#Pt15 pre-Tx MLR defined alloreactivity

This part of R script is published previously, and is available in the GitHub repository at <https://github.com/Aleksobrad/Fu-J-et-al.-LGVHR-manuscript>

#Pt15 post-Tx MLR defined alloreactivity

|  |  |
| --- | --- |
| Abbreviation | Full name |
| D4U | PreTx\_Donor\_CD4\_unstim |
| D4L | PreTx\_Donor\_CD4\_CFSElow |
| D8U | PreTx\_Donor\_CD8\_unstim |
| D8L | PreTx\_Donor\_CD8\_CFSElow |
| R4U | PreTx\_Recipient\_CD4\_unstim |
| R4L | PreTx\_Recipient\_CD4\_CFSElow |
| R8U | PreTx\_Recipient\_CD8\_unstim |
| R8L | PreTx\_Recipient\_CD8\_CFSElow |
| R'4U | PostTx\_Recipient\_CD4\_unstim |
| R'4L | PostTx\_Recipient\_CD4\_CFSElow |
| R'8U | PostTx\_Recipient\_CD8\_unstim |
| R'8L | PostTx\_Recipient\_CD8\_CFSElow |

data <- read.delim("P:/CCTI\_USERS/Jianing Fu/Adaptive samples and analysis/Pt21 MVTx/Pt21 12262019.tsv")

source('P:/CCTI\_USERS/Jianing Fu/Adaptive samples and analysis/! summary of R codes/alloreactivity\_printtofile 09132017.R')

names(data)

[1] "Rearrangement" "Sum..Templates."

[3] "Present.In" "Pt21\_MVTx\_Bx\_duodenum\_POD243\_sorted\_T"

[5] "Pt21\_MVTx\_Bx\_stomach\_POD243\_sorted\_T" "Pt21\_MVTx\_LN\_gut\_associated\_POD262\_sorted\_T"

[7] "Pt21\_MVTx\_POD306\_PBMC\_R.4L\_D\_SP" "Pt21\_MVTx\_POD306\_PBMC\_R.4U"

[9] "Pt21\_MVTx\_POD306\_PBMC\_R.8L\_D\_SP" "Pt21\_MVTx\_POD306\_PBMC\_R.8U"

[11] "Pt21\_MVTx\_POD626\_LN\_T" "Pt21\_MVTx\_POD626\_PBMC\_T"

[13] "Pt21\_MVTx\_POD626\_ileum\_Bx" "Pt21\_Post\_PBMC\_day0011"

[15] "Pt21\_Post\_PBMC\_day0023" "Pt21\_Post\_PBMC\_day0109"

[17] "Pt21\_Post\_PBMC\_day0262" "Pt21\_Post\_ileum\_day0023"

[19] "Pt21\_Post\_ileum\_day0109" "Pt21\_Post\_ileum\_day0243"

[21] "Pt21\_Post\_ileum\_day0262" "Pt21\_Pre\_Donor\_CD4\_CFSElo"

[23] "Pt21\_Pre\_Donor\_CD4\_unstim" "Pt21\_Pre\_Donor\_CD8\_CFSElo"

[25] "Pt21\_Pre\_Donor\_CD8\_unstim" "Pt21\_Pre\_Recipient\_CD4\_CFSElo"

[27] "Pt21\_Pre\_Recipient\_CD4\_unstim" "Pt21\_Pre\_Recipient\_CD8\_CFSElo"

