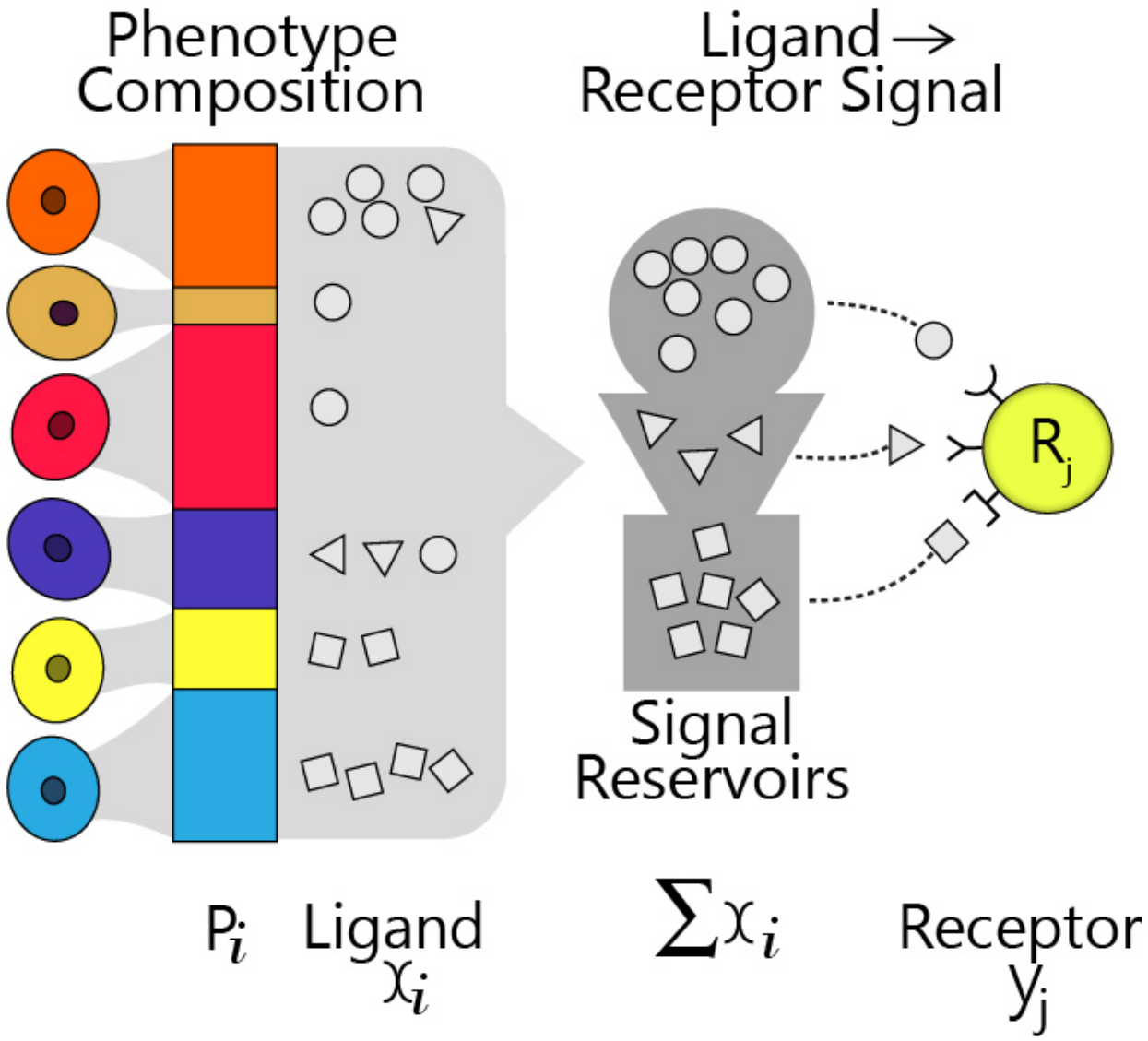
Twister example

Jason Griffiths

7 February 2022

Background on TWISTER

We developed TWISTER to uncover networks of communication pathways between populations of cancer and normal cells within a tumor microenvironemnt. Using single-cell data, we identify and annotate phenotypically distinct cell types and reveal the communication between cancer and normal cells. Unlike existing models that measure cell-cell communication between individual cells,TWISTER measures ecosystem-wide combined signaling to a receiving cell from the diverse normal cell sub- populations and heterogeneous cancer lineages. This reveals how both phenotypic and compositional changes modify communication and impact treatment response. Further, this approach contrasts the communication states across many biopsies, rather than being limited to individual samples, providing comparative insights into communication trends during treatment and identifying pathways that distinguish resistant vs. sensitive tumors.



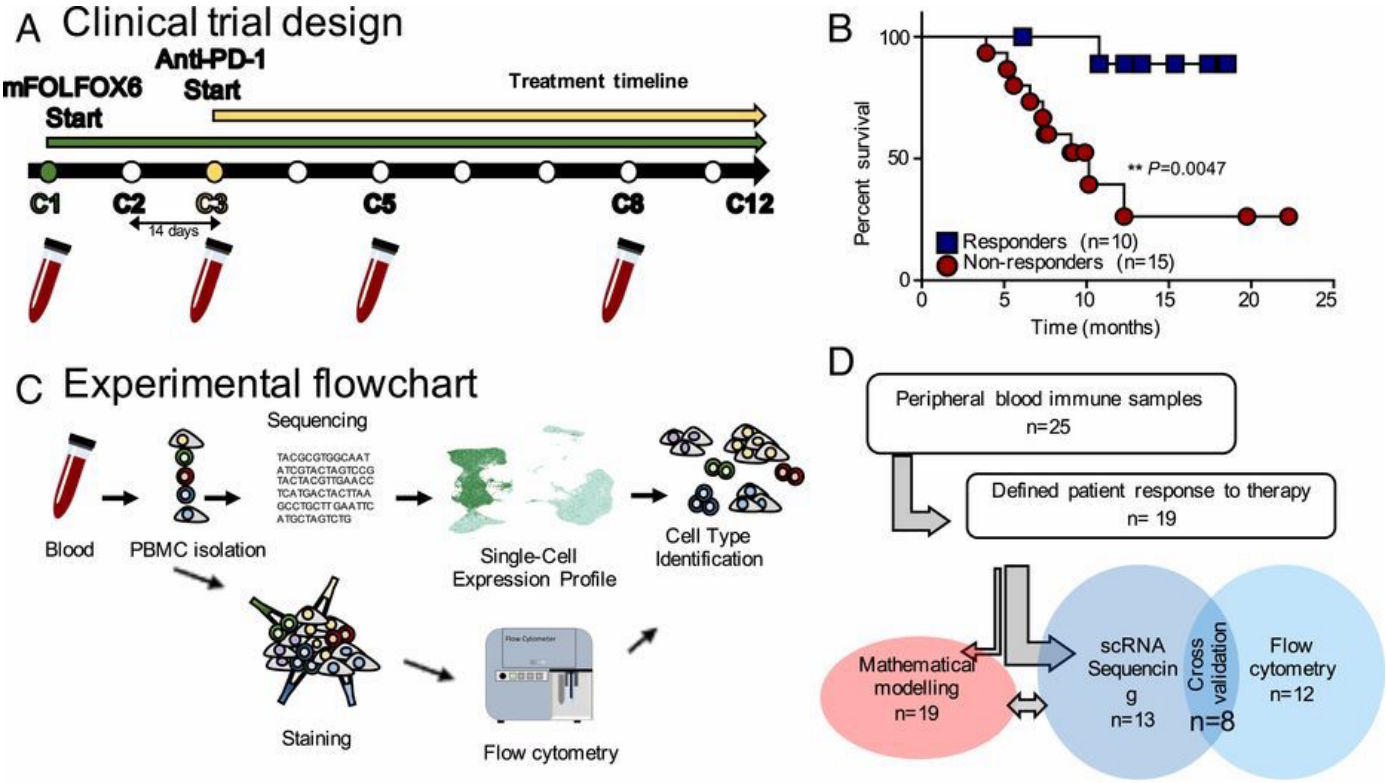
Tumor-wide Integration of Signaling to Each Receiver: TWISTER

The example dataset

To show how the TWISTER communication analysis can be run, we will use data from one of our previously published papers exploring the phenotypic evolution of circulating immune cells of responsive and non-responsive gastro-intestinal tumors during immunotherapy (https://www.pnas.org/content/117/27/16072.short (https://www.pnas.org/content/117/27/16072.short)). In this study, peripheral blood samples were taken before, during and after treatment and single cell RNA sequencing was performed on all samples of a cohort of 13 patients.

Cell type identfication and annotation was performed using an immune classifier, umap analysis and marker gene assessment. Cell annotations were verified using two public datasets, based on the consistency of transcriptional profiles. Using this appraoch, cellular phenotypic states were defined with high resolution, including a diversity of T cell activation states (naive, memory and effector) and different types of myeloid cells (monocytes, M1/M2 macrophages and dendritic cells). The heterogeneity of cell phenotypes was characterized using UMAP analysis. For each of the 70781 cells we have a profile of gene expression, a cell type annotation and a umap phenotype characterization.

We can explore communication within these tumors, using this scRNA seq information and a data base of cell-cell ligand-receptor interactions (published by Ramilowski et al 2015).



Phenotypic evolution of peripheral immune cells during immunotherapy

Setting up the R environment

Load some packages.

```
rm(list=ls())
require(abind); require(data.table); require(dplyr); require(ggplot2); require(tidy
r); require(igraph); require(parallel); require("ggalluvial")
```

Reading in data

We need the follwoing datasets: 1) A data base of established communication pathways indicating the ligand and receptor genes involved in each pathway.

