

TCA_genetic_correlation_check

September 2, 2019

```
In [1]: library(glmnet)
        library(matrixStats)
        library(abind)
        library(pracma)
        library(matrixcalc)
        library(TCA)
        library(MCMCpack)
        library(ggplot2)
```

```
Loading required package: Matrix
Loading required package: foreach
Loaded glmnet 2.0-18
```

```
Attaching package: pracma
```

```
The following objects are masked from package:Matrix:
```

```
    expm, lu, tril, triu
```

```
Loading required package: coda
```

```
Loading required package: MASS
```

```
##
```

```
## Markov Chain Monte Carlo Package (MCMCpack)
```

```
## Copyright (C) 2003-2019 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park
```

```
##
```

```
## Support provided by the U.S. National Science Foundation
```

```
## (Grants SES-0350646 and SES-0350613)
```

```
##
```

```
Attaching package: MCMCpack
```

```
The following object is masked from package:matrixcalc:
```

```
    vech
```

```
The following object is masked from package:pracma:
```

procrustes

```
In [246]: summary_statistics <- function(mdl,train_X,test_X,train_c1,test_c1,
                                         train_G,test_G,train_Z,test_Z,beta,
                                         gamma_c1,gamma_c2,model=1){
  # TCA estimation & parameters
  Z_hat_tca = tensor(train_G,mdl,log_file=NULL,verbose=FALSE,debug=FALSE)
  rmse_tca = t(do.call(rbind,lapply(1:length(Z_hat_tca),
                                     function(x) sqrt(sum((Z_hat_tca[[x]]-train_Z[,x])^2))))
  Z_hat_cor_tca = sapply(1:length(Z_hat_tca), function(x) cor(unlist(Z_hat_tca[x]),train_Z[,x]))
  cell_type = ncol(mdl$W)
  if(model==1){
    tca_beta = t(matrix(mdl$gammas_hat[,grepl("_SNP",colnames(mdl$gammas_hat))],ncol=length(Z_hat_tca)))
    beta_hat_cor_tca = lapply(1:cell_type,function(x)
                              cor(beta[x,],mdl$gammas_hat[,grepl(paste("Cell_type",x,".G",sep=""),colnames(mdl$gammas_hat))]))
  }else{ # model 2 does not predict beta for snps
    tca_beta = NULL
    beta_hat_cor_tca = NULL
  }
  gamma_hat_cor_tca = cor(array(gamma_c1,
                                dim=c(length(gamma_c1),length(Z_hat_tca),length(Z_hat_tca))),
                           mdl$gammas_hat[,grepl('male|smok|age|nerd|self', colnames(mdl$gammas_hat))])
  if(model==1){
    pred = cbind(train_X,train_c1)
    test_pred = cbind(test_X,test_c1)
  }else{ # model 2 does not use train_X/test_X as C1
    pred = train_c1
    test_pred = test_c1
  }
  # TCA parameter direct estimation <----> C1 & X
  Z_hat_train_second_eq_tca = sapply(1:cell_type,function(x) pred %%% mdl$gammas_hat[[x]])
  Z_hat_train_second_eq_cor_tca = diag(cor(train_Z,Z_hat_train_second_eq_tca))
  Z_hat_test_second_eq_tca = sapply(1:cell_type,function(x) test_pred %%% mdl$gammas_hat[[x]])
  Z_hat_test_second_eq_cor_tca = diag(cor(test_Z,Z_hat_test_second_eq_tca))

  # Lasso bulk data regression <----> X only
  glmnet.mdl.X.cv <- cv.glmnet(x=train_X,y=t(train_G),nfolds=5)
  glmnet.mdl.X <- glmnet(x=train_X,y=t(train_G),lambda=glmnet.mdl.X.cv$lambda.min)
  beta_full_X_bulk <- as.numeric(glmnet.mdl.X$beta)
  # extract non zero predictors and recompute
  predictors.X <- colnames(train_X)[which(beta_full_X_bulk!=0)]
  beta_X <- as.matrix(c(glmnet.mdl.X$a0,as.matrix(glmnet.mdl.X$beta[predictors.X,])))
  bias_one <- numeric(nrow(train_X))+1
  G_hat_train_lasso <- cbind(bias_one,train_X[,predictors.X]) %%% beta_X
  G_hat_train_cor_bulk <- cor(t(train_G),G_hat_train_lasso)
  G_hat_test_lasso <- cbind(numeric(nrow(test_X))+1,test_X[,predictors.X]) %%% beta_X
```

```

G_hat_test_cor_bulk <- cor(t(test_G),G_hat_test_lasso)

# cell type specific lasso
corrs = numeric(ncol(mdl$W))
corrs.real = numeric(ncol(mdl$W))
corrs.beta = numeric(ncol(mdl$W))
corrs.test.real = numeric(ncol(mdl$W))
dev_ratio = numeric(ncol(mdl$W))
rmse_lasso = numeric(ncol(mdl$W))
beta_full_cell = matrix(0,nrow=ncol(mdl$W),ncol=dim(train_X)[2])
Z_hat_test_lasso = matrix(0,nrow=nrow(test_Z),ncol=ncol(test_Z))
for (h in 1:ncol(mdl$W)){

  glmnet.mdl.cv <- cv.glmnet(x=train_X,y=Z_hat_tca[[h]],standardize=FALSE,alpha=
  rmse_lasso[h] = sqrt(glmnet.mdl.cv$cvm[glmnet.mdl.cv$lambda == glmnet.mdl.cv$lambda.min])
  glmnet.mdl <- glmnet(x=train_X,y=Z_hat_tca[[h]],standardize=FALSE,alpha=1,lambda=glmnet.mdl.cv$lambda.min)
  dev_ratio[h] <- glmnet.mdl$dev.ratio
  beta.full <- as.numeric(glmnet.mdl$beta)
  beta_full_cell[h,] <- beta.full
  predictors <- colnames(train_X)[which(beta.full != 0)]
  beta_lasso <- as.matrix(c(glmnet.mdl$a0,as.matrix(glmnet.mdl$beta[predictors,])))
  Z_hat_train_lasso <- cbind(numeric(nrow(train_X))+1,train_X[,predictors]) %*%
  Z_hat_test_lasso[,h] <- cbind(numeric(nrow(test_X))+1,test_X[,predictors]) %*%
  Z_hat_test_lasso[,h] = signif(Z_hat_test_lasso[,h], digits = 5)
  if(sum(beta_lasso)==0 | sd(Z_hat_train_lasso)==0){
    # model 2 lasso forces SNPs effect to become zero.
    # If there were no predictors, then cor would be NaN. Which also means
    # no snps is correlated with Z. cor just set to 0.
    corrs[h] = 0
    corrs.real[h] = 0
    corrs.beta[h] = 0
    corrs.test.real[h] = 0
  }else{
    corrs[h] <- cor(t(Z_hat_tca[[h]]),Z_hat_train_lasso)
    corrs.real[h] <- cor(train_Z[,h],Z_hat_train_lasso)
    corrs.beta[h] <- cor(beta[h,],beta.full)
    corrs.test.real[h] <- cor(test_Z[,h],Z_hat_test_lasso[,h])
  }
}

# Precision & Recall for lasso
Binary_True=(beta!=0)
Binary_Pred_Lasso=(beta_full_cell!=0)
TP = do.call(rbind,lapply(1:cell_type,function(x) sum(as.integer(Binary_True[x,])&
FP = do.call(rbind,lapply(1:cell_type,function(x) sum(as.integer(!Binary_True[x,])
FN = do.call(rbind,lapply(1:cell_type,function(x) sum(as.integer(Binary_True[x,])&
precision_lasso = TP/(TP+FP)
recall_lasso = TP/(TP+FN)

```

```

return(list('Z_hat_test_lasso'=Z_hat_test_lasso, 'Z_hat_tca'=Z_hat_tca, 'Z_hat_cor_t
          'Z_hat_train_second_eq_cor_tca'=Z_hat_train_second_eq_cor_tca, 'Z_hat_te
          'G_hat_train_cor_bulk'=G_hat_train_cor_bulk, 'G_hat_test_cor_bulk'=G_hat
          'cor_lasso_tca'=corrs, 'cor_lasso_real_train'=corrs.real, 'cor_lasso_real
          'cor_beta_lasso_real'=corrs.beta, 'beta_hat_lasso'=beta_full_cell, 'dev_r
          'precision_lasso'=precision_lasso, 'recall_lasso'=recall_lasso, 'rmse_las
}

```

```

In [247]: generate_params <- function(cell_her=TRUE, gene_cor=FALSE, bulk_her=FALSE, seed=1, N=5000,
      set.seed(1)
      if(length(herr_arr)!=0){
        M = length(herr_arr)
      }else{
        M = M
      }
      K = 4
      N = N
      pc_num = 2
      # cis_snps_nums = floor(runif(M, min = 150, max = 400))
      D = D
      cis_snps_nums = array(D, M)
      sigma_g = 0.01
      sigma_z = 0.1

      if(gene_cor){ # varying genetic correlation across genes
        corr_seq = seq(from=0, to=pslab, length.out=M)
      }else{
        corr_seq = seq(from=0, to=0, length.out=M)
      }
      pslab = matrix(pslab, nrow=M, ncol=K)
      corr_matrix = lapply(1:M, function(x) matrix(corr_seq[x], nrow=K, ncol=K))
      for(i in 1:M){
        diag(corr_matrix[[i]])=1
      }

      if(cell_her){ # varying heritability across genes
        heritability_cell_specific = do.call(rbind, lapply(1:K, function(x) herr_arr))
      }else{
        heritability_cell_specific = matrix(her, nrow=K, ncol=M)
      }

      if(bulk_her){ # varying bulk heritability across genes
        heritability_bulk = seq(from=0.1, to=her_bulk, length.out=M)
      }else{
        heritability_bulk = array(her_bulk, M)
      }

      MAF = lapply(1:length(cis_snps_nums), function(x) runif(cis_snps_nums[x], min=0.1, ma

```

```

id_name = sapply(1:N, function(x) paste('SAMPLE',toString(x),sep = ''))
cell_type_name=sapply(1:K, function(x) paste('Cell_type',x,sep = ''))

### generate C1 and C2
male = matrix(rbinom(N, 1, 0.5),nrow=N)
smoking = matrix(rbinom(N, 2, 0.2),nrow=N)
smoking = (smoking - min(smoking))/(max(smoking))
age = matrix(sapply(sapply(rnorm(N,50,20), function(x) floor(x)), function(x) if
age = (age-min(age))/(max(age)-min(age))
nerdiness = matrix(rbinom(N,2,0.2),nrow=N)
nerdiness = (nerdiness-min(nerdiness))/max(nerdiness)
selfishness = matrix(rbinom(N,2,0.2),nrow=N)
selfishness = (selfishness-min(selfishness))/max(selfishness)
c1 = scale(cbind(male, smoking, age,nerdiness,selfishness))
c2 = scale(matrix(rnorm(N*pc_num), nrow=N))

rownames(c1) = id_name
colnames(c1) = c('male','smoking','age','nerdiness','selfishness')
rownames(c2) = id_name
colnames(c2) = sapply(1:pc_num, function(x) paste('PC',toString(x),sep = ''))

p1 = dim(c1)[2]
sigma_gamma = sqrt((1-heritibility_cell_specific[1,]-sigma_z^2)/p1)
var_beta = heritibility_cell_specific[1,]*(p1*sigma_gamma^2+sigma_z^2)/(1-heritibi

# cell type proportion, from the real data estimated
W_alpha = c(26.553683792256,17.6621467979005,4.48671525658667,1.56874856517803)#,0
#W_alpha = 50.4500920752719
W_xsi = c(0.526335685426257,0.350091468050216,0.088933737720289,0.0310950585152043
if(K <= length(W_alpha)){
  W = rdirichlet(N, W_alpha[1:K])
}else{ #more cell type prop
  W = rdirichlet(N, runif(K, 0, 1))
}
colnames(W) = cell_type_name
rownames(W) = id_name

alpha = W_alpha
alpha_0 = sum(alpha)
alpha_tilde = alpha/alpha_0
m2_alpha = alpha_tilde %*% t(alpha_tilde)*alpha_0/(alpha_0+1)
diag(m2_alpha) = alpha_tilde*(1-alpha_tilde)/(alpha_0+1)+alpha_tilde^2

return(list(
  'M'=M, 'K'=K, 'N'=N, 'pc_num'=pc_num, 'D'=D, 'id_name'=id_name, 'cell_name'=cell_type_name,
  'her'=heritibility_cell_specific, 'her_bulk'=heritibility_bulk, 'pslab'=pslab,
  'MAF'=MAF, 'c1'=c1, 'c2'=c2, 'W'=W, 'sigma_gamma'=sigma_gamma, 'sigma_g'=sigma_g, 'm2_alpha'=m2_alpha,
  'sigma_z'=sigma_z, 'gene_cor'=corr_seq, 'beta_cor'=corr_matrix, 'var_beta'=var_beta

