Supplementary Information

Supplementary Information for "*Small molecule metabolome identifies potential therapeutic targets against COVID-19.*"

Authors: Sean Bennet, Martin Kaufmann, Calvin Sjaarda, Kaede Takami, Katya Douchant, Emily Moslinger, Henry Wong, D Reed, A Ellis, S Vanner, Robert I. Colautti,#, Prameet M. Sheth

> NOTE: Raw data and fully reproducible code for this project are available on GitHub: http://bit.ly/COVID-Metabolomics (http://bit.ly/COVID-Metabolomics)

Setup

Basic setup for plotting and data handling

```
library(tidyverse) # Tools for data science (graphing, data reorganizing, etc.)
library(ropls)

# Some custom graphing stuff
source("./theme_pub.R")
theme_set(theme_pub())
```

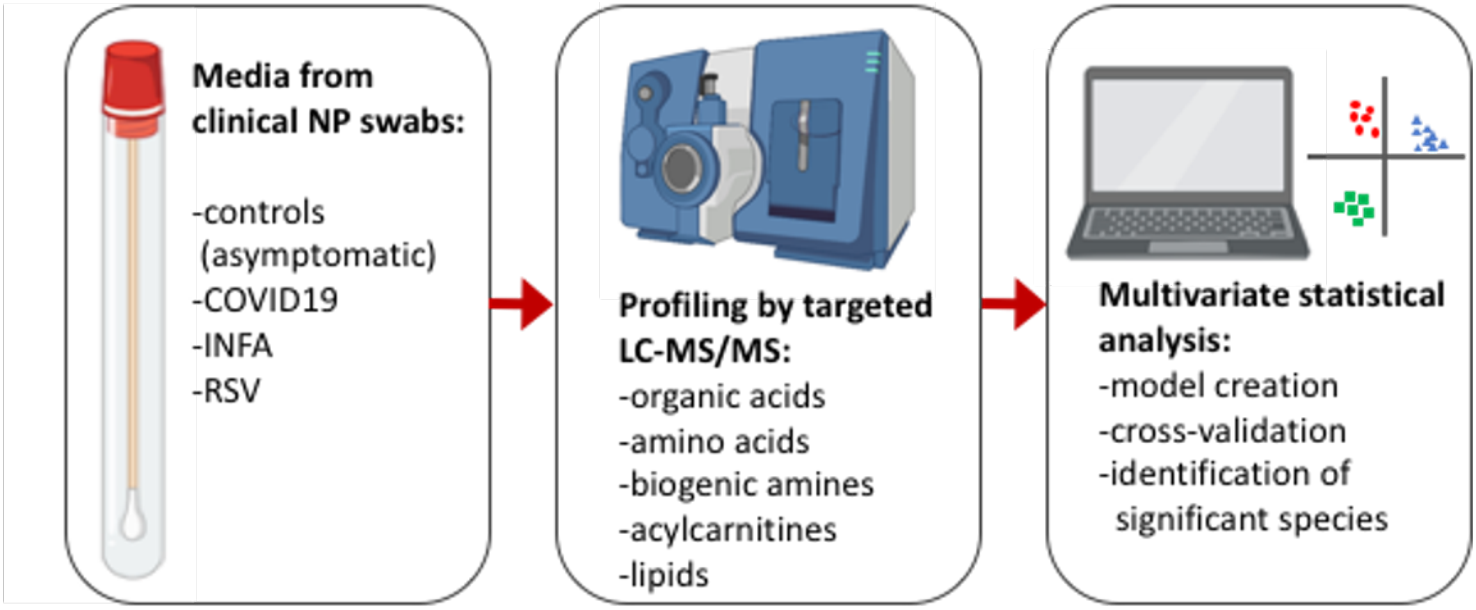
User-parameters

```
flipResp<-T # If True, reverse main axis scaling for respiratory model
flipCOVID<-T # If True, reverse main axis scaling for COVID model
```

Load data

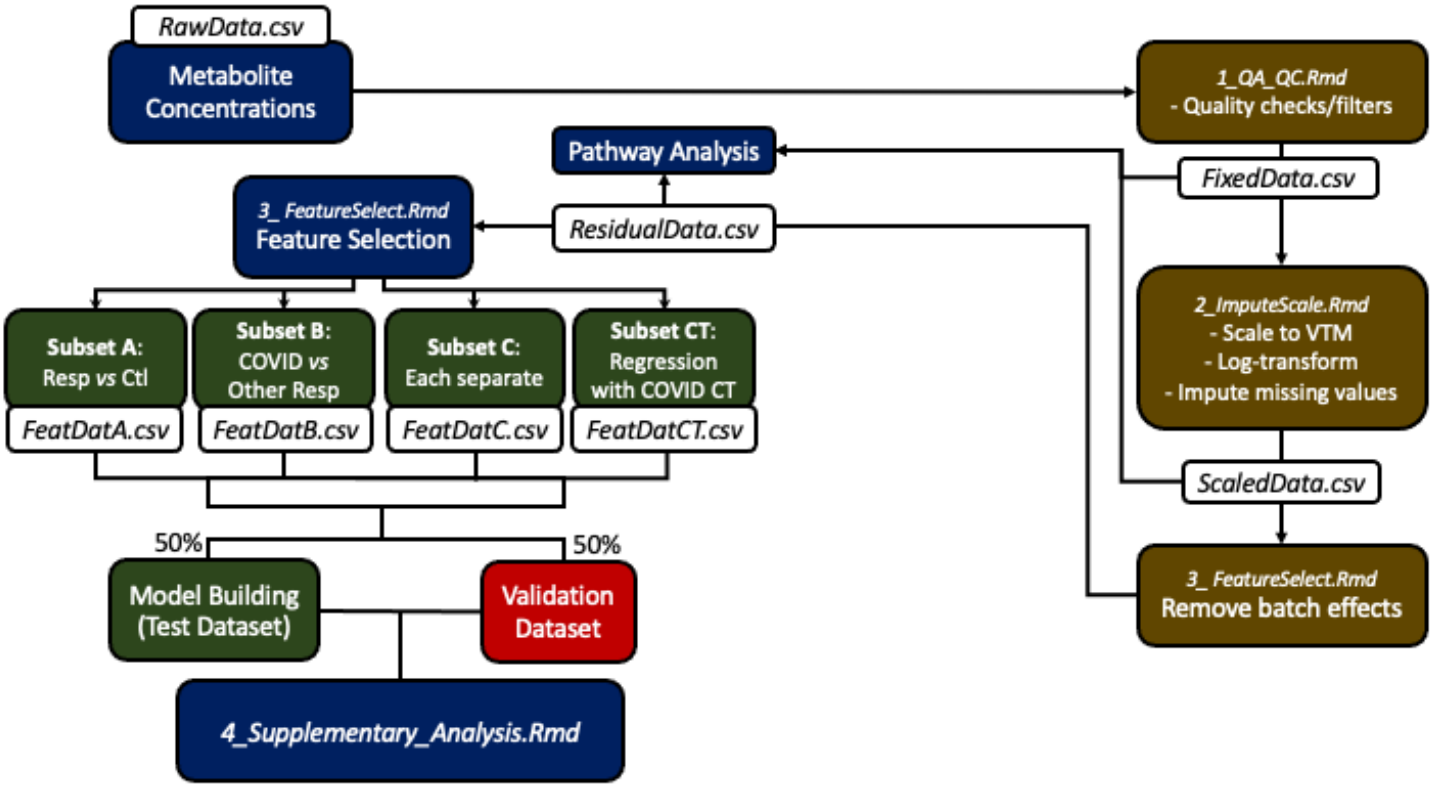
```
featDatA<-read.csv("./data/FeatDatA.csv") # Features selected from resDat using Subse
t A
featDatB<-read.csv("./data/FeatDatB.csv") # Features selected from resDat using Subse
t B
featDatC<-read.csv("./data/FeatDatC.csv") # Features selected from resDat using Subse
t C
featDatCT<-read.csv("./data/FeatDatCT.csv") # Features selected from resDat using Sub
set based on correlation with CT
```

Project Overview



Project Overview

Pipeline Details



Analysis Pipeline

PLS-DA - all groups

Full Partial Least Squares Discriminant Analysis with all 4 groups (and 3 orthogonal predictor axes)

NOTE: This is used for graphing purposes only. For predictive models, see OPLS-DA models, below.

