TCA_heritibility_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 [9]:
```
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]),tra
    cell_type = ncol(mdl$W)
    if(model==1){
        tca_beta = t(matrix(mdl$gammas_hat[,grepl("_SNP",colnames(mdl$gammas_hat))],ncol
        beta_hat_cor_tca = lapply(1:cell_type,function(x)
            cor(beta[x,],mdl$gammas_hat[,grepl(paste("Cell_type",x,".G",sep=''),colnames
    }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),
                            mdl$gammas_hat[,grepl('male|smok|age', colnames(mdl$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$gammas_hat[,
    Z_hat_train_second_eq_cor_tca = diag(cor(train_Z,Z_hat_train_second_eq_tca))
    Z_hat_test_second_eq_tca = sapply(1:cell_type,function(x) test_pred %*% mdl$gammas_h
    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 recorrelate
    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=1,
    rmse_lasso[h] = sqrt(glmnet.mdl.cv$cvm[glmnet.mdl.cv$lambda == glmnet.mdl.cv$lam
    glmnet.mdl <- glmnet(x=train_X,y=Z_hat_tca[[h]],standardize=FALSE,alpha=1,lambda
    dev_ratio[h] <- glmnet.mdl$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$beta[predictors,])))
    Z_hat_train_lasso <- cbind(numeric(nrow(train_X))+1,train_X[,predictors]) %*% be
    Z_hat_test_lasso[,h] <- cbind(numeric(nrow(test_X))+1,test_X[,predictors]) %*% b
    Z_hat_test_lasso[,h] = signif(Z_hat_test_lasso[,h], digits = 5)
    if(sum(beta_lasso)==0 | sd(Z_hat_train_lasso)==0){
        # model 2 lasso forces SNPs effect to become zero.
        # If there were no predictors, then cor would be NaN. Which also means
        # no snps is correlated with Z. cor just set to 0.
        corrs[h] = 0
        corrs.real[h] = 0
        corrs.beta[h] = 0
        corrs.test.real[h] = 0
    }else{
        corrs[h] <- cor(t(Z_hat_tca[[h]]),Z_hat_train_lasso)
        corrs.real[h] <- cor(train_Z[,h],Z_hat_train_lasso)
        corrs.beta[h] <- cor(beta[h,],beta.full)
        corrs.test.real[h] <- cor(test_Z[,h],Z_hat_test_lasso[,h])
    }
}
# Precision & Recall for lasso
Binary_True=(beta!=0)
Binary_Pred_Lasso=(beta_full_cell!=0)
TP = do.call(rbind,lapply(1:params$K,function(x) sum(as.integer(Binary_True[x,]&Bina
FP = do.call(rbind,lapply(1:params$K,function(x) sum(as.integer((!Binary_True[x,])&B
FN = do.call(rbind,lapply(1:params$K,function(x) sum(as.integer(Binary_True[x,]&(!Bi
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_tca
                    'Z_hat_train_second_eq_cor_tca'=Z_hat_train_second_eq_cor_tca,'Z_hat_test
                    'G_hat_train_cor_bulk'=G_hat_train_cor_bulk,'G_hat_test_cor_bulk'=G_hat_t
                    'cor_lasso_tca'=corrs,'cor_lasso_real_train'=corrs.real,'cor_lasso_real_t
                    'cor_beta_lasso_real'=corrs.beta,'beta_hat_lasso'=beta_full_cell,'dev_rat
                    'precision_lasso'=precision_lasso,'recall_lasso'=recall_lasso,'rmse_lasso
    }
```

In [67]:
```
generate_params <- function(cell_her=TRUE,gene_cor=FALSE,bulk_her=FALSE,seed=1,N=5000,M
    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){ # varing 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){ # varing heribility across genes
        heritibility_cell_specific = do.call(rbind,lapply(1:K,function(x) herr_arr))
    }else{
        heritibility_cell_specific = matrix(her,nrow=K,ncol=M)
    }

    if(bulk_her){ # varing bulk heritibility across genes
        heritibility_bulk = seq(from=0.1,to=her_bulk,length.out=M)
    }else{
        heritibility_bulk = array(her_bulk,M)
    }

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

```
    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))
    c1 = scale(cbind(male, smoking, age))
    c2 = scale(matrix(rnorm(N*pc_num), nrow=N))

    rownames(c1) = id_name
    colnames(c1) = c('male','smoking','age')
    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-heritibil

    # 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
        '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
        'sigma_z'=sigma_z,'beta_cor'=corr_matrix,'var_beta'=var_beta,'W_alpha'=W_alpha[
}
```

In [70]:
```
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 %*%

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))
gamma_c1 = do.call(rbind,lapply(1:cell_type,function(x) rnorm(dim(par$c1)[2],mean=0
# enforce the c1_gamma variance
c1gamma_var = dim(par$c1)[2]*(par$sigma_gamma[g])^2
gamma_c1 = do.call(rbind,
                   lapply(1:cell_type,function(x) gamma_c1[x,]*(sqrt(c1gamma_var)/s
c1_gamma = par$c1 %*% t(gamma_c1)

Z = epsilon_z + mu_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'=t(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)
```

In [89]:
```
herr_arr=c(1:40)/100
params = generate_params(cell_her=TRUE,gene_cor=FALSE,bulk_her=FALSE,
                         seed=1,N=5000,M=10,herr_arr=herr_arr,
                         D=250,pslab=0.5,her=0.05,
                         her_bulk=0.6)
data = lapply(1:params$M,function(x) one_gene(par=params,x))
```

In [195]:
```
# # heritability is not correct.
# xbeta = data[[1]]$X %*% t(data[[1]]$beta)
# #apply(xbeta,2,mean)
# params$her[,1]
# ss_xbeta=apply(xbeta*xbeta,2,sum)
# ss_z = apply(data[[1]]$Z*data[[1]]$Z,2,sum)
# ss_xbeta/ss_z
# # Is it nominator or denominator?
# ss_xbeta
# params$var_beta[,1]*(params$N-1)*params$D
# cgamma = params$c1 %*% data[[1]]$gamma_c1
# print('next come ss of cgamma')
# ss_cgamma = apply(cgamma*cgamma,2,sum)
# (params$N-1) * 3*0.1*0.1
# ss_cgamma
# print('next come ss of epsilonz')
```

```
# ss_epsilonz = apply((data[[1]]$epsilon_z)^2,2,sum)
# (params$N-1)*0.1*0.1
# ss_epsilonz
```

In [72]:
```
# The reason why the covariance beta version does not perform
# The numerical value of variance is too small. And TCA does not perform any more
# G mean and var

# See overlapp version of notebook to copy the data generation function here
# params_p = generate_params_primitive(cont_her=TRUE,seed=1)
# data_p = lapply(1:params_p$M,function(x) one_gene_primitive(par=params_p,x))

# t(do.call(rbind,lapply(1:length(data),function(x) var(unlist(data[[x]]$G)))))
# t(do.call(rbind,lapply(1:length(data),function(x) var(unlist(data_p[[x]]$G)))))
# cov(data_p[[1]]$Z)
# cov(data[[1]]$Z)

# xbeta = data[[10]]$X %*% t(data[[10]]$beta)
# xbeta_p = data_p[[10]]$X %*% t(data_p[[10]]$beta)
# var(xbeta)
# var(xbeta_p)
```

A matrix: 1 Œ 10 of type dbl 0.6814677 0.6773699 0.6758098 0.6720685 0.6714916 0.6664994 0.66596

A matrix: 1 Œ 10 of type dbl 0.6809166 0.6796567 0.6792632 0.6791135 0.6800521 0.6770347 0.67699

A matrix: 4 Œ 4 of type dbl

	Cell_type1	Cell_type2	Cell_type3	Cell_type4
Cell_type1	0.9972474	0.5316513	0.549621568	-0.468061177
Cell_type2	0.5316513	0.9965857	0.688312735	-0.643856850
Cell_type3	0.5496216	0.6883127	0.998415003	0.007293881
Cell_type4	-0.4680612	-0.6438568	0.007293881	0.993795467

A matrix: 4 Œ 4 of type dbl

	Cell_type1	Cell_type2	Cell_type3	Cell_type4
Cell_type1	0.9967180	0.5351977	0.550816975	-0.477529910
Cell_type2	0.5351977	0.9973839	0.689083863	-0.651152905
Cell_type3	0.5508170	0.6890839	1.003236678	-0.004356544
Cell_type4	-0.4775299	-0.6511529	-0.004356544	0.997140218

0.0.1 Plotting Begins for Heritability Check

In [168]:
```
herr_arr=c(1:40)/100
params = generate_params(cell_her=TRUE,gene_cor=FALSE,bulk_her=FALSE,
                         seed=1,N=5000,M=40,herr_arr=herr_arr,
                         D=250,pslab=0.5,her=0.05,
                         her_bulk=0.6)
data = lapply(1:params$M,function(x) one_gene(par=params,x))
```

In [90]:
```
# 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=tra
                                                  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=tra
                                                  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"
```