```

```
}
```

```
In [248]: one_gene <- function(par,g,seed=1){
  set.seed(seed)
  cell_type = par$K
  n_snps = par$D
  her = par$her[,g]
  cor = par$beta_cor[[g]]
  pslab = par$pslab[g,]
  var_beta = array(par$var_beta[g],par$K)

  maf = par$MAF[[g]]
  dummy_var = 10

  var_matrix=sqrt(var_beta%*%t(var_beta))
  Sigma_beta = cor*var_matrix/(1-pslab)^2
  diag(Sigma_beta) = diag(Sigma_beta) *(1-pslab)
  beta_ = mvrnorm(n_snps,mu=rep(0,par$K),Sigma=Sigma_beta,tol=1e-4,empirical=TRUE)
  rmask = do.call(rbind,lapply(1:cell_type,function(x) rbinom(n_snps,1,1-pslab[x])))
  beta = t(beta_)*rmask

  X = t(do.call(rbind,lapply(1:n_snps,function(x) rbinom(par$N,2,maf[x]))))
  colnames(X) = sapply(1:n_snps, function(x) paste('G',g,'_SNP',toString(x),sep = ''))
  rownames(X) = par$id_name
  X = scale(X)
  # enforce variance of samples by adjusting sd of beta
  xbeta_var = n_snps*var_beta[1]
  beta = do.call(rbind,
    lapply(1:cell_type,function(x) beta[x,]*(sqrt(xbeta_var)/sd(X %*%
    t(beta))))

  mu_z = X%*%t(beta)
  epsilon_z = array(rnorm(par$N*cell_type,mean=0,sd=par$sigma_z), c(par$N,cell_type))
  epsilon_z = apply(epsilon_z,2,function(x) x*par$sigma_z/sd(x))
  Sigma_gamma_c1 = matrix(0,nrow=cell_type,ncol=cell_type)
  diag(Sigma_gamma_c1) = par$sigma_gamma[g]^2
  gamma_c1 = mvrnorm(dim(par$c1)[2],mu=rep(0,cell_type),Sigma=Sigma_gamma_c1,tol=1e-4)
  # gamma_c1 = do.call(rbind,lapply(1:cell_type,function(x) rnorm(dim(par$c1)[2],mean=0,sd=par$sigma_gamma[g])))
  # enforce variance of samples by adjusting sd of gamma_c1
  # enforce the c1_gamma variance
  c1gamma_var = dim(par$c1)[2]*(par$sigma_gamma[g])^2
  gamma_c1 = do.call(rbind,
    lapply(1:cell_type,function(x) gamma_c1[,x]*(sqrt(c1gamma_var)/sd(gamma_c1))))
  c1_gamma = par$c1 %*% t(gamma_c1)

  Z = mu_z + epsilon_z + c1_gamma#
  rownames(Z) = par$id_name
  colnames(Z) = par$cell_name
```

```

### Generate Gene expressions
G = rowSums((par$W)*Z)
epsilon_G = rnorm(par$N,mean=0,sd=par$sigma_g)
epsilon_G = epsilon_G*par$sigma_g/sd(epsilon_G)

# use bulk level heritability to calculate sigma_gamma
bulk_her_nom = sum(hadamard.prod(par$m2_alpha,var(mu_z)))
bulk_her_c1_gamma = sum(hadamard.prod(par$m2_alpha,var(c1_gamma)))
bulk_her_epsilon_z = sum(hadamard.prod(par$m2_alpha,var(epsilon_z)))
bulk_her_z = sum(hadamard.prod(par$m2_alpha,var(Z)))

bulk_her_denom = sum(hadamard.prod(par$m2_alpha,var(Z)))+var(epsilon_G)
bulk_her_val = par$her_bulk[g]*her[1]
sd_gamma_c2 = sqrt((bulk_her_nom/bulk_her_val-bulk_her_denom)/par$pc_num)

gamma_c2 = matrix(rnorm(par$pc_num,mean=0,sd=sd_gamma_c2),nrow=par$pc_num,ncol=1)
c2gamma_var = par$pc_num * (sd_gamma_c2)^2
gamma_c2 = gamma_c2*(sqrt(c2gamma_var)/sd(par$c2%%gamma_c2))
c2_gamma = par$c2 %% gamma_c2

G = t(G + epsilon_G) # + c2_gamma
real_denom = var(t(G))
colnames(G) <- par$id_name
rownames(G) <- paste('gene',g,sep='_')
G = as.data.frame(G)

real_bulk_her = bulk_her_nom/(bulk_her_denom+var(c2_gamma))

return(list('X'=X,'beta'=beta,'gamma_c1'=gamma_c1,'bulk_her'=real_bulk_her,
           'gamma_c2'=gamma_c2,'Z'=Z,'G'=G,'epsilon_z'=epsilon_z))
}
#data = one_gene(par=params,1)

```

0.0.1 Plotting Begins for Genetic Correlation Check

```

In [249]: params = generate_params(cell_her=FALSE,gene_cor=TRUE,bulk_her=FALSE,
                                seed=1,N=5000,M=10,
                                D=250,pslab=0.5,her=0.95,
                                her_bulk=0.6)

data = lapply(1:params$M,function(x) one_gene(par=params,x))

```

```

In [250]: params$gene_cor = t(do.call(rbind,lapply(1:length(data),function(x) params$beta_cor[[x]]
temp = array(0,length(data))
for(i in 1:length(data)){
  cor_act = cor(data[[i]]$X %*% t(data[[i]]$beta))
  diag(cor_act) = params$gene_cor[i]
  temp[i] = signif(mean(cor_act),3)
  print(paste(temp[i],signif(params$gene_cor[i],3)))
}

```

```

    }
    params$gene_cor_real = temp

[1] "0.00328 0"
[1] "0.0759 0.0556"
[1] "0.119 0.111"
[1] "0.19 0.167"
[1] "0.228 0.222"
[1] "0.301 0.278"
[1] "0.34 0.333"
[1] "0.404 0.389"
[1] "0.452 0.444"
[1] "0.498 0.5"

```

```

In [267]: lapply(1:length(data),function(x) cor(data[[x]]$Z))
          lapply(1:length(data),function(x) cor(data[[x]]$X*%t(data[[x]]$beta)))
          # lapply(1:length(data),function(x) cor(params$c1*%t(data[[x]]$gamma_c1)))

```

		Cell_type1	Cell_type2	Cell_type3	Cell_type4
1. A matrix: 4 × 4 of type dbl	Cell_type1	1.0000000000	-0.0003317917	-0.00213270	0.01213545
	Cell_type2	-0.0003317917	1.0000000000	0.02956018	0.03453880
	Cell_type3	-0.0021327002	0.0295601768	1.00000000	-0.06838377
	Cell_type4	0.0121354480	0.0345387986	-0.06838377	1.00000000
		Cell_type1	Cell_type2	Cell_type3	Cell_type4
2. A matrix: 4 × 4 of type dbl	Cell_type1	1.00000000	0.04009888	0.04259027	0.08975178
	Cell_type2	0.04009888	1.00000000	0.02878084	0.14921212
	Cell_type3	0.04259027	0.02878084	1.00000000	0.13229550
	Cell_type4	0.08975178	0.14921212	0.13229550	1.00000000
		Cell_type1	Cell_type2	Cell_type3	Cell_type4
3. A matrix: 4 × 4 of type dbl	Cell_type1	1.00000000	0.08992166	0.08861244	0.1286508
	Cell_type2	0.08992166	1.00000000	0.10006931	0.1446577
	Cell_type3	0.08861244	0.10006931	1.00000000	0.1419878
	Cell_type4	0.12865083	0.14465772	0.14198778	1.0000000
		Cell_type1	Cell_type2	Cell_type3	Cell_type4
4. A matrix: 4 × 4 of type dbl	Cell_type1	1.0000000	0.1302058	0.1561839	0.2015018
	Cell_type2	0.1302058	1.0000000	0.1700036	0.2387678
	Cell_type3	0.1561839	0.1700036	1.0000000	0.2306276
	Cell_type4	0.2015018	0.2387678	0.2306276	1.0000000
		Cell_type1	Cell_type2	Cell_type3	Cell_type4
5. A matrix: 4 × 4 of type dbl	Cell_type1	1.0000000	0.1700356	0.2334695	0.2257214
	Cell_type2	0.1700356	1.0000000	0.2349993	0.2146365
	Cell_type3	0.2334695	0.2349993	1.0000000	0.2207478
	Cell_type4	0.2257214	0.2146365	0.2207478	1.0000000

