

# iRPCA of e12 coronary progenitors

## Load data

Log-transformed counts per million .Rdata object for the e12 fetal coronary progenitor dataset should be downloaded from Figshare via the following link: <https://ndownloader.figshare.com/files/11086496> MD5: a6a4606279435b694e70e8d6e9997a7c

Save the file in the data/ folder

## all r1

PC13 separates a small number of contaminating Hbb-b1+ cells. Remove these cells from further analysis.

```
cordir='all_r1/'
system(paste0('mkdir ',cordir));
system(paste0('mkdir ',file.path(cordir,'biplots')));

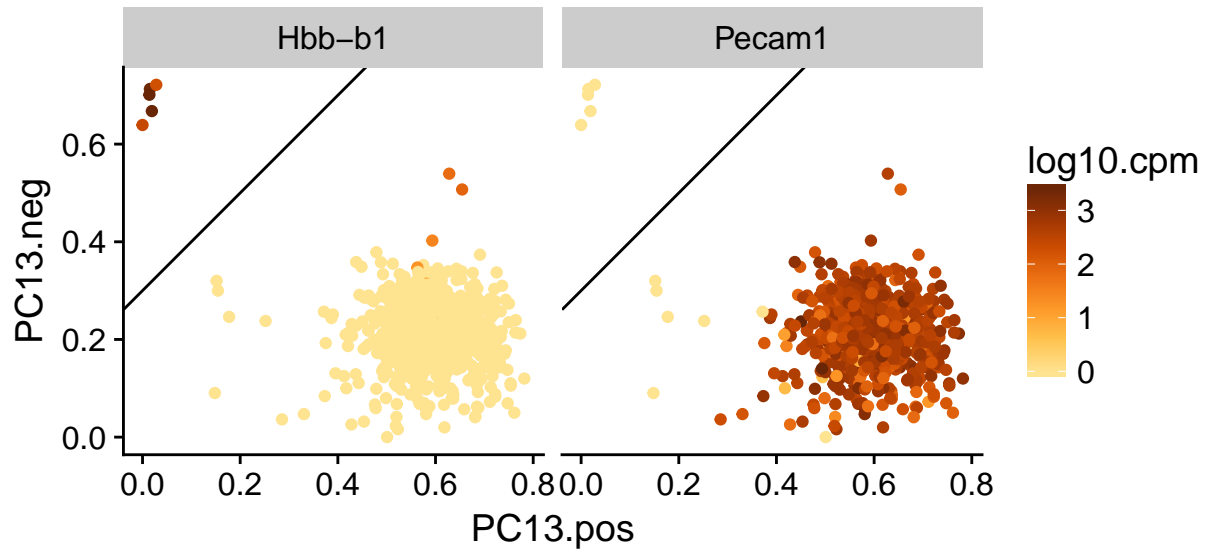
dt.pca = copy(dt.fv1_NS.good)
setkey(dt.pca)

n.pcs = 15
if(recalculate.pca){
  pca.d1 <- dimplots.pca.lowMem(dt.pca,cell.info,cordir,suffix='',max.genes = 30,ncp=n.pcs,
                                min.cells.detect=1,n.cores = 8)
  plot.ggpairs.wExptCols(cell.info,dir=cordir,suffix='',num.pcs = n.pcs,height=25,width=25,sigType = 'S')
}

# remove some hemoglobin-high RBC precursors
pc.scores = fread(paste0(cordir, 'PC_allscores.csv'))
dt.pca.2 = merge(dt.pca[gene %in% c('Hbb-b1','Pecam1')],
                 pc.scores,
                 by='cell.name')

cutoff.point <- 0.3
if(with(dt.pca.2[gene=='Hbb-b1'], cor(log10.cpm, PC13.score)) < 0) cutoff.point <- -cutoff.point

ggplot(dt.pca.2, aes(PC13.pos, PC13.neg, color = log10.cpm)) + geom_point() +
  geom_abline(intercept = -cutoff.point) +
  scale_color_gradientn(colours=brewer.pal(9,'YlOrBr')[3:9]) +
  facet_wrap(~gene)+
  coord_fixed()
```

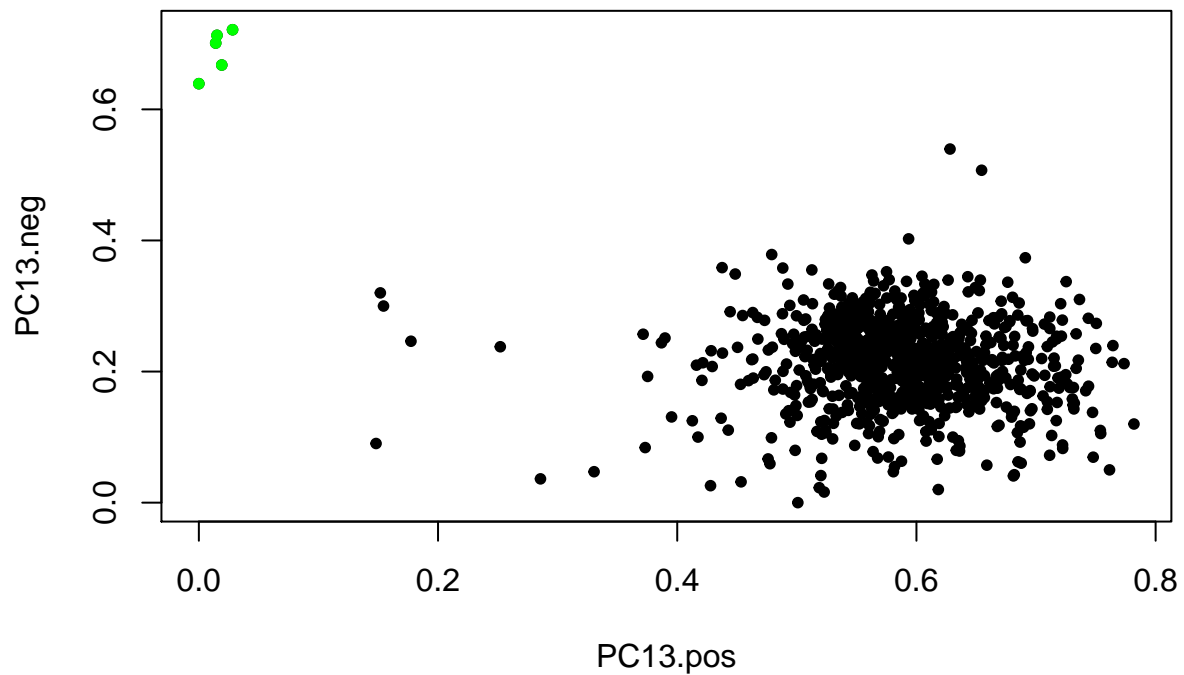


```
ggsave('all_r1/all_r1_hbbHi_separation.pdf',height = 3, width = 4)

if(cutoff.point > 0){
  cells.hbb.hi = pc.scores[ PC13.score > cutoff.point, cell.name]
} else{
  cells.hbb.hi = pc.scores[ PC13.score < cutoff.point, cell.name]
}

save(cells.hbb.hi, file = 'all_r1/cells_hbb_hi.RData')
write.table(cells.hbb.hi, file = 'all_r1/cells_hbb_hi.txt')

with(pc.scores, plot(PC13.pos, PC13.neg, pch = 20))
with(pc.scores[cell.name %in% cells.hbb.hi],points(PC13.pos, PC13.neg, col = 'green', pch = 20 ))
```



## all r2 (endocardial vs Aplnr)

2. rPCA, 15 PCs, all genes minus cell cycle (noCC). (endocard\_refine\_r1)

a. PC15 and PC2 separate a group of Gpr126+ endocardial cells that appears continuous with the Kcne- g

b. Remove the Kcne+ group that is not continuous w/Gpr126 popuulation to further refine the Gpr126+ po

```
rm(list=ls()[ls() %like% 'cells.*'])
cordir='all_r2/'
system(paste0('mkdir ',cordir));
system(paste0('mkdir ',file.path(cordir,'biplots')));

load('all_r1/cells_hbb_hi.RData')

dt.pca = dt.fv1_NS.good[!cell.name %in% cells.hbb.hi][!gene %in% genes.cc]
setkey(dt.pca)

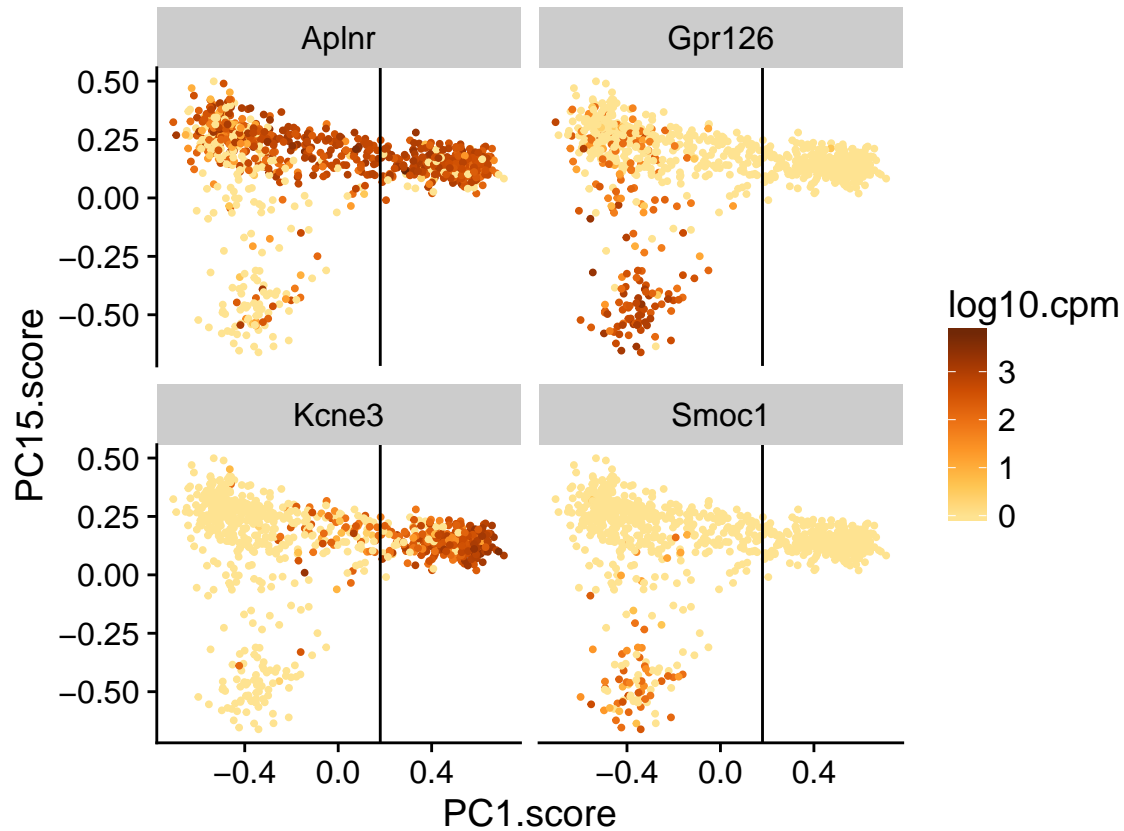
n.pcs = 15
if(recalculate.pca){

  pca.d1 <- dimplots.pca.lowMem(dt.pca,cell.info,cordir,suffix='',max.genes = 30,ncp=n.pcs,
                                min.cells.detect=1,n.cores = 8)
  plot.ggpairs.wExptCols(cell.info,dir=cordir,suffix='',num.pcs = n.pcs,height=25,width=25,sigType = 'S')

  make.facet.biplots(dt.pca,c(paste0('PC',1),paste0('PC',15)),cordir,size=1.5, plot.pdf = T,
                      custom.genes = c('Aplnr','Smoc1','Pdghfra','A2m','Fbln2','Fbln5','Sfrp2','Kcne3','Gja5',
                                         'Cxcr7','Gja5'),height = 6, width = 9)
}
## remove coronary cells
pc.scores = fread(paste0(cordir, 'PC_allscores.csv'))
dt.pca.2 = merge(dt.pca[gene %in% c('Kcne3','Aplnr','Smoc1','Gpr126')],
                 pc.scores,
                 by='cell.name')

cutoff.point <- 0.18
if(with(dt.pca.2[gene=='Kcne3'], cor(log10.cpm, PC1.score)) < 0) cutoff.point <- -cutoff.point

ggplot(dt.pca.2, aes(PC1.score, PC15.score, color = log10.cpm)) +
  geom_point(size=.7) +
  geom_vline(xintercept = cutoff.point) +
  scale_color_gradientn(colours=brewer.pal(9,'YlOrBr')[3:9]) +
  coord_fixed() +
  facet_wrap(~gene)
```

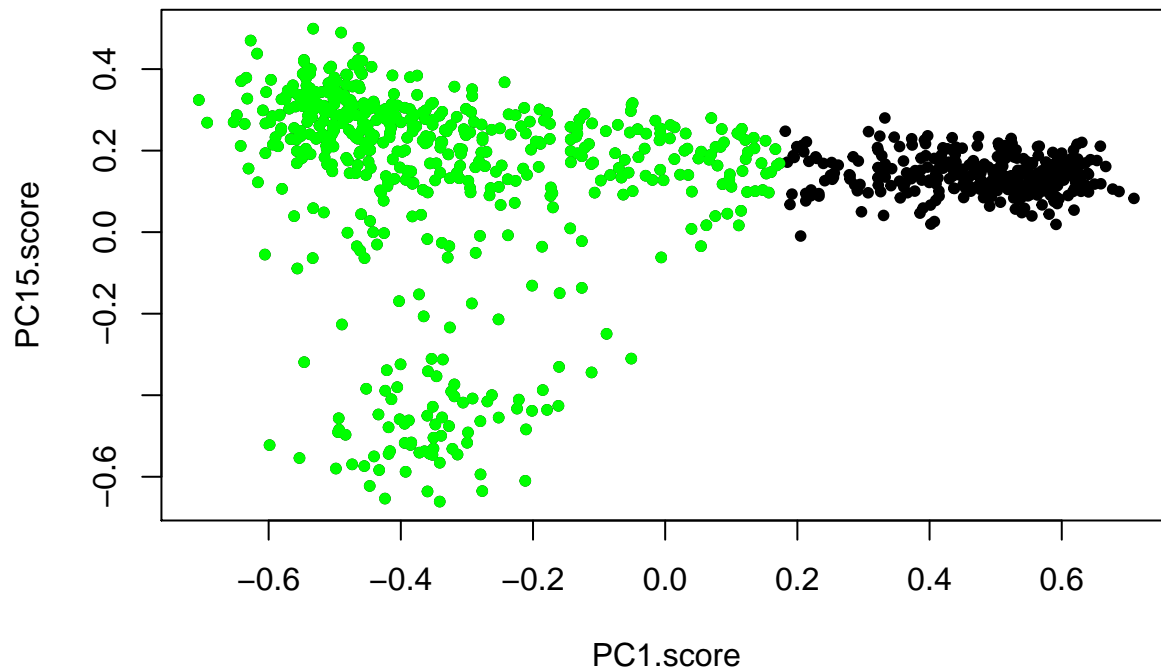


```
ggsave('all_r2/cells_kcne3_all_r2_separation.pdf',height = 3, width = 4)
```

```
if(cutoff.point > 0){
cells.kcne3.all_r2 = pc.scores[ PC1.score > cutoff.point, cell.name]
} else{
cells.kcne3.all_r2 = pc.scores[ PC1.score < cutoff.point, cell.name]
}
```

```
save(cells.kcne3.all_r2, file = 'all_r2/cells_kcne3_all_r2.RData')
write.table(cells.kcne3.all_r2, file = 'all_r2/cells_kcne3_all_r2.txt')
```

```
with(pc.scores, plot(PC1.score, PC15.score, pch = 20)) +
  with(pc.scores[!cell.name %in% cells.kcne3.all_r2],points(PC1.score, PC15.score, col = 'green', pch =
```



```
## integer(0)
```

## endocard refine 1

3. rPCA, 15 PCs, noCC.

a. PC2 and PC5 separate the Gpr126+ population from the Cldn11+ population.

b. Gpr126+ population appears continuous with the Apln-, PC5-high part of the Cldn11+ population.

c. Remove the Apln+ population (PC2 > -0.1 and PC5 < -0.1) to further refine the Gpr126+ population.

```
cordir='endocard_refine_r1/'
system(paste0('mkdir ',cordir));
system(paste0('mkdir ',file.path(cordir,'biplots')));

load('all_r1/cells_hbb_hi.RData')
load('all_r2/cells_kcne3_all_r2.RData')

dt.pca = dt.fv1_NS.good[!cell.name %in% c(cells.hbb.hi, cells.kcne3.all_r2)]
dt.pca = dt.pca[!gene %in% genes.cc]
setkey(dt.pca)

n.pcs = 15
if(recalculate.pca){
pca.d1 <- dimplots.pca.lowMem(dt.pca,cell.info,cordir,suffix='',max.genes = 30,ncp=n.pcs,
                             min.cells.detect=1,n.cores = 8)
plot.ggpairs.wExptCols(cell.info,dir=cordir,suffix='',num.pcs = n.pcs,height=25,width=25,sigType = 'Sig')

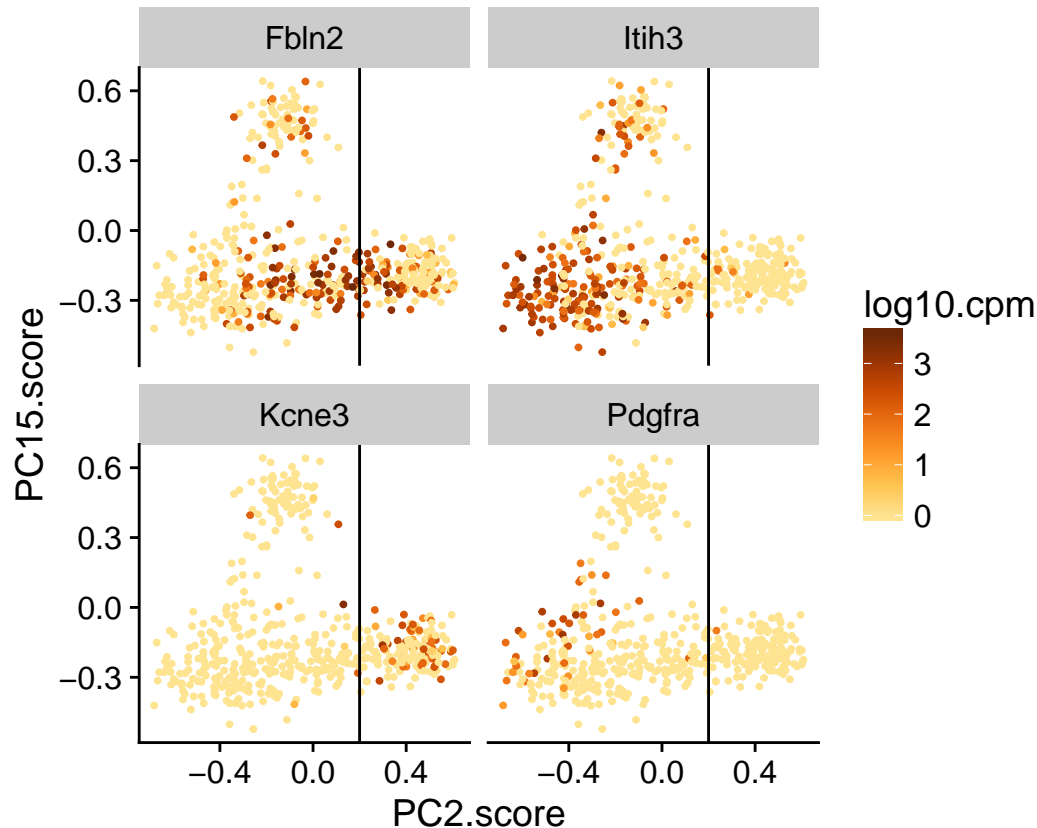
make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',15)),cordir,size=1.5, plot.pdf = T,num.genes = 1)
}

# Further refinement of endocardial population
pc.scores = fread(paste0(cordir, 'PC_allscores.csv'))
```

```
dt.pca.2 = merge(dt.pca[gene %in% c('Pdgfra', 'Fbln2', 'Itih3', 'Kcne3')],
                pc.scores, by='cell.name')

cutoff.point <- 0.2
if(with(dt.pca.2[gene=='Kcne3'], cor(log10.cpm, PC2.score)) < 0) cutoff.point <- -cutoff.point

ggplot(dt.pca.2, aes(PC2.score, PC15.score, color = log10.cpm)) + geom_point(size = .7) +
  scale_color_gradientn(colours=brewer.pal(9, 'YlOrBr')[3:9]) +
  facet_wrap(~gene) + geom_vline(xintercept = cutoff.point) +
  coord_fixed()
```