[29] "Pt21\_Pre\_Recipient\_CD8\_unstim"

|  |  |
| --- | --- |
| Sample name (Pt: patient) | updated rename |
| Pt21\_MVTx\_SP\_GVH\_D4L | Pt21\_Pre\_Donor\_CD4\_CFSElo |
| Pt21\_MVTx\_SP\_D4U | Pt21\_Pre\_Donor\_CD4\_unstim |
| Pt21\_MVTx\_SP\_GVH\_D8L | Pt21\_Pre\_Donor\_CD8\_CFSElo |
| Pt21\_MVTx\_SP\_D8U | Pt21\_Pre\_Donor\_CD8\_unstim |
| Pt21\_MVTx\_Bx\_POD23 | Pt21\_Post\_ileum\_day0023 |
| Pt21\_MVTx\_Bx\_POD109 | Pt21\_Post\_ileum\_day0109 |
| Pt21\_MVTx\_Bx\_ileum\_POD243\_sorted T | Pt21\_Post\_ileum\_day0243 |
| Pt21\_MVTx\_Bx\_ileum stoma\_POD262\_sorted T | Pt21\_Post\_ileum\_day0262 |
| Pt21\_MVTx\_PBMC\_POD11 | Pt21\_Post\_PBMC\_day0011 |
| Pt21\_MVTx\_PBMC\_POD23 | Pt21\_Post\_PBMC\_day0023 |
| Pt21\_MVTx\_PBMC\_POD109 | Pt21\_Post\_PBMC\_day0109 |
| Pt21\_MVTx\_PBMC\_POD262\_sorted T | Pt21\_Post\_PBMC\_day0262 |
| Pt21\_MVTx\_SP\_HVG\_R4L | Pt21\_Pre\_Recipient\_CD4\_CFSElo |
| Pt21\_MVTx\_SP\_R4U | Pt21\_Pre\_Recipient\_CD4\_unstim |
| Pt21\_MVTx\_SP\_HVG\_R8L | Pt21\_Pre\_Recipient\_CD8\_CFSElo |
| Pt21\_MVTx\_SP\_R8U | Pt21\_Pre\_Recipient\_CD8\_unstim |

# resolve CD4 and CD8 ambiguous clones first (ratio=5), then followed by resolve donor and recipient ambiguous clones.

#D4U vs D8U

normalize(data[,c(23,25)])

ratio=5

c1indices=which(data[,23]>0 & data[,25]>0 & data[,23]>ratio\*data[,25])

c2indices=which(data[,23]>0 & data[,25]>0 & data[,25]>ratio\*data[,23])

ambiindices=which(data[,23]>0 & data[,25]>0)

ambiindices=setdiff(setdiff(ambiindices,c1indices),c2indices)

data[c1indices,25]=0

data[c2indices,23]=0

data[ambiindices,c(23,25)]=0

#D4L vs D8L

normalize(data[,c(22,24)])

ratio=5

c1indices=which(data[,22]>0 & data[,24]>0 & data[,22]>ratio\*data[,24])

c2indices=which(data[,22]>0 & data[,24]>0 & data[,24]>ratio\*data[,22])

ambiindices=which(data[,22]>0 & data[,24]>0)

ambiindices=setdiff(setdiff(ambiindices,c1indices),c2indices)

data[c1indices,24]=0

data[c2indices,22]=0

data[ambiindices,c(22,24)]=0

#R4U vs R8U

normalize(data[,c(27,29)])

ratio=5

c1indices=which(data[,27]>0 & data[,29]>0 & data[,27]>ratio\*data[,29])

c2indices=which(data[,27]>0 & data[,29]>0 & data[,29]>ratio\*data[,27])

ambiindices=which(data[,27]>0 & data[,29]>0)

ambiindices=setdiff(setdiff(ambiindices,c1indices),c2indices)

data[c1indices,29]=0

data[c2indices,27]=0

data[ambiindices,c(27,29)]=0

#R4L vs R8L

normalize(data[,c(26,28)])

ratio=5

c1indices=which(data[,26]>0 & data[,28]>0 & data[,26]>ratio\*data[,28])

c2indices=which(data[,26]>0 & data[,28]>0 & data[,28]>ratio\*data[,26])

ambiindices=which(data[,26]>0 & data[,28]>0)

ambiindices=setdiff(setdiff(ambiindices,c1indices),c2indices)

data[c1indices,28]=0

data[c2indices,26]=0

data[ambiindices,c(26,28)]=0

rownames(data)=data[,1]

x=data

#remove donor/recipient ambiguous clones, either real shared clones, or ambiguous raised by CFSE-MLR sorting error

ambiguous=which((x[,23]>0 | x[,25]>0)&(x[,27]>0 | x[,29]>0))

ambiguous=union(ambiguous,which((x[,23]>0 | x[,25]>0)&(x[,26]>0 | x[,28]>0)))

ambiguous=union(ambiguous,which((x[,27]>0 | x[,29]>0)&(x[,22]>0 | x[,24]>0)))