6. A matrix: 4 × 4 of type dbl		Cell_type1	Cell_type2	Cell_type3	Cell_type4
	Cell_type1	1.0000000	0.1921008	0.2631345	0.3168799
	Cell_type2	0.1921008	1.0000000	0.2844620	0.3489239
	Cell_type3	0.2631345	0.2844620	1.0000000	0.3522017
	Cell_type4	0.3168799	0.3489239	0.3522017	1.0000000
7. A matrix: 4 × 4 of type dbl		Cell_type1	Cell_type2	Cell_type3	Cell_type4
	Cell_type1	1.0000000	0.2264679	0.2873016	0.3773279
	Cell_type2	0.2264679	1.0000000	0.2952850	0.3971018
	Cell_type3	0.2873016	0.2952850	1.0000000	0.3729353
	Cell_type4	0.3773279	0.3971018	0.3729353	1.0000000
8. A matrix: 4 × 4 of type dbl		Cell_type1	Cell_type2	Cell_type3	Cell_type4
	Cell_type1	1.0000000	0.2853901	0.3513460	0.4337017
	Cell_type2	0.2853901	1.0000000	0.3862238	0.4324173
	Cell_type3	0.3513460	0.3862238	1.0000000	0.4355035
	Cell_type4	0.4337017	0.4324173	0.4355035	1.0000000
9. A matrix: 4 × 4 of type dbl		Cell_type1	Cell_type2	Cell_type3	Cell_type4
	Cell_type1	1.0000000	0.3183749	0.3596084	0.5074710
	Cell_type2	0.3183749	1.0000000	0.4544970	0.4856864
	Cell_type3	0.3596084	0.4544970	1.0000000	0.4722043
	Cell_type4	0.5074710	0.4856864	0.4722043	1.0000000
10. A matrix: 4 × 4 of type dbl		Cell_type1	Cell_type2	Cell_type3	Cell_type4
	Cell_type1	1.0000000	0.3515752	0.4180559	0.5512974
	Cell_type2	0.3515752	1.0000000	0.4787724	0.5215446
	Cell_type3	0.4180559	0.4787724	1.0000000	0.5029956
	Cell_type4	0.5512974	0.5215446	0.5029956	1.0000000
1. A matrix: 4 × 4 of type dbl		1.000000000	0.002289842	-0.001864556	0.02211653
		0.002289842	1.000000000	0.039548699	0.03978192
		-0.001864556	0.039548699	1.000000000	-0.07563841
		0.022116530	0.039781920	-0.075638410	1.000000000
2. A matrix: 4 × 4 of type dbl		1.000000000	0.04113077	0.03897940	0.1021882
		0.04113077	1.000000000	0.02489338	0.1568722
		0.03897940	0.02489338	1.000000000	0.1322040
		0.10218820	0.15687217	0.13220403	1.0000000
3. A matrix: 4 × 4 of type dbl		1.000000000	0.1027956	0.09144727	0.1288701
		0.10279565	1.00000000	0.10778292	0.1477207
		0.09144727	0.1077829	1.000000000	0.1499510
		0.12887007	0.1477207	0.14995103	1.0000000
4. A matrix: 4 × 4 of type dbl		1.00000000	0.1426765	0.1626129	0.2087614
		0.1426765	1.00000000	0.1829341	0.2502022
		0.1626129	0.1829341	1.00000000	0.2408941
		0.2087614	0.2502022	0.2408941	1.0000000

5. A matrix: 4 CE 4 of type dbl	1.0000000	0.1738589	0.2494111	0.2449476
	0.1738589	1.0000000	0.2505405	0.2241523
	0.2494111	0.2505405	1.0000000	0.2368979
	0.2449476	0.2241523	0.2368979	1.0000000
6. A matrix: 4 CE 4 of type dbl	1.0000000	0.2078771	0.2763432	0.3295500
	0.2078771	1.0000000	0.3038295	0.3624918
	0.2763432	0.3038295	1.0000000	0.3697754
	0.3295500	0.3624918	0.3697754	1.0000000
7. A matrix: 4 CE 4 of type dbl	1.0000000	0.2402361	0.2977960	0.4027493
	0.2402361	1.0000000	0.3076504	0.4202813
	0.2977960	0.3076504	1.0000000	0.3850410
	0.4027493	0.4202813	0.3850410	1.0000000
8. A matrix: 4 CE 4 of type dbl	1.0000000	0.3049494	0.3721249	0.4518753
	0.3049494	1.0000000	0.4089703	0.4544544
	0.3721249	0.4089703	1.0000000	0.4583280
	0.4518753	0.4544544	0.4583280	1.0000000
9. A matrix: 4 CE 4 of type dbl	1.0000000	0.3335088	0.3717337	0.5350763
	0.3335088	1.0000000	0.4771118	0.5148450
	0.3717337	0.4771118	1.0000000	0.4939253
	0.5350763	0.5148450	0.4939253	1.0000000
10. A matrix: 4 CE 4 of type dbl	1.0000000	0.3711668	0.4469332	0.5756336
	0.3711668	1.0000000	0.5082334	0.5446554
	0.4469332	0.5082334	1.0000000	0.5336249
	0.5756336	0.5446554	0.5336249	1.0000000

```
In [252]: # do TCA-TWAS for every gene
prop = 0.7
t_prop = floor(prop*params$N)
li = list()
for(g in 1:length(data)){
  print(paste('gene',g,sep='_'))
  train_X = (data[[g]]$X)[1:t_prop,]
  test_X = data[[g]]$X[(t_prop+1):params$N,]
  train_W = params$W[1:t_prop,]
  test_W = params$W[(t_prop+1):params$N,]
  train_G = data[[g]]$G[,1:t_prop]
  test_G = data[[g]]$G[, (t_prop+1):params$N]
  train_c1 = params$c1[1:t_prop,]
  test_c1 = params$c1[(t_prop+1):params$N,]
  train_c2 = params$c2[1:t_prop,]
  test_c2 = params$c2[(t_prop+1):params$N,]
  train_Z = data[[g]]$Z[1:t_prop,]
  test_Z = data[[g]]$Z[(t_prop+1):params$N,]
  beta = data[[g]]$beta
  gamma_c1 = data[[g]]$gamma_c1
```

```

gamma_c2 = data[[g]]$gamma_c2
tca.mdl1 = tca(X=train_G,W=train_W,C1=cbind(train_X,train_c1),verbose=FALSE)
tca.mdl2 = tca(X=train_G,W=train_W,C1=train_c1,verbose=FALSE)
tca.mdl1.summary = summary_statistics mdl=tca.mdl1,train_X=train_X,test_X=test_X,
                                     train_c1=train_c1,test_c1=test_c1,train_G=train_G,
                                     test_G=test_G,train_Z=train_Z,test_Z=test_Z,
                                     beta=beta,gamma_c1=gamma_c1,gamma_c2=gamma_c2
tca.mdl2.summary = summary_statistics mdl=tca.mdl2,train_X=train_X,test_X=test_X,
                                     train_c1=train_c1,test_c1=test_c1,train_G=train_G,
                                     test_G=test_G,train_Z=train_Z,test_Z=test_Z,
                                     beta=beta,gamma_c1=gamma_c1,gamma_c2=gamma_c2
li[[g]] = list('s1'=tca.mdl1.summary, 's2'=tca.mdl2.summary)
}

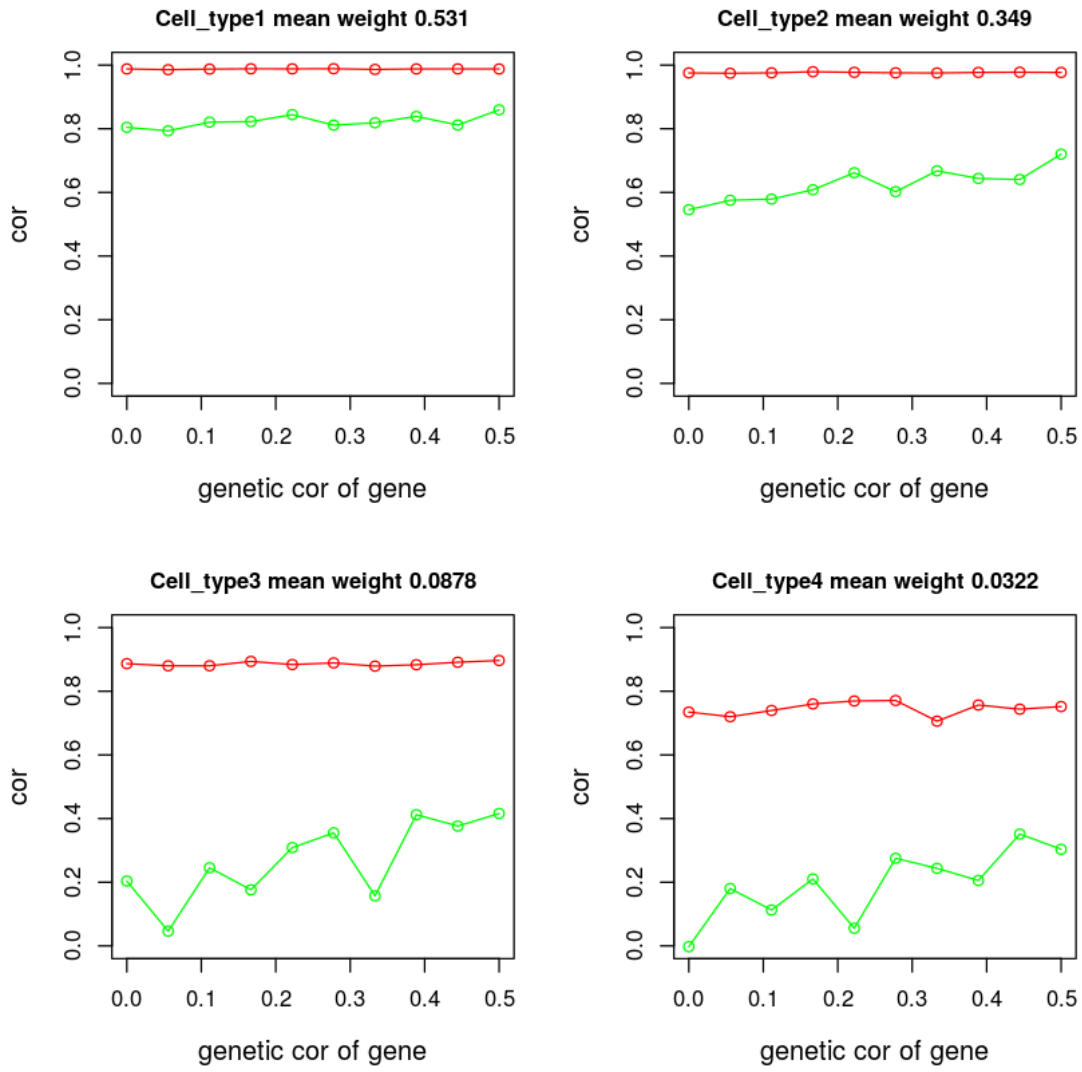
[1] "gene_1"
[1] "gene_2"
[1] "gene_3"
[1] "gene_4"
[1] "gene_5"
[1] "gene_6"
[1] "gene_7"
[1] "gene_8"
[1] "gene_9"
[1] "gene_10"

In [253]: mean_weight = signif(apply(params$W,2,mean),digits=3)
Z_hat_cor_tca1 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$Z_hat_cor_
Z_hat_cor_tca2 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s2$Z_hat_cor_

par(mfrow=c(2,2),mar=c(4, 3.8, 4, 2) + 0.1)
opts=params$cell_name
for(i in 1:length(params$cell_name)){
  heading = paste(opts[i], 'mean weight', mean_weight[i], sep=' ')
  plot(params$gene_cor,Z_hat_cor_tca1[,i],xlab='genetic cor of gene',
        ylab='cor',cex.lab=1.2,ylim=c(0,1),yaxs='r',type='o',col='red')
  title(main=heading,font.main=2,cex.main=1,line=1,cex.lab=1)
  lines(params$gene_cor,Z_hat_cor_tca2[,i],col='green',type='o')
}
mtext("TCA's Z_hat <Cor> Ground truth train Z",side=3,adj=6,cex=0.9,line=24,font=2)
par(mfrow=c(2,1),mar=c(0,0,0,0))
plot(1,type='n',axes=FALSE,xlab='',ylab='')
legend(x='top',inset=0,legend=c('mdl1:SNPs+C1','mdl2:C1'),
       col=c('red','green'),lwd=2,cex=0.7,horiz=TRUE)