```
Warning message in cor(t(train_G), G_hat_train_lasso):
the standard deviation is zeroWarning message in cor(t(test_G), G_hat_test_lasso):
the standard deviation is zero
```

```
[1] "gene_2"
```

```
Warning message in cor(t(train_G), G_hat_train_lasso):
the standard deviation is zeroWarning message in cor(t(test_G), G_hat_test_lasso):
the standard deviation is zeroWarning message in cor(t(train_G), G_hat_train_lasso):
the standard deviation is zeroWarning message in cor(t(test_G), G_hat_test_lasso):
the standard deviation is zeroWarning message in cor(test_Z[, h], Z_hat_test_lasso[, h]):
the standard deviation is zero
```

```
[1] "gene_3"
[1] "gene_4"
```

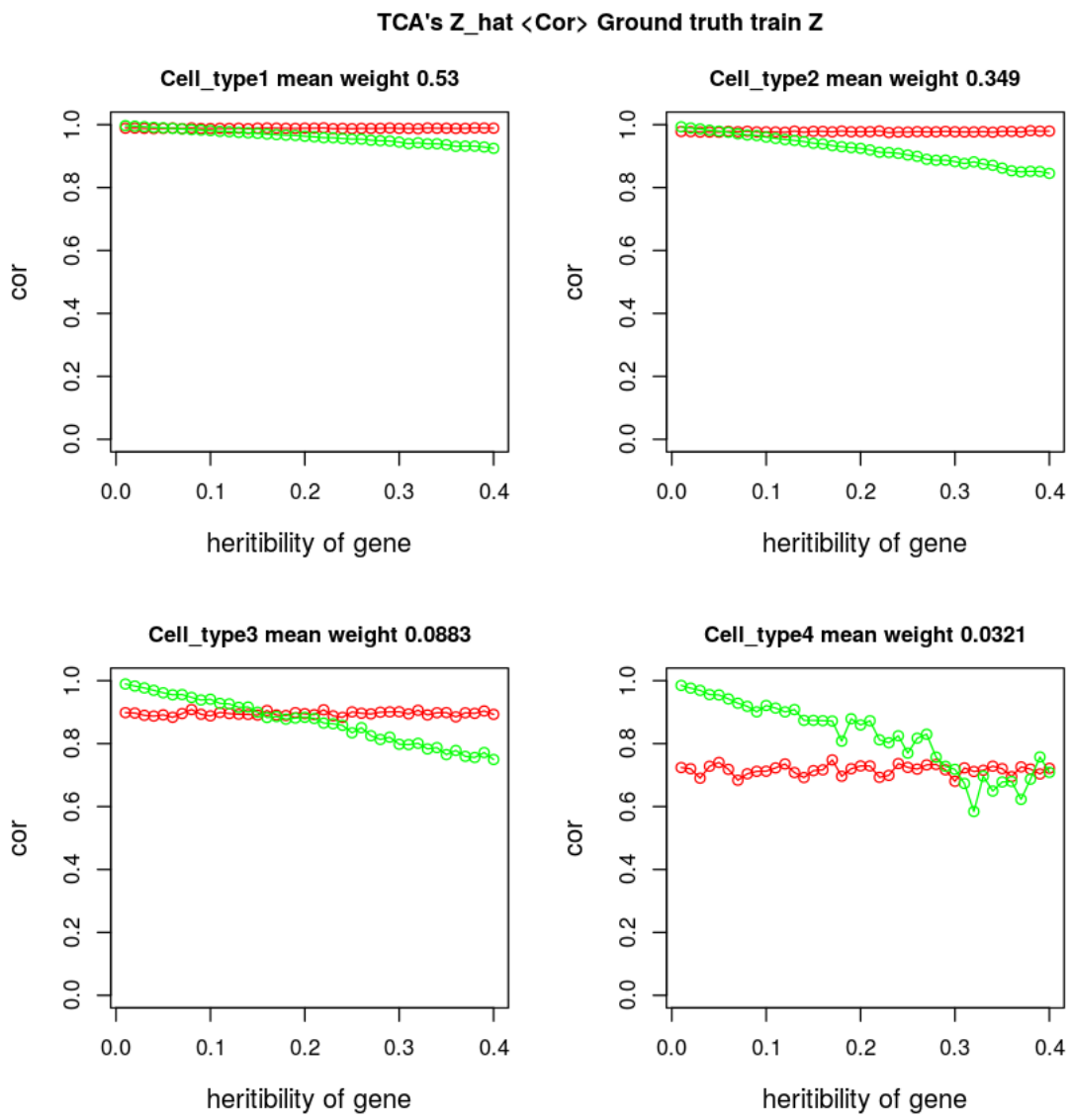
```
Warning message in cor(test_Z[, h], Z_hat_test_lasso[, h]):
the standard deviation is zero

[1] "gene_5"
[1] "gene_6"
[1] "gene_7"
[1] "gene_8"


Warning message in cor(test_Z[, h], Z_hat_test_lasso[, h]):
the standard deviation is zero

[1] "gene_9"


Warning message in cor(test_Z[, h], Z_hat_test_lasso[, h]):
the standard deviation is zero

[1] "gene_10"
[1] "gene_11"
[1] "gene_12"
[1] "gene_13"
[1] "gene_14"
[1] "gene_15"


Warning message in cor(test_Z[, h], Z_hat_test_lasso[, h]):
the standard deviation is zero

[1] "gene_16"


Warning message in cor(test_Z[, h], Z_hat_test_lasso[, h]):
the standard deviation is zeroWarning message in cor(test_Z[, h], Z_hat_test_lasso[, h]):
the standard deviation is zero

[1] "gene_17"


Warning message in cor(test_Z[, h], Z_hat_test_lasso[, h]):
the standard deviation is zero

[1] "gene_18"
[1] "gene_19"
[1] "gene_20"
[1] "gene_21"
[1] "gene_22"
[1] "gene_23"
[1] "gene_24"
[1] "gene_25"
```

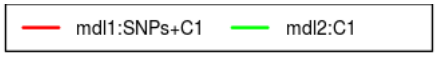
```
[1] "gene_26"
[1] "gene_27"
[1] "gene_28"
[1] "gene_29"
[1] "gene_30"
[1] "gene_31"
[1] "gene_32"
[1] "gene_33"
[1] "gene_34"
```

```
Warning message in cor(test_Z[, h], Z_hat_test_lasso[, h]):
the standard deviation is zero
```

```
[1] "gene_35"
[1] "gene_36"
[1] "gene_37"
[1] "gene_38"
[1] "gene_39"
[1] "gene_40"
```

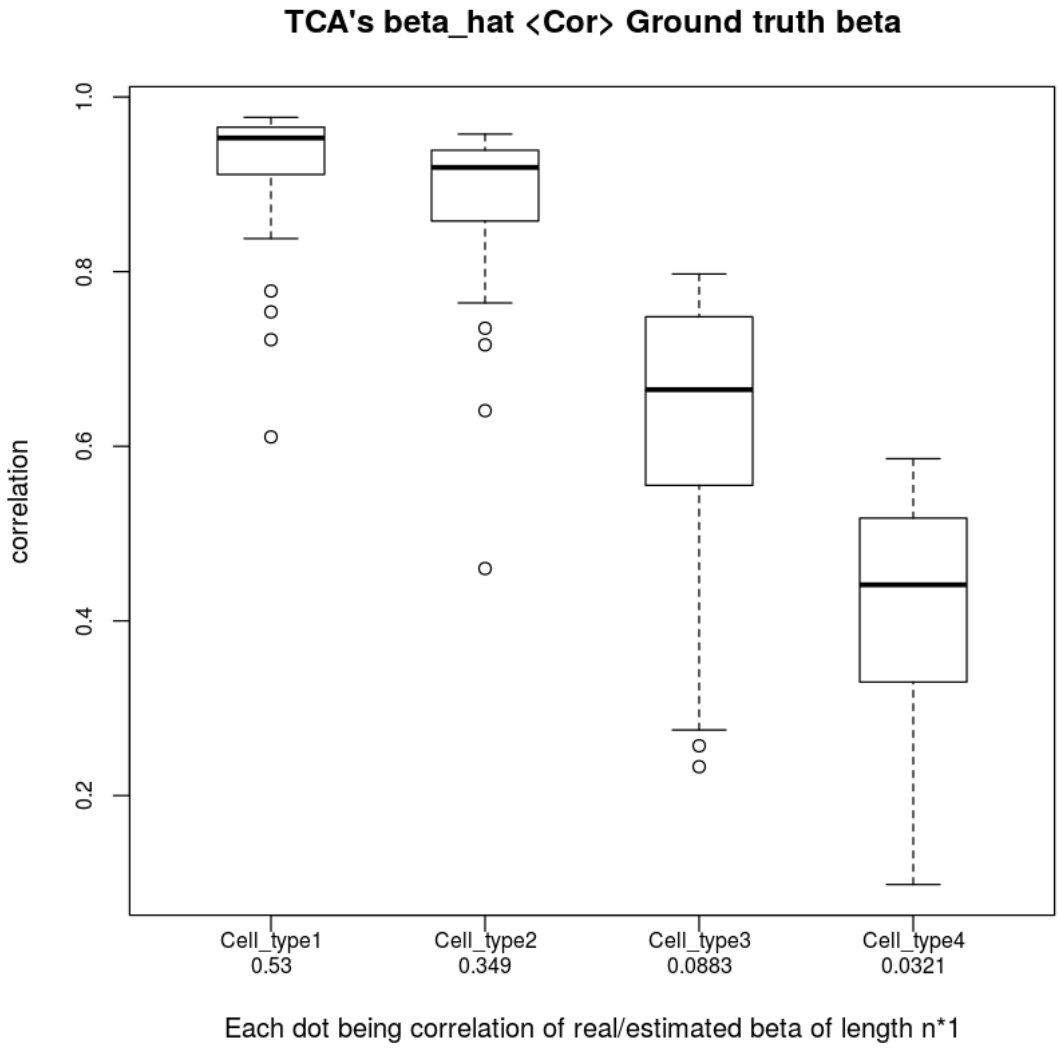
```
In [105]: mean_weight = signif(apply(params$W,2,mean),digits=3)
          Z_hat_cor_tca1 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$Z_hat_cor_
          Z_hat_cor_tca2 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s2$Z_hat_cor_