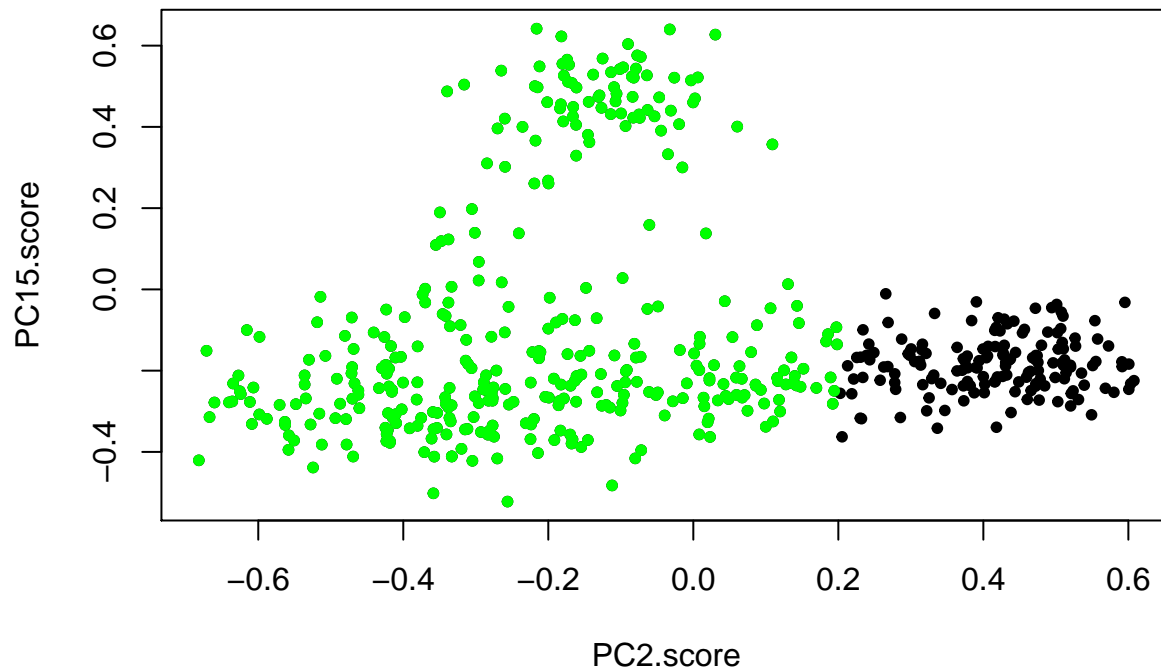


```
ggsave('endocard_refine_r1/endocard_refine_r1_separation.pdf', height = 6, width = 9)

if(cutoff.point > 0){
  cells.endocardRefine.r1.v2 = pc.scores[ PC2.score < cutoff.point, cell.name]
} else{
  cells.endocardRefine.r1.v2 = pc.scores[ PC2.score > cutoff.point, cell.name]
}
cells.endocardRefine.r1.v2 = pc.scores[ PC2.score < cutoff.point, cell.name]

save(cells.endocardRefine.r1.v2, file = 'endocard_refine_r1/cells_endocard_refine_r1.RData')
write.table(cells.endocardRefine.r1.v2, file = 'endocard_refine_r1/cells_endocard_refine_r1.txt')

with(pc.scores, plot(PC2.score, PC15.score, pch = 20)) +
  with(pc.scores[cell.name %in% cells.endocardRefine.r1.v2], points(PC2.score, PC15.score, col = 'green'))
```



```
## integer(0)
```

## endocard refine 2 (remove Aplnr+ population)

There is possibly a continuum between the endocardial (Gpr126) population and the rest of the cells. Remove the Aplnr+ population using PC2 and PC5, because it is not directly continuous with the endocardial population, to continue analyzing the transition between endocardial and other cell types.

```
cordir='endocard_refine_r2/'
system(paste0('mkdir ',cordir));
system(paste0('mkdir ',file.path(cordir,'biplots')));

load('endocard_refine_r1/cells_endocard_refine_r1.RData')

dt.pca = dt.fv1_NS.good[cell.name %in% c(cells.endocardRefine.r1.v2)]
dt.pca = dt.pca[!gene %in% genes.cc]
setkey(dt.pca)

n.pcs = 15
if(recalculate.pca){

pca.d1 <- dimplots.pca.lowMem(dt.pca,cell.info,cordir,suffix='',max.genes = 30,ncp=n.pcs,
                             min.cells.detect=1,n.cores = 8)
plot.ggpairs.wExptCols(cell.info,dir=cordir,suffix='',num.pcs = n.pcs,
                       height=25,width=25,sigType = 'Sig',annot.expt = F, plot.read.depth = F)
plot.ggpairs.wExptCols(cell.info,dir=cordir,suffix='',num.pcs = n.pcs,
                       height=25,width=25,sigType = 'PC',annot.expt = F, plot.read.depth = F)

make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',5)),cordir,size=2, plot.pdf = T, num.genes = 10)
```

```

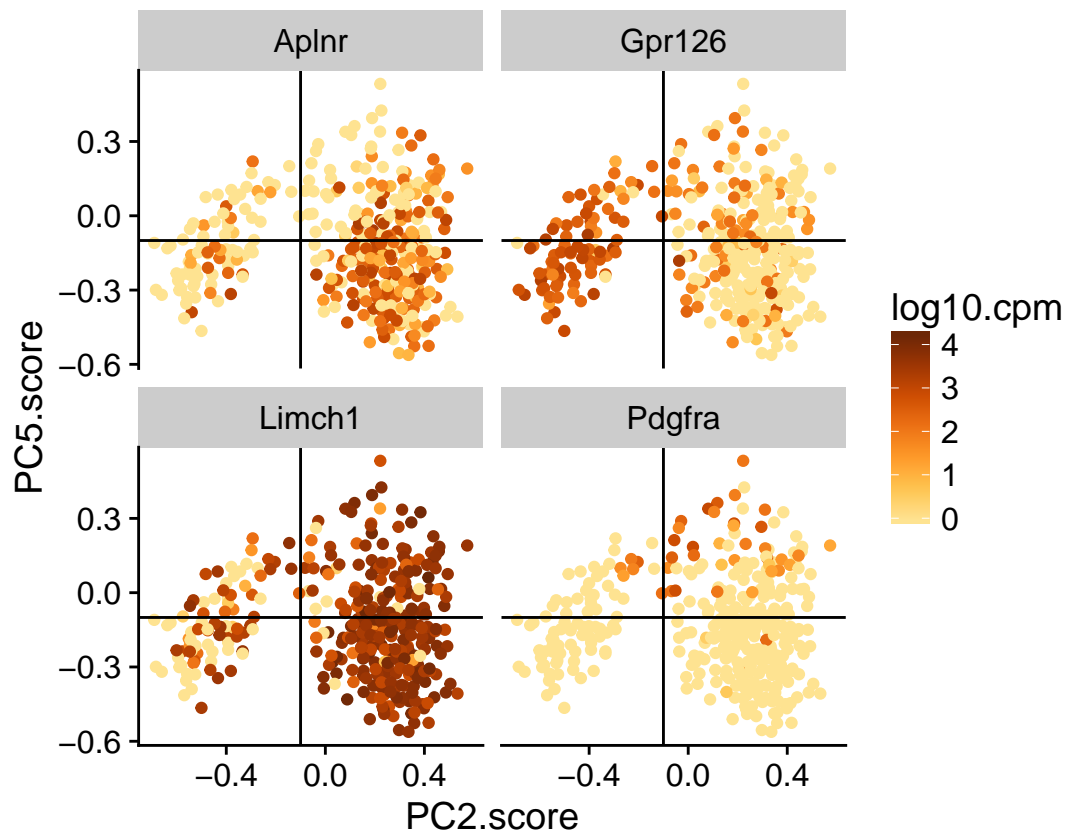
make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',4)),cordir,size=1.5, plot.pdf = T,
                    custom.genes = c('Gpr126','Fbln2','Pdgfra','Aplnr','A2m','Nr2f2'), height = 6, width
}

# Further refinement of endocardial population
rm(pc.scores, dt.pca.2)
pc.scores = fread(paste0(cordir, 'PC_allscores.csv'))
dt.pca.2 = merge(dt.pca[gene %in% c('Pdgfra','Gpr126','Limch1','Aplnr')], pc.scores,by='cell.name')

cutoff.1 <- -.1
if(with(dt.pca.2[gene=='Aplnr'], cor(log10.cpm, PC2.score)) < 0) cutoff.1 <- -cutoff.1
cutoff.2 <- -.1
if(with(dt.pca.2[gene=='Pdgfra'], cor(log10.cpm, PC5.score)) < 0) cutoff.2 <- -cutoff.2

ggplot(dt.pca.2, aes(PC2.score, PC5.score, color = log10.cpm)) + geom_point() +
  geom_hline(yintercept = cutoff.2) +
  geom_vline(xintercept = cutoff.1) +
  scale_color_gradientn(colours=brewer.pal(9,'YlOrBr')[3:9]) +
  facet_wrap(~gene) +
  coord_fixed()

```



```

ggsave('endocard_refine_r2/endocard_refine_r2_separation.pdf',height = 9, width = 11)

if(cutoff.1 > 0){
  cells.aplnr.hi <- pc.scores[PC2.score > cutoff.1, cell.name]
} else{
  cells.aplnr.hi <- pc.scores[PC2.score < cutoff.1, cell.name]
}

```



```

}

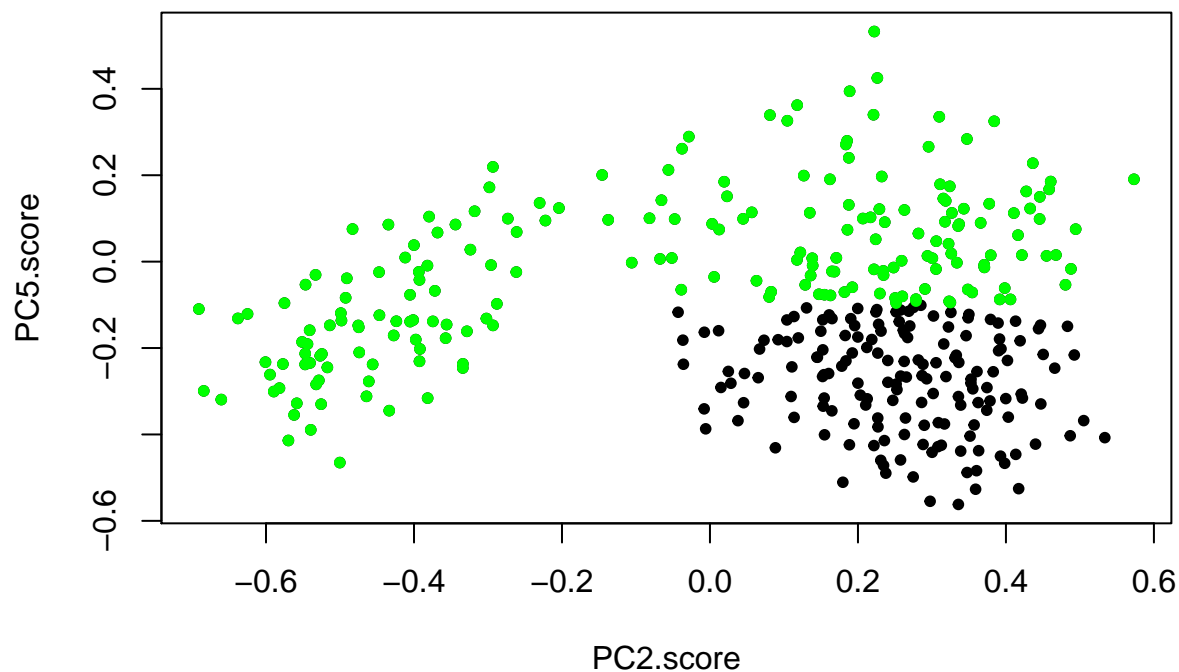
if(cutoff.2 > 0){
  cells.pdgfra.hi <- pc.scores[PC5.score < cutoff.2, cell.name]
} else{
  cells.pdgfra.hi <- pc.scores[PC5.score > cutoff.2, cell.name]
}

cells.endocardRefine.r2.v2 = union(cells.pdgfra.hi, cells.aplnr.hi)

save(cells.endocardRefine.r2.v2, file = 'endocard_refine_r2/cells_endocard_refine_r2.RData')
write.table(cells.endocardRefine.r2.v2, file = 'endocard_refine_r2/cells_endocard_refine_r2.txt')

plt=with(pc.scores, plot(PC2.score, PC5.score, pch = 20)) + with(pc.scores[cell.name %in% cells.endocar

```



```

plt

## integer(0)

pdf('endocard_refine_r2/endocard_refine_r2_separation_verification.pdf', height = 4, width = 4)
plt

## integer(0)

dev.off()

## pdf
## 2

```

## endocard refine 3 (define small Fbln5+ population)

PC8 separates a small subpopulation of Fbln5+ cells; these may be an interesting progenitor population. However, there are too few Fbln5+ cells to study.

```
cordir='endocard_refine_r3/'
system(paste0('mkdir ',cordir));
system(paste0('mkdir ',file.path(cordir,'biplots')));

load('endocard_refine_r2/cells_endocard_refine_r2.RData')

dt.pca = dt.fv1_NS.good[cell.name %in% c(cells.endocardRefine.r2.v2)]
dt.pca = dt.pca[!gene %in% genes.cc]
dt.pca = dt.pca[!gene %in% c('Rn45s','Lars2','Malat1')]
setkey(dt.pca)

n.pcs = 10
if(recalculate.pca){

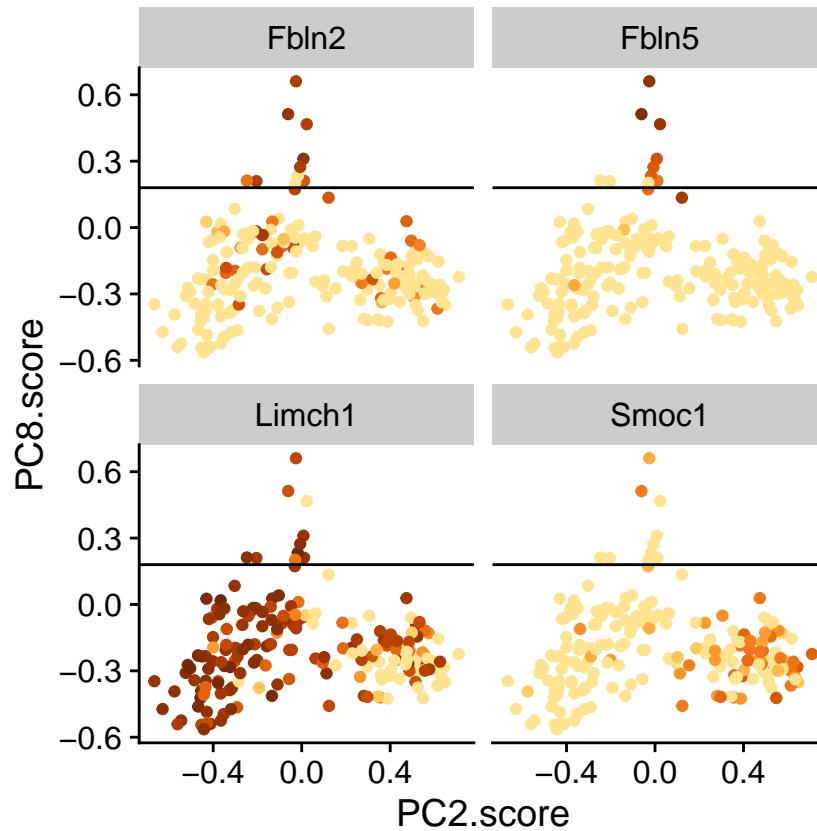
pca.d1 <- dimplots.pca.lowMem(dt.pca,cell.info,cordir,suffix='',max.genes = 30,npc=n.pcs,
                             min.cells.detect=1,n.cores = 8)
plot.ggpairs.wExptCols(cell.info,dir=cordir,suffix='',num.pcs = n.pcs,height=25,width=25,sigType = 'Sig

make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',8)),cordir,size=2, plot.pdf = T)
}

# Further refinement of endocardial population
pc.scores = fread(paste0(cordir, 'PC_allscores.csv'))
dt.pca.2 = merge(dt.pca[gene %in% c('Fbln2','Fbln5','Smoc1','Limch1')],
                 pc.scores,
                 by='cell.name')

cutoff.1 <- .18
if(with(dt.pca.2[gene=='Fbln5'], cor(log10.cpm, PC8.score)) < 0) cutoff.1 <- -cutoff.1

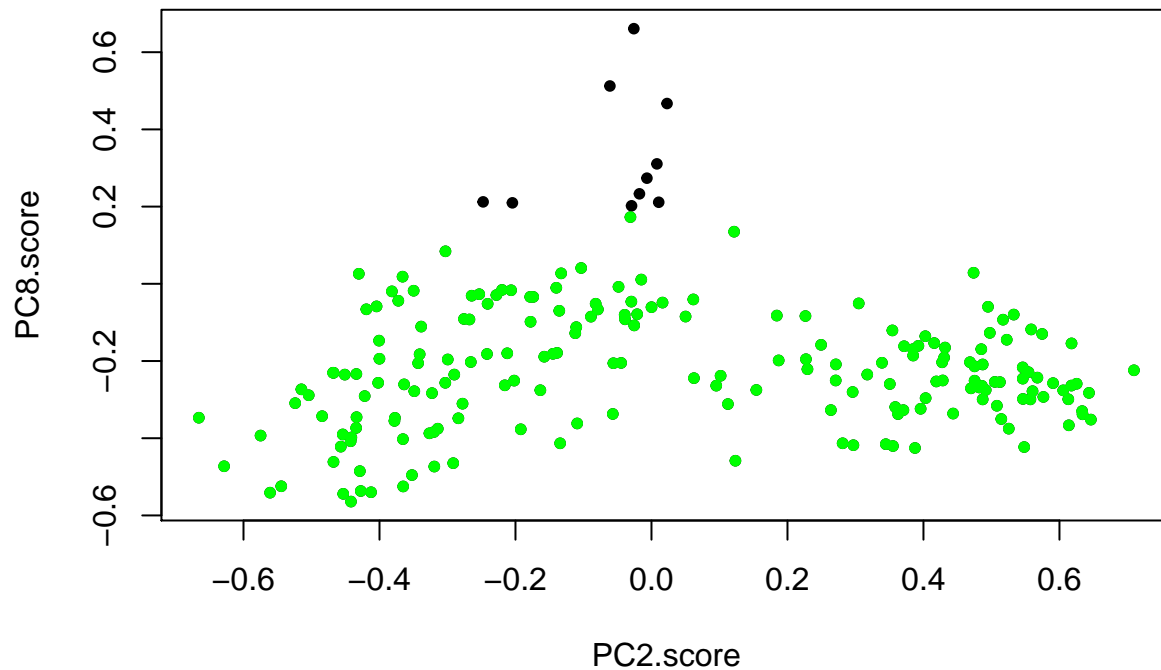
ggplot(dt.pca.2, aes(PC2.score, PC8.score, color = log10.cpm)) + geom_point() +
  scale_color_gradientn(colours=brewer.pal(9,'YlOrBr')[3:9]) +
  facet_wrap(~gene) + coord_fixed() +
  theme(legend.position = 'none')+
  geom_hline(yintercept = cutoff.1)
```



```
ggsave('endocard_refine_r3/endo_refine_r3_separation.pdf',height = 9, width = 11)
```

```
if(cutoff.1 > 0){
  cells.endocardRefine.r3.v2.Fbln5 = pc.scores[PC8.score > cutoff.1, cell.name]
  cells.endocardRefine.r3.v2.rest = pc.scores[PC8.score <= cutoff.1, cell.name]
} else{
  cells.endocardRefine.r3.v2.Fbln5 = pc.scores[PC8.score < cutoff.1, cell.name]
  cells.endocardRefine.r3.v2.rest = pc.scores[PC8.score >= cutoff.1, cell.name]
}
```

```
plt=with(pc.scores, plot(PC2.score, PC8.score, pch = 20)) +
  with(pc.scores[cell.name %in% cells.endocardRefine.r3.v2.rest],points(PC2.score, PC8.score, col = 'gr
```



```
plt
```

```
## integer(0)
```

```
pdf('endocard_refine_r3/endocard_refine_r3_separation_verification.pdf', height = 4, width = 4)
```

```
plt
```

```
## integer(0)
```

```
dev.off()
```

```
## pdf
```

```
## 2
```

```
save(cells.endocardRefine.r3.v2.Fbln5, file = 'endocard_refine_r3/cells_.endocardRefine.r3.v2.Fbln5.Rdata')
```

```
save(cells.endocardRefine.r3.v2.rest, file = 'endocard_refine_r3/cells.endocardRefine.r3.v2.rest.Rdata')
```