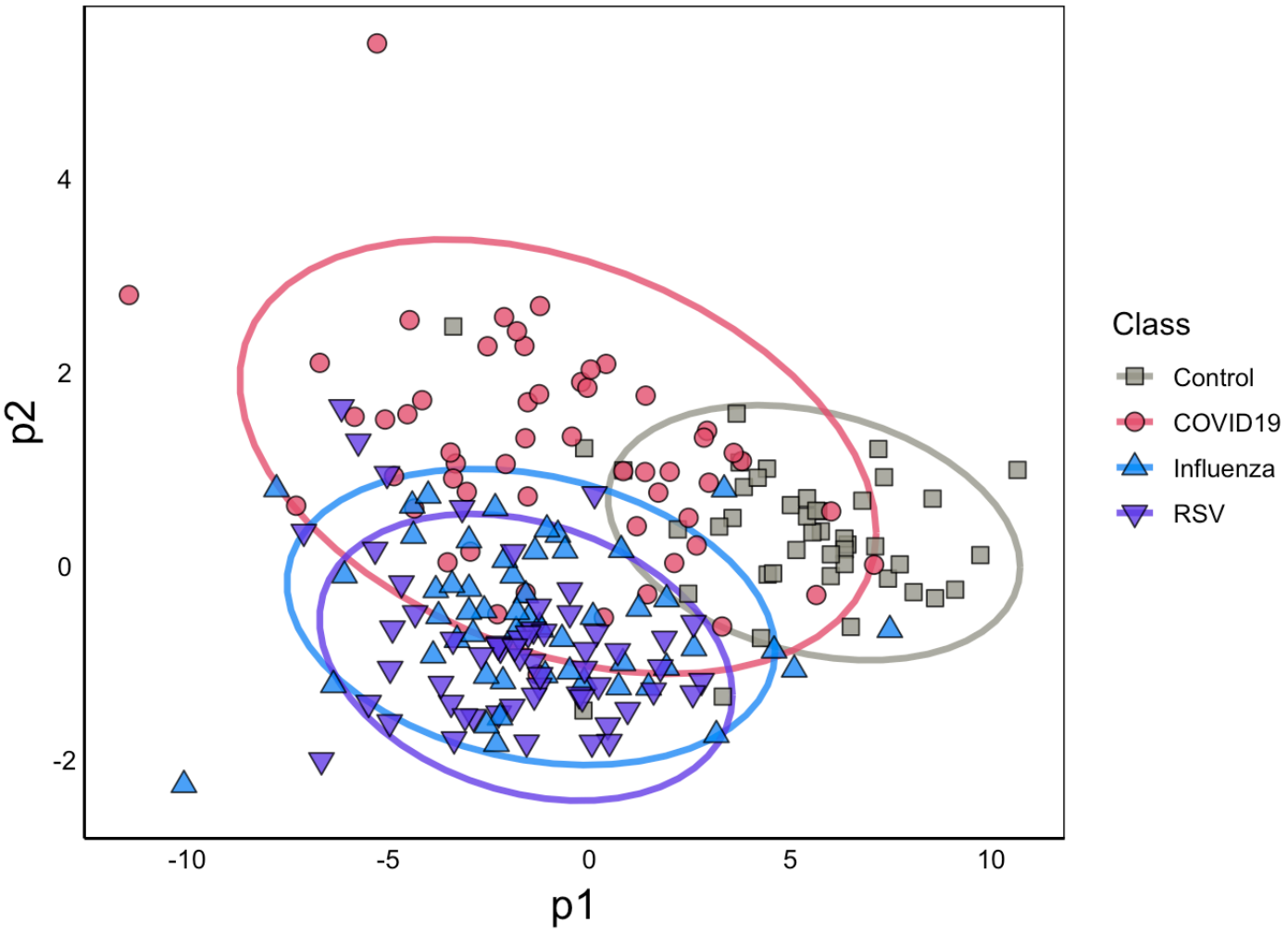
```
# RDA Model
# Setup for grid search with Leave-One-Out Cross-Validation (LOOC using the test-buil
ding subset of data)
FULLdat<-featDatC %>% # Dataset with new encoding
  filter(Class.name %in% c("Control","COVID19","Influenza","RSV")) %>% # Remove VTM
  column_to_rownames("Sample.Name")

DescNames<-c("Batch.Number","Class.name","Sex","Age","CT","OrigClass") # Response Var
iable
Concs<-names(FULLdat)[!names(FULLdat) %in% DescNames] # Predictor Variables


# Organize data for opls
metData<-FULLdat[,Concs] # Metabolite data
patClass<-FULLdat[,"Class.name"] # Predictors
# Set row.names
names(patClass)<-row.names(FULLdat)

# Model of full data for plotting
FULLmod<-opls(metData, patClass, predI=3, fig.pdfC="none")
```

```
## PLS-DA
## 210 samples x 31 variables and 1 response
## standard scaling of predictors and response(s)
##       R2X(cum) R2Y(cum) Q2(cum) RMSEE pre ort pR2Y  pQ2
## Total    0.663    0.395   0.334 0.343   3   0 0.05 0.05
```

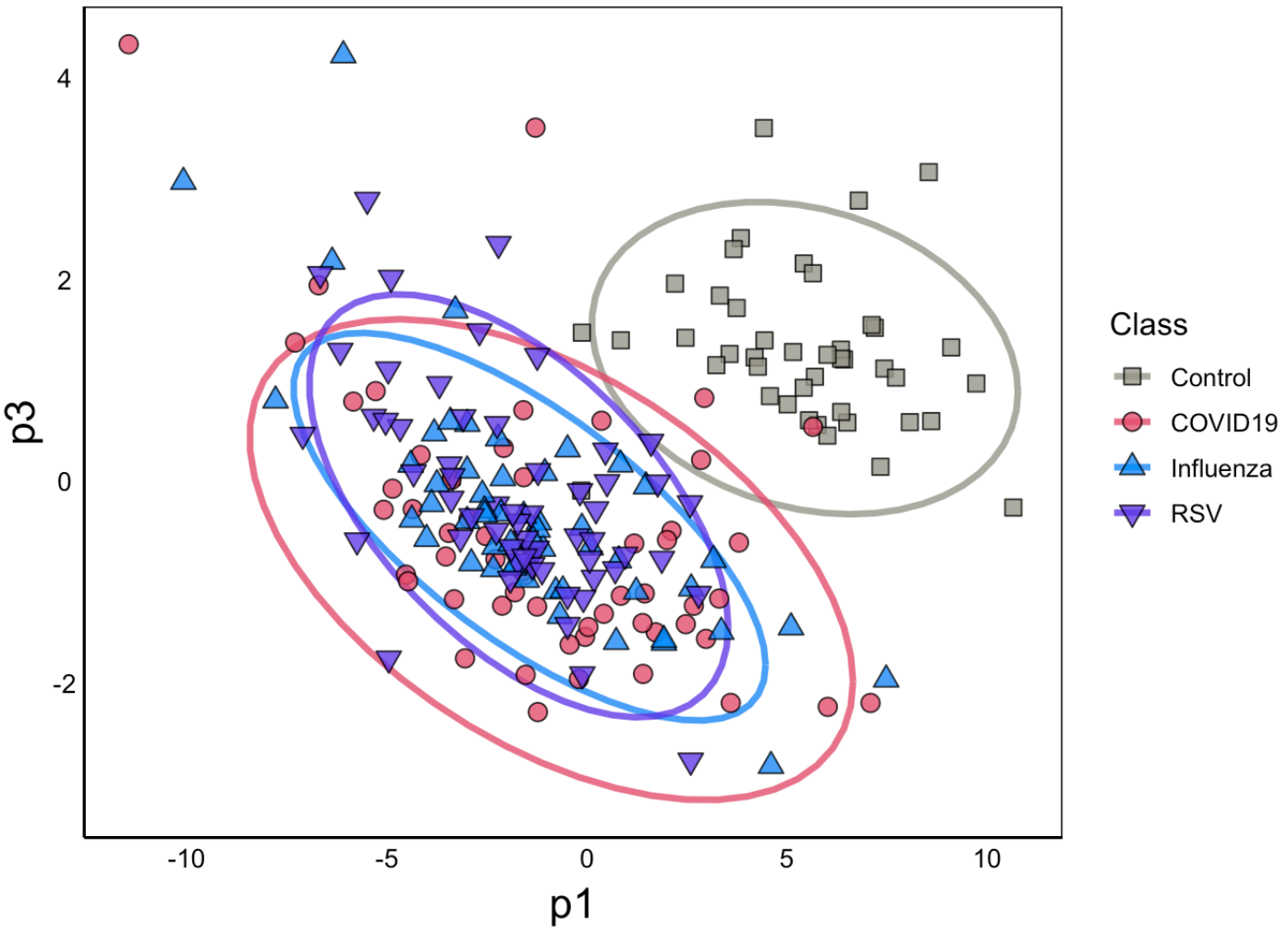
NOTE: No confusion matrix is calculated here (no cross-validation). The purpose is to see whether samples form distinct groups, and factor loadings, rather than to generate and test predictions from the model (that is done below).

PLS-DA axis plots

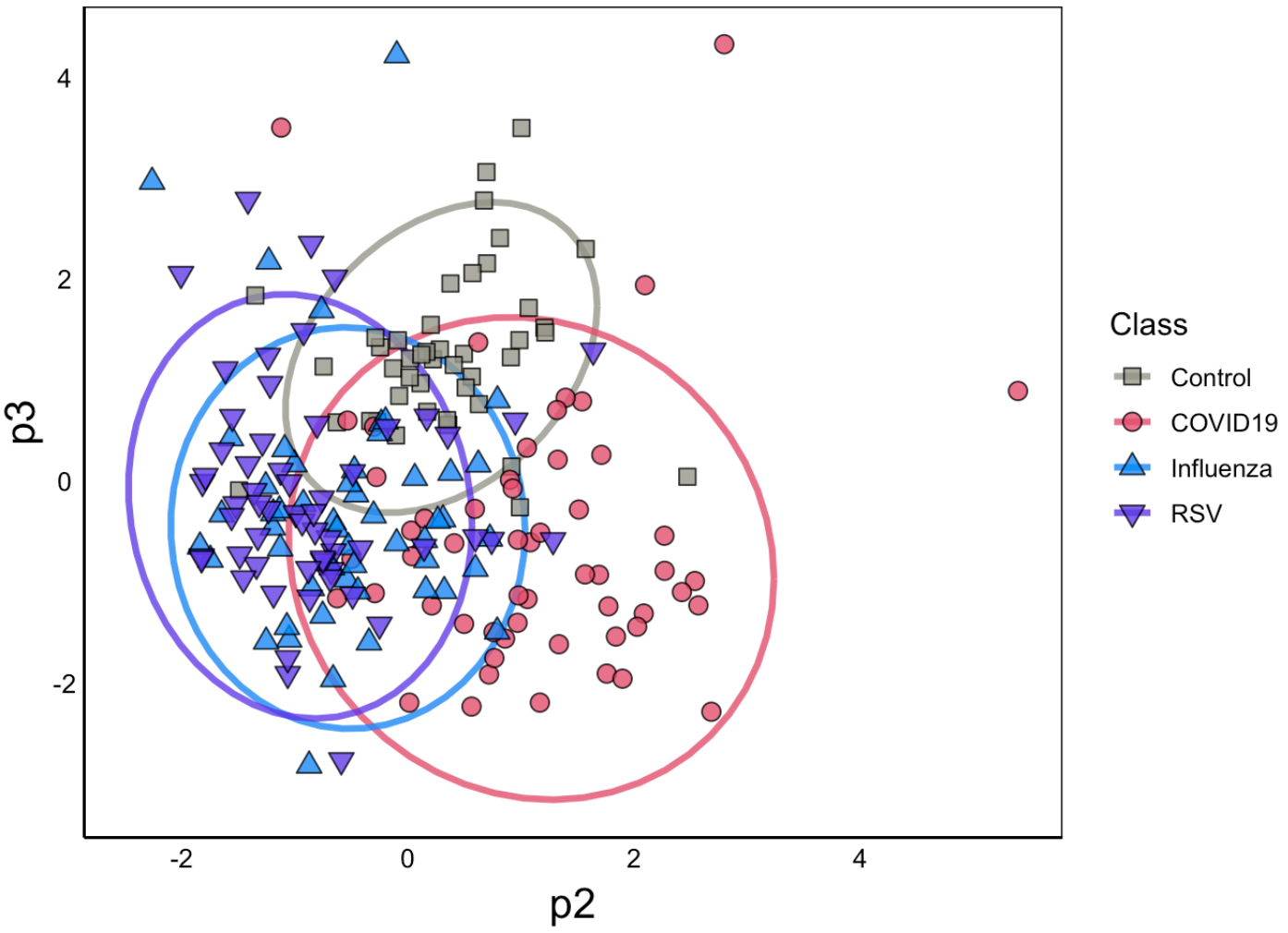
```
pDatF<-as.data.frame(FULLmod@scoreMN)
pDatF$Class<-as.factor(FULLdat$Class.name)
pDatF$Age<-as.factor(FULLdat$Age)
pDatF$Sex<-as.factor(FULLdat$Sex)

ggplot(aes(x=p1,y=p2,group=Class,fill=Class,shape=Class),data=pDatF) +
  stat_ellipse(aes(colour=Class),size=1.2, alpha=0.8) +
  geom_point(size=3,alpha=0.8) +
  scale_fill_manual(values=c("#989788","#E54F6D","#008BF8","#623CEA","#E7EBC5")) +
  scale_colour_manual(values=c("#989788","#E54F6D","#008BF8","#623CEA")) +
  scale_shape_manual(values=c(22,21,24,25,22))
```




```
ggplot(aes(x=p1,y=p3,group=Class,fill=Class,shape=Class),data=pDatF) +
  stat_ellipse(aes(colour=Class),size=1.2, alpha=0.8) +
  geom_point(size=3,alpha=0.8) +
  scale_fill_manual(values=c("#989788","#E54F6D","#008BF8","#623CEA","#E7EBC5")) +
  scale_colour_manual(values=c("#989788","#E54F6D","#008BF8","#623CEA")) +
  scale_shape_manual(values=c(22,21,24,25,22))
```




```
ggplot(aes(x=p2,y=p3,group=Class,fill=Class,shape=Class),data=pDatF) +
  stat_ellipse(aes(colour=Class),size=1.2, alpha=0.8) +
  geom_point(size=3,alpha=0.8) +
  scale_fill_manual(values=c("#989788","#E54F6D","#008BF8","#623CEA","#E7EBC5")) +
  scale_colour_manual(values=c("#989788","#E54F6D","#008BF8","#623CEA")) +
  scale_shape_manual(values=c(22,21,24,25,22))
```