2. Cell metadata indiating the fine resolution cell type annotation of each cell in the sample cohort and the clinical metadata that goes with the cells/samples.
3. Count per million (CPM) gene expression.
4. Phenotype landscape Umap coordinates.

```
# Load Ligand Receptor database list of Ramilowski et al 2015
load( "/Users/jason/Dropbox/Cancer_pheno_evo/data/FELINE2/LigandReceptor/Filtered_Human-2015-Ramilowski-LR-pairs.RData")
LRgenelist <- unique( c(LRpairsFiltered$HPMR.Receptor, LRpairsFiltered$HPMR.Ligand) )
LRpairsFiltered[1:5,] %>% dplyr::select(Pair.Name, HPMR.Ligand, HPMR.Receptor)
```

```
##       Pair.Name HPMR.Ligand HPMR.Receptor
## 1:     A2M_LRP1         A2M          LRP1
## 2: AANAT_MTNR1A       AANAT        MTNR1A
## 3: AANAT_MTNR1B       AANAT        MTNR1B
## 4:    ACE_AGTR2         ACE         AGTR2
## 5:   ACE_BDKRB2         ACE        BDKRB2
```

```
# Load cell metadata with fine resolution cell subtype annotation and clinical information
load( file= "/Users/jason/Dropbox/PD1 Analysis/Lance/PD1_combined/PD1_paper_cells_analysed.RData")
Meta.dd2[1, ]
```

```
##                        Cell.ID Time.Point     Responder rowID Cluster
## 1: 14546X1_S1_AAACCTGCAAAGCAAT         C1 Non.Reponsder     1      T0
##    Patient.ID Major_cluster   Sub_cluster Cell.class.for.normalisation
## 1:     HJD33E        T_Cell T_Cell_CD4_EM                   lymphocyte
##    Contaminant Intermediate_classification_AB Intermediate_classification_2_0
## 1:       FALSE                     T_Cell_CD4                  T_Cell_CD4_EM
##    Intermediate_Clusters_ML
## 1:               T_Cell_CD4
```

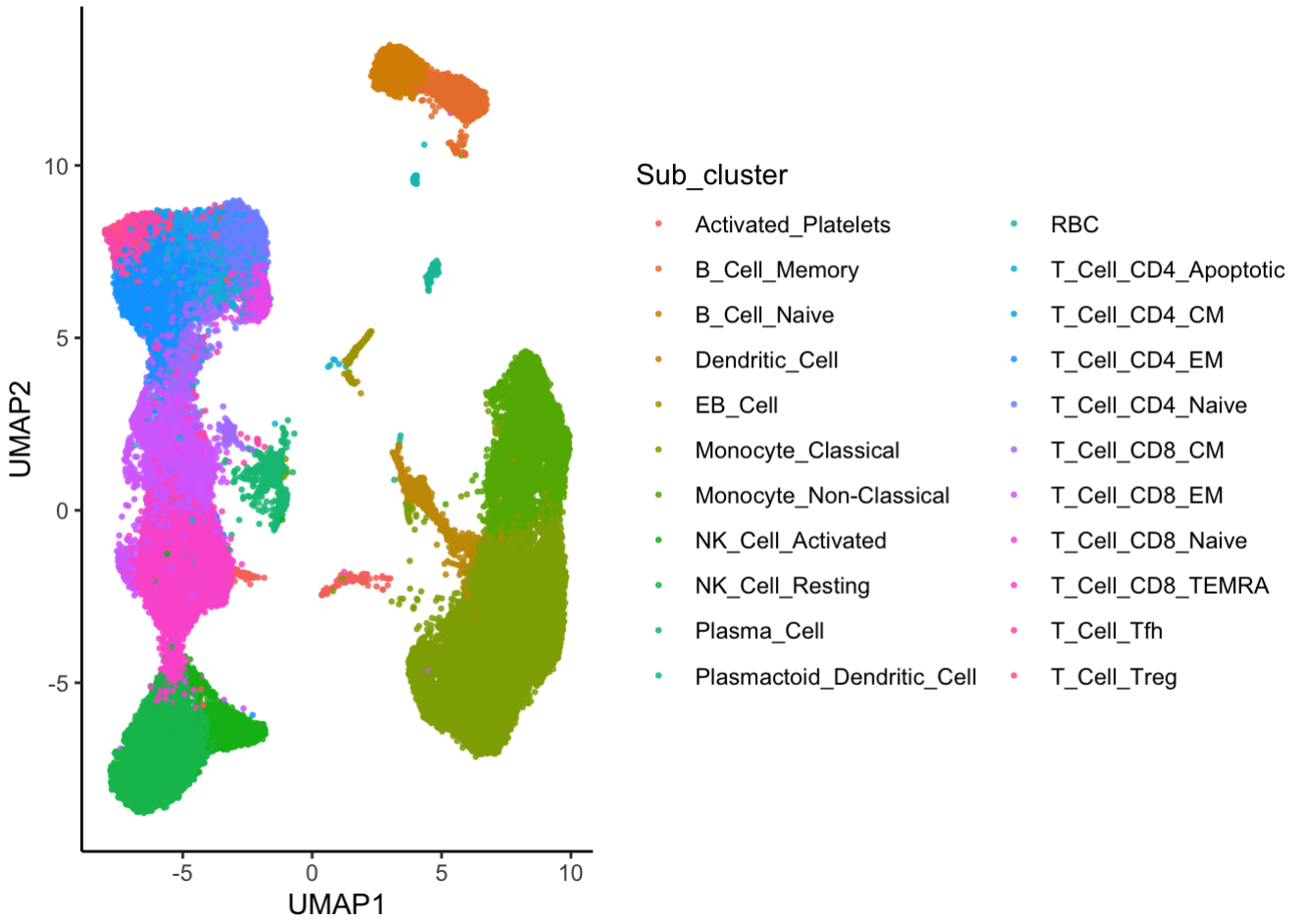
```
# Read count per million (CPM) gene expression data and join umap locations to this data or the metadata
CPM  <- fread(file= "/Users/jason/Dropbox/PD1 Analysis/Lance/PD1_combined/Raw Counts/PD1.RawCounts.wUMAPCoords.txt" )
CPM[1:10, 1:10]
```

```
##         UMAP1       UMAP2                    Cell.ID RP11.34P13.7 RP11.34P13.8
##  1: -7.049916  6.21744633 14546X1_S1_AAACCTGCAAAGCAAT            0            0
##  2: -5.962549 -5.81142330 14546X1_S1_AAACCTGCATCGATTG            0            0
##  3: -4.080601 -1.00315988 14546X1_S1_AAACCTGGTAAGGGCT            0            0
##  4: -3.504583 -0.98200071 14546X1_S1_AAACGGGTCCAATGGT            0            0
##  5: -5.601706  6.02359104 14546X1_S1_AAAGATGTCCGAACGC            0            0
##  6: -3.813487 -0.07468483 14546X1_S1_AAAGATGTCTAGCACA            0            0
##  7: -2.022975 -6.34682274 14546X1_S1_AAAGCAACACACGCTG            0            0
##  8: -4.639802  8.28241920 14546X1_S1_AAAGCAACAGGATCGA            0            0
##  9: -5.118358  8.36736965 14546X1_S1_AAAGTAGCAGGATCGA            0            0
## 10: -7.230830 -7.69101620 14546X1_S1_AAAGTAGCATTAGCCA            0            0
##     AL627309.1 AP006222.2 RP4.669L17.10 RP11.206L10.3 RP11.206L10.5
##  1:          0   0.000000             0             0             0
##  2:          0   0.000000             0             0             0
##  3:          0   0.000000             0             0             0
##  4:          0   0.000000             0             0             0
##  5:          0   0.000000             0             0             0
##  6:          0   0.000000             0             0             0
##  7:          0   0.000000             0             0             0
##  8:          0   0.000000             0             0             0
##  9:          0   1.033635             0             0             0
## 10:          0   0.000000             0             0             0
```

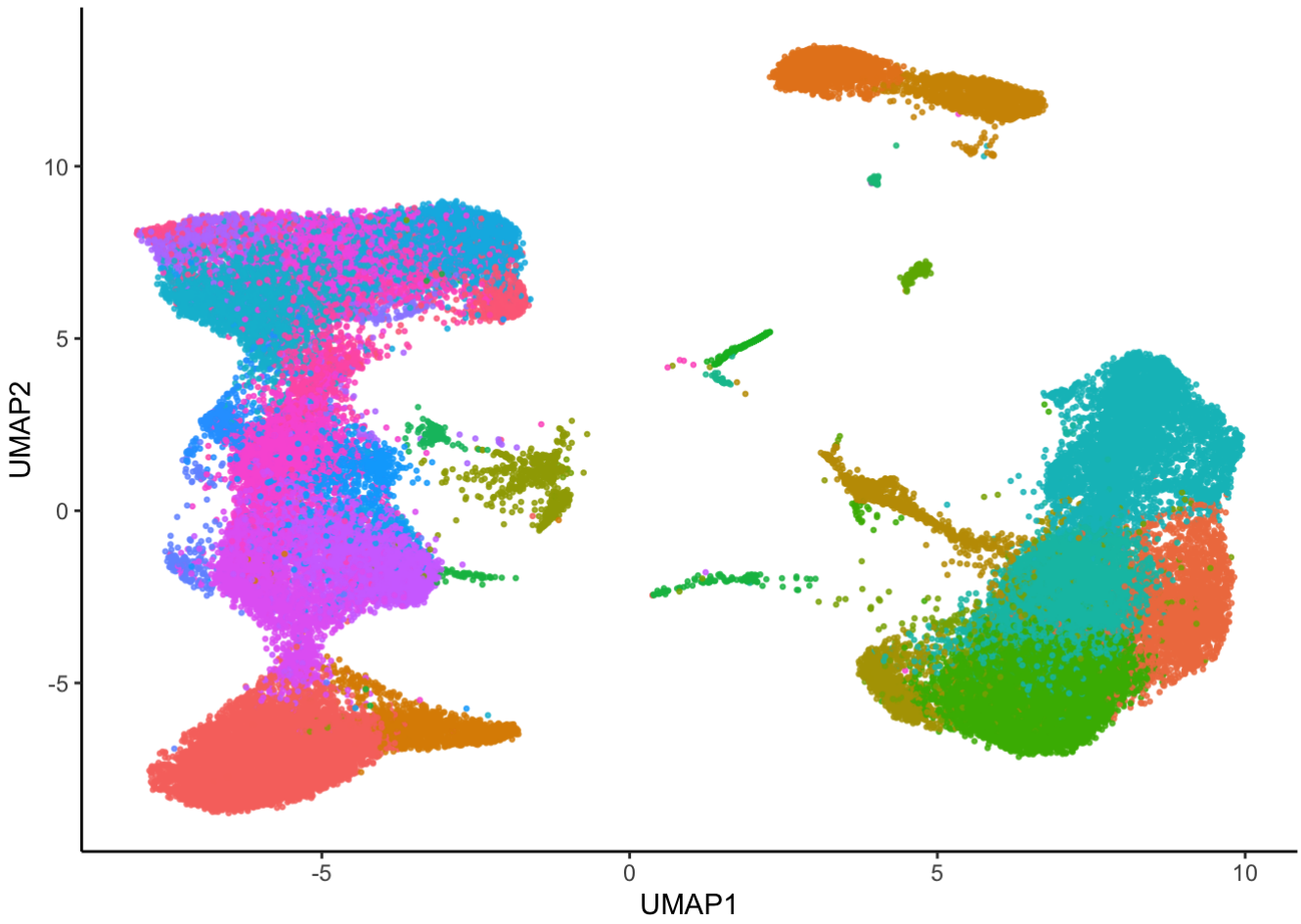
Visualize cellular phenotypic heterogeneity

