Xeva Tutorial

Arvind Mer

2017-01-18

Load Xeva

Load Xeva library and data.

```
library(Xeva)
data(lpdx)
head(modelInfo(lpdx))
```

```
##                                  model.id                      donor
## PHLC1106_P5.501.A1.1 PHLC1106_P5.501.A1.1 11101S-213RC-312S(F)-412S-
## PHLC1106_P5.504.A4.1 PHLC1106_P5.504.A4.1 11101S-213RC-312S(F)-412S-
## PHLC1106_P5.506.B1.1 PHLC1106_P5.506.B1.1 11101S-213RC-312S(F)-412S-
## PHLC1106_P5.507.B2.1 PHLC1106_P5.507.B2.1 11101S-213RC-312S(F)-412S-
## PHLC1106_P5.508.B3.1 PHLC1106_P5.508.B3.1 11101S-213RC-312S(F)-412S-
## PHLC1106_P5.511.C1.1 PHLC1106_P5.511.C1.1 11101S-213RC-312S(F)-412S-
##                           dob sex     PHLC biobase.id patient.id
## PHLC1106_P5.501.A1.1 Aug31.14   F PHLC1106   PHLC1106   PHLC1106
## PHLC1106_P5.504.A4.1 Aug31.14   F PHLC1106   PHLC1106   PHLC1106
## PHLC1106_P5.506.B1.1 Aug31.14   F PHLC1106   PHLC1106   PHLC1106
## PHLC1106_P5.507.B2.1 Aug31.14   F PHLC1106   PHLC1106   PHLC1106
## PHLC1106_P5.508.B3.1 Aug31.14   F PHLC1106   PHLC1106   PHLC1106
## PHLC1106_P5.511.C1.1 Sep14.14   F PHLC1106   PHLC1106   PHLC1106
```

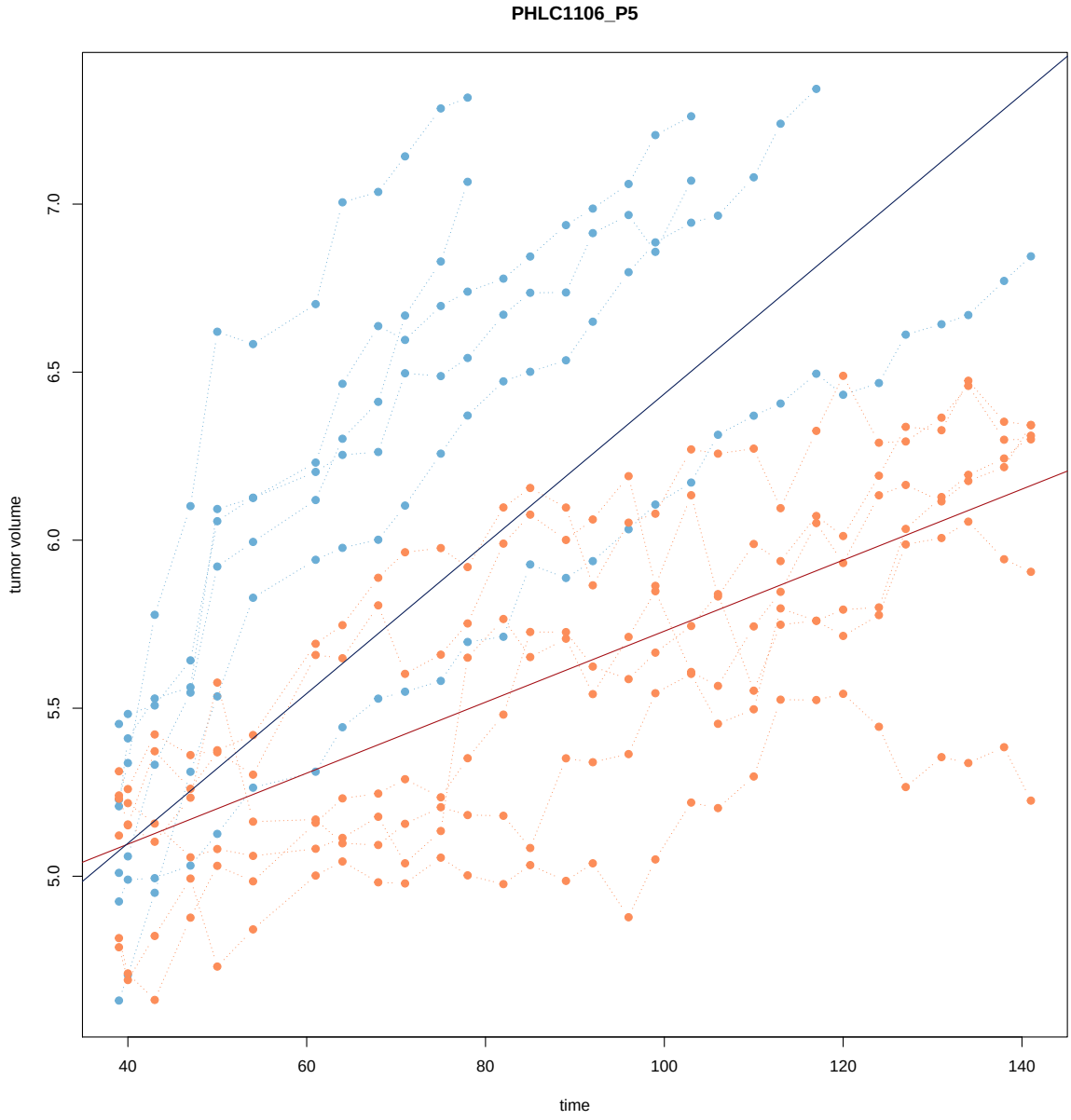
Models which belongs to same batch are in one list which is stored in expDesign slot. For example

```
print(batchNames(lpdx))
```

```
##   PHLC1106_P5    PHLC111_P7    PHLC119_P5    PHLC153_P6    PHLC181_P7
## "PHLC1106_P5" "PHLC111_P7" "PHLC119_P5" "PHLC153_P6" "PHLC181_P7"
##    PHLC189_P5    PHLC191_P5    PHLC191_P7    PHLC196_P5    PHLC215_P5
##  "PHLC189_P5" "PHLC191_P5" "PHLC191_P7" "PHLC196_P5" "PHLC215_P5"
##    PHLC229_P6    PHLC235_P4    PHLC655_P7     PHLC82_P5
##  "PHLC229_P6" "PHLC235_P4" "PHLC655_P7"   "PHLC82_P5"
```