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

#additionally remove stim vs unstim ambiguous within recipient preTx (R4L, R4U, R8L, R8U) or donor preTx (D4L, D4U, D8L, D8U) samples; which can be considered as additional step of optimization for CD4 and CD8 sorting error.

#R4L vs R8U

normalize(x[,c(26,29)])

ratio=2

c1indices=which(x[,26]>0 & x[,29]>0 & x[,26]>ratio\*x[,29])

c2indices=which(x[,26]>0 & x[,29]>0 & x[,29]>ratio\*x[,26])

ambiindices=which(x[,26]>0 & x[,29]>0)

ambiindices=setdiff(setdiff(ambiindices,c1indices),c2indices)

x[c1indices,29]=0

x[c2indices,26]=0

x[ambiindices,c(26,29)]=0

#R8L vs R4U

normalize(x[,c(28,27)])

ratio=2

c1indices=which(x[,28]>0 & x[,27]>0 & x[,28]>ratio\*x[,27])

c2indices=which(x[,28]>0 & x[,27]>0 & x[,27]>ratio\*x[,28])

ambiindices=which(x[,28]>0 & x[,27]>0)

ambiindices=setdiff(setdiff(ambiindices,c1indices),c2indices)

x[c1indices,27]=0

x[c2indices,28]=0

x[ambiindices,c(28,27)]=0

#D4L vs D8U

normalize(x[,c(22,25)])

ratio=2

c1indices=which(x[,22]>0 & x[,25]>0 & x[,22]>ratio\*x[,25])

c2indices=which(x[,22]>0 & x[,25]>0 & x[,25]>ratio\*x[,22])

ambiindices=which(x[,22]>0 & x[,25]>0)

ambiindices=setdiff(setdiff(ambiindices,c1indices),c2indices)

x[c1indices,25]=0

x[c2indices,22]=0

x[ambiindices,c(22,25)]=0

#D8L vs D4U

normalize(x[,c(24,23)])

ratio=2

c1indices=which(x[,24]>0 & x[,23]>0 & x[,24]>ratio\*x[,23])

c2indices=which(x[,24]>0 & x[,23]>0 & x[,23]>ratio\*x[,24])

ambiindices=which(x[,24]>0 & x[,23]>0)

ambiindices=setdiff(setdiff(ambiindices,c1indices),c2indices)

x[c1indices,23]=0

x[c2indices,24]=0

x[ambiindices,c(24,23)]=0

#resolve ambiguous raised by post-Tx unstim and CFSE-MLR

#post-Tx R4U (8) vs R8U (10)

normalize(x[,c(8,10)])

ratio=5

c1indices=which(x[,8]>0 & x[,10]>0 & x[,8]>ratio\*x[,10])

c2indices=which(x[,8]>0 & x[,10]>0 & x[,10]>ratio\*x[,8])

ambiindices=which(x[,8]>0 & x[,10]>0)

ambiindices=setdiff(setdiff(ambiindices,c1indices),c2indices)

x[c1indices,10]=0

x[c2indices,8]=0

x[ambiindices,c(8,10)]=0

#post-Tx R4L (7) vs R8L (9)

normalize(x[,c(7,9)])

ratio=5

c1indices=which(x[,7]>0 & x[,9]>0 & x[,7]>ratio\*x[,9])

c2indices=which(x[,7]>0 & x[,9]>0 & x[,9]>ratio\*x[,7])

ambiindices=which(x[,7]>0 & x[,9]>0)

ambiindices=setdiff(setdiff(ambiindices,c1indices),c2indices)