## endocard refine 4 (remove small Fbln2 population)

PC4 separates a small population of Fbln2+ cells. These are likely SVv cells.

```
rm(list=ls()[ls() %like% 'cells.*'])
```

```
cordir='endocard_refine_r4/'
```

```
system(paste0('mkdir ',cordir));
```

```
system(paste0('mkdir ',file.path(cordir,'biplots')));
```

```
load('endocard_refine_r3/cells.endocardRefine.r3.v2.rest.Rdata')
```

```
dt.pca = dt.fv1_NS.good[cell.name %in% c(cells.endocardRefine.r3.v2.rest)]
```

```
dt.pca = dt.pca[!gene %in% genes.cc]
```

```
dt.pca = dt.pca[!gene %in% c('Rn45s','Lars2','Malat1')]
```

```
setkey(dt.pca)
```

```

n.pcs = 10
if(recalculate.pca){

pca.d1 <- dimplots.pca.lowMem(dt.pca,cell.info,cordir,suffix='',max.genes = 30,ncp=n.pcs,
                             min.cells.detect=1,n.cores = 8)
plot.ggpairs.wExptCols(cell.info,dir=cordir,suffix='',num.pcs = n.pcs,height=25,width=25,sigType = 'Sig')

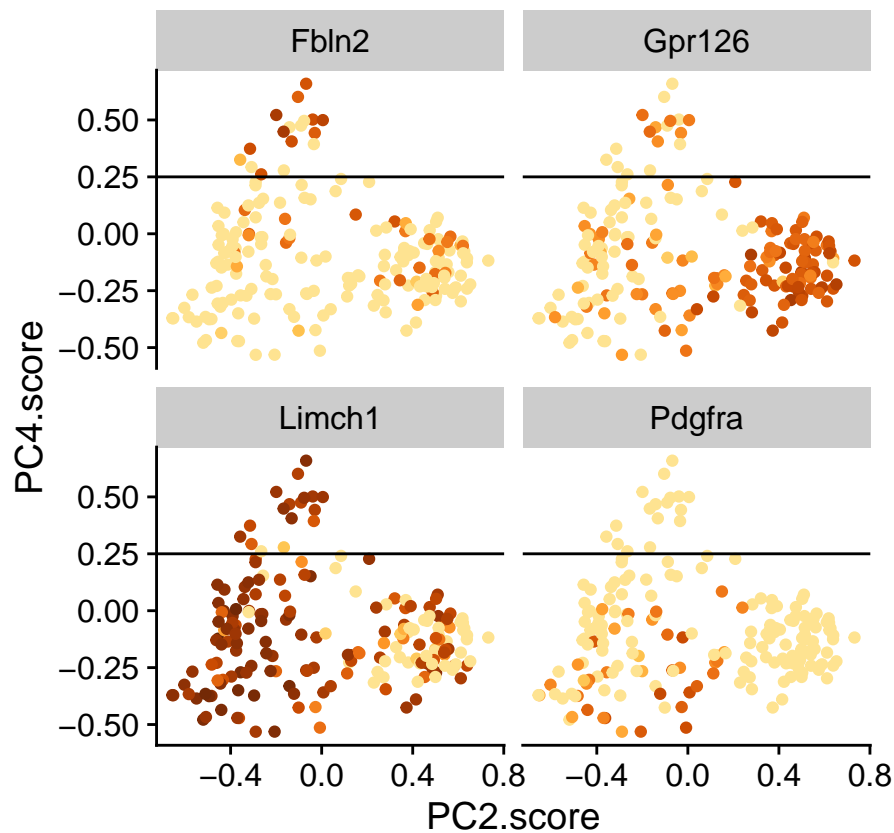
make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',4)),cordir,size=2, plot.pdf = T)
}

pc.scores = fread(paste0(cordir, 'PC_allscores.csv'))
dt.pca.2 = merge(dt.pca[gene %in% c('Fbln2','Pdgfra','Gpr126','Limch1')],
                 pc.scores,
                 by='cell.name')

cutoff.1 <- .25
if(with(dt.pca.2[gene=='Fbln2'], cor(log10.cpm, PC4.score)) < 0) cutoff.1 <- -cutoff.1

ggplot(dt.pca.2, aes(PC2.score, PC4.score, color = log10.cpm)) + geom_point() +
  scale_color_gradientn(colours=brewer.pal(9,'YlOrBr')[3:9]) +
  facet_wrap(~gene) + coord_fixed() +
  theme(legend.position = 'none')+
  geom_hline(yintercept = cutoff.1) +
  coord_fixed()

```



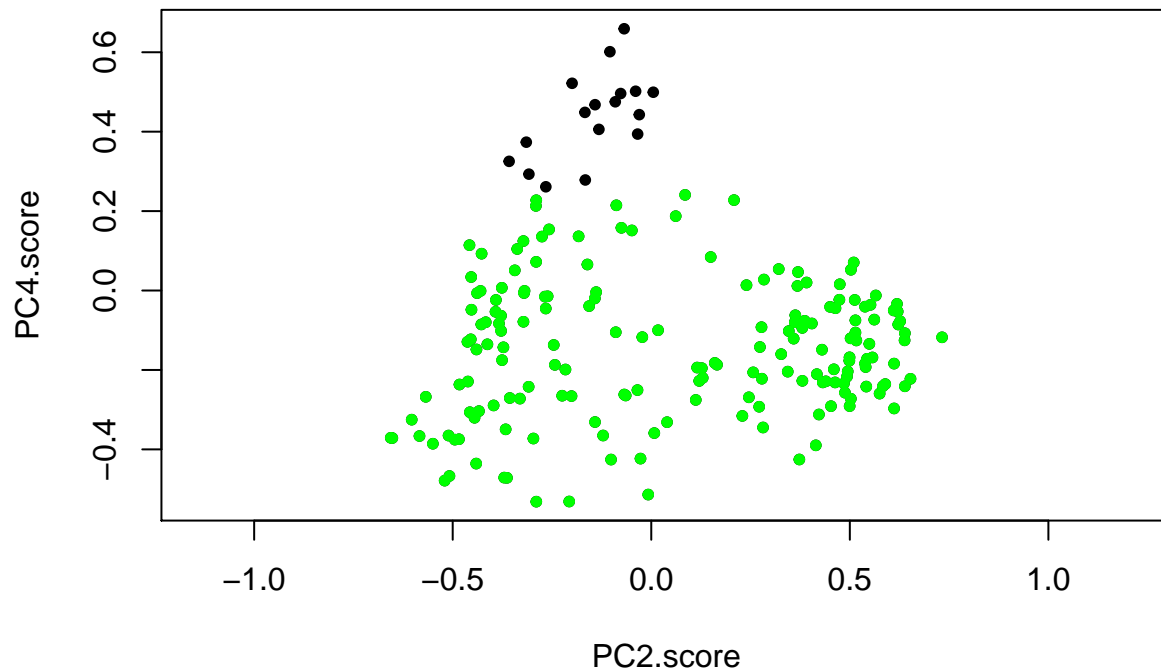
```

ggsave('endocard_refine_r4/endo_refine_r4_separation.pdf',height = 9, width = 11)

if(cutoff.1 > 0){
  cells.endocardRefine.r4.v2.Fbln2 = pc.scores[PC4.score > cutoff.1, cell.name]
  cells.endocardRefine.r4.v2.rest = pc.scores[PC4.score <= cutoff.1, cell.name]
} else{
  cells.endocardRefine.r4.v2.Fbln2 = pc.scores[PC4.score < cutoff.1, cell.name]
  cells.endocardRefine.r4.v2.rest = pc.scores[PC4.score >= cutoff.1, cell.name]
}

plt=with(pc.scores, plot(PC2.score, PC4.score, pch = 20, asp=1)) +
  with(pc.scores[cell.name %in% cells.endocardRefine.r4.v2.rest],points(PC2.score, PC4.score, col = 'gr

```



```

pdf('endocard_refine_r4/endocard_refine_r4_separation_verification.pdf', height = 4, width = 4)
plt

## integer(0)
dev.off()

## pdf
## 2
#
save(cells.endocardRefine.r4.v2.Fbln2, file = 'endocard_refine_r4/cells_.endocardRefine.r4.v2.Fbln5.Rda')
save(cells.endocardRefine.r4.v2.rest, file = 'endocard_refine_r4/cells.endocardRefine.r4.v2.rest.Rdata')

```

## endocard refine 5 (define endocardial population)

PC1 is cell cycle Endocardial population appears to be continuous with primarily the Pdgfra mesenchymal population. Use PC2.score to define endocardial cells (Gpr126-hi). Use PC3.score to define the Pdgfra

mesenchymal population (for now).

```
rm(list=ls()[ls() %like% 'cells.*'])
cendir='endocard_refine_r5/'
system(paste0('mkdir ',cendir));
system(paste0('mkdir ',file.path(cendir,'biplots')));

load('endocard_refine_r4/cells.endocardRefine.r4.v2.rest.Rdata')

dt.pca = dt.fv1_NS.good[cell.name %in% c(cells.endocardRefine.r4.v2.rest)]
dt.pca = dt.pca[!gene %in% genes.cc]
dt.pca = dt.pca[!gene %in% c('Rn45s','Lars2','Malat1')]
setkey(dt.pca)

n.pcs = 10
if(recalculate.pca){

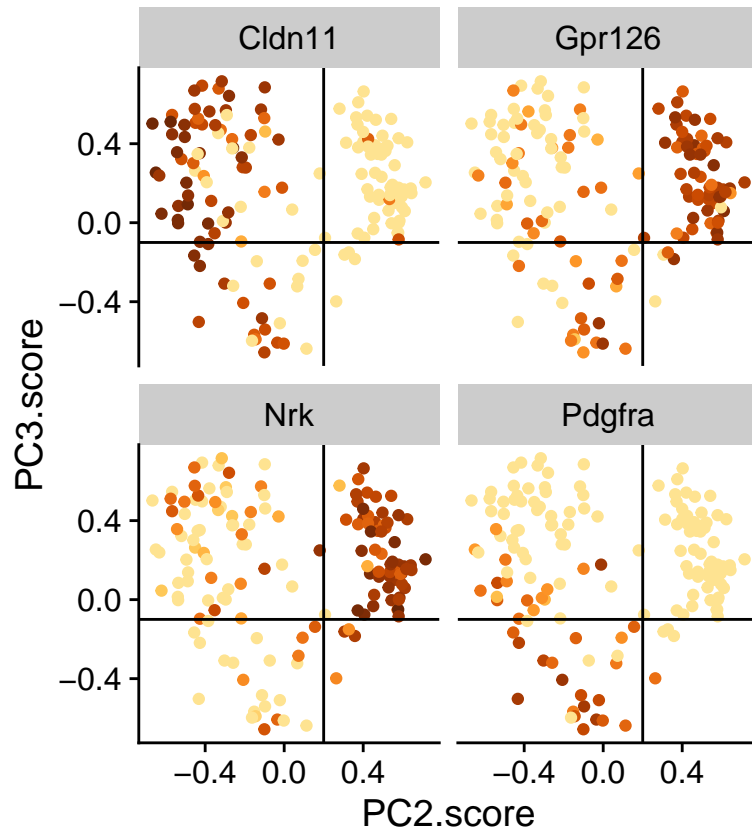
pca.d1 <- dimplots.pca.lowMem(dt.pca,cell.info,cendir,suffix='',max.genes = 30,ncp=n.pcs,
                             min.cells.detect=1,n.cores = 8, min.numgenes = 3500)
plot.ggpairs.wExptCols(cell.info,dir=cendir,suffix='',num.pcs = n.pcs,height=25,width=25,sigType = 'Sig

make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',10)),cendir,size=2, plot.pdf = T)
make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',3)),cendir,size=2, plot.pdf = T)
}
# Define the endocardial population based on PC2 (corresponds to Nr4a1/Pdgfra- transition).
pc.scores = fread(paste0(cendir, 'PC_allscores.csv'))
dt.pca.2 = merge(dt.pca[gene %in% c('Nr4a1','Pdgfra','Gpr126','Cldn11')],
                 pc.scores,
                 by='cell.name')

cutoff.1 <- .2 # Endocardial separation
if(with(dt.pca.2[gene=='Gpr126'], cor(log10.cpm, PC2.score)) < 0) cutoff.1 <- -cutoff.1

cutoff.2 <- .1 # Pdgfra separation (not defining Pdgfra cells here since there is a lot of other hetero
if(with(dt.pca.2[gene=='Pdgfra'], cor(log10.cpm, PC3.score)) < 0) cutoff.2 <- -cutoff.2

ggplot(dt.pca.2, aes(PC2.score, PC3.score, color = log10.cpm)) + geom_point() +
  scale_color_gradientn(colours=brewer.pal(9,'YlOrBr')[3:9]) +
  facet_wrap(~gene) +
  coord_fixed() +
  theme(legend.position = 'none')+
  geom_vline(xintercept = cutoff.1) +
  geom_hline(yintercept = cutoff.2)
```



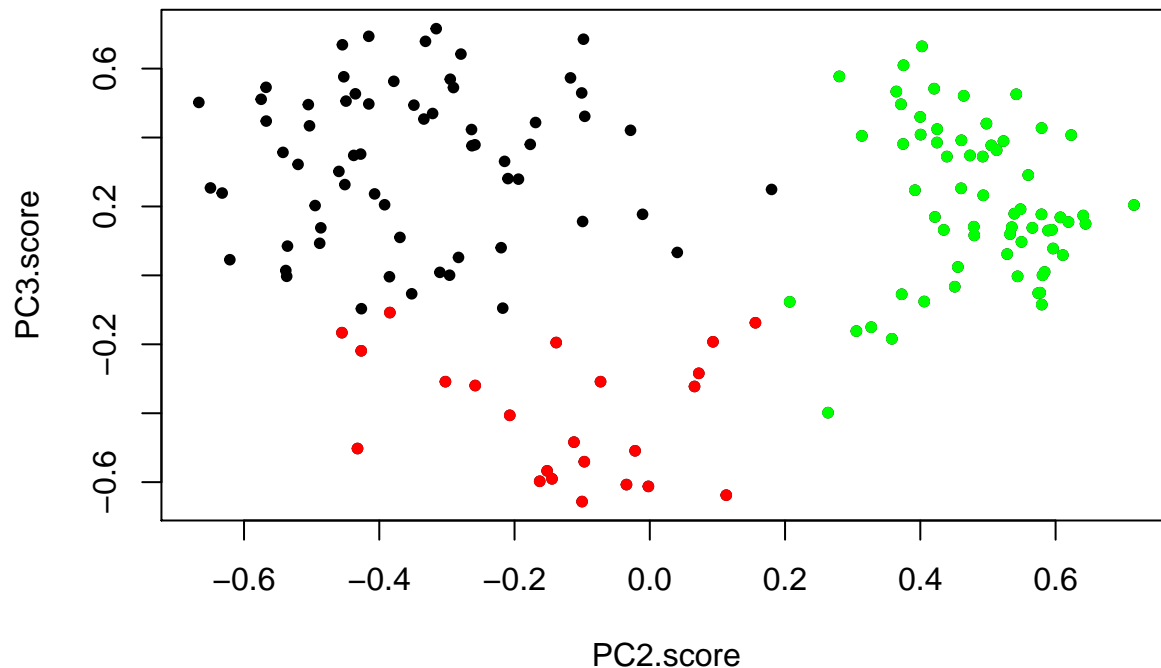
```
ggsave('endocard_refine_r5/endo_refine_r5_separation.pdf',height = 9, width = 11)
```

```
# Use PC2 to separate
if(cutoff.1 < 0){
  cells.endocardRefine.r5.v2.endocard = pc.scores[PC2.score < cutoff.1, cell.name]
} else{
  cells.endocardRefine.r5.v2.endocard = pc.scores[PC2.score > cutoff.1, cell.name]
}

if(cutoff.2 > 0){
  cells.endocardRefine.r5.v2.pdgfra = pc.scores[(!cell.name %in% cells.endocardRefine.r5.v2.endocard) &
} else{
  cells.endocardRefine.r5.v2.pdgfra = pc.scores[(!cell.name %in% cells.endocardRefine.r5.v2.endocard)
}

# verify endocardial definition.
with(pc.scores, plot(PC2.score, PC3.score, pch = 20))+
  with(pc.scores[cell.name %in% cells.endocardRefine.r5.v2.endocard],points(PC2.score, PC3.score, col = 'orange'))+
  with(pc.scores[cell.name %in% cells.endocardRefine.r5.v2.pdgfra],points(PC2.score, PC3.score, col = 'yellow'))
```





```
## integer(0)
```

```
save(cells.endocardRefine.r5.v2.endocard, file = 'endocard_refine_r5/cells_endocardRefine.r5.v2.endocard.Rdata')
write.table(cells.endocardRefine.r5.v2.endocard, file = 'endocard_refine_r5/cells_endocardRefine.r5.v2.endocard.csv')
save(cells.endocardRefine.r5.v2.pdgfra, file = 'endocard_refine_r5/cells_endocardRefine.r5.v2.pdgfra.Rdata')
```

## all\_r3: (no endocard; remove valve)

After defining the endocardial population (and two very small groups of cells: Fbln5+ and Hbb+) in the previous steps, analyze the rest of the cells with rPCA.

PC2 and PC3 scores show a continuum from valve-like (Corin+) to CV Plexus (Apln+)-like, connected by at least 2 distinct intermediate populations (Fbln2+, A2m+); however, there are too many intermediate populations for PCA to clearly separate them all.

Use PC2.score and PC3.score to remove the Corin+/Fbln2- valve population and run rPCA on the rest of the cells so that heterogeneity in the rest of the continuum can be better revealed.

```
rm(list=ls()[ls() %like% 'cells.*'])
corder='all_r3/'
system(paste0('mkdir ',corder));
system(paste0('mkdir ',file.path(corder,'biplots')));

load('all_r1/cells_hbb_hi.RData')
load('endocard_refine_r5/cells_endocardRefine.r5.v2.endocard.Rdata')
load('endocard_refine_r5/cells_endocardRefine.r5.v2.pdgfra.Rdata')
load('endocard_refine_r4/cells_.endocardRefine.r4.v2.Fbln5.Rdata')

dt.pca = dt.fv1_NS.good[!cell.name %in% c(cells.hbb.hi,
                                           cells.endocardRefine.r5.v2.endocard,
                                           cells.endocardRefine.r5.v2.pdgfra,
                                           cells.endocardRefine.r4.v2.Fbln2)] [!gene %in% genes.cc]

setkey(dt.pca)
```

```

#
n.pcs = 15
if(recalculate.pca){

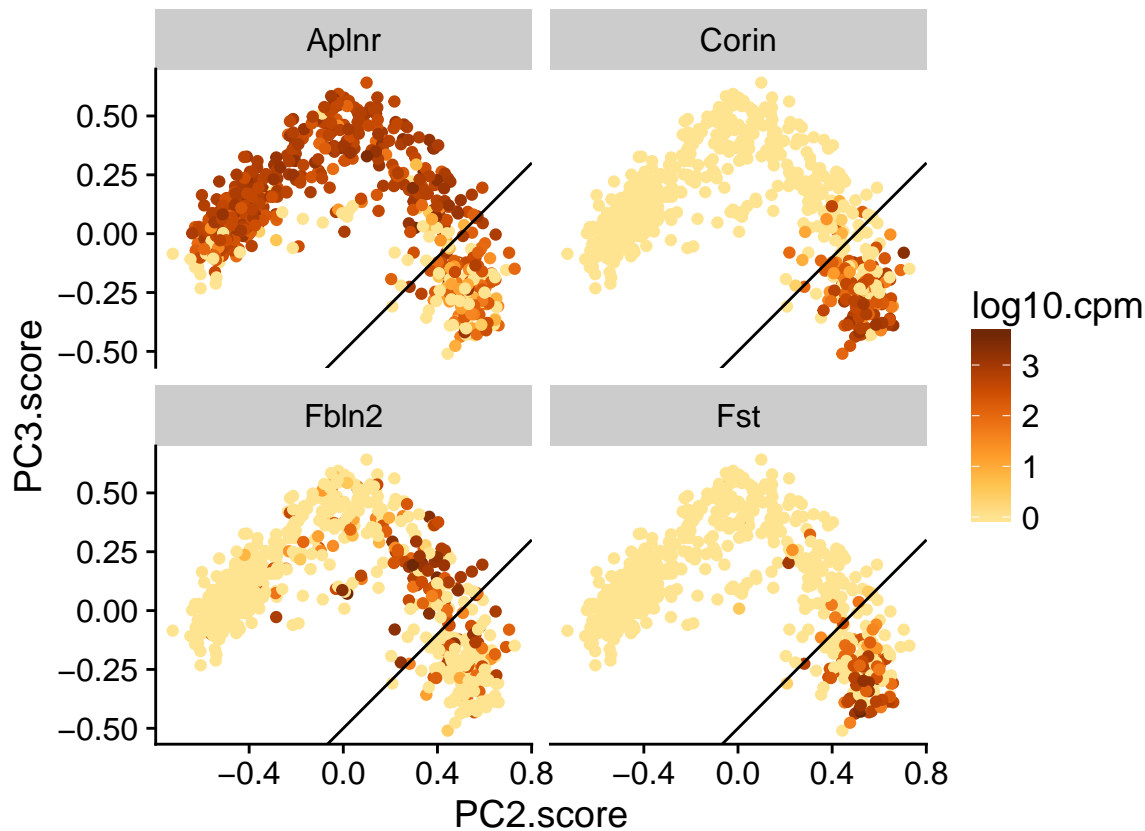
pca.d1 <- dimplots.pca.lowMem(dt.pca,cell.info,cordir,suffix='',max.genes = 30,ncp=n.pcs,
                             min.cells.detect=1,n.cores = 8, min.numgenes=3500)
plot.ggpairs.wExptCols(cell.info,dir=cordir,suffix='',num.pcs = n.pcs,height=25,width=25,sigType = 'Sig

make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',3)),cordir,size=1.5, plot.pdf = T)
}

rm(pc.scores, dt.pca.2)
pc.scores = fread(paste0(cordir, 'PC_allscores.csv'))
dt.pca.2 = merge(dt.pca[gene %in% c('Aplnr','Fst','Corin','Fbln2')],
                 pc.scores[,c('cell.name',colnames(pc.scores)[colnames(pc.scores) %like% "PC[0-9]"])], w
                 by = 'cell.name')

intercept=-.5
plt=ggplot(dt.pca.2, aes(PC2.score, PC3.score, color = log10.cpm)) + geom_point() +
  scale_color_gradientn(colours=brewer.pal(9,'YlOrBr')[3:9]) +
  facet_wrap(~gene) + coord_fixed() +
  geom_abline(intercept = intercept)
plt

