

2.2 Perfect separation (Brucella/Salmonella)

Nicolas Bennett

2015-05-29

The previous investigation, concerned with comparing several glm packages showed issues with perfect separation, which poses problems for finding ML estimates for the affected variables (they don't exist, as the corresponding coefficients are allowed to grow to infinity). The question remains whether this situation is specific to those circumstances or if it can be replicated in many different settings.

In terms of glm routines, as the main interest lies in detection of complete separation, only working with the standard glm function would suffice. For comparison, also *glmnet* and *bayesglm* are included.

```
mtor.loc <- findWells(pathogens=c("brucella", "salmonella"),
                      experiments="du-k1", contents="MTOR")
## there are 8 wells remaining:
##   J101-2C H6   SIRNA DHARMACON_L-003008-00_A 2475 MTOR
##   J104-2C H6   SIRNA DHARMACON_L-003008-00_B 2475 MTOR
##   J107-2C H6   SIRNA DHARMACON_L-003008-00_C 2475 MTOR
##   J110-2C H6   SIRNA DHARMACON_L-003008-00_D 2475 MTOR
##   J101-2L H6   SIRNA DHARMACON_L-003008-00_A 2475 MTOR
##   J104-2L H6   SIRNA DHARMACON_L-003008-00_B 2475 MTOR
##   J110-2L H6   SIRNA DHARMACON_L-003008-00_D 2475 MTOR
##   J107-2L H6   SIRNA DHARMACON_L-003008-00_C 2475 MTOR
other.loc <- findWells(plates=sapply(mtor.loc, getBarcode),
                      well.names=c("H7", "I6"))
## there are 16 wells remaining:
##   J101-2C H7   SIRNA DHARMACON_L-007730-00_A 3984 LIMK1
##   J101-2C I6   SIRNA DHARMACON_L-004259-00_A 4139 MARK1
##   J104-2C H7   SIRNA DHARMACON_L-007730-00_B 3984 LIMK1
##   J104-2C I6   SIRNA DHARMACON_L-004259-00_B 4139 MARK1
##   J107-2C H7   SIRNA DHARMACON_L-007730-00_C 3984 LIMK1
##   J107-2C I6   SIRNA DHARMACON_L-004259-00_C 4139 MARK1
##   J110-2C H7   SIRNA DHARMACON_L-007730-00_D 3984 LIMK1
##   J110-2C I6   SIRNA DHARMACON_L-004259-00_D 4139 MARK1
##   J101-2L H7   SIRNA DHARMACON_L-007730-00_A 3984 LIMK1
##   J101-2L I6   SIRNA DHARMACON_L-004259-00_A 4139 MARK1
##   J104-2L H7   SIRNA DHARMACON_L-007730-00_B 3984 LIMK1
##   J104-2L I6   SIRNA DHARMACON_L-004259-00_B 4139 MARK1
##   J110-2L H7   SIRNA DHARMACON_L-007730-00_D 3984 LIMK1
##   J110-2L I6   SIRNA DHARMACON_L-004259-00_D 4139 MARK1
##   J107-2L H7   SIRNA DHARMACON_L-007730-00_C 3984 LIMK1
##   J107-2L I6   SIRNA DHARMACON_L-004259-00_C 4139 MARK1
scram.loc <- findWells(plates=sapply(mtor.loc, getBarcode),
                      contents="SCRAMBLED", well.names="H2")
## there are 8 wells remaining:
##   J101-2C H2   CONTROL SCRAMBLED none ON-TARGETplus Non-targeting Pool
##   J104-2C H2   CONTROL SCRAMBLED none ON-TARGETplus Non-targeting Pool
##   J107-2C H2   CONTROL SCRAMBLED none ON-TARGETplus Non-targeting Pool
##   J110-2C H2   CONTROL SCRAMBLED none ON-TARGETplus Non-targeting Pool
##   J101-2L H2   CONTROL SCRAMBLED none ON-TARGETplus Non-targeting Pool
##   J104-2L H2   CONTROL SCRAMBLED none ON-TARGETplus Non-targeting Pool
```

```

##    J110-2L  H2    CONTROL  SCRAMBLED  none  ON-TARGETplus  Non-targeting  Pool
##    J107-2L  H2    CONTROL  SCRAMBLED  none  ON-TARGETplus  Non-targeting  Pool

data <- getSingleCellData(c(mtor.loc, other.loc, scram.loc))
## for plate J101-2C all requested data was loaded from cached well files.
## for plate J104-2C all requested data was loaded from cached well files.
## for plate J107-2C all requested data was loaded from cached well files.
## for plate J110-2C all requested data was loaded from cached well files.
## for plate J101-2L all requested data was loaded from cached well files.
## for plate J104-2L all requested data was loaded from cached well files.
## for plate J110-2L all requested data was loaded from cached well files.
## for plate J107-2L all requested data was loaded from cached well files.

h6 <- lapply(data, function(x) {
  return(list(meta=x$H6$meta, data=meltData(cleanData(x$H6, "lower"))))
})
## well H6 (J101-2L):
##   keeping 1 images (3) despite count.cells not in [45, 146] but 159.
## well H6 (J110-2L):
##   keeping 1 images (4) despite count.cells not in [40, 154] but 162.
## well H6 (J107-2L):
##   keeping 3 images (2, 7, 9) despite count.cells not in [56, 163] but 168, 169, 166.
h7 <- lapply(data, function(x) {
  return(list(meta=x$H7$meta, data=meltData(cleanData(x$H7, "lower"))))
})
## well H7 (J110-2L):
##   discarding 1 images (5) because count.cells not in [40, 154] but 37.
## well H7 (J107-2L):
##   discarding 1 images (1) because count.cells not in [56, 163] but 42.
i6 <- lapply(data, function(x) {
  return(list(meta=x$I6$meta, data=meltData(cleanData(x$I6, "lower"))))
})
## well I6 (J104-2L):
##   keeping 1 images (2) despite count.cells not in [43.75, 130] but 149.
## well I6 (J107-2L):
##   keeping 1 images (2) despite count.cells not in [56, 163] but 165.
##   discarding 1 images (7) because count.cells not in [56, 163] but 53.
h2 <- lapply(data, function(x) {
  return(list(meta=x$H2$meta, data=meltData(cleanData(x$H2, "lower"))))
})
## well H2 (J101-2C):
##   discarding 2 images (8, 9) because count.cells not in [54, 433] but 7, 4.
## well H2 (J110-2L):
##   discarding 1 images (3) because count.cells not in [40, 154] but 37.
rm(data)

