haoharryfeng / **ISLET**

Type / to search

Code Issues Pull requests Actions Projects Security Insights

ISLET / R / dataprep.R

haoharryfeng 0.99.6 8b8e137 · 2 years ago History

Code Blame 87 lines (68 loc) · 3.43 KB Raw

```r

###function to read-in and check data from case and control: observed expression, proportion, and sample-to-subject relationship

dataPrep<-function(dat_se){
  message("Begin: working on data preparation as the input for ISLET algorithm.")
#  if (missing(case_dat_se) || missing(ctrl_dat_se))
#    stop("SummarizedExperiment objects from both groups are needed.")
#  if (length(intersect(colData(case_dat_se)[,1], colData(ctrl_dat_se)[,1]))>0)
#    stop("Subject IDs across case group and control group must be unique.")
#  if (ncol(colData(case_dat_se))!=ncol(colData(ctrl_dat_se)))
#    stop("Case group and control group must have the same number of cell types.")

  #subject id between cases and ctrls should also be unique, check and implement this later
  #check for negative values, implement later

    if (!is(dat_se, "SummarizedExperiment"))
        stop("The input dataset must be a SummarizedExperiment object.")
    if (length(unique(colData(dat_se)$group)) != 2)
        stop("There must be two groups (case/ctrl) in the input SummarizedExperiment object.")
    if (unique(colData(dat_se)$group)[1] != "case" || unique(colData(dat_se)$group)[2] != "ctrl")
        stop("The names for the two groups in comparison should be labeled as
                 `case` and `ctrl` in the input SummarizedExperiment object.")


  #separate cases and controls
    idx <- which(colData(dat_se)$group == "case")

    case_dat_se <- SummarizedExperiment(assays=list(counts=assays(dat_se)$counts[, idx]),
                                        colData=colData(dat_se)[idx, -1])
    ctrl_dat_se <- SummarizedExperiment(assays=list(counts=assays(dat_se)$counts[, -idx]),
                                        colData=colData(dat_se)[-idx, -1])



  #K = number of cell types
  K <- ncol(colData(case_dat_se))-1

  #N1 = number of samples for group 1
  N1 <- ncol(assays(case_dat_se)$counts)
  #N1 = number of samples for group 2
  N2 <- ncol(assays(ctrl_dat_se)$counts)
  #NS = total number of Samples for group 1&2
  NS <- N1 + N2
  #NU = total number of Unique subjects for group 1&2
  caseUN <- length(unique(colData(case_dat_se)[, 1]))
  ctrlUN <- length(unique(colData(ctrl_dat_se)[, 1]))
  NU <- caseUN + ctrlUN



  X_sub1 <- as.matrix(rbind(colData(case_dat_se)[, -1], colData(ctrl_dat_se)[, -1]))
  X_sub2 <- rbind(matrix(1,  nrow=N1, ncol=K), matrix(0, nrow=N2, ncol=K))*X_sub1
#  X_sub2 = X_sub2[,1:para]
  X_0 <- cbind(X_sub1, X_sub2)
  X_list <- lapply(1, function(x){return(X_0)})
  X <- bdiag(X_list)
```

Subject_ID	CT1	CT2	…
sample_ID 1	1		
sample_ID 2	1		
…			

N_1: # of case

N_2: # of ctrl

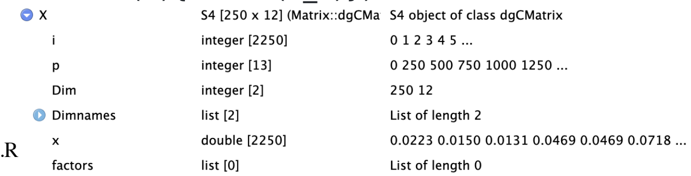
$$\begin{array}{c|ccc} & \text{CT1} & \text{CT2} & \cdots \\ \hline \text{sample1} & & & \\ \text{sample2} & & & \\ \vdots & & & \\ \text{sample}N & & & \end{array}$$