Examine the umap phenotype landscape and add cell type annotations.

```
# Visualize cell type heterogeneity at different resolutions
umap_vis_dd <- merge( CPM %>% dplyr::select(Cell.ID, UMAP1, UMAP2) , Meta.dd2 , by=
"Cell.ID")
ggplot(umap_vis_dd, aes(UMAP1, UMAP2, col= Sub_cluster)) + theme_classic() + geom_poi
nt(alpha= 0.8, size= 0.5)
```




```
ggplot(umap_vis_dd, aes(UMAP1, UMAP2, col= Cluster)) + theme_classic() + geom_point(a
lpha= 0.8, size= 0.5) + theme(legend.position= "none")
```



Specify the patient and timepoint codes to select cells from one sample of a tumor

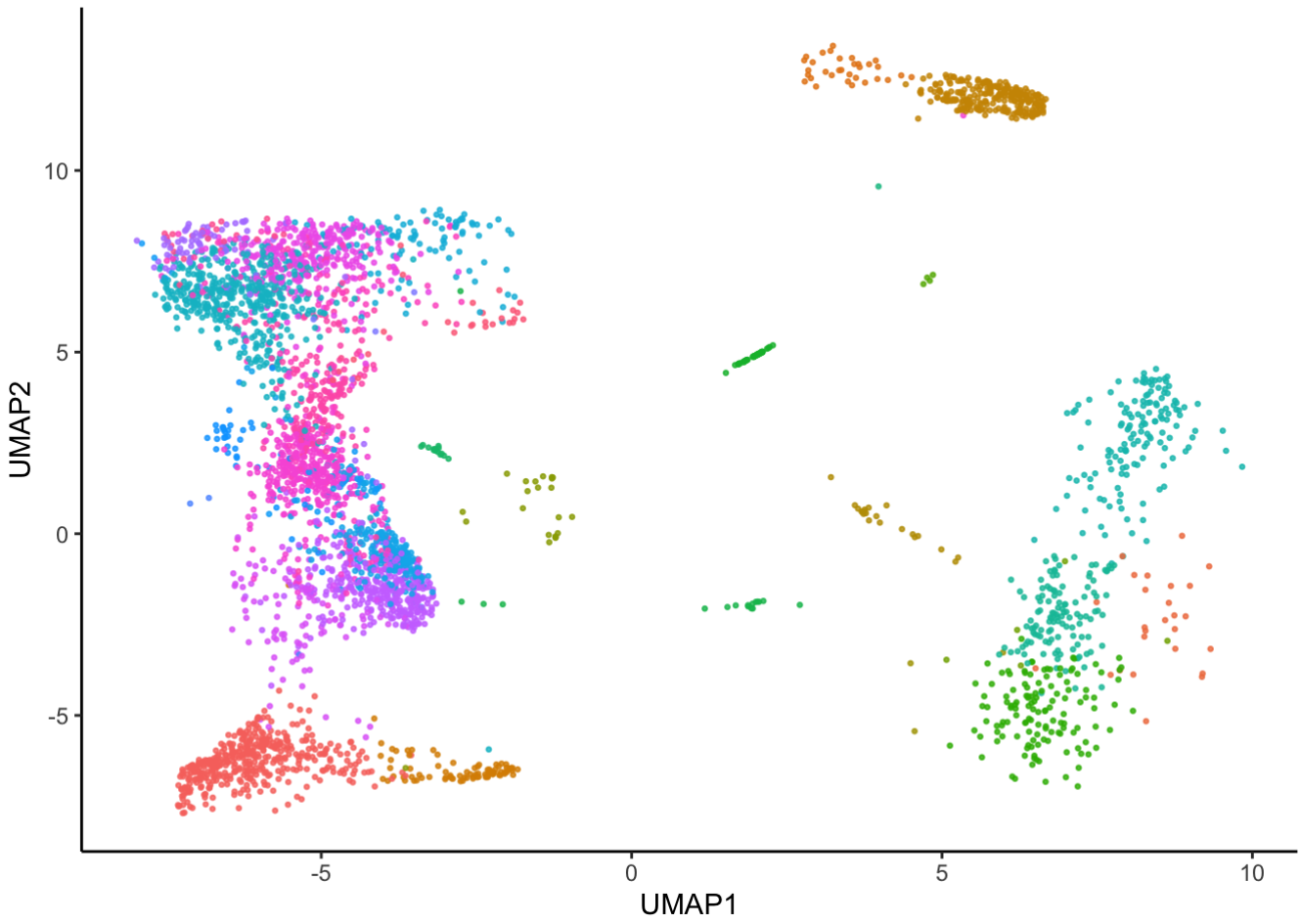
This parameter seting can be used to cycle through all samples when anaysing a cohort of samples

```
pars <- c(Patient = "HJD33E", TimePoint="C1")
```

Subset the data

Extract cell metadata and cpm data for cells from one sample of a tumor.

```
# Extract subset of cells sampled from one patient: meta and CPM data
WhichCells <- Meta.dd2[Patient.ID %in% pars["Patient"]]
CPMsubset <- CPM[Cell.ID %in% WhichCells$Cell.ID,] # CPMsubset[1:10,1:10]
rm(list="CPM")


### Unique units (uu): clusters of phenotypically similar cells (all cell types) and
 their umap discretization level
uu <- unique( WhichCells %>% dplyr::select( c("Major_cluster", "Intermediate_Clusters
_ML" ,"Sub_cluster","Cluster") ) )
ggplot(umap_vis_dd[Cell.ID %in% WhichCells$Cell.ID,], aes(UMAP1, UMAP2, col=Cluster))
+ theme_classic() + geom_point(alpha=0.8, size=.5) + theme(legend.position="none")
```



Extract ligand-receptor expression data to use in communication analysis & identify which communication pathways we can measure

First extract genes in the CPM data that are listed in the ligand-receptor database as being involved in ligand-receptor communication pathways. Then check that both the ligand and the receptor of each communication pathway are present in the CPM dataset. Keep communication pathways for which we have data on both the ligand and receptor.

```
### Extract Ligand and Receptor genes in the L-R database that are present in the CPM
data
# Subset Ligand Receptor genes and umap coordinates from CPM data
LRcpm <- CPMsubset[, c("UMAP1", "UMAP2", "Cell.ID", LRgenelist[ LRgenelist %in% names
(CPMsubset) ]), with= FALSE] #LRcpm[1:3,1:4]    # select just the ligand receptor gen
e expression
# List the genes in the LR cm dataset
LRGene.ID <- LRgenelist[ LRgenelist %in% names(CPMsubset) ]
rm(list= "CPMsubset")


# Identify which receptor and ligand pairs are both represented in the dataset
# copy the dataset as we will add to it an indicator if each ligand/receptor is prese
nt and then retain those cases where both are present
LRpairsFiltered_i <- LRpairsFiltered
LRpairsFiltered_i[, LigandPresent:= 0 ]
LRpairsFiltered_i[, ReceptorPresent:= 0 ]
LRpairsFiltered_i[LRpairsFiltered_i$HPMR.Ligand %in% LRGene.ID, LigandPresent:= 1 ]
LRpairsFiltered_i[LRpairsFiltered_i$HPMR.Receptor %in% LRGene.ID, ReceptorPresent:= 1
]
# Selecte the ligand-receptor gene pairs for which we can calculate communication sco
res because we have both genes
LRpairsFiltered2_i <- LRpairsFiltered_i[ LigandPresent== 1 & ReceptorPresent== 1 ]
LRpairsFiltered2_i[1:10,]
```

```
##        Pair.Name                      Ligand.Name
##  1:     A2M_LRP1           alpha-2-macroglobulin
##  2:   ADAM10_AXL ADAM metallopeptidase domain 10
##  3: ADAM12_ITGA9 ADAM metallopeptidase domain 12
##  4: ADAM12_ITGB1 ADAM metallopeptidase domain 12
##  5:  ADAM12_SDC4 ADAM metallopeptidase domain 12
##  6: ADAM15_ITGA5 ADAM metallopeptidase domain 15
##  7: ADAM15_ITGA9 ADAM metallopeptidase domain 15
##  8: ADAM15_ITGAV ADAM metallopeptidase domain 15
##  9: ADAM15_ITGB1 ADAM metallopeptidase domain 15
## 10: ADAM15_ITGB3 ADAM metallopeptidase domain 15
##
Receptor.Name
##  1:                                          low density lipoprotein receptor-rel
ated protein 1
##  2:                                                                   AXL receptor t
yrosine kinase
##  3:                                                                              int
egrin, alpha 9
##  4: integrin, beta 1 (fibronectin receptor, beta polypeptide, antigen CD29 include
s MDF2, MSK12)
##  5:
syndecan 4
##  6:                                integrin, alpha 5 (fibronectin receptor, alph
a polypeptide)
##  7:                                                                              int
egrin, alpha 9
##  8:                                                                              int
egrin, alpha V
##  9: integrin, beta 1 (fibronectin receptor, beta polypeptide, antigen CD29 include
s MDF2, MSK12)
## 10:                                integrin, beta 3 (platelet glycoprotein IIIa,
antigen CD61)
##     HPMR.Ligand HPMR.Receptor Pair.Source        Pair.Evidence LigandPresent
##  1:         A2M          LRP1       known literature supported             1
##  2:      ADAM10           AXL       novel literature supported             1
##  3:      ADAM12         ITGA9       known literature supported             1
##  4:      ADAM12         ITGB1       known literature supported             1
##  5:      ADAM12          SDC4       known literature supported             1
##  6:      ADAM15         ITGA5       known literature supported             1
##  7:      ADAM15         ITGA9       known literature supported             1
##  8:      ADAM15         ITGAV       novel literature supported             1
##  9:      ADAM15         ITGB1       known literature supported             1
## 10:      ADAM15         ITGB3       known literature supported             1
##     ReceptorPresent
##  1:               1
##  2:               1
##  3:               1
##  4:               1
##  5:               1
##  6:               1
##  7:               1
##  8:               1
##  9:               1
## 10:               1
```

```
# determine the number of pathways
nrow(LRpairsFiltered2_i)
```

```
## [1] 655
```

