Fit Qin-lab natural isolates using bootstraps

h qin

2017 April 24 - June 11

```
rm(list=ls())
host = "Applejack" #"Ridgeside"
if (host == "AppleJack") {
 setwd("/Users/hqin/github/bmc_netwk_aging_manuscript/R1/0.nat.rls.fitting");
}
if (host == "Ridgeside") {
 setwd("/home/hqin/github/bmc_netwk_aging_manuscript/R1/0.nat.rls.fitting");
}

library('flexsurv')
```

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

```
source("../lifespan.r")
```

Set seed for consistency for publication purpose.

```
set.seed(20170101)  #remove to test for different bootstraps
```

parse the strains from files

```
RUNS = 100; #bootstrap runs
files = list.files(path="../qinlab_rls/", pattern="rls.tab")
tmp1 = gsub("\\d{6}.", "", files)
redundant_strains = gsub(".rls.tab", "", tmp1)
strains = sort( unique( redundant_strains ))
strains
```

```
##  [1] "101S"           "BY4716"         "BY4741"         "BY4742"
##  [5] "BY4743"         "JSBY4741"       "M1-2"           "M13"
##  [9] "M14"            "M2-8"           "M22"            "M32"
## [13] "M34"            "M5"             "M8"             "RM112N"
## [17] "S288c"          "SGU57"          "sir2D.4741a"    "sir2D.4742"
## [21] "sir2DSIR2.4742" "SK1"            "W303"           "YPS128"
## [25] "YPS163"
```

Take files from natural isolates

```
my.strains=c("101S", "M1-2","M13","M14","M2-8","M22","M32","M34","M5","M8","RM112N","S288c","SGU57", "Y
files2=c();
for( i in 1:length(my.strains)){
 files2 = c( files2, files[grep(my.strains[i], files)]);
}
```

```
report = data.frame(cbind(my.strains))
report$samplesize = NA; report$R=NA; report$t0=NA; report$n=NA; report$G=NA; report$longfilename=NA;

files = files2;
strains = my.strains;
```

Now, fit all RLS data sets by strains

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

for( i in 1:length(report[,1])){
#for( i in 3:4){
  my.files = files[grep(strains[i], files)]
  report$longfilename[i] = paste(my.files, collapse = "::");
  tb = read.table( paste("../qinlab_rls/",my.files[1],sep=''), sep="\t")
  if( length(my.files)> 1){
    for( fi in 2:length(my.files)) {
      tmp.tb = read.table( paste("../qinlab_rls/",my.files[fi],sep=''), sep="\t")
      tb = rbind( tb, tmp.tb)
    }
  }
  report$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, 4), 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]
}#i loop

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='');
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
```

```
##    my.strains samplesize           R       t0        n          G
## 1        101S         85 0.002505451 35.99604 6.686567 0.16043672
## 2        M1-2         54 0.003389155 40.47858 7.077706 0.15126428
## 3         M13         70 0.003418251 40.80201 7.364152 0.15665075
## 4         M14         60 0.003484255 55.06556 6.558044 0.10172215
## 5        M2-8        105 0.003409035 42.35209 7.998702 0.16660562
## 6         M22         60 0.003311774 46.31594 6.873676 0.12749105
## 7         M32         60 0.002698475 34.32367 7.039921 0.17693307
```

