Sequencing: Technical topics

Koen Van den Berge

7/5/2021

Contents

1 Aliasing

Suppose we are working with the following experimental design. Studying the effect of alcoholism on gene expression, researchers gather RNA-seq data from four alcoholic individuals and four healthy individuals. For each individual, they obtain RNA-seq data from a blood sample as well as liver tissue. The research question relates to differential expression between the diseased versus control conditions, in both tissues.

If we would like to model all data simultaneously, then we could imagine a design such as `~ patient + disease*tissue`, where

- `disease` is a binary indicator referring to alcoholic versus control sample.
- `tissue` defines if the sample is a liver or blood sample.
- `patient` defines the individual donor the sample comes from.

Let's try this, by simulating random data for one gene.

```
set.seed(2)
patient <- factor(rep(letters[1:8], each=2)) # 2 samples per patient
disease <- factor(c(rep("healthy",8), rep("alcohol",8))) # first four are healthy, next four are alcoho
tissue <- factor(rep(c("blood", "liver"), 8)) # one liver and one blood sample for each

table(patient, disease, tissue)
```

```
## , , tissue = blood
##
##        disease
## patient alcohol healthy
##       a       0       1
##       b       0       1
##       c       0       1
```

```
##       d       0       1
##       e       1       0
##       f       1       0
##       g       1       0
##       h       1       0
##
## , , tissue = liver
##
##        disease
## patient alcohol healthy
##       a       0       1
##       b       0       1
##       c       0       1
##       d       0       1
##       e       1       0
##       f       1       0
##       g       1       0
##       h       1       0
```

```
## simulate data for one gene
n <- 16
y <- rpois(n = n, lambda = 50)

## fit a Poisson model
m <- glm(y ~ patient + disease*tissue,
         family = "poisson")
summary(m)
```

```
##
## Call:
## glm(formula = y ~ patient + disease * tissue, family = "poisson")
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.52772  -0.43544   0.00013   0.44162   1.34650
##
## Coefficients: (1 not defined because of singularities)
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  3.76900    0.11916  31.631   <2e-16 ***
## patientb                     0.06744    0.14999   0.450   0.6530
## patientc                     0.06744    0.14999   0.450   0.6530
## patientd                     0.27304    0.14310   1.908   0.0564 .
## patiente                     0.16449    0.16224   1.014   0.3107
## patientf                     0.02565    0.16644   0.154   0.8775
## patientg                    -0.01784    0.16785  -0.106   0.9154
## patienth                     0.05706    0.16544   0.345   0.7302
## diseasehealthy                    NA         NA      NA       NA
## tissueliver                  0.10807    0.10155   1.064   0.2872
## diseasehealthy:tissueliver -0.12374    0.14407  -0.859   0.3904
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
```

```
##     Null deviance: 16.1200  on 15  degrees of freedom
## Residual deviance:  8.8417  on  6  degrees of freedom
## AIC: 120.16
##
## Number of Fisher Scoring iterations: 4
```

We find that one of the coefficients is `NA`! This is obviously not because we're dealing with `NA` values in the data as we've just simulated the response variable.

Instead, one of the parameters, in this case the parameter distinguising diseased from healthy patients **cannot be estimated as it is a linear combination of other parameters**. In our case, estimating the diseased effect would use information that is already used to estimate the patient-level intercepts. In other words, **once you know the patient, you immediately also know the diseased status**, so estimating the diseased vs healthy effect on top of the patient effect provides no additional information if we have already estimated the patient-level effects. This concept is called aliasing, and is common in RNA-seq experiments with complex experimental designs.