          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$her[i,],Z_hat_cor_tca1[,i],xlab='heritibility of gene',
                   ylab='cor',cex.lab=1.2,ylim=c(0,1),yaxs='r',type='o',col='red')
              title(main=heading,font.main=2,cex.main=1,line=1,cex.lab=1)
              lines(params$her[i,],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)
```

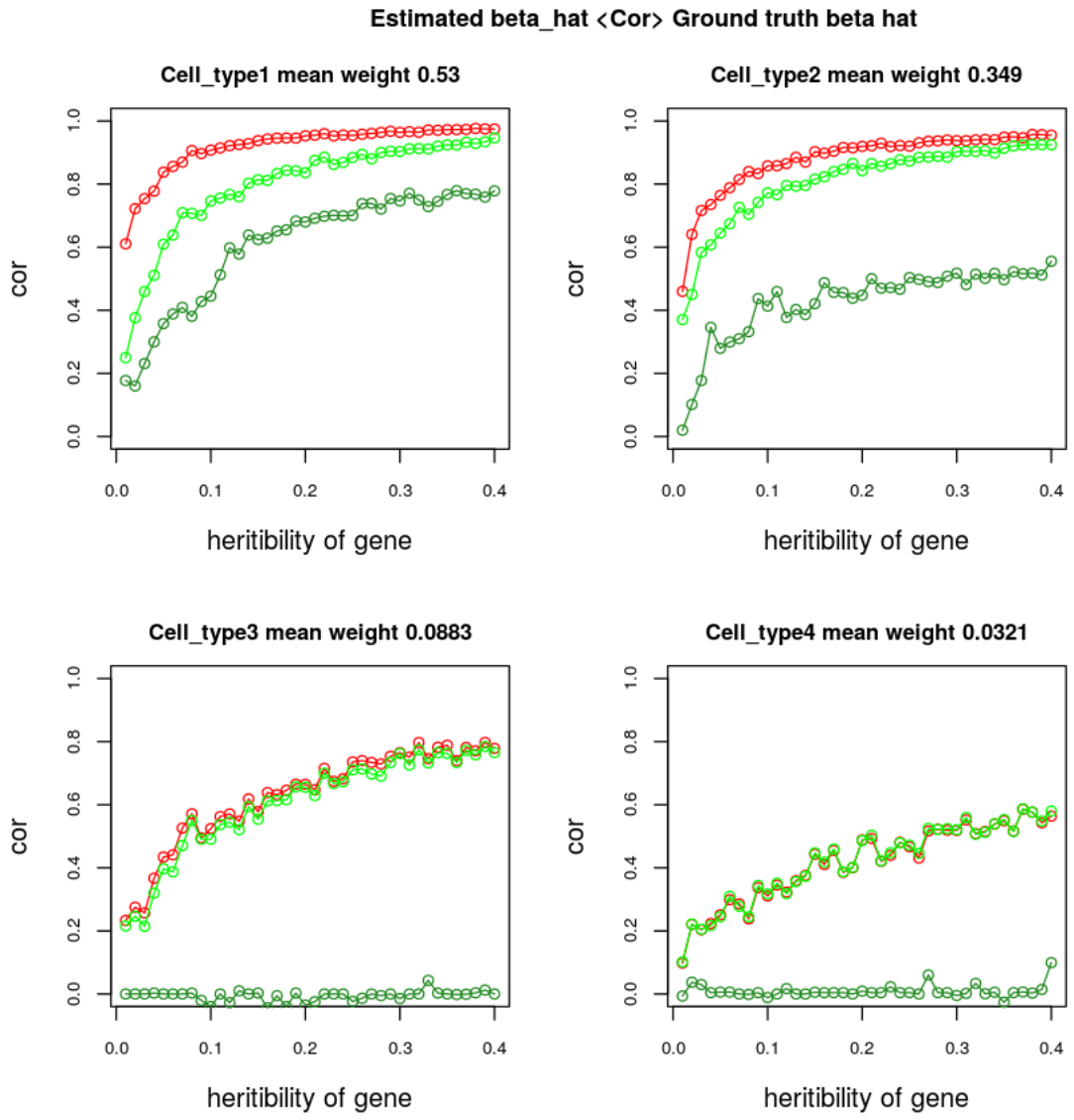

TCA's Z_hat <Cor> Ground truth train Z
Cell_type1 mean weight 0.53
Cell_type2 mean weight 0.349
Cell_type3 mean weight 0.0883
Cell_type4 mean weight 0.0321
cor
heritibility of gene
0.0
0.2
0.4
0.6
0.8
1.0
0.1
0.3

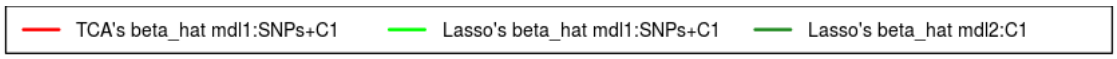


In [106]:
```
beta_hat_cor_tca = do.call(rbind,lapply(1:length(data),function(x) unlist(li[[x]]$s1$b
colnames(beta_hat_cor_tca) = paste(params$cell_name,mean_weight,sep='\n')
cor_beta_lasso_real1 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$cor_
cor_beta_lasso_real2 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$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")
# Heritibility and correlation
par(mfrow=c(2,2),mar=c(4, 3.8, 4, 2) + 0.1)
```

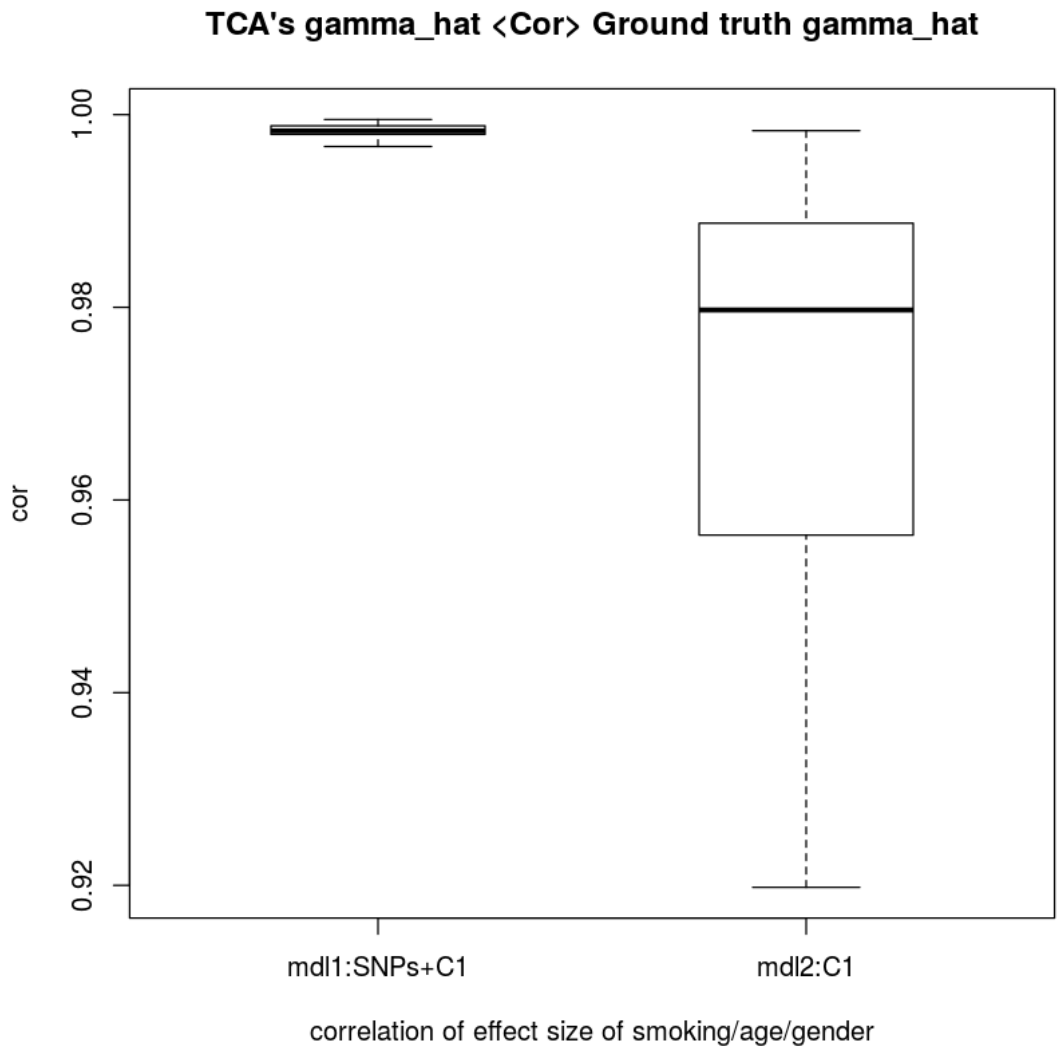
```
opts=params$cell_name
for(i in 1:length(params$cell_name)){
    heading = paste(opts[i],'mean weight',mean_weight[i],sep=' ')
    plot(params$her[i,],beta_hat_cor_tca[,i],xlab='heritibility of gene',
         ylab='cor',cex.lab=1.2,ylim=c(0,1),yaxs='r',type='o',col='red')