```
## 8         M34         58 0.002762630 31.44145 6.677334 0.18155728
## 9          M5        166 0.003434030 73.99028 7.761917 0.09198913
## 10         M8         60 0.001828786 30.65496 6.077315 0.16688372
## 11     RM112N         59 0.002527789 55.44863 6.202258 0.09464606
## 12      S288c         41 0.005077696 56.77646 7.885613 0.12478848
## 13      SGU57         58 0.006519542 58.03551 7.879940 0.12195836
## 14     YPS128         69 0.002613473 41.90105 6.475758 0.13165058
## 15     YPS163        130 0.002296450 37.25939 6.404451 0.14547179
##    longfilename    avgLS     stdLS        CV  GompGFlex    GompRFlex
## 1            15 31.45824  7.414184 0.2360298 0.14183796 0.0011886014
## 2             9 27.89889  9.088393 0.3271839 0.12535089 0.0025636829
## 3             7 26.59871  9.150567 0.3452559 0.12325312 0.0030381156
## 4            10 36.32417 12.787126 0.3531547 0.09198143 0.0021270344
## 5             4 24.76743  8.056250 0.3255815 0.11812658 0.0043352445
## 6            11 32.07050 10.140110 0.3171061 0.11150913 0.0020152844
## 7             6 28.14783  6.664404 0.2368370 0.14566819 0.0015725113
## 8             1 27.02103  8.062000 0.2993340 0.16048962 0.0013442040
## 9             5 36.85289 12.990472 0.3527247 0.06803908 0.0040096710
## 10            8 34.78883  6.781575 0.1951294 0.16453160 0.0003590377
## 11           12 44.13085 12.975729 0.2948886 0.09121338 0.0010544096
## 12           14 26.36390 10.201881 0.3879796 0.09101058 0.0062074368
## 13           13 23.89845 10.516315 0.4414499 0.09131664 0.0077070530
## 14            2 35.08014  9.586650 0.2738812 0.12013655 0.0011318327
## 15            3 34.41446  8.563412 0.2490014 0.13442802 0.0008118923
##    GompLogLikFlex GompAICFlex WeibShapeFlex WeibRateFlex WeibLogLikFlex
## 1         -292.604      589.20      4.941140     34.24623       -289.332
## 2         -192.909      389.76      3.635913     30.95623       -194.592
## 3         -250.000      504.00      3.162652     29.31791       -258.419
## 4         -233.435      470.91      3.369193     40.53819       -236.454
## 5         -372.001      748.01      3.417174     27.54976       -367.165
## 6         -221.673      447.31      3.713709     35.57107       -222.881
## 7         -203.847      411.68      4.597658     30.76086       -199.428
## 8         -196.896      397.81      4.124638     29.73866       -202.132
## 9         -669.603     1343.21      3.120367     41.22948       -658.682
## 10        -200.250      404.45      6.028250     37.50598       -198.654
## 11        -231.399      466.79      4.104189     48.67951       -233.248
## 12        -153.402      310.80      2.856687     29.48898       -152.826
## 13        -215.925      435.81      2.487367     26.87244       -217.746
## 14        -251.437      506.89      4.383940     38.57445       -251.798
## 15        -459.429      922.83      4.836988     37.60504       -459.809
##    WeibAICFlex             BestModel
## 1       582.72 <2=10:Gomp=16:Weib=74
## 2       393.19  <2=25:Gomp=69:Weib=6
## 3       520.85   <2=5:Gomp=88:Weib=7
## 4       476.90   <2=7:Gomp=91:Weib=2
## 5       738.35   <2=5:Gomp=2:Weib=93
## 6       449.76 <2=32:Gomp=51:Weib=17
## 7       402.89          <2=1:Weib=99
## 8       408.27          <2=7:Gomp=93
## 9      1321.34              Weib=100
## 10      401.29  <2=22:Gomp=6:Weib=72
## 11      470.48  <2=23:Gomp=70:Weib=7
## 12      309.61 <2=31:Gomp=26:Weib=43
## 13      439.51 <2=15:Gomp=73:Weib=12
```

```
## 14      507.62 <2=38:Gomp=40:Weib=22
## 15      923.64 <2=21:Gomp=44:Weib=35
```

StdDev