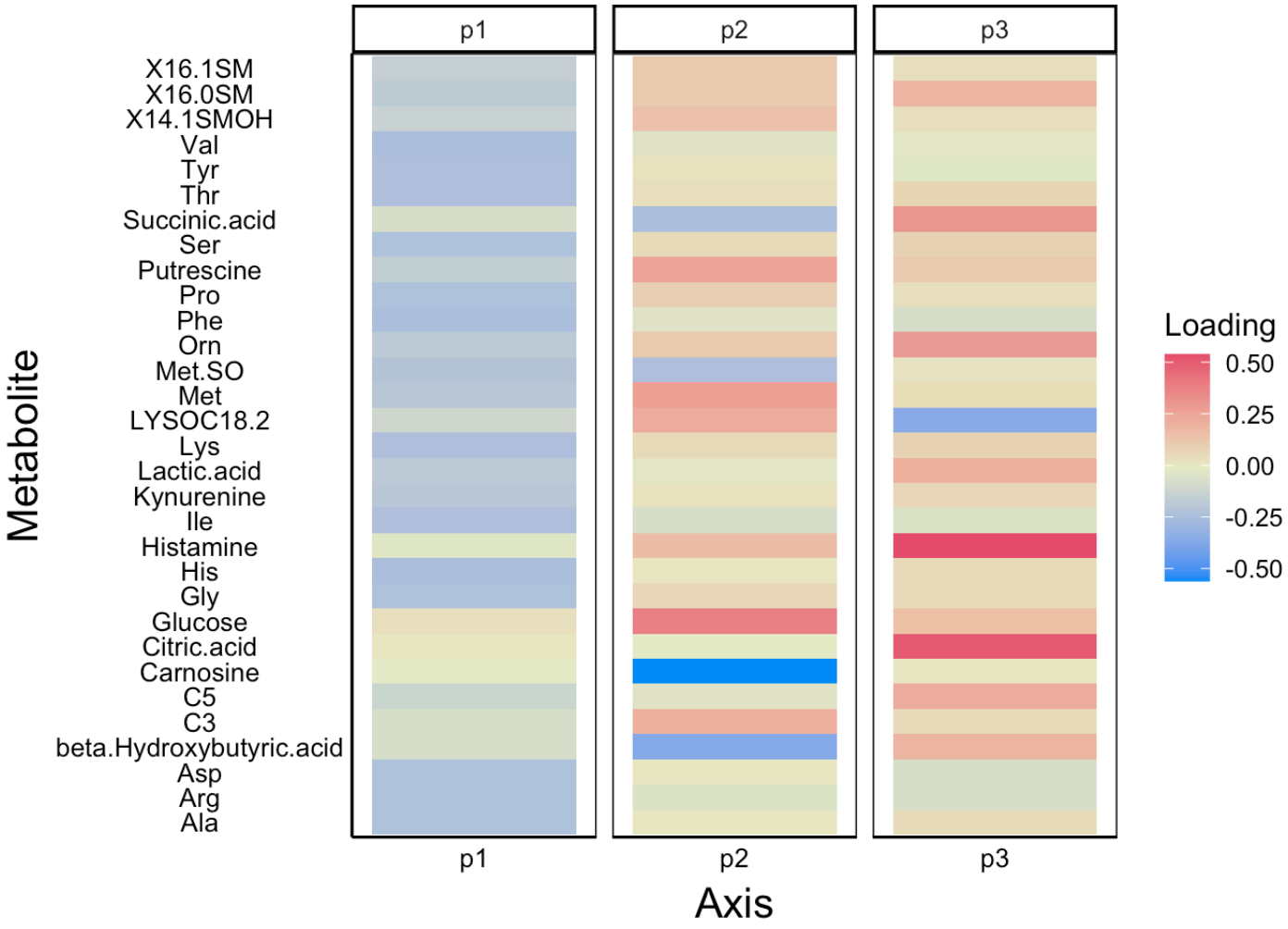


Export PLD-DA plotting data

```
#write.csv(pDatF,"./pDat/FULLdat.csv")
```

Graph of loadings:

```
Loadings<-as.data.frame(FULLmod@loadingMN)
Loadings$Metabolite<-row.names(Loadings)
heatDat<-gather(Loadings,Axis,Loading,all_of(names(Loadings[-4])))
heatDat<-as.data.frame(heatDat)

ggplot(aes(x=Axis,y=Metabolite,fill=Loading),data=heatDat) + geom_tile() +
  facet_grid(~ Axis, scales = "free_x", space = "free_x") +
  scale_fill_gradientn(colours=c("#008BF8","#E7EBC5","#E54F6D"))
```



Export FULL model loadings data

```
#write.csv(heatDat,"./pDat/FULLload.csv")
```

OPLS-DA with FS

Orthogonal PLS used here for models based on two bins. NOTE: the x-axis is the orthogonal predictor, a second (y-axis) is added only for plotting purposes.

Control vs All respiratory

```
# Respiratory Only
RESPdat<-featDatA %>% # Dataset with new encoding
  filter(Class.name %in% c("Control","COVID19","Influenza","RSV")) %>%
  column_to_rownames("Sample.Name")
RESPdat$OrigClass<-RESPdat$Class.name
RESPdat$Class.name<-recode_factor(RESPdat$Class.name, COVID19 = "Resp",
                                  Influenza = "Resp", RSV = "Resp")

DescNames<-c("Batch.Number","Class.name","Sex","Age","CT","OrigClass") # Response Var
iable
Concs<-names(RESPdat)[!names(RESPdat) %in% DescNames] # Predictor Variables

# Organize data for opls
metData<-RESPdat[,Concs] # Metabolite data
patClass<-RESPdat[,"Class.name"] # Predictors
# Set row.names
names(patClass)<-row.names(RESPdat)
# opls model
set.seed(4325)
OPLSMod<-opls(metData, patClass, subset="odd", fig.pdfC="none")
```

```
## Warning: 'permI' set to 0 because train/test partition is selected
```

```
## PLS-DA
## 105 samples x 28 variables and 1 response
## standard scaling of predictors and response(s)
##       R2X(cum) R2Y(cum) Q2(cum) RMSEE RMSEP pre ort
## Total    0.749    0.828   0.787 0.172 0.231   3   0
```

```
trainSet <- getSubsetVi(OPLSMod)

print("Fitted Model")
```

```
## [1] "Fitted Model"
```

```
table(patClass[trainSet],fitted(OPLSMod))
```

```
##
##           Resp Control
##   Resp      82       1
##   Control    0      22
```

```
print("Test Data")
```

```
## [1] "Test Data"
```

```
TestFit<-table(patClass[-trainSet],
      predict(OPLSMod, metData[-trainSet, ]))

TestFit
```

```
##
##           Resp Control
##   Resp      82       1
##   Control    3      19
```

```
TP<-TestFit[1] # True Positive
FP<-sum(TestFit[2])# False Positive
FN<-sum(TestFit[3]) # False Negative
TN<-sum(TestFit)-TP-FP-FN# True Negative

# Model of full data for plotting
pOPLSMod<-opls(metData, patClass, fig.pdfC="none")
```

```
## PLS-DA
## 210 samples x 28 variables and 1 response
## standard scaling of predictors and response(s)
##       R2X(cum) R2Y(cum) Q2(cum) RMSEE pre ort pR2Y  pQ2
## Total    0.722    0.771    0.72 0.197   3   0 0.05 0.05
```