x[c1indices,9]=0

x[c2indices,7]=0

x[ambiindices,c(7,9)]=0

#remove donor/recipient ambiguous clones, either real shared clones, or ambiguous raised by CFSE-MLR sorting error

ambiguous=which((x[,23]>0 | x[,25]>0)&(x[,8]>0 | x[,10]>0))

ambiguous=union(ambiguous,which((x[,23]>0 | x[,25]>0)&(x[,7]>0 | x[,9]>0)))

ambiguous=union(ambiguous,which((x[,8]>0 | x[,10]>0)&(x[,22]>0 | x[,24]>0)))

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

#additionally remove stim vs unstim ambiguous within recipient postTx (R’4L, R’4U, R’8L, R’8U) samples; which can be considered as additional step of optimization for CD4 and CD8 sorting error.

#R'4L vs R'8U

normalize(x[,c(7,10)])

ratio=2

c1indices=which(x[,7]>0 & x[,10]>0 & x[,7]>ratio\*x[,10])

c2indices=which(x[,7]>0 & x[,10]>0 & x[,10]>ratio\*x[,7])

ambiindices=which(x[,7]>0 & x[,10]>0)

ambiindices=setdiff(setdiff(ambiindices,c1indices),c2indices)

x[c1indices,10]=0

x[c2indices,7]=0

x[ambiindices,c(7,10)]=0

#R'8L vs R'4U

normalize(x[,c(9,8)])

ratio=2

c1indices=which(x[,9]>0 & x[,8]>0 & x[,9]>ratio\*x[,8])

c2indices=which(x[,9]>0 & x[,8]>0 & x[,8]>ratio\*x[,9])

ambiindices=which(x[,9]>0 & x[,8]>0)

ambiindices=setdiff(setdiff(ambiindices,c1indices),c2indices)

x[c1indices,8]=0

x[c2indices,9]=0

x[ambiindices,c(9,8)]=0

write.table(x,file ="P:/CCTI\_USERS/Jianing Fu/Adaptive samples and analysis/Pt21 MVTx/Pt21 12262019 resolve ambiguous updated with postTx MLR.tsv",quote=F,row.names=F,col.names=T, sep="\t")

# add "fold=2" below, use 2 fold expansion (default 5 fold)

# add "freq=0.00002" below, use freq=0.00002 for template counts; use default freq=0.00001 for read counts

#post-Tx POD306 HVG direction

# For CD4 HvG, cd4.HVG=x[,c(unstim(R4U),stim(R4L))]

# For CD8 HvG, cd8.HVG=x[,c(unstim(R8U),stim(R8L))]

cd4.HVG= x[,c(8,7)]

cd8.HVG= x[,c(10,9)]

allo.HVG=listAlloreactive(cd4.HVG,cd8.HVG, fold=2, freq=0.00002)

length(allo.HVG[[1]])

[1] 3258

length(allo.HVG[[2]])

[1] 2157

write.table(allo.HVG[[1]],file ="P:/CCTI\_USERS/Jianing Fu/Adaptive samples and analysis/Pt21 MVTx/Pt21 CD4 post-Tx POD306 HVGlist.txt",quote=F,row.names=F,col.names=F, sep="\t")

write.table(allo.HVG[[2]],file ="P:/CCTI\_USERS/Jianing Fu/Adaptive samples and analysis/Pt21 MVTx/Pt21 CD8 post-Tx POD306 HVGlist.txt",quote=F,row.names=F,col.names=F, sep="\t")

rCD4mappable=rownames(x[(x[,8]+x[,7])>0,])

rCD8mappable=rownames(x[(x[,10]+x[,9])>0,])

CD4nonHVG=setdiff(rCD4mappable,allo.HVG[[1]])