```
BootstrapStd
```

```
##    my.strains samplesize            R        t0         n           G
## 1        101S          0 0.0009287405  5.479713 0.6974831 0.022866994
## 2        M1-2          0 0.0009980579  5.276955 0.7602881 0.017968586
## 3         M13          0 0.0009459944  4.195255 0.6396622 0.015279534
## 4         M14          0 0.0010014401  6.604037 0.6372519 0.012057134
## 5        M2-8          0 0.0006344560  3.989513 0.1091360 0.014718183
## 6         M22          0 0.0010404733 11.837460 1.5075316 0.015948835
## 7         M32          0 0.0011194964  2.586144 0.7143603 0.024292330
## 8         M34          0 0.0007362679  3.544114 0.6159757 0.018066724
## 9          M5          0 0.0005916447  7.219866 0.3813192 0.007859491
## 10         M8          0 0.0003484946  2.919832 0.2071864 0.014669925
## 11     RM112N          0 0.0006463488  5.986260 0.3746111 0.009729632
## 12      S288c          0 0.0016711154 12.922973 1.3414738 0.023097209
## 13      SGU57          0 0.0021172202 19.651135 1.5046710 0.017847788
## 14     YPS128          0 0.0007754037  3.888046 0.4863120 0.015269929
## 15     YPS163          0 0.0004804929  2.624209 0.4280389 0.011935765
##    longfilename     avgLS     stdLS         CV  GompGFlex    GompRFlex
## 1             0 0.8145652 0.6935133 0.02504680 0.02380020 0.0006263407
## 2             0 1.2904696 0.8190490 0.03951375 0.01707491 0.0011734232
## 3             0 1.0640702 0.8726972 0.04234235 0.01207602 0.0010961058
## 4             0 1.6213699 0.8832416 0.03409446 0.01068784 0.0009291640
## 5             0 0.7334054 0.5917975 0.02604108 0.01095486 0.0010385633
## 6             0 1.3592553 0.7678484 0.03111031 0.01055153 0.0007328389
## 7             0 0.9733216 0.6534801 0.02281649 0.01109063 0.0004621282
## 8             0 0.9969735 0.8140843 0.03715807 0.01798446 0.0006980044
## 9             0 0.9422285 0.5958022 0.01861178 0.00679012 0.0008521899
## 10            0 0.9690498 0.6401819 0.01971980 0.01594485 0.0001833472
## 11            0 1.7506204 1.1804215 0.03295971 0.01003443 0.0004660643
## 12            0 1.5006540 1.1151878 0.04538956 0.01579110 0.0020188426
## 13            0 1.3190327 0.8063968 0.04257855 0.01148507 0.0023363275
## 14            0 1.1245440 0.8631905 0.02934451 0.01138383 0.0004510274
## 15            0 0.6989248 0.4736352 0.01567590 0.01000082 0.0002297927
##    GompLogLikFlex GompAICFlex WeibShapeFlex WeibRateFlex WeibLogLikFlex
## 1       10.040795   20.103269     0.6118343    0.7991631       8.453998
## 2        5.052674   10.111619     0.5445300    1.2478190       5.264550
## 3        5.107392   10.197048     0.5175619    1.1412121       9.055014
## 4        4.422243    8.859276     0.4008239    1.6226350       4.283122
## 5        6.815764   13.635585     0.2868792    0.7848509       7.468214
## 6        4.054075    8.143529     0.4353666    1.3545984       4.936643
## 7        4.313953    8.673517     0.3661096    1.0628450       5.162590
## 8        5.446413   10.926471     0.6325299    0.9340973       7.057482
## 9        8.619410   17.228878     0.1810164    1.0062292       7.133282
## 10       5.314465   10.603387     0.5951988    0.9761665       5.354530
## 11       5.138201   10.270380     0.5215999    1.7111870       5.692623
## 12       4.135825    8.298716     0.4090500    1.6389505       4.664818
## 13       3.544191    7.107678     0.2945314    1.4349958       4.403819
## 14       5.665947   11.381267     0.5023200    1.0705033       6.280722
## 15       7.528428   15.066891     0.3347178    0.6817798       7.470658
```

```
##    WeibAICFlex BestModel
## 1    16.946189      Weib
## 2    10.509923      Gomp
## 3    18.131604        <2
## 4     8.552275      Gomp
## 5    14.919565      Weib
## 6     9.867936      Weib
## 7    10.321004      Weib
## 8    14.105558      Gomp
## 9    14.319740      Weib
## 10   10.710417        <2
## 11   11.353351      Gomp
## 12    9.314803      Weib
## 13    8.840329      Gomp
## 14   12.550327      Gomp
## 15   14.965631        <2
```

Merge the two tables

```
BootstrapMean$Rstd = BootstrapStd$R
BootstrapMean$t0std = BootstrapStd$t0
BootstrapMean$nstd = BootstrapStd$n
BootstrapMean$Gstd = BootstrapStd$G
names(BootstrapMean)
```

```
##  [1] "my.strains"     "samplesize"     "R"              "t0"
##  [5] "n"              "G"              "longfilename"   "avgLS"
##  [9] "stdLS"          "CV"             "GompGFlex"      "GompRFlex"
## [13] "GompLogLikFlex" "GompAICFlex"    "WeibShapeFlex"  "WeibRateFlex"
## [17] "WeibLogLikFlex" "WeibAICFlex"    "BestModel"      "Rstd"
## [21] "t0std"          "nstd"           "Gstd"
```

Reorganized the columns

```
BootstrapMean = BootstrapMean[, c( "my.strains",  "samplesize",  "R", "Rstd",  "t0",  "t0std",  "n",  "
                                   "Gstd", "avgLS",  "stdLS",   "BestModel",
                                   "longfilename",   "CV",  "GompGFlex",  "GompRFlex",  "GompLogLikFlex
 "GompAICFlex",  "WeibShapeFlex",  "WeibRateFlex",  "WeibLogLikFlex",  "WeibAICFlex"  )];
BootstrapMean
```