```



```

save_plot('all_r3/all_r3_valve_separation.pdf',plt,ncol=2, nrow=2, base_height = 3.5)

cells.allr3.valve = pc.scores[PC3.score < PC2.score + intercept, cell.name]

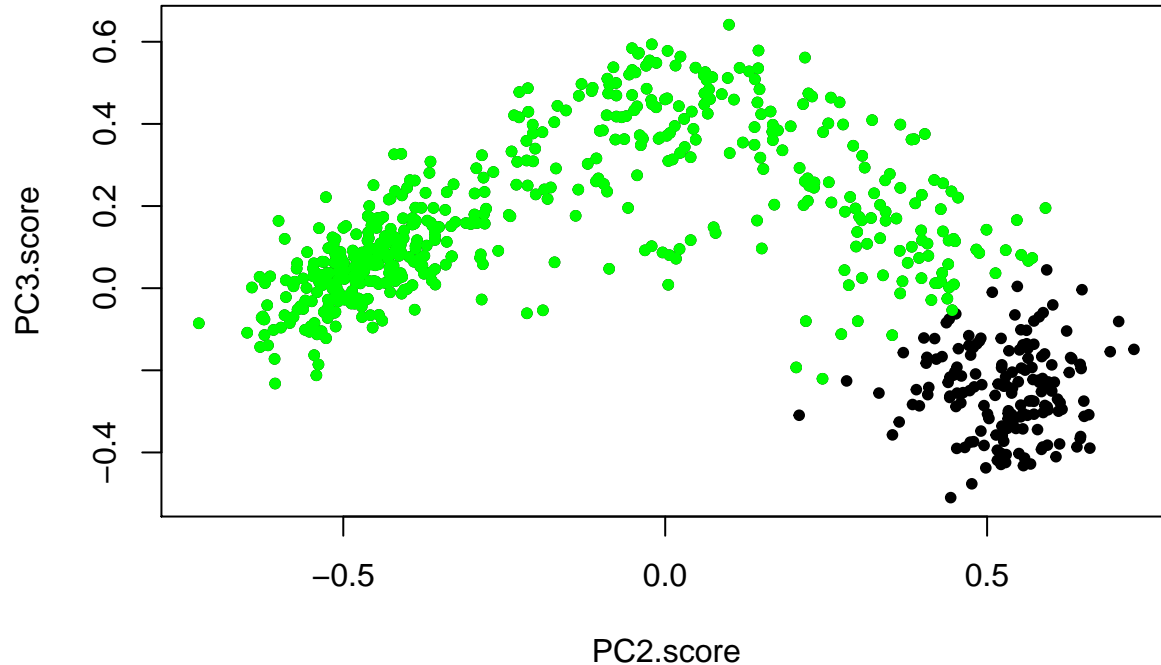
```

```

cells.allr3.rest = pc.scores[!cell.name %in% cells.allr3.valve, cell.name]

# Verify separation of valve-like population
with(pc.scores, plot(PC2.score, PC3.score, pch = 20)) +
  with(pc.scores[cell.name %in% cells.allr3.rest], points(PC2.score, PC3.score, col = 'green', pch = 20))

```



```

## integer(0)
save(cells.allr3.valve, file = 'all_r3/all_r3_valve.RData')
write.table(cells.allr3.valve, file = 'all_r3/all_r3_valve.txt')
save(cells.allr3.rest, file = 'all_r3/all_r3_rest.RData')

```

## all\_r4 (remove 3 immune cells)

PC15 separates 3 contaminating immune cells - remove them (too few cells to study)

```

rm(list=ls()[ls() %like% 'cells.*'])
cordin='all_r4/'
system(paste0('mkdir ',cordin));
system(paste0('mkdir ',file.path(cordin,'biplots')));

load('all_r3/all_r3_rest.RData')

dt.pca = dt.fv1_NS.good[cell.name %in% cells.allr3.rest][!gene %in% genes.cc]
setkey(dt.pca)

n.pcs = 15
if(recalculate.pca){

pca.d1 <- dimplots.pca.lowMem(dt.pca,cell.info,cordin,suffix='',max.genes = 30,ncp=n.pcs,
                             min.cells.detect=1,n.cores = 8)

```

```

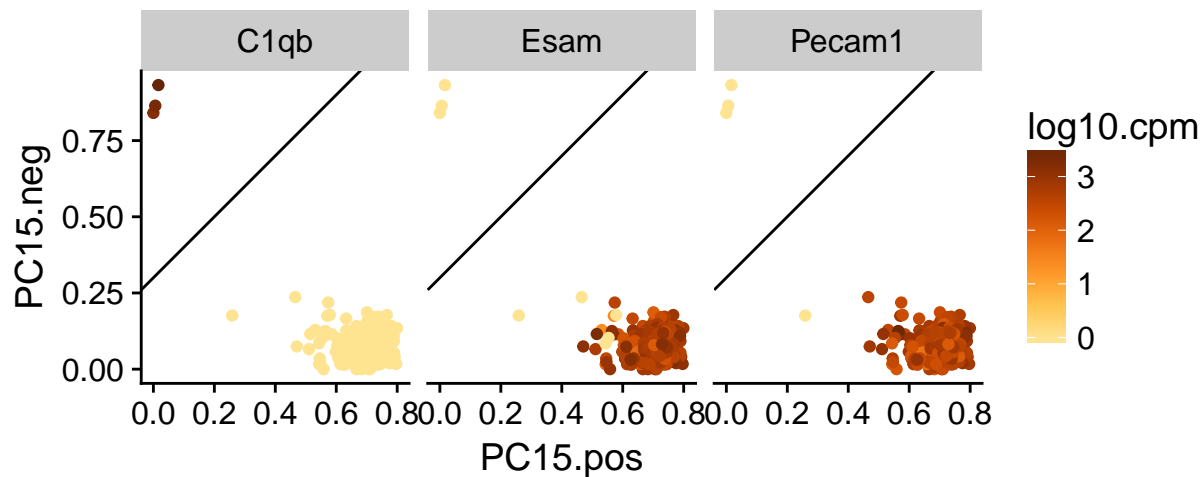
plot.ggpairs.wExptCols(cell.info,dir=cordir,suffix='',num.pcs = n.pcs,height=25,width=25,sigType = 'Sig

make.facet.biplots(dt.pca,c(paste0('PC',15),paste0('PC',15)), cordir,size=1.5, plot.pdf = T, ax.suf =c(
}

# remove 3 immune cells
pc.scores = fread(paste0(cordir, 'PC_allscores.csv'))
dt.pca.2 = merge(dt.pca[gene %in% c('Pecam1','C1qb','Esam')],
                pc.scores[,c('cell.name',colnames(pc.scores)[colnames(pc.scores) %like% "PC[0-9]"])], w
                by = 'cell.name')

cutoff <- -.3
plt=ggplot(dt.pca.2, aes(PC15.pos, PC15.neg, color = log10.cpm)) + geom_point() +
  scale_color_gradientn(colours=brewer.pal(9,'YlOrBr')[3:9]) +
  facet_wrap(~gene) + coord_fixed() +
  geom_abline(intercept = -cutoff)
plt

```



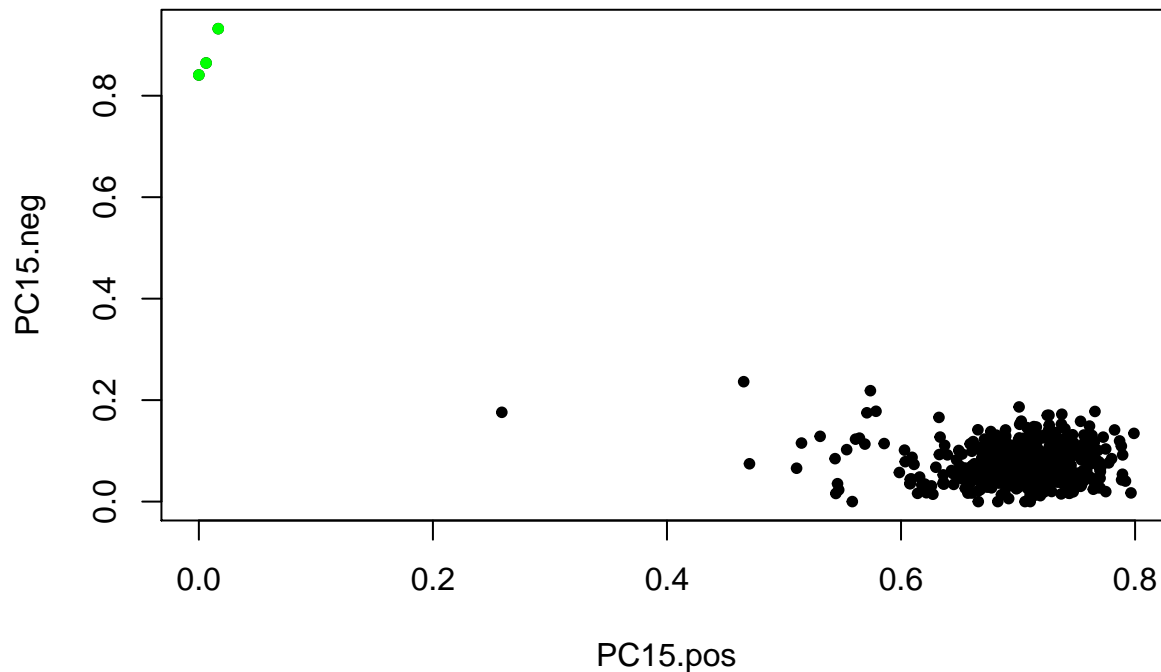
```

save_plot('all_r4/all_r4_separation.pdf',plt,ncol=3, nrow=1, base_height = 2.5)

# define the immune cells
cells.allr4.immune = pc.scores[PC15.score < cutoff, cell.name]
cells.allr4.rest = pc.scores[PC15.score >= cutoff, cell.name]

with(pc.scores, plot(PC15.pos, PC15.neg, pch = 20))
with(pc.scores[cell.name %in% cells.allr4.immune],points(PC15.pos, PC15.neg, col = 'green', pch = 20 ))

```



```
save(cells.allr4.immune, file = 'all_r4/all_r4_immune.RData')
write.table(cells.allr4.immune, file = 'all_r4/all_r4_immune.txt')
save(cells.allr4.rest, file = 'all_r4/all_r4_rest.RData')
write.table(cells.allr4.rest, file = 'all_r4/all_r4_rest.txt')
```

## all\_r5 (remove SVv)

PC2 and PC7 separates the Fbln2+ SVv population, which is strongly continuous with the A2m+ SVc population. There are also a few cells that appear continuous with the Apln+ CV Plexus population. The SVc and CV Plexus populations also appear strongly continuous with one another.

Separate and remove the SVv population using PC2 and PC7 to better reveal heterogeneity in the SVc-CV Plexus continuum.

```
rm(list=ls()[ls() %like% 'cells.*'])
cordin='all_r5/'
system(paste0('mkdir ',cordin));
system(paste0('mkdir ',file.path(cordin,'biplots')));

load('all_r4/all_r4_rest.RData')

dt.pca = dt.fv1_NS.good[cell.name %in% c(cells.allr4.rest)][!gene %in% genes.cc]
setkey(dt.pca)
cell.info = copy(fv1_NS.cell.info)
cell.info[, experiment := sort.plate]
cell.info[, experiment_color := sort.plate_color]
#
n.pcs = 15
if(recalculate.pca){

pca.d1 <- dimplots.pca.lowMem(dt.pca,cell.info,cordin,suffix='',max.genes = 30,ncp=n.pcs,
```

```

min.cells.detect=1,n.cores = 8)
plot.ggpairs.wExptCols(cell.info,dir=cordir,suffix='',num.pcs = n.pcs,height=25,width=25,sigType = 'Sig')
plot.ggpairs.wExptCols(cell.info,dir=cordir,suffix='',num.pcs = n.pcs,height=25,width=25,sigType = 'PC')

# make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',8)), cordir,size=1.5, plot.pdf = T, num.genes = 30)
make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',7)), cordir,size=1.5, plot.pdf = T, num.genes = 30)
}

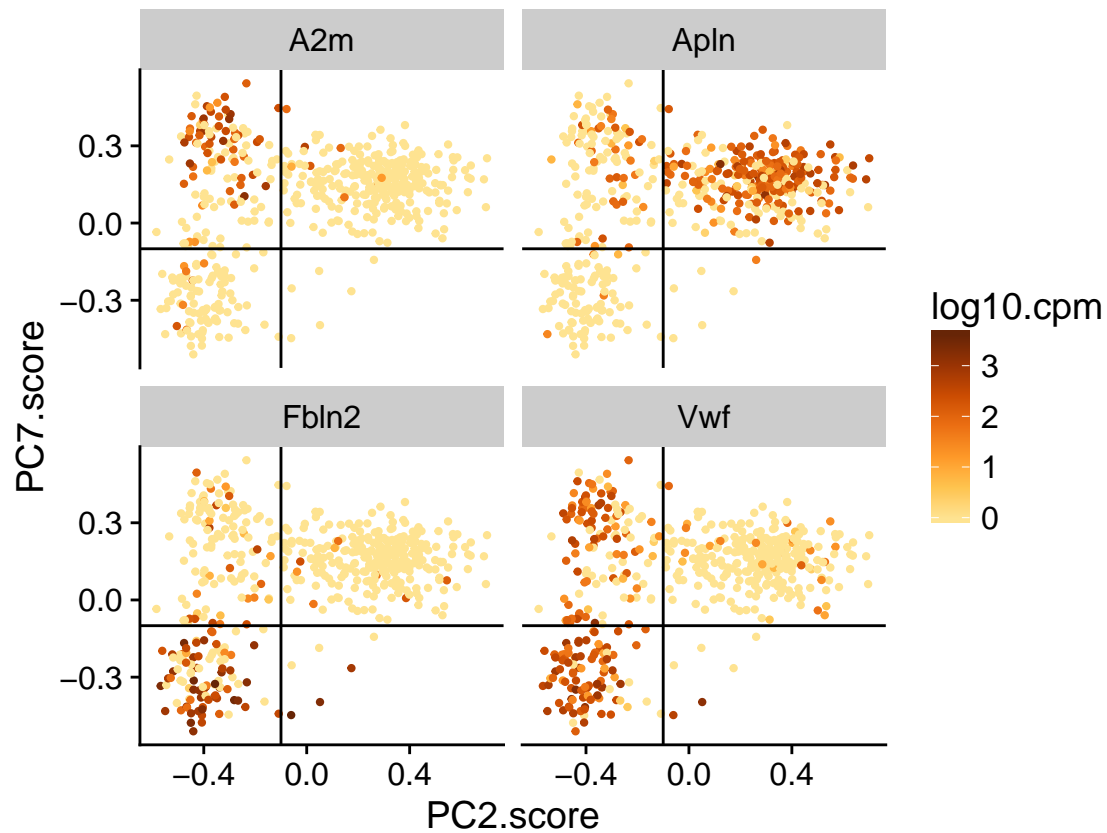
# Plot genes on PC pos/neg biplots
# registerDoMC(8)
# foreach(i=1:n.pcs) %dopar% {
#   make.facet.biplots(dt.pca,c(paste0('PC',i),paste0('PC',i)),cordir,ax.suf = c('.pos','.neg'),size=1.5, plot.pdf = T, num.genes = 30)
#   height = 18,width = 22,num.genes = 30, plot.pdf = T)
# }

pc.scores = fread(paste0(cordir, 'PC_allscores.csv'))
dt.pca.2 = merge(dt.pca[gene %in% c('Vwf','A2m','Apln','Fbln2')],
  pc.scores[,c('cell.name',colnames(pc.scores)[colnames(pc.scores) %like% "PC[0-9]"])],
  by = 'cell.name')

cutoff.pc7 <- -.1
cutoff.pc2 <- -.1

plt=ggplot(dt.pca.2, aes(PC2.score, PC7.score, color = log10.cpm)) + geom_point(size=.8) +
  scale_color_gradientn(colours=brewer.pal(9,'YlOrBr')[3:9]) +
  facet_wrap(~gene) + coord_fixed() +
  geom_hline(yintercept = cutoff.pc7) +
  geom_vline(xintercept = cutoff.pc2)
plt

```



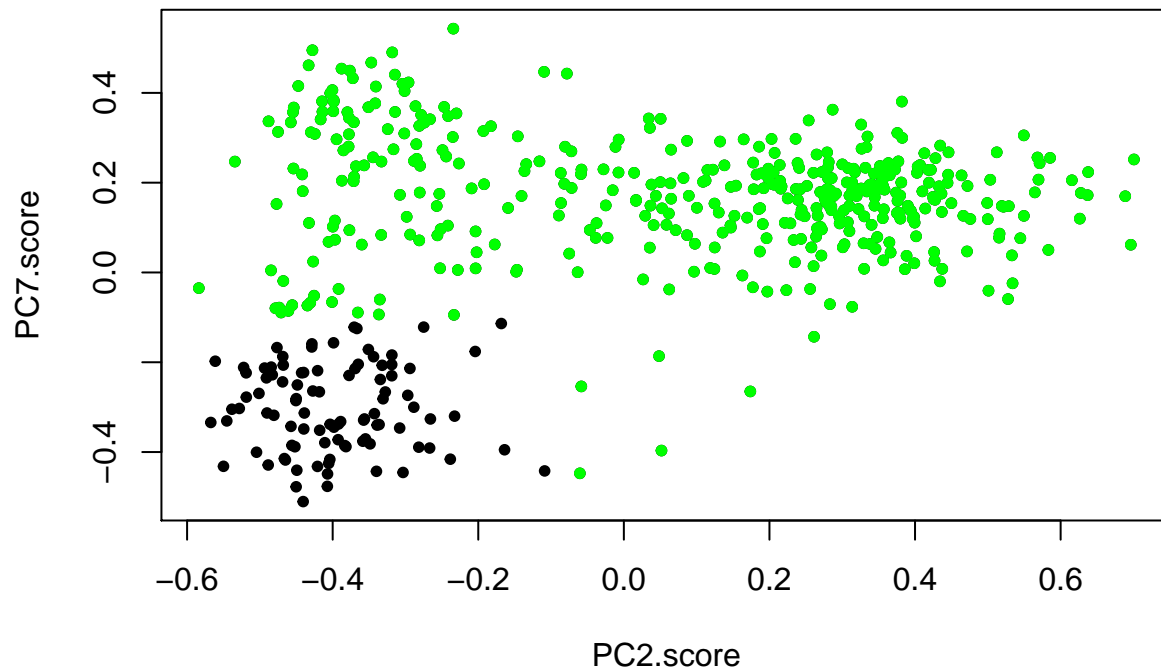
```
save_plot('all_r5/all_r5_separation.pdf',plt,ncol=2, nrow=2, base_height = 3)
```

```
cells.allr5.SVv = pc.scores[(PC2.score < cutoff.pc2) & (PC7.score < cutoff.pc7), cell.name]
cells.allr5.rest = pc.scores[!cell.name %in% cells.allr5.SVv, cell.name]
```

```
save(cells.allr5.SVv, file = 'all_r5/all_r5_SVv.RData')
write.table(cells.allr5.SVv, file = 'all_r5/all_r5_SVv.txt')
save(cells.allr5.rest, file = 'all_r5/all_r5_rest.RData')
write.table(cells.allr5.rest, file = 'all_r5/all_r5_rest.txt')
```

```
with(pc.scores, plot(PC2.score, PC7.score, pch = 20))
```

```
with(pc.scores[cell.name %in% cells.allr5.rest],points(PC2.score, PC7.score, col = 'green', pch = 20 ))
```



## all\_r6 (remove SVc)

PC2 and PC3 reveal a continuum from SVc to CV Plexus and from CV Plexus to a pre-artery (Gja4+/Nr2f2-) population.