```

TCA's \hat{Z} <Cor> Ground truth train Z



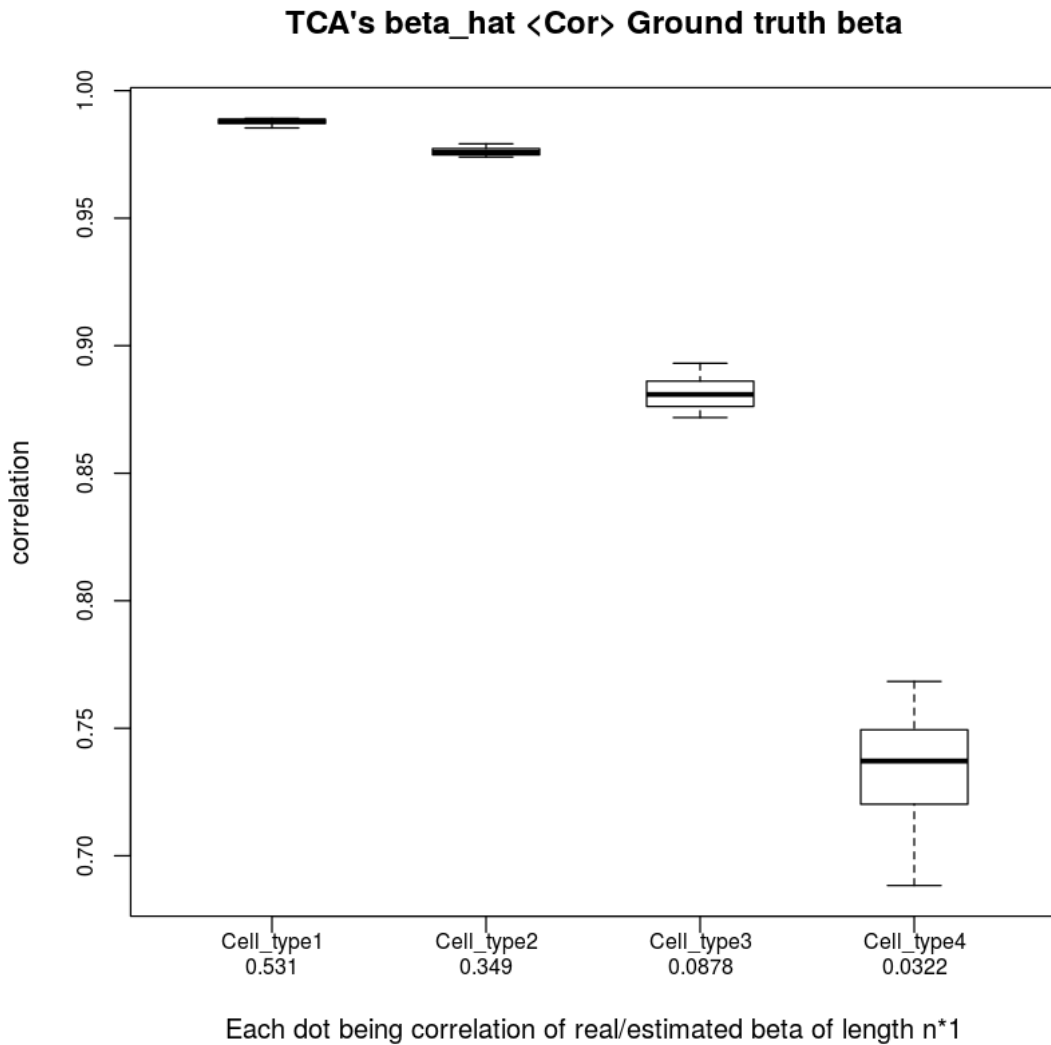


```
In [254]: beta_hat_cor_tca = do.call(rbind,lapply(1:length(data),function(x) unlist(li[[x]]$s1$b
colnames(beta_hat_cor_tca) = paste(params$cell_name,mean_weight,sep='\n')
cor_beta_lasso_real1 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$cor_
cor_beta_lasso_real2 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s2$cor_
#colnames(cor_beta_lasso_real1) = paste(params$cell_name,mean_weight,sep='\n')
#colnames(cor_beta_lasso_real2) = paste(params$cell_name,mean_weight,sep='\n')
# Cell type weight and correlation
par(cex.axis=.8)
boxplot(beta_hat_cor_tca,ylab='correlation',
        xlab='Each dot being correlation of real/estimated beta of length n*1',
        boxwex=0.5,main="TCA's beta_hat <Cor> Ground truth beta")
# Heritability and correlation
par(mfrow=c(2,2),mar=c(4, 3.8, 4, 2) + 0.1)
```

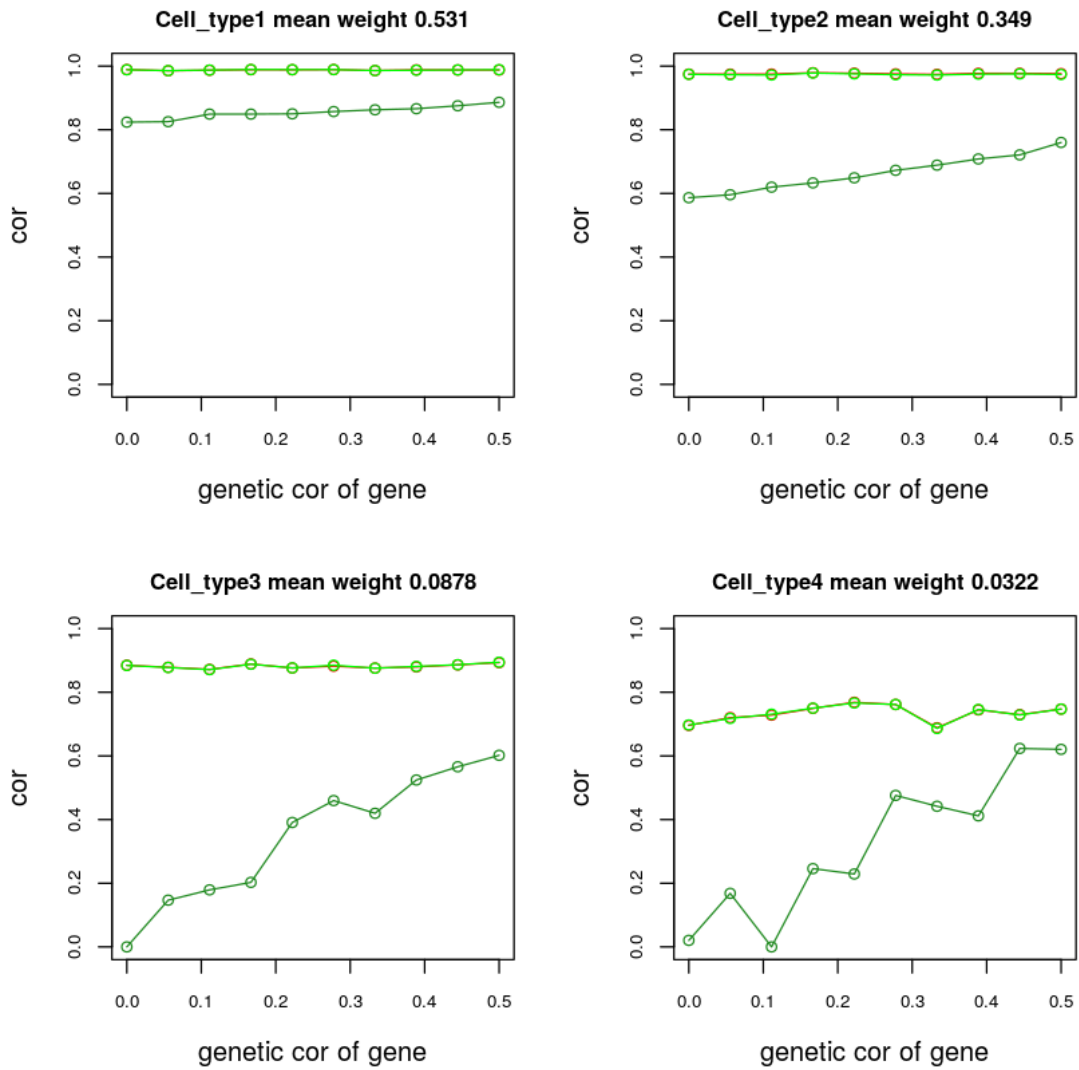
```

opts=params$cell_name
for(i in 1:length(params$cell_name)){
  heading = paste(opts[i], 'mean weight', mean_weight[i], sep=' ')
  plot(params$gene_cor, beta_hat_cor_tca[,i], xlab='genetic cor of gene',
        ylab='cor', cex.lab=1.2, ylim=c(0,1), yaxs='r', type='o', col='red')
  lines(params$gene_cor, cor_beta_lasso_real1[,i], type='o', col='green')
  lines(params$gene_cor, cor_beta_lasso_real2[,i], type='o', col='forestgreen')
  title(main=heading, font.main=2, cex.main=1, line=1, cex.lab=1)
}
mtext("Estimated beta_hat <Cor> Ground truth beta hat", side=3, adj=2, cex=0.9, line=24, font=1)
par(mfrow=c(2,1), mar=c(0,0,0,0))
plot(1, type='n', axes=FALSE, xlab='', ylab='')
legend(x='top', inset=0, legend=c("TCA's beta_hat mdl1:SNPs+C1", "Lasso's beta_hat mdl1:SNPs+C1"),
       col=c('red', 'green', 'forestgreen'), lwd=2, cex=0.7, horiz=TRUE)