```
##    my.strains samplesize           R         Rstd       t0     t0std
## 1        101S         85 0.002505451 0.0009287405 35.99604  5.479713
## 2        M1-2         54 0.003389155 0.0009980579 40.47858  5.276955
## 3         M13         70 0.003418251 0.0009459944 40.80201  4.195255
## 4         M14         60 0.003484255 0.0010014401 55.06556  6.604037
## 5        M2-8        105 0.003409035 0.0006344560 42.35209  3.989513
## 6         M22         60 0.003311774 0.0010404733 46.31594 11.837460
## 7         M32         60 0.002698475 0.0011194964 34.32367  2.586144
## 8         M34         58 0.002762630 0.0007362679 31.44145  3.544114
## 9          M5        166 0.003434030 0.0005916447 73.99028  7.219866
## 10         M8         60 0.001828786 0.0003484946 30.65496  2.919832
## 11     RM112N         59 0.002527789 0.0006463488 55.44863  5.986260
## 12      S288c         41 0.005077696 0.0016711154 56.77646 12.922973
## 13      SGU57         58 0.006519542 0.0021172202 58.03551 19.651135
## 14     YPS128         69 0.002613473 0.0007754037 41.90105  3.888046
## 15     YPS163        130 0.002296450 0.0004804929 37.25939  2.624209
```

```
##           n      nstd          G        Gstd    avgLS     stdLS
## 1  6.686567 0.6974831 0.16043672 0.022866994 31.45824  7.414184
## 2  7.077706 0.7602881 0.15126428 0.017968586 27.89889  9.088393
## 3  7.364152 0.6396622 0.15665075 0.015279534 26.59871  9.150567
## 4  6.558044 0.6372519 0.10172215 0.012057134 36.32417 12.787126
## 5  7.998702 0.1091360 0.16660562 0.014718183 24.76743  8.056250
## 6  6.873676 1.5075316 0.12749105 0.015948835 32.07050 10.140110
## 7  7.039921 0.7143603 0.17693307 0.024292330 28.14783  6.664404
## 8  6.677334 0.6159757 0.18155728 0.018066724 27.02103  8.062000
## 9  7.761917 0.3813192 0.09198913 0.007859491 36.85289 12.990472
## 10 6.077315 0.2071864 0.16688372 0.014669925 34.78883  6.781575
## 11 6.202258 0.3746111 0.09464606 0.009729632 44.13085 12.975729
## 12 7.885613 1.3414738 0.12478848 0.023097209 26.36390 10.201881
## 13 7.879940 1.5046710 0.12195836 0.017847788 23.89845 10.516315
## 14 6.475758 0.4863120 0.13165058 0.015269929 35.08014  9.586650
## 15 6.404451 0.4280389 0.14547179 0.011935765 34.41446  8.563412
##                BestModel longfilename        CV  GompGFlex    GompRFlex
## 1  <2=10:Gomp=16:Weib=74           15 0.2360298 0.14183796 0.0011886014
## 2   <2=25:Gomp=69:Weib=6            9 0.3271839 0.12535089 0.0025636829
## 3    <2=5:Gomp=88:Weib=7            7 0.3452559 0.12325312 0.0030381156
## 4    <2=7:Gomp=91:Weib=2           10 0.3531547 0.09198143 0.0021270344
## 5    <2=5:Gomp=2:Weib=93            4 0.3255815 0.11812658 0.0043352445
## 6  <2=32:Gomp=51:Weib=17           11 0.3171061 0.11150913 0.0020152844
## 7           <2=1:Weib=99            6 0.2368370 0.14566819 0.0015725113
## 8           <2=7:Gomp=93            1 0.2993340 0.16048962 0.0013442040
## 9              Weib=100            5 0.3527247 0.06803908 0.0040096710
## 10  <2=22:Gomp=6:Weib=72            8 0.1951294 0.16453160 0.0003590377
## 11  <2=23:Gomp=70:Weib=7           12 0.2948886 0.09121338 0.0010544096
## 12 <2=31:Gomp=26:Weib=43           14 0.3879796 0.09101058 0.0062074368
## 13 <2=15:Gomp=73:Weib=12           13 0.4414499 0.09131664 0.0077070530
## 14 <2=38:Gomp=40:Weib=22            2 0.2738812 0.12013655 0.0011318327
## 15 <2=21:Gomp=44:Weib=35            3 0.2490014 0.13442802 0.0008118923
##    GompLogLikFlex GompAICFlex WeibShapeFlex WeibRateFlex WeibLogLikFlex
## 1         -292.604      589.20      4.941140     34.24623       -289.332
## 2         -192.909      389.76      3.635913     30.95623       -194.592
## 3         -250.000      504.00      3.162652     29.31791       -258.419
## 4         -233.435      470.91      3.369193     40.53819       -236.454
## 5         -372.001      748.01      3.417174     27.54976       -367.165
## 6         -221.673      447.31      3.713709     35.57107       -222.881
## 7         -203.847      411.68      4.597658     30.76086       -199.428
## 8         -196.896      397.81      4.124638     29.73866       -202.132
## 9         -669.603     1343.21      3.120367     41.22948       -658.682
## 10        -200.250      404.45      6.028250     37.50598       -198.654
## 11        -231.399      466.79      4.104189     48.67951       -233.248
## 12        -153.402      310.80      2.856687     29.48898       -152.826
## 13        -215.925      435.81      2.487367     26.87244       -217.746
## 14        -251.437      506.89      4.383940     38.57445       -251.798
## 15        -459.429      922.83      4.836988     37.60504       -459.809
##    WeibAICFlex
## 1       582.72
## 2       393.19
## 3       520.85
## 4       476.90
## 5       738.35
```