Separate the SVc population by PC2.score to analyze better the CV Plexus-Arterial transition. Define a boundary based on the transition from Mrc1+ (SVc) to Apln+ (CV Plexus)

```
rm(list = ls()[ls() %like% "cells.*"])
cordir = "all_r6/"
system(paste0("mkdir ", cordir))
system(paste0("mkdir ", file.path(cordir, "biplots")))

load("all_r5/all_r5_rest.RData")

dt.pca = dt.fv1_NS.good[cell.name %in% cells.allr5.rest][!gene %in% genes.cc]
setkey(dt.pca)

n.pcs = 10
if (recalculate.pca) {

  pca.d1 <- dimplots.pca.lowMem(dt.pca, cell.info, cordir, suffix = "", max.genes = 30,
    ncp = n.pcs, min.cells.detect = 1, n.cores = 8, min.numgenes = 3500)
  plot.ggpairs.wExptCols(cell.info, dir = cordir, suffix = "", num.pcs = n.pcs,
    height = 25, width = 25, sigType = "Sig", annot.expt = F)

  # registerDoMC(8) foreach(i=1:n.pcs) %dopar% {
  # make.facet.biplots(dt.pca, c(paste0('PC', i), paste0('PC', i)), cordir, ax.suf =
  # c('.pos', '.neg'), size=1.5, height = 18, width = 22, num.genes = 30, plot.pdf
  # = T) }
  make.facet.biplots(dt.pca, c(paste0("PC", 2), paste0("PC", 3)), cordir,
```



```

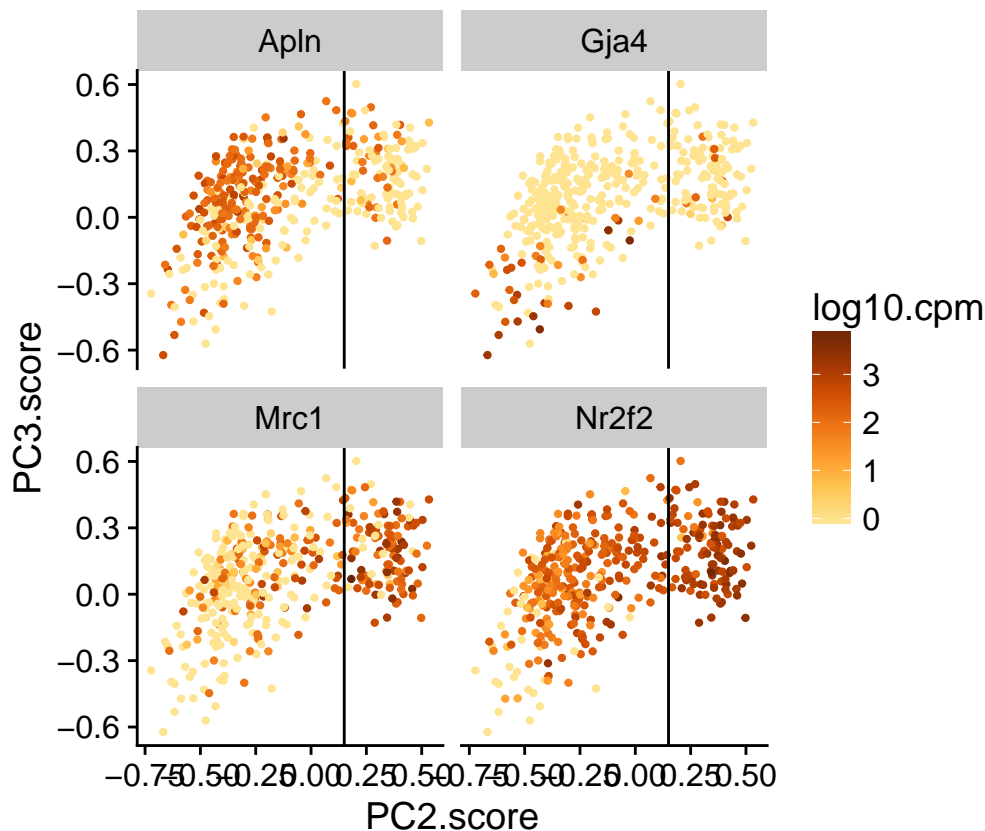
    size = 1.5, plot.pdf = T)
}

pc.scores = fread(paste0(cordir, "PC_allscores.csv"))
dt.pca.2 = merge(dt.pca[gene %in% c("Mrc1", "Gja4", "Nr2f2", "Apln")], pc.scores,
  by = "cell.name")

# Cutoff to define Svc-CV Plexus boundary
cutoff.1 = 0.15

plt = ggplot(dt.pca.2, aes(PC2.score, PC3.score, color = log10.cpm)) + geom_point(size = 0.8) +
  scale_color_gradientn(colours = brewer.pal(9, "YlOrBr")[3:9]) + facet_wrap(~gene) +
  coord_fixed() + geom_vline(xintercept = cutoff.1)
plt

```



```

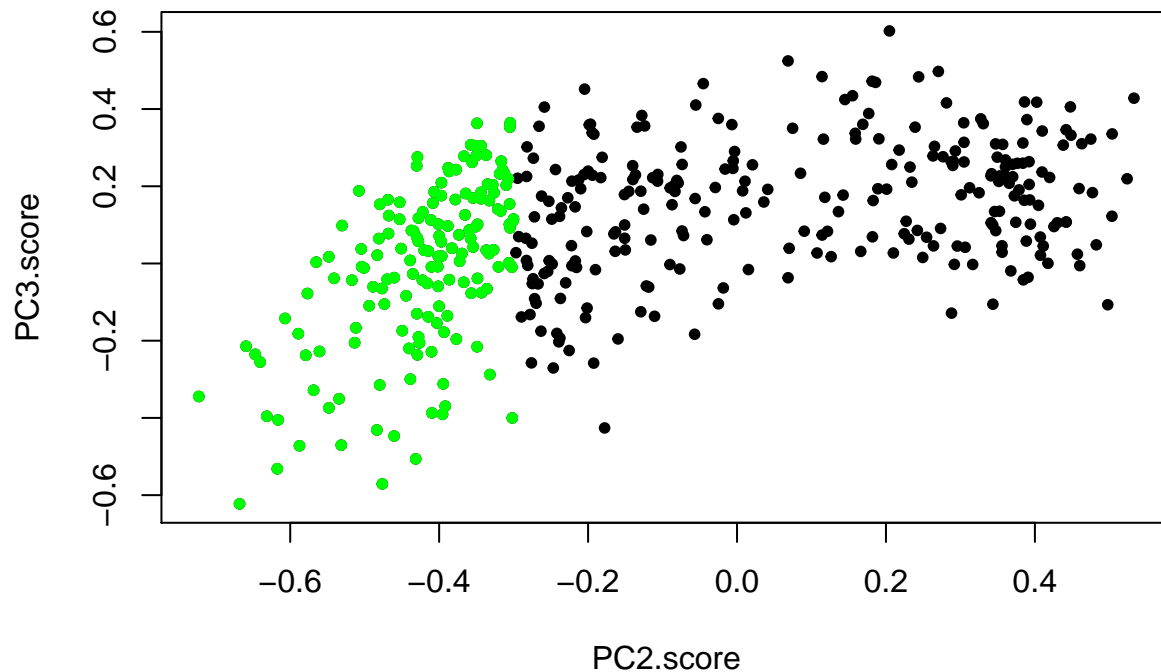
save_plot("all_r6/all_r6_separation.pdf", plt, ncol = 2, nrow = 2, base_height = 2.5)

cells.allr6.Svc = pc.scores[PC2.score > cutoff, cell.name]
cells.allr6.rest = pc.scores[!cell.name %in% cells.allr6.Svc, cell.name]

save(cells.allr6.Svc, file = "all_r6/all_r6_Svc.RData")
write.table(cells.allr6.Svc, file = "all_r6/all_r6_Svc.txt")
save(cells.allr6.rest, file = "all_r6/all_r6_rest.RData")
write.table(cells.allr6.rest, file = "all_r6/all_r6_rest.txt")

```

```
with(pc.scores, plot(PC2.score, PC3.score, pch = 20))
with(pc.scores[cell.name %in% cells.allr6.rest], points(PC2.score, PC3.score,
  col = "green", pch = 20))
```



## all\_r7 (Define Arterial cells)

PC3 separates pre-artery cells from CV plexus cells. Define the pre-artery population here based on PC3 and the transition from Nr2f2+ (CV Plexus) to Gja4+ (pre-artery).

```
rm(list=ls()[ls() %like% 'cells.*'])
cordir='all_r7/'
system(paste0('mkdir ', cordir));
system(paste0('mkdir ', file.path(cordir, 'biplots')));

load('all_r6/all_r6_rest.RData')

dt.pca = dt.fv1_NS.good[cell.name %in% cells.allr6.rest][!gene %in% genes.cc]
setkey(dt.pca)
cell.info = copy(fv1_NS.cell.info)
cell.info[, experiment := sort.plate]
cell.info[, experiment_color := sort.plate_color]
cell.info[, sum.counts := num.mouse.reads]

n.pcs = 10
if(recalculate.pca){

pca.d1 <- dimplots.pca.lowMem(dt.pca, cell.info, cordir, suffix='', max.genes = 30, ncp=n.pcs,
  min.cells.detect=1, n.cores = 8, min.numgenes = 3500)
plot.ggpairs.wExptCols(cell.info, dir=cordir, suffix='', num.pcs = n.pcs, height=25, width=25, sigType = 'Sig

# make.facet.biplots(dt.pca, c(paste0('PC', 1), paste0('PC', 3)), cordir, size=1.5, plot.pdf = T)
```

```

# make.facet.biplots(dt.pca,c(paste0('PC',1),paste0('PC',7)),cordir,size=1.5, plot.pdf = T)
# make.facet.biplots(dt.pca,c(paste0('PC',3),paste0('PC',7)),cordir,size=1.5, plot.pdf = T)

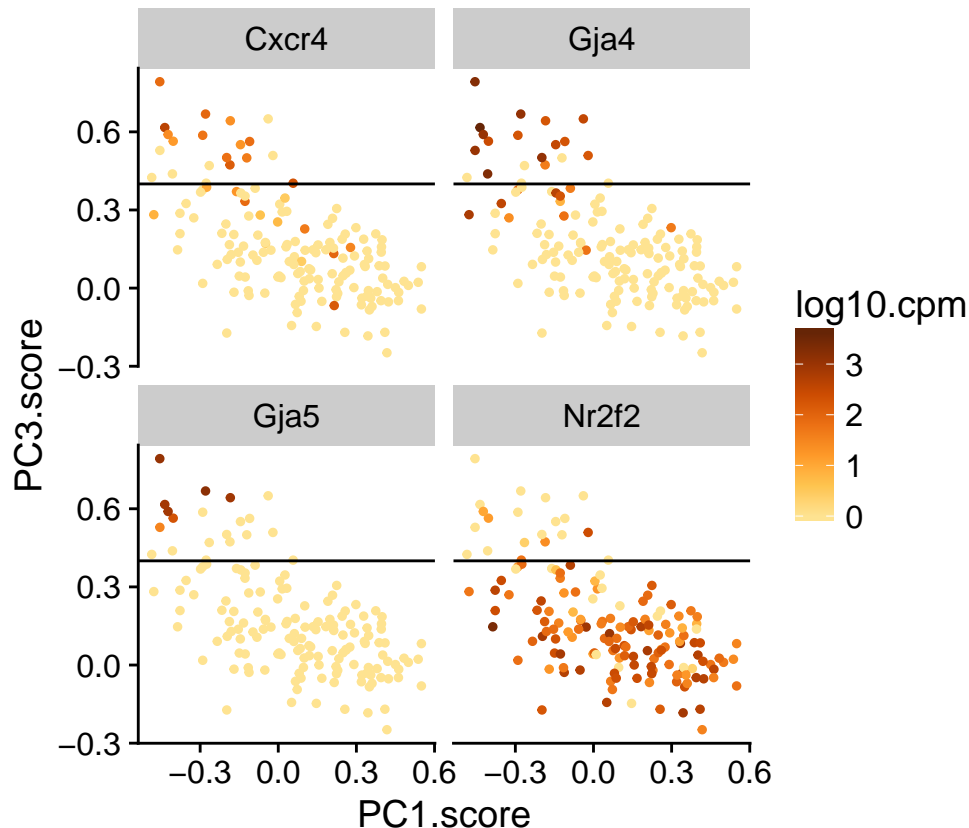
# make.facet.biplots(dt.pca,c(paste0('PC',3),paste0('PC',10)),cordir,size=1.5, plot.pdf = T)
# make.facet.biplots(dt.pca,c(paste0('PC',3),paste0('PC',9)),cordir,size=1.5, plot.pdf = T)
#
# registerDoMC(detectCores())
# foreach(i=1:n.pcs) %dopar% {
#   make.facet.biplots(dt.pca,c(paste0('PC',i),paste0('PC',i)),cordir,ax.suf = c('.pos','.neg'),size=1.5,
#     height = 12,width = 15,num.genes = 15, plot.pdf = T, suffix = '')
# }

}

# Define arterial cells
pc.scores = fread(paste0(cordir, 'PC_allscores.csv'))
dt.pca.2 = merge(dt.pca[gene %in% c('Cxcr4','Nr2f2','Gja4','Gja5')],
  pc.scores[,c('cell.name',colnames(pc.scores)[colnames(pc.scores) %like% "PC[0-9]"])], w
  by = 'cell.name')

cutoff <- .4
plt=ggplot(dt.pca.2, aes(PC1.score, PC3.score, color = log10.cpm)) + geom_point(size=1) +
  scale_color_gradientn(colours=brewer.pal(9,'YlOrBr')[3:9]) +
  facet_wrap(~gene) + coord_fixed() +
  geom_hline(yintercept = cutoff)
plt

```



```

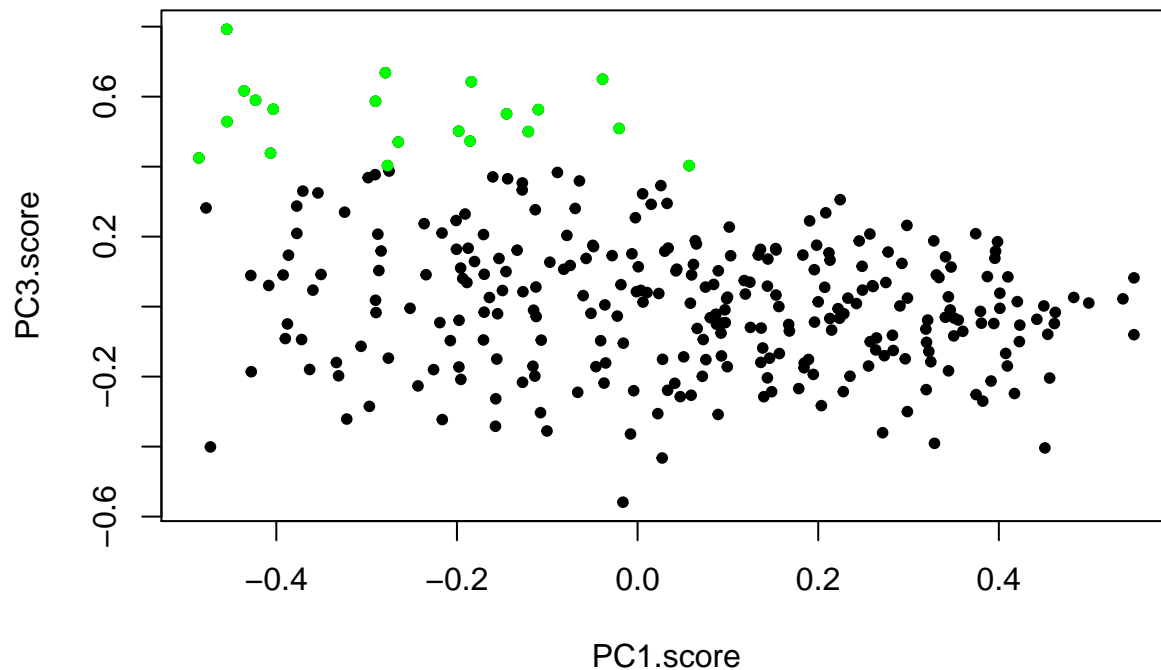
save_plot('all_r7/all_r7_separation.pdf',plt,ncol=2, nrow=2, base_height = 2.5)

if(cutoff > 0){
  cells.allr7.art1 = pc.scores[(PC3.score > cutoff), cell.name]
}else{
  cells.allr7.art1 = pc.scores[(PC3.score < cutoff), cell.name]
}
cells.allr7.rest = pc.scores[!cell.name %in% cells.allr7.art1, cell.name]

save(cells.allr7.art1, file = 'all_r7/all_r7_art1.Rdata')
write.table(cells.allr7.art1, file = 'all_r7/all_r7_art1.txt')
save(cells.allr7.rest, file = 'all_r7/all_r7_rest.RData')
write.table(cells.allr7.rest, file = 'all_r7/all_r7_rest.txt')

with(pc.scores, plot(PC1.score, PC3.score, pch = 20))
with(pc.scores[cell.name %in% cells.allr7.art1],points(PC1.score, PC3.score, col = 'green', pch = 20 ))

```



## all\_r8 (define Tbx1+ population)

PC10 separates a small population of Tbx1+ cells.

```

rm(list=ls()[ls() %like% 'cells.*'])
cordir='all_r8/'
system(paste0('mkdir ',cordir));
system(paste0('mkdir ',file.path(cordir,'biplots')));

load('all_r7/all_r7_rest.RData')

dt.pca = dt.fv1_NS.good[cell.name %in% cells.allr7.rest][!gene %in% genes.cc]
setkey(dt.pca)

```

```

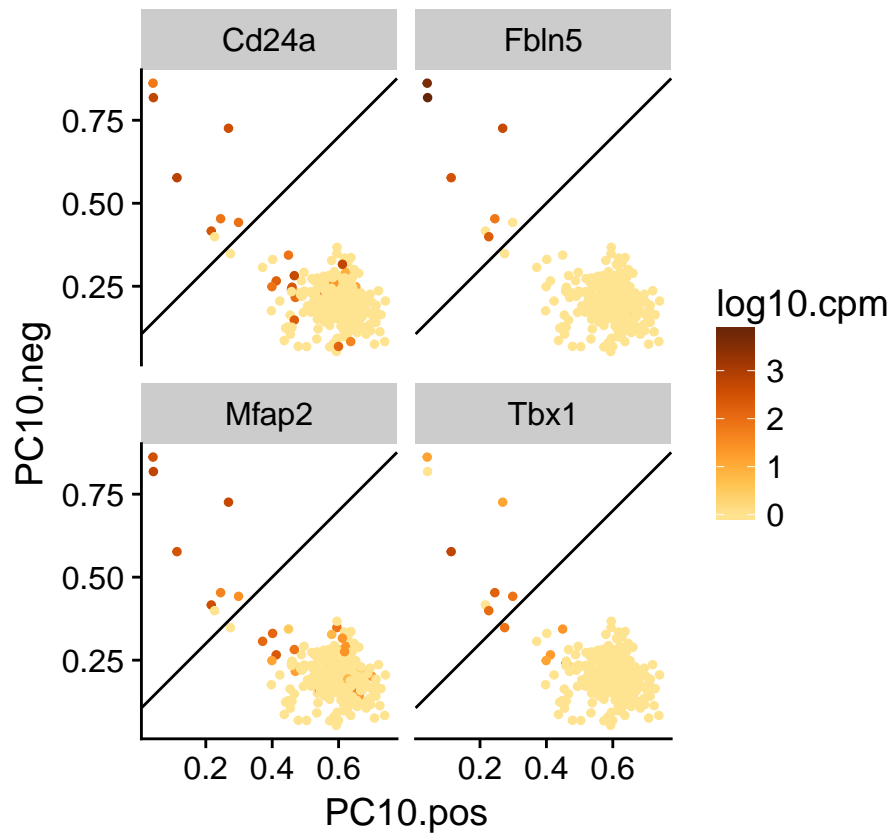
cell.info = copy(fv1_NS.cell.info)
cell.info[, experiment := sort.plate]
cell.info[, experiment_color := sort.plate_color]
cell.info[, sum.counts := num.mouse.reads]
#
n.pcs = 10
if(recalculate.pca){

pca.d1 <- dimplots.pca.lowMem(dt.pca,cell.info,cordir,suffix='',max.genes = 30,ncp=n.pcs,
                             min.cells.detect=1,n.cores = 8, min.numgenes = 3500)
plot.ggpairs.wExptCols(cell.info,dir=cordir,suffix='',num.pcs = n.pcs,height=25,width=25,sigType = 'Sig
# plot.ggpairs.wExptCols(cell.info,dir=cordir,suffix='',num.pcs = n.pcs,height=25,width=25,sigType = 'P
#
# make.facet.biplots(dt.pca,c(paste0('PC',1),paste0('PC',2)),cordir,size=1.5, plot.pdf = T)
# make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',4)),cordir,size=1.5, plot.pdf = T)
#
make.facet.biplots(dt.pca,c(paste0('PC',1),paste0('PC',10)),cordir,size=1.5, plot.pdf = T)
# make.facet.biplots(dt.pca,c(paste0('PC',3),paste0('PC',10)),cordir,size=1.5, plot.pdf = T)
# make.facet.biplots(dt.pca,c(paste0('PC',3),paste0('PC',9)),cordir,size=1.5, plot.pdf = T)
#
# registerDoMC(detectCores())
# foreach(i=c(1:2,10)) %dopar% {
#   make.facet.biplots(dt.pca,c(paste0('PC',i),paste0('PC',i)),cordir,ax.suf = c('.pos','neg'),size=1.
#   height = 12,width = 15,num.genes = 15, plot.pdf = T, suffix = '')
# }
}

pc.scores = fread(paste0(cordir, 'PC_allscores.csv'))
dt.pca.2 = merge(dt.pca[gene %in% c('Tbx1','Fbln5','Cd24a','Mfap2')],
                 pc.scores[,c('cell.name',colnames(pc.scores)[colnames(pc.scores) %like% "PC[0-9]"])], w
                 by = 'cell.name')

cutoff=-.1
plt=ggplot(dt.pca.2, aes(PC10.pos, PC10.neg, color = log10.cpm)) + geom_point(size=1) +
  scale_color_gradientn(colours=brewer.pal(9,'YlOrBr')[3:9]) +
  facet_wrap(~gene) + coord_fixed() +
  geom_abline(intercept = -cutoff)
plt