CD8nonHVG=setdiff(rCD8mappable,allo.HVG[[2]])

write.table(CD4nonHVG,file ="P:/CCTI\_USERS/Jianing Fu/Adaptive samples and analysis/Pt21 MVTx/Pt21 CD4nonHVG rmappable list post-Tx POD306.txt",quote=F,row.names=F,col.names=F, sep="\t")

write.table(CD8nonHVG,file ="P:/CCTI\_USERS/Jianing Fu/Adaptive samples and analysis/Pt21 MVTx/Pt21 CD8nonHVG rmappable list post-Tx POD306.txt",quote=F,row.names=F,col.names=F, sep="\t")

length(rCD4mappable)

[1] 27923

length(rCD8mappable)

[1] 14064

length(CD4nonHVG)

[1] 24665

length(CD8nonHVG)

[1] 11907

rmappable=rownames(x[(x[,8]+x[,7]+x[,10]+x[,9])>0,])

length(rmappable)

[1] 41987

write.table(rmappable,file ="P:/CCTI\_USERS/Jianing Fu/Adaptive samples and analysis/Pt21 MVTx/Pt21 recipient mappable list post-Tx POD306.txt",quote=F,row.names=F,col.names=F, sep="\t")

length(rCD4mappable)+length(rCD8mappable)

[1] 41987

length(intersect(rCD4mappable,rCD8mappable))

[1] 0

#Pt15 integrate pre- and post-Tx MLRs

Pt21\_new <- read.delim("~/Desktop/10x TRM 2021/pre post MLR analysis/Pt21 12262019 resolve ambiguous updated with postTx MLR.tsv")

colnames(Pt21\_new)

[1] "Rearrangement" "Sum..Templates."

[3] "Present.In" "Pt21\_MVTx\_Bx\_duodenum\_POD243\_sorted\_T"

[5] "Pt21\_MVTx\_Bx\_stomach\_POD243\_sorted\_T" "Pt21\_MVTx\_LN\_gut\_associated\_POD262\_sorted\_T"

[7] "Pt21\_MVTx\_POD306\_PBMC\_R.4L\_D\_SP" "Pt21\_MVTx\_POD306\_PBMC\_R.4U"

[9] "Pt21\_MVTx\_POD306\_PBMC\_R.8L\_D\_SP" "Pt21\_MVTx\_POD306\_PBMC\_R.8U"

[11] "Pt21\_MVTx\_POD626\_LN\_T" "Pt21\_MVTx\_POD626\_PBMC\_T"

[13] "Pt21\_MVTx\_POD626\_ileum\_Bx" "Pt21\_Post\_PBMC\_day0011"

[15] "Pt21\_Post\_PBMC\_day0023" "Pt21\_Post\_PBMC\_day0109"

[17] "Pt21\_Post\_PBMC\_day0262" "Pt21\_Post\_ileum\_day0023"

[19] "Pt21\_Post\_ileum\_day0109" "Pt21\_Post\_ileum\_day0243"

[21] "Pt21\_Post\_ileum\_day0262" "Pt21\_Pre\_Donor\_CD4\_CFSElo"

[23] "Pt21\_Pre\_Donor\_CD4\_unstim" "Pt21\_Pre\_Donor\_CD8\_CFSElo"

[25] "Pt21\_Pre\_Donor\_CD8\_unstim" "Pt21\_Pre\_Recipient\_CD4\_CFSElo"

[27] "Pt21\_Pre\_Recipient\_CD4\_unstim" "Pt21\_Pre\_Recipient\_CD8\_CFSElo"

[29] "Pt21\_Pre\_Recipient\_CD8\_unstim"

