Bootstrap and fit mutants RLS with binomial aging model

h qin

May 4 2017 - 20 June, 2017

```
rm(list=ls())
host = "Applejack" #"Ridgeside"
if (host == "AppleJack") {
 setwd("/Users/hqin/github/bmc_netwk_aging_manuscript/R1/1.mutants")
}
if (host == "Ridgeside") {
 setwd("/hong/hqin/github/bmc_netwk_aging_manuscript/R1/1.mutants")
}

library('flexsurv')
```

```
## Loading required package: survival
```

```
library('stringr')
source("../lifespan.r")
```

parse the strains from files

```
set.seed(20170701) #for repeatability

RUNS = 100; #bootstrap runs

mydir = 'rls.qin'   # Qin lab rls
files = list.files(path=paste(mydir,"/", sep=''), pattern="csv")

genotypes = c();
media = c();
for( i in 1:length(files)) {
 elements = unlist(str_split(files[i], "_"))
 genotypes = c(genotypes, elements[1])
 media = c(media, elements[length(elements)])
}
genotypes = unique(genotypes)
genotypes
```

```
## [1] "4741.csv"       "4742.csv"       "4743.csv"       "sir2.csv"
## [5] "sir2SIR2OE.csv"
```

```
#media
```

Now, fit all RLS data

```
for( BootstrapCount in 1:RUNS ) {#!!!!!!

report = data.frame(files)
report$samplesize = NA; report$R=NA; report$t0=NA; report$n=NA; report$G=NA; #report$longfilename=NA;

for( i in 1:length(report[,1])){
#for( i in 3:4){
  tb = read.csv( paste(mydir,"/",files[i] ,sep=''), sep="\t")
  #tb = read.table( paste("../qinlab_rls/",my.files[1],sep=''), sep="\t")
  report$samplesize[i] = length(tb[,1])

  #boostrap
  tb[,1] = sample(tb[,1], replace=TRUE); # BOOTSTRAP HERE

  GompFlex = flexsurvreg(formula = Surv(tb[,1]) ~ 1, dist = 'gompertz')
  WeibFlex = flexsurvreg(formula = Surv(tb[,1]) ~ 1, dist = 'weibull')

  report$avgLS[i] =  mean(tb[,1])
  report$stdLS[i] =  sd(tb[,1])
  report$CV[i] = report$stdLS[i] / report$avgLS[i]

  report$GompGFlex[i] = GompFlex$res[1,1]
  report$GompRFlex[i] = GompFlex$res[2,1]
  report$GompLogLikFlex[i] = round(GompFlex$loglik, 1)
  report$GompAICFlex[i] = round(GompFlex$AIC)

  report$WeibShapeFlex[i] = WeibFlex$res[1,1]
  report$WeibRateFlex[i] = WeibFlex$res[2,1]
  report$WeibLogLikFlex[i] = round(WeibFlex$loglik, 1)
  report$WeibAICFlex[i] = round(WeibFlex$AIC)

  #set initial values
  Rhat = report$GompRFlex[i]; # 'i' was missing. a bug costed HQ a whole afternoon.
  Ghat = report$GompGFlex[i];
  nhat = 6;
  t0= (nhat-1)/Ghat;
  fitBinom = optim ( c(Rhat, t0, nhat),  llh.binomialMortality.single.run,
                     lifespan=tb[,1],
                     #method='SANN') #SANN needs control
                     method="L-BFGS-B",
                     lower=c(1E-10, 1, 1), upper=c(1,200,20) );
  report[i, c("R", "t0", "n")] = fitBinom$par[1:3]
  report$G[i] = (report$n[i] - 1)/report$t0[i]
}

report$BestModel = ifelse(report$GompAICFlex < report$WeibAICFlex, "Gomp", "Weib")
report$BestModel = ifelse(abs(report$GompAICFlex - report$WeibAICFlex)<2, "<2", report$BestModel)

outname = paste("bootstrap/", BootstrapCount, ".csv", sep='');  #has to be changed here !!!
write.csv(report, file = outname, row.names = FALSE);
}#end of boostrap loop
```