X_sub1 : proportion ∈ (0,1) $(N_1+N_2)\times K$

X_sub2: $\left[\quad // \quad\right]$ } 前 N_1 行和 X_sub1 相同 } 后 N_2 行全是 0 ∈ [0,1) $(N_1+N_2)\times K$

$$\begin{array}{r} \\ N_1\text{行} \\ \\ N_2\text{行} \\ \\ \end{array} \overbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & \cdots & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}}^{K\text{列}}$$

元素逐个对应相乘



```
X           S4 [250 x 12] (Matrix::dgCMa  S4 object of class dgCMatrix
  i         integer [2250]                0 1 2 3 4 5 ...
  p         integer [13]                  0 250 500 750 1000 1250 ...
  Dim       integer [2]                   250 12
  Dimnames  list [2]                      List of length 2
  x         double [2250]                 0.0223 0.0150 0.0131 0.0469 0.0469 0.0718 ...
  factors   list [0]                      List of length 0
```

X_0:

$$X = \begin{pmatrix} \theta_{111} & \theta_{112} & \dots & \theta_{11K} & z_1\theta_{111} & z_1\theta_{112} & \dots & z_1\theta_{11K} \\ \theta_{121} & \theta_{122} & \dots & \theta_{12K} & z_1\theta_{121} & z_1\theta_{122} & \dots & z_1\theta_{12K} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \theta_{1T_11} & \theta_{1T_12} & \dots & \theta_{1T_1K} & z_1\theta_{1T_11} & z_1\theta_{1T_12} & \dots & z_1\theta_{1T_1K} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \theta_{J11} & \theta_{J12} & \dots & \theta_{J1K} & z_J\theta_{J11} & z_J\theta_{J12} & \dots & z_J\theta_{J1K} \\ \theta_{J21} & \theta_{J22} & \dots & \theta_{J2K} & z_J\theta_{J21} & z_J\theta_{J22} & \dots & z_J\theta_{J2K} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \theta_{JT_J1} & \theta_{JT_J2} & \dots & \theta_{JT_JK} & z_J\theta_{JT_J1} & z_J\theta_{JT_J2} & \dots & z_J\theta_{JT_JK} \end{pmatrix}_{N\times 2K}$$

$N = NS$ (# of sample)

```r
  #obtain a vector of unique subject IDs, for all, to use later
  sub_id <- c(colData(case_dat_se)[, 1], colData(ctrl_dat_se)[, 1])

  propm <- as.matrix(rbind(colData(case_dat_se)[, -1], colData(ctrl_dat_se)[, -1]))
 # propd = apply(propm, MARGIN = 2, makea, sub_id = sub_id, X = X, NU = NU, simplify = F)
  propd <- apply(X=propm, MARGIN=2, FUN=makea,
                 ind_id=sub_id, datX=X, aNU=NU, simplify=FALSE)

  A_0 <- do.call(cbind, propd)
  #A_list=lapply(1,function(x){return(A_0)})
  A<-bdiag(A_0)

  CT<-colnames(propm)

  datuse <- inputSet(exp_case=assays(case_dat_se)$counts,
                     exp_ctrl=assays(ctrl_dat_se)$counts,
                     X=X,
                     A=A,
                     K=K,
                     NS=NS, # of sample
                     NU=NU, # of subject
                     case_num=caseUN,
                     ctrl_num=ctrlUN,
                     CT=CT,
                     SubjectID=sub_id,
                     type='intercept'
                     )
  message("Complete: data preparation for ISLET.")
  return(datuse)
}
```

对列

$$\boldsymbol{a}_{11} = \begin{bmatrix} \theta_{111} \\ \theta_{121} \\ \vdots \\ \theta_{1T_11} \end{bmatrix}$$