```
## 6       449.76
## 7       402.89
## 8       408.27
## 9      1321.34
## 10      401.29
## 11      470.48
## 12      309.61
## 13      439.51
## 14      507.62
## 15      923.64
```

Merge mean and std for publication

```
BootstrapMeanPublishing = data.frame( BootstrapMean[, c("my.strains")] )
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
```

```
##    BootstrapMean...c..my.strains...          RwithStd     t0withStd
## 1                              101S  0.0025 +/- 9e-04    36 +/- 5.5
## 2                             M1-2  0.0034 +/- 0.001  40.5 +/- 5.3
## 3                              M13  0.0034 +/- 9e-04  40.8 +/- 4.2
## 4                              M14  0.0035 +/- 0.001  55.1 +/- 6.6
## 5                             M2-8  0.0034 +/- 6e-04    42.4 +/- 4
## 6                              M22  0.0033 +/- 0.001 46.3 +/- 11.8
## 7                              M32 0.0027 +/- 0.0011  34.3 +/- 2.6
## 8                              M34  0.0028 +/- 7e-04  31.4 +/- 3.5
## 9                               M5  0.0034 +/- 6e-04    74 +/- 7.2
## 10                              M8  0.0018 +/- 3e-04  30.7 +/- 2.9
## 11                          RM112N  0.0025 +/- 6e-04    55.4 +/- 6
## 12                           S288c 0.0051 +/- 0.0017 56.8 +/- 12.9
## 13                           SGU57 0.0065 +/- 0.0021   58 +/- 19.7
## 14                          YPS128  0.0026 +/- 8e-04  41.9 +/- 3.9
## 15                          YPS163  0.0023 +/- 5e-04  37.3 +/- 2.6
##         nwithStd       GwithStd
## 1  6.7 +/- 0.697 0.16 +/- 0.023
## 2   7.1 +/- 0.76 0.15 +/- 0.018
## 3   7.4 +/- 0.64 0.16 +/- 0.015
## 4  6.6 +/- 0.637  0.1 +/- 0.012
## 5    8 +/- 0.109 0.17 +/- 0.015
## 6  6.9 +/- 1.508 0.13 +/- 0.016
## 7    7 +/- 0.714 0.18 +/- 0.024
## 8  6.7 +/- 0.616 0.18 +/- 0.018
## 9  7.8 +/- 0.381 0.09 +/- 0.008
## 10 6.1 +/- 0.207 0.17 +/- 0.015
## 11 6.2 +/- 0.375  0.09 +/- 0.01
## 12 7.9 +/- 1.341 0.12 +/- 0.023
## 13 7.9 +/- 1.505 0.12 +/- 0.018
## 14 6.5 +/- 0.486 0.13 +/- 0.015
## 15 6.4 +/- 0.428 0.15 +/- 0.012
```

output