Merge cell annotation metadata and ligand-receptor gene expression for cells from a specific timepoint and patient sample

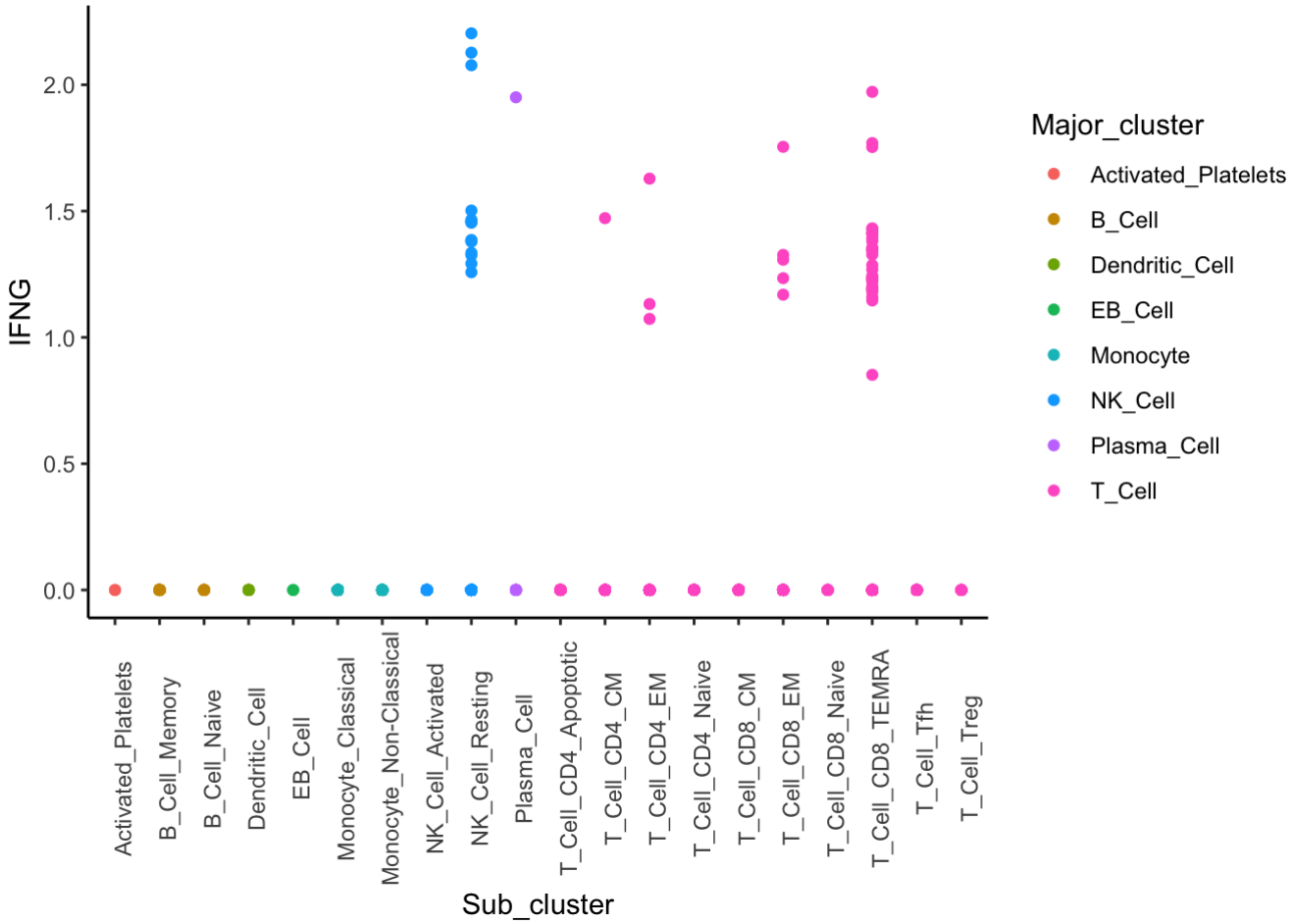
Bring all the different data types together and reorganise the structure.

```
# Specify timepoint
tau <- pars["TimePoint"]
# Phenotype classifications of all cells in the sample of the tumor at this timepoint
phenotypes_i <- WhichCells[Time.Point == tau]
## Extract clinical data of the patient for merging
clin_i <- phenotypes_i[1, ] %>% dplyr::select(Patient.ID, Time.Point, Responder)

# Merge cell metadata (annotaitons and umap location), sample information and express
ion of Ligand-Receptor genes
dd2 <-  merge(phenotypes_i, LRcpm, by= "Cell.ID")
# Count the total number of cells in this sample
dd2[ ,samplesize_it:= nrow(dd2) ]   #dd2[,1:40]
# Gather genes into long format and remove wide copy to save memory
dd3 <- data.table( gather( dd2, gene, expression, all_of(LRGene.ID) ) )
rm(list= c("dd2"))
dd3[1:3,]
```

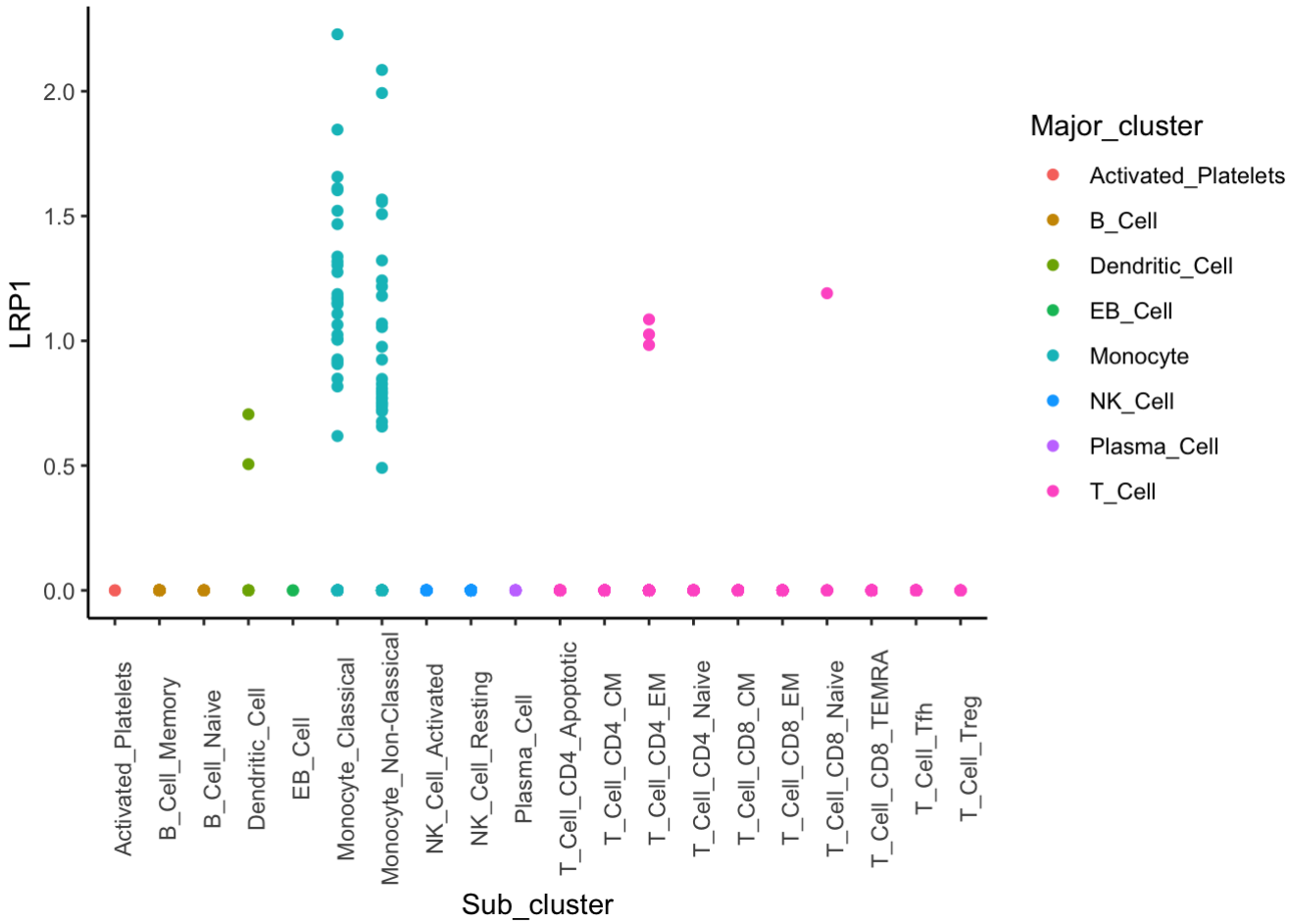
```
##                        Cell.ID Time.Point     Responder rowID Cluster
## 1: 14546X1_S1_AAACCTGCAAAGCAAT         C1 Non.Reponsder     1      T0
## 2: 14546X1_S1_AAACCTGCATCGATTG         C1 Non.Reponsder     2       1
## 3: 14546X1_S1_AAACCTGGTAAGGGCT         C1 Non.Reponsder     3      T5
##    Patient.ID Major_cluster     Sub_cluster Cell.class.for.normalisation
## 1:     HJD33E        T_Cell   T_Cell_CD4_EM                   lymphocyte
## 2:     HJD33E       NK_Cell NK_Cell_Resting                   lymphocyte
## 3:     HJD33E        T_Cell   T_Cell_CD8_EM                   lymphocyte
##    Contaminant Intermediate_classification_AB Intermediate_classification_2_0
## 1:       FALSE                     T_Cell_CD4                   T_Cell_CD4_EM
## 2:       FALSE                        NK_Cell                         NK_Cell
## 3:       FALSE                     T_Cell_CD8                      T_Cell_CD8
##    Intermediate_Clusters_ML     UMAP1     UMAP2 samplesize_it gene expression
## 1:               T_Cell_CD4 -7.049916  6.217446          1343 LRP1          0
## 2:                  NK_Cell -5.962549 -5.811423          1343 LRP1          0
## 3:               T_Cell_CD8 -4.080601 -1.003160          1343 LRP1          0
```

