

```

***** Simulation-based and covariate-adjusted sample size and power
calculations for two-sample RNA-seq data using generalized linear models
*****



## simulation R code

library(MASS)

##### Functions #####
##### **** A Function of estimating new size of alpha star for testing multiple
genes ****.
# T: Total number of genes in the analysis
# T1: Total true DEGs
# fdr: cotrolling false discovery rate

alphaStar<-function(power=0.8,fdr, T, T1) {
  (T1*power*fdr)/((T-T1)*(1-fdr))
}

##### **** Function of power and sample size calculation using glm model without
ajusted covariantes****

# rho: Fold change, a ratio of mean reads u1 in treatment group and u0 in
control group (u1/u0)
# mu0: mean read counts in control group
# k: ratio of sample size n1 in treatment and n0 in control groups (k=n1/n0),
k=1 for balanced design
# phi: dispersion parameter in a negtive bionomial disribution in GLM
# w: ratio of average size factor in two condition groups (s1/s0)
# alpha: significance size of alpha and 0.05 as default
# power: detection power and 0.80 as default
# flag: flag = 0 for testing a single gene and flag = 1 or else for testing
multiple genes

ss.fun1<-function(rho, mu0,k,phi,w,alpha=0.05,power=0.8,n,int=5000,tol=0.001,
flag=0){

  if(flag==0){ # for a single gene
    alpha<-alpha # table 1 for a single gene
  }
  else{
    alpha<-alphaStar(power=0.8,fdr=0.05, T=10000, T1=200) # table 4 for
multiple genes
  }

  count<-0
  a<-0.1
  while (a>tol){
    sh<-1/phi;
    u1<-rho*w*mu0;
    if(k==1){
      n1=n2=n
    }

```

```

t0<-rep(0,n)
t1<-rep(1,n)
ta1<-c(t0,t1)

for (i in 1:int){
  set.seed(i)
  pbo<-rnbinom(n,size=sh, mu=mu0) # control data
  d1<-rnbinom(n, size=sh, mu=u1)   # treatment data
  r1<-c(pbo,d1)

  x1<-cbind(ta1,r1) # regression matrix
  dt1<-as.data.frame(x1) # data for GLM

  fm1 <- glm.nb(r1 ~ ta1, data = dt1) # GLM with NB distribution
  pval<-summary(fm1)$coef[2,4] # extract p-value

  if(pval<=alpha/2){ count<-count+1} # two side test pval < 0.025 ad
default
}

p<-count/int # empiracal power estimated

if(p>0.7999){
  a<-0.001} # stop
else{
  a=0.1
  count=0
  n=n+1} # repeat while loop
}
return(c(n, p))
}

## The sample size (N) and power estimation for Testing a single gene given
the following parameters.
# The results are listed in Table 1 when FC = 2.

rho<-2 # FC =2
k=1      # balanced design
w=1      # equal read depth
i=0
phi<-c(0.1, 0.2, 0.5) # dispersion
t.mu0<-c(5, 10)      # mean read counts of the gene in control group
l0<-length(phi)
l1<-length(t.mu0)
res1<-matrix(0, nrow=l0*l1, ncol=2)
row.names(res1)<-rep(c(5,10),l0)
for (j in 1:l0){

  for(l in 1:l1){
    i<-i+1
    res1[i,]<-ss.fun1(rho=rho, k=k, w=w, mu0=t.mu0[l], phi=phi[j], n=6, flag=0)
  }
}

```

```

res1 # results for estimated sample size and actual power are listed in Table
1 when FC=2.

# mu0      N      power
# [,1]    [,2]
# 5       10     0.8320
# 10      7      0.8406
# 5       14     0.8274
# 10      11     0.8306
# 5       25     0.8034
# 10      23     0.8088
write.csv(res1,"rho.2_ss_power_table1_singleGene.csv")

## The sample size (N) and power estimation for testing multiple gene given
the parameters as follow.
## The results are listed in Table 4 when FC = 2.

rho<-2 # FC =2
k=1      # balanced design
w=1      # equal read depth
i=0
phi<-c(0.1, 0.2, 0.5) # dispersion
t.mu0<-c(5, 10)      # mean read counts of the gene in control group
l0<-length(phi)
l1<-length(t.u0)
res2<-matrix(0, nrow=l0*l1, ncol=2)
row.names(res2)<-rep(c(5,10),l0)
for (j in 1:l0){

  for(l in 1:l1){
    i<-i+1
    res2[i,]<-ss.fun1(rho=rho, k=k, w=w, mu0=t.mu0[l], phi=phi[j], n=6, flag=1)
  }
}
res2 #results for estimated sample size and actual power and listed in Table
4
# mu0      N      power
# [,1]    [,2]
# 5       20     0.8310
# 10      14     0.8294
# 5       28     0.8140
# 10      21     0.8000
# 5       51     0.8062
# 10      46     0.8128

write.csv(res2,"rho.2_ss_power_table3_FDR.csv")

```