#summarize boostrap results. Pick row-col value from every file into a buffer, then mean and stddev.

```
BootstrapMean = report;
BootstrapStd = report;
rownames = names(report);
for( col in 2:length(report[1,])) {
  for ( row in 1:length(report[,1])) {
    buffer = c();
    for( BootstrapCount in 1:RUNS ) {
      filename = paste("bootstrap/", BootstrapCount, ".csv", sep='');
      tb = read.csv(filename)
      if( rownames[col] == "BestModel" ) {
        buffer = as.character( c(buffer, as.character(tb[row,col])) );
      } else {
        buffer = c(buffer, tb[row, col]);
      }
    }
    if( rownames[col] == "BestModel" ) {
      tmp = table( buffer );
      BootstrapMean[row,col] = paste(names(tmp), tmp, sep="=", collapse = ":");
    } else {
     BootstrapMean[row,col] = mean(buffer);
     BootstrapStd[row,col] = sqrt(var(buffer));
    }
  }
}
```

Means

```
BootstrapMean
```

```
##            files samplesize           R       t0        n          G
## 1       4741.csv         60 0.006323298 72.58452 7.738004 0.09715719
## 2       4742.csv         90 0.011874943 65.06404 7.058902 0.14585171
## 3       4743.csv         89 0.003207802 60.51096 7.737274 0.11261760
## 4       sir2.csv        117 0.006624094 22.91419 8.183248 0.32162797
## 5 sir2SIR2OE.csv         69 0.010577514 40.02936 7.694039 0.18216080
##      avgLS     stdLS        CV  GompGFlex   GompRFlex GompLogLikFlex
## 1 28.30517 11.665463 0.4131086 0.06877678 0.008073506       -234.543
## 2 20.79956  9.205718 0.4425275 0.06819314 0.016160087       -334.661
## 3 33.65719 10.577376 0.3143664 0.08395783 0.003565259       -343.124
## 4 12.73111  4.267768 0.3354932 0.20838131 0.010467182       -342.693
## 5 16.01725  6.655730 0.4157750 0.12157129 0.013860248       -231.107
##   GompAICFlex WeibShapeFlex WeibRateFlex WeibLogLikFlex WeibAICFlex
## 1      473.09      2.586579     31.71666       -231.944      467.91
## 2      673.29      2.398341     23.47735       -322.562      649.13
## 3      690.29      3.474995     37.39557       -335.498      675.00
## 4      689.37      3.260032     14.18239       -335.266      674.56
## 5      466.20      2.624786     18.08153       -225.537      455.02
##                BestModel
## 1 <2=10:Gomp=16:Weib=74
## 2              Weib=100
## 3              Weib=100
## 4   <2=1:Gomp=3:Weib=96
## 5              Weib=100
```

StdDev

```
BootstrapStd
```

```
##            files samplesize            R        t0         n          G
## 1       4741.csv          0 0.0020718175 15.198077 0.7325374 0.02125096
## 2       4742.csv          0 0.0037234447 25.735774 1.7737868 0.14052447
## 3       4743.csv          0 0.0007331497  8.124890 0.5915531 0.01331728
## 4       sir2.csv          0 0.0017142799  3.835767 0.6798113 0.04952520
## 5 sir2SIR2OE.csv          0 0.0026113291  7.476001 0.8755283 0.08487013
##       avgLS     stdLS         CV   GompGFlex    GompRFlex GompLogLikFlex
## 1 1.7019361 1.1092254 0.04223864 0.011011771 0.0022415214       4.951344
## 2 1.0135667 0.8304394 0.03282293 0.008789604 0.0021187035       6.278340
## 3 1.0778634 0.8505697 0.02465002 0.011754763 0.0009828806       7.901312
## 4 0.4250728 0.2518860 0.02117658 0.018883120 0.0022112394       6.546126
## 5 0.8380106 0.4884831 0.02581652 0.010402926 0.0027254347       3.620153
##   GompAICFlex WeibShapeFlex WeibRateFlex WeibLogLikFlex WeibAICFlex
## 1    9.916923     0.2937836    1.9419193       5.470274   10.968361
## 2   12.502359     0.1597660    1.1598830       6.643578   13.324736
## 3   15.778322     0.2943262    1.1963116       6.918092   13.825756
## 4   13.045108     0.2159291    0.4605846       6.591305   13.231673
## 5    7.244643     0.1629840    0.9413241       4.724996    9.440917
##   BestModel
## 1      Weib
## 2      Weib
## 3      Weib
## 4      Weib
## 5      Weib
```