Visualize expression of communication genes in specific cell types

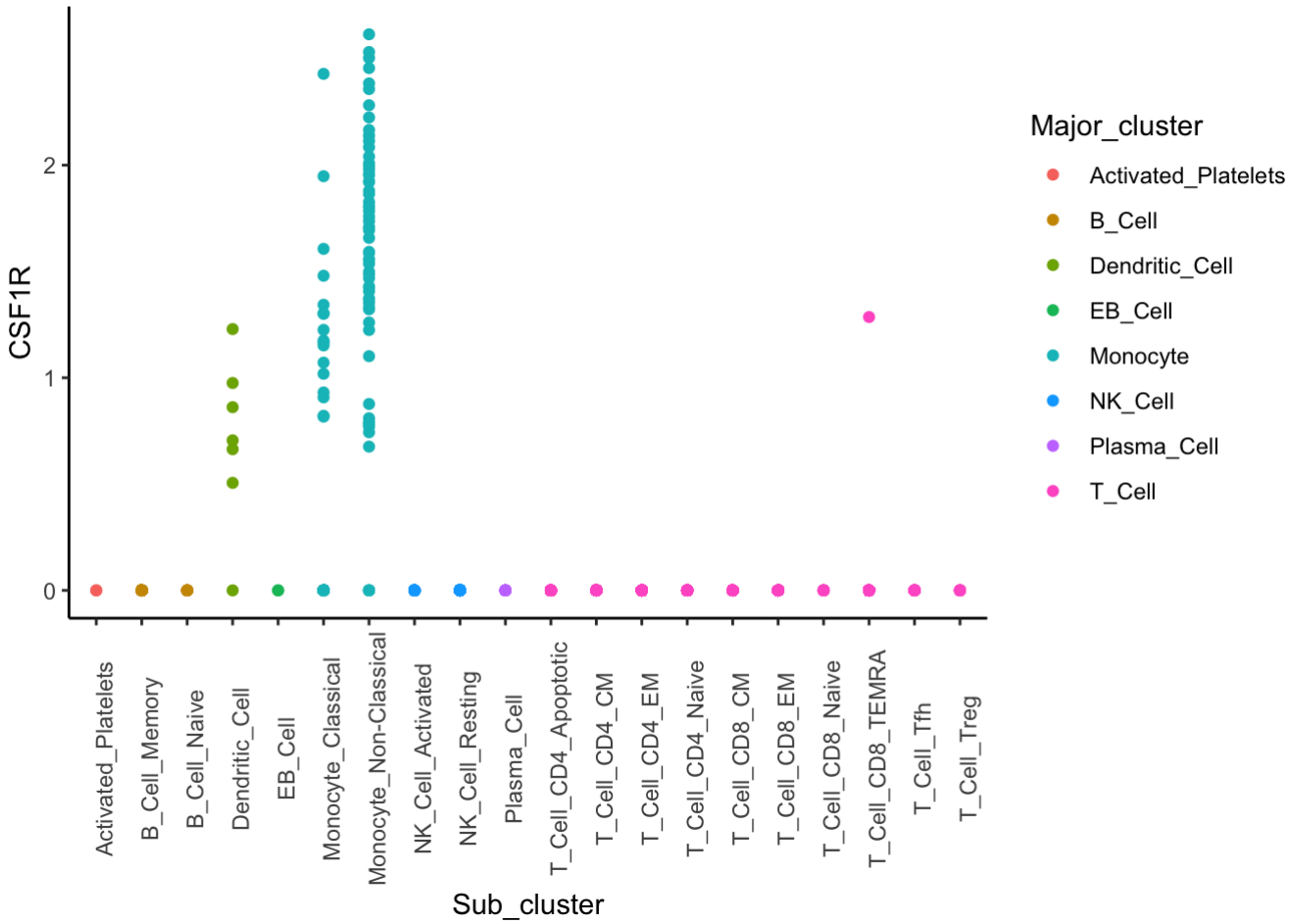
```
# Visualise specific genes as required
ggplot(dd3[gene=="IFNG"], aes(y= expression, x= Sub_cluster, col= Major_cluster) ) +
 geom_point()+ylab("IFNG") + theme_classic()+theme(axis.text.x = element_text(angle=9
0))
```




```
ggplot(dd3[gene=="LRP1"], aes(y= expression, x= Sub_cluster, col= Major_cluster) ) +
 geom_point()+ylab("LRP1") + theme_classic()+theme(axis.text.x = element_text(angle=9
0))
```




```
ggplot(dd3[gene=="CSF1R"], aes(y= expression, x= Sub_cluster, col= Major_cluster) ) +
geom_point()+ylab("CSF1R") + theme_classic()+theme(axis.text.x = element_text(angle=9
0))
```



Calculate the ligand and receptor expression within each cell type and the frequency of each cell type in the sample

To circumvent the issues of sparsity/frop out and noise associated with low read depth, we calculate the average expression of genes for clusters of phenotypically similar cells of each cell type. We track how many of each cell type there are because the numerous cell types will contribute more to communication that other types that are hardly present.

```
### Summarise the average expression of genes in each cell discretization class
grps <- c("gene", "samplesize_it", "Major_cluster", "Intermediate_Clusters_ML" , "Sub
_cluster", "Cluster")
dd4 <- data.table( dd3 %>% dplyr::group_by_(.dots = grps) %>% dplyr::summarise( expre
ssion_bar:= mean(expression), countofvalues = n() ) )
```

```
## `summarise()` has grouped output by 'gene', 'samplesize_it', 'Major_cluster', 'Int
ermediate_Clusters_ML', 'Sub_cluster'. You can override using the `.groups` argument.
```

```
rm(list= "dd3")
```

Translate cell type counts into relative abundances (frequencies)

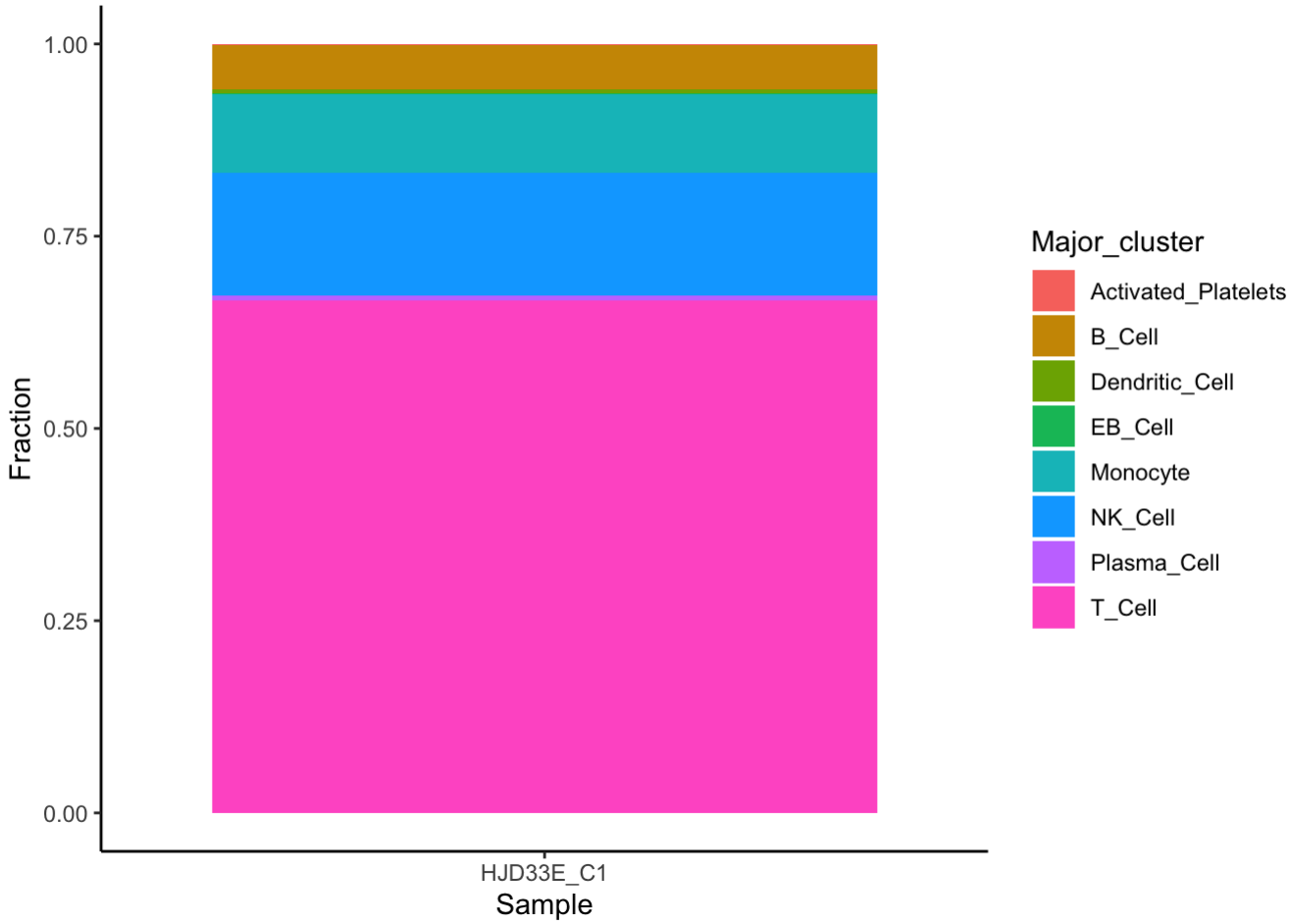
Add in the clinical data (important for contrastingmany samples) and a key for the cell type clusters named “key_”. Communication analyses will use the key to keep track of who is sending and receiving signals from who? At this stage, communication pairs are still not joined and so we have a dataset showing the expression of each ligand or receptor in each cell type (indicated by key_) and quantification of the relative abundance of cell types.

```
## Merge clinical data with average expression data
dd5 <- data.table(clin_i, dd4  )
rm(list= c("dd4"))

# Add the fraction that each celltype contributes to the total tumor sample
dd5[, FracSample:= countofvalues/samplesize_it ]
# Specify that the Cluster column is the key to work with
dd5$key_ <- as.character(dd5$Cluster)

dd5[1:10,]
```