```

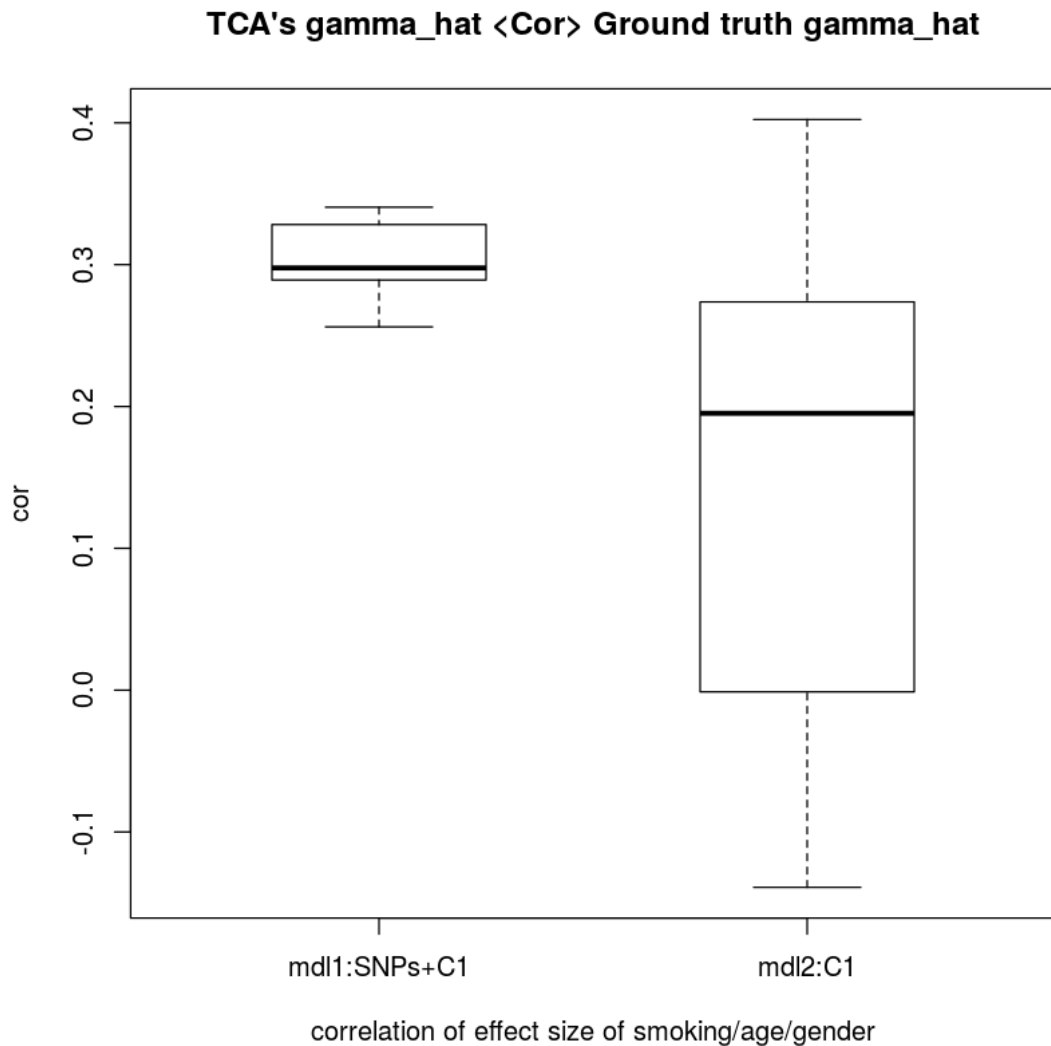


Estimated beta_hat <Cor> Ground truth beta hat



— TCA's beta_hat mdl1:SNPs+C1	— Lasso's beta_hat mdl1:SNPs+C1	— Lasso's beta_hat mdl2:C1
--	--	---

```
In [255]: gamma_hat_cor_tca1 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$gamma_hat))
gamma_hat_cor_tca2 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s2$gamma_hat))
gamma_cor = cbind(gamma_hat_cor_tca1,gamma_hat_cor_tca2)
colnames(gamma_cor) = c('mdl1:SNPs+C1','mdl2:C1')
boxplot(gamma_cor,main="TCA's gamma_hat <Cor> Ground truth gamma_hat",
        ylab='cor',boxwex=0.5,
        xlab='correlation of effect size of smoking/age/gender')
```

```
In [256]: Z_hat_train_seq_eq_cor_tca1 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$
Z_hat_train_seq_eq_cor_tca2 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$
Z_hat_test_seq_eq_cor_tca1 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$
Z_hat_test_seq_eq_cor_tca2 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$

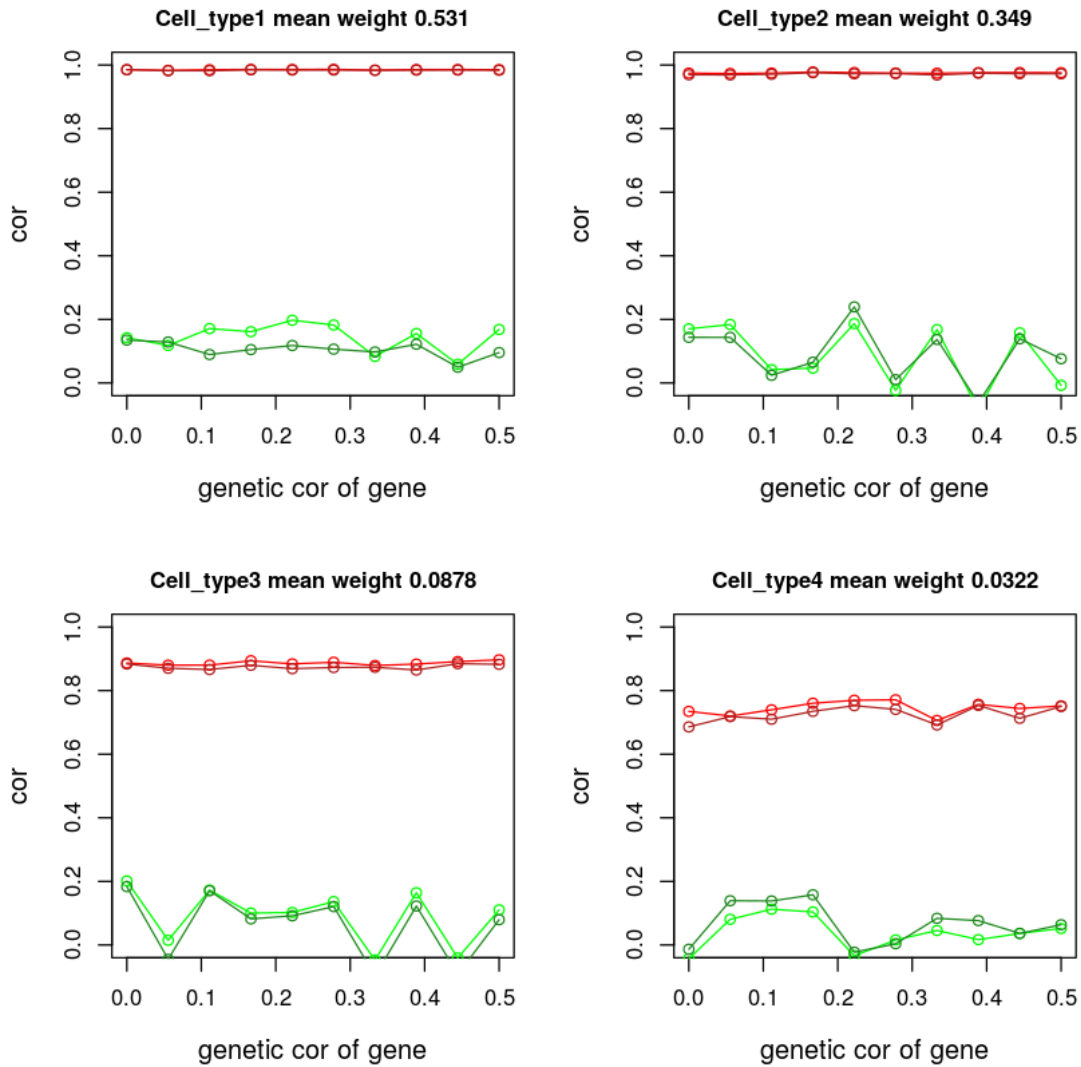
par(mfrow=c(2,2),mar=c(4, 3.8, 4, 2) + 0.1)
opts=params$cell_name
for(i in 1:params$K){
  heading = paste(opts[i], 'mean weight', mean_weight[i], sep=' ')
  plot(params$gene_cor,Z_hat_train_seq_eq_cor_tca1[,i],xlab='genetic cor of gene',
        ylab='cor',cex.lab=1.2,ylim=c(0,1),yaxs='r',type='o',col='red')
  title(main=heading,font.main=2,cex.main=1,line=1,cex.lab=1)
  lines(params$gene_cor,Z_hat_train_seq_eq_cor_tca2[,i],col='green',type='o')
```

```

    lines(params$gene_cor,Z_hat_test_seq_eq_cor_tca1[,i],col='firebrick',type='o')
    lines(params$gene_cor,Z_hat_test_seq_eq_cor_tca2[,i],col='forestgreen',type='o')
  }
  mtext("TCA's Z_hat 2nd eq <Cor> Ground truth train Z",side=3,adj=3,cex=0.9,line=24,font=1)
  par(mfrow=c(2,1),mar=c(0,0,0,0))
  plot(1,type='n',axes=FALSE,xlab='',ylab='')
  legend(x='top',inset=0,legend=c('mdl1 train:SNPs+C1','mdl2 train:C1','mdl1 test:SNPs+C1','mdl2 test:C1'),
        col=c('red','green','firebrick','forestgreen'),lwd=1,cex=0.7,hORIZ=TRUE)

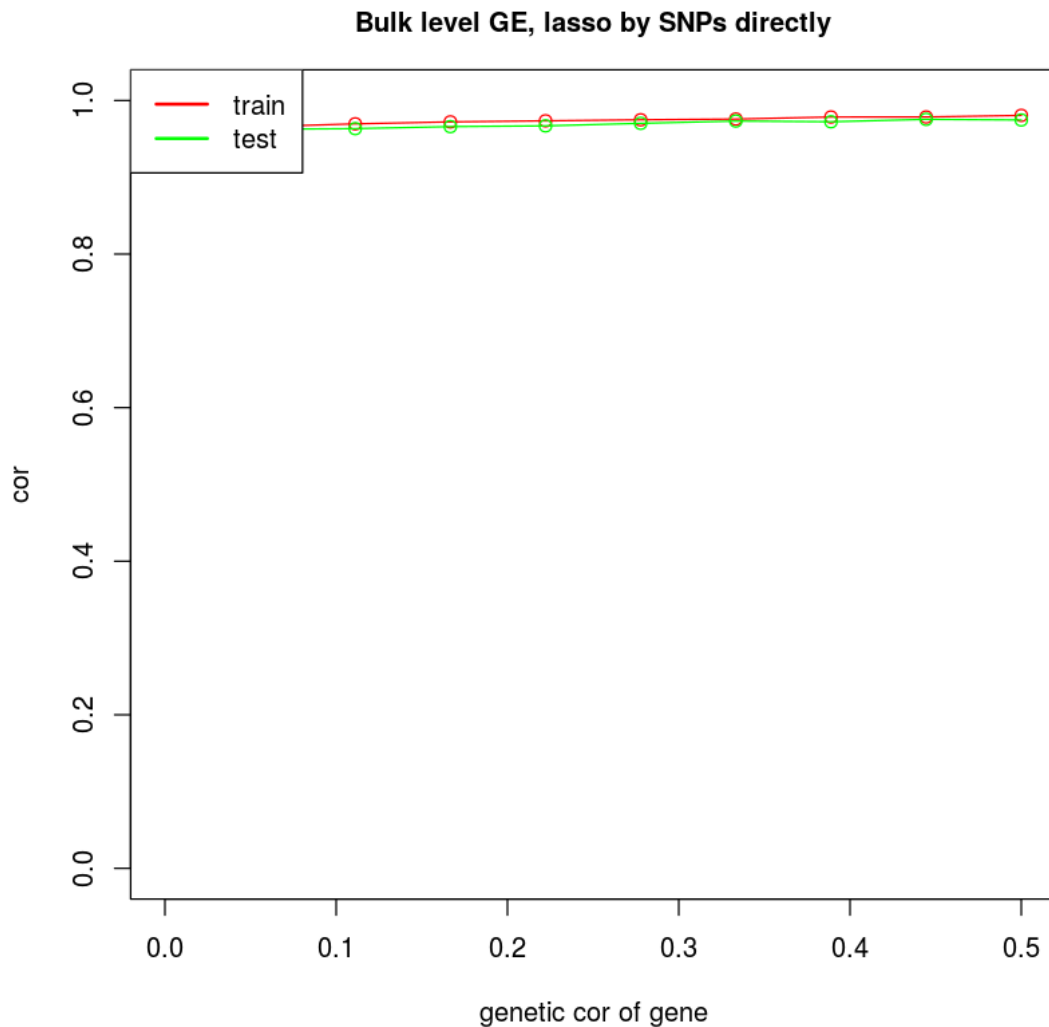
```

TCA's Z_hat 2nd eq <Cor> Ground truth train Z





```
In [257]: G_hat_train_cor_bulk = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$G_hat_train_cor_bulk))
G_hat_test_cor_bulk = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$G_hat_test_cor_bulk))
plot(params$gene_cor,G_hat_train_cor_bulk,type='o',col='red',xlab='genetic cor of gene',ylab='G_hat_train_cor_bulk')
mtext("Bulk level GE, lasso by SNPs directly",font=2,line=1)
lines(params$gene_cor,G_hat_test_cor_bulk,col='green',type='o')
legend(x='topleft',inset=0,legend=c('train','test'),
      col=c('red','green'),lwd=2,cex=1,hORIZ=FALSE)
```



