DiSNEP

Brian Jo Hsuan Lee

5/5/2022

Load association scores, base network, and similarity matrix

```
ass_score = read_csv("/Users/beelee/Desktop/Columbia/Spring_2022/P8139-Statistical_Genetic_Modeling/Pro
  data.frame()

base_network = read_csv("/Users/beelee/Desktop/Columbia/Spring_2022/P8139-Statistical_Genetic_Modeling/
rownames(base_network) = colnames(base_network)

gene_sim = read_csv("/Users/beelee/Desktop/Columbia/Spring_2022/P8139-Statistical_Genetic_Modeling/Proj
rownames(gene_sim) = colnames(gene_sim)
```

Modify the diffuse matrix function a little

```
diffus_matrix=function(s0,adjacency,alpha=0.75,iter=10,difference=1e-6){

  require(SMUT)

  gene=intersect(rownames(s0),rownames(adjacency))

  s0=s0[rownames(s0) %in% gene,]
  s0=s0[,colnames(s0) %in% gene]
  s0=s0[,order(colnames(s0))]
  s0=s0[order(rownames(s0)),]

  adjacency=adjacency[rownames(adjacency) %in% gene,]
  adjacency=adjacency[,colnames(adjacency) %in% gene]
  adjacency=adjacency[,order(colnames(adjacency))]
  adjacency=adjacency[order(rownames(adjacency)),]

  diag(adjacency)=0
  adjacency=t(t(adjacency)/colSums(adjacency))

  ## added line of code to ensure every entry has a value. may be the source of our undesirable output
  adjacency[is.na(adjacency)] = 0

  snet_1=s0
  snet=snet_1

  for(kk in 1:iter){

    snet_1<-alpha*eigenMapMatMult(adjacency,snet)+(1-alpha)*(s0)
    diff=max(abs(snet_1-snet))
    print(c("iteration:",kk,"difference:",diff))
    if(diff<difference){return(snet_1)}
    snet=snet_1

  }

  return(snet_1)
}
```

Modify the diffuse vector function a little as well

```
diffus_vec=function(signals,snet,type,beta=0.75,iter=10,difference=1e-6,top=100){

  require(SMUT)

  if (type=="pvalue"){
    signals[,2]=-qnorm(signals[,2])
    colnames(signals)=c("gene","score")
  }else{
    colnames(signals)=c("gene","score")
  }

  snet=t(t(snet)/(colSums(snet)+.Machine$double.eps))
  snet[is.na(snet)] = 0

  p=as.matrix(signals$score)

  ## added line of code to ensure every entry has a value
  p[is.na(p)] = 0

  p1=p
  j=1

  repeat{

    p=p1
    p1<-beta*eigenMapMatMult(snet,p)+(1-beta)*(signals$score)

    ## added line of code to ensure every entry has a value
    p1[is.na(p1)] = 0

    p_diff=sum(abs(p1-p))
    j=j+1

    if (j>iter){break}
    if (p_diff<1e-6) {break}

  }
  res=data.frame(gene=signals$gene,score=p1)
  res=res[order(res$score,decreasing = T),]
  res=res[1:top,]
  return(res)
}
```

DiSNEP on our data! Save the ranked gene list.

```
# Step 1. Due to divergence, limit the iteration to 5..
SE = diffus_matrix(base_network, gene_sim, alpha=0.75, iter=5, difference=1e-6)
```

```
## [1] "iteration:"       "1"                "difference:"      "145115.637137223"
## [1] "iteration:"      "2"               "difference:"     "177253.16217909"
## [1] "iteration:"       "3"                "difference:"      "27641831.7818143"
## [1] "iteration:"       "4"                "difference:"      "40812119.4054088"
## [1] "iteration:"       "5"                "difference:"      "5231037707.14657"
```

```
SE_post = post_process(SE,percent=0.9)

# Step 2
res = diffus_vec(ass_score, SE_post, type="pvalue", beta=0.75, iter=10, difference=1e-6, top=500)

write_csv(res, file = "/Users/beelee/Desktop/Columbia/Spring_2022/P8139-Statistical_Genetic_Modeling/Pr
```

Run GeneWanderer on our data save the ranked gene list as well.

```
gw_res = diffus_vec(ass_score, base_network, type="pvalue", beta=0.75, iter=10, difference=1e-6, top=50
write_csv(gw_res, file = "/Users/beelee/Desktop/Columbia/Spring_2022/P8139-Statistical_Genetic_Modeling/
```

Acquire truly associated genes from DisGeNET database. Save the output.

```
true_brca = read_excel("~/Downloads/C0346153_disease_gda_summary.xlsx")
true_brca = true_brca$Gene
write_lines(true_brca, file = "/Users/beelee/Desktop/Columbia/Spring_2022/P8139-Statistical_Genetic_Mode
```

Example code provided

```
data("s0")
data("adjacency")
data("signals")

se=diffus_matrix(s0,adjacency,alpha=0.75,iter=10, difference=1e-6)
```

```
## [1] "iteration:"       "1"                "difference:"      "2755.92911180153"
## [1] "iteration:"       "2"                "difference:"      "73.1243863021488"
## [1] "iteration:"       "3"                "difference:"      "10.3336023977259"
## [1] "iteration:"       "4"                "difference:"      "1.51222523768021"
## [1] "iteration:"        "5"                 "difference:"
## [4] "0.226582906332339"
## [1] "iteration:"         "6"                  "difference:"
## [4] "0.0343595235240173"
## [1] "iteration:"          "7"                   "difference:"
## [4] "0.00562975263434851"
## [1] "iteration:"          "8"                   "difference:"
## [4] "0.00110371353771654"
## [1] "iteration:"          "9"                   "difference:"
## [4] "0.00021658314553008"
## [1] "iteration:"           "10"                   "difference:"
## [4] "4.25188717514402e-05"
```

```
se_post=post_process(se,percent=0.9)
res=diffus_vec(signals,se_post,type="pvalue", beta=0.75, iter=10, difference=1e-6, top=100)
```