```

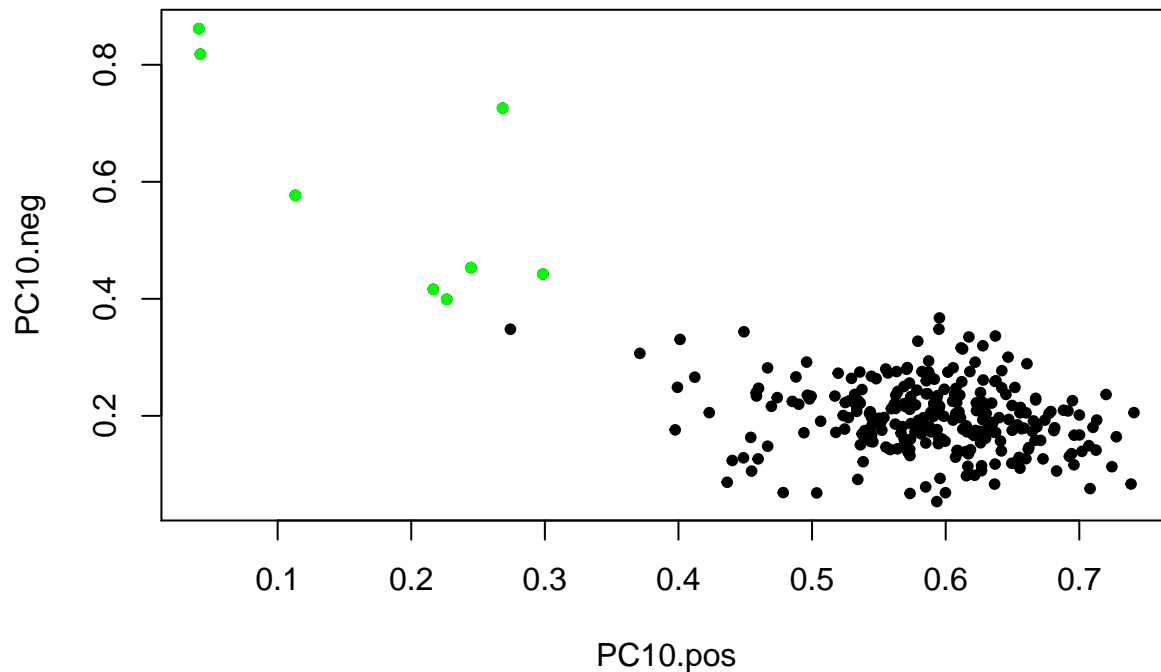


```
save_plot('all_r8/all_r8_separation.pdf',plt,ncol=2, nrow=2, base_height = 2.5)

if(cutoff < 0){
  cells.allr8.Tbx1 = pc.scores[PC10.score < cutoff, cell.name]
}else{
  cells.allr8.Tbx1 = pc.scores[PC10.score > cutoff, cell.name]
}
cells.allr8.rest = pc.scores[!cell.name %in% cells.allr8.Tbx1, cell.name]

save(cells.allr8.Tbx1, file = 'all_r8/all_r8_Tbx1.RData')
write.table(cells.allr8.Tbx1, file = 'all_r8/all_r8_Tbx1.txt')
save(cells.allr8.rest, file = 'all_r8/all_r8_rest.RData')
write.table(cells.allr8.rest, file = 'all_r8/all_r8_rest.txt')

with(pc.scores, plot(PC10.pos, PC10.neg, pch = 20))
with(pc.scores[cell.name %in% cells.allr8.Tbx1],points(PC10.pos, PC10.neg, col = 'green', pch = 20 ))
```



## all\_r9 (define CV1/CV2)

PC2 separates CV Plexus cells expressing retinal tip-associated markers (Id2, Rgs2) and retinal stalk-associated markers (Apln). Separate them at the transition from Rgs2+ to Apln+ (CV1 cells are Rgs+, CV2 cells are Apln+) using PC2.

```
rm(list=ls()[ls() %like% 'cells.*'])
cordir='all_r9/'
system(paste0('mkdir ',cordir));
system(paste0('mkdir ',file.path(cordir,'biplots')));

load('all_r8/all_r8_rest.RData')

dt.pca = dt.fv1_NS.good[cell.name %in% cells.allr8.rest][!gene %in% genes.cc]
setkey(dt.pca)
cell.info = copy(fv1_NS.cell.info)
cell.info[, experiment := sort.plate]
cell.info[, experiment_color := sort.plate_color]
cell.info[, sum.counts := num.mouse.reads]
# #
n.pcs = 10
if(recalculate.pca){

pca.d1 <- dimplots.pca.lowMem(dt.pca,cell.info,cordir,suffix='',max.genes = 30,ncp=n.pcs,
                             min.cells.detect=1,n.cores = 8, min.numgenes = 3500)
plot.ggpairs.wExptCols(cell.info,dir=cordir,suffix='',num.pcs = n.pcs,height=25,width=25,sigType = 'Sig
# plot.ggpairs.wExptCols(cell.info,dir=cordir,suffix='',num.pcs = n.pcs,height=25,width=25,sigType = 'P
#
make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',2)),cordir,size=1.5, plot.pdf = T, ax.suf = c('.
#
# registerDoMC(detectCores())
```

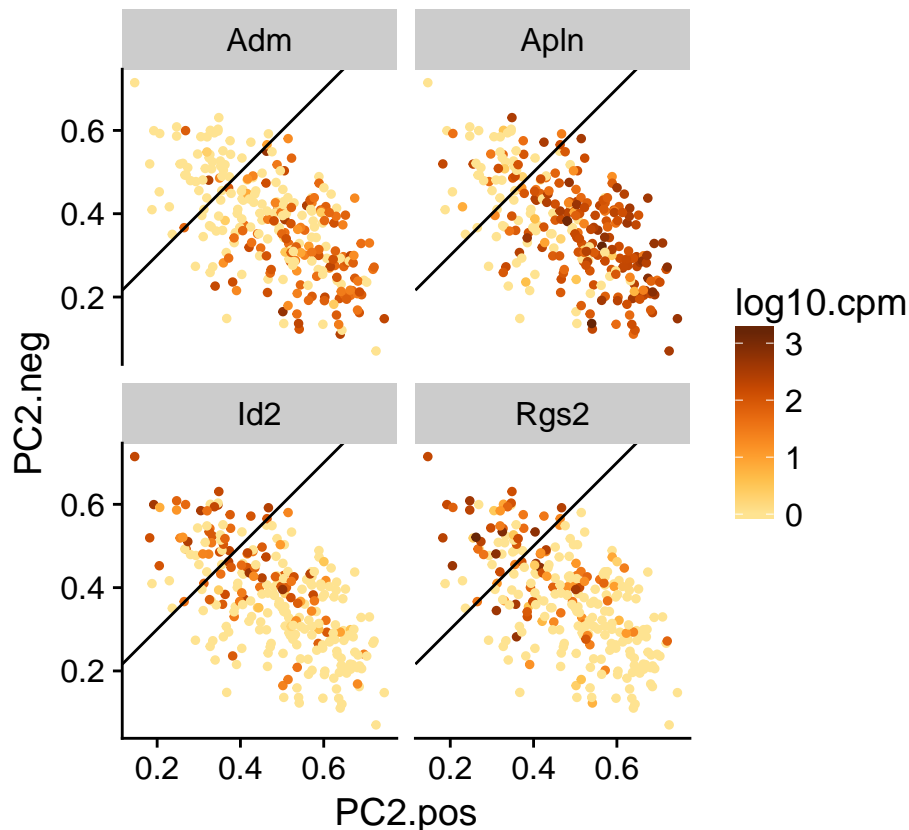
```

# foreach(i=c(1:2,10)) %dopar% {
#   make.facet.biplots(dt.pca,c(paste0('PC',i),paste0('PC',i)),cordir,ax.suf = c('.pos','.neg'),size=1.
#   height = 12,width = 15,num.genes = 15, plot.pdf = T, suffix = '')
# }
}

pc.scores = fread(paste0(cordir, 'PC_allscores.csv'))
dt.pca.2 = merge(dt.pca[gene %in% c('Apln','Adm','Id2','Rgs2')],
  pc.scores[,c('cell.name',colnames(pc.scores)[colnames(pc.scores) %like% "PC[0-9]"])], w
  by = 'cell.name')

cutoff=-.1
plt=ggplot(dt.pca.2, aes(PC2.pos, PC2.neg, color = log10.cpm)) + geom_point(size=1) +
  scale_color_gradientn(colours=brewer.pal(9,'YlOrBr')[3:9]) +
  facet_wrap(~gene) + coord_fixed() +
  geom_abline(intercept = -cutoff)
plt

```



```

save_plot(paste0(cordir,'all_r9_separation.pdf'),plt,ncol=2, nrow=2, base_height = 2.5)

# define CV1 and CV2. Make sure CV1 is Apln+
cells.allr9.cv2 = pc.scores[PC2.score < cutoff, cell.name]
cells.allr9.cv1 = pc.scores[!cell.name %in% cells.allr9.cv2, cell.name]

save(cells.allr9.cv2, file = paste0(cordir,'all_r9_CV2.RData'))
write.table(cells.allr9.cv2, file = paste0(cordir,'all_r9_CV2.txt'))

```

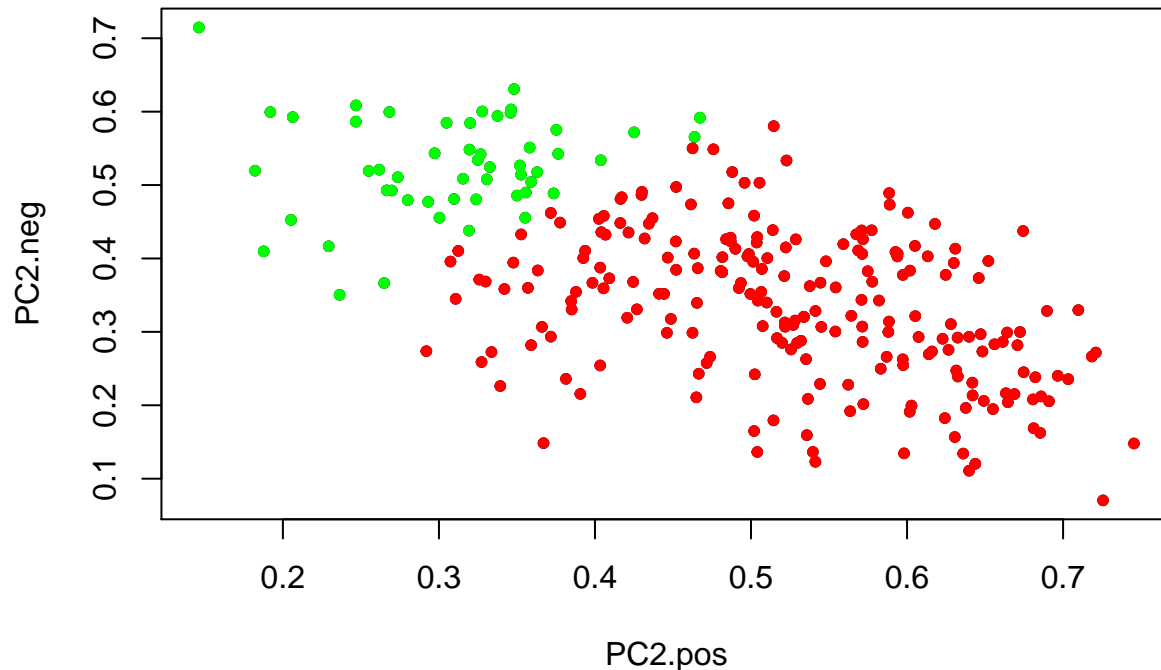


```

save(cells.allr9.cv1, file = paste0(cordir, 'all_r9_CV1.RData'))
write.table(cells.allr9.cv1, file = paste0(cordir, 'all_r9_CV1.txt'))

with(pc.scores, plot(PC2.pos, PC2.neg, pch = 20))
with(pc.scores[cell.name %in% cells.allr9.cv2], points(PC2.pos, PC2.neg, col = 'green', pch = 20))
with(pc.scores[cell.name %in% cells.allr9.cv1], points(PC2.pos, PC2.neg, col = 'red', pch = 20))

```



## SVv-SVc (define SVc)

In all\_r4, the cutoff separating SVc from CV plexus was defined. Now define the cutoff between SVc and SVv and define SVc cells. PC2 clearly separates SVv (Fbln2+) from SVc (Mrc1+). Separate them at the transition between expression of these two genes using PC2.

The boundaries between SVc and the two populations it is continuous with have now been defined and thus the population is now defined.

```

rm(list=ls()[ls() %like% 'cells.*'])
cordir='SVv_SVc/'
system(paste0('mkdir ', cordir));
system(paste0('mkdir ', file.path(cordir, 'biplots')));

load('all_r5/all_r5_SVv.RData')
load('all_r6/all_r6_SVc.RData')

dt.pca = dt.fv1_NS.good[cell.name %in% c(cells.allr5.SVv, cells.allr6.SVc)][!gene %in% genes.cc]
setkey(dt.pca)
cell.info = copy(fv1_NS.cell.info)
cell.info[, experiment := sort.plate]
cell.info[, experiment_color := sort.plate_color]

```

```

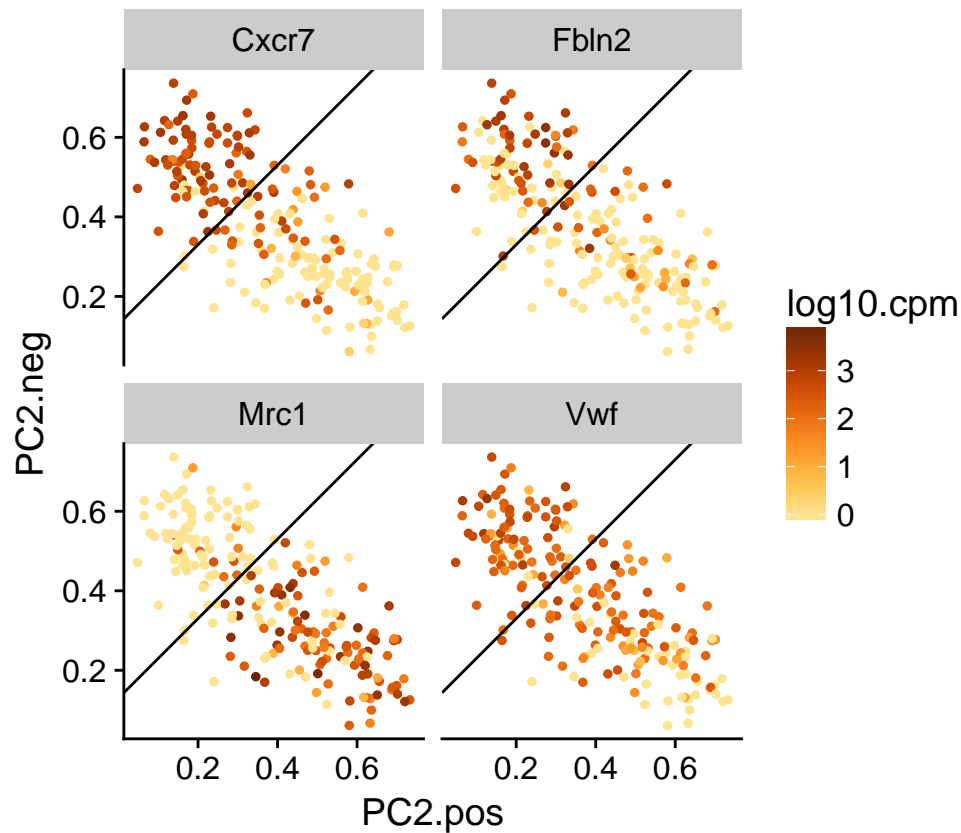
cell.info[, sum.counts := num.mouse.reads]

#
n.pcs = 10
if(recalculate.pca){
pca.d1 <- dimplots.pca.lowMem(dt.pca,cell.info,cordir,suffix='',max.genes = 30,ncp=n.pcs,
                             min.cells.detect=1,n.cores = 8, min.numgenes = 3500)
plot.ggpairs.wExptCols(cell.info,dir=cordir,suffix='',num.pcs = n.pcs,height=25,width=25,sigType = 'Sig
#
# make.facet.biplots(dt.pca,c(paste0('PC',1),paste0('PC',2)),cordir,size=1.5, plot.pdf = T)
# make.facet.biplots(dt.pca,c(paste0('PC',1),paste0('PC',3)),cordir,size=1.5, plot.pdf = T)
# make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',3)),cordir,size=1.5, plot.pdf = T)
# make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',9)),cordir,size=1.5, plot.pdf = T)
#
# registerDoMC(detectCores())
# foreach(i=c(1:2,10)) %dopar% {
#   make.facet.biplots(dt.pca,c(paste0('PC',i),paste0('PC',i)),cordir,ax.suf = c('.pos','.neg'),size=1.
#   height = 12,width = 15,num.genes = 15, plot.pdf = T, suffix = '')
# }
}

pc.scores = fread(paste0(cordir, 'PC_allscores.csv'))
dt.pca.2 = merge(dt.pca[gene %in% c('Fbln2','Cxcr7','Vwf','Mrc1')],
                 pc.scores[,c('cell.name',colnames(pc.scores)[colnames(pc.scores) %like% "PC[0-9]"])], w
                 by = 'cell.name')

cutoff = -.13
plt=ggplot(dt.pca.2, aes(PC2.pos, PC2.neg, color = log10.cpm)) + geom_point(size=1) +
  scale_color_gradientn(colours=brewer.pal(9,'YlOrBr')[3:9]) +
  facet_wrap(~gene) + coord_fixed() +
  geom_abline(intercept = -cutoff)
plt

```

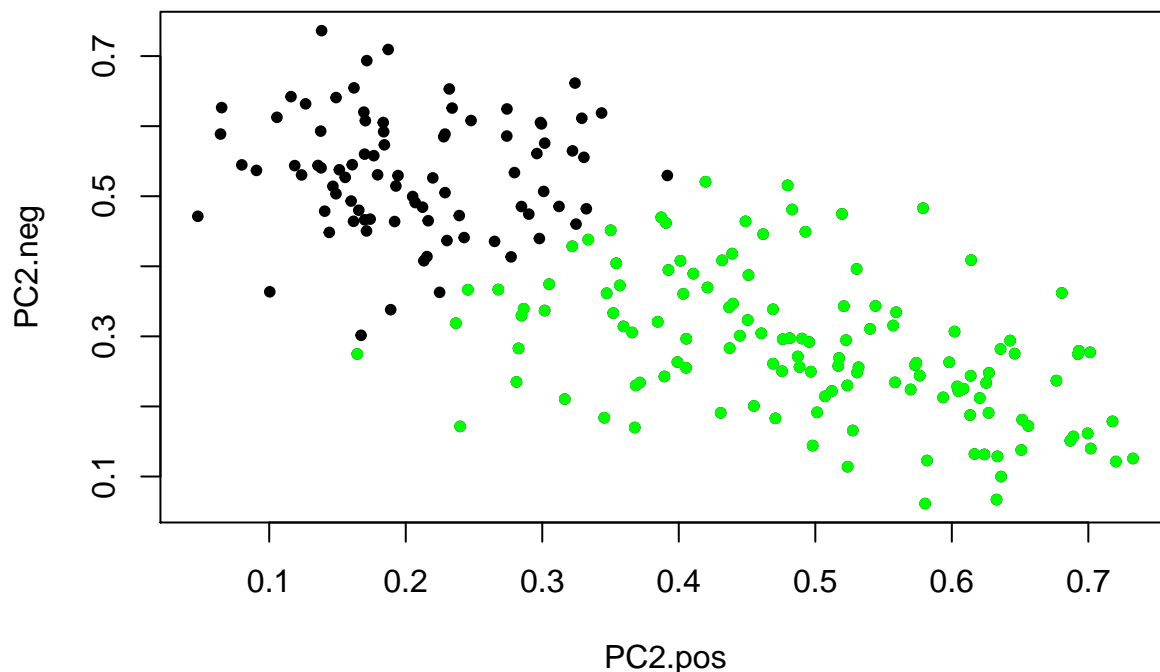


```
save_plot(paste0(cordir,'separation.pdf'),plt,ncol=2, nrow=2, base_height = 2.5)

# Define boundary between SVc and SVv. Make sure SVv is Fbln2+ and SVc is Mrc1+!
cells.allSVv_SVc.SVc = pc.scores[PC2.score > cutoff, cell.name]
cells.allSVv_SVc.rest = pc.scores[!cell.name %in% cells.allSVv_SVc.SVc, cell.name]

save(cells.allSVv_SVc.SVc, file = paste0(cordir,'all_SVv_SVc_SVc.RData'))
write.table(cells.allSVv_SVc.SVc, file = paste0(cordir,'all_SVv_SVc_SVc.txt'))
save(cells.allSVv_SVc.rest, file = paste0(cordir,'all_SVv_SVc_rest.RData'))
write.table(cells.allSVv_SVc.rest, file = paste0(cordir,'all_SVv_SVc_rest.txt'))

with(pc.scores, plot(PC2.pos, PC2.neg, pch = 20))
with(pc.scores[cell.name %in% cells.allSVv_SVc.SVc],points(PC2.pos, PC2.neg, col = 'green', pch = 20 ))
```



## All r10 (remove SVv)

All cells minus the populations already defined: endocardial, SVc, CV Plexus, Arterial, Tbx1, Fbln5, immune, and Hbb+. R

PC2: strongest (most correlated) signature: Aplnr-Fst SVv-valve gradient.