Merge the two tables

```
BootstrapMean$Rstd = BootstrapStd$R
BootstrapMean$t0std = BootstrapStd$t0
BootstrapMean$nstd = BootstrapStd$n
BootstrapMean$Gstd = BootstrapStd$G
BootstrapMean$avgLSstd = BootstrapStd$avgLS
BootstrapMean$GompRFlexStd = BootstrapStd$GompRFlex
BootstrapMean$GompGFlexStd = BootstrapStd$GompGFlex
names(BootstrapMean)
```

```
##  [1] "files"          "samplesize"     "R"              "t0"
##  [5] "n"              "G"              "avgLS"          "stdLS"
##  [9] "CV"             "GompGFlex"      "GompRFlex"      "GompLogLikFlex"
## [13] "GompAICFlex"    "WeibShapeFlex"  "WeibRateFlex"   "WeibLogLikFlex"
## [17] "WeibAICFlex"    "BestModel"      "Rstd"           "t0std"
## [21] "nstd"           "Gstd"           "avgLSstd"       "GompRFlexStd"
## [25] "GompGFlexStd"
```

Reorganized the columns

```
BootstrapMean = BootstrapMean[, c( "files",  "samplesize",  "R", "Rstd",  "t0",  "t0std",  "n",  "nstd"
                                   "Gstd", "avgLS",  "avgLSstd", "stdLS",   "BestModel",  "CV",
                                    "GompRFlex", "GompRFlexStd", "GompGFlex",  "GompGFlexStd",
                                   "GompLogLikFlex",  "GompAICFlex",  "WeibShapeFlex",  "WeibRateFlex",
                                   "WeibLogLikFlex",  "WeibAICFlex"  )];
BootstrapMean
```

```
##             files samplesize           R         Rstd       t0     t0std
## 1       4741.csv         60 0.006323298 0.0020718175 72.58452 15.198077
## 2       4742.csv         90 0.011874943 0.0037234447 65.06404 25.735774
## 3       4743.csv         89 0.003207802 0.0007331497 60.51096  8.124890
## 4       sir2.csv        117 0.006624094 0.0017142799 22.91419  3.835767
## 5 sir2SIR2OE.csv         69 0.010577514 0.0026113291 40.02936  7.476001
##          n      nstd          G       Gstd    avgLS  avgLSstd     stdLS
## 1 7.738004 0.7325374 0.09715719 0.02125096 28.30517 1.7019361 11.665463
## 2 7.058902 1.7737868 0.14585171 0.14052447 20.79956 1.0135667  9.205718
## 3 7.737274 0.5915531 0.11261760 0.01331728 33.65719 1.0778634 10.577376
## 4 8.183248 0.6798113 0.32162797 0.04952520 12.73111 0.4250728  4.267768
## 5 7.694039 0.8755283 0.18216080 0.08487013 16.01725 0.8380106  6.655730
##               BestModel        CV   GompRFlex GompRFlexStd  GompGFlex
## 1 <2=10:Gomp=16:Weib=74 0.4131086 0.008073506 0.0022415214 0.06877678
## 2              Weib=100 0.4425275 0.016160087 0.0021187035 0.06819314
## 3              Weib=100 0.3143664 0.003565259 0.0009828806 0.08395783
## 4   <2=1:Gomp=3:Weib=96 0.3354932 0.010467182 0.0022112394 0.20838131
## 5              Weib=100 0.4157750 0.013860248 0.0027254347 0.12157129
##   GompGFlexStd GompLogLikFlex GompAICFlex WeibShapeFlex WeibRateFlex
## 1  0.011011771       -234.543      473.09      2.586579     31.71666
## 2  0.008789604       -334.661      673.29      2.398341     23.47735
## 3  0.011754763       -343.124      690.29      3.474995     37.39557
## 4  0.018883120       -342.693      689.37      3.260032     14.18239
## 5  0.010402926       -231.107      466.20      2.624786     18.08153
##   WeibLogLikFlex WeibAICFlex
## 1       -231.944      467.91
## 2       -322.562      649.13
## 3       -335.498      675.00
## 4       -335.266      674.56
## 5       -225.537      455.02
```