    lines(params$her[i,],cor_beta_lasso_real1[,i],type='o',col='green')
    lines(params$her[i,],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,fo
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:S
       col=c('red','green','forestgreen'),lwd=2,cex=0.7,horiz=TRUE)
```




Estimated beta_hat <Cor> Ground truth beta hat
Cell_type1 mean weight 0.53
Cell_type2 mean weight 0.349
Cell_type3 mean weight 0.0883
Cell_type4 mean weight 0.0321
cor
heritibility of gene
0.0
0.2
0.4
0.6
0.8
1.0
0.1
0.3

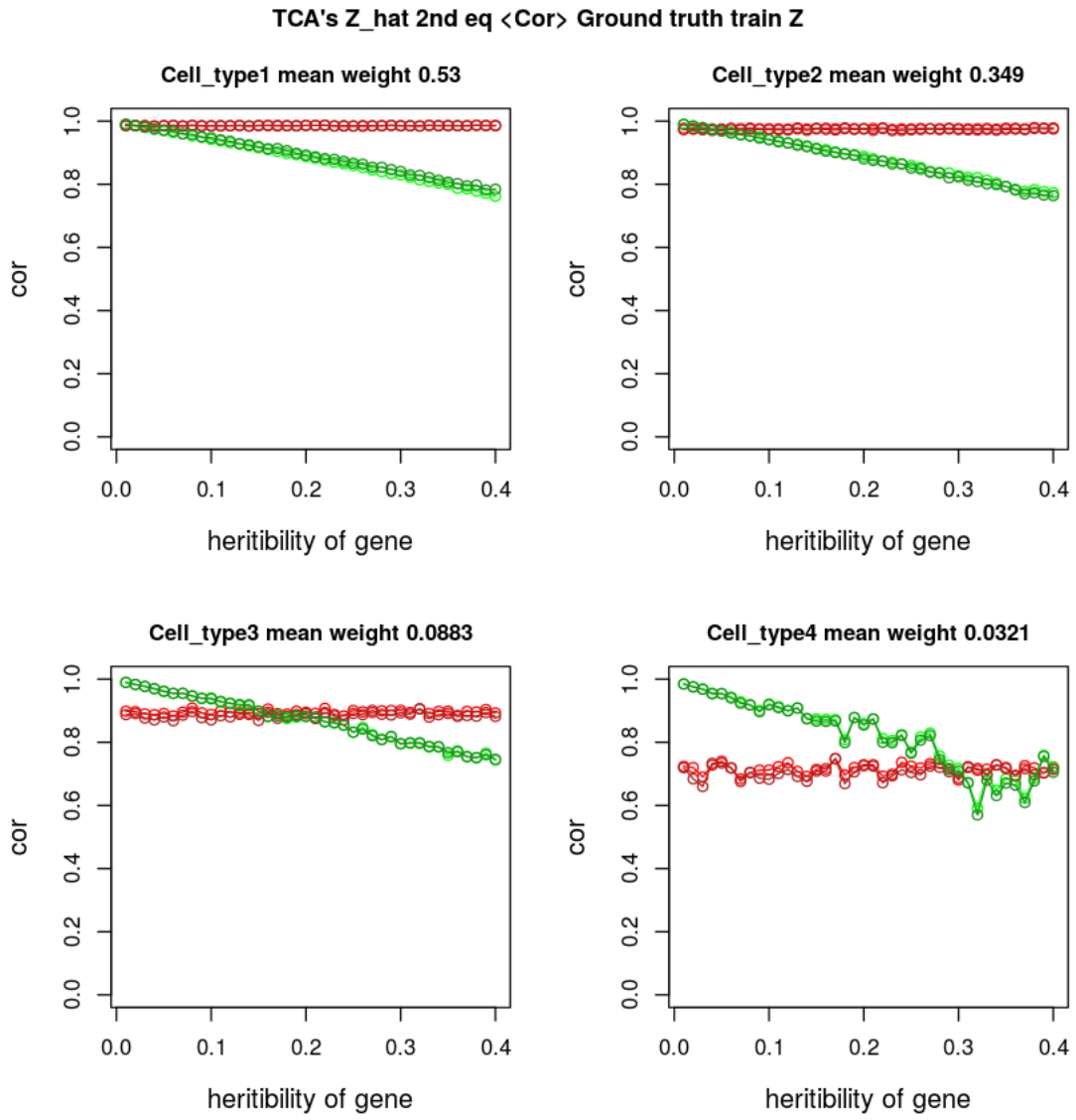


In [107]:
```
gamma_hat_cor_tca1 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$gamma_
gamma_hat_cor_tca2 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s2$gamma_
gamma_cor = cbind(gamma_hat_cor_tca1,gamma_hat_cor_tca2)
colnames(gamma_cor) = c('mdl1:SNPs+C1','mdl2:C1')
boxplot(gamma_cor,main="TCA's gamma_hat <Cor> Ground truth gamma_hat",
        ylab='cor',boxwex=0.5,
        xlab='correlation of effect size of smoking/age/gender')
```

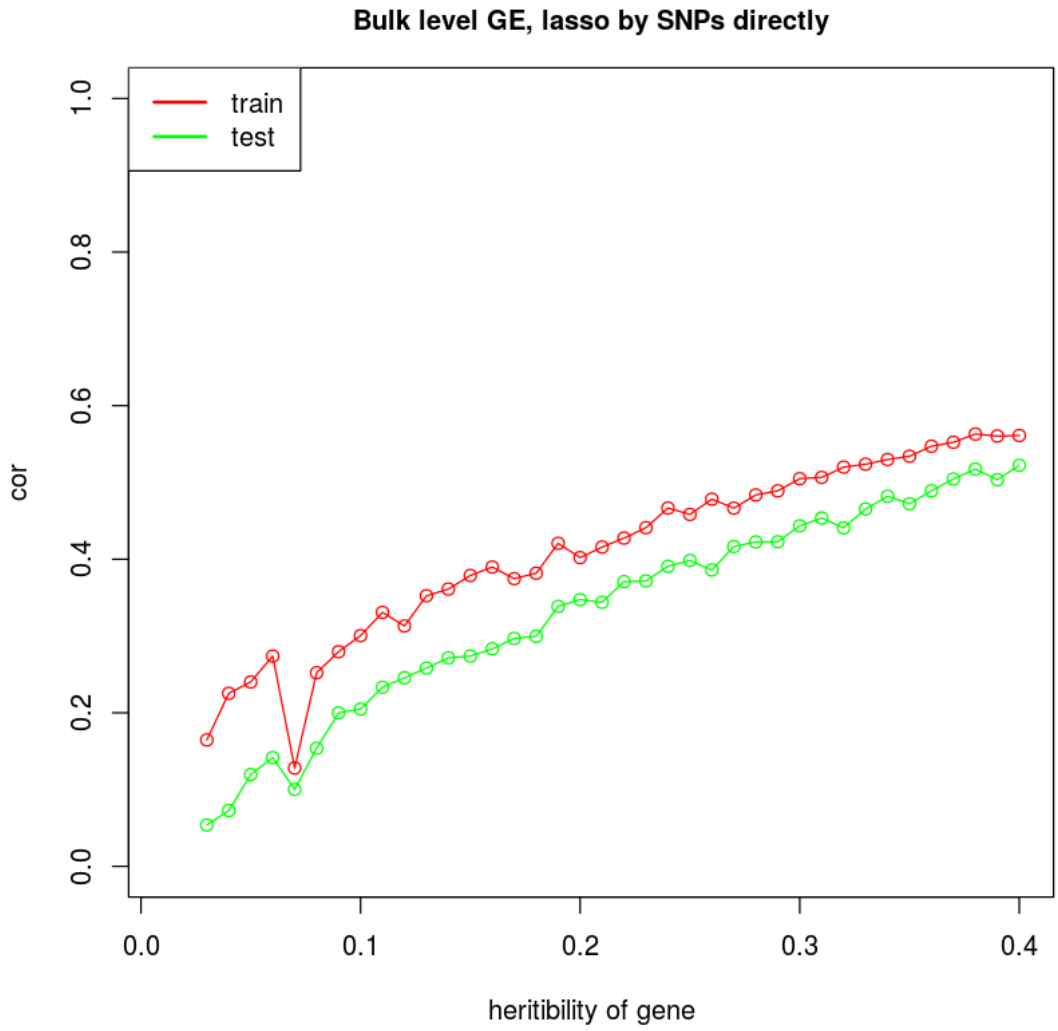


In [108]:
```
Z_hat_train_seq_eq_cor_tca1 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$
Z_hat_train_seq_eq_cor_tca2 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$
Z_hat_test_seq_eq_cor_tca1 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s
Z_hat_test_seq_eq_cor_tca2 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s