```
write.csv(BootstrapMean, file="sandbox/Bootstrap_summary.csv", row.names = FALSE)
write.csv(BootstrapMeanPublishing, file="sandbox/Bootstrap_summary_for_publication.csv", row.names = FA
```

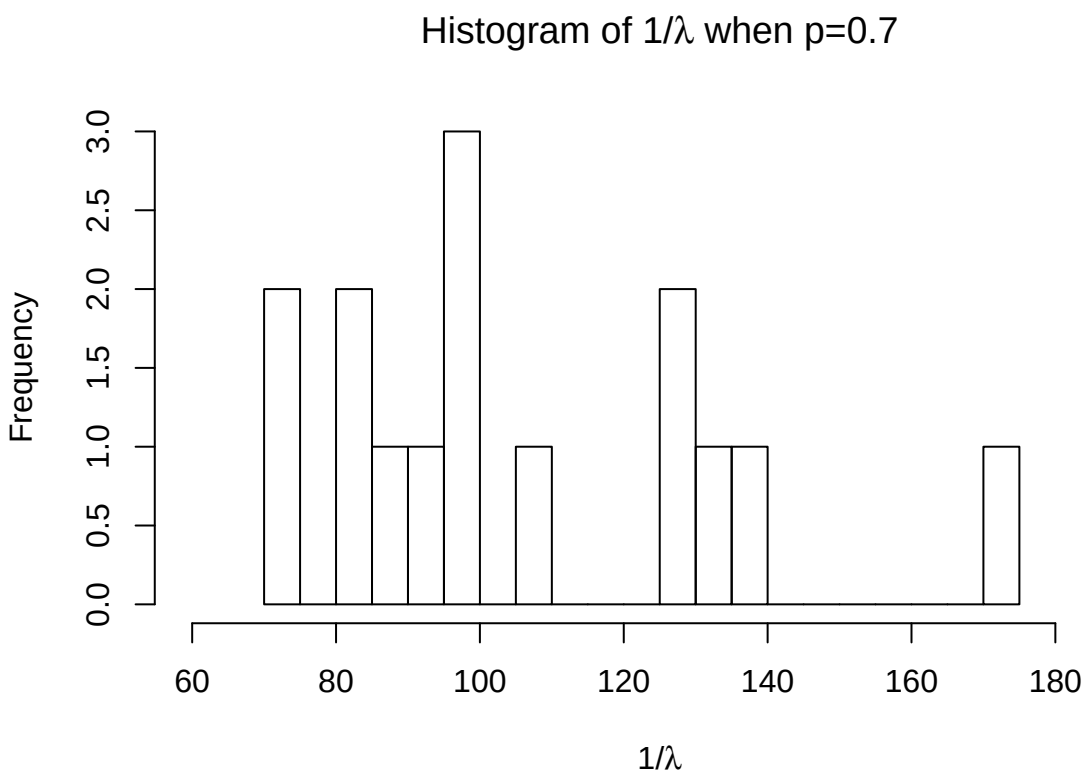
Calculate lambda based on t0 = (1-p)/(p Lambda) So, 1/lambda = t0 * p / (1-p)

```
p = 0.7
BootstrapMean$One.over.lambdaP07 = BootstrapMean$t0 * p/ (1-p)
BootstrapMean[, c("t0", "One.over.lambdaP07")]
```

```
##          t0 One.over.lambdaP07
## 1  35.99604           83.99076
## 2  40.47858           94.45003
## 3  40.80201           95.20470
## 4  55.06556          128.48632
## 5  42.35209           98.82154
## 6  46.31594          108.07053
## 7  34.32367           80.08855
## 8  31.44145           73.36337
## 9  73.99028          172.64399
## 10 30.65496           71.52823
## 11 55.44863          129.38014
## 12 56.77646          132.47841
## 13 58.03551          135.41619
## 14 41.90105           97.76911
## 15 37.25939           86.93857
```

Histogram of 1/lambda for p=0.7

```
hist(BootstrapMean$One.over.lambdaP07, br=20, xlim=c(60,190), xlab=expression(paste("1/",lambda) ), mai
```

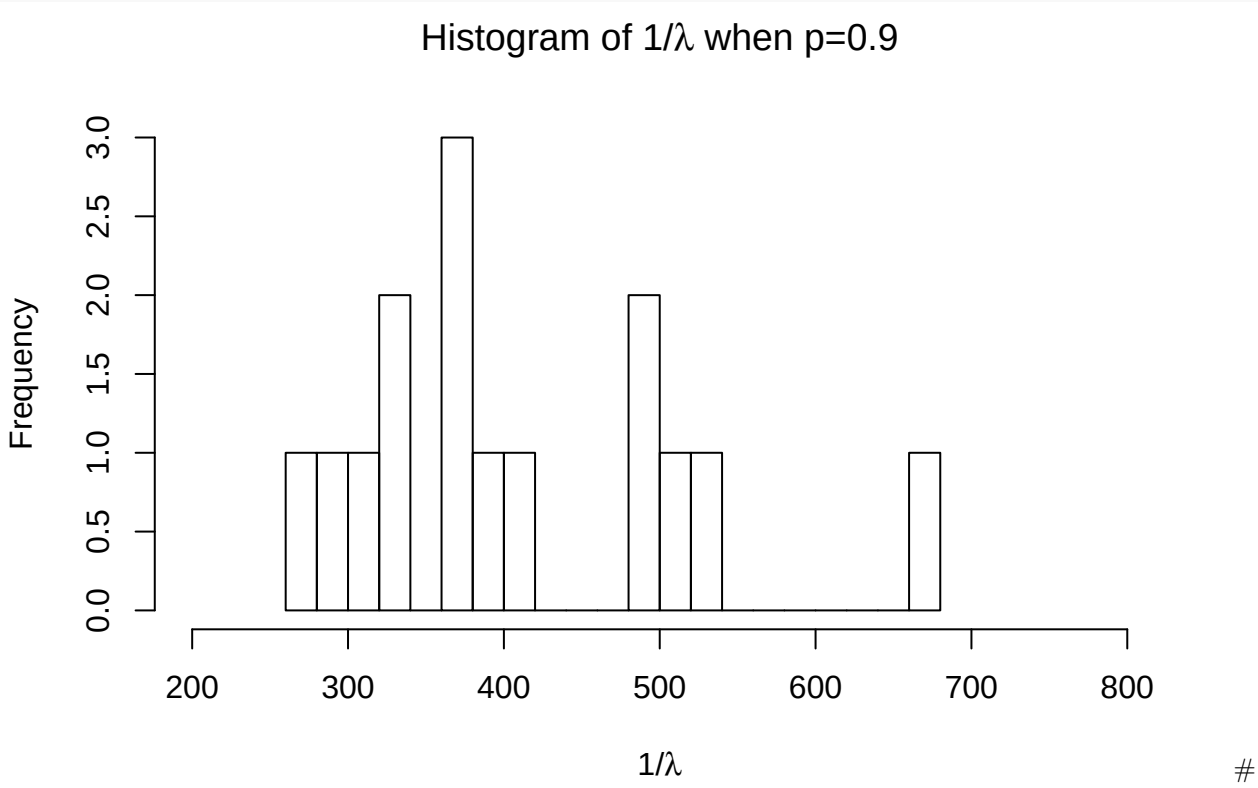