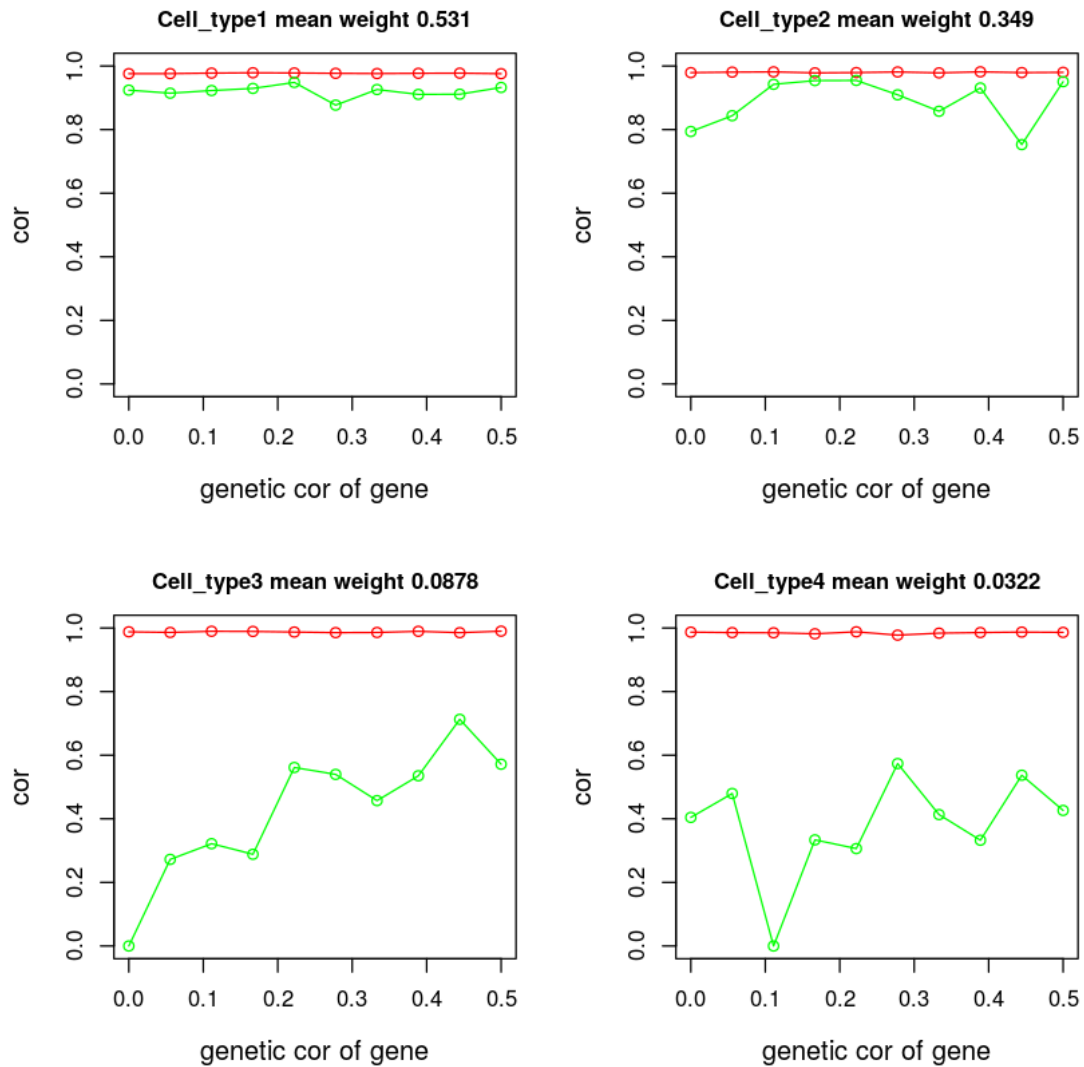
```
In [258]: cor_lasso_tca1 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$cor_lasso_
cor_lasso_tca2 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s2$cor_lasso_
par(mfrow=c(2,2),mar=c(4, 3.8, 4, 2) + 0.1)
opts=params$cell_name
for(i in 1:length(params$cell_name)){
  heading = paste(opts[i], 'mean weight', mean_weight[i], sep=' ')
  plot(params$gene_cor, cor_lasso_tca1[,i], xlab='genetic cor of gene',
        ylab='cor', cex.lab=1.2, ylim=c(0,1), yaxs='r', type='o', col='red')
  title(main=heading, font.main=2, cex.main=1, line=1, cex.lab=1)
  lines(params$gene_cor, cor_lasso_tca2[,i], col='green', type='o')
}
mtext("TCA's Z_hat <Cor> Lasso Z_hat", side=3, adj=-7, cex=0.9, line=24, font=2)
```

```

par(mfrow=c(2,1),mar=c(0,0,0,0))
plot(1,type='n',axes=FALSE,xlab='',ylab='')
legend(x='top',inset=0,legend=c('mdl1:SNPs+C1','mdl2:C1'),
      col=c('red','green'),lwd=2,cex=0.7,horiz=TRUE)

```

TCA's Z_{hat} <Cor> Lasso Z_{hat}





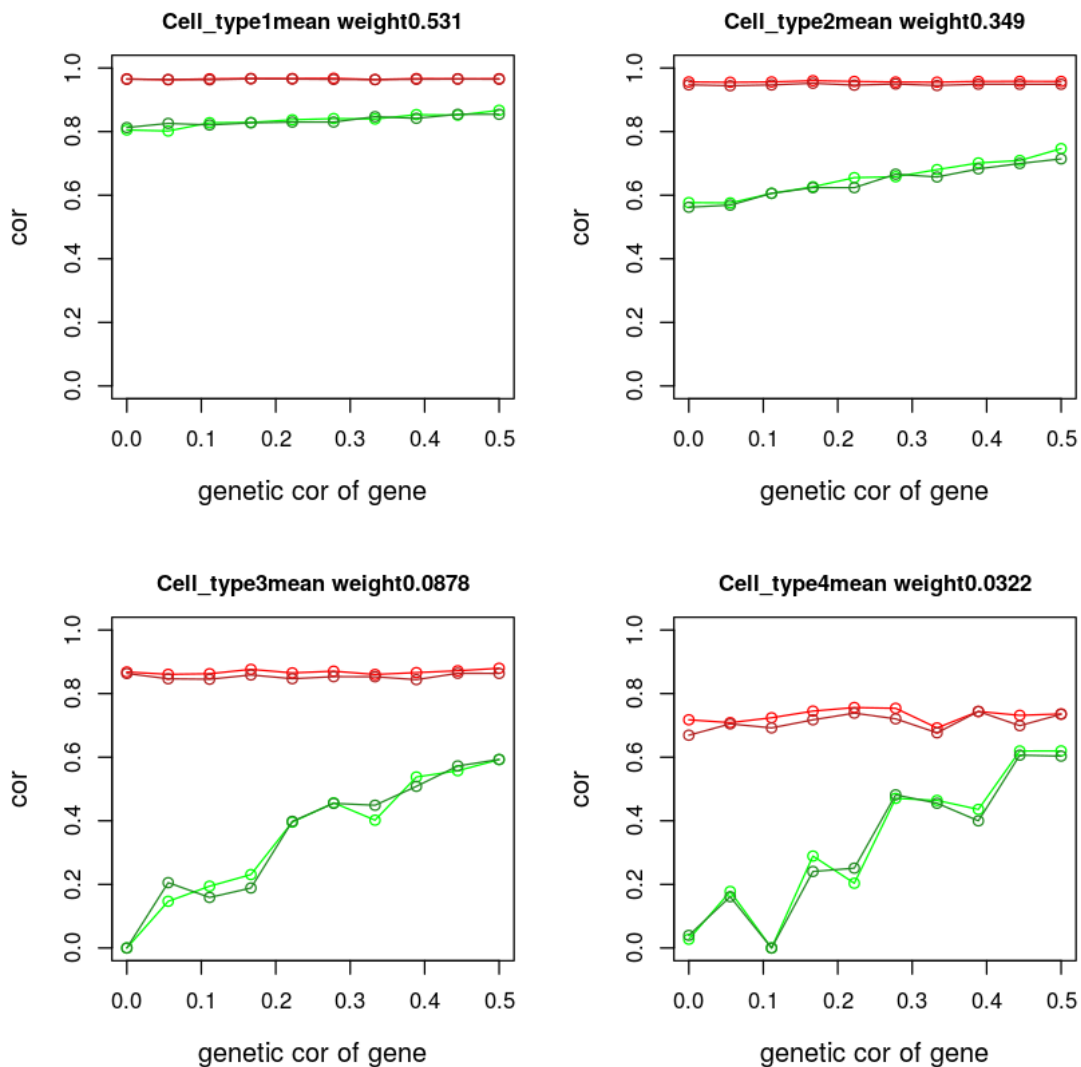
```
In [259]: cor_lasso_real_train1 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$cor_
cor_lasso_real_train2 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s2$cor_
cor_lasso_real_test1 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$cor_
cor_lasso_real_test2 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s2$cor_
colnames(cor_lasso_real_train1)=paste(params$cell_name,'\n',mean_weight)
colnames(cor_lasso_real_train2)=paste(params$cell_name,'\n',mean_weight)
colnames(cor_lasso_real_test1)=paste(params$cell_name,'\n',mean_weight)
colnames(cor_lasso_real_test2)=paste(params$cell_name,'\n',mean_weight)
par(mfrow=c(2,2),mar=c(4,3.8,4,2)+0.1)
opts=params$cell_name
for(i in 1:length(params$cell_name)){
  heading=paste(opts[i], 'mean weight',mean_weight[i],sep='')
  plot(params$gene_cor,cor_lasso_real_train1[,i],xlab='genetic cor of gene',
```

```

        ylab='cor',cex.lab=1.2,ylim=c(0,1),yaxs='r',type='o',col='red')
    title(main=heading,font.main=2,cex.main=1,line=1,cex.lab=1)
    lines(params$gene_cor,cor_lasso_real_train2[,i],col='green',type='o')
    lines(params$gene_cor,cor_lasso_real_test1[,i],col='firebrick',type='o')
    lines(params$gene_cor,cor_lasso_real_test2[,i],col='forestgreen',type='o')
}
mtext("Lasso Z_hat <Cor> Ground Truth Z",side=3,adj=7,cex=1,line=24,font=2)
par(mfrow=c(2,1),mar=c(0,0,0,0))
plot(1,type='n',axes=FALSE,xlab='',ylab='')
legend(x='top',inset=0,legend=c('mdl1 train:SNPs+C1','mdl2 train:C1','mdl1 test:SNPs+C1',
    col=c('red','green','firebrick','forestgreen'),lwd=2,cex=0.7,hORIZ=TRUE)