|  |  |
| --- | --- |
| Sample name (Pt: patient) | updated rename |
| Pt21\_MVTx\_SP\_GVH\_D4L | Pt21\_Pre\_Donor\_CD4\_CFSElo |
| Pt21\_MVTx\_SP\_D4U | Pt21\_Pre\_Donor\_CD4\_unstim |
| Pt21\_MVTx\_SP\_GVH\_D8L | Pt21\_Pre\_Donor\_CD8\_CFSElo |
| Pt21\_MVTx\_SP\_D8U | Pt21\_Pre\_Donor\_CD8\_unstim |
| Pt21\_MVTx\_Bx\_POD23 | Pt21\_Post\_ileum\_day0023 |
| Pt21\_MVTx\_Bx\_POD109 | Pt21\_Post\_ileum\_day0109 |
| Pt21\_MVTx\_Bx\_ileum\_POD243\_sorted T | Pt21\_Post\_ileum\_day0243 |
| Pt21\_MVTx\_Bx\_ileum stoma\_POD262\_sorted T | Pt21\_Post\_ileum\_day0262 |
| Pt21\_MVTx\_PBMC\_POD11 | Pt21\_Post\_PBMC\_day0011 |
| Pt21\_MVTx\_PBMC\_POD23 | Pt21\_Post\_PBMC\_day0023 |
| Pt21\_MVTx\_PBMC\_POD109 | Pt21\_Post\_PBMC\_day0109 |
| Pt21\_MVTx\_PBMC\_POD262\_sorted T | Pt21\_Post\_PBMC\_day0262 |
| Pt21\_MVTx\_SP\_HVG\_R4L | Pt21\_Pre\_Recipient\_CD4\_CFSElo |
| Pt21\_MVTx\_SP\_R4U | Pt21\_Pre\_Recipient\_CD4\_unstim |
| Pt21\_MVTx\_SP\_HVG\_R8L | Pt21\_Pre\_Recipient\_CD8\_CFSElo |
| Pt21\_MVTx\_SP\_R8U | Pt21\_Pre\_Recipient\_CD8\_unstim |

rcd4 <- Pt21\_new[,c(27,26)]

rcd4 <- normalize(rcd4)

rCD4HVG <- rownames(rcd4[rcd4[,2]>0.00002 & rcd4[,2] > rcd4[,1]\*2,])

rCD4NonHVG <- setdiff(rownames(Pt21\_new[Pt21\_new[,27]>0|Pt21\_new[,26]>0,]), rCD4HVG)

rcd8 <- Pt21\_new[,c(29,28)]

rcd8 <- normalize(rcd8)

rCD8HVG <- rownames(rcd8[rcd8[,2]>0.00002 & rcd8[,2] > rcd8[,1]\*2,])

rCD8NonHVG <- setdiff(rownames(Pt21\_new[Pt21\_new[,29]>0|Pt21\_new[,28]>0,]), rCD8HVG)

dcd4 <- Pt21\_new[,c(23,22)]

dcd4 <- normalize(dcd4)

rCD4GVH <- rownames(dcd4[dcd4[,2]>0.00002 & dcd4[,2] > dcd4[,1]\*2,])

rCD4NonGVH <- setdiff(rownames(Pt21\_new[Pt21\_new[,23]>0|Pt21\_new[,22]>0,]), rCD4GVH)

dcd8 <- Pt21\_new[,c(25,24)]

dcd8 <- normalize(dcd8)

rCD8GVH <- rownames(dcd8[dcd8[,2]>0.00002 & dcd8[,2] > dcd8[,1]\*2,])

rCD8NonGVH <- setdiff(rownames(Pt21\_new[Pt21\_new[,25]>0|Pt21\_new[,24]>0,]), rCD8GVH)

r2cd4 <- Pt21\_new[,c(8,7)]

r2cd4 <- normalize(r2cd4)

r2CD4HVG <- rownames(r2cd4[r2cd4[,2]>0.00002 & r2cd4[,2] > r2cd4[,1]\*2,])

r2CD4NonHVG <- setdiff(rownames(Pt21\_new[Pt21\_new[,8]>0|Pt21\_new[,7]>0,]), r2CD4HVG)

r2cd8 <- Pt21\_new[,c(10,9)]

r2cd8 <- normalize(r2cd8)

r2CD8HVG <- rownames(r2cd8[r2cd8[,2]>0.00002 & r2cd8[,2] > r2cd8[,1]\*2,])

r2CD8NonHVG <- setdiff(rownames(Pt21\_new[Pt21\_new[,10]>0|Pt21\_new[,9]>0,]), r2CD8HVG)