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$her[i,],Z_hat_train_seq_eq_cor_tca1[,i],xlab='heritibility of gene',
         ylab='cor',cex.lab=1.2,ylim=c(0,1),yaxs='r',type='o',col='red')
    title(main=heading,font.main=2,cex.main=1,line=1,cex.lab=1)
    lines(params$her[i,],Z_hat_train_seq_eq_cor_tca2[,i],col='green',type='o')
```

```
    lines(params$her[i,],Z_hat_test_seq_eq_cor_tca1[,i],col='firebrick',type='o')
    lines(params$her[i,],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,fon
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+C
       col=c('red','green','firebrick','forestgreen'),lwd=1,cex=0.7,horiz=TRUE)
```

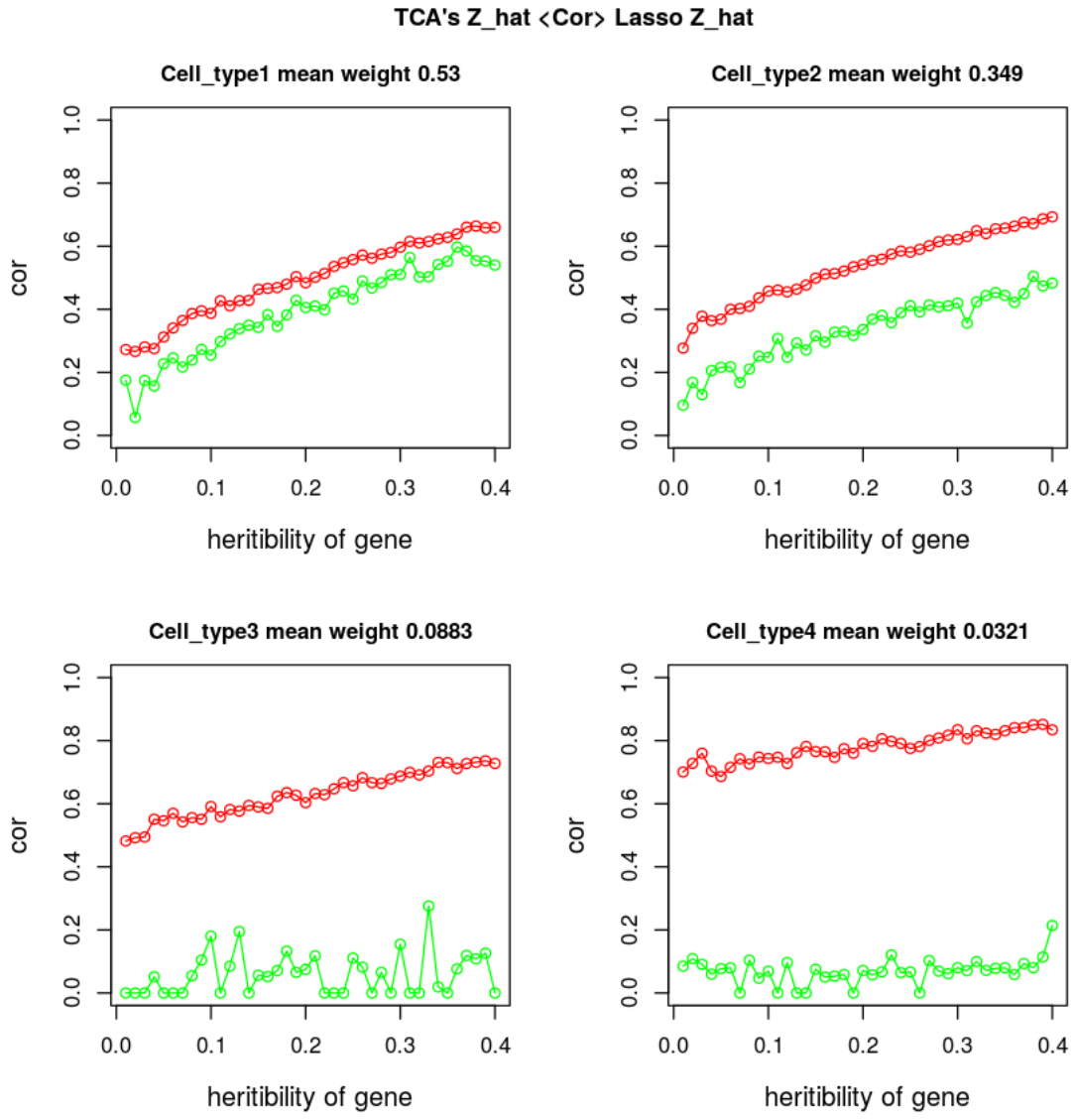


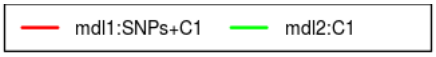
In [109]:
```
G_hat_train_cor_bulk = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$G_ha
G_hat_test_cor_bulk = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$G_hat
plot(params$her[1,],G_hat_train_cor_bulk,type='o',col='red',xlab='heritibility of gene
mtext("Bulk level GE, lasso by SNPs directly",font=2,line=1)
lines(params$her[1,],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 [110]: 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$her[i,],cor_lasso_tca1[,i],xlab='heritibility of gene',
                   ylab='cor',cex.lab=1.2,ylim=c(0,1),yaxs='r',type='o',col='red')
              title(main=heading,font.main=2,cex.main=1,line=1,cex.lab=1)
              lines(params$her[i,],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)
```

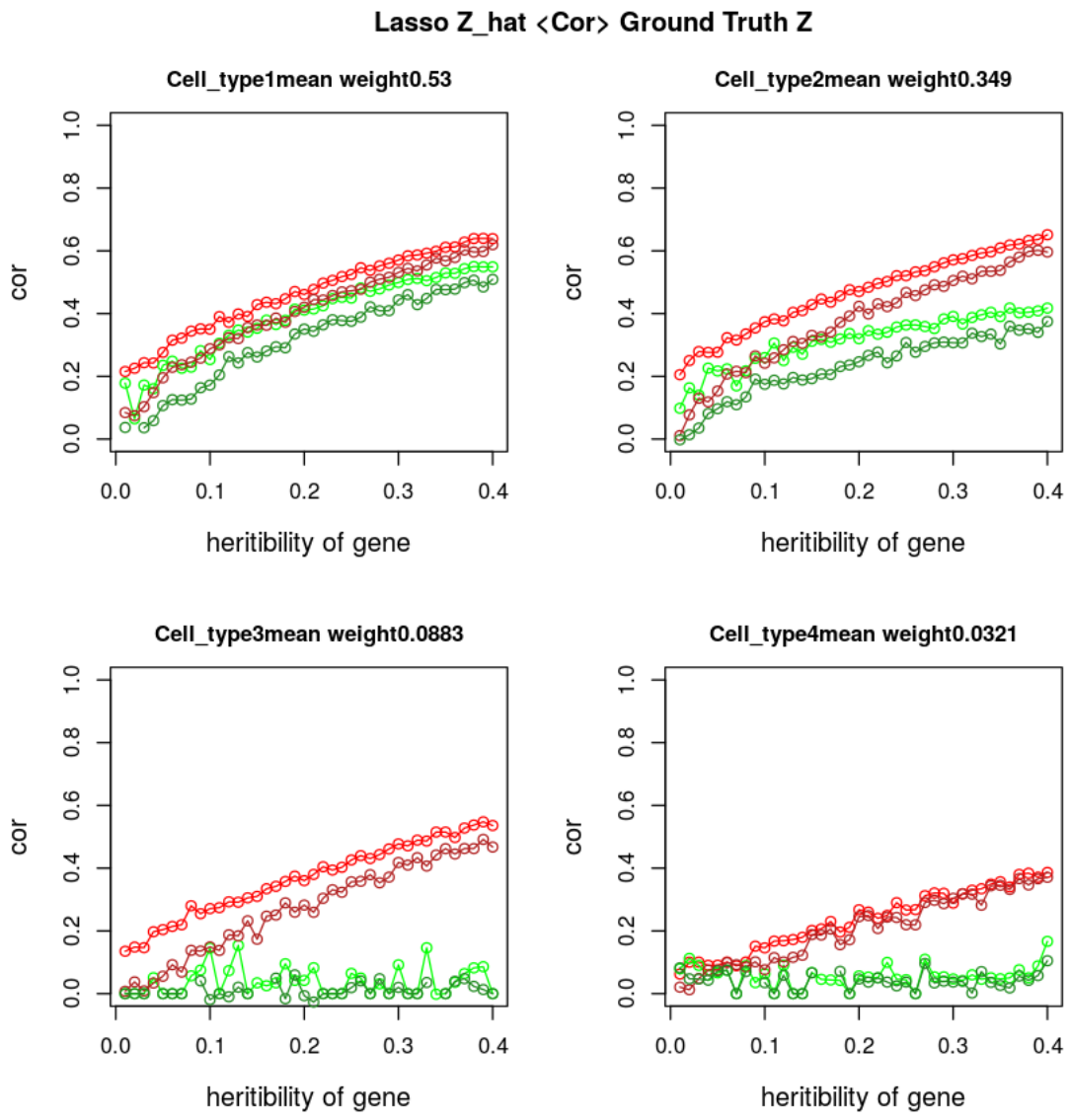
```