```

Lasso Z_hat <Cor> Ground Truth Z





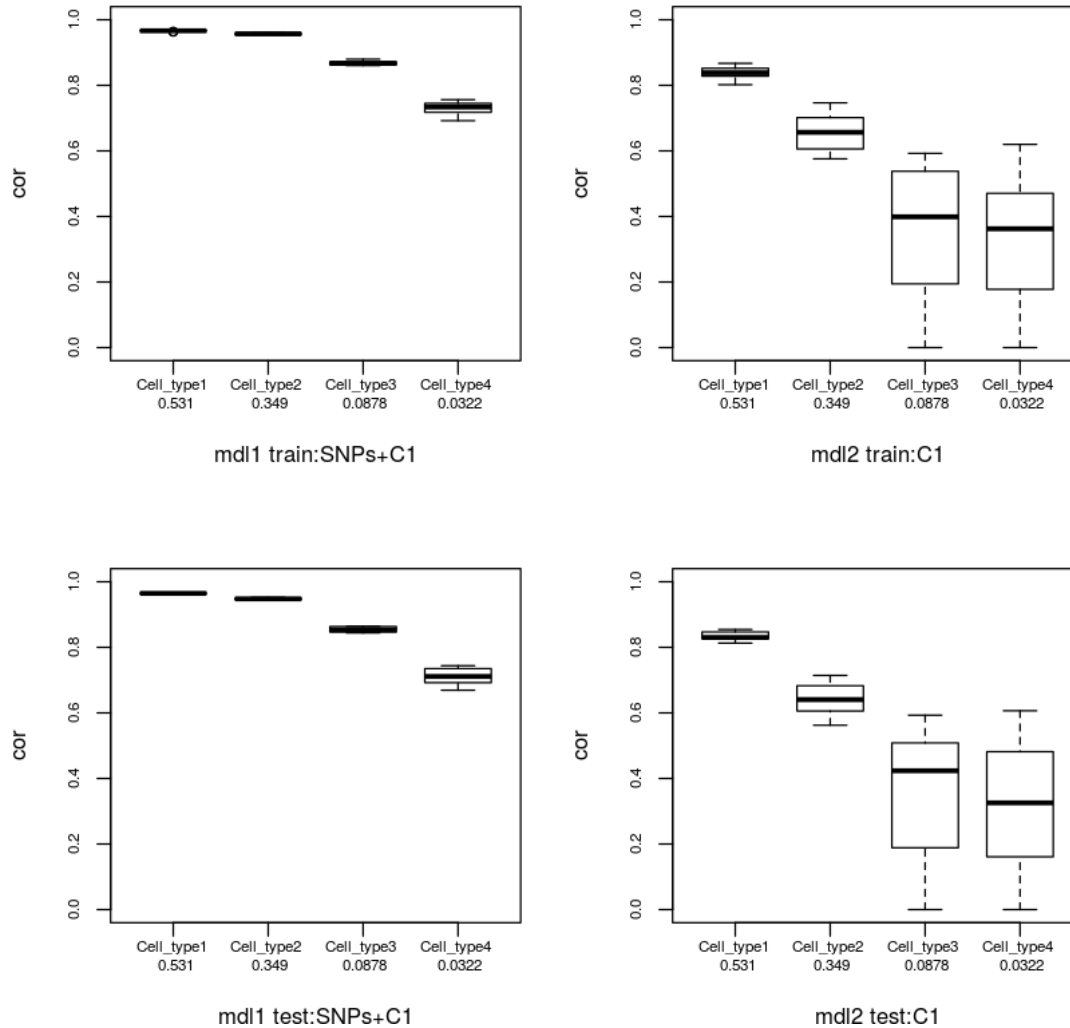
```
In [260]: # zhat_cor = data.frame(heritability=params$gene_cor, cor=cor_lasso_real_test1)
# colnames(zhat_cor) = c('heritability', 'cor_celltype1', 'cor_celltype2', 'cor_celltype3')
# ggplot(zhat_cor, aes(heritability)) + geom_line(
#   aes(y=cor_celltype1, color=hue_pal()(4)[1])) + geom_line(
#   aes(y=cor_celltype2, color=hue_pal()(4)[2])) + geom_line(
#   aes(y=cor_celltype3, color=hue_pal()(4)[3])) + geom_line(
#   aes(y=cor_celltype4, color=hue_pal()(4)[4])) + ggtitle(
#     "Correlation: TWAS Z_hat and real Z") + theme(
#     axis.text=element_text(size=18),
#     axis.title=element_text(size=18, face="bold"),
#     plot.title = element_text(size=15, face="bold", hjust=0.5)) + labs(
#     y="Correlation") + scale_color_discrete(name="Cell Type Weights", labels=c(0.08, 0.3, 0.3, 0.3))
# # ggsave("/home/elessar/Documents/activities/hackathon/urop/UCLA/csst/tca-twas/pipel")
```



```
# # plot = last_plot(),dpi = 300, limitsize = TRUE)
```

```
In [261]: par(mar=c(0,0,0,0),cex.axis=.7)
par(mfrow=c(2,2),mar=c(3.8, 3.7, 4, 2))
boxplot(cor_lasso_real_train1,ylab='cor',
        xlab='mdl1 train:SNPs+C1',boxwex=0.7,ylim=c(0,1))
boxplot(cor_lasso_real_train2,ylab='cor',
        xlab='mdl2 train:C1',boxwex=0.7,ylim=c(0,1))
boxplot(cor_lasso_real_test1,ylab='cor',
        xlab='mdl1 test:SNPs+C1',ylim=c(0,1),boxwex=0.7)
boxplot(cor_lasso_real_test2,ylab='cor',
        xlab='mdl2 test:C1',boxwex=0.7,ylim=c(0,1))
mtext("Lasso Z_hat <Cor> Ground Truth Z",side=3,adj=8,cex=1,line=23,font=2)
```

Lasso Z_hat <Cor> Ground Truth Z

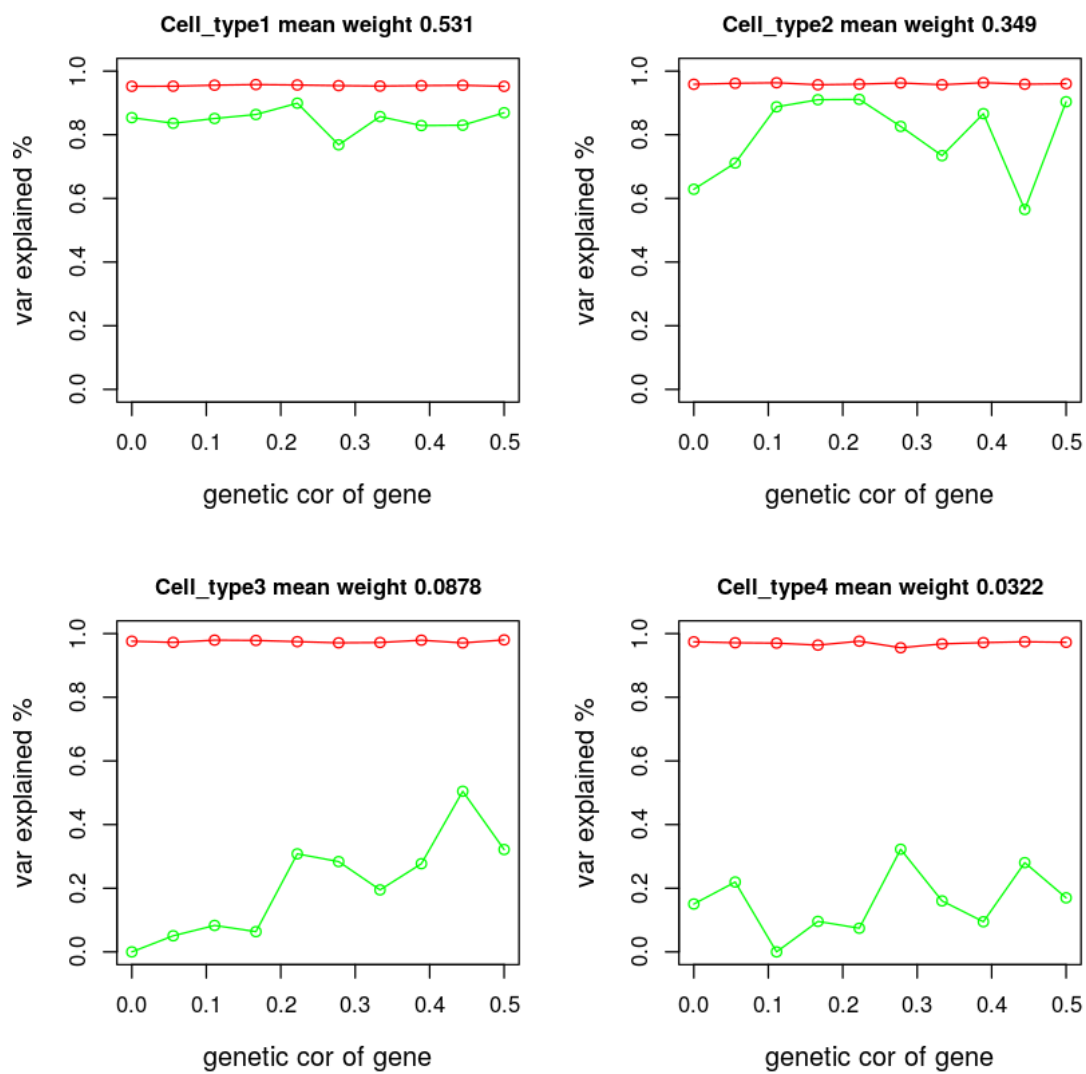


```

In [262]: lasso_dev_rat1 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$dev_rat))
lasso_dev_rat2 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s2$dev_rat))
colnames(lasso_dev_rat1) = paste(params$cell_name,'\n',mean_weight)
colnames(lasso_dev_rat2) = paste(params$cell_name,'\n',mean_weight)
par(mfrow=c(2,2),mar=c(4, 3.8, 4, 2) + 0.1)
opts=params$cell_name
for(i in 1:params$K){
  heading = paste(opts[i], 'mean weight', mean_weight[i], sep=' ')
  plot(params$gene_cor,lasso_dev_rat1[,i],xlab='genetic cor of gene',
        ylab='var explained %',cex.lab=1.2,ylim=c(0,1),yaxs='r',type='o',col='red')
  title(main=heading,font.main=2,cex.main=1,line=1,cex.lab=1)
  lines(params$gene_cor,lasso_dev_rat2[,i],col='green',type='o')
}
mtext("Lasso Z_hat <dev exp> Ground Truth Z",side=3,adj=3.5,cex=1,line=23.5,font=2)
par(mfrow=c(2,1),mar=c(0,0,0,0))
plot(1,type='n',axes=FALSE,xlab='',ylab='')
legend(x='top',inset=0,legend=c('mdl1 train:SNPs+C1','mdl2 train:C1'),
       col=c('red','green'),lwd=2,cex=0.7,horiz=TRUE)