第1个subject 在第1个cell type 的different proportion across time

N = NS = NU × T

$$\boldsymbol{A} = \begin{pmatrix} \mathbf{a}_{11} & 0 & 0 & 0 & \dots & \mathbf{a}_{1K} & 0 & 0 & 0 \\ 0 & \mathbf{a}_{21} & 0 & 0 & \dots & 0 & \mathbf{a}_{2K} & 0 & 0 \\ 0 & 0 & \ddots & 0 & \dots & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \mathbf{a}_{J1} & \dots & 0 & 0 & 0 & \mathbf{a}_{JK} \end{pmatrix}_{N\times Q} \qquad (5)$$

Q = NU × K

where $\mathbf{a}_{jk} := (\theta_{j_1k}, \theta_{j_2k}, \cdots, \theta_{jT_jk})'$ is simply a reorganized vector of cell type proportions, to align with random effect $\boldsymbol{u}$.

$u \in \mathbb{R}^{Q\times 1}$

Q = 50×6 = 300 in our case

NS = 250

NU = 50

T = 5

makea:

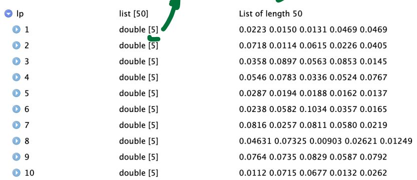
```r
###function to make the design matrix [A] for random effect
#updated on 05/31/2022 to reflect the change in ID order
#user should sort their data by subject ID
#makea <- function(onectprop, ind_id = sub_id, datX = X, aNU = NU){
makea <- function(onectprop, ind_id, datX , aNU){
  lp <- split(onectprop, ind_id)
  a1 <- matrix(0, nrow=nrow(datX), ncol=aNU)
#  ct_sub=table(sub_id)[as.character(unique(sub_id))]
#  lp=lp[names(ct_sub)]
  chk <- unique(ind_id) #chk should have the length of NU
  lp<-lp[as.character(chk)]
  count <- rep(0, length(chk))
  for(i in seq_len(aNU)){
    tmp <- sum(ind_id == chk[i])
    count[i] <- tmp
  }

  for(i in seq_len(aNU)){
    s <- 1+sum(count[0:(i-1)])
    e <- sum(count[seq_len(i)])
    a1[s:e, i] <- lp[[i]]
  }
  return(a1)
```

one cell type proportion

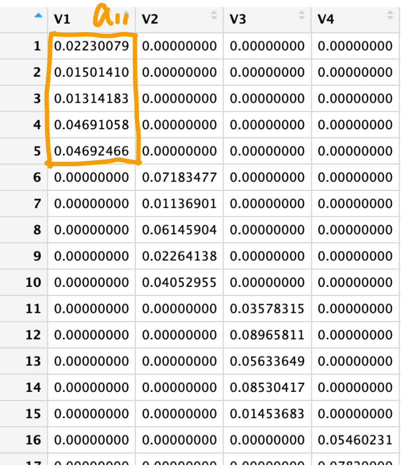
$\rightarrow 0^{NS\times NU}$

5个时间点



```
lp        list [50]     List of length 50
  1       double [5]    0.0223 0.0150 0.0131 0.0469 0.0469
  2       double [5]    0.0718 0.0114 0.0615 0.0226 0.0405
  3       double [5]    0.0358 0.0897 0.0563 0.0853 0.0145
  4       double [5]    0.0546 0.0783 0.0336 0.0524 0.0767
  5       double [5]    0.0287 0.0194 0.0188 0.0162 0.0137
  6       double [5]    0.0238 0.0582 0.1034 0.0357 0.0165
  7       double [5]    0.0816 0.0257 0.0811 0.0580 0.0219
  8       double [5]    0.04631 0.07325 0.00903 0.02621 0.01249
  9       double [5]    0.0764 0.0735 0.0829 0.0587 0.0792
  10      double [5]    0.0112 0.0715 0.0677 0.0132 0.0262
```

k=1 (第一个cell type)