par(mfrow=c(2,1),mar=c(0,0,0,0))
plot(1,type='n',axes=FALSE,xlab='',ylab='')
legend(x='top',inset=0,legend=c('mdl1:SNPs+C1','mdl2:C1'),
       col=c('red','green'),lwd=2,cex=0.7,horiz=TRUE)
```





In [111]:
```
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$her[i,],cor_lasso_real_train1[,i],xlab='heritibility of gene',
```

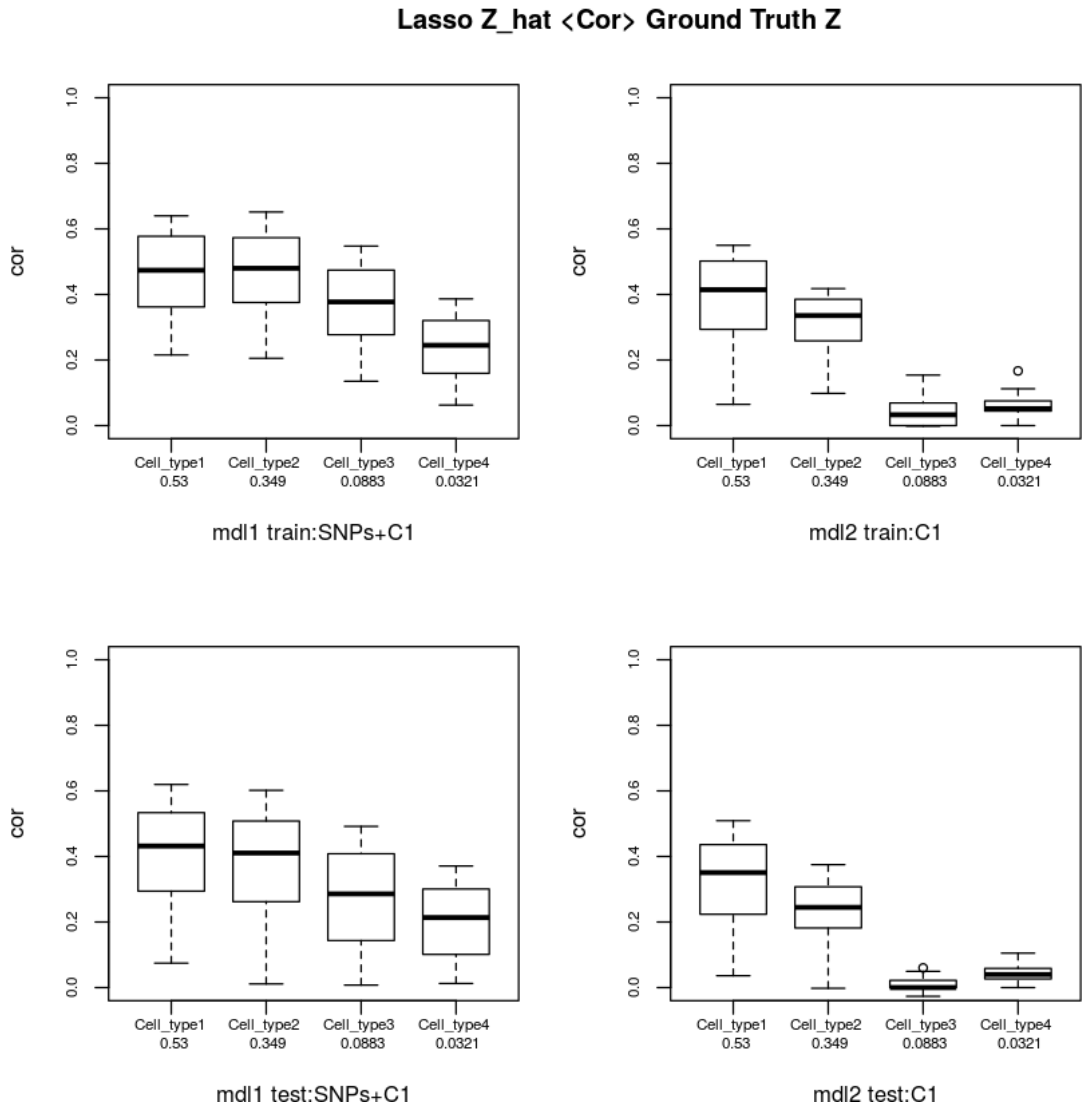
```
        ylab='cor',cex.lab=1.2,ylim=c(0,1),yaxs='r',type='o',col='red')
    title(main=heading,font.main=2,cex.main=1,line=1,cex.lab=1)
    lines(params$her[i,],cor_lasso_real_train2[,i],col='green',type='o')
    lines(params$her[i,],cor_lasso_real_test1[,i],col='firebrick',type='o')
    lines(params$her[i,],cor_lasso_real_test2[,i],col='forestgreen',type='o')
}
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+C
       col=c('red','green','firebrick','forestgreen'),lwd=2,cex=0.7,horiz=TRUE)
```



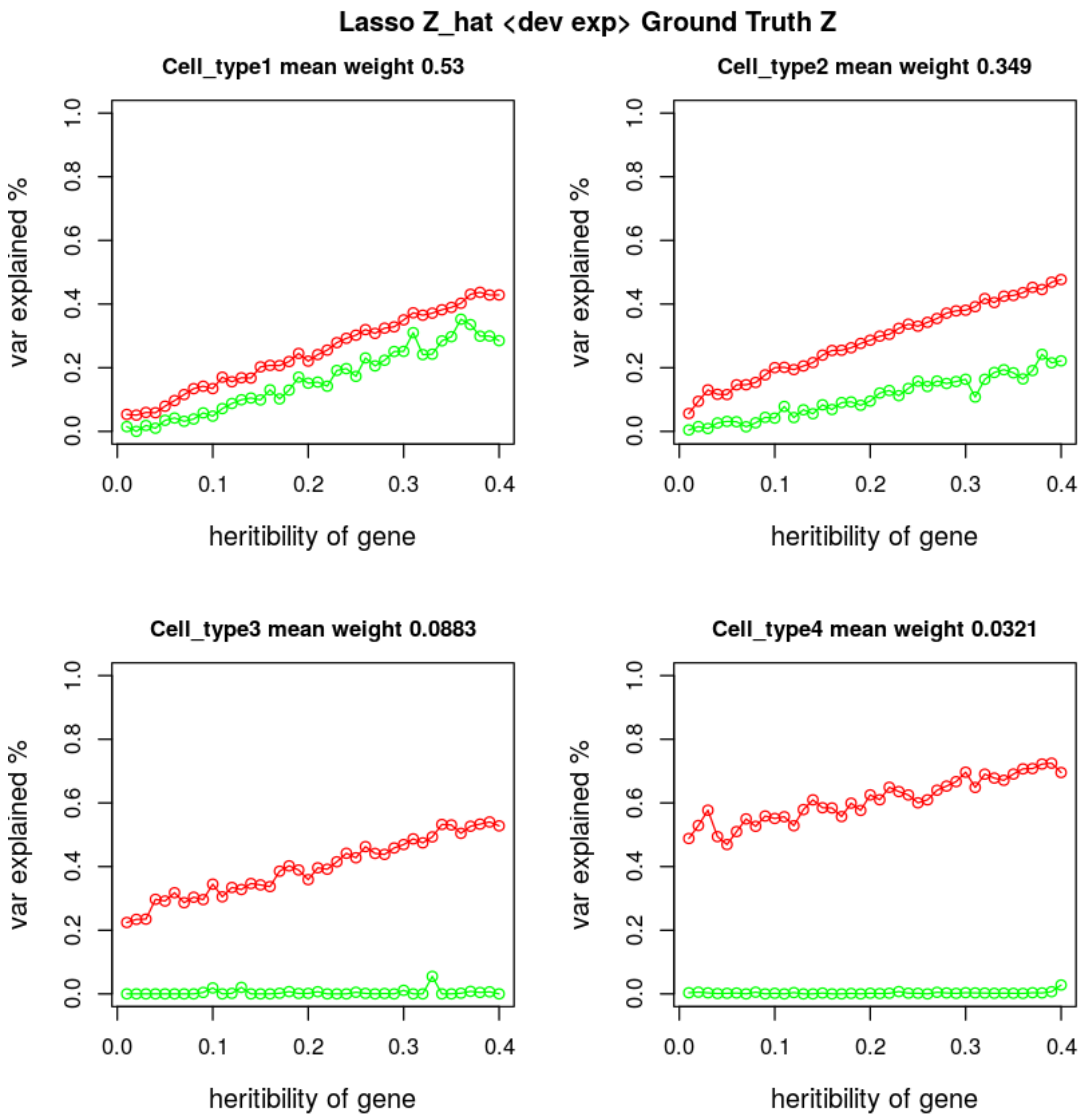
In [98]:
```
# zhat_cor = data.frame(heritability=params$her[i,],cor=cor_lasso_real_test1)
# colnames(zhat_cor) = c('heritability','cor_celltype1','cor_celltype2','cor_celltype3'
# ggplot(zhat_cor,aes(heritability))+geom_line(
#     aes(y=cor_celltype1,color=hue_pal()(4)[1]))+geom_line(
#     aes(y=cor_celltype2,color=hue_pal()(4)[2]))+geom_line(
#     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(0.08,0.35
# # ggsave("/home/elessar/Documents/activities/hackathon/urop/UCLA/csst/tca-twas/pipeli
```

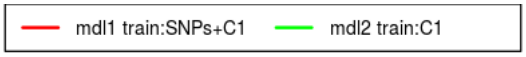
```
# #         plot = last_plot(),dpi = 300, limitsize = TRUE)
```

In [113]:
```
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)
```



In [114]:
```
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$her[i,],lasso_dev_rat1[,i],xlab='heritibility of gene',
         ylab='var explained %',cex.lab=1.2,ylim=c(0,1),yaxs='r',type='o',col='red')
    title(main=heading,font.main=2,cex.main=1,line=1,cex.lab=1)
    lines(params$her[i,],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)
```

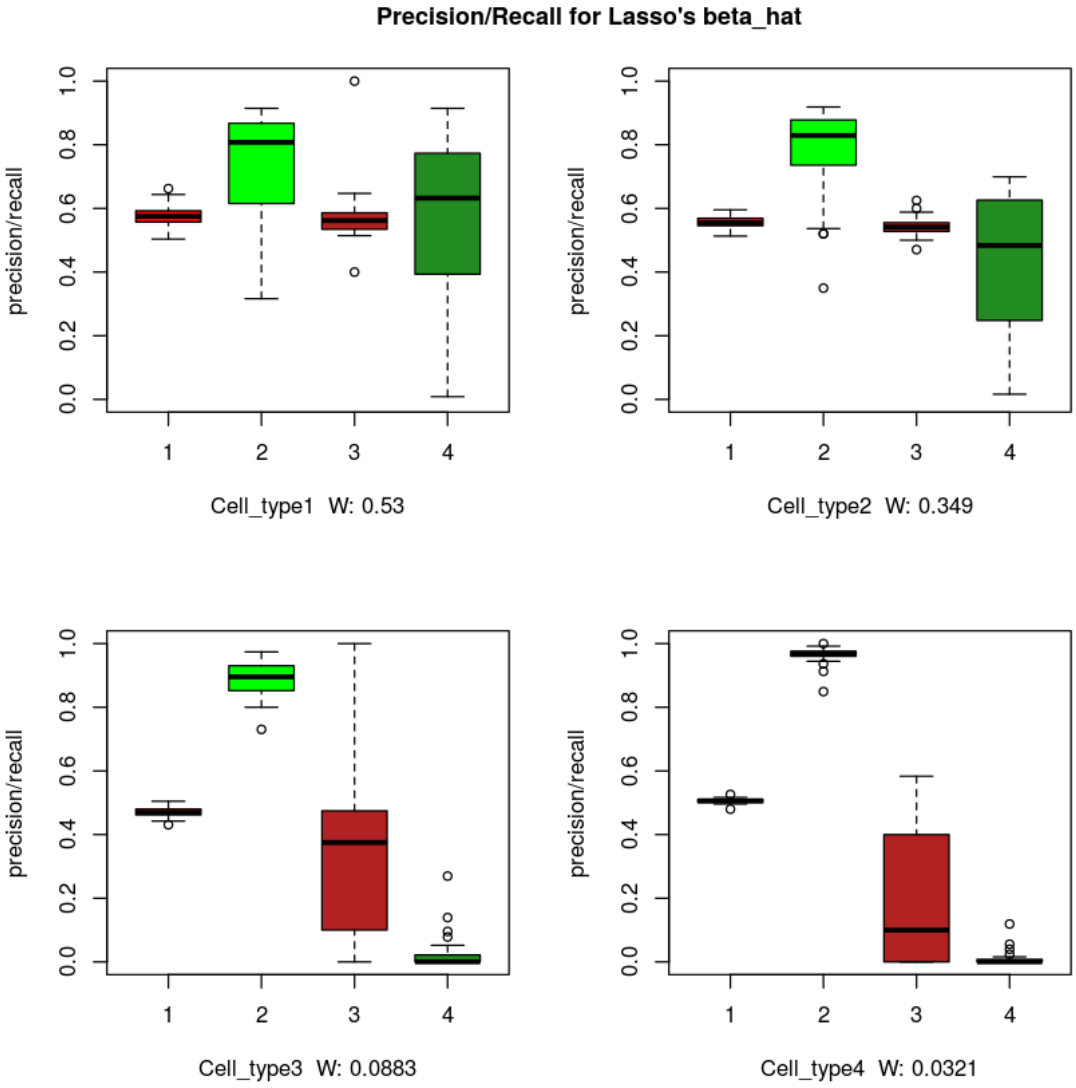

Lasso Z_hat <dev exp> Ground Truth Z
Cell_type1 mean weight 0.53
Cell_type2 mean weight 0.349
Cell_type3 mean weight 0.0883
Cell_type4 mean weight 0.0321
var explained %
heritibility of gene
0.0
0.2
0.4
0.6
0.8
1.0
0.1
0.3



In [101]:
```
# lasso_dev_rat1 = do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$dev_rat)
# colnames(lasso_dev_rat1) = paste(params$cell_name,'\n',mean_weight)
# zhat_dev_exp = data.frame(heritability=params$her[i,],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 Explaned: 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(
```

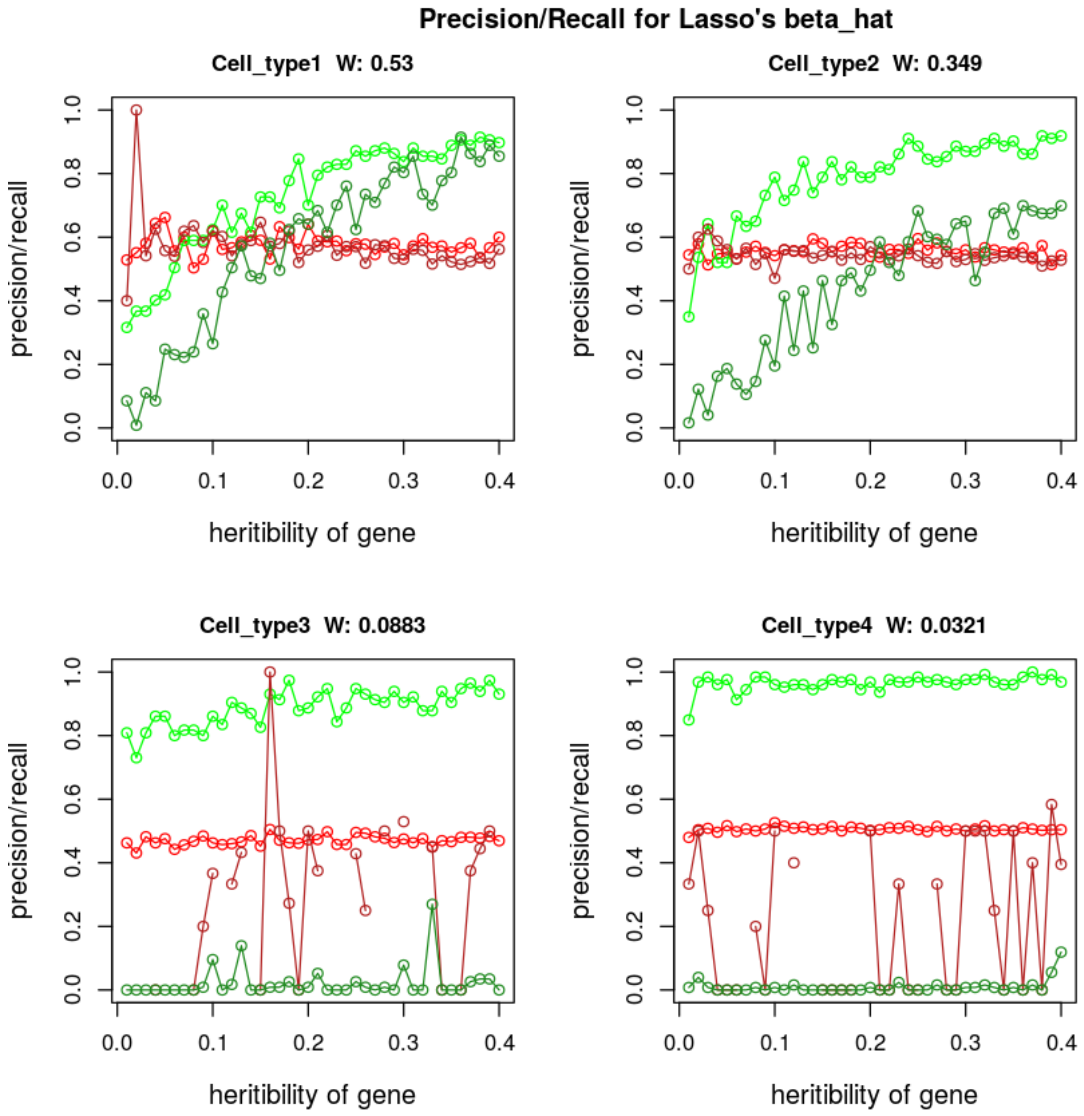
```
#     y="Dev Explained %")+scale_color_discrete(name="Cell Type Weights",labels=c(0.08
# # ggsave("/home/elessar/Documents/activities/hackathon/urop/UCLA/csst/tca-twas/pipel
# #        plot = last_plot(),dpi = 300, limitsize = TRUE)
```

In [117]:
```
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 beta_hat
precision/recall
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
Cell_type1 W: 0.53
Cell_type2 W: 0.349
Cell_type3 W: 0.0883
Cell_type4 W: 0.0321