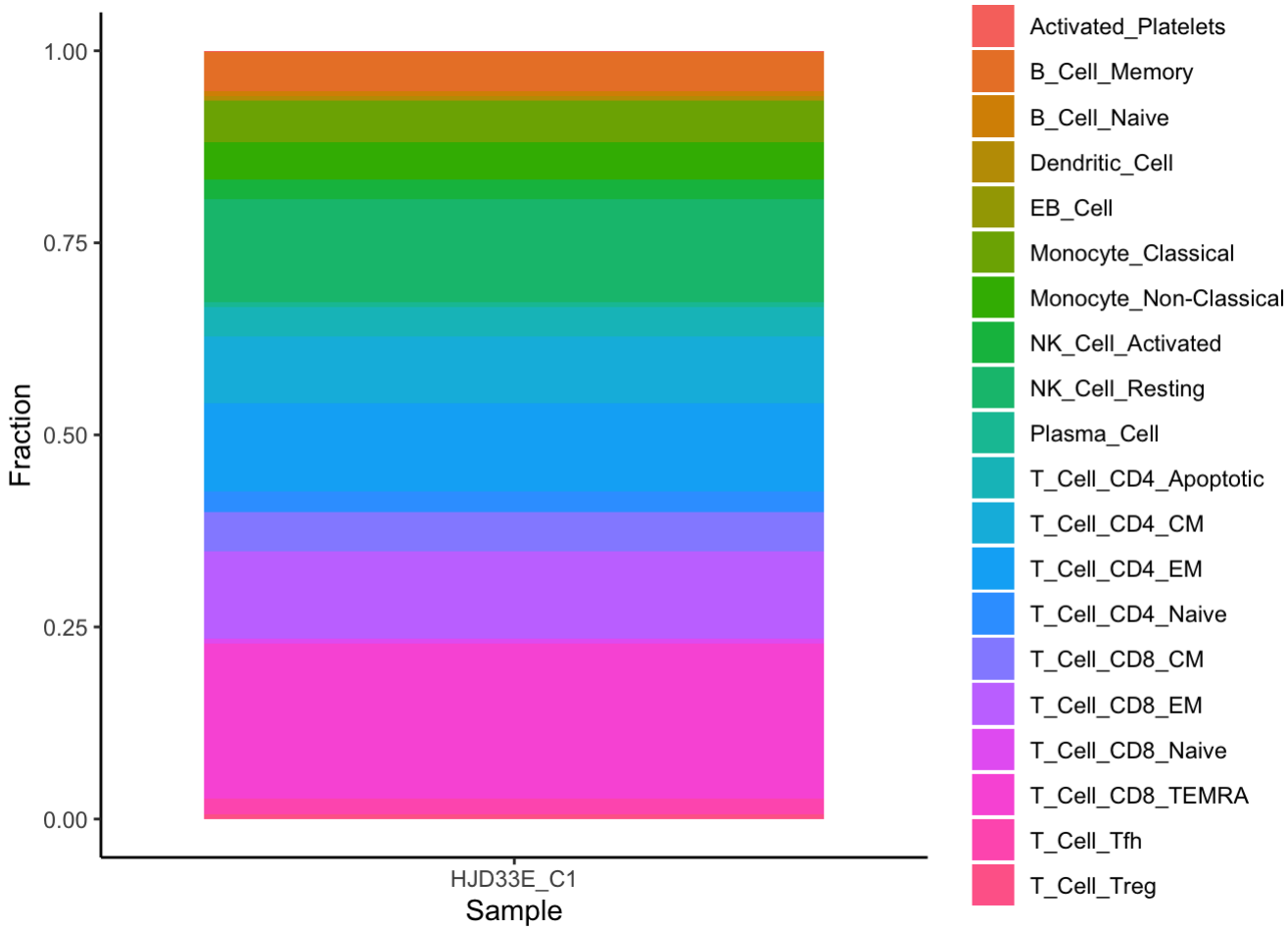
```
##     Patient.ID Time.Point     Responder gene samplesize_it       Major_cluster
##  1:     HJD33E         C1 Non.Reponsder  A2M          1343 Activated_Platelets
##  2:     HJD33E         C1 Non.Reponsder  A2M          1343               B_Cell
##  3:     HJD33E         C1 Non.Reponsder  A2M          1343               B_Cell
##  4:     HJD33E         C1 Non.Reponsder  A2M          1343      Dendritic_Cell
##  5:     HJD33E         C1 Non.Reponsder  A2M          1343              EB_Cell
##  6:     HJD33E         C1 Non.Reponsder  A2M          1343             Monocyte
##  7:     HJD33E         C1 Non.Reponsder  A2M          1343             Monocyte
##  8:     HJD33E         C1 Non.Reponsder  A2M          1343             Monocyte
##  9:     HJD33E         C1 Non.Reponsder  A2M          1343             Monocyte
## 10:     HJD33E         C1 Non.Reponsder  A2M          1343             Monocyte
##     Intermediate_Clusters_ML         Sub_cluster Cluster expression_bar
##  1:      Activated_Platelets Activated_Platelets      22              0
##  2:                  B_Cell       B_Cell_Memory      13              0
##  3:                  B_Cell        B_Cell_Naive      11              0
##  4:           Dendritic_Cell      Dendritic_Cell      14              0
##  5:                 EB_Cell             EB_Cell      20              0
##  6:                Monocyte  Monocyte_Classical      10              0
##  7:                Monocyte  Monocyte_Classical      15              0
##  8:                Monocyte  Monocyte_Classical      18              0
##  9:                Monocyte  Monocyte_Classical       2              0
## 10:                Monocyte  Monocyte_Classical       3              0
##     countofvalues   FracSample key_
##  1:             2 0.0014892033   22
##  2:            69 0.0513775130   13
##  3:             8 0.0059568131   11
##  4:             7 0.0052122115   14
##  5:             2 0.0014892033   20
##  6:             3 0.0022338049   10
##  7:             1 0.0007446016   15
##  8:             2 0.0014892033   18
##  9:            35 0.0260610573    2
## 10:            31 0.0230826508    3
```

Visualize cell type composition of tumor sample

```
# Composition visualization
ggplot( dd5[gene== dd5$gene[1]] , aes( x= FracSample, y= paste( Patient.ID, Time.Poin
t,sep="_"), fill= Major_cluster)) + geom_bar(stat= "identity") + theme_classic() + co
ord_flip() + ylab("Sample") +xlab("Fraction")
```




```
ggplot( dd5[gene== dd5$gene[1]] , aes( x= FracSample, y= paste( Patient.ID, Time.Poin
t,sep="_"), fill= Sub_cluster)) + geom_bar(stat= "identity") +theme_classic() + coord
_flip() + ylab("Sample")+ xlab("Fraction")
```



Measuring communication

Quantification of communication via each Ligand-Receptor communication pathway is done sequentially by measuring the receiving cell type's receptor expression and the production of ligands by each other cell type (given their relative abundance in the tumor).

```
#Run communication analysis
## Pre-define look ups for all of the combinations of gene and cell types (discretiza
tion class)
# Two copies of the look up of unique units (uu) where we will modify column names in
2 different ways
lu2 <- uu %>% dplyr::select("Intermediate_Clusters_ML" , "Sub_cluster", "Cluster")
setnames(lu2, old= c("Cluster", "Sub_cluster", "Intermediate_Clusters_ML"), new= c("R
eceptor", "ReceptorCelltype", "ReceptorPhenoCelltype"))
lu2i <- uu %>% dplyr::select("Intermediate_Clusters_ML" , "Sub_cluster", "Cluster")
setnames(lu2i, old= c("Cluster", "Sub_cluster", "Intermediate_Clusters_ML"), new= c(
"Ligand", "LigandCelltype", "LigandPhenoCelltype"))
lu2[1:4,]
```

```
##    ReceptorPhenoCelltype ReceptorCelltype Receptor
## 1:            T_Cell_CD4    T_Cell_CD4_EM       T0
## 2:               NK_Cell  NK_Cell_Resting        1
## 3:            T_Cell_CD8    T_Cell_CD8_EM       T5
## 4:            T_Cell_CD8 T_Cell_CD8_TEMRA      T10
```

```
lu2i[1:4,]
```

```
##    LigandPhenoCelltype   LigandCelltype Ligand
## 1:          T_Cell_CD4    T_Cell_CD4_EM     T0
## 2:             NK_Cell  NK_Cell_Resting      1
## 3:          T_Cell_CD8    T_Cell_CD8_EM     T5
## 4:          T_Cell_CD8 T_Cell_CD8_TEMRA    T10
```

```
# Perform communication analysis
Communication <- rbindlist(mclapply( 1:nrow(LRpairsFiltered2_i)   , function(p){
  # Names of ligand and receptor
  LRnms <- unname( unlist( LRpairsFiltered2_i[p][ , HPMR.Receptor, HPMR.Ligand ] ) )
  # Select expression of the pair in each cell subtype
  LRpair_it_dd <-  data.table( dd5[gene %in% LRnms ] %>% spread(gene, expression_bar)
,  LRpairsFiltered2_i[p] %>% dplyr::select(HPMR.Receptor, HPMR.Ligand, Pair.Name))
  setnames(LRpair_it_dd , old= LRnms, new= c("Ligand", "Receptor"))
  setcolorder(LRpair_it_dd,c( names(dd5[1] %>% dplyr::select(-c("gene", "expression_b
ar"))), "Ligand", "Receptor", "HPMR.Receptor", "HPMR.Ligand", "Pair.Name" )  )

  # Calculate the expression of the signaller cell type by multiplying single cell av
erage expression by the cell number
  LRpair_it_dd[ , Ligand_N:= Ligand * FracSample]

  # Caclulate ligand-receptor signalling between each cell type: outer product matrix
-> Signaler on the cols and receiver cell class on the rows
  Rmat <- LRpair_it_dd[, Receptor] %*% t( unlist( LRpair_it_dd[, "Ligand_N"] ) )
  rownames(Rmat) <- colnames(Rmat) <- LRpair_it_dd$key_
  RmatperSignaller <- LRpair_it_dd[, Receptor] %*% t( unlist( LRpair_it_dd[, "Ligand"
] ) )
  rownames(RmatperSignaller) <- colnames(RmatperSignaller) <- LRpair_it_dd$key_

  # Marginalise signalling matrix to calculate signal transduction to cells of each r
eceiver cell type
  LRpair_it_dd[ , Transduction := rowSums(Rmat) ]
  LRpair_it_dd[ , TransductionperSignaller := rowSums(RmatperSignaller) ]

  # Reformat the ligand-receptor signalling matrix into a long dataframe
  Rmatlong <- data.table( gather(as.data.table(Rmat, keep.rownames = T), Ligand, Sign
al, -1) )[Signal > 0]
  setnames(Rmatlong, old= "rn", new= "Receptor")

  # Merge ligand-receptor signalling with cell type information
  Rmatlong2 <- merge( merge(Rmatlong, lu2, by= "Receptor"), lu2i , by= "Ligand")
  # Merge ligand-receptor signalling with transduction data and all clinical informat
ion
  setnames(Rmatlong2, old= c("Ligand", "Receptor"),new= c("key_signaller", "key_"))
  output <- data.table( merge( LRpair_it_dd, Rmatlong2, by= "key_", all.x= T )  )
  return( output )
  #cat(p);cat("    ")
},mc.cores= detectCores()-2))
PatientID <- Communication[1]$Patient.ID
cat("Saving output for :        patient "); cat(PatientID); cat("       time     ") ;
cat(tau)
```

```
## Saving output for :        patient
```

```
## HJD33E
```

```
##        time
```

```
## C1
```

```
savenm <- paste0("ImmuneCommunicationResults__","PatientID_",PatientID,"__", "Day_",t
au,".RData")
saveloc <- "/Users/jason/Dropbox/Cancer_pheno_evo/data/FELINE2/ImmuneCommunicationOut
put2/"
#save( PatientID, Communication, tau,  uu, dd5,file=paste0(saveloc,savenm))
```