Merge mean and std for publication

```
BootstrapMeanPublishing = data.frame( BootstrapMean[, c("files")] )
BootstrapMeanPublishing$RwithStd = as.vector( paste(round(BootstrapMean$R, 4), round(BootstrapMean$Rstd
BootstrapMeanPublishing$t0withStd = as.vector( paste(round(BootstrapMean$t0, 1), round(BootstrapMean$t0s
BootstrapMeanPublishing$nwithStd = as.vector( paste(round(BootstrapMean$n, 1), round(BootstrapMean$nstd
BootstrapMeanPublishing$GwithStd = as.vector( paste(round(BootstrapMean$G, 2), round(BootstrapMean$Gstd
BootstrapMeanPublishing$GompRFlexwithStd = as.vector( paste(round(BootstrapMean$GompRFlex, 3), round(Boo
BootstrapMeanPublishing$GompGFlexwithStd = as.vector( paste(round(BootstrapMean$GompGFlex, 2), round(Boo

BootstrapMeanPublishing$avgLSwithStd = as.vector( paste(round(BootstrapMean$avgLS, 2), round(BootstrapMe

BootstrapMeanPublishing
```

```
##   BootstrapMean...c..files...          RwithStd     t0withStd
## 1                    4741.csv 0.0063 +/- 0.0021 72.6 +/- 15.2
## 2                    4742.csv 0.0119 +/- 0.0037 65.1 +/- 25.7
## 3                    4743.csv  0.0032 +/- 7e-04  60.5 +/- 8.1
## 4                    sir2.csv 0.0066 +/- 0.0017  22.9 +/- 3.8
## 5              sir2SIR2OE.csv 0.0106 +/- 0.0026    40 +/- 7.5
##        nwithStd       GwithStd GompRFlexwithStd GompGFlexwithStd
## 1 7.7 +/- 0.733  0.1 +/- 0.021  0.008 +/- 0.002   0.07 +/- 0.011
## 2 7.1 +/- 1.774 0.15 +/- 0.141  0.016 +/- 0.002   0.07 +/- 0.009
## 3 7.7 +/- 0.592 0.11 +/- 0.013  0.004 +/- 0.001   0.08 +/- 0.012
## 4  8.2 +/- 0.68  0.32 +/- 0.05   0.01 +/- 0.002   0.21 +/- 0.019
## 5 7.7 +/- 0.876 0.18 +/- 0.085  0.014 +/- 0.003    0.12 +/- 0.01
##      avgLSwithStd
## 1 28.31 +/- 1.702
## 2  20.8 +/- 1.014
## 3 33.66 +/- 1.078
## 4 12.73 +/- 0.425
## 5 16.02 +/- 0.838
```

output

```
write.csv(BootstrapMean, file=paste('sandbox/', mydir, '_Bootstrap_summary.csv', sep=''), row.names = F
write.csv(BootstrapMeanPublishing, file=paste('sandbox/', mydir, '_Bootstrap_summary_Publications.csv',
```

How can t0 and R both increases in pooled data set? What does this mean for p, and lambda?

Do pooled data contain heterogenous noises?