CD4HVG <- setdiff(rCD4HVG,c(rCD4NonHVG,rCD4GVH,rCD4NonGVH,rCD8HVG,rCD8NonHVG,rCD8GVH,rCD8NonGVH,r2CD8HVG,r2CD8NonHVG))

CD4NonHVG <- setdiff(rCD4NonHVG,c(rCD4HVG,rCD4GVH,rCD4NonGVH,rCD8HVG,rCD8NonHVG,rCD8GVH,rCD8NonGVH,r2CD8HVG,r2CD8NonHVG))

CD8HVG <- setdiff(rCD8HVG,c(rCD4NonHVG,rCD4GVH,rCD4NonGVH,rCD4HVG,rCD8NonHVG,rCD8GVH,rCD8NonGVH,r2CD4HVG,r2CD4NonHVG))

CD8NonHVG <- setdiff(rCD8NonHVG,c(rCD4HVG,rCD4GVH,rCD4NonGVH,rCD8HVG,rCD4NonHVG,rCD8GVH,rCD8NonGVH,r2CD4HVG,r2CD4NonHVG))

unmappable <- setdiff(rownames(Pt21\_new), c(rCD4HVG,rCD4NonHVG,rCD4GVH,rCD4NonGVH,rCD8HVG,rCD8NonHVG,rCD8GVH,rCD8NonGVH))

CD4H2VG <- setdiff(r2CD4HVG,c(r2CD4NonHVG,r2CD8HVG,r2CD8NonHVG, rCD8HVG,rCD8NonHVG))

CD4NonH2VG <- setdiff(r2CD4NonHVG,c(r2CD4HVG,r2CD8HVG,r2CD8NonHVG, rCD8HVG,rCD8NonHVG))

CD8H2VG <- setdiff(r2CD8HVG,c(r2CD4NonHVG,r2CD4HVG,r2CD8NonHVG, rCD4HVG,rCD4NonHVG))

CD8NonH2VG <- setdiff(r2CD8NonHVG,c(r2CD4HVG,r2CD8HVG,r2CD4NonHVG, rCD4HVG,rCD4NonHVG))

unmappable2 <- setdiff(rownames(Pt21\_new), c(r2CD4HVG,r2CD4NonHVG,r2CD8HVG,r2CD8NonHVG))

CD4presistentHVG <- intersect(CD4HVG, CD4H2VG)

CD4acquiredHVG <- intersect(CD4NonHVG, CD4H2VG)

CD4denovoHVG <- intersect(unmappable, CD4H2VG)

CD4tolerantHVG <- intersect(CD4HVG, CD4NonH2VG)

CD4persistentNonHVG <- intersect(CD4NonHVG, CD4NonH2VG)

CD4denovoNonHVG <- intersect(unmappable, CD4NonH2VG)

CD4missingHVG <- intersect(CD4HVG, unmappable2)

CD4missingNonHVG <- intersect(CD4NonHVG, unmappable2)

Unmappable <- intersect(unmappable, unmappable2)

CD8presistentHVG <- intersect(CD8HVG, CD8H2VG)

CD8acquiredHVG <- intersect(CD8NonHVG, CD8H2VG)

CD8denovoHVG <- intersect(unmappable, CD8H2VG)

CD8tolerantHVG <- intersect(CD8HVG, CD8NonH2VG)

CD8persistentNonHVG <- intersect(CD8NonHVG, CD8NonH2VG)

CD8denovoNonHVG <- intersect(unmappable, CD8NonH2VG)

CD8missingHVG <- intersect(CD8HVG, unmappable2)

