

Negative Binomial Modeling of Yeast Two-Hybrid Interactions

Valeria Velasquez-Zapata

ORCID: 0000-0002-2756-2156

Mejoramiento Genético Vegetal, Uso y Aprovechamiento de la Agrobiodiversidad (MGVA),
Corporación Colombiana de Investigación Agropecuaria (AGROSAVIA),
Centro de Investigación la Selva

June 27th 2024

Abstract

To investigate protein-protein interactions, a novel yeast two-hybrid next generation screening (Y2H-NGIS) assay has been developed. Using next generation sequencing it is possible to test thousands of interactions in batch. The assay enriches for interacting preys compared to controls, and a bait control is employed to validate results. Count data from this assay can be modeled using Negative Binomial distribution. Here I propose two approximations to the Negative Binomial modeling of Y2H-NGIS data, using differential expression analysis and processing count tables from multiple biological replicates. A frequentist and Bayesian approximations are compared and tested in a dataset using barley and powdery mildew effector dataset. The goal is to compare selection and control conditions to identify genuine interactions and minimize false positives inherent in this biological system.

1 Introduction

Hordeum vulgare, commonly referred to as barley, is frequently infected by the fungi, *Blumeria graminis* (powdery mildew). Once the fungi is established, it produces a structure known as a haustorium. This structure serves two functions. In addition to feeding, the haustorium also produces effector proteins that regulate the host's immune response. These effector proteins have been identified using mass spectrometry and bioinformatic pipelines.

These effectors are being screened for protein-protein interactions using a yeast 2-hybrid assay. The objective of these experiments is to identify what barley proteins (preys) interact at the molecular level with those effector proteins (baits). This system after selection will enrich the yeast populations that contain the interacting preys as compared with an experiment with no selection. Additionally, a bait control is proposed where a random bait is used in the screening.

The method described in (Pashkova et al, 2016), utilizes next generation sequencing to screen the yeast 2 hybrid results for each bait selection and one control sample with no selection, ranking the candidate protein interactions to identify the enriched preys in the selected conditions, and from that obtain the candidate proteins that interact with a specific bait.

In my approach to analyze the data from this experiment, I approximate this problem to a differential expression analysis. After a bioinformatic processing pipeline, my input data for this modelling is a count table for n genes under each condition, using 3 biological replicates. My objective is then to compare the counting data from selection and control to identify the enriched preys and discard false positives, which is known to be high because of the characteristics of this biological system. I propose a frequentist and a Bayesian solution to the problem and compare their residuals and running times.

2 Mathematical modeling

To start with the modeling, I propose a hierarchical Bayesian model which I will compare with a maximum likelihood estimation. I choose the negative binomial distribution model, which accounts for a high variability by using the parameter of over-dispersion(ϕ) in addition to the mean(μ). Under this model, the variance and the mean are related as follows: $\frac{\sigma^2}{\mu} = 1 + \phi\mu$.

Definition of variables:

y_i = number of counts for a gene

r = dispersion = $\frac{1}{\phi}$

ϕ = over dispersion

p = the probability of success in each event (true positive)

μ = expected number of counts for each gene

Negative binomial pdf: $f(y)_{NB} = f(y, r, p) = P(Y = y)$, with

$$\begin{aligned} f(y)_{NB} &= \frac{\Gamma(r+y)}{\Gamma(r)y!} p^y (1-p)^r \quad \text{with} \quad \mu = \frac{pr}{1-p} \quad \phi = \frac{1}{r} \\ &= \frac{\Gamma(\phi^{-1}+y)}{\Gamma(\phi^{-1})y!} \left(\frac{\phi\mu}{1+\phi\mu} \right)^y \left(\frac{1}{1+\phi\mu} \right)^{\phi^{-1}} \\ &= \frac{\Gamma(y+\phi^{-1})}{\Gamma(\phi^{-1})y!} \left(\frac{(\phi\mu)^y}{(1+\mu\phi)^{y+\phi^{-1}}} \right) \end{aligned} \quad (1)$$

I will use a generalized linear model (GLM) to establish a relationship between the counts data (y) and the covariates (x), which in the simplest model has a value of zero in the control and 1 for the selected condition.

$$Y_i | x_i, \beta, \phi \sim NB(\mu, \phi), \mu = \exp(x_i^T \beta) \quad (2)$$

Suppose the experimental design of this particular experiment involves three replicates of controls and three experimental bait screens. The three control trials can be referred to as Y_{cik} , with c standing for control, i indicating the which control replicate and k representing the gene. The three experimental bait trials are represented by Y_{bik} with b representing b, i indicating the bait replicate, and k indicating the gene.

