|  |
| --- |
| > setwd('P:/CCTI\_USERS/Jianing Fu/Adaptive samples and analysis/Pt18 MVTx')  > data=read.table('Pt18 02072018.tsv',header=T)  > rownames(data)=data[,1]  > names(data)  [1] "nucleotide" "total" "Pt18\_MVTx\_BM\_POD357"  [4] "Pt18\_MVTx\_BM\_POD357\_Donor\_CD45." "Pt18\_MVTx\_Bx\_POD105" "Pt18\_MVTx\_Bx\_POD18"  [7] "Pt18\_MVTx\_Bx\_POD307" "Pt18\_MVTx\_Bx\_POD35" "Pt18\_MVTx\_Bx\_POD5.7"  [10] "Pt18\_MVTx\_Bx\_Tx\_colon\_POD357" "Pt18\_MVTx\_Bx\_duodenum\_POD357" "Pt18\_MVTx\_Bx\_ileum\_POD357"  [13] "Pt18\_MVTx\_Bx\_native\_colon\_POD357" "Pt18\_MVTx\_Bx\_stomach\_POD357" "Pt18\_MVTx\_PBMC\_POD14"  [16] "Pt18\_MVTx\_PBMC\_POD28" "Pt18\_MVTx\_PBMC\_POD314" "Pt18\_MVTx\_PBMC\_POD357"  [19] "Pt18\_MVTx\_PBMC\_POD357\_Donor\_CD45." "Pt18\_MVTx\_PBMC\_POD5" "Pt18\_MVTx\_PBMC\_POD7"  [22] "Pt18\_MVTx\_PBMC\_POD98" "Pt18\_MVTx\_SP\_D4U" "Pt18\_MVTx\_SP\_D8U"  [25] "Pt18\_MVTx\_SP\_GVH\_D4L" "Pt18\_MVTx\_SP\_GVH\_D8L" "Pt18\_MVTx\_SP\_HVG\_R4L"  [28] "Pt18\_MVTx\_SP\_HVG\_R8L" "Pt18\_MVTx\_SP\_R4U" "Pt18\_MVTx\_SP\_R8U"  > x=data  > ambiguous=which((x[,29]>0 | x[,30]>0)&(x[,23]>0 | x[,24]>0))  > ambiguous=union(ambiguous,which((x[,29]>0 | x[,30]>0)&(x[,25]>0 | x[,26]>0)))  > ambiguous=union(ambiguous,which((x[,23]>0 | x[,24]>0)&(x[,27]>0 | x[,28]>0)))  > x=x[setdiff(1:nrow(x),ambiguous),]  >  > # venn diagram of Pt18 POD357 PBMC  > a=rownames(x[x[,23]>0 | x[,24]>0,])  > b=rownames(x[x[,19]>0,])  > c=rownames(x[x[,25]>0 | x[,26]>0,])  > venn\_diagram(a,b,c)  [1] "a:" "111176"  [1] "b:" "1820"  [1] "c:" "27271"  [1] "a,b:" "68"  [1] "a,c:" "483"  [1] "b,c:" "1"  [1] "a,b,c:" "1"  > cloneCal(x[,23])  [1] 0.037  > cloneCal(x[,24])  [1] 0.041  > cloneCal(x[,25])  [1] 0.18  > cloneCal(x[,26])  [1] 0.22  > donormappable=which((x[,23]>0 | x[,24]>0 | x[,25]>0 | x[,26]>0))  > x=x[setdiff(1:nrow(x),donormappable),]  > cloneCal(x[,19])  [1] 0.092 |

setwd('P:/CCTI\_USERS/Jianing Fu/Adaptive samples and analysis')

data=read.table('Pt15 05042017.tsv',header=T)

setwd('P:/CCTI\_USERS/Jianing Fu/Adaptive samples and analysis')

data2=read.table('Pt15 05042017 table 2.tsv',header=T)

p15=merge(data,data2,by="nucleotide", all=T)

p15[is.na(p15)]=0

write.table(p15,file ="P:/CCTI\_USERS/Jianing Fu/Adaptive samples and analysis/Pt15 combined.tsv",quote=F,row.names=F,col.names=F, sep="\t")

> setwd('P:/CCTI\_USERS/Jianing Fu/Adaptive samples and analysis')

> data=read.table('Pt15 combined.tsv',header=T)

> source('P:/CCTI\_USERS/Jianing Fu/Adaptive samples and analysis/09-12-2017 JSD ileum colon native colon/alloreactivity\_printtofile 09132017.R')

> rownames(data)=data[,1]

> x=data

> names(data)

[1] "nucleotide" "total.x"

[3] "Pt15MVTx.GVH.D4L" "Pt15MVTx.GVH.D8L"

[5] "Pt15MVTx.GVH.R4L" "Pt15MVTx.GVH.R8L"

[7] "Pt15MVTx.PBMCs.POD143.Donor\_T" "Pt15MVTx.PBMCs.POD143.Recipient\_T"

[9] "Pt15MVTx.pre.Tx.donor\_MLN" "Pt15MVTx.unstim.D4"

[11] "Pt15MVTx.unstim.D8" "Pt15MVTx.unstim.R4"

