

R Code for GLMs and model comparison in Dalziel et al. 2018

Read data:

```
flu <- read.csv("influenza_cities.csv")
names(flu)
```

```
## [1] "zip"           "population_size" "kappa"
## [4] "nu"           "avgsh"           "sdsh"
## [7] "lon"          "lat"             "h_crowding"
## [10] "w_crowding"
```

```
keep <- complete.cases(flu)
flu <- flu[keep,]
rm(keep)
```

Fit crowding as a function of population size:

```
#residential crowding:
hcrowd_pop <- glm(log(h_crowding) ~ log(population_size), data = flu)
summary(hcrowd_pop)
```

```
##
## Call:
## glm(formula = log(h_crowding) ~ log(population_size), data = flu)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.22610  -0.28681  -0.03627   0.30772   2.29737
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.63869    0.28781  -2.219   0.0269 *
## log(population_size)  0.41169    0.02264  18.187  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.2014433)
##
##      Null deviance: 186.90  on 598  degrees of freedom
## Residual deviance: 120.26  on 597  degrees of freedom
## AIC: 744.14
##
## Number of Fisher Scoring iterations: 2
```

```
#daytime crowding:
wcrowd_pop <- glm(log(w_crowding) ~ log(population_size), data = flu)
summary(wcrowd_pop)
```

```
##
## Call:
## glm(formula = log(w_crowding) ~ log(population_size), data = flu)
##
```

```
## Deviance Residuals:
##      Min        1Q      Median        3Q        Max
## -1.1860   -0.5105   -0.1579    0.3313    3.7485
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.17904    0.46372  -0.386    0.7
## log(population_size)  0.53780    0.03647  14.746 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.5229336)
##
##      Null deviance: 425.90  on 598  degrees of freedom
## Residual deviance: 312.19  on 597  degrees of freedom
## AIC: 1315.6
##
## Number of Fisher Scoring iterations: 2
```

Fit base transmission potential (kappa) as a function of population size and crowding

```
kappa_pop <- glm(log(kappa) ~ log(population_size), data = flu)
summary(kappa_pop)
```

```
##
## Call:
## glm(formula = log(kappa) ~ log(population_size), data = flu)
##
## Deviance Residuals:
##      Min        1Q      Median        3Q        Max
## -0.43489   -0.07958    0.00826    0.07874    0.37473
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.050138    0.075369  -13.93 <2e-16 ***
## log(population_size)  0.231399    0.005928   39.04 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.01381437)
##
##      Null deviance: 29.2983  on 598  degrees of freedom
## Residual deviance:  8.2472  on 597  degrees of freedom
## AIC: -861.06
##
## Number of Fisher Scoring iterations: 2
```

```
kappa_popXcrowd <- glm(log(kappa) ~ log(population_size)*log(h_crowding)*log(w_crowding), data = flu)
summary(kappa_popXcrowd)
```

```
##
## Call:
## glm(formula = log(kappa) ~ log(population_size) * log(h_crowding) *
##      log(w_crowding), data = flu)
##
## Deviance Residuals:
```

```
##      Min      1Q      Median      3Q      Max
## -0.45469 -0.07227  0.00981  0.07629  0.34636
##
## Coefficients:
##
##              Estimate Std. Error
## (Intercept)      6.463602   3.644330
## log(population_size) -0.371283   0.300631
## log(h_crowding)     -1.593476   0.739319
## log(w_crowding)     -0.838555   0.578620
## log(population_size):log(h_crowding)  0.127801   0.060458
## log(population_size):log(w_crowding)  0.063371   0.046952
## log(h_crowding):log(w_crowding)      0.184695   0.114347
## log(population_size):log(h_crowding):log(w_crowding) -0.013984   0.009224
##
##              t value Pr(>|t|)
## (Intercept)      1.774   0.0766 .
## log(population_size) -1.235   0.2173
## log(h_crowding)     -2.155   0.0315 *
## log(w_crowding)     -1.449   0.1478
## log(population_size):log(h_crowding)  2.114   0.0349 *
## log(population_size):log(w_crowding)  1.350   0.1776
## log(h_crowding):log(w_crowding)      1.615   0.1068
## log(population_size):log(h_crowding):log(w_crowding) -1.516   0.1300
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.01285076)
##
##      Null deviance: 29.2983  on 598  degrees of freedom
## Residual deviance:  7.5948  on 591  degrees of freedom
## AIC: -898.42
##
## Number of Fisher Scoring iterations: 2
```