I am interested in identifying the interactors of a bait when I compare the control and the bait screen. To model this problem, I will compare a Bayesian to a frequentist approach. As a first step I need to state an hypothesis test. The hypothesis testing can be stated as:

H_o : For an specific gene k, there is no difference in the enrichment levels of the control and bait screen samples

H_a : For an specific gene, there are differences in the enrichment levels of the control and bait screen samples

Mathematically, we can write these hypotheses as:

$H_o : Y_i \sim NB(\mu, \phi)$ with $i \in \{1, \dots, 6\}$

$H_a : Y_{ci} \sim NB(\mu_c, \phi_c)$ and $Y_{bi} \sim NB(\mu_b, \phi_b)$ with $i \in \{1, 2, 3\}$

2.1 Frequentist solution

I will use a Likelihood Ratio Test LRT and select a statistic which is sensitive to the null hypothesis, as follows:

$$\Lambda(Y_1, Y_2, \dots, Y_n) = \frac{L(\mu_o, \phi_o \mid Y_1, Y_2, \dots, Y_6)}{L(\mu_c, \phi_c, \mu_b, \phi_b \mid Y_{c1}, Y_{c2}, Y_{c3}, Y_{b1}, Y_{b2}, Y_{b3})}$$

The likelihood ratio test statistic $\lambda = -2 \log \Lambda \sim \chi_2^2$ has an asymptotic chi-squared distribution with two degrees of freedom as the difference in the number of parameters.

In appendix A I present all the details of the derivations to solve the system of equations and get MLE for ϕ, μ . In summary I will implement these equations

$$L(\phi, \mu | Y_1, \dots, Y_n) = \prod_{i=1}^n \frac{\Gamma(y_i + \phi^{-1})}{\Gamma(\phi^{-1}) y_i!} \left(\frac{(\phi \mu)^{y_i}}{(1 + \mu \phi)^{y_i + \phi^{-1}}} \right) \quad (3)$$

Derivative of the log-likelihood equation with respect to μ :

$$\begin{aligned} \frac{dl(y_i | \phi, \mu)}{d\mu} &= \sum_{i=1}^n \left[y_i \left(\frac{1}{\mu} - \frac{1}{\phi^{-1} + \mu} \right) + \phi^{-1} \left(\frac{-1}{\phi^{-1} + \mu} \right) \right] = 0 \dots \\ \hat{\mu} &= \frac{\sum y_i}{n} \end{aligned} \quad (4)$$

Derivative of the log-likelihood with respect to ϕ :

$$\begin{aligned} \frac{dl(\vec{y}, \phi, \mu)}{d\phi} &= \sum_{i=1}^n \left[\sum_{j=0}^{y_i-1} \frac{1}{j + \phi^{-1}} (-\phi^{-2}) + y_i \left[-\frac{1}{\phi^{-1} + \mu} (-\phi^{-2}) \right] + \phi^{-1} \left[\frac{1}{\phi^{-1}} (-\phi^{-2}) \right. \right. \\ &\quad \left. \left. + \frac{1}{\phi^{-1} + \mu} (-\phi^{-2}) \right] + (-\phi^{-2}) [\log(\phi^{-1}) - \log(\phi^{-1} + \mu)] \right] = 0 \dots \\ \hat{\phi} &= \sum_{i=1}^n \left[\left(\sum_{j=0}^{y_i-1} \frac{1}{j\phi + 1} \right) + \frac{\mu - y_i}{1 + \mu\phi} - \frac{1}{\phi} \log(1 + \mu\phi) \right] \end{aligned} \quad (5)$$

To finish solving this equation I can use a numerical method as for example Newton-Rhapson method, With both MLE equations, I can estimate each parameter per each gene, calculate the likelihoods and the LTR can be performed to get the p-values of each gene.