Explore communication ouput

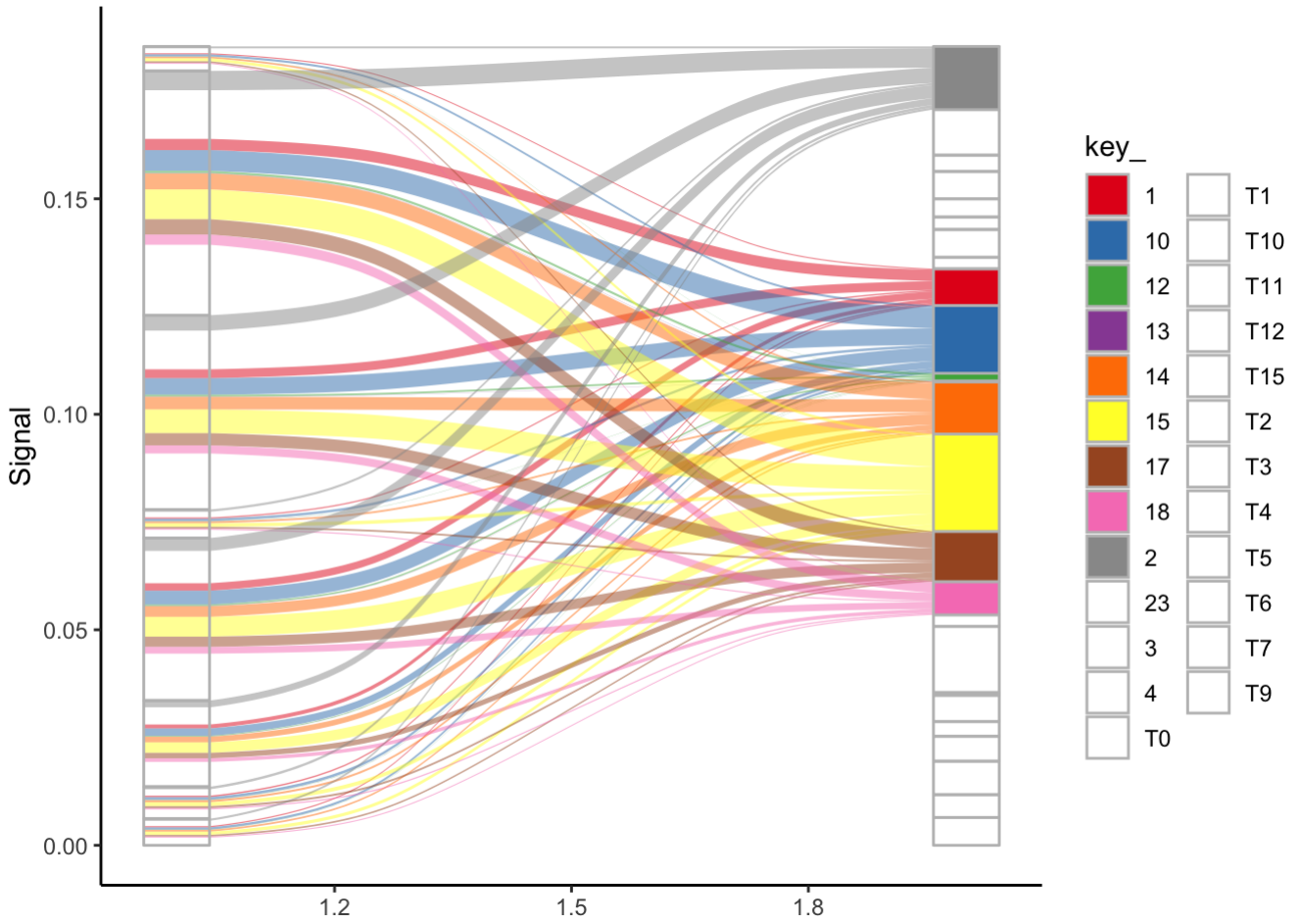
Here I am choosing a very specific example to show how the data is structured. However, the output produces communication scores for all pathways of communication between cell types.

```
#na.omit(Communication)[1:11]
na.omit(Communication)[Pair.Name=="TNF_TNFRSF1A"][ReceptorCelltype=="Monocyte_Classic
al"][LigandCelltype=="Monocyte_Classical"][key_==2&key_signaller==2]
```

```
##    key_ Patient.ID Time.Point     Responder samplesize_it Major_cluster
## 1:    2     HJD33E         C1 Non.Reponsder          1343      Monocyte
##    Intermediate_Clusters_ML        Sub_cluster Cluster countofvalues FracSample
## 1:                Monocyte Monocyte_Classical       2            35 0.02606106
##        Ligand  Receptor HPMR.Receptor HPMR.Ligand    Pair.Name     Ligand_N
## 1: 0.02930103 0.5883893      TNFRSF1A         TNF TNF_TNFRSF1A 0.0007636158
##    Transduction TransductionperSignaller key_signaller       Signal
## 1:   0.01465986                0.2416826             2 0.0004493033
##    ReceptorPhenoCelltype   ReceptorCelltype LigandPhenoCelltype
## 1:              Monocyte Monocyte_Classical            Monocyte
##        LigandCelltype
## 1: Monocyte_Classical
```

Exemplify the communication network

```
 ggplot(na.omit(Communication)[Pair.Name=="TNF_TNFRSF1A"] ,
        aes(y = Signal, axis1 = key_signaller, axis2 = key_,fill= key_)) +theme_class
ic()+
   geom_alluvium( width = 1/12) +#aes(fill = Admit),
   geom_stratum(width = 1/12,  color = "grey")+
   scale_fill_brewer(type = "qual", palette = "Set1")
```




```
rm(list=c("dd5", "lu2", "lu2i", "Communication"))
```

In this dataset, the “Ligand” and “Recpetor” columns indicate the average ligand and receptor expression of the focal cell type. The FracSample column indicatses the relative frequency of the cell type.

key_ = the identifier of the focal signal receiving cells.

key_signaller = the identifier of the signal sending cell population.

Ligand_N = the total signal sent by all cells of that cell type = Ligand*FracSample.

Signal = the communication that the fcal cell receives from communication with a given cell type = Receptor*Ligand_N

Transduction = the total signal received by the focal cell type from all cell types = sum of all signals from all cell types

TransductionperSignaller= individual level variation of this calcualtion = sum_j(Receptor*Ligand_j)

The last four columns are essential for keeping trck of who is communicating with who (ReceptorPhenoCelltype:LigandCelltype).

Calculate community wide communication to individuals of the receiving cell type

The community-wide communication information (CCI) database summarises communications from all cell types to individual cells of all other cell types via each LR communication pathway.

```
# For each sample in the communication output folder, calulate the strength of commun
ciation between each cell type via each communication pathway
filenamesCCI <- list.files(saveloc)
CCI <- rbindlist(lapply(1:length(filenamesCCI), function(ii){
  load(file= paste0(saveloc, filenamesCCI[ii]))
  cat(ii);
  SumComm <- data.table( Communication %>%
                           group_by(Patient.ID, LigandPhenoCelltype, ReceptorPhenoCel
ltype,    Time.Point , Pair.Name, Responder)%>%
                           dplyr::summarise(TransductionMu= weighted.mean(Signal,coun
tofvalues),  Receptor= weighted.mean(Receptor), Ligandtot= sum(Ligand), Ligand_Ntot=
 sum(Ligand_N))  )[!( is.na(LigandPhenoCelltype)|is.na(ReceptorPhenoCelltype) ) ]
}))
```

```
## 1
```

```
## `summarise()` has grouped output by 'Patient.ID', 'LigandPhenoCelltype', 'Receptor
PhenoCelltype', 'Time.Point', 'Pair.Name'. You can override using the `.groups` argum
ent.
```

```
# Scale TME wide communication data (TransductionMu) for each communication pathway t
o make data comparable across communication types
CCI[, scaleTransduction:= scale(TransductionMu, center=F), by= c("Pair.Name")]
CCI[!is.finite(TransductionMu), scaleTransduction:= 0]
#save(CCI,WhichCells, uu, perIndiv, file ="/Users/jason/Dropbox/Cancer_pheno_evo/dat
a/FELINE2/Communication output merged/PopulationCommunicationMerged.RData" )
CCI[1,]
```