[13] "Pt15MVTx.unstim.R8" "Pt15\_MVTx\_PBMC\_POD83"

[15] "Pt15\_MVTx\_PBMC\_POD83\_MLR\_D3L" "total.y"

[17] "Pt15MVTx.Bx.POD17" "Pt15MVTx.Bx.POD27"

[19] "Pt15MVTx.Bx.POD55" "Pt15MVTx.PBMCs.POD11"

[21] "Pt15MVTx.PBMCs.POD19" "Pt15MVTx.PBMCs.POD26"

[23] "Pt15MVTx\_Bx\_POD237" "Pt15MVTx\_PBMC\_POD255"

> ambiguous=which((x[,13]>0 | x[,12]>0)&(x[,11]>0 | x[,10]>0))

> ambiguous=union(ambiguous,which((x[,13]>0 | x[,12]>0)&(x[,3]>0 | x[,4]>0)))

> ambiguous=union(ambiguous,which((x[,11]>0 | x[,10]>0)&(x[,5]>0 | x[,6]>0)))

> x=x[setdiff(1:nrow(x),ambiguous),]

> # venn diagram of Pt15 POD143

> a=rownames(x[x[,7]>0,])

>

> b=rownames(x[x[,9]>0,])

>

> c=rownames(x[x[,11]>0 | x[,10]>0 | x[,3]>0 | x[,4]>0,])

>

> venn\_diagram(a,b,c)

[1] "a:" "48042"

[1] "b:" "116905"

[1] "c:" "251773"

[1] "a,b:" "134"

[1] "a,c:" "460"

[1] "b,c:" "3562"

[1] "a,b,c:" "268"

> #cloneCal(x[,3]) #clonality of a given column

> cloneCal(x[,10])

[1] 0.032

> cloneCal(x[,11])

[1] 0.06

> cloneCal(x[,9])

[1] 0.025

> #clonality of donor non-mappable clones in POD143 (0.023)

> donormappable=which((x[,11]>0 | x[,10]>0 | x[,3]>0 | x[,4]>0 | x[,9]>0))

> x=x[setdiff(1:nrow(x),donormappable),]

> cloneCal(x[,7])   
#x changed at this step, need to rerun the remove ambiguous clones.(data: 1069040 obs. Of 24 variables, after remove ambiguous clones, x: 1068348 obs. Of 24 variables)

[1] 0.023

> length(which(x[,10]>0))

[1] 121515

> length(which(x[,11]>0))

[1] 121919

> length(which(x[,7]>0))

[1] 48904

> length(which(x[,9]>0))

[1] 120869

> #abundancePlot(x[,c(10,11,9)])

> abundancePlot(normalize(x[,c(10,11,9)]))

> d\_derived=rownames(x[x[,11]+x[,10]+x[,3]+x[,4]+x[,9]>0,])

> pbmc143=rownames(x[x[,7]>0,])

> nonmappable=setdiff(pbmc143,d\_derived)

> mappable=intersect(pbmc143,d\_derived)

> length(nonmappable)

[1] 48042

> length(mappable)

[1] 862

**#Add in the following function into the background code file (**alloreactivity\_printtofile 09132017.R**).**

#removes the single highest clone frequency from the abundance plot regression according to Susan and Boris et. al. Jci Insight. 2018 In press.

getAbundancePlotSlope<-function(x){

  x=x/sum(x)

  x=x[x>0]

  out=as.data.frame(table(x[x<max(x)]))

  x=log10(as.numeric(as.character(out[,1])))

  y=log10(out[,2])

  return(lm(y~x)$coefficients[2])

}

#This command gives the abundance plot slope from a single input vector, so you could run it as getAbundancePlotSlope(data[,column]) or on any arbitrary number of columns at once using the apply() function [e.g. apply(data,2,getAbundancePlotSlope)]. This runs the same way as the other diversity functions like cloneCal, simpsonCal, entropyCal, getR20, etc...

#Mac

> setwd('~/Desktop/gut HSC paper')

> data=read.table('Pt15 combined.tsv',header=T)

> source('~/Desktop/gut HSC paper/alloreactivity\_printtofile 09132017.R')

> rownames(data)=data[,1]

> x=data

> names(data)

[1] "nucleotide" "total.x"

[3] "Pt15MVTx.GVH.D4L" "Pt15MVTx.GVH.D8L"

[5] "Pt15MVTx.GVH.R4L" "Pt15MVTx.GVH.R8L"

[7] "Pt15MVTx.PBMCs.POD143.Donor\_T" "Pt15MVTx.PBMCs.POD143.Recipient\_T"

[9] "Pt15MVTx.pre.Tx.donor\_MLN" "Pt15MVTx.unstim.D4"

[11] "Pt15MVTx.unstim.D8" "Pt15MVTx.unstim.R4"

[13] "Pt15MVTx.unstim.R8" "Pt15\_MVTx\_PBMC\_POD83"

[15] "Pt15\_MVTx\_PBMC\_POD83\_MLR\_D3L" "total.y"

[17] "Pt15MVTx.Bx.POD17" "Pt15MVTx.Bx.POD27"