Now I have defined the elements we require to solve this problem and I can proceed to generate an algorithm which can be applied to a toy dataset. With all the elements to solve this problem, I define the following algorithm:

1. Calculate the likelihood
 - a) Determine the log likelihood
 - b) Compute the Maximum Likelihood of the data w.r.t parameters ϕ, μ
2. For each gene,
 - a) Using the calculated MLEs, plug in the MLEs into the likelihood for different datasets
 - b) Perform the likelihood ratio test
3. Get predicted values, p-values and carry out multiple testing correction

2.2 Bayesian solution

To solve the problem using Bayesian statistics I propose a hierarchical model using log links. I will use the model proposed by (Fu, 2016). We start with the data model that was already stated in equations (1) and (2) where the counts can be modelled using negative binomial distribution and the mean has a log link to a regression model with a desired number of covariates. For this model there is only one covariate that codifies selection ($x=1$) and control ($x=0$).

$$\begin{aligned} Y_i | x_i, \beta, \phi &\sim NB(\mu, \phi), \\ \log(\mu) &= \beta_0 + x_1 \beta_1 \end{aligned} \tag{6}$$

For the next level we have the process model for ϕ, β_j . According to Fu, 2016 the natural prior for β_j is a multinormal distribution, and a Gamma for ϕ as follows:

$$\begin{aligned} \beta_j | m_j, \Sigma_j &\sim N(m_j, \Sigma_j) \\ \phi | a, b &\sim G(a, b) \end{aligned} \tag{7}$$

Now we have the parameter level choosing conjugate hyperprior. For the means of the multinormal distribution for β_j we have another multinormal distribution with a smaller variance which is controlled by the hyperparameter α , the variance vector is distributed with an Inverse-Wishart distribution that is equivalent to a conjugate prior for the variance of a multivariate normal. Finally we have the parameters that define the gamma distribution for ϕ . These are also conjugate priors

$$\begin{aligned} m_j | \mu, \Sigma_j, \alpha &\sim N(m_j, \Sigma_j / \alpha) \\ \Sigma_j | \Omega, \nu &\sim IW(\Omega, \nu) \\ a_j &\propto s^{a-1} / \Gamma(a) \\ b_j | p, q &\sim G(p, q) \end{aligned} \tag{8}$$

Selection of hyperpriors: we assumed that no expert knowledge was available so I set a non-informative prior distribution as in Fu 2016:

$$\mu = [0, 0], \alpha = 1e - 6, \Sigma = \text{Diag}[1], \Omega = (\alpha + 1)\Sigma, \nu = 4, s = 0.001, l = p = q = 1, a = 2$$

Now we have defined the elements we require to solve this problem and we can proceed to generate an algorithm which can be applied to a toy dataset. I will use the Gibbs sampler

from JAGS (Plummer, 2003) and then estimate predictors and p-values for the hypothesis test:

1. Determine priors and normalization method
 - a) Use non-informative priors
 - b) Use a library-size normalization and filter out results
2. Use Gibbs sampling to calculate posterior distributions,
 - a) Calculate posterior distribution for the parameters using JAGS
 - b) Optimize the model to avoid autocorrelation
3. Get predicted values, p-values and carry out multiple testing correction

3 Implementation

I tested this methodology with the sequencing data obtained from (Velasquez-Zapata et al 2021). The data comes from a bait identified as Mla 1-161. After filtering, normalizing and running the analysis I got a total of 26 genes identified in the library. These include protein kinases, receptors, chloroplastic proteins, among others.

There were challenges to this implementation as for example zero counts so I decided to filter out genes with zero counts across replicates. The second challenge in the implementation of the Bayesian approach was autocorrelation, I found I had to reparameterize the problem with the mean and the dispersion parameter instead of p and dispersion. Increasing thinning also helped to decrease the problem, as it can be observed in figure 1. A third problem I found was related to adaptation of the model, for which I had to increase the adaptation number of iterations.

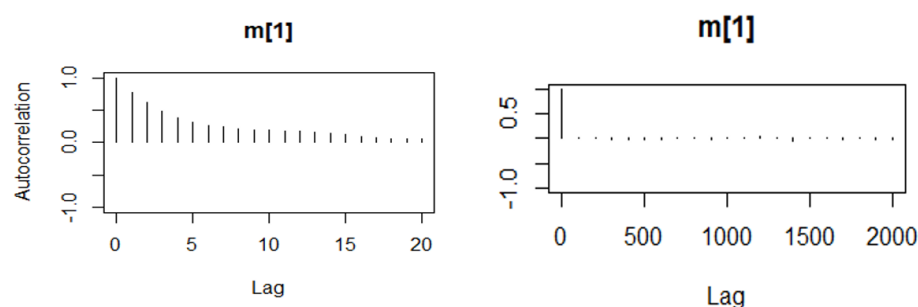


Figure 1. Autocorrelation plots for the mean parameter before and after reparameterization and thinning.