```
p = 0.9
BootstrapMean$One.over.lambdaP09 = BootstrapMean$t0 * p/ (1-p)
BootstrapMean[, c("t0", "One.over.lambdaP09")]
```

```
##          t0 One.over.lambdaP09
## 1  35.99604           323.9643
## 2  40.47858           364.3073
## 3  40.80201           367.2181
## 4  55.06556           495.5901
## 5  42.35209           381.1688
## 6  46.31594           416.8435
## 7  34.32367           308.9130
## 8  31.44145           282.9730
## 9  73.99028           665.9125
## 10 30.65496           275.8946
## 11 55.44863           499.0377
## 12 56.77646           510.9881
## 13 58.03551           522.3196
## 14 41.90105           377.1094
## 15 37.25939           335.3345
```

Histogram 1/lambda, p=0.9

```
hist(BootstrapMean$One.over.lambdaP09, br=20, xlim=c(200,800), xlab=expression(paste("1/",lambda) ), ma
```



#

Strehler-Mildvan correlation in using bootstrap results

```
# BootstrapMean
summary(lm(log10(BootstrapMean$GompRFlex) ~ BootstrapMean$GompGFlex))
```

```
##
## Call:
```

```
## lm(formula = log10(BootstrapMean$GompRFlex) ~ BootstrapMean$GompGFlex)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49942 -0.20689 -0.02195  0.20883  0.36533
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -1.7110     0.3204  -5.340 0.000134 ***
## BootstrapMean$GompGFlex  -8.4039     2.6337  -3.191 0.007091 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2766 on 13 degrees of freedom
## Multiple R-squared:  0.4392, Adjusted R-squared:  0.3961
## F-statistic: 10.18 on 1 and 13 DF,  p-value: 0.007091
```

```
summary(lm(log10(BootstrapMean$R) ~ BootstrapMean$GompGFlex))
```

```
##
## Call:
## lm(formula = log10(BootstrapMean$R) ~ BootstrapMean$GompGFlex)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.177513 -0.079147  0.002811  0.054411  0.234284
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -2.1366     0.1251 -17.080 2.75e-10 ***
## BootstrapMean$GompGFlex  -3.1043     1.0283  -3.019  0.00987 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.108 on 13 degrees of freedom
## Multiple R-squared:  0.4121, Adjusted R-squared:  0.3669
## F-statistic: 9.114 on 1 and 13 DF,  p-value: 0.009871
```

```
summary(lm(log10(BootstrapMean$R) ~ BootstrapMean$G)) #G from t0 and n
```

```
##
## Call:
## lm(formula = log10(BootstrapMean$R) ~ BootstrapMean$G)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.188886 -0.077273 -0.003401  0.059639  0.289279
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -2.2745     0.1718 -13.238 6.38e-09 ***
## BootstrapMean$G  -1.6447     1.2030  -1.367    0.195
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.1317 on 13 degrees of freedom
## Multiple R-squared:  0.1257, Adjusted R-squared:  0.05846
## F-statistic: 1.869 on 1 and 13 DF,  p-value: 0.1947
```

```
summary(lm(log10(BootstrapMean$GompGFlex) ~ BootstrapMean$t0)) #G from t0 and n
```

```
##
## Call:
## lm(formula = log10(BootstrapMean$GompGFlex) ~ BootstrapMean$t0)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.016469 -0.012257 -0.005318  0.008555  0.024096
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -0.5370301  0.0150244  -35.74 2.29e-14 ***
## BootstrapMean$t0 -0.0088343  0.0003205  -27.56 6.43e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01454 on 13 degrees of freedom
## Multiple R-squared:  0.9832, Adjusted R-squared:  0.9819
## F-statistic: 759.7 on 1 and 13 DF,  p-value: 6.43e-13
```

