W. S. (1952). Multidimensional scaling: I. theory and method. *Psychometrika*, 17:401–419.

Appendix A

Supplemental Figures and Tables

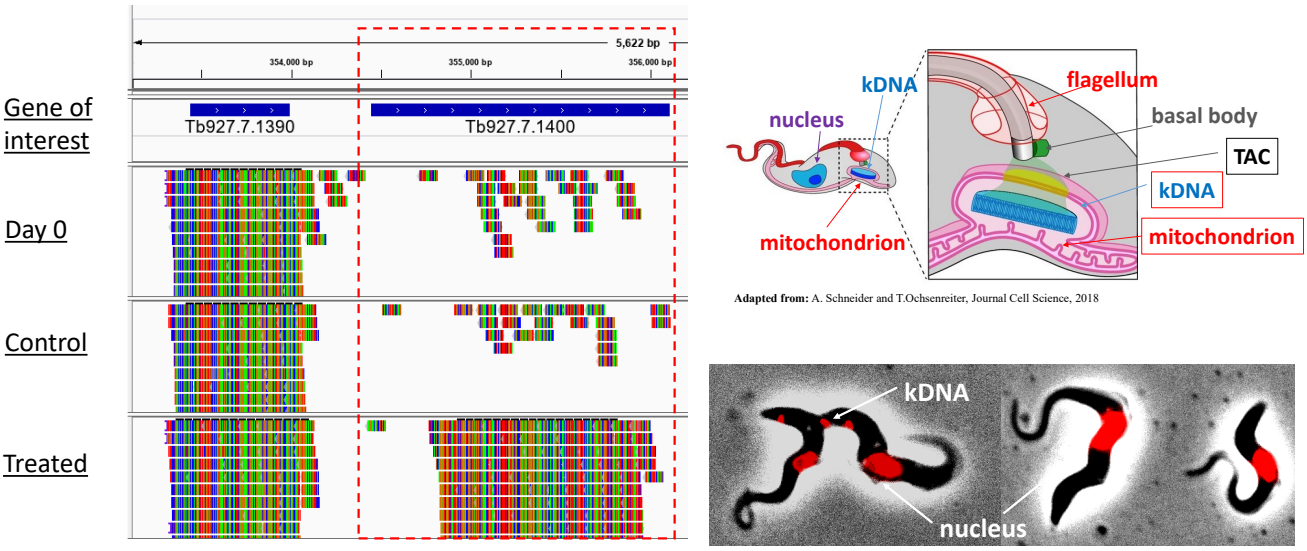
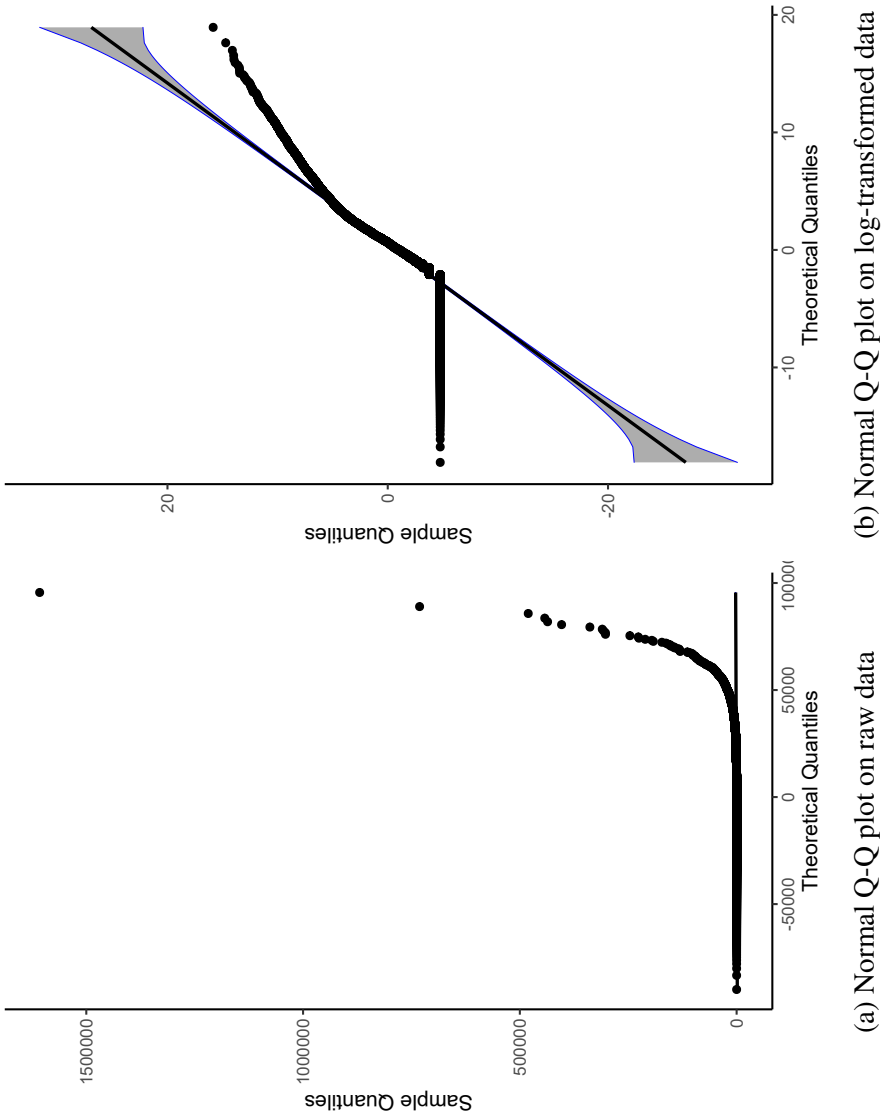
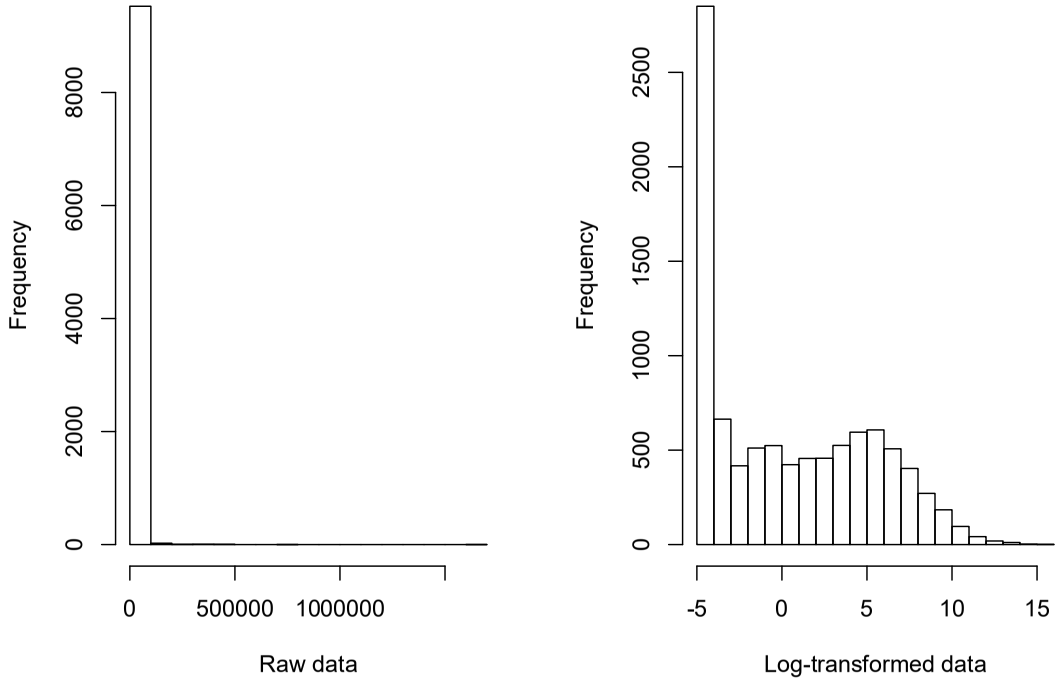