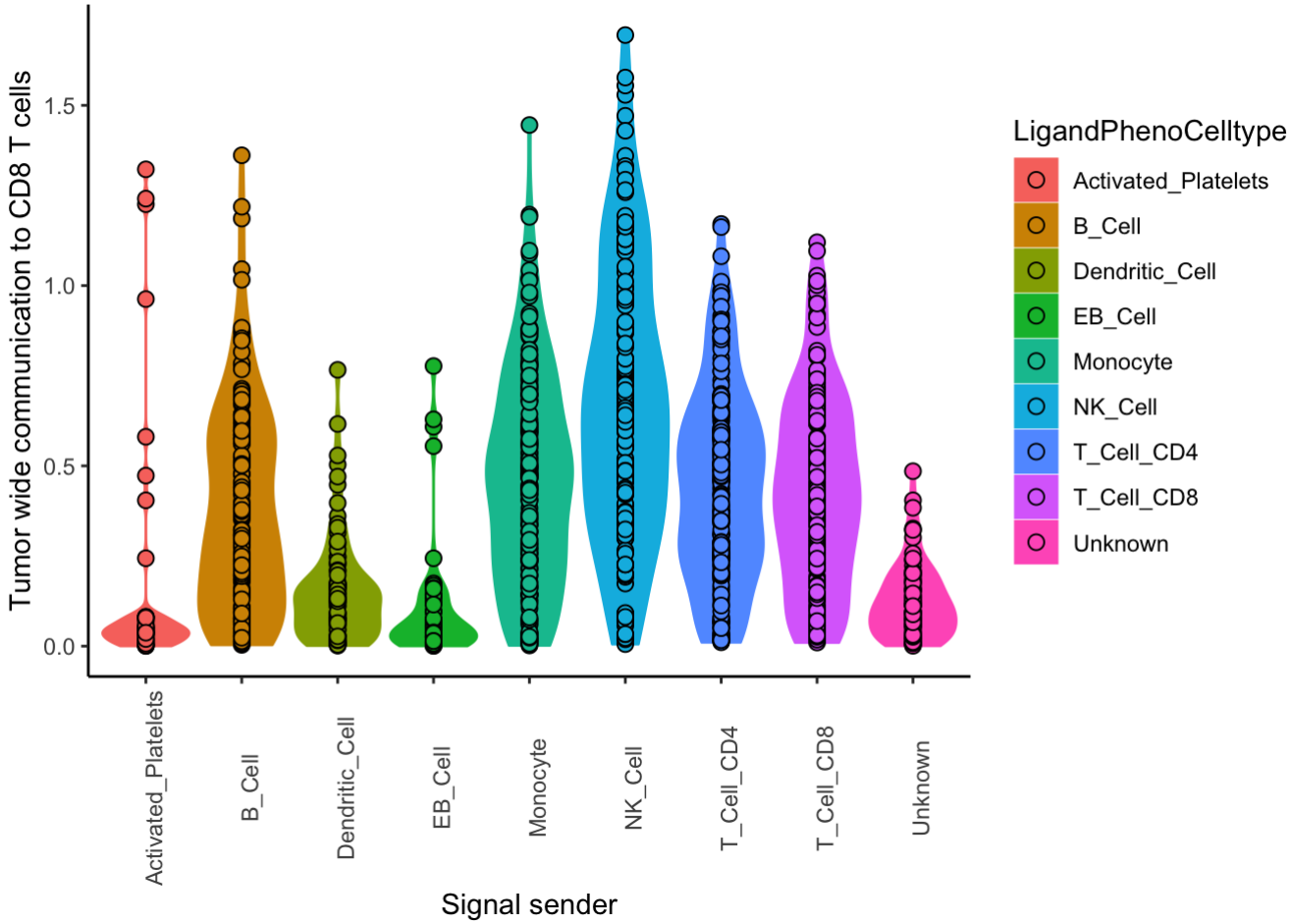
```
##    Patient.ID LigandPhenoCelltype ReceptorPhenoCelltype Time.Point   Pair.Name
## 1:     HJD33E Activated_Platelets   Activated_Platelets         C1 CALR_ITGA2B
##        Responder TransductionMu Receptor Ligandtot Ligand_Ntot
## 1: Non.Reponsder    0.003985008 2.013365  1.329085 0.001979277
##    scaleTransduction
## 1:        0.07976069
```

The most important columns are: 1) TransductionMu This is the amount of communication the average cell of the receiving cell type (indicated by the ReceptorPhenoCelltype column) gets from the population of cells of the signal sending cell type (indicated by the LigandPhenoCelltype). 2) scaleTransduction This standardizes the TransductionMu score to make different communication pathways comparable within the dataset.

Visualize TME communication

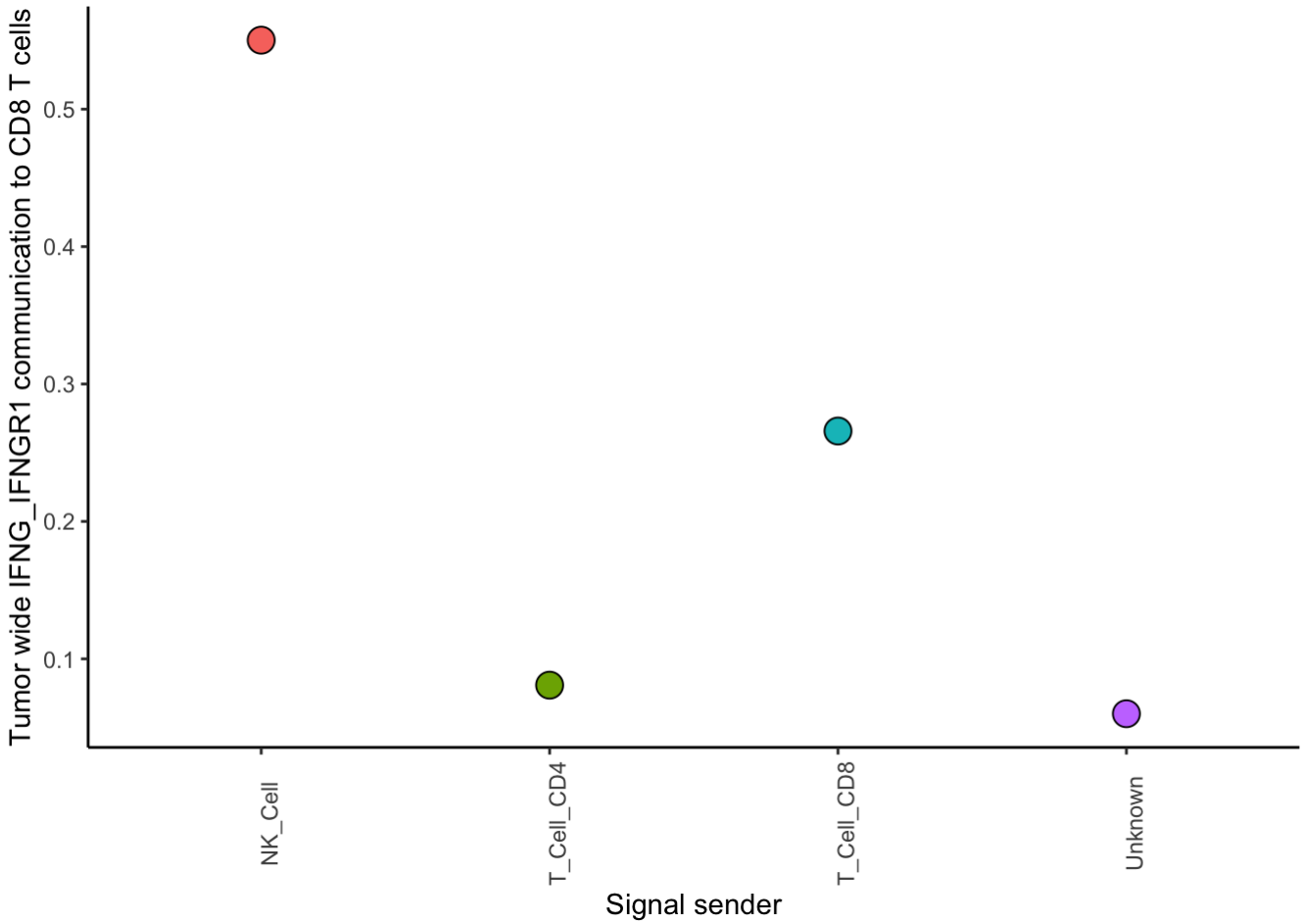
Using the CCI database of communication, we can now start asking many different biological questions. One example is to ask: which cell types are communicating most strongly with a given cell type; say CD8 T cells?

```
#For a given receiver cell type, plot the strength of communications sent by other ce
ll types via all the differnt pathways.
ggplot(CCI[ReceptorPhenoCelltype== "T_Cell_CD8"],
       aes(y= log(1 + scaleTransduction), x= LigandPhenoCelltype , col= LigandPhenoCe
lltype ,fill= LigandPhenoCelltype) ) + theme_classic() +
  geom_violin(scale = "width")+
  geom_point(pch=21, col= "black", size= 2.5) +
  labs(y= "Tumor wide communication to CD8 T cells",x="Signal sender") +
  theme(axis.text.x = element_text(angle=90))
```



Another example is to ask who is sending a signal via a certain communication pathway to a focal cell type (e.g who sends IFNgamma signals to CD8 T cells)?

```
ggplot(CCI[Pair.Name== "IFNG_IFNGR1"][ReceptorPhenoCelltype== "T_Cell_CD8"],
       aes(y= log(1 + scaleTransduction), x= LigandPhenoCelltype ,col= LigandPhenoCel
ltype ,fill= LigandPhenoCelltype) ) + theme_classic() +
  geom_point(pch= 21, col= "black",size= 4.5) + theme(legend.position= "none") +
  labs(y= "Tumor wide IFNG_IFNGR1 communication to CD8 T cells",x= "Signal sender") +
  theme(axis.text.x = element_text(angle=90))
```



Visualize TME wide communication

Use network graphs to visualize comunication. First define a network with nodes that are the cell types and edges that represent communication. The edges are directed, meaning that the communication goes from one cell type to another and is not equal in both directions. We add weights to the edges to represent the strength of communiction

```
# Select data to plot
CCI_plot <- CCI[][Time.Point== pars["TimePoint"]][][ order(LigandPhenoCelltype, Recep
torPhenoCelltype) ]

# Construct directed graph
g <- graph.data.frame(CCI_plot %>% dplyr::select(-Patient.ID), directed= TRUE)
# Add weights to edges of the graph
E(g)$weight <- CCI_plot$scaleTransduction
# Specifcy color of nodes
V(g)$color <- ggsci::pal_npg("nrc")(1)
# Generate circle layout
n <- length( unique(CCI_plot$ReceptorPhenoCelltype) ) -1
pts.circle <- t( sapply(1:n, function(r)c(cos(2*r*pi/n), sin(2*r*pi/n))) )
NodeList <- data.table( c("Dendritic_Cell", "Monocyte", "T_Cell_CD4", "T_Cell_CD8",
"NK_Cell", "B_Cell", "EB_Cell" ,"Activated_Platelets", "Unknown" ) ,
                        c(0, pts.circle[,1] )  ,  c(0, pts.circle[,2] ) )
presloc <- NodeList[na.omit((match(names(V(g) ), NodeList$V1  ))), ]

# Visualize communication with weighted graph
plot.igraph(g, rescale= FALSE,
            layout = as.matrix(presloc %>% select(-V1))   ,
            xlim = c(-1,1), ylim =c(-1,1),
            vertices= NodeList[V1 %in% presloc$V1],
            edge.label.color = adjustcolor("black", 0.5),
            edge.color= adjustcolor("black", 0.5),
            edge.width= 0.1*E(g)$weight )
```