PC9: Pdgfra+ cells PC10: cell cycle

```
cordir='all_r10/'
rm(list=ls()[ls() %like% 'cells.*'])
load('all_r1/cells_hbb_hi.RData')
load("endocard_refine_r5/cells_endocardRefine.r5.v2.endocard.Rdata")
load("all_r8/all_r8_Tbx1.RData")
load("all_r9/all_r9_CV1.RData")
load("all_r9/all_r9_CV2.RData")
load("SVv_SVc/all_SVv_SVc_SVc.RData")
load("all_r7/all_r7_artl.Rdata")
load("all_r4/all_r4_immune.RData")

cells.already.defined = c(cells.allr4.immune, cells.hbb.hi, cells.endocardRefine.r5.v2.endocard, cells.)

dt.pca = dt.fv1_NS.good[!cell.name %in% c(cells.already.defined)][!gene %in% genes.cc]
setkey(dt.pca)

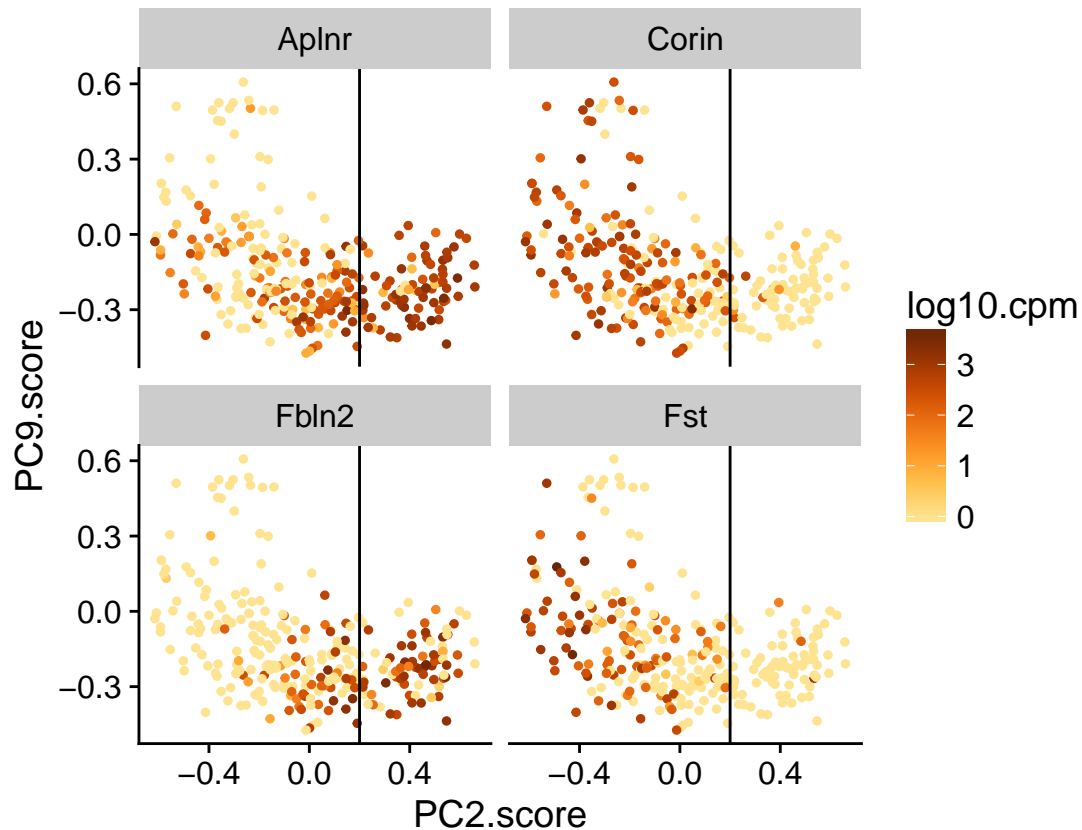
n.pcs = 10
if(recalculate.pca){
pca.d1 <- dimplots.pca.lowMem(dt.pca, cell.info, cordir, suffix='', max.genes = 30, ncp=n.pcs,
                             min.cells.detect=1, n.cores = 8, min.numgenes = 3500)
plot.ggpairs.wExptCols(cell.info, dir=cordir, suffix='', num.pcs = n.pcs, height=25, width=25, sigType = 'Sig
# plot.ggpairs.wExptCols(cell.info, dir=cordir, suffix='', num.pcs = n.pcs, height=25, width=25, sigType = 'P
#
```

```

make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',9)),cordir,size=1.5, plot.pdf = T)
# make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',5)),cordir,size=1.5, plot.pdf = T)
# # make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',3)),cordir,size=1.5, plot.pdf = T)
# # make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',9)),cordir,size=1.5, plot.pdf = T)
#
# registerDoMC(detectCores())
# foreach(i=1:n.pcs) %dopar% {
#   make.facet.biplots(dt.pca,c(paste0('PC',i),paste0('PC',i)),cordir,ax.suf = c('.pos','.neg'),size=1.5,
#     height = 12,width = 15,num.genes = 15, plot.pdf = T, suffix = '')
# }
#
}
pc.scores = fread(paste0(cordir, 'PC_allscores.csv'))
dt.pca.2 = merge(dt.pca[gene %in% c('Aplnr','Fbln2','Fst','Corin')],
  pc.scores[,c('cell.name',colnames(pc.scores)[colnames(pc.scores) %like% "PC[0-9]"])], w
  by = 'cell.name')

cutoff = .2
plt=ggplot(dt.pca.2, aes(PC2.score, PC9.score, color = log10.cpm)) + geom_point(size=1) +
  scale_color_gradientn(colours=brewer.pal(9,'YlOrBr')[3:9]) +
  facet_wrap(~gene) + coord_fixed() +
  geom_vline(xintercept = cutoff)
plt

```



```

save_plot(paste0(cordir,'separation.pdf'),plt,ncol=2, nrow=2, base_height = 2.5)

```

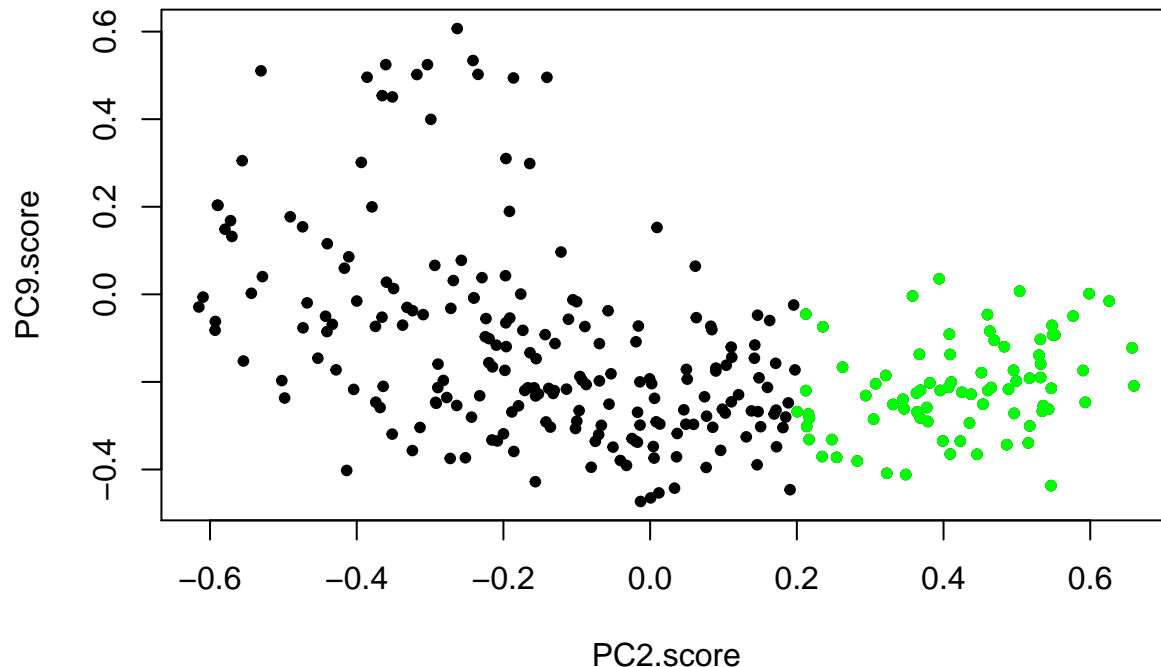
```

cells.allr10.aplnr = pc.scores[PC2.score > cutoff, cell.name]
cells.allr10.rest = pc.scores[!cell.name %in% cells.allr10.aplnr, cell.name]

save(cells.allr10.aplnr, file = paste0(cordir, 'all_r10_SVv.RData'))
write.table(cells.allr10.aplnr, file = paste0(cordir, 'all_r10_SVv.txt'))
save(cells.allr10.rest, file = paste0(cordir, 'all_r10_rest.RData'))
write.table(cells.allr10.rest, file = paste0(cordir, 'all_r10_rest.txt'))

with(pc.scores, plot(PC2.score, PC9.score, pch = 20))
with(pc.scores[cell.name %in% cells.allr10.aplnr], points(PC2.score, PC9.score, col = 'green', pch = 20))

```



## All r11 (Define mesenchymal 1/2)

All cells minus endocardial, SVc, CV Plexus, Arterial, Tbx1, Fbln5, immune, RBC, most of SVv (left: mesenchyme and valve)

PC2 and PC3 reveal a 2-D continuum: one continuum from Mesenchymal-like (Pdgfra+) to venous valve-like (Nrp1+). A second continuum separates Mesenchymal cells into two parts: one Wnt2+ (Mes1) and one Postn+ (Mes2). Define mesenchymal cells (Mes1 and Mes2) here because the Pdgfra signature is well-separated and there is little other heterogeneity.

```

cordir='all_r11/'
rm(list=ls()[ls() %like% 'cells.*'])
load('all_r10/all_r10_rest.RData')

dt.pca = dt.fv1_NS.good[cell.name %in% c(cells.allr10.rest)][!gene %in% genes.cc]
setkey(dt.pca)

n.pcs = 10
if(recalculate.pca){
pca.d1 <- dimplots.pca.lowMem(dt.pca, cell.info, cordir, suffix='', max.genes = 30, ncp=n.pcs,

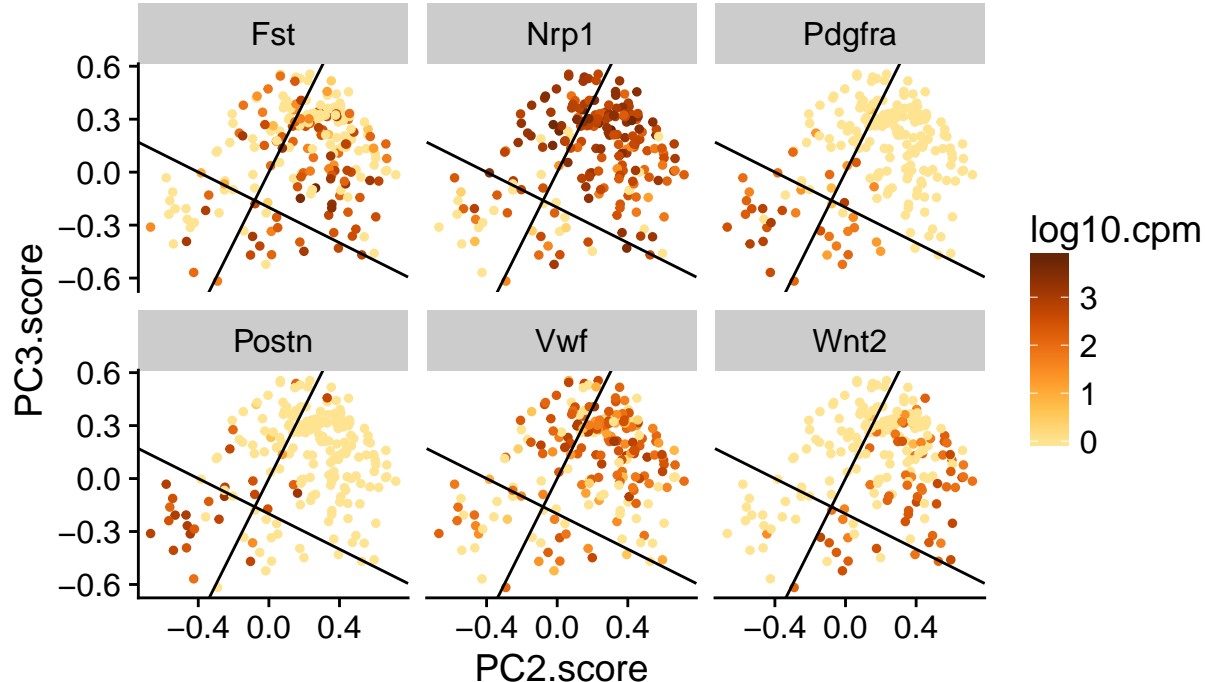
```

```

min.cells.detect=1,n.cores = 8, min.numgenes = 3500)
plot.ggpairs.wExptCols(cell.info,dir=cordir,suffix='',num.pcs = n.pcs,height=25,width=25,sigType = 'Sig
# plot.ggpairs.wExptCols(cell.info,dir=cordir,suffix='',num.pcs = n.pcs,height=25,width=25,sigType = 'P
#
# make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',4)),cordir,size=1.5, plot.pdf = T)
# # make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',5)),cordir,size=1.5, plot.pdf = T)
# # make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',3)),cordir,size=1.5, plot.pdf = T)
# # make.facet.biplots(dt.pca,c(paste0('PC',2),paste0('PC',9)),cordir,size=1.5, plot.pdf = T)
#
# registerDoMC(detectCores())
# foreach(i=1:n.pcs) %dopar% {
#   make.facet.biplots(dt.pca,c(paste0('PC',i),paste0('PC',i)),cordir,ax.suf = c('.pos','.neg'),size=1.
#   height = 12,width = 15,num.genes = 15, plot.pdf = T, suffix = '')
# }
}
pc.scores = fread(paste0(cordir, 'PC_allscores.csv'))
dt.pca.2 = merge(dt.pca[gene %in% c('Postn','Pdgfra','Nrp1','Fst', 'Vwf','Wnt2')],
  pc.scores[,c('cell.name',colnames(pc.scores)[colnames(pc.scores) %like% "PC[0-9]"])], w
  by = 'cell.name')

plt=ggplot(dt.pca.2, aes(PC2.score, PC3.score, color = log10.cpm)) + geom_point(size=1) +
  scale_color_gradientn(colours=brewer.pal(9,'YlOrBr')[3:9]) +
  facet_wrap(~gene) + coord_fixed() +
  geom_abline(slope=-.5, intercept = -.2) +
  geom_abline(slope=2, intercept = 0)
plt

```



```

save_plot(paste0(cordir,'separation.pdf'),plt,ncol=3, nrow=2, base_height = 2.5)
#
cells.allr11.Mes1 = pc.scores[(PC3.score < -.5*PC2.score -.2) & (PC3.score < 2*PC2.score), cell.name]
cells.allr11.Mes2 = pc.scores[(PC3.score < -.5*PC2.score -.2) & (PC3.score >= 2*PC2.score), cell.name]
cells.allr11.rest = pc.scores[!cell.name %in% c(cells.allr11.Mes1, cells.allr11.Mes2), cell.name]

```

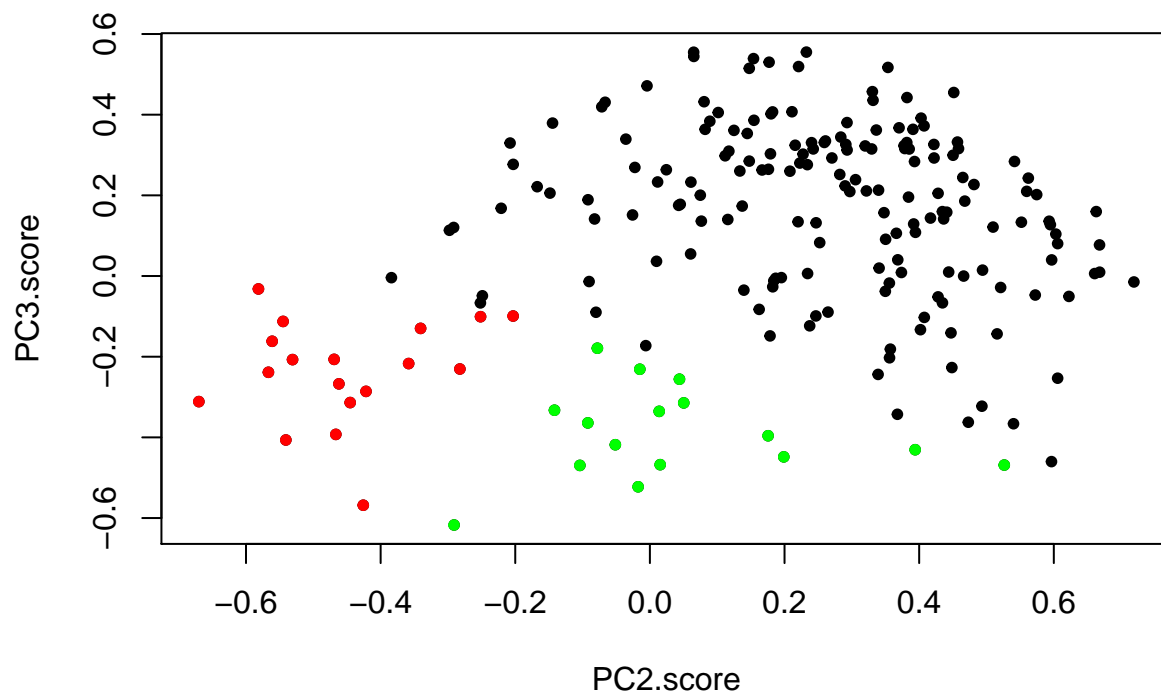
```

save(cells.allr11.Mes1, file = paste0(cordir, 'all_r11_Mes1.RData'))
write.table(cells.allr11.Mes1, file = paste0(cordir, 'all_r11_Mes1.txt'))
save(cells.allr11.Mes2, file = paste0(cordir, 'all_r11_Mes2.RData'))
write.table(cells.allr11.Mes2, file = paste0(cordir, 'all_r11_Mes2.txt'))

save(cells.allr11.rest, file = paste0(cordir, 'all_r11_rest.RData'))
write.table(cells.allr11.rest, file = paste0(cordir, 'all_r11_rest.txt'))

with(pc.scores, plot(PC2.score, PC3.score, pch = 20))
with(pc.scores[cell.name %in% cells.allr11.Mes1], points(PC2.score, PC3.score, col = 'green', pch = 20))
with(pc.scores[cell.name %in% cells.allr11.Mes2], points(PC2.score, PC3.score, col = 'red', pch = 20))

```



## all\_r12 (define VV1, and VV2)

All cells minus previously-defined populations: endocardial, SVc, CV Plexus, Arterial, Fbln5, immune, RBC, and Mes1/Mes2.

The top PCs show two main gradients: SVv (Fbln2+) to venous valve (Corin+/Fst+). There is also a very clear gradient within venous valve: VV1 (Cfh+) to VV2 (Wnt2+). This is a matching gradient to the Mes1/Mes2 gradient - it could reflect spatial variation in the venous valve and mesenchymal progenitors.

Separate cells into SVv, VV1, and VV2 using PC2 and PC3.