To calculate angle between the treatment and control samples of this batch

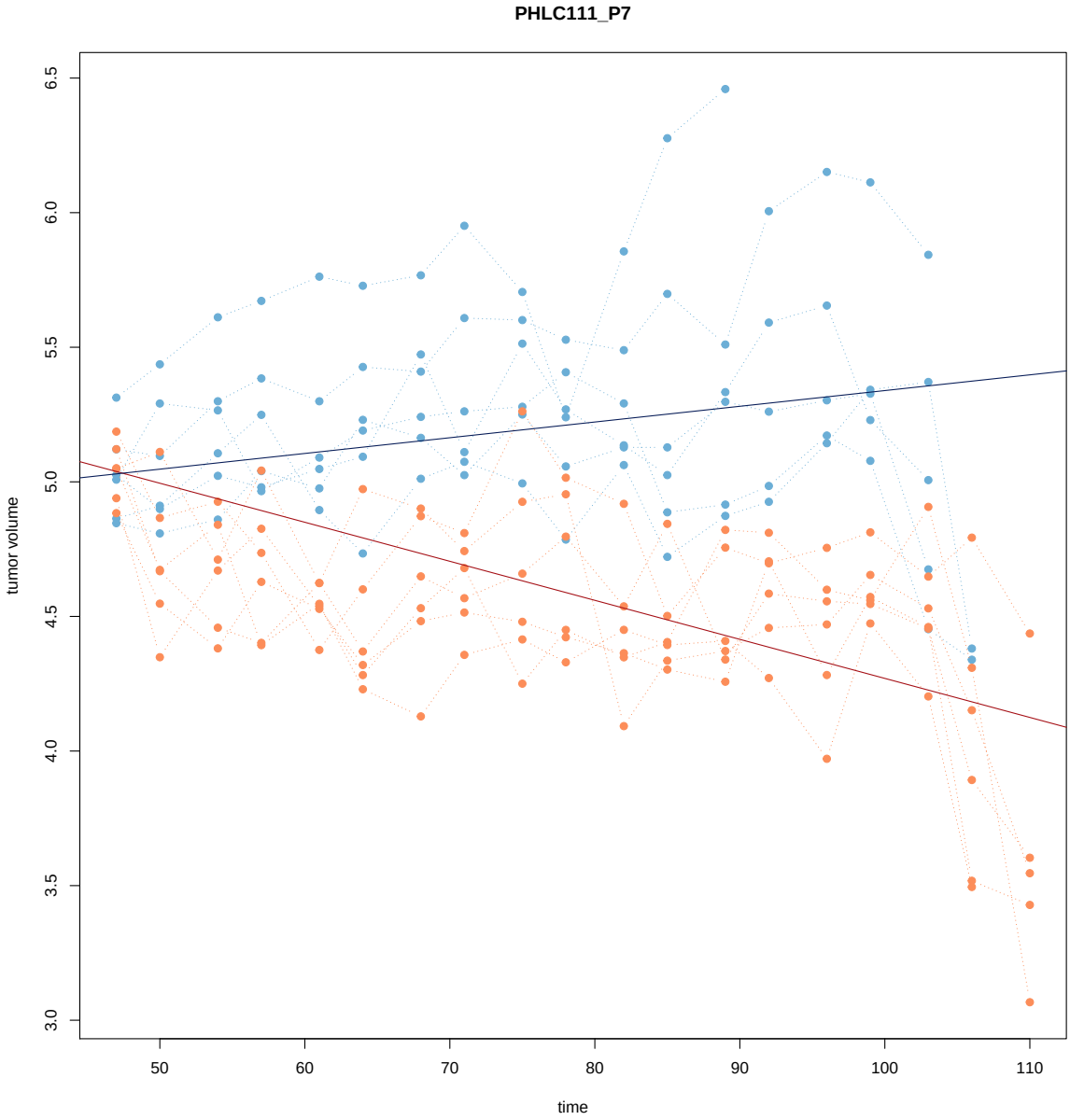
```
batchNames <- batchNames(lpdx)
expDesign  <- expDesign(lpdx, batchNames[1])
ang <- calculateAngle(lpdx, expDesign, treatment.only = TRUE, plot=TRUE)
```