```

Lasso Z_hat <dev exp> Ground Truth Z





```
In [263]: # lasso_dev_rat1 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$dev_rat))
# colnames(lasso_dev_rat1) = paste(params$cell_name,'\n',mean_weight)
# zhat_dev_exp = data.frame(heritability=params$gene_cor,lasso_dev_rat1)
# colnames(zhat_dev_exp) = c('heritability','dev_celltype1','dev_celltype2','dev_cellt
# ggplot(zhat_dev_exp,aes(heritability))+geom_line(
#   aes(y=dev_celltype1,color=hue_pal()(4)[1]))+geom_line(
#   aes(y=dev_celltype2,color=hue_pal()(4)[2]))+geom_line(
#   aes(y=dev_celltype3,color=hue_pal()(4)[3]))+geom_line(
#   aes(y=dev_celltype4,color=hue_pal()(4)[4]))+ggtitle(
#     "Deviance Explained: TWAS deviance explained")+theme(
#     axis.text=element_text(size=18),
#     axis.title=element_text(size=18,face="bold"),
#     plot.title = element_text(size=15,face="bold",hjust=0.5))+labs(
```

```

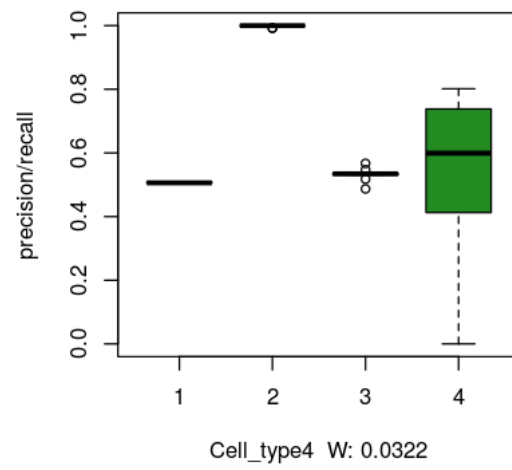
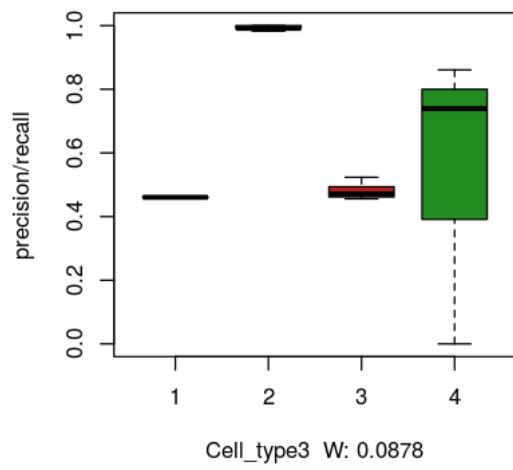
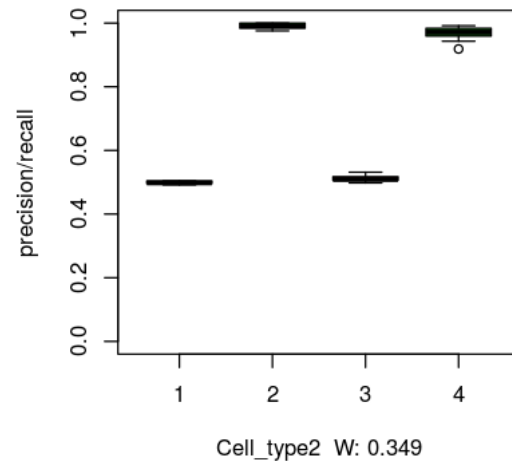
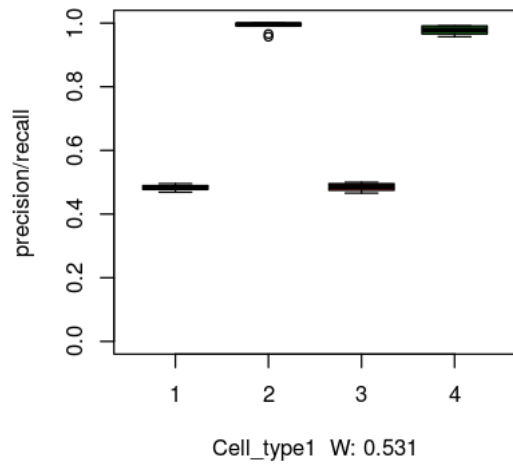
#      y="Dev Explained %")+scale_color_discrete(name="Cell Type Weights",labels=c(0.08
# # ggsave("/home/elessar/Documents/activities/hackathon/urop/UCLA/csst/tca-twas/pipel
# #      plot = last_plot(),dpi = 300, limitsize = TRUE)

In [264]: beta_precision_lasso1 =matrix(do.call(rbind,lapply(1:length(data),function(x) li[[x]]$
beta_precision_lasso2 = matrix(do.call(rbind,lapply(1:length(data),function(x) li[[x]]$
beta_recall_lasso1 =matrix(do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$
beta_recall_lasso2 = matrix(do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s2
prec_rec_frame = lapply(1:params$K,function(x) t(rbind(beta_precision_lasso1[x,],beta_

par(mfrow=c(2,2),mar=c(4, 3.8, 4, 2) + 0.1)
opts=paste(params$cell_name,' W:',mean_weight)
for(i in 1:params$K){
  boxplot(prec_rec_frame[[i]],ylab='precision/recall',
          xlab=opts[i],boxwex=0.7,ylim=c(0,1),col=c('red','green','firebrick','forestgre
})
mtext("Precision/Recall for Lasso's beta_hat",side=3,adj=12.5,cex=0.9,line=22.5,font=2)
par(mfrow=c(2,1),mar=c(0,0,0,0))
plot(1,type='n',axes=FALSE,xlab='',ylab='')
legend(x='top',inset=0,legend=c('precision mdl1:SNPs+C1','recall mdl1:SNPs+C1','precis
      col=c('red','green','firebrick','forestgreen'),lwd=1,cex=0.65,hORIZ=TRUE)

```

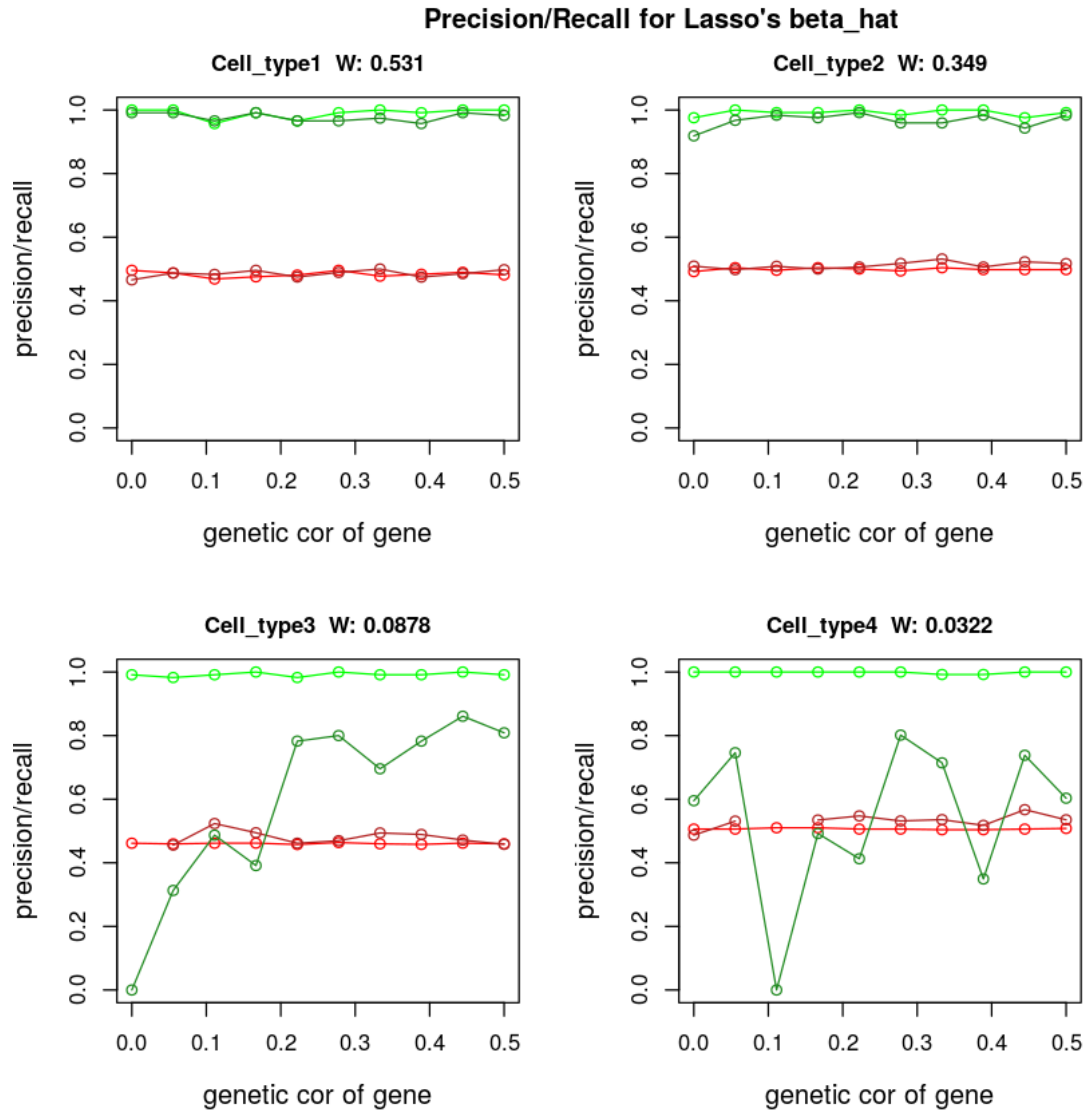
Precision/Recall for Lasso's β_{hat}





```
In [265]: par(mfrow=c(2,2),mar=c(4, 3.8, 4, 2) + 0.1)
  for(i in 1:params$K){
    plot(params$gene_cor,prec_rec_frame[[i]][,1],xlab='genetic cor of gene',
          ylab='precision/recall',cex.lab=1.2,ylim=c(0,1),yaxs='r',type='o',col='red')
    title(main=opts[i],font.main=2,cex.main=1,line=1,cex.lab=1)
    lines(params$gene_cor,prec_rec_frame[[i]][,2],col='green',type='o')
    lines(params$gene_cor,prec_rec_frame[[i]][,3],col='firebrick',type='o')
    lines(params$gene_cor,prec_rec_frame[[i]][,4],col='forestgreen',type='o')
  }
  mtext("Precision/Recall for Lasso's beta_hat",side=3,adj=3.5,cex=1,line=23.5,font=2)
  par(mfrow=c(2,1),mar=c(0,0,0,0))
  plot(1,type='n',axes=FALSE,xlab='',ylab='')
  legend(x='top',inset=0,legend=c('precision mdl1:SNPs+C1','recall mdl1:SNPs+C1','precis
```

```
col=c('red','green','firebrick','forestgreen'),lwd=1,cex=0.65,hORIZ=TRUE)
```





```
In [266]: rmse_lasso1 =matrix(do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$rmse_lasso1)),nrow=length(data))
rmse_lasso2 = matrix(do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s2$rmse_lasso2)),nrow=length(data))
rmse_tca1 =matrix(do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$rmse_tca1)),nrow=length(data))
rmse_tca2 = matrix(do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s2$rmse_tca2)),nrow=length(data))
rmse_frame = lapply(1:params$K,function(x) t(rbind(rmse_lasso1[x,],rmse_lasso2[x,],rmse_tca1[x,],rmse_tca2[x,])))

par(mfrow=c(2,2),mar=c(4, 3.8, 4, 2) + 0.1)
opts=paste(params$cell_name, ' W:',mean_weight)
for(i in 1:params$K){
  boxplot(rmse_frame[[i]],ylab='rmse',
          xlab=opts[i],boxwex=0.7,ylim=c(0,1),col=c('red','green','firebrick','forestgreen'))
}
mtext("RMSE for lasso / TCA",side=3,adj=-1.5,cex=0.9,line=22.5,font=2)
```

```

par(mfrow=c(2,1),mar=c(0,0,0,0))
plot(1,type='n',axes=FALSE,xlab='',ylab='')
legend(x='top',inset=0,legend=c('lasso mdl1:SNPs+C1','lasso mdl2:C1','tca mdl1:SNPs+C1',
col=c('red','green','firebrick','forestgreen'),lwd=1,cex=0.65,hORIZ=TRUE)

```