Define predicted kappas from census data using both population size and crowding to use when we want a measure of kappa that derives from census data (not incidence data):

```
flu$kpred <- kappa_popXcrowd$fitted.values
```

Define *excess kappa* as the residual variation in base transmission not explained by population size and *excess crowding* as the residual variation in crowding not explained by population size:

```
flu$exkappa <- kappa_pop$residuals
flu$exhcrowd <- hcrowd_pop$residuals
flu$exwcrowd <- wcrowd_pop$residuals
```

Fit excess kappa as a function of excess crowding:

```
exkappa_exhcrowd <- glm(exkappa ~ exhcrowd, data = flu)
summary(exkappa_exhcrowd)
```

```
##
## Call:
## glm(formula = exkappa ~ exhcrowd, data = flu)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.45881 -0.07414  0.01359  0.07671  0.31138
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.185e-15  4.694e-03   0.000      1
## exhcrowd    5.540e-02  1.048e-02   5.289 1.73e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.01319609)
##
## Null deviance: 8.2472 on 598 degrees of freedom
## Residual deviance: 7.8781 on 597 degrees of freedom
## AIC: -888.49
##
## Number of Fisher Scoring iterations: 2
exkappa_exwcrowd <- glm(exkappa ~ exwcrowd, data = flu)
summary(exkappa_exwcrowd)
```

```
##
## Call:
## glm(formula = exkappa ~ exwcrowd, data = flu)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44068 -0.07568  0.00679  0.07955  0.36937
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.284e-15  4.783e-03   0.00  1.0000
## exwcrowd    1.444e-02  6.626e-03   2.18  0.0297 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.01370529)
##
## Null deviance: 8.2472 on 598 degrees of freedom
## Residual deviance: 8.1821 on 597 degrees of freedom
## AIC: -865.81
##
## Number of Fisher Scoring iterations: 2
```

Fit epidemic intensity to population size, transmission potential (estimated independently from census data), and specific humidity:

```
nu_pop <- glm(log(nu) ~ log(population_size), data = flu)
nu_sh <- glm(log(nu) ~ avgsh * sdsh + I(avgsh^2) + I(sdsh^2), data = flu)
nu_kappa <- glm(log(nu) ~ log(kpred), data = flu)
nu_popXsh <- glm(log(nu) ~ log(population_size) * avgsh * sdsh + I(avgsh^2) + I(sdsh^2), data = flu)
nu_kappaXsh <- glm(log(nu) ~ log(kpred) * avgsh * sdsh + I(avgsh^2) + I(sdsh^2), data = flu)
```

Do model comparison for the intensity GLMs

```
modcomp <- AIC(nu_pop, nu_sh, nu_kappa, nu_popXsh, nu_kappaXsh)
modcomp$Model <- c("Population size (P)", "Specific humidity (SH)", "Base transmission (K)", "P x SH", "K x SH", "P x K")
modcomp <- modcomp[,c(3,1,2)]
```

```
rownames(modcomp) <- NULL
modcomp <- modcomp[order(modcomp$AIC),]
modcomp$deltaAIC <- modcomp$AIC - modcomp$AIC[1]
modcomp
```

```
##           Model df      AIC deltaAIC
## 5           K x SH 11  931.4079   0.0000
## 4           P x SH 11  969.3593  37.9514
## 2 Specific humidity (SH) 7 1121.0814 189.6735
## 3 Base transmission (K) 3 1142.9572 211.5493
## 1 Population size (P) 3 1183.9871 252.5793
```

Bar plot of deltaAICs for intensity models

```
pal <- 1 #brewer.pal(5, "Set1")
x <- barplot(modcomp$deltaAIC, xlab = expression(paste(Delta, "AIC")), ylab="Model", srt=45, horiz=T, col=pal,
  text(rep(30,5), x, modcomp$Model, pos=4, col='white', cex=0.95)
  text(35, x[2], modcomp$Model[2], pos=4, font=1, cex=0.95)
  text(35, x[1], modcomp$Model[1], pos=4, font=2, cex=0.95)
```