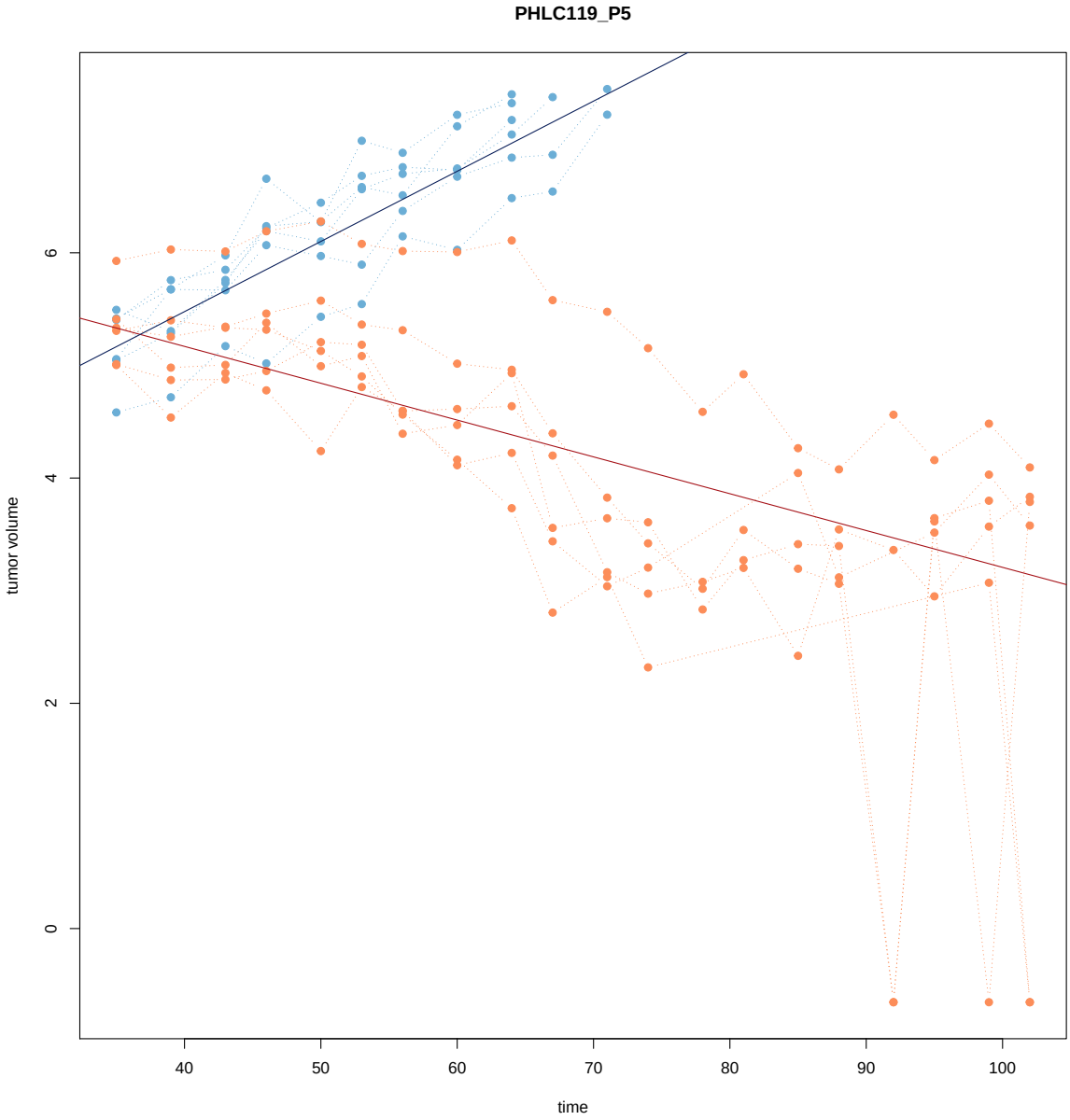


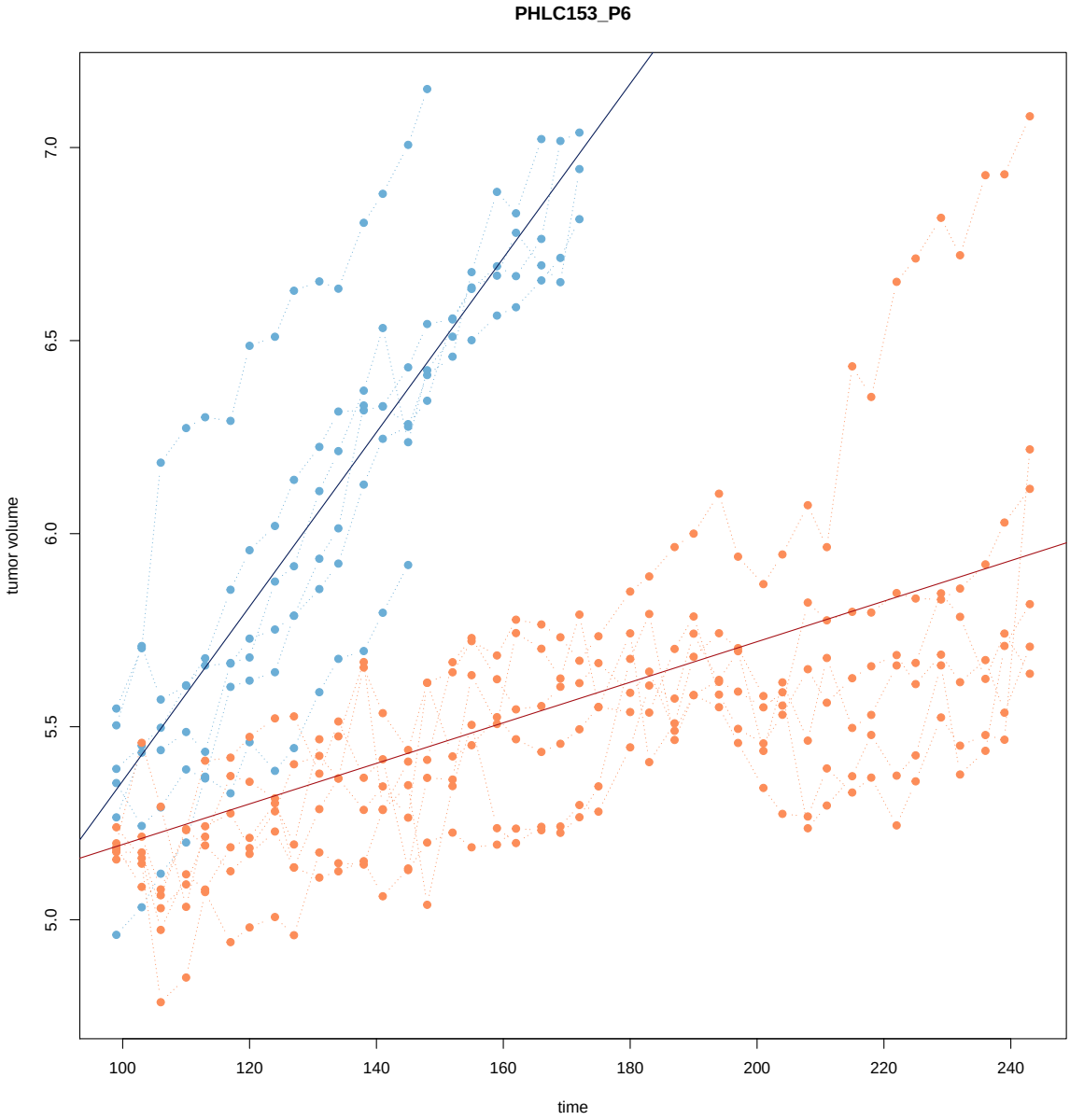

```
print(ang)