```

First, wells containing siRNA for the gene *MTOR* are searched for within the kinome-wide Dharmacon unpooled screens (replicate 1) for brucella and salmonella. Then on the plates containing those wells, scrambled control experiments are looked up (in a well located close to the *MTOR*). Additionally, two further groups of wells in close vicinity of the *MTOR* well are looked up: one in the same row, but next column and one in the same column but one row down. The data for the resulting 32 wells is loaded, cleaned up and melted into data frames.

```

dat1.bruc <- suppressMessages(makeRankFull(prepareDataforGlm(
  h6[["J101-2C"]][extract_itex]data[/extract_itex]mat[extract_itex]Cells, h7[["J101-2C"]][extract_itex]data[/extract_itex]mat[extract_itex]Cells)
))
## Warning in prepareDataforGlm(h6[["J101-2C"]][extract_itex]data[/extract_itex]mat[extract_itex]Cells,
## h7[["J101-2C"]][extract_itex]data[/extract_itex]mat[extract_itex]Cells): removed 25 variables containing Na/NaN.
## Warning in makeRankFull(prepareDataforGlm(h6[["J101-2C"]][extract_itex]data[/extract_itex]mat[extract_itex]Cells, :
## removed 46 zero variance variables.
## Warning in makeRankFull(prepareDataforGlm(h6[["J101-2C"]][extract_itex]data[/extract_itex]mat[extract_itex]Cells, :
## removed 9 variables due to highly correlation (>0.9999)
dat1.salm <- suppressMessages(makeRankFull(prepareDataforGlm(
  h6[["J101-2L"]][extract_itex]data[/extract_itex]mat[extract_itex]Cells, h7[["J101-2L"]][extract_itex]data[/extract_itex]mat[extract_itex]Cells)
))
## Warning in prepareDataforGlm(h6[["J101-2L"]][extract_itex]data[/extract_itex]mat[extract_itex]Cells,
## h7[["J101-2L"]][extract_itex]data[/extract_itex]mat[extract_itex]Cells): removed 5 variables containing Na/NaN.
## Warning in makeRankFull(prepareDataforGlm(h6[["J101-2L"]][extract_itex]data[/extract_itex]mat[extract_itex]Cells, :
## removed 40 zero variance variables.
## Warning in makeRankFull(prepareDataforGlm(h6[["J101-2L"]][extract_itex]data[/extract_itex]mat[extract_itex]Cells, :
## removed 14 variables due to highly correlation (>0.9999)
glm111 <- glmRegular(dat1.bruc)
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
glm112 <- glmGlmnet(dat1.bruc)
glm113 <- glmBayesglm(dat1.bruc)
## Warning: fitted probabilities numerically 0 or 1 occurred
glm121 <- glmRegular(dat1.salm)
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
glm122 <- glmGlmnet(dat1.salm)
glm123 <- glmBayesglm(dat1.salm)
## Warning: fitted probabilities numerically 0 or 1 occurred