Accuracy

```
(TP+TN)/(sum(TestFit))
```

```
## [1] 0.9619048
```

Sensitivity

```
(TP)/(TP+FN)
```

```
## [1] 0.9879518
```

Specificity

```
TN/(TN+FP)
```

```
## [1] 0.8636364
```

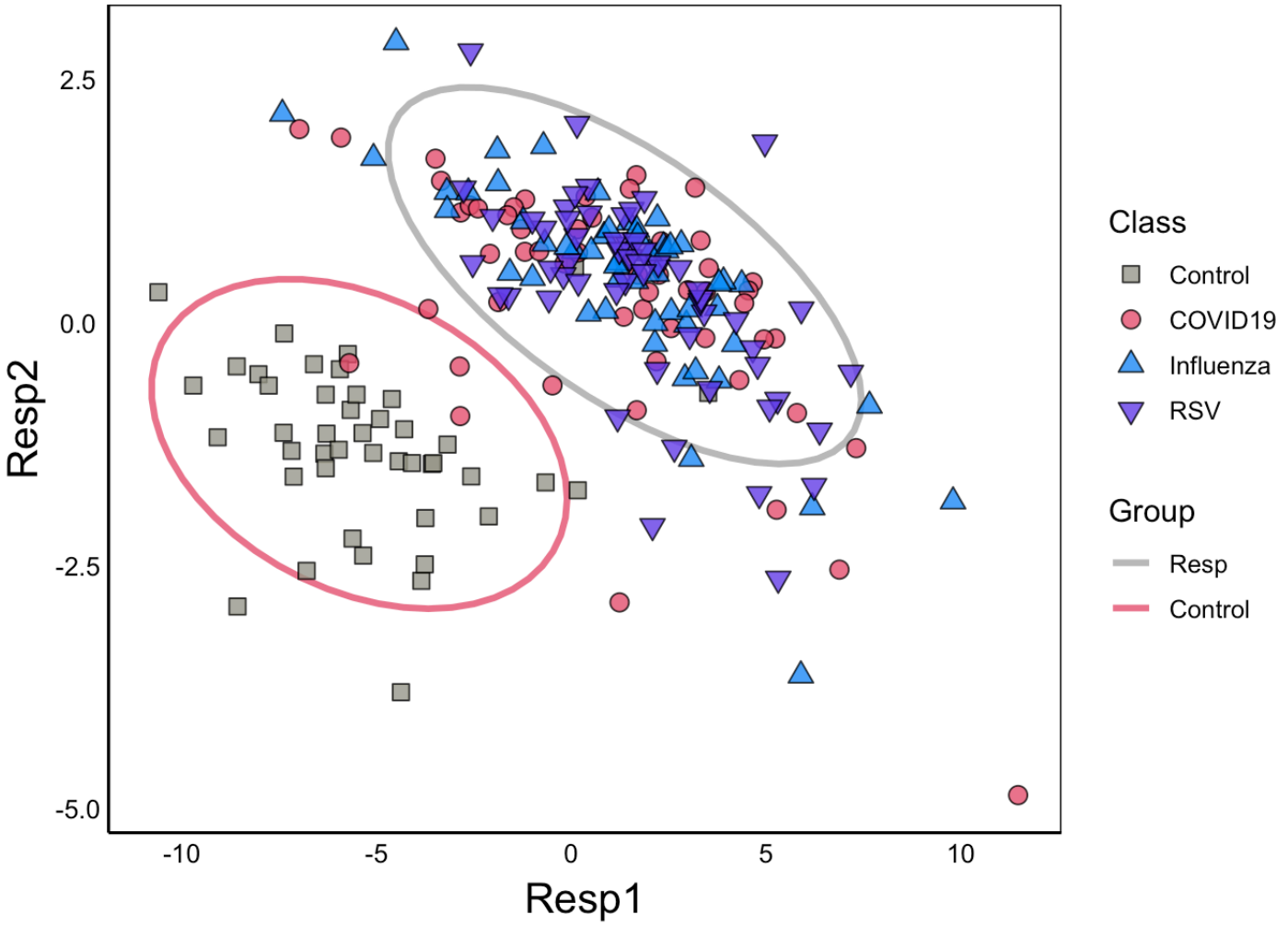
Plot data

NOTE: plot for training data only

```
pDat<-as.data.frame(pOPLSMod@scoreMN)
if(flipResp==T){
  pDat<-pDat*-1
}
pDat$Class<-RESPdat$OrigClass
pDat$Group<-RESPdat$Class.name
pDat$Age<-RESPdat$Age
pDat$Sex<-RESPdat$Sex
names(pDat)<-gsub("p([0-9])","Resp\\1",names(pDat))

#ggplot(aes(x=Full1,y=Full2,group=Class),data=pDat) +
#  geom_point(aes(colour=Class),size=3,alpha=0.7) + scale_colour_brewer(palette = "Set1")

#ggplot(aes(x=Full3,y=Full4,group=Class),data=pDat) +
#  geom_point(aes(colour=Class),size=3,alpha=0.7) + scale_colour_brewer(palette = "Set1")

#ggplot(aes(x=Full1,y=Full5,group=Class),data=pDat) +
#  geom_point(aes(colour=Class),size=3,alpha=0.7) + scale_colour_brewer(palette = "Set1")
```

Plot Control vs All Resp

```
ggplot(aes(x=Resp1,y=Resp2),data=pDat) +
  stat_ellipse(aes(colour=Group),size=1.2, alpha=0.8) +
  geom_point(aes(fill=Class,shape=Class),size=3,alpha=0.8) +
  scale_fill_manual(values=c("#989788","#E54F6D","#008BF8","#623CEA","#E7EBC5")) +
  scale_colour_manual(values=c("grey65","#E54F6D")) +
  scale_shape_manual(values=c(22,21,24,25,22))
```

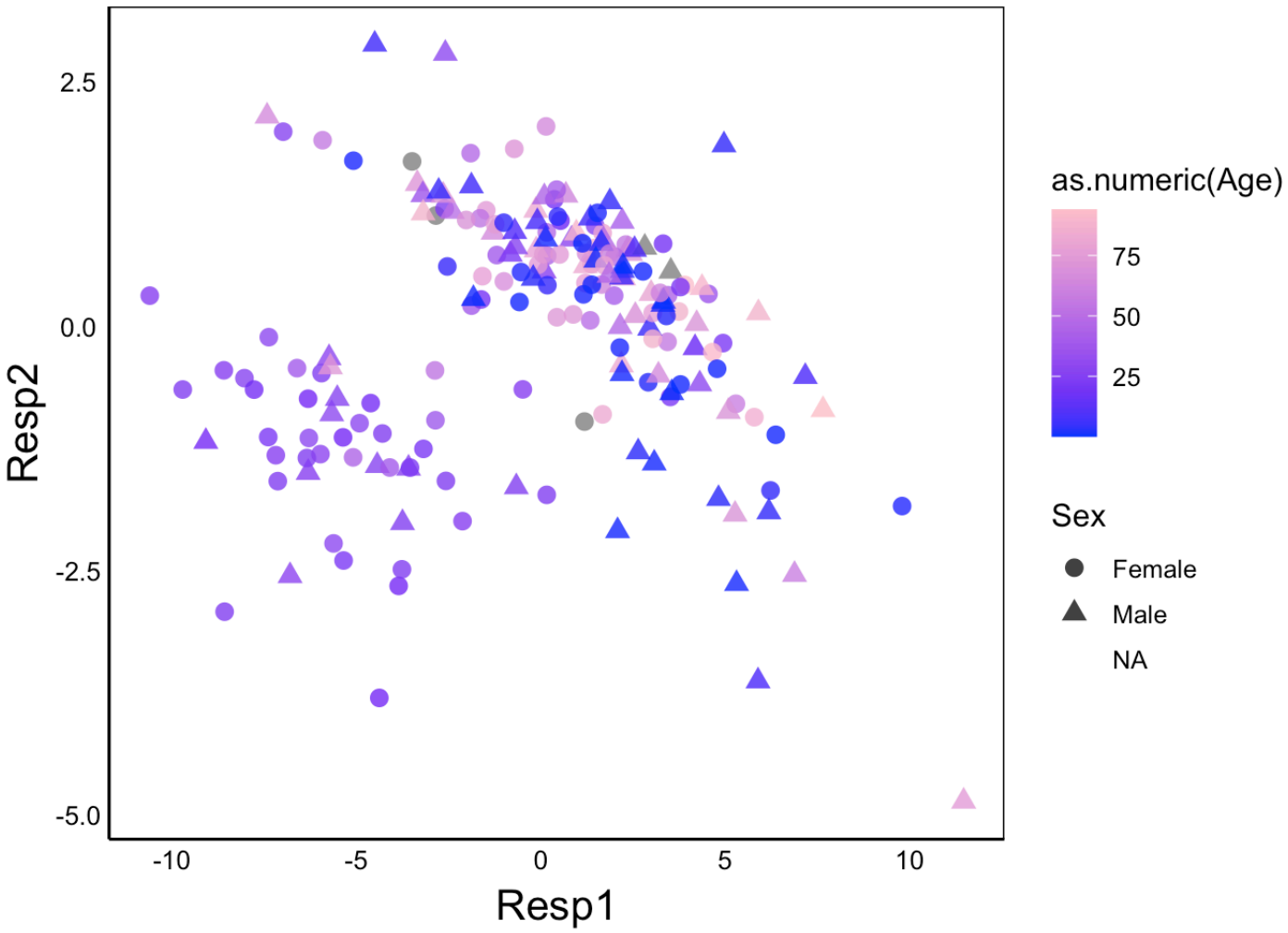


And the same showing age & sex

```
ggplot(aes(x=Resp1,y=Resp2,group=Class),data=pDat) +
geom_point(aes(colour=as.numeric(Age),shape=Sex),size=3,alpha=0.8) +
  scale_colour_gradient(low="blue",high="pink")
```

```
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
```

```
## Warning: Removed 16 rows containing missing values (geom_point).
```