```

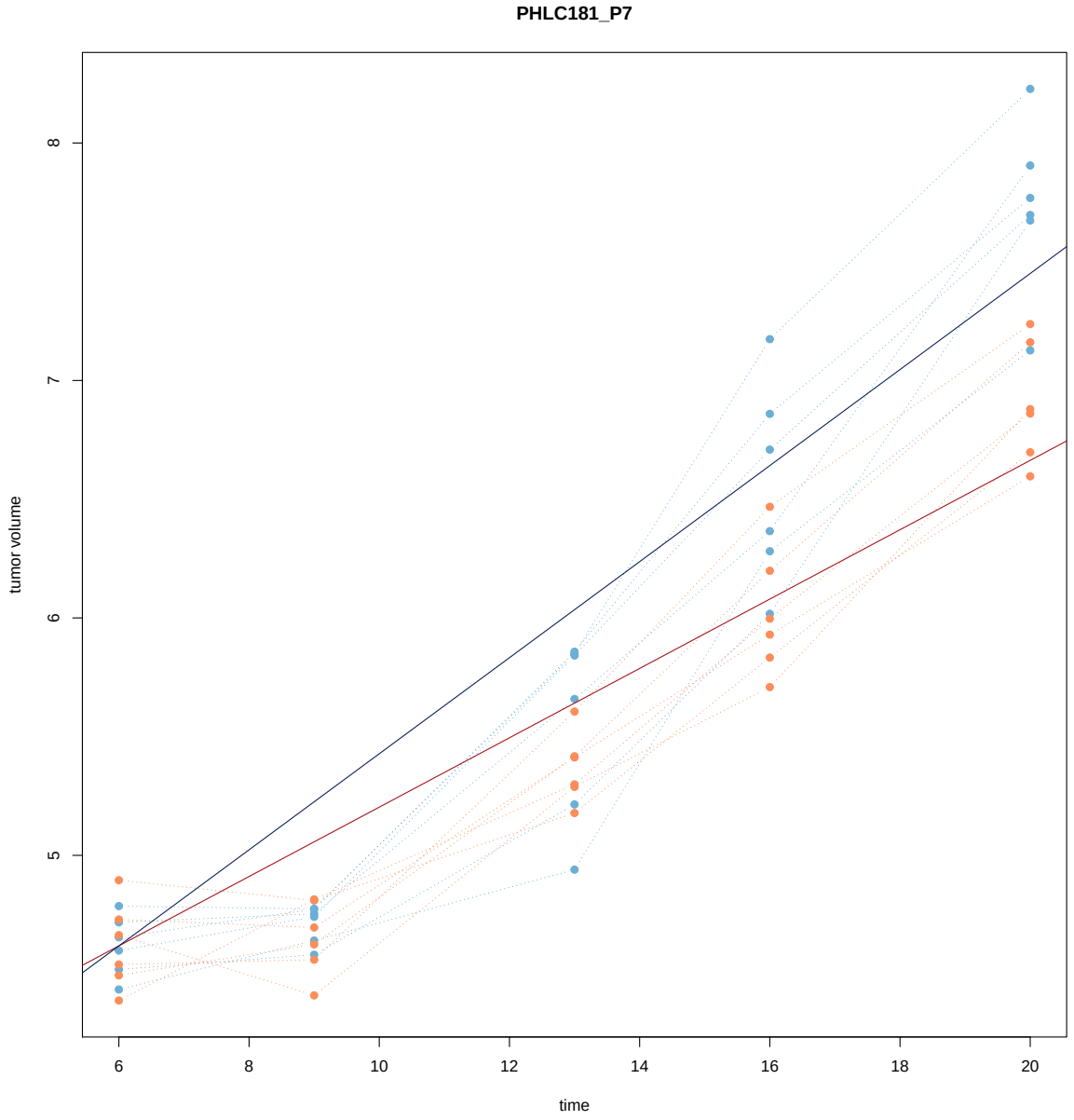
```
## $PHLC1106_P5
## [1] 0.671583
```