	V1	V2	V3	V4
1	0.02230079	0.00000000	0.00000000	0.00000000
2	0.01501410	0.00000000	0.00000000	0.00000000
3	0.01314183	0.00000000	0.00000000	0.00000000
4	0.04691058	0.00000000	0.00000000	0.00000000
5	0.04692466	0.00000000	0.00000000	0.00000000
6	0.00000000	0.07183477	0.00000000	0.00000000
7	0.00000000	0.01136901	0.00000000	0.00000000
8	0.00000000	0.06145904	0.00000000	0.00000000
9	0.00000000	0.02264138	0.00000000	0.00000000
10	0.00000000	0.04052955	0.00000000	0.00000000
11	0.00000000	0.00000000	0.03578315	0.00000000
12	0.00000000	0.00000000	0.08965811	0.00000000
13	0.00000000	0.00000000	0.05633649	0.00000000
14	0.00000000	0.00000000	0.08530417	0.00000000
15	0.00000000	0.00000000	0.01453683	0.00000000
16	0.00000000	0.00000000	0.00000000	0.05460231
17	0.00000000	0.00000000	0.00000000	[illegible]

a_{11} … V50

a_2

a_{12}

Gene1

$$y = \begin{bmatrix} 1 \\ 2 \\ \vdots \\ N \end{bmatrix} \text{\# of samples}$$

$$\boldsymbol{y} = \boldsymbol{X\beta} + \underbrace{\boldsymbol{A}\underline{\boldsymbol{u}}}_{N\times 1} + \boldsymbol{\varepsilon} \qquad (3)$$

where $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma_0^2\boldsymbol{I})$ are the residuals. Here, $\boldsymbol{X}$ and $\boldsymbol{A}$ are the design matrices for the fixed-effect $\boldsymbol{\beta}$ and random-effect $\boldsymbol{u}$, respectively, where $\boldsymbol{\beta} = (m_1, m_2, \cdots, m_K, \beta_1, \beta_2, \cdots, \beta_K)'$ has two components: $(m_1, m_2, \cdots, m_K)$ are the baseline average gene expression in the control group, and $(\beta_1, \beta_2, \cdots, \beta_K)$ are the difference between the case group and the control group. The random effect $\boldsymbol{u} = (u_{11}, u_{21}, \cdots, u_{J1}, u_{12}, u_{22}, \cdots, u_{J2}, \cdots, u_{1K}, u_{2K}, \cdots, u_{JK})'$ captures the individual-level gene expression deviance from the group-level mean, for each cell type. The design matrices $\boldsymbol{X}$ and $\boldsymbol{A}$ are in the form:

haoharryfeng / **ISLET**

Type / to search

Code Issues Pull requests Actions Projects Security Insights

ISLET / R / **dataprep_slope.R**

haoharryfeng 0.99.6

8b8e137 · 2 years ago History

Code Blame 90 lines (73 loc) · 3.89 KB

Raw

```r
###function to read-in and check data from case and control: observed expression, proportion, and sample-to-subject relationship

dataPrepSlope<-function(dat_se){
    message("Begin: working on data preparation as the input for ISLET algorithm.")
#    if (missing(case_dat_se) || missing(ctrl_dat_se))
#        stop("SummarizedExperiment objects from both groups are needed.")
#    if (length(intersect(colData(case_dat_se)[,1], colData(ctrl_dat_se)[,1]))>0)
#        stop("Subject IDs across case group and control group must be unique.")
#    if (ncol(colData(case_dat_se))!=ncol(colData(ctrl_dat_se)))
#        stop("Case group and control group must have the same number of cell types.")

    #subject id between cases and ctrls should also be unique, check and implement this later
    #check for negative values, implement later
    if (!is(dat_se, "SummarizedExperiment"))
        stop("The input dataset must be a SummarizedExperiment object.")
    if (length(unique(colData(dat_se)$group)) != 2)
        stop("There must be two groups (case/ctrl) in the input SummarizedExperiment object.")
    if (unique(colData(dat_se)$group)[1] != "case" || unique(colData(dat_se)$group)[2] != "ctrl")
        stop("The names for the two groups in comparison should be
             labeled as `case` and `ctrl` in the input SummarizedExperiment object.")

    idx <- which(colData(dat_se)$group == "case")

    case_dat_se <- SummarizedExperiment(assays=list(counts=assays(dat_se)$counts[, idx]),
                                       colData=colData(dat_se)[idx, -1])
    ctrl_dat_se <- SummarizedExperiment(assays=list(counts=assays(dat_se)$counts[, -idx]),
                                       colData=colData(dat_se)[-idx, -1])




    #K = number of cell types
    K <- ncol(colData(case_dat_se))-2

    #N1 = number of samples for group 1
    N1 <- ncol(assays(case_dat_se)$counts)
    #N1 = number of samples for group 2
    N2 <- ncol(assays(ctrl_dat_se)$counts)
    #NS = total number of Samples for group 1&2
    NS <- N1 + N2
    #NU = total number of Unique subjects for group 1&2
    caseUN <- length(unique(colData(case_dat_se)[, 1]))
    ctrlUN <- length(unique(colData(ctrl_dat_se)[, 1]))
    NU <- caseUN + ctrlUN
    case_age<-colData(case_dat_se)[, 2] ## The first two columns are: subject ID and sample age.
    ctrl_age<-colData(ctrl_dat_se)[, 2]


    X_sub1 <- as.matrix(rbind(colData(case_dat_se)[, -(seq_len(2))], colData(ctrl_dat_se)[, -(seq_len(2))]))
    X_sub2 <- rbind(matrix(1, nrow=N1, ncol=K), matrix(0, nrow=N2, ncol=K))*X_sub1
    X_age <- c(case_age, ctrl_age)
    X_sub3 <- X_sub1*X_age
    X_sub4 <- rbind(matrix(1, nrow=N1, ncol=K), matrix(0, nrow=N2, ncol=K))*X_sub1*X_age
    ## This is difference in slope between two groups
    X_sub4 <- X_sub4
    X_0 <- cbind(X_sub1, X_sub2, X_sub3, X_sub4)
```

与dataprep基本相同. 此处 $\beta = (m_1, \cdots, m_K, \beta_1, \cdots, \beta_K, \alpha_1, \cdots, \alpha_K, \gamma_1, \cdots, \gamma_K)^T \in \mathbb{R}^{4K}$

$$X_0 = \left(\text{dataprep中的} X_0 \ \vdots \ \begin{matrix} \theta_{i,\cdot} \cdot t_i \\ X\_sub3 \end{matrix} \ \vdots \ \begin{matrix} \theta_{i,\cdot} \cdot t_i \cdot Z_i \\ X\_sub4 \end{matrix} \right) \in [0,1]^{N \times 4K}$$

$\beta \in \mathbb{R}^{4k \times G}$

```
    X_list <- lapply(1, function(x){return(X_0)})
    X <- bdiag(X_list)

    #obtain a vector of unique subject IDs, for all, to use later
    sub_id <- c(colData(case_dat_se)[, 1], colData(ctrl_dat_se)[, 1])

    propm <- as.matrix(rbind(colData(case_dat_se)[, -(seq_len(2))], colData(ctrl_dat_se)[, -(seq_len(2))]))
    # propd = apply(propm, MARGIN = 2, makea, sub_id = sub_id, X = X, NU = NU, simplify = F)
    propd <- apply(X=propm, MARGIN=2, FUN=makea,
                   ind_id=sub_id, datX=X, aNU=NU, simplify=FALSE)

    A_0 <- do.call(cbind, propd)
    #A_list=lapply(1,function(x){return(A_0)})
    A<-bdiag(A_0)

    CT<-colnames(propm)

    datuse <- inputSet(exp_case=assays(case_dat_se)$counts,
                  exp_ctrl=assays(ctrl_dat_se)$counts,
                  X=X,
                  A=A,
                  K=K,
                  NS=NS,
                  NU=NU,
                  case_num=caseUN,
                  ctrl_num=ctrlUN,
                  CT=CT,
                  SubjectID=sub_id,
                  type='slope'
                  )
    message("Complete: data preparation for ISLET.")
    return(datuse)
}
```