Export OPLS data

```
#write.csv(pDat,"./pDat/RESPdat.csv")
```

COVID vs Other respiratory

```
# Respiratory Only
COVIDdat<-featDatB %>% # Dataset with new encoding
    filter(Class.name %in% c("COVID19","Influenza","RSV")) %>%
    column_to_rownames("Sample.Name")
COVIDdat$OrigClass<-COVIDdat$Class.name
COVIDdat$Class.name<-gsub("Influenza|RSV","Other Resp",COVIDdat$Class.name)

DescNames<-c("Batch.Number","Class.name","Sex","Age","CT","OrigClass") # Response Var
iable
Concs<-names(COVIDdat)[!names(COVIDdat) %in% DescNames] # Predictor Variables

# Organize data for opls
metData<-COVIDdat[,Concs] # Metabolite data
patClass<-COVIDdat[,"Class.name"] # Predictors
# Set row.names
names(patClass)<-row.names(COVIDdat)
# opls model
OPLSMod2<-opls(metData, patClass, predI = 2, subset="odd", fig.pdfC="none")
```

```
## Warning: 'permI' set to 0 because train/test partition is selected
```

```
## PLS-DA
## 84 samples x 5 variables and 1 response
## standard scaling of predictors and response(s)
##       R2X(cum) R2Y(cum) Q2(cum) RMSEE RMSEP pre ort
## Total    0.536    0.442   0.301 0.359  0.38   2   0
```

```
trainSet <- getSubsetVi(OPLSMod2)

print("Fitted Model")
```

```
## [1] "Fitted Model"
```

```
table(patClass[trainSet],fitted(OPLSMod2))
```

```
##
##              COVID19 Other Resp
##   COVID19         21          7
##   Other Resp       6         50
```

```
print("Test Data")
```

```
## [1] "Test Data"
```

```
TestFit<-table(patClass[-trainSet],
      predict(OPLSMod2, metData[-trainSet, ]))

TestFit
```

```
##
##              COVID19 Other Resp
##   COVID19         20          7
##   Other Resp       5         50
```

```
TP<-TestFit[1] # True Positive
FP<-sum(TestFit[2])# False Positive
FN<-sum(TestFit[3]) # False Negative
TN<-sum(TestFit)-TP-FP-FN# True Negative

# Model for plotting full dataset
pOPLSMod2<-opls(metData, patClass,fig.pdfC="none")
```

```
## PLS-DA
## 166 samples x 5 variables and 1 response
## standard scaling of predictors and response(s)
##       R2X(cum) R2Y(cum) Q2(cum) RMSEE pre ort pR2Y  pQ2
## Total    0.566    0.404   0.341 0.367   2   0 0.05 0.05
```

Accuracy

```
(TP+TN)/(sum(TestFit))
```

```
## [1] 0.8536585
```

Sensitivity

```
(TP)/(TP+FN)
```

```
## [1] 0.7407407
```

Specificity

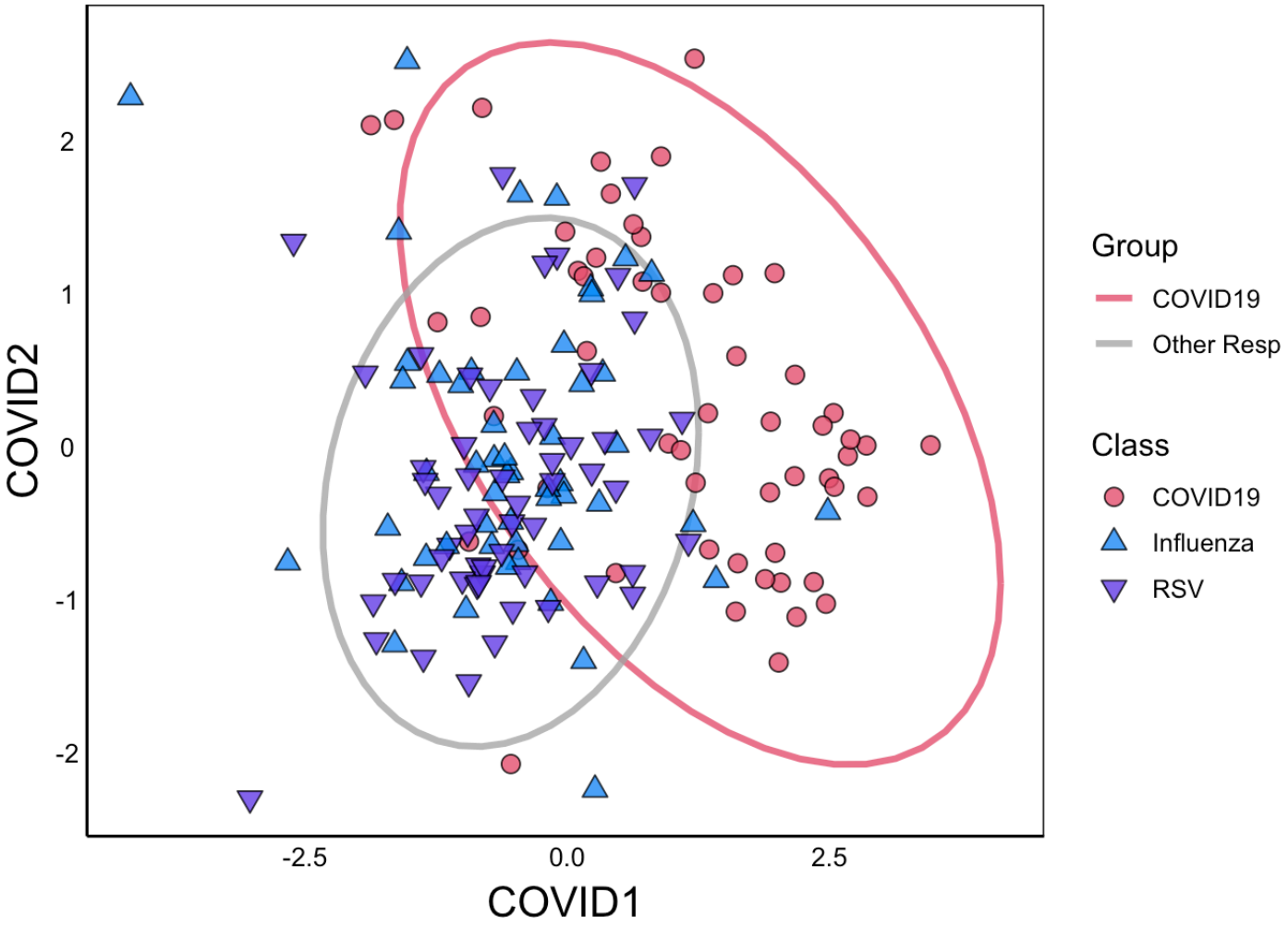
```
TN/(TN+FP)
```

```
## [1] 0.9090909
```

Plot COVID vs other Respiratory

NOTE: plot for training data only

```
pDat2<-as.data.frame(pOPLSMod2@scoreMN)
if(flipCOVID==T){
  pDat2<-pDat2*-1
}
pDat2$Class<-COVIDdat$OrigClass
pDat2$Group<-COVIDdat$Class.name
pDat2$Age<-COVIDdat$Age
pDat2$Sex<-COVIDdat$Sex
names(pDat2)<-gsub("p([0-9])","COVID\\1",names(pDat2))

ggplot(aes(x=COVID1,y=COVID2),data=pDat2) +
  stat_ellipse(aes(colour=Group),size=1.2, alpha=0.8) +
  geom_point(aes(fill=Class,shape=Class),size=3,alpha=0.8) +
  scale_fill_manual(values=c("#E54F6D","#008BF8","#623CEA")) +
  scale_colour_manual(values=c("#E54F6D","grey65")) +
  scale_shape_manual(values=c(21,24,25))
```

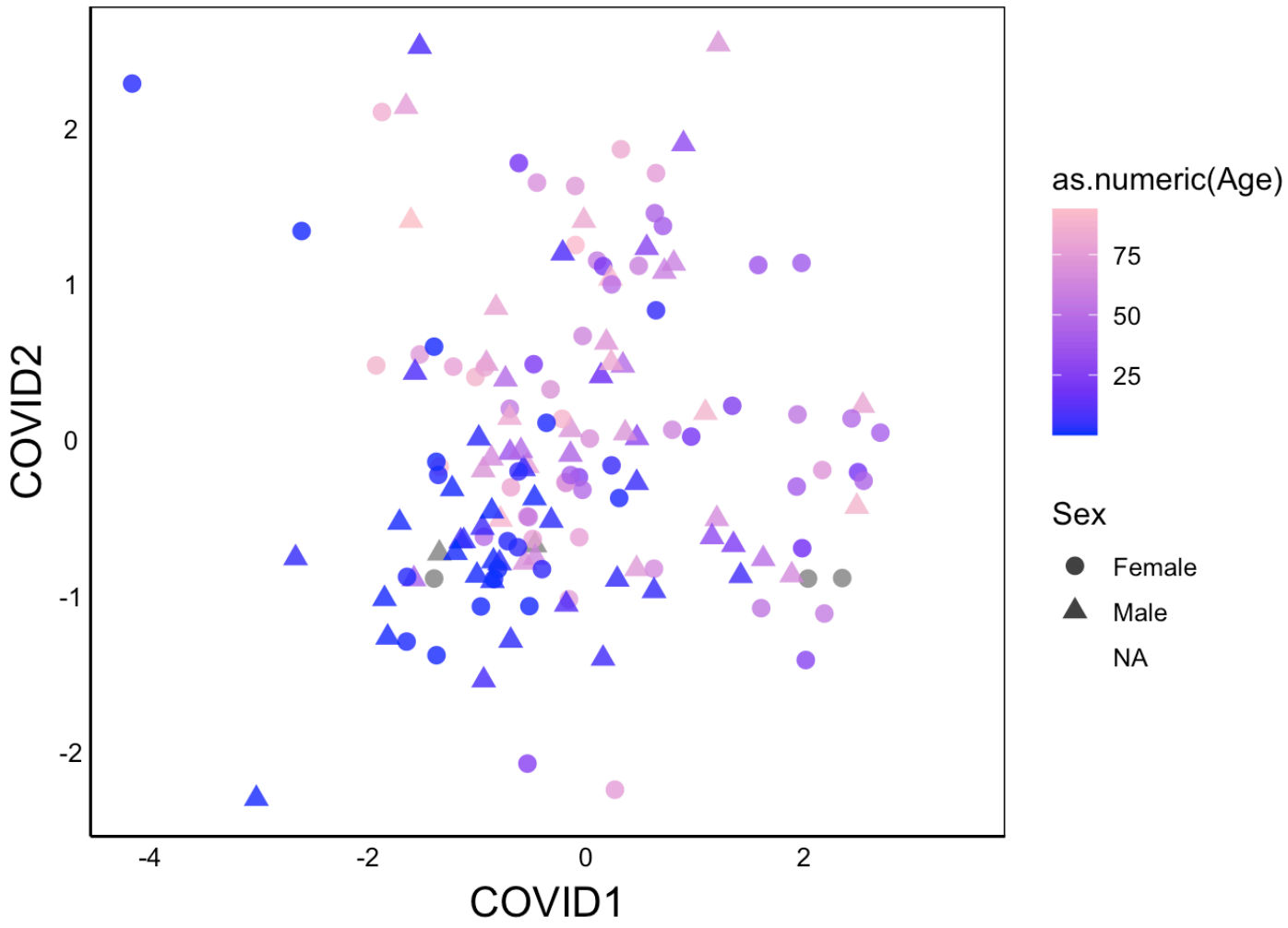


Age & sex

```
ggplot(aes(x=COVID1,y=COVID2,group=Class),data=pDat2) +
geom_point(aes(colour=as.numeric(Age),shape=Sex),size=3,alpha=0.8) +
  scale_colour_gradient(low="blue",high="pink")
```

```
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
```

```
## Warning: Removed 16 rows containing missing values (geom_point).
```



