ICeDT

Immune Cell Deconvolution in Tumor tissues

Update for version 0.99.1, from 2019/02/25 to 2019/03/04

Nomenclature

- C: variables for Consistent genes, CM or M was used in some cases.
- A: variables for Abberant genes
- Use lower case at the begining of the variable name. For example Sigma2C is changed to sigma2C.
- Y: observed expression data from bulk (mixed) samples
- X: observed expression data from purified samples, assume at least two samples per cell type.
- Z: estimated cell type-specific expression for each cell type. Previously there were notations such Z_0 and Z_1 and they are the same. Now the notations are unified to be Z.

Simulation codes

1. Move the code Simulation_Machinery.R into the R package as function simFunc.
2. Rename "Simulation_Machinery.RData" to "mean_var_relation.RData" and move it into the R package.
3. Remove functions weight_creation, meanFun, and IQRFun.
4. Modity the function to set muX of tumor to be -20. Prevoiusly muX was set to be the same value as other cell types, and then the simulated gene expression from tumor was set to 0. This has two consequence

- the ouptut of `tumor_mu` is wrong since it is muX
- for any abberant genes, the gene expression from tumor is not 0.

ICeDT algorithm

1. Combine the codes in *_initFit.R, *_UpdateFunctions.R, and *_FittingAlgorithm.R into one file.
2. Combine HS2_UpdateWgts_All_* and HS2_UpdateWgts_Single_*, to write HS2_UpdateWgts_Single_* within a for loop. Prevoiusly apply was used. For a complex function like that, apply will not be faster than for loop.
3. Change function names or parameter names

- change function name ICeDT_fit_noWgt_noRef to ICeDT_noWgt_noRef, remove parameter Subj_CO, and change parameter RhoConv_CO to rhoConverge.
- HS2_UpdateWgts_* to updateWgts_
- HS_* to HS
- p_m or Pm or to propC, which is the mixture proportio of consistent genes.

4. Z_star was only used in function HS_GradFunc_Fix, but it is one of the parameters for several other functions of gradient and liklihood. Remove it as a parameter and calculate within the function of HS_GradFunc_Fix

Strcuture of function ICeDT_noWgt_noRef

1. Check input
2. Given observed cell type-specific gene expression of multiple samples per cell type, calculate cell type-specific expression per cell type.
3. Call `PropPlus_Update` to update rho, sigma2C, sigma2A, and propC.

- Iteratively update weights (each gene's probability being consistent) using function `updateWgts` (E-step) and estimate three parameters: cell type proportions, sigma2c, and sigma2A using function `updatePropn_All` (M-step).
  - `updatePropn_Single` iteratively estimate sigma2c, sigma2A, and cell type compositions. The latter were estimated using Augmented Lagrangian Minimization Algorithm implmented in function `alabama/auglag`.

4. Update weights one more time

Handeling the special cell type (fixed cell type) of tumor cells.

1. When estimating cell type-specific gene expression, force gene expression from tumor cells to be 0. No matter what is the observed gene expression from tumor cells.

```
Z     = exp(CT_MU + CT_var/2)
Z[,1] = rep(0, nG)
```

2. `PropPlus_Update` takes initial estimates of rho as relative proportions from all the other cell types other than tumor cells.
3. In function `updateWgts`, add 0 as the proportion from tumor to the rho vector.

```
updateWgts <- function(logY, rho_init, sigma2C, sigma2A, Z, propC){

  EM_wgt = matrix(NA, nrow=nrow(logY), ncol=ncol(logY))

  for(i in 1:ncol(logY)){

    logY_i     = logY[,i]
    rho_init_i = rho_init[,i]

    #-----------------------------------------------#
    # Cmu_ij for Consistent Marker Gene Probs      #
    # Amu_ij for Aberrant Marker Gene Probs        #
    #-----------------------------------------------#

    eta_ij = Z %*% matrix(c(0, rho_init_i), ncol=1)
    Cmu_ij = log(eta_ij) - sigma2C[i]/2
    Amu_ij = log(eta_ij) - sigma2A[i]/2
    C_lLik = dnorm(logY_i, mean = Cmu_ij, sd = sqrt(sigma2C[i]), log = TRUE)
    A_lLik = dnorm(logY_i, mean = Amu_ij, sd = sqrt(sigma2A[i]), log = TRUE)

    #----------------------------------#
    # Compiling Weights                #
    #----------------------------------#
    EM_wgt[,i] = 1/(1+((1-propC[i])/propC[i])*exp(A_lLik-C_lLik))
  }
```

```
  return(EM_wgt)
}
```

4. In function `updatePropn_Single`, always calculate expected expression in bulk tumor by assuming proportion from tumor is 0.

```
  eta_ij     = drop(Z %*% matrix(c(0,urho_1), ncol=1))
  log_eta_ij = log(eta_ij)

  sigma2C_1  = sigma2_Update(logY = logY, log_eta_ij = log_eta_ij,
                             EM_wgt = EM_wgt, AB_Up = FALSE)
  sigma2A_1  = sigma2_Update(logY = logY, log_eta_ij = log_eta_ij,
                            EM_wgt = EM_wgt, AB_Up = TRUE)

  mu_ijC = log_eta_ij - sigma2C_1/2
  mu_ijA = log_eta_ij - sigma2A_1/2

  logLik_1 = sum(EM_wgt*dnorm(logY,mean=mu_ijC,sd=sqrt(sigma2C_1),log=TRUE))+
    sum((1-EM_wgt)*dnorm(logY,mean=mu_ijA,sd=sqrt(sigma2A_1),log=TRUE))
```

When estimating rho and `useRho` is TRUE, first estimate rho given proportion of tumor cells, and the normalize it so that its summation is 1.

```
    if(useRho){
      auglagOut = auglag(par = urho_0[-c(nCell)], fn = logLik_Fix,
                         gr = gradFunc_Fix, hin = hin_func_Fix,
                         hin.jac = hin_jacob_Fix, logY = logY,
                         rho_i0 = rho_i0, Z=Z, sigma2C = sigma2C_0,
                         sigma2A = sigma2A_0, EM_wgt = EM_wgt,
                         control.optim = list(fnscale=-1),
                         control.outer = list(trace=FALSE))

      urho_1[c(1:(nCell-1))] = auglagOut$par
      urho_1[nCell] = 1-rho_i0-sum(auglagOut$par)

      urho_1 = correctRho(est = urho_1, total = 1-rho_i0)
    }
```

The objective function for optimation do consider the gene expression from tumor cells. This is the only place where gene expressoin from tumor cells (1st column of Z) is used.

```
logLik_Fix <- function(x, logY, rho_i0, Z, sigma2C, sigma2A, EM_wgt){
  rho  = c(0, x, 1-rho_i0-sum(x))

  eta_ij = Z %*% rho
  mu_ijC = log(eta_ij) - sigma2C/2
  mu_ijA = log(eta_ij) - sigma2A/2

  out = sum(EM_wgt*dnorm(logY, mean = mu_ijC, sd = sqrt(sigma2C), log = TRUE)) +
    sum((1-EM_wgt)*dnorm(logY, mean = mu_ijA, sd = sqrt(sigma2A), log = TRUE))

  return(out)
}
```

However, since in function `ICeDT_noWgt_noRef`, `Z[,1]` is initilized by 0,

To be fixed.

1. function `alabama/auglag` gives warning message when choosing parameters outside the constraints, for example, setting rho to be larger than 1.