PC10 separates a small group of cycling cells.

```

cordir='all_r12/'
rm(list=ls()[ls() %like% 'cells.*'])
load('all_r11/all_r11_rest.RData')
load('all_r10/all_r10_SVv.RData')

dt.pca = dt.fv1_NS.good[cell.name %in% c(cells.allr11.rest, cells.allr10.aplnr)][!gene %in% genes.cc]

```



```

setkey(dt.pca)
cell.info = copy(fv1_NS.cell.info)
cell.info[, experiment := sort.plate]
cell.info[, experiment_color := sort.plate_color]
cell.info[, sum.counts := num.mouse.reads]

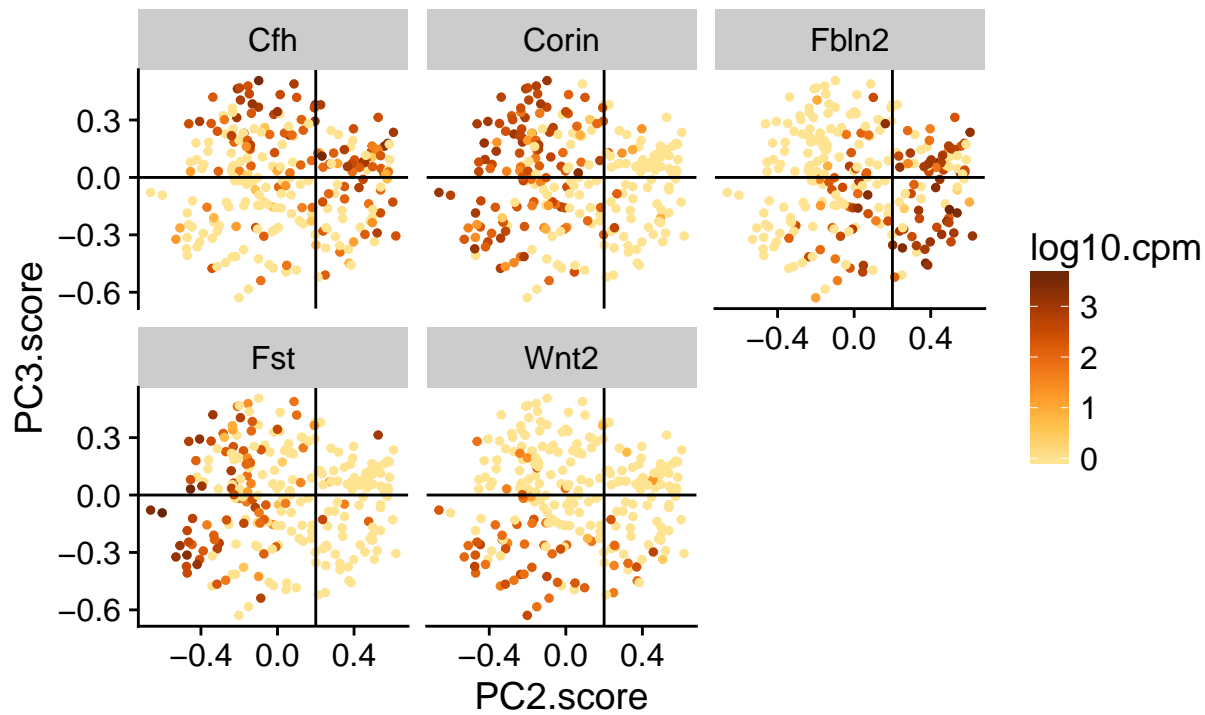
# n.pcs = 10
if(recalculate.pca){
pca.d1 <- dimplots.pca.lowMem(dt.pca, cell.info, cordir, suffix='', max.genes = 30, ncp=n.pcs,
                             min.cells.detect=1, n.cores = 8, min.numgenes = 3500)
plot.ggpairs.wExptCols(cell.info, dir=cordir, suffix='', num.pcs = n.pcs, height=25, width=25, sigType = 'Sig')
make.facet.biplots(dt.pca, c(paste0('PC', 2), paste0('PC', 3)), cordir, size=1.5, plot.pdf = T)
# make.facet.biplots(dt.pca, c(paste0('PC', 2), paste0('PC', 9)), cordir, size=1.5, plot.pdf = T)
}

pc.scores = fread(paste0(cordir, 'PC_allscores.csv'))
dt.pca.2 = merge(dt.pca[gene %in% c('Wnt2', 'Cfh', 'Corin', 'Fbln2', 'Fst')],
                pc.scores,
                by = 'cell.name')

# Determine cutoffs for SVv and VV1/VV2
cutoff.SVv <- .2
cutoff.VV <- 0
plt=ggplot(dt.pca.2, aes(PC2.score, PC3.score, color = log10.cpm)) + geom_point(size=1) +
  scale_color_gradientn(colours=brewer.pal(9, 'YlOrBr')[3:9]) +
  facet_wrap(~gene) + coord_fixed() +
  geom_vline(xintercept = cutoff.SVv) +
  geom_hline(yintercept = cutoff.VV)

plt

```



```
save_plot(paste0(cordir, 'separation.pdf'), plt, ncol=2, nrow=2, base_height = 2.5)

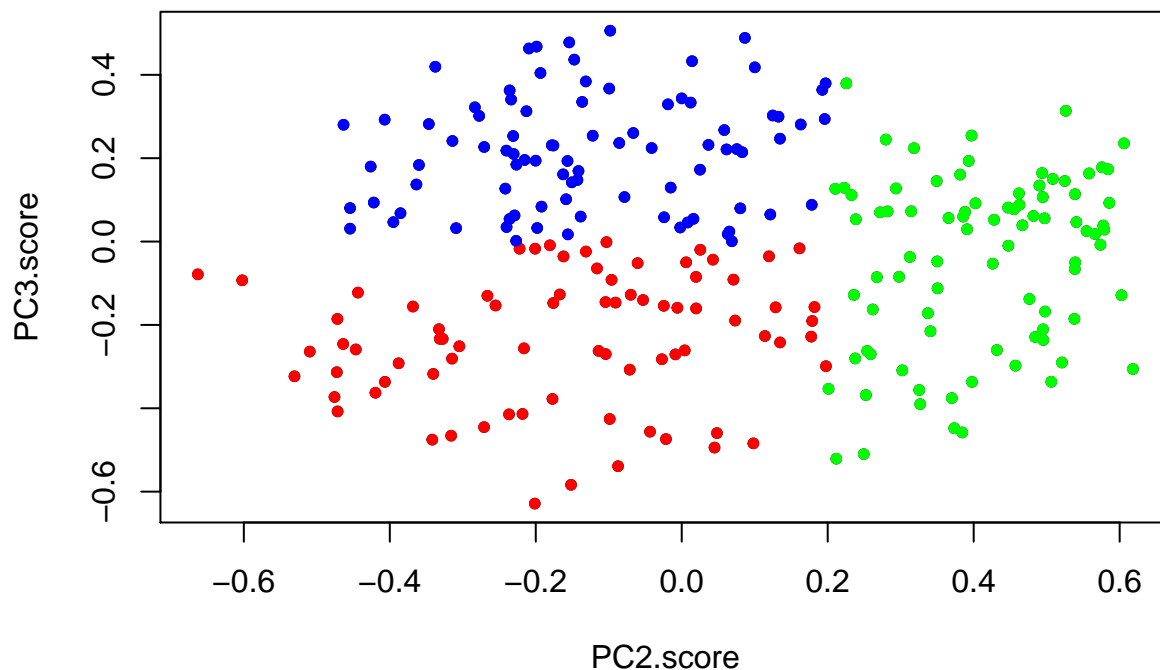
# Separate SVv cells - make sure they are Fbln2-hi!
# VV1 is defined as Wnt2+, and VV2 is Cfh+
cells.allr12.SVv = pc.scores[PC2.score > cutoff.SVv, cell.name]
cells.allr12.VV1 = pc.scores[!cell.name %in% cells.allr12.SVv][PC3.score < cutoff.VV, cell.name]
cells.allr12.VV2 = pc.scores[!cell.name %in% cells.allr12.SVv][PC3.score > cutoff.VV, cell.name]

save(cells.allr12.SVv, file = paste0(cordir, 'all_r12_SVv.RData'))
write.table(cells.allr12.SVv, file = paste0(cordir, 'all_r12_SVv.txt'))

save(cells.allr12.VV1, file = paste0(cordir, 'all_r12_VV1.RData'))
write.table(cells.allr12.VV1, file = paste0(cordir, 'all_r12_VV1.txt'))

save(cells.allr12.VV2, file = paste0(cordir, 'all_r12_VV2.RData'))
write.table(cells.allr12.VV2, file = paste0(cordir, 'all_r12_VV2.txt'))

with(pc.scores, plot(PC2.score, PC3.score, pch = 20))
with(pc.scores[cell.name %in% cells.allr12.SVv], points(PC2.score, PC3.score, col = 'green', pch = 20 ))
with(pc.scores[cell.name %in% cells.allr12.VV1], points(PC2.score, PC3.score, col = 'red', pch = 20 ))
with(pc.scores[cell.name %in% cells.allr12.VV2], points(PC2.score, PC3.score, col = 'blue', pch = 20 ))
```



## Define subtypes based on iRPCA populations

```
cells.preartery <- as.character(read.table('all_r7/all_r7_art1.txt')[,1])
cells.cv1 <- as.character(read.table('all_r9/all_r9_CV1.txt')[,1])
cells.cv2 <- as.character(read.table('all_r9/all_r9_CV2.txt')[,1])
cells.svc <- as.character(read.table('SVv_SVc/all_SVv_SVc_SVc.txt')[,1])
cells.endocardial <- as.character(read.table('endocard_refine_r5/cells_endocardRefine.r5.v2.endocard.tx
cells.svv <- as.character(read.table('all_r12/all_r12_SVv.txt')[,1])
cells.mes1 <- as.character(read.table('all_r11/all_r11_Mes1.txt')[,1])
cells.mes2 <- as.character(read.table('all_r11/all_r11_Mes2.txt')[,1])
cells.vv1 <- as.character(read.table('all_r12/all_r12_VV1.txt')[,1])
cells.vv2 <- as.character(read.table('all_r12/all_r12_VV2.txt')[,1])

cell.info[cell.name %in% cells.preartery, subtype := 'Arterial']
cell.info[cell.name %in% cells.cv1, subtype := 'CV1']
cell.info[cell.name %in% cells.cv2, subtype := 'CV2']
cell.info[cell.name %in% cells.svc, subtype := 'SVC']
cell.info[cell.name %in% cells.svv, subtype := 'SVv']
cell.info[cell.name %in% cells.endocardial, subtype := 'Endocardial']
cell.info[cell.name %in% cells.vv1, subtype := 'Venous.Valve.1']
cell.info[cell.name %in% cells.vv2, subtype := 'Venous.Valve.2']
cell.info[cell.name %in% cells.mes1, subtype := 'Mesenchymal.1']
cell.info[cell.name %in% cells.mes2, subtype := 'Mesenchymal.2']

# add colors
tmp1.cell.info <- cell.info[!is.na(subtype)][!subtype %in% c('SVC','Arterial','CV1','CV2')]
tmp1.cell.info[, subtype_color := brewer.pal(8, 'Dark2')[factor(subtype)]]

tmp3.cell.info <- cell.info[!is.na(subtype)][subtype %in% c('SVC','Arterial','CV1','CV2')]
```

```

tmp3.cell.info[subtype=="SVc", subtype_color:=rgb(119, 207, 244, maxColorValue = 255)]
tmp3.cell.info[subtype=="Arterial", subtype_color:=rgb(242, 122, 129, maxColorValue = 255)]
tmp3.cell.info[subtype=="CV1", subtype_color:=rgb(141,113, 252, maxColorValue = 255)]
tmp3.cell.info[subtype=="CV2", subtype_color:=rgb(246,155,235,maxColorValue = 255)]

e12.subtype.info <- rbindlist(list(tmp1.cell.info, tmp3.cell.info), use.names = T, fill = F)
e12.subtype.info[, subtype := factor(subtype, levels=c('Arterial','CV1','CV2','SVc','SVv','Venous.Valve'))]
unique(e12.subtype.info[,.(subtype, subtype_color)])

##           subtype subtype_color
## 1:           SVv      #E7298A
## 2: Mesenchymal.1      #D95F02
## 3: Venous.Valve.2      #E6AB02
## 4: Venous.Valve.1      #66A61E
## 5: Mesenchymal.2      #7570B3
## 6:   Endocardial      #1B9E77
## 7:           CV1      #8D71FC
## 8:           SVc      #77CFF4
## 9:   Arterial      #F27A81
## 10:          CV2      #F69BEB

save(e12.subtype.info, file = 'e12_subtypeInfo.Rdata')

```

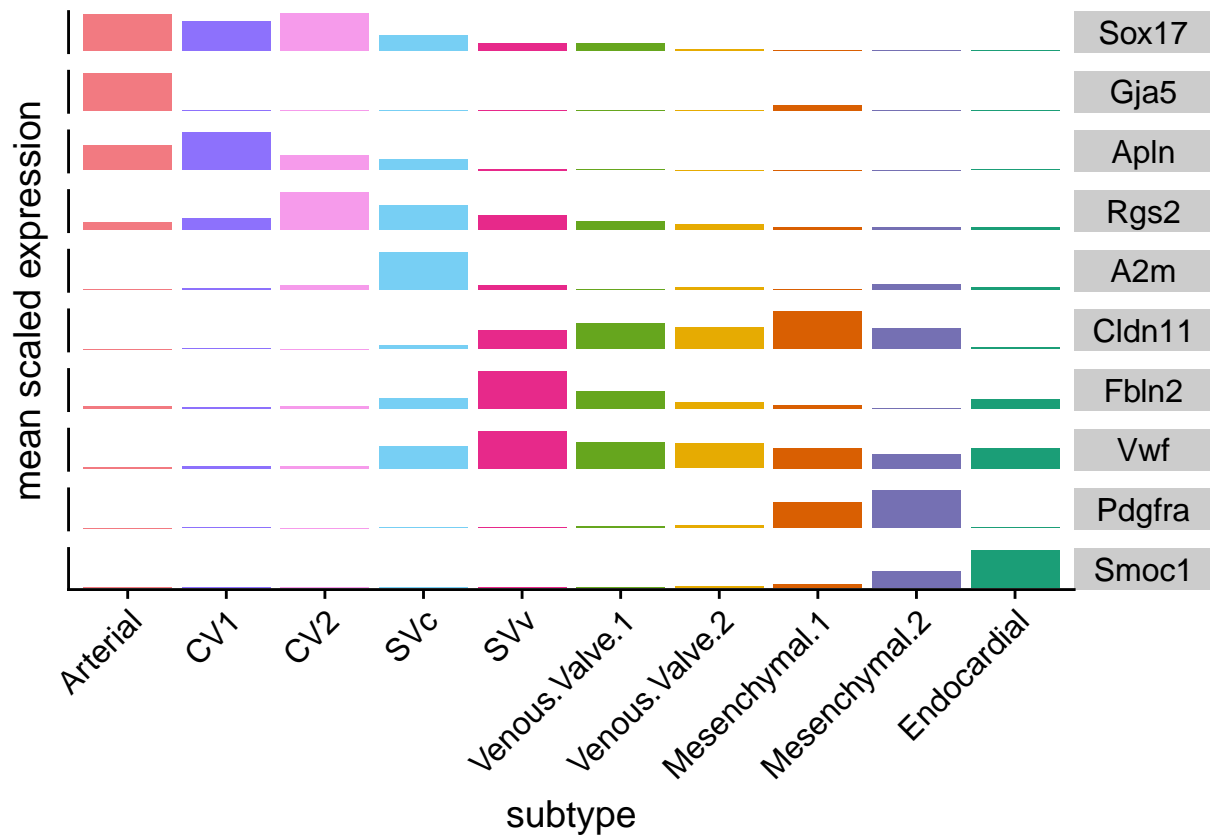
## Expression of validated marker genes across subtypes

```

load('e12_subtypeInfo.Rdata')
genes.plot <- c("Sox17","Gja5","Apln","Rgs2","A2m","Cldn11","Fbln2","Vwf","Pdgfra","Smoc1")
dt.plot <- dt.fv1_NS.good[genes %in% genes.plot]
dt.plot <- merge(dt.plot, e12.subtype.info, by='cell.name')
dt.plot[, gene:=factor(gene, levels=genes.plot)]

gene.means <- dt.plot[, mean(log10.cpm), by=.(gene, subtype, subtype_color)]
gene.means[, V1:=V1/max(V1), by=.(gene)]
ggplot(gene.means, aes(subtype, V1, fill=subtype_color)) +
  geom_bar(stat='identity') +
  scale_fill_identity() +
  facet_grid(gene~.) +
  theme(axis.text.x = element_text(angle=45, hjust=1),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(), strip.text.y = element_text(angle=0)) +
  ylab('mean scaled expression')

```



```
sessionInfo()
```

```
## R version 3.4.3 (2017-11-30)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS High Sierra 10.13.2
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] parallel stats graphics grDevices utils datasets methods
## [8] base
##
## other attached packages:
## [1] Seurat_2.3.0 Rmisc_1.5 lattice_0.20-35
## [4] ggrepel_0.7.0 dplyr_0.7.4 Matrix_1.2-12
## [7] doMC_1.3.5 iterators_1.0.9 foreach_1.4.4
## [10] plyr_1.8.4 RColorBrewer_1.1-2 reshape2_1.4.3
## [13] cowplot_0.9.2 ggplot2_2.2.1 data.table_1.10.4-3
##
## loaded via a namespace (and not attached):
## [1] snow_0.4-2 backports_1.1.2 Hmisc_4.1-1
## [4] VGAM_1.0-5 sn_1.5-1 igraph_1.2.1
## [7] lazyeval_0.2.1 splines_3.4.3 digest_0.6.14
```

## [10] htmltools_0.3.6	lars_1.2	gdata_2.18.0
## [13] magrittr_1.5	checkmate_1.8.5	cluster_2.0.6
## [16] mixtools_1.1.0	ROCR_1.0-7	sfsmisc_1.1-1
## [19] recipes_0.1.2	gower_0.1.2	dimRed_0.1.0
## [22] R.utils_2.6.0	colorspace_1.3-2	bindr_0.1
## [25] survival_2.41-3	zoo_1.8-1	ape_5.0
## [28] glue_1.2.0	DRR_0.0.3	gtable_0.2.0
## [31] ipred_0.9-6	kernlab_0.9-25	ddalpha_1.3.1
## [34] prabclus_2.2-6	DEoptimR_1.0-8	scales_0.5.0
## [37] mvtnorm_1.0-6	Rcpp_0.12.16	metap_0.8
## [40] dtw_1.18-1	htmlTable_1.11.2	tclust_1.3-1
## [43] foreign_0.8-69	proxy_0.4-21	mclust_5.4
## [46] SDMTools_1.1-221	Formula_1.2-2	stats4_3.4.3
## [49] tsne_0.1-3	lava_1.6	prodlim_1.6.1
## [52] htmlwidgets_1.0	FNN_1.1	gplots_3.0.1
## [55] fpc_2.1-11	acepack_1.4.1	modeltools_0.2-21
## [58] ica_1.0-1	pkgconfig_2.0.1	R.methodsS3_1.7.1
## [61] flexmix_2.3-14	nnet_7.3-12	caret_6.0-78
## [64] tidyselect_0.2.4	labeling_0.3	rlang_0.2.0
## [67] munsell_0.4.3	tools_3.4.3	ranger_0.9.0
## [70] broom_0.4.3	ggridges_0.4.1	evaluate_0.10.1
## [73] stringr_1.3.0	yaml_2.1.16	ModelMetrics_1.1.0
## [76] knitr_1.18	fitdistrplus_1.0-9	robustbase_0.92-8
## [79] caTools_1.17.1	purrr_0.2.4	RANN_2.5.1
## [82] bindrcpp_0.2	pbapply_1.3-4	nlme_3.1-131
## [85] formatR_1.5	R.oo_1.21.0	RcppRoll_0.2.2
## [88] compiler_3.4.3	rstudioapi_0.7	png_0.1-7
## [91] tibble_1.4.2	stringi_1.1.7	trimcluster_0.1-2
## [94] psych_1.7.8	diffusionMap_1.1-0	pillar_1.2.1
## [97] lmtest_0.9-35	bitops_1.0-6	irlba_2.3.2
## [100] R6_2.2.2	latticeExtra_0.6-28	KernSmooth_2.23-15
## [103] gridExtra_2.3	codetools_0.2-15	MASS_7.3-48
## [106] gtools_3.5.0	assertthat_0.2.0	CVST_0.2-1
## [109] rprojroot_1.3-2	withr_2.1.1	mnormt_1.5-5
## [112] diptest_0.75-7	doSNOW_1.0.16	grid_3.4.3
## [115] rpart_4.1-12	timeDate_3042.101	tidyr_0.8.0
## [118] class_7.3-14	rmarkdown_1.8	segmented_0.5-3.0
## [121] Rtsne_0.13	numDeriv_2016.8-1	scatterplot3d_0.3-40
## [124] lubridate_1.7.1	base64enc_0.1-3	