Export OPLS data

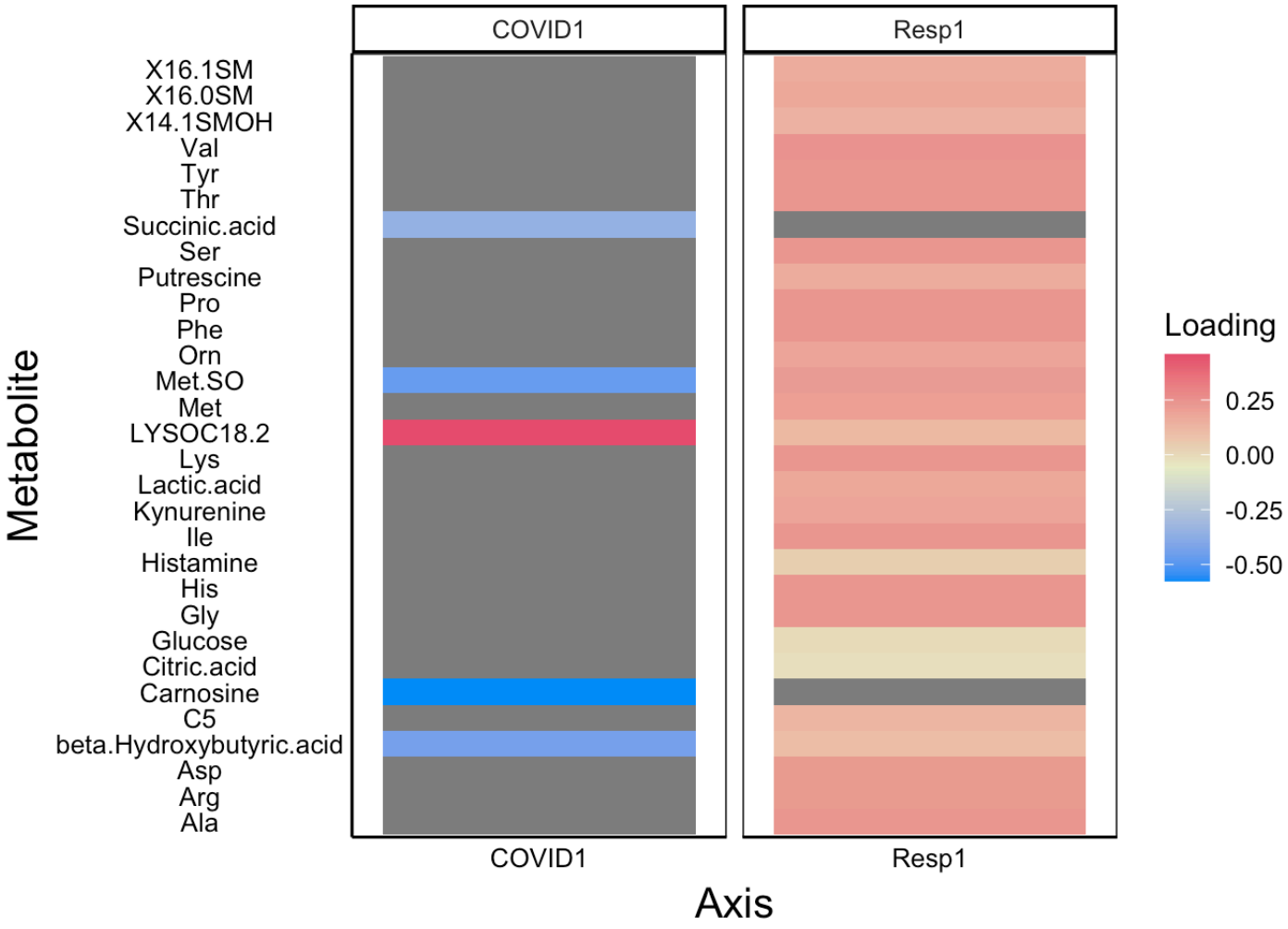
```
#write.csv(pDat2,"./pDat/COVIDdat.csv")
```

Loadings

Loading for both OPLS models

RESP = Control vs all respiratory COVID = COVID vs other respiratory

```
Loadings<-as.data.frame(OPLSMod@loadingMN)
names(Loadings)<-gsub("p","Resp",names(Loadings))
if(flipResp==T){
  Loadings<-Loadings*-1
}
cLoadings<-as.data.frame(OPLSMod2@loadingMN)
names(cLoadings)<-gsub("p","COVID",names(cLoadings))
if(flipCOVID==T){
  cLoadings<-cLoadings*-1
}
heatDat<-full_join(rownames_to_column(Loadings), rownames_to_column(cLoadings), by =
"rowname")
heatDat<-gather(heatDat,Axis,Loading,all_of(names(heatDat)[-1]))
names(heatDat)[1]<-"Metabolite"
heatDat<-as.data.frame(heatDat[heatDat$Axis %in%
                                 c("COVID1","Resp1"), ])

ggplot(aes(x=Axis,y=Metabolite,fill=Loading),data=heatDat) + geom_tile() +
  facet_grid(~ Axis, scales = "free_x", space = "free_x") +
  scale_fill_gradientn(colours=c("#008BF8","#E7EBC5","#E54F6D"))
```

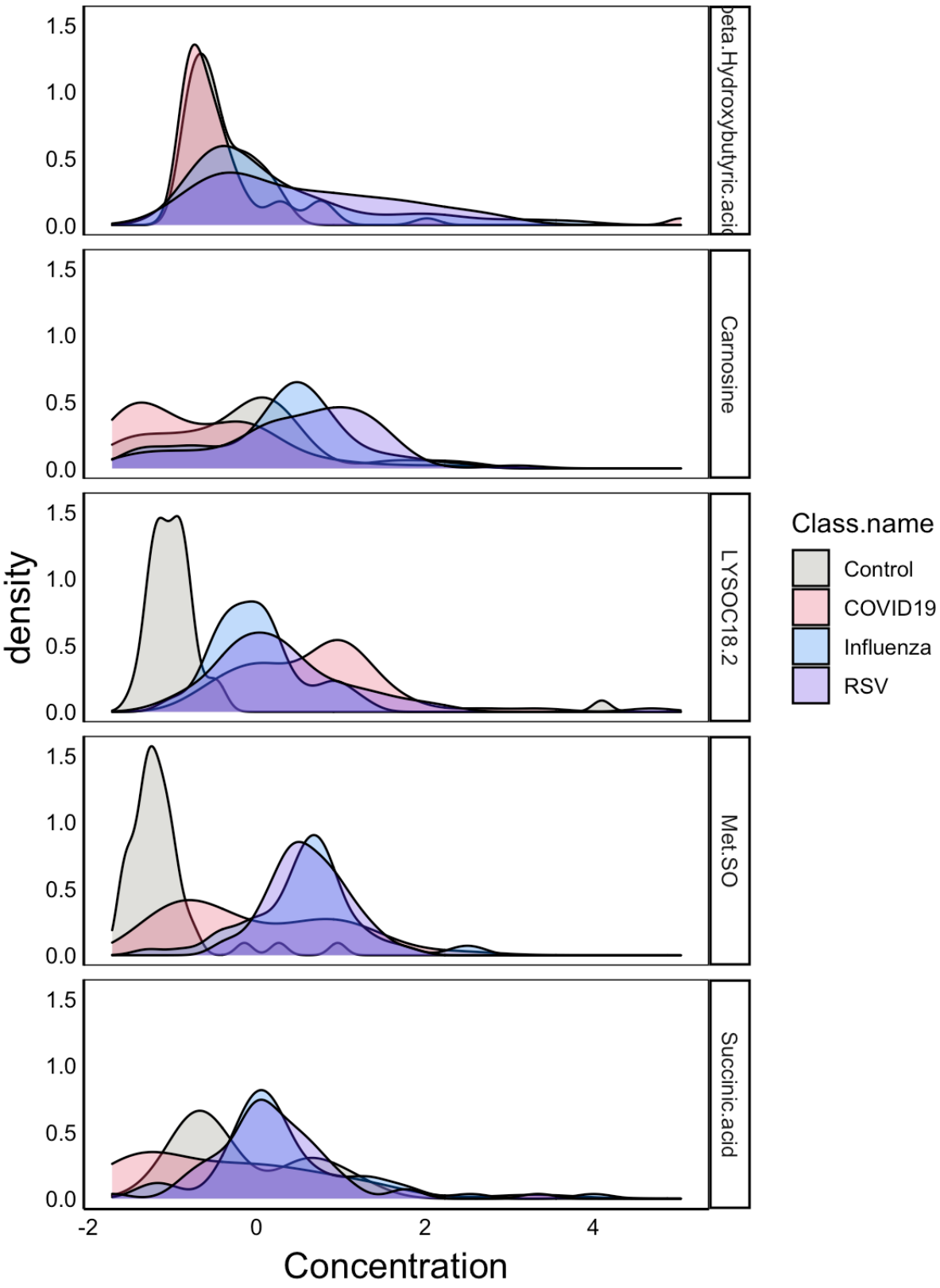


Export OPLS data

```
#write.csv(heatDat,"./pDat/OPLSload.csv")
```

Histogram of significant metabolites

```
pDat3<-gather(FULLdat,Metabolite,Concentration,
              all_of(c("Succinic.acid","Met.SO","LYSOC18.2","Carnosine","beta.Hydroxy
butyric.acid")))
ggplot(aes(x=Concentration,group=Class.name),data=pDat3) +
  geom_density(aes(fill=Class.name),alpha=0.3) + facet_grid(Metabolite ~ .) +
  scale_fill_manual(values=c("#989788","#E54F6D","#008BF8","#623CEA","#E7EBC5"))
```