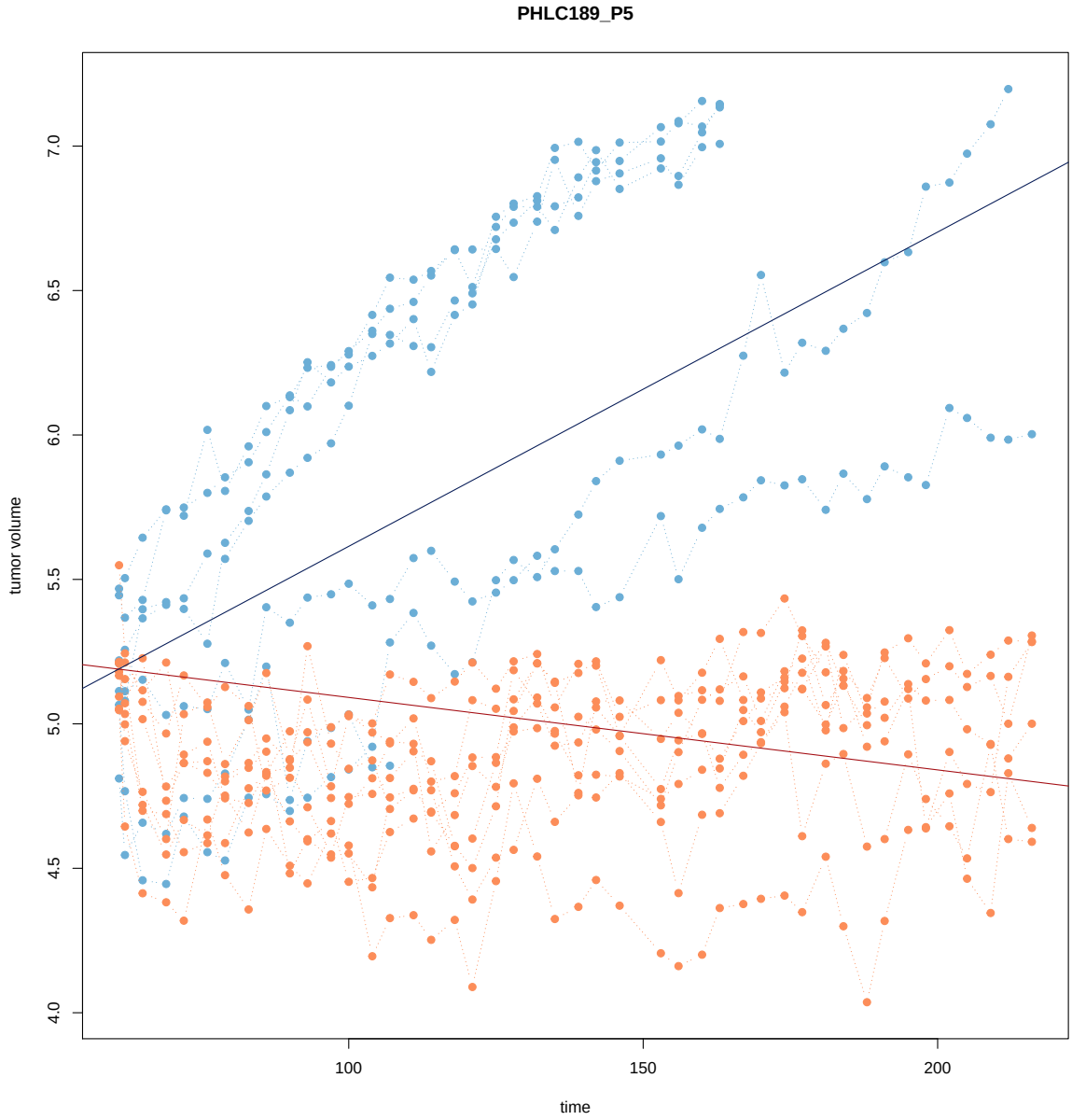
```
#par(mfrow=c(5,3))
for(I in batchNames)
{
  expDesign  <- expDesign(lpdx, I)
  ang <- calculateAngle(lpdx, expDesign, treatment.only = TRUE, plot=TRUE)
#  print(ang)
}
```