4 Results and Discussion

After getting all the estimators from maximum likelihood and the Bayesian analysis I predicted values from each model using the MLEs and the mean of the posterior distributions for μ, ϕ . Figure 2 shows these results. We can observe that the predicted values are within the range of the observed values and visually both models look very similar at that level.

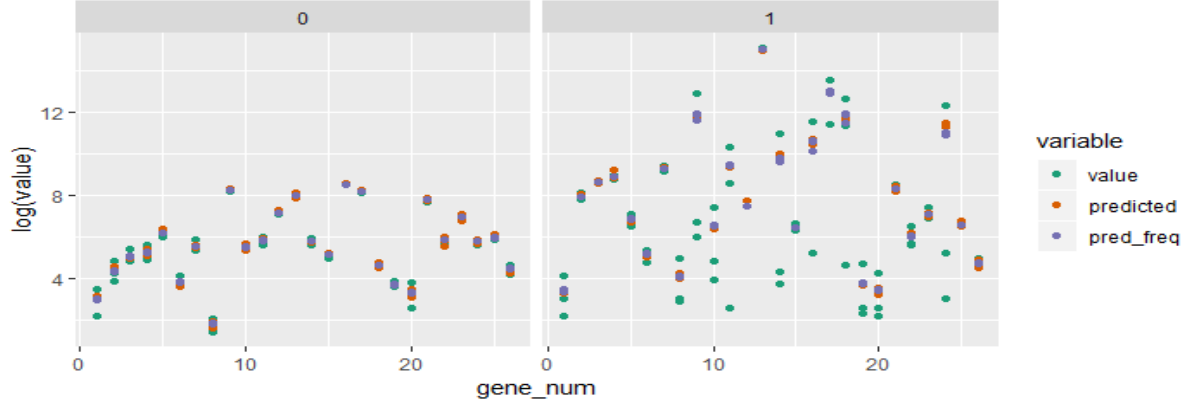


Figure 2. Observed data and predicted values for genes 1-26. Green is experimental data, orange is Bayesian predicted and purple MLE predicted. Control(0), Selected (1)

When the results are faceted by type of selection it is possible to observe that the selected sample is the one with more variation in the observations and then more variation in the predictors too. After this, residuals were calculated as $\log(\text{value}) - \log(\text{predicted})$ and plotted to compare the methods. Figure 3 shows the results and we can see that both methods behave very similarly. Both have a majority of residuals around zero with a flat slope, however there is a group of residuals which are very skewed from this behavior. These must coincide with genes with a high variation their replicates from the selected sample, confirming what we observed in the predicted plot.

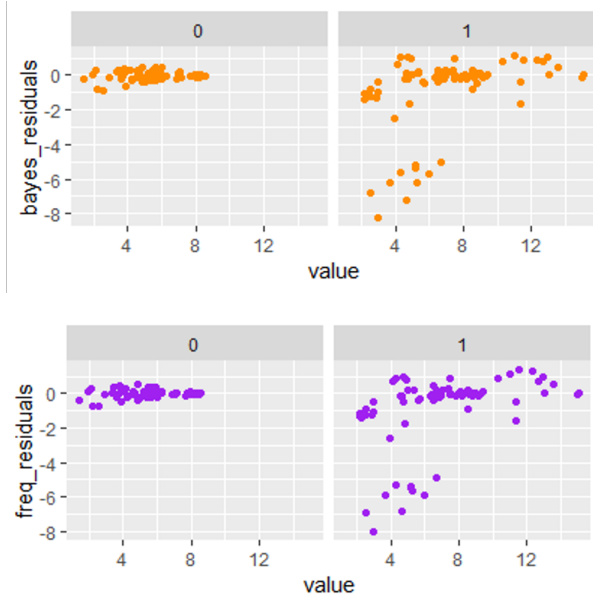


Figure 3. Residuals plots from the Bayesian and Frequentist models.

Fu (2016) analyzed both methods for a counting example of regional crash counts. He found that as the number of covariates increases the Bayesian models has better performance by decreasing the error. For the model that was implemented here I only had one covariate so I could not see that effect. Additionally, using more informative priors can help to improve the Bayesian model.

As I commented, I filtered out several cases that certainly the Bayesian model could have handled better than the frequentist approach. Those cases include very low counts and zero counts across selected replicates. Both models fail to replicate counts when there is no consistency across biological replicates, as we can see in the residuals plots.