Export Metabolite Data

```
#write.csv(FULLdat,"./pDat/Metabolites.csv")
```

Other stuff (Exploratory)

CT Correlations

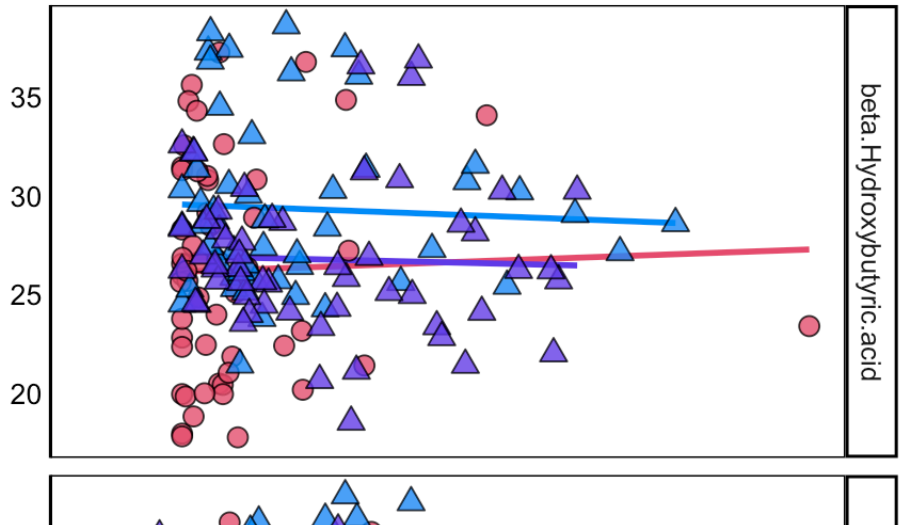
Do the major metabolites from COVID1 correlate with CT value in COVID and other respiratory patients?

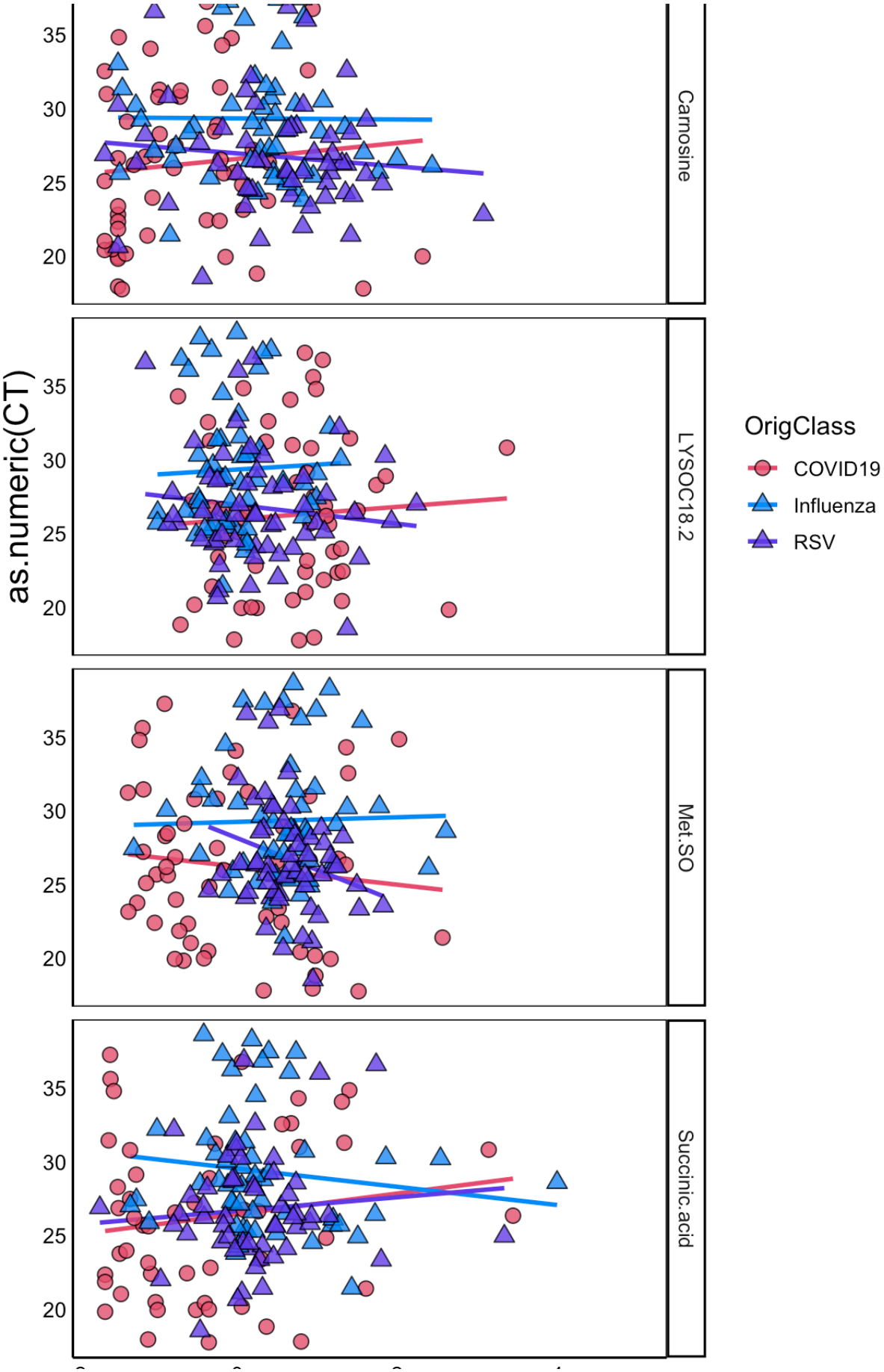
```
COVIDmet<-c("Succinic.acid",
            "Carnosine","Met.SO","LYSOC18.2","beta.Hydroxybutyric.acid")
CTdat<-COVIDdat[,c("OrigClass","Sex","Age","CT",COVIDmet)] %>%
  gather(Metab,Conc,all_of(COVIDmet))

ggplot(aes(x=Conc,y=as.numeric(CT)),data=CTdat) +
  geom_smooth(aes(group=OrigClass,colour=OrigClass),se=F,method="lm") +
  geom_point(aes(group=OrigClass,fill=OrigClass,shape=OrigClass),
             size=3,alpha=0.8) +
  scale_fill_manual(values=c("#E54F6D","#008BF8","#623CEA")) +
  scale_colour_manual(values=c("#E54F6D","#008BF8","#623CEA")) +
  scale_shape_manual(values=c(21,24,24)) +
  facet_grid(Metab~.,scales="free")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 25 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 25 rows containing missing values (geom_point).
```




as.numeric(CT)
35
30
25
20
Carnosine
LYSOC18.2
Met.SO
Succinic.acid
OrigClass
COVID19
Influenza
RSV

Statistical tests

```
for(Met in unique(CTdat$Metab)){
  for(Cls in unique(CTdat$OrigClass)){
    StatDat<-CTdat[CTdat$Metab == Met &
                     CTdat$OrigClass == Cls,]
    print(anova(lm(Conc~as.numeric(CT), data=StatDat)))
    StatDat<-NA
  }
}
```

```
## Analysis of Variance Table
##
## Response: Conc
##                Df Sum Sq Mean Sq F value Pr(>F)
## as.numeric(CT)  1  1.857   1.857  1.2914  0.261
## Residuals      52 74.775   1.438
## Analysis of Variance Table
##
## Response: Conc
##                Df Sum Sq Mean Sq F value Pr(>F)
## as.numeric(CT)  1  0.748 0.74751  0.8974  0.348
## Residuals      50 41.649 0.83298
## Analysis of Variance Table
##
## Response: Conc
##                Df  Sum Sq Mean Sq F value Pr(>F)
## as.numeric(CT)  1  0.2763 0.27626  0.4752 0.4936
## Residuals      53 30.8101 0.58132
## Analysis of Variance Table
##
## Response: Conc
##                Df Sum Sq Mean Sq F value Pr(>F)
## as.numeric(CT)  1  0.325 0.32495  0.4108 0.5244
## Residuals      52 41.133 0.79101
## Analysis of Variance Table
##
## Response: Conc
##                Df Sum Sq Mean Sq F value Pr(>F)
## as.numeric(CT)  1  0.002 0.00228  0.0029 0.9572
## Residuals      50 39.210 0.78421
## Analysis of Variance Table
```

```
##
## Response: Conc
##                Df Sum Sq Mean Sq F value Pr(>F)
## as.numeric(CT)  1  0.612 0.61189  0.6768 0.4144
## Residuals      53 47.918 0.90411
## Analysis of Variance Table
##
## Response: Conc
##                Df Sum Sq Mean Sq F value Pr(>F)
## as.numeric(CT)  1  0.648 0.64801  0.6642 0.4188
## Residuals      52 50.733 0.97564
## Analysis of Variance Table
##
## Response: Conc
##                Df  Sum Sq Mean Sq F value Pr(>F)
## as.numeric(CT)  1  0.0166 0.01662  0.0333 0.8559
## Residuals      50 24.9443 0.49889
## Analysis of Variance Table
##
## Response: Conc
##                Df Sum Sq Mean Sq F value  Pr(>F)
## as.numeric(CT)  1 0.6493 0.64928  3.5978 0.06331 .
## Residuals      53 9.5646 0.18046
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Analysis of Variance Table
##
## Response: Conc
##                Df Sum Sq Mean Sq F value Pr(>F)
## as.numeric(CT)  1  0.137 0.13702  0.2065 0.6515
## Residuals      52 34.512 0.66368
## Analysis of Variance Table
##
## Response: Conc
##                Df  Sum Sq  Mean Sq F value Pr(>F)
## as.numeric(CT)  1  0.0228 0.022824  0.0829 0.7746
## Residuals      50 13.7639 0.275278
## Analysis of Variance Table
##
## Response: Conc
##                Df  Sum Sq Mean Sq F value Pr(>F)
## as.numeric(CT)  1  0.4731 0.47306  0.8661 0.3562
## Residuals      53 28.9470 0.54617
## Analysis of Variance Table
##
## Response: Conc
##                Df Sum Sq Mean Sq F value Pr(>F)
```

```
## as.numeric(CT)  1  0.049 0.04939  0.0569 0.8124
## Residuals      52 45.145 0.86817
## Analysis of Variance Table
##
## Response: Conc
##                Df Sum Sq Mean Sq F value Pr(>F)
## as.numeric(CT)  1  0.173  0.1728  0.1397 0.7102
## Residuals      50 61.866  1.2373
## Analysis of Variance Table
##
## Response: Conc
##                Df Sum Sq Mean Sq F value Pr(>F)
## as.numeric(CT)  1  0.086 0.08638  0.0729 0.7883
## Residuals      53 62.828 1.18544
```