CD8missingNonHVG <- intersect(CD8NonHVG, unmappable2)

matrix1 <- as.data.frame(matrix(nrow=17,ncol=17))

rownames(matrix1) <- c("CD4presistentHVG","CD4acquiredHVG","CD4denovoHVG",

"CD4tolerantHVG","CD4persistentNonHVG","CD4denovoNonHVG",

"CD4missingHVG","CD4missingNonHVG","Unmappable",

"CD8presistentHVG","CD8acquiredHVG","CD8denovoHVG",

"CD8tolerantHVG","CD8persistentNonHVG","CD8denovoNonHVG",

"CD8missingHVG","CD8missingNonHVG") -> colnames(matrix1)

subsets <- list(CD4presistentHVG,CD4acquiredHVG,CD4denovoHVG,

CD4tolerantHVG,CD4persistentNonHVG,CD4denovoNonHVG,

CD4missingHVG,CD4missingNonHVG,Unmappable,

CD8presistentHVG,CD8acquiredHVG,CD8denovoHVG,

CD8tolerantHVG,CD8persistentNonHVG,CD8denovoNonHVG,

CD8missingHVG,CD8missingNonHVG)

for (m in 1:17){

for (n in 1:17) {

matrix1[m,n] <- length(intersect(subsets[[m]], subsets[[n]]))

}

}

write.table(matrix1, file ="~/Desktop/10x TRM 2021/pre post MLR analysis/Pt21 17 subsets uniqueclonenubmer\_JF.tsv",row.names=T,col.names=T, sep="\t")

matrix2 <- as.data.frame(matrix(nrow=17\*4,ncol=(ncol(Pt21\_new)-2)))

rownames(matrix2) <- paste0(rep(c("CD4presistentHVG","CD4acquiredHVG","CD4denovoHVG",

"CD4tolerantHVG","CD4persistentNonHVG","CD4denovoNonHVG",

"CD4missingHVG","CD4missingNonHVG","Unmappable",

"CD8presistentHVG","CD8acquiredHVG","CD8denovoHVG",

"CD8tolerantHVG","CD8persistentNonHVG","CD8denovoNonHVG",

"CD8missingHVG","CD8missingNonHVG"),4), c(rep("template#", 17), rep("template%", 17), rep("uniqueclone#", 17), rep("uniqueclone%", 17)))

colnames(matrix2) <- colnames(Pt21\_new[,4:ncol(Pt21\_new)])

length1 <- function(x){

return(length(which(x>0)))

}

Pt21\_new\_v2 <- Pt21\_new

Pt21\_new\_v2 <- Pt21\_new\_v2[rownames(Pt21\_new\_v2) %in% c(CD4presistentHVG,CD4acquiredHVG,CD4denovoHVG,

CD4tolerantHVG,CD4persistentNonHVG,CD4denovoNonHVG,

CD4missingHVG,CD4missingNonHVG,Unmappable,

CD8presistentHVG,CD8acquiredHVG,CD8denovoHVG,

CD8tolerantHVG,CD8persistentNonHVG,CD8denovoNonHVG,

CD8missingHVG,CD8missingNonHVG), ]

for (m in 1:17) {

for (n in 4:ncol(Pt21\_new\_v2)) {

matrix2[m,n-2] <- sum(Pt21\_new\_v2[subsets[[m]], n])

matrix2[m+17,n-2] <- sum(Pt21\_new\_v2[subsets[[m]], n])/sum(Pt21\_new\_v2[,n])\*100

matrix2[m+17\*2,n-2] <- length1(Pt21\_new\_v2[subsets[[m]], n])

matrix2[m+17\*3,n-2] <- length1(Pt21\_new\_v2[subsets[[m]], n])/length1(Pt21\_new\_v2[, n])\*100

}

}

write.table(matrix2, file ="~/Desktop/10x TRM 2021/pre post MLR analysis/Pt21 17 subsets number% in uniqueclone and template\_JF\_2.tsv",row.names=T,col.names=T, sep="\t")

### matrix first row: column names need slightly adjust. calculation takes about ~10-15mins

#total template counts

apply(Pt21\_new[,c(4:29)],2,sum)

#generate data matrix

matrix<-apply(Pt21\_new[,c(4:29)],2, sum)

write.table(matrix,file ="~/Desktop/10x TRM 2021/pre post MLR analysis/Pt21 total template counts.tsv",quote=F,row.names=T,col.names=T, sep="\t")