```

As previously, *MTOR* wells were always compared to scrambled wells, this time the *MTOR* well H6 is compared to a neighboring well H7 for both a brucella plate (J101-2C) and a salmonella plate (J101-2L). The resulting prediction accuracies and Matthews correlation coefficients are:

- regular, brucella: 0.97 and 0.92
- glmnet, brucella: 1 and 1
- bayesglm, brucella: 1 and 0.99
- regular, salmonella: 0.85 and 0.7
- glmnet, salmonella: 0.89 and 0.78
- bayesglm, salmonella: 0.91 and 0.82

Both the convergence issues and perfect separation of previous experiments comparing *MTOR* against scrambled wells remain.

```

dat2.bruc <- suppressMessages(makeRankFull(prepareDataforGlm(
  do.call(rbind, lapply(h6, function(x) {
    if(getPathogen(x[/extract_itex]meta) == "Brucella") return(x[/extract_itex]data[/extract_itex]mat[extract_itex]Cells) else return(NULL)
  })),
  do.call(rbind, lapply(h7, function(x) {
    if(getPathogen(x[/extract_itex]meta) == "Brucella") return(x[/extract_itex]data[/extract_itex]mat[extract_itex]Cells) else return(NULL)
  })))

```

```

))
## Warning in prepareDataforGlm(do.call(rbind, lapply(h6, function(x) {:
## removed 25 variables containing Na/NaN.
## Warning in makeRankFull(prepareDataforGlm(do.call(rbind, lapply(h6,
## function(x) {: removed 42 zero variance variables.
## Warning in makeRankFull(prepareDataforGlm(do.call(rbind, lapply(h6,
## function(x) {: removed 12 variables due to highly correlation (>0.9999)
dat2.salm <- suppressMessages(makeRankFull(prepareDataforGlm(
  do.call(rbind, lapply(h6, function(x) {
    if(getPathogen(x$meta) == "Salmonella") return(x$data$mat$Cells)
    else return(NULL)
  })),
  do.call(rbind, lapply(h7, function(x) {
    if(getPathogen(x$meta) == "Salmonella") return(x$data$mat$Cells)
    else return(NULL)
  })))
))
## Warning in prepareDataforGlm(do.call(rbind, lapply(h6, function(x) {:
## removed 5 variables containing Na/NaN.
## Warning in makeRankFull(prepareDataforGlm(do.call(rbind, lapply(h6,
## function(x) {: removed 40 zero variance variables.
## Warning in makeRankFull(prepareDataforGlm(do.call(rbind, lapply(h6,
## function(x) {: removed 14 variables due to highly correlation (>0.9999)

glm211 <- glmRegular(dat2.bruc)
glm212 <- glmGlmnet(dat2.bruc)
glm213 <- glmBayesglm(dat2.bruc)
glm221 <- glmRegular(dat2.salm)
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
glm222 <- glmGlmnet(dat2.salm)
glm223 <- glmBayesglm(dat2.salm)

```

In this iteration, the same wells are compared, but instead of only using data from single wells, all available wells are combined (4 each). The resulting prediction accuracies and Matthews correlation coefficients are:

- regular, brucella: 0.79 and 0.56
- glmnet, brucella: 0.79 and 0.56
- bayesglm, brucella: 0.79 and 0.56
- regular, salmonella: 0.81 and 0.57
- glmnet, salmonella: 0.83 and 0.6
- bayesglm, salmonella: 0.82 and 0.58

Both the convergence issues and perfect separation of previous experiments comparing *MTOR* against scrambled wells remain.