Stats summary: Influenze has significantly higher CT count overall, but no effect of

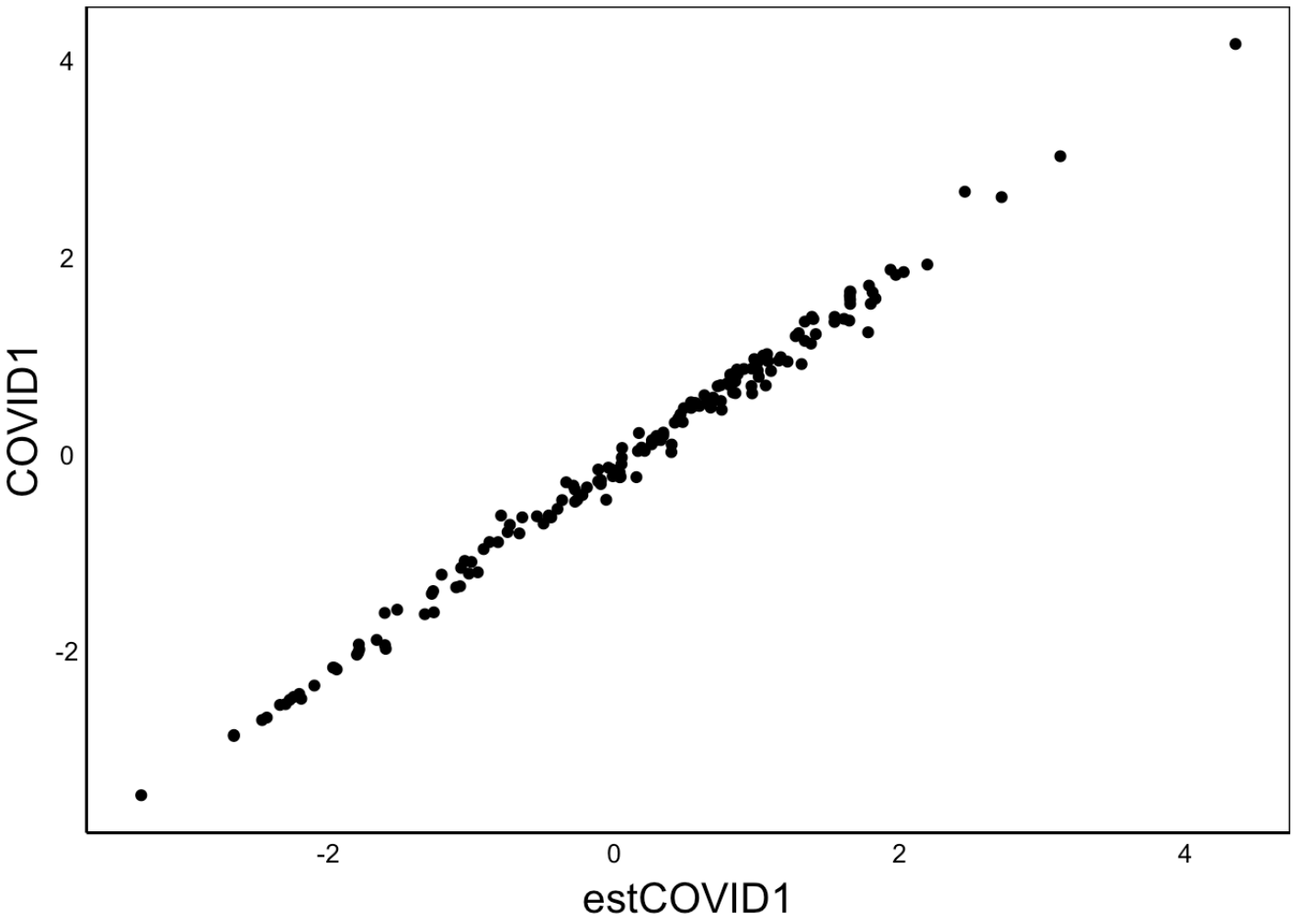
What about COVID1 axis from OPLS-DA model – does it predict CT values?

Setup:

```
COVIDmet<-c("Carnosine","Met.SO","beta.Hydroxybutyric.acid",
            "LYSOC18.2","Succinic.acid")
CTcomp<-COVIDdat[,COVIDmet]

CTcomp$estCOVID1<-rowSums(t(OPLSMod2@loadingMN[,1]*t(CTcomp)))

CTcomp$Sample<-rownames(CTcomp)
CTcomp$CT<-COVIDdat$CT
CTcomp$OrigClass<-COVIDdat$OrigClass
pDat2$Sample<-rownames(pDat2)

if(flipCOVID==T){
  pDat2$COVID1<-pDat2$COVID1*-1
}

pDat3<-full_join(pDat2[,c("Sample","COVID1")],
                      CTcomp[,c("Sample","estCOVID1","OrigClass","CT")],by="Sample")
```

Double-check proper calculation of COVID1 in full dataset

```
qplot(x=estCOVID1,y=COVID1,data=pDat3)
```



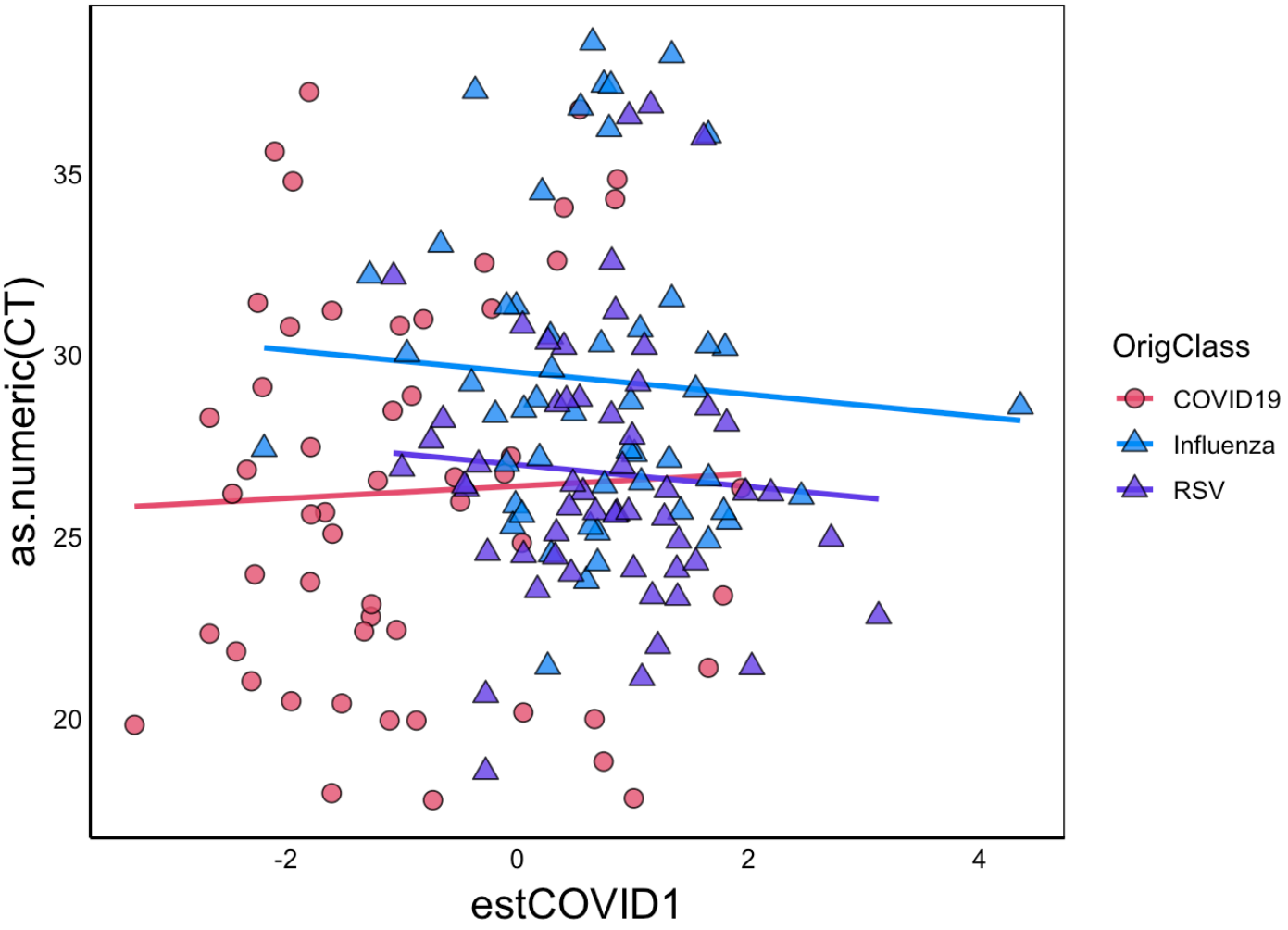
Test for CT correlation

```
ggplot(aes(x=estCOVID1,y=as.numeric(CT)),data=pDat3) +
  geom_smooth(aes(group=OrigClass,colour=OrigClass),se=F,method="lm") +
  geom_point(aes(group=OrigClass,fill=OrigClass,shape=OrigClass),
             size=3,alpha=0.8) +
  scale_fill_manual(values=c("#E54F6D","#008BF8","#623CEA")) +
  scale_colour_manual(values=c("#E54F6D","#008BF8","#623CEA")) +
  scale_shape_manual(values=c(21,24,24))
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 5 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```


as.numeric(CT)
35
30
25
20
-2
0
2
4
estCOVID1
OrigClass
COVID19
Influenza
RSV

NOPE