While to understand the origin of the aliasing it is crucial to understand the relationship between the variables in the experimental design, we can also investigate it in detail using the `alias` function, to give us an idea.

```
alias(m)
```

```
## Model :
## y ~ patient + disease * tissue
##
## Complete :
##                 (Intercept) patientb patientc patientd patiente patientf
## diseasehealthy  1           0        0        0       -1       -1
##                 patientg patienth tissueliver diseasehealthy:tissueliver
## diseasehealthy -1       -1        0           0
```

We see that the effect `diseasehealthy` is a linear combination of the the intercept minus the patient-specific effects of the alcoholic patients. This makes sense! For clarity, let's reproduce this using our design matrix.

```
X <- model.matrix(~ patient + disease*tissue)

## these are indeed the same.
X[,"diseasehealthy"]
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
##  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0
```

```
X[,"(Intercept)"] - X[,"patiente"] - X[,"patientf"] - X[,"patientg"] - X[,"patienth"]
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
##  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0
```

Since one of our parameters is a linear combination of other parameters, it cannot be estimated simultaneously with the other parameters. In this case, we can actually drop the `tissue` effect from the model, since we know that it is already included in the `patient` effect. The most convenient way will be to make a new variable that represents the interaction between patient and tissue. This covariate will contain all the information we need. Below, we fit it using a model without an intercept; this eases interpretation as each coefficient can be interpreted as the average expression for that particular tissue of that particular patient.

```
patientTissue <- as.factor(paste0(tissue, "_", patient))
patientTissue
```

```
##  [1] blood_a liver_a blood_b liver_b blood_c liver_c blood_d liver_d blood_e
## [10] liver_e blood_f liver_f blood_g liver_g blood_h liver_h
## 16 Levels: blood_a blood_b blood_c blood_d blood_e blood_f blood_g ... liver_h
```

```
m2 <- glm(y ~ -1 + patientTissue,
          family = "poisson")

summary(m2)
```

```
##
## Call:
## glm(formula = y ~ -1 + patientTissue, family = "poisson")
##
## Deviance Residuals:
##  [1]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## patientTissueblood_a    3.7612     0.1525   24.66   <2e-16 ***
## patientTissueblood_b    3.7377     0.1543   24.22   <2e-16 ***
## patientTissueblood_c    3.7842     0.1508   25.10   <2e-16 ***
## patientTissueblood_d    4.1589     0.1250   33.27   <2e-16 ***
## patientTissueblood_e    3.9512     0.1387   28.49   <2e-16 ***
## patientTissueblood_f    3.8501     0.1459   26.39   <2e-16 ***
## patientTissueblood_g    3.4965     0.1741   20.09   <2e-16 ***
## patientTissueblood_h    3.9512     0.1387   28.49   <2e-16 ***
## patientTissueliver_a    3.7612     0.1525   24.66   <2e-16 ***
## patientTissueliver_b    3.9120     0.1414   27.66   <2e-16 ***
## patientTissueliver_c    3.8712     0.1443   26.82   <2e-16 ***
## patientTissueliver_d    3.8918     0.1429   27.24   <2e-16 ***
## patientTissueliver_e    4.0254     0.1336   30.12   <2e-16 ***
## patientTissueliver_f    3.8501     0.1459   26.39   <2e-16 ***
## patientTissueliver_g    4.0431     0.1325   30.52   <2e-16 ***
## patientTissueliver_h    3.8067     0.1491   25.54   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance:  4.4893e+03  on 16  degrees of freedom
## Residual deviance: -2.0206e-14  on  0  degrees of freedom
## AIC: 123.32
##
## Number of Fisher Scoring iterations: 3
```

Question. What do the p-values mean here? Why are they all significant?

Answer.

Since now we are working with an intercept-free model, each coefficient represents the average gene expression of that patient for the particular tissue. The coefficients therefore can no longer be interpreted as a difference with respect to the intercept on the linear predictor scale. The null hypothesis is still the same as usual, i.e., the p-value corresponding to β_2 tests the null hypothesis $H_0 : \beta_2 = 0$. Thus, the test is now assessing whether the average gene expression is different from zero (rather than different from the intercept).

We see that all coefficients can now be estimated.

2 Independent filtering

3 Other approaches to modeling counts: limma-voom