That variation in the counts in the selected samples and the different replicates comes from the experimental conditions. These are inherent to the yeast test and the bioinformatic pipeline implemented to analyze the raw data, so improvements in these areas will certainly make more reproducible count matrices. There is one factor that we cannot control and it is related with random mutations in yeast that would allow a cell to reproduce even when it does not carry the interacting prey. That randomness is associated with the test, so identifying those outliers is important to decide is take them out of the analysis or flag them as we compare the p-values.

On the other side, the Bayesian analysis is more computationally expensive than the frequentist so both implementations have advantages and disadvantages. On the Bayesian side the optimization of the running parameters can be challenging as I experienced when I tried to run it, and on the frequentist side there are limitations on the number of cases that it can handle. In order to make a reproducible analysis it is necessary to take into account those factors and decide according to the specifics of the dataset.

5 Conclusions

The negative binomial regression is an useful way of analyzing count data. As a generalization of the Poisson regression it allows for accounting for variance from environmental factors represented by the dispersion parameter. Many systems in biology and bioinformatics are modelled using this distribution and improvements in their implementation and accuracy is required to predict better the biology in the background.

As a future direction I am interested in testing this methodology to compare different baits, which requires a different hypothesis test and regression model that allows for several bait-control screenings. This will provide more information about non-specific preys which are interacting with a large number of baits, and this can be used to identify other false positives in the system.

6 Bibliography

Fu, S (2016). A hierarchical Bayesian approach to negative binomial regression. *Methods and Applications of Analysis* 22(4), 409-428.

<http://dx.doi.org/10.4310/MAA.2015.v22.n4.a4>

Pashkova, N., Peterson, T. A., Krishnamani, V., Breheny, P., Stamnes, M., and Piper, R. C. (2016). DEEPN as an Approach for Batch Processing of Yeast 2-Hybrid Interactions. *Cell Reports*, 17(1), 303–315. <https://doi.org/10.1016/j.celrep.2016.08.095>

Plummer, M. (2003). JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling.

Suter, B., Zhang, X., Gustavo Pesce, C., Mendelsohn, A. R., Dinesh-Kumar, S. P., Mao, J. H. (2015). Next-generation sequencing for binary protein-protein interactions. *Frontiers in Genetics*, 6(DEC), 1–6. <https://doi.org/10.3389/fgene.2015.00346>

Velásquez-Zapata, V., Elmore, J.M., Banerjee, S., Dorman, K.S. and Wise, R.P. (2021) Next-generation yeast-two-hybrid analysis with Y2H-SCORES identifies novel interactors of the MLA immune receptor. *PLoS Comput. Biol.*, 17, e1008890.

7 Appendix A

Derivation for maximum likelihood estimation: the first step would be to find the likelihood of this problem, using the negative binomial distribution and assuming that all samples are iid.

$$L(\phi, \mu | Y_1, \dots, Y_n) = \prod_{i=1}^n \frac{\Gamma(y_i + \phi^{-1})}{\Gamma(\phi^{-1}) y_i!} \left(\frac{(\phi\mu)^{y_i}}{(1 + \mu\phi)^{y_i + \phi^{-1}}} \right) \quad (9)$$

Now, we calculate the log-likelihood.

$$\begin{aligned} l(\phi, \mu | Y_1, \dots, Y_n) &= \sum_{i=1}^n \left[\left[\log \frac{\Gamma(\phi^{-1} + y_i)}{\Gamma(\phi^{-1})} \right] - \log(y_i!) + y_i [\log(\mu) - \log(\phi^{-1} + \mu)] + \phi^{-1} [\log(\phi^{-1}) - \log(\phi^{-1} + \mu)] \right] \\ &= \sum_{i=1}^n \left[\left[\log \frac{\Gamma(\phi^{-1} + y_i)}{\Gamma(\phi^{-1})} \right] - \log(y_i!) + y_i \log \mu + y_i \log(\phi) - (y_i + \phi^{-1}) \log(1 + \phi\mu) \right] \\ &= \sum_{i=1}^n \left[\left[\sum_{j=0}^{y_i-1} \log(j + \phi^{-1}) \right] - \log(y_i!) + y_i \log \mu + y_i \log(\phi) - (y_i + \phi^{-1}) \log(1 + \phi\mu) \right] \end{aligned} \quad (10)$$

With

$$\log \frac{\Gamma(\phi^{-1} + y_i)}{\Gamma(\phi^{-1})} = \log \frac{(\phi^{-1} + y_i - 1)!}{(\phi^{-1} - 1)!} = \log \frac{(\phi^{-1} + y_i - 1) \dots (\phi^{-1}) (\phi^{-1} - 1)!}{(\phi^{-1} - 1)!} = \sum_{j=0}^{y_i-1} \log(j + \phi^{-1}) \quad (11)$$

The following step is to use the log-likelihood equation to calculate the MLE estimators for the parameters. Now, we maximize it with respect to the parameters μ and ϕ .