[19] "Pt15MVTx.Bx.POD55" "Pt15MVTx.PBMCs.POD11"

[21] "Pt15MVTx.PBMCs.POD19" "Pt15MVTx.PBMCs.POD26"

[23] "Pt15MVTx\_Bx\_POD237" "Pt15MVTx\_PBMC\_POD255"

> ambiguous=which((x[,13]>0 | x[,12]>0)&(x[,11]>0 | x[,10]>0))

> ambiguous=union(ambiguous,which((x[,13]>0 | x[,12]>0)&(x[,3]>0 | x[,4]>0)))

> ambiguous=union(ambiguous,which((x[,11]>0 | x[,10]>0)&(x[,5]>0 | x[,6]>0)))

> x=x[setdiff(1:nrow(x),ambiguous),]

> a=rownames(x[x[,7]>0,])

>

> b=rownames(x[x[,9]>0,])

>

> c=rownames(x[x[,11]>0 | x[,10]>0 | x[,3]>0 | x[,4]>0,])

>

> venn\_diagram(a,b,c)

[1] "a:" "48042"

[1] "b:" "116905"

[1] "c:" "251773"

[1] "a,b:" "134"

[1] "a,c:" "460"

[1] "b,c:" "3562"

[1] "a,b,c:" "268"

> cloneCal(x[,10])

[1] 0.032

> cloneCal(x[,11])

[1] 0.06

> cloneCal(x[,9])

[1] 0.025

> source('~/Desktop/gut HSC paper/alloreactivity\_printtofile 09132017.R')

> getAbundancePlotSlope(x[,10])

x

**-2.519687**

> getAbundancePlotSlope(x[,11])

x

**-1.528721**

> getAbundancePlotSlope(x[,9])

x

**-2.3024**

> donormappable=which((x[,11]>0 | x[,10]>0 | x[,3]>0 | x[,4]>0 | x[,9]>0))

> x=x[setdiff(1:nrow(x),donormappable),]

> cloneCal(x[,7])

[1] 0.023

> getAbundancePlotSlope(x[,7])

x

**-2.090745**

>

setwd('P:/CCTI\_USERS/Jianing Fu/Adaptive samples and analysis')

data=read.table('Pt7 12062016.tsv',header=T)

[1] "nucleotide" "total" "Itx.Pt7.PBMC.POD253" "Itx.Pt7.POD101.136.DM" [5] "Itx.Pt7.POD101.136.DN" "Itx.Pt7.POD22.50.D" "Itx.Pt7.POD22.50.R4" "Itx.Pt7.POD22.50.R8" [9] "Itx.Pt7.POD225.Bx" "Itx.Pt7.POD24.Bx" "Itx.Pt7.PreTx.4U" "Itx.Pt7.PreTx.8U" [13] "Itx.Pt7.PreTx.D4L" "Itx.Pt7.PreTx.D4U" "Itx.Pt7.PreTx.D8L" "Itx.Pt7.PreTx.D8U" [17] "Itx.Pt7.PreTx.R4L" "Itx.Pt7.PreTx.R8L"

rownames(data)=data[,1]

x=data

#remove donor/recipient ambiguous clones, either real shared clones, or ambiguous raised by CFSE-MLR sorting error # manually removing 2 ambiguous clones between sorted donor memory vs naïve T cells in Pt7 PBMCs.

ambiguous=which((x[,4]>0)&(x[,5]>0))

ambiguous=union(ambiguous,which((x[,12]>0 | x[,11]>0)&(x[,14]>0 | x[,16]>0)))

ambiguous=union(ambiguous,which((x[,12]>0 | x[,11]>0)&(x[,13]>0 | x[,15]>0)))

ambiguous=union(ambiguous,which((x[,14]>0 | x[,16]>0)&(x[,17]>0 | x[,18]>0)))

x=x[setdiff(1:nrow(x),ambiguous),]

#cloneCal(x[,3]) #clonality of a given column

cloneCal(x[,4])

0.15

cloneCal(x[,5])

0.061

> #Bx POD24

> cloneCal(x[,10])

[1] 0.14

> #D4U

> cloneCal(x[,14])

[1] 0.061

> #D8U

> cloneCal(x[,16])

[1] 0.15

> #D4L

> cloneCal(x[,13])

[1] 0.093

> #D8L

> cloneCal(x[,15])

[1] 0.18

#venn diagram

# a=rownames(data[data[,3]>0,]) # the clones present in column 3

# b=rownames(data[data[,4]>0,]) # the clones present in column 4

# c=rownames(data[data[,5]>0,]) # the clones present in column 5

# venn diagram of Memory T

a=rownames(x[x[,4]>0,])

b=rownames(x[x[,10]>0,])

c=rownames(x[x[,14]>0 | x[,16]>0 | x[,13]>0 | x[,15]>0,])

venn\_diagram(a,b,c)

# venn diagram of Naive T

a=rownames(x[x[,5]>0,])

b=rownames(x[x[,10]>0,])

c=rownames(x[x[,14]>0 | x[,16]>0 | x[,13]>0 | x[,15]>0,])

venn\_diagram(a,b,c)