```
summary(lm(log10(BootstrapMean$R) ~ BootstrapMean$n)) #G from t0 and n
```

```
##
## Call:
## lm(formula = log10(BootstrapMean$R) ~ BootstrapMean$n)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.132065 -0.057526  0.000259  0.043924  0.169621
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -3.68903    0.24459 -15.083 1.29e-09 ***
## BootstrapMean$n  0.16924    0.03482   4.861 0.000311 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0839 on 13 degrees of freedom
## Multiple R-squared:  0.6451, Adjusted R-squared:  0.6178
## F-statistic: 23.63 on 1 and 13 DF,  p-value: 0.0003109
```

Does t0 mediate GFlex ~ log(RFlex)? Yes.

```
summary(lm( BootstrapMean$GompGFlex ~ BootstrapMean$t0 ))
```

```
##
## Call:
## lm(formula = BootstrapMean$GompGFlex ~ BootstrapMean$t0)
##
```

```
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.007271 -0.004972 -0.002378  0.002244  0.013517
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.2202732  0.0075774   29.07 3.25e-13 ***
## BootstrapMean$t0 -0.0022402  0.0001616  -13.86 3.65e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.007334 on 13 degrees of freedom
## Multiple R-squared:  0.9366, Adjusted R-squared:  0.9317
## F-statistic: 192.1 on 1 and 13 DF,  p-value: 3.65e-09
```

```
summary(lm(log10(BootstrapMean$GompRFlex) ~ BootstrapMean$t0 ))
```

```
##
## Call:
## lm(formula = log10(BootstrapMean$GompRFlex) ~ BootstrapMean$t0)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46908 -0.19237 -0.00486  0.23982  0.39995
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -3.533595   0.299440 -11.801 2.55e-08 ***
## BootstrapMean$t0  0.018197   0.006388   2.849   0.0137 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2898 on 13 degrees of freedom
## Multiple R-squared:  0.3843, Adjusted R-squared:  0.337
## F-statistic: 8.115 on 1 and 13 DF,  p-value: 0.01369
```

```
summary(lm(log10(BootstrapMean$GompRFlex) ~ BootstrapMean$GompGFlex ))
```

```
##
## Call:
## lm(formula = log10(BootstrapMean$GompRFlex) ~ BootstrapMean$GompGFlex)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49942 -0.20689 -0.02195  0.20883  0.36533
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -1.7110     0.3204  -5.340 0.000134 ***
## BootstrapMean$GompGFlex  -8.4039     2.6337  -3.191 0.007091 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2766 on 13 degrees of freedom
## Multiple R-squared:  0.4392, Adjusted R-squared:  0.3961
```

```
## F-statistic: 10.18 on 1 and 13 DF,  p-value: 0.007091
```

```
summary(lm(log10(BootstrapMean$GompGFlex) ~ BootstrapMean$t0 + log10(BootstrapMean$GompRFlex)))
```

```
##
## Call:
## lm(formula = log10(BootstrapMean$GompGFlex) ~ BootstrapMean$t0 +
##     log10(BootstrapMean$GompRFlex))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.017722 -0.010619 -0.005119  0.010194  0.021283
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -0.5734526  0.0523756 -10.949 1.33e-07 ***
## BootstrapMean$t0               -0.0086468  0.0004161 -20.781 8.91e-11 ***
## log10(BootstrapMean$GompRFlex) -0.0103075  0.0141753  -0.727    0.481
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01481 on 12 degrees of freedom
## Multiple R-squared:  0.9839, Adjusted R-squared:  0.9812
## F-statistic: 366.4 on 2 and 12 DF,  p-value: 1.75e-11
```

```
summary(lm(log10(BootstrapMean$GompRFlex) ~ BootstrapMean$t0 + log10(BootstrapMean$GompGFlex))) #not si
```

```
##
## Call:
## lm(formula = log10(BootstrapMean$GompRFlex) ~ BootstrapMean$t0 +
##     log10(BootstrapMean$GompGFlex))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.50584 -0.21093 -0.01843  0.20102  0.40655
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -5.73236    3.03919  -1.886   0.0837 .
## BootstrapMean$t0                -0.01797    0.05017  -0.358   0.7264
## log10(BootstrapMean$GompGFlex) -4.09431    5.63067  -0.727   0.4811
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2952 on 12 degrees of freedom
## Multiple R-squared:  0.4103, Adjusted R-squared:  0.312
## F-statistic: 4.175 on 2 and 12 DF,  p-value: 0.04205
```