Figure S1 Description of the dataset. In treated experimental group genes of interest (that encode proteins that are localised in outlined part of *T. brucei* cell shown on the right, such as TAC, kDNA and mitochondrion, should be enriched for reads versus other two groups (D0 and Control). Example of TAC component is shown on the left.



(a) Normal Q-Q plot on raw data (b) Normal Q-Q plot on log-transformed data

Figure S2 Visualisation of data normality using QQ plot



(a) Histogram using raw data (b) Histogram using log-transformed data

Figure S3 Frequency distributions for raw and log transformed data (Sample 1 from Treated group

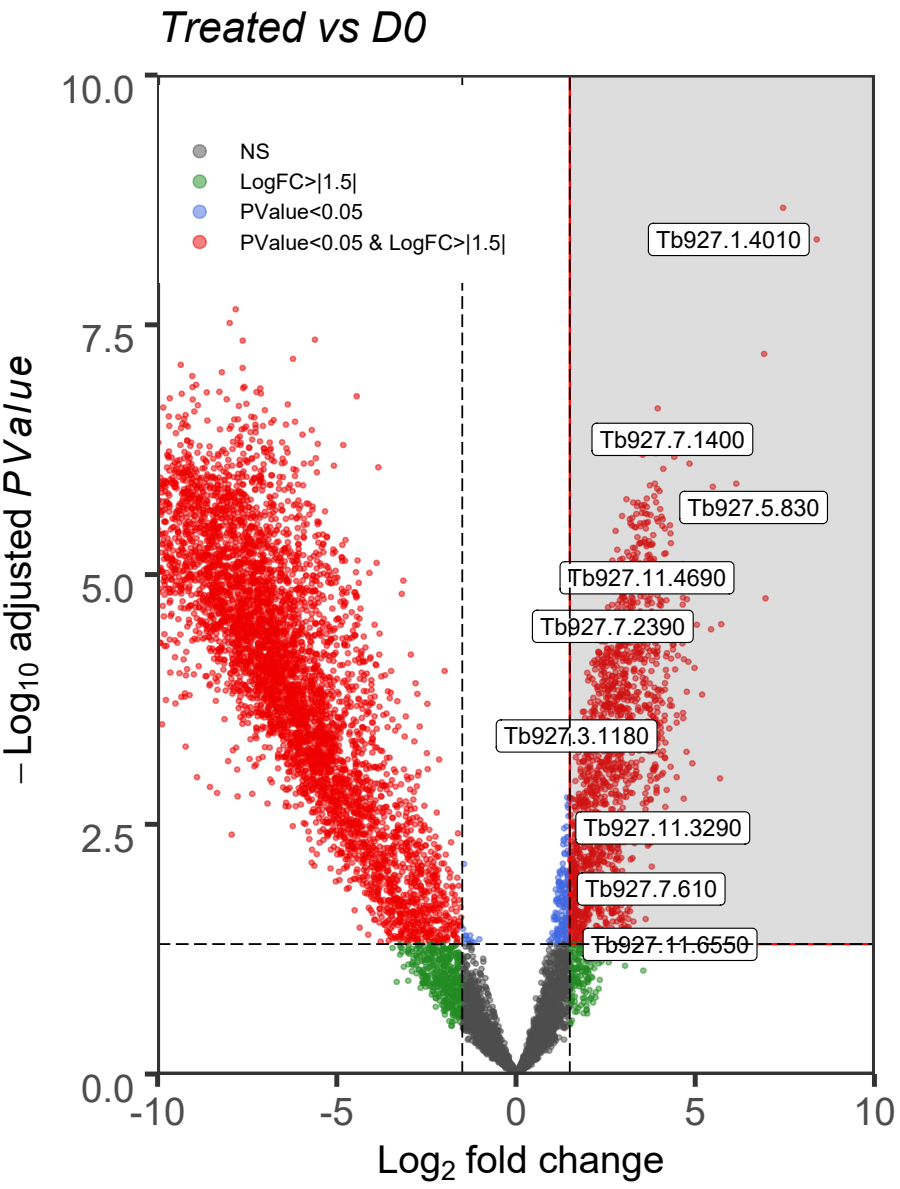


Figure S4 Volcano plot showing differentially expressed genes. Highlighted gene IDs are known kDNA factors.

Appendix B

Stan model code

```
#saved as s1.stan
data {
  int<lower=1> G;         #number of genes
  int<lower=1> S;          #number of samples
  int<lower=0> expression[G, S]; #matrix of G × S containing the expression data
  vector<lower=0, upper=1>[S] design; #this is binary design matrix
  vector<lower=0>[G] eff_length; #effective length for each gene
}
transformed data {
  vector<lower=-1, upper=1>[S] design2 = 2 * design - rep_vector(1, S);
  real <lower=0> expected_gene_mean = sum(expression[, 1])/sum(eff_length);
  #expression of all genes (sum of all counts)/divided by effective size(coverage)
  vector<lower=0>[G] log_eff_length = log(eff_length);
}
```

Define parameters to estimate.

```
parameters {
  real mu_alpha;
  real<lower=0>  sigma_alpha;
  vector[G] alpha;
  vector[G] beta;
  real<lower=0>  sigma_beta;
  vector[S-1] log_norm_factors_ref;
  real<lower=0, upper=2000000000> phi; # dispersion parameter
}
transformed parameters {
  vector[S] log_norm_factors = append_row(0, log_norm_factors_ref);
}
```

Model description

```
model {
  mu_alpha ~ normal(log(expected_gene_mean), 1.17);
  sigma_alpha ~ normal(0, 2);
  alpha ~ normal(mu_alpha, sigma_alpha);
  sigma_beta ~ std_normal(); #normal 0,1 in stan code
  beta ~ normal(0, sigma_beta);
  log_norm_factors_ref ~ normal(0, 0.05);
  phi ~ uniform(0, 2000000000);
  for (i in 1:G) {
    for (j in 1:S) {
      expression[i, j] ~ neg_binomial_2_log(alpha[i] +
      beta[i] * design2[j] + log_eff_length[i] +
      log_norm_factors[j], phi);
    }
  }
}
generated quantities {
  matrix[G, S] log_lik;
  int expression_rep[G, S];
  for (i in 1:G) {
    for (j in 1:S) {
      log_lik[i, j] = neg_binomial_2_log_lpmf(expression[i, j] | alpha[i]
      + beta[i] * design[j] + log_norm_factors[j] + log_eff_length[i], phi);

      expression_rep[i, j] = neg_binomial_2_log_rng(alpha[i]
      + beta[i] * design[j] + log_norm_factors[j] + log_eff_length[i], phi);
    }
  }
}
```