

# Chicken Proteomics Data

- “Chicken Corneocyte Cross-Linked Proteome,” Robert H. Rice, Brett R. Winters, Blythe P. Durbin-Johnson, and David M. Rocke, *J. Proteome Res.*, 2013, 12 (2), pp 771–776.
- 32 samples
- Four chickens
- Beak, Claw, Feather, Scale
- Soluble fraction, insoluble fraction
- 224 identified proteins

# Chicken.Proteomics.xlsx

Column	Content
1 (A)	Row Number (Obs)
2 (B)	Protein
3 (C)	Accession Number
4 (D)	MW in kDa
5-36 (E-AJ)	Sample number 1-32

# Chicken.Factors.txt

- 32 by 3 table with a header row
- Columns are Chicken (1-4), Component (Beak, Claw, Feather, Scale), and Fraction (Soluble, Insoluble).

# Data processing

- Read Chicken.Proteomics.xls into R
  - You could save the sheet as a tab delimited text file and read with `read.delim()`
  - You could save the sheet as a .csv file and read with `read.csv()`
  - You could install and load the CRAN library `xlsx`
    - You need to install java for this to work
    - `chick <- read.xlsx("Chicken.Proteomics.xlsx", sheetIndex = 1, stringsAsFactors = F)`
- Extract columns 5–36 as a matrix of counts

# Data processing

- Read Chicken.Factors.txt using `read.delim()`.
- Replace the “Chicken” variable with a factor (otherwise “Chicken” is a number).
- Omit proteins (rows of the data matrix) where the number of zeroes is too high (for example, greater than 75% of the sample size, which is  $(0.75)(32) = 24$ ).

```

require(xlsx)
require(multcomp)
chick <- read.xlsx("Chicken.Proteomics.xlsx",sheetIndex=1,stringsAsFactors=F)
chickmat <- chick[,5:36]
vars <- read.delim("Chicken.Factors.txt")
vars$Chicken <- as.factor(vars$Chicken)
print(table(apply(chickmat==0,1,sum)))
rowzeroes <- apply(chickmat==0,1,sum)
maxz <- 24
chickmat <- chickmat[rowzeroes <= maxz,]

glms <- function(mat,vars)
{
  m <- nrow(mat)
  pvmat <- matrix(rep(0,3*m),ncol=3)
  for (i in 1:m)
  {
    new.dat <- data.frame(vars,t(mat[i,]))
    names(new.dat)[4] <- "y"
    gobj <- glm(y~Chicken+Component+Fraction,data=new.dat,family=quasipoisson)
    pvs <- drop1(gobj,test="Chisq")[-1,4]
    pvmat[i,] <- pvs
  }
  return(pvmat)
}

```

# Exercise 1

- Fit the main effects model. The Chicken effect is not important in itself, but is included to account for animal-to-animal variability. The main important effect is Component, but if Fraction is also significant, there may be an interaction and if there is a correlation, the effects could be confounded. There are 3 p-values to save.

# Exercise 2

- Fit the interaction model  $\text{Chicken} + \text{Component} * \text{Fraction}$ . Using `drop1()`, there is only one p-value saved, the one for the interaction. Considering the interaction, there are two ways for a protein to be interesting:
  - The  $\text{Component}:\text{Fraction}$  term is significant or
  - The interaction is not significant, but in the main effects model,  $\text{Component}$  is significant.
- Using FDR-adjusted p-values and a threshold of 0.10, find all proteins that are significant by this standard.



# Exercise 3

- Fit the model `Chicken+Component` to each fraction separately. The `drop1()` command will yield two p-values in this case. The p-value for `Component` is the one of interest.
- Using the same threshold as in the last problem, find all proteins that are significant either in the soluble fraction or in the insoluble fraction. You can perform the FDR p-value adjustment either for the two fraction separately or from the combined list.
- Compare the two lists (Exercise 2 and Exercise 3).

# Exercise 4

- Use the multcomp package to find differences among different components.
- Do this separately for soluble and insoluble proteins.
- For each protein where there is a significant effect of component for the soluble fraction, find which differences are significant.
- Do the same for the insoluble fraction

```
> ghlobj <- glht(gobj,linfct=mcp(Component = "Tukey"))
> summary(ghlobj)
> plot(ghlobj)
> summary(ghlobj)$test$pvalues # vector of length 6
```

## Simultaneous Tests for General Linear Hypotheses

### Multiple Comparisons of Means: Tukey Contrasts

```
Fit: glm(formula = y ~ Chicken + Component, family = quasipoisson,
  data = new.dat, subset = (Fraction == "Soluble"))
```

#### Linear Hypotheses:

	Estimate	Std. Error	z value	Pr(> z )	
Claw - Beak == 0	-0.1560	0.1273	-1.226	0.586	
Feather - Beak == 0	-2.8904	0.3766	-7.674	<1e-04	***
Scale - Beak == 0	-1.0033	0.1668	-6.014	<1e-04	***
Feather - Claw == 0	-2.7344	0.3783	-7.228	<1e-04	***
Scale - Claw == 0	-0.8473	0.1706	-4.968	<1e-04	***
Scale - Feather == 0	1.8871	0.3934	4.797	<1e-04	***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
(Adjusted p values reported -- single-step method)

```
> tapply(tmp.df$y,list(tmp.df$Component,tmp.df$Fraction),mean)
```

	Insoluble	Soluble
Beak	78.25	67.50
Claw	85.25	57.75
Feather	18.50	3.75
Scale	74.50	24.75