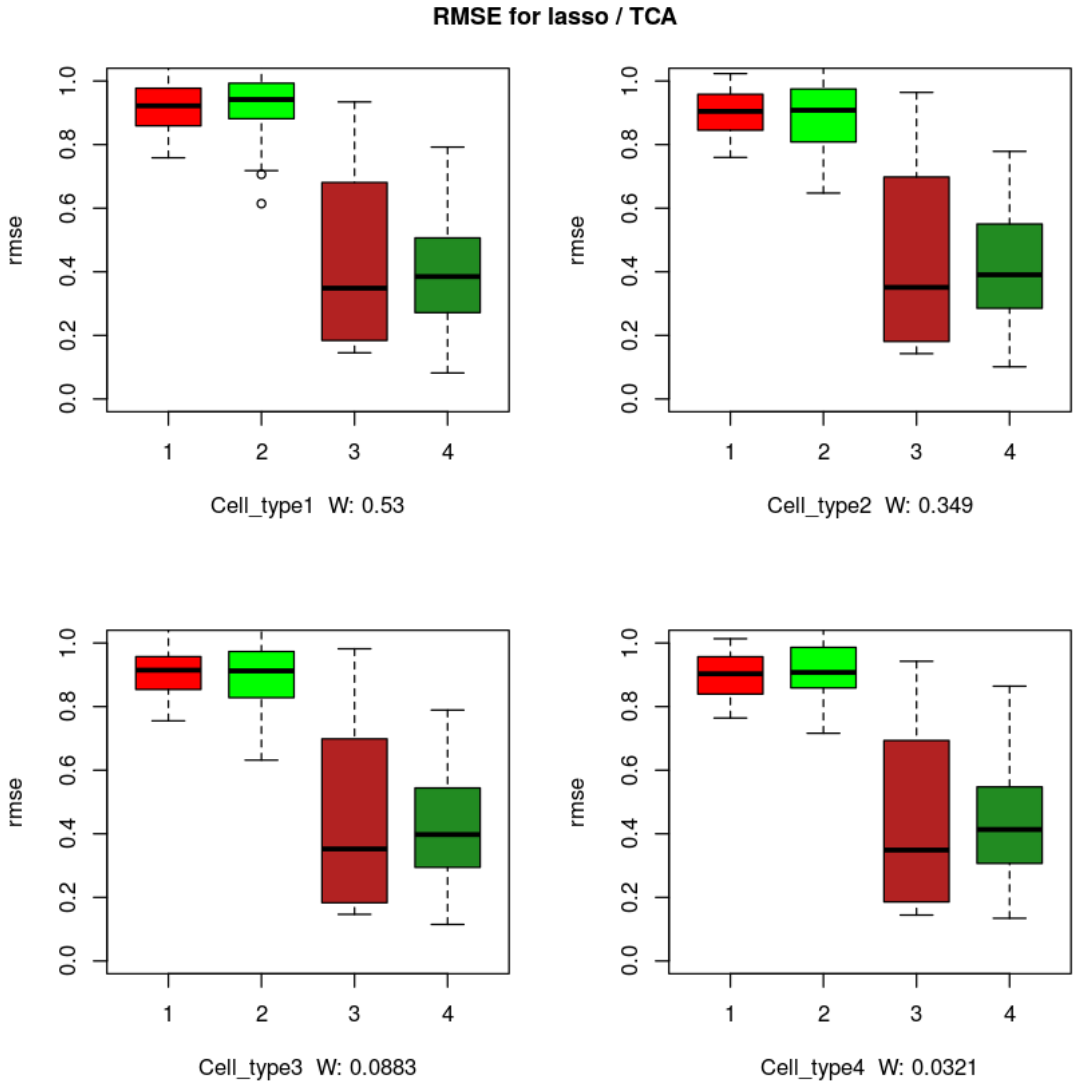
In [119]:
```
par(mfrow=c(2,2),mar=c(4, 3.8, 4, 2) + 0.1)
for(i in 1:params$K){
    plot(params$her[i,],prec_rec_frame[[i]][,1],xlab='heritibility of gene',
         ylab='precision/recall',cex.lab=1.2,ylim=c(0,1),yaxs='r',type='o',col='red')
    title(main=opts[i],font.main=2,cex.main=1,line=1,cex.lab=1)
    lines(params$her[i,],prec_rec_frame[[i]][,2],col='green',type='o')
    lines(params$her[i,],prec_rec_frame[[i]][,3],col='firebrick',type='o')
    lines(params$her[i,],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 [120]:
```
rmse_lasso1 =matrix(do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$rmse_la
rmse_lasso2 = matrix(do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s2$rmse_l
rmse_tca1 =matrix(do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s1$rmse_tca)
rmse_tca2 = matrix(do.call(rbind,lapply(1:length(data),function(x) li[[x]]$s2$rmse_tca
rmse_frame = lapply(1:params$K,function(x) t(rbind(rmse_lasso1[x,],rmse_lasso2[x,],rms

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','forestgre
}
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)
```




lasso mdl1:SNPs+C1
lasso mdl2:C1
tca mdl1:SNPs+C1
tca mdl2:C1