Derivative of the log-likelihood equation with respect to μ :

$$\begin{aligned}
\frac{dl(y_i|\phi, \mu)}{d\mu} &= \sum_{i=1}^n \left[y_i \left(\frac{1}{\mu} - \frac{1}{\phi^{-1} + \mu} \right) + \phi^{-1} \left(\frac{-1}{\phi^{-1} + \mu} \right) \right] = 0 \\
&= \sum_{i=1}^n \left[\frac{y_i}{\mu} - \frac{y_i}{\phi^{-1} + \mu} - \frac{\phi^{-1}}{\phi^{-1} + \mu} \right] = 0 \\
&= \sum_{i=1}^n \left[\frac{y_i}{\mu} - \frac{(y_i + \phi^{-1})}{\phi^{-1} + \mu} \right] = 0 \\
\sum_{i=1}^n \frac{y_i}{\mu} &= \sum_{i=1}^n \frac{(y_i + \phi^{-1})}{\phi^{-1} + \mu} \\
\frac{1}{\mu} \sum_{i=1}^n y_i &= \frac{1}{\phi^{-1} + \mu} \sum_{i=1}^n (y_i + \phi^{-1}) \\
(\phi^{-1} + \mu) \sum_{i=1}^n y_i &= \mu \sum_{i=1}^n (y_i + \phi^{-1}) \tag{12} \\
\phi^{-1} \sum_{i=1}^n y_i &= \mu \sum_{i=1}^n y_i = \mu \sum_{i=1}^n (y_i + \phi^{-1}) \\
\phi^{-1} \sum_{i=1}^n y_i &= \mu \sum_{i=1}^n (y_i + \phi^{-1}) + \mu \left(\sum_{i=1}^n y_i \right) - \mu \left[\sum_{i=1}^n y_i + \phi^{-1} - y_i \right] \\
\mu &= \frac{\phi^{-1} \sum_{i=1}^n y_i}{\sum_{i=1}^n \phi^{-1}} \\
\mu &= \frac{\phi^{-1} \sum_{i=1}^n y_i}{\phi^{-1} n} \\
\hat{\mu} &= \frac{\sum y_i}{n}
\end{aligned}$$

Derivative of the log-likelihood with respect to ϕ :

$$\begin{aligned}
\frac{dl(\vec{y}, \phi, \mu)}{d\phi} &= \sum_{i=1}^n \left[\sum_{j=0}^{y_i-1} \frac{1}{j + \phi^{-1}} (-\phi^{-2}) + y_i \left[-\frac{1}{\phi^{-1} + \mu} (-\phi^{-2}) \right] + \phi^{-1} \left[\frac{1}{\phi^{-1}} (-\phi^{-2}) \right. \right. \\
&\quad \left. \left. + \frac{1}{\phi^{-1} + \mu} (-\phi^{-2}) \right] + (-\phi^{-2}) [\log(\phi^{-1}) - \log(\phi^{-1} + \mu)] \right] = 0 \\
&= \sum_{i=1}^n \left[\sum_{j=0}^{y_i-1} \frac{1}{j + \phi^{-1}} - \frac{y_i}{\phi^{-1} + \mu} + 1 - \frac{\phi^{-1}}{\phi^{-1} + \mu} + \log(\phi^{-1}) - \log(\phi^{-1} + \mu) \right] = 0 \\
&= \sum_{i=1}^n \left[\sum_{j=0}^{y_i-1} \frac{\phi}{j\phi + 1} + \frac{\phi^{-1} + \mu - y_i - \phi^{-1}}{\phi^{-1} + \mu} - \log(1 + \mu\phi) \right] = 0 \\
&= \sum_{i=1}^n \left[\sum_{j=0}^{y_i-1} \frac{\phi}{j\phi + 1} + \frac{\phi(\mu - y_i)}{1 + \mu\phi} - \log(1 + \mu\phi) \right] = 0 \\
\hat{\phi} &= \sum_{i=1}^n \left[\left(\sum_{j=0}^{y_i-1} \frac{1}{j\phi + 1} \right) + \frac{\mu - y_i}{1 + \mu\phi} - \frac{1}{\phi} \log(1 + \mu\phi) \right]
\end{aligned} \tag{13}$$