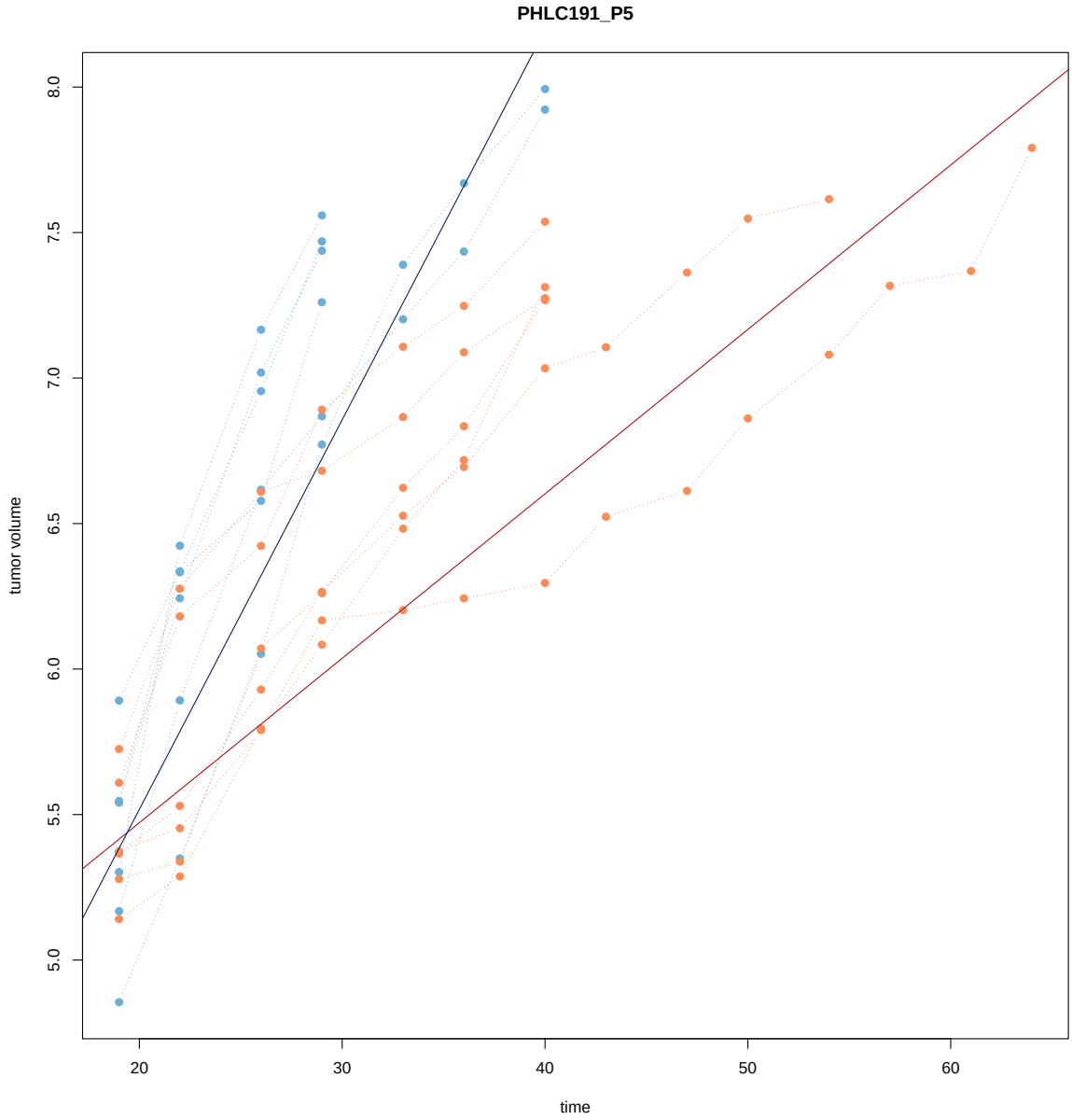

PHLC111_P7
tumor volume
time
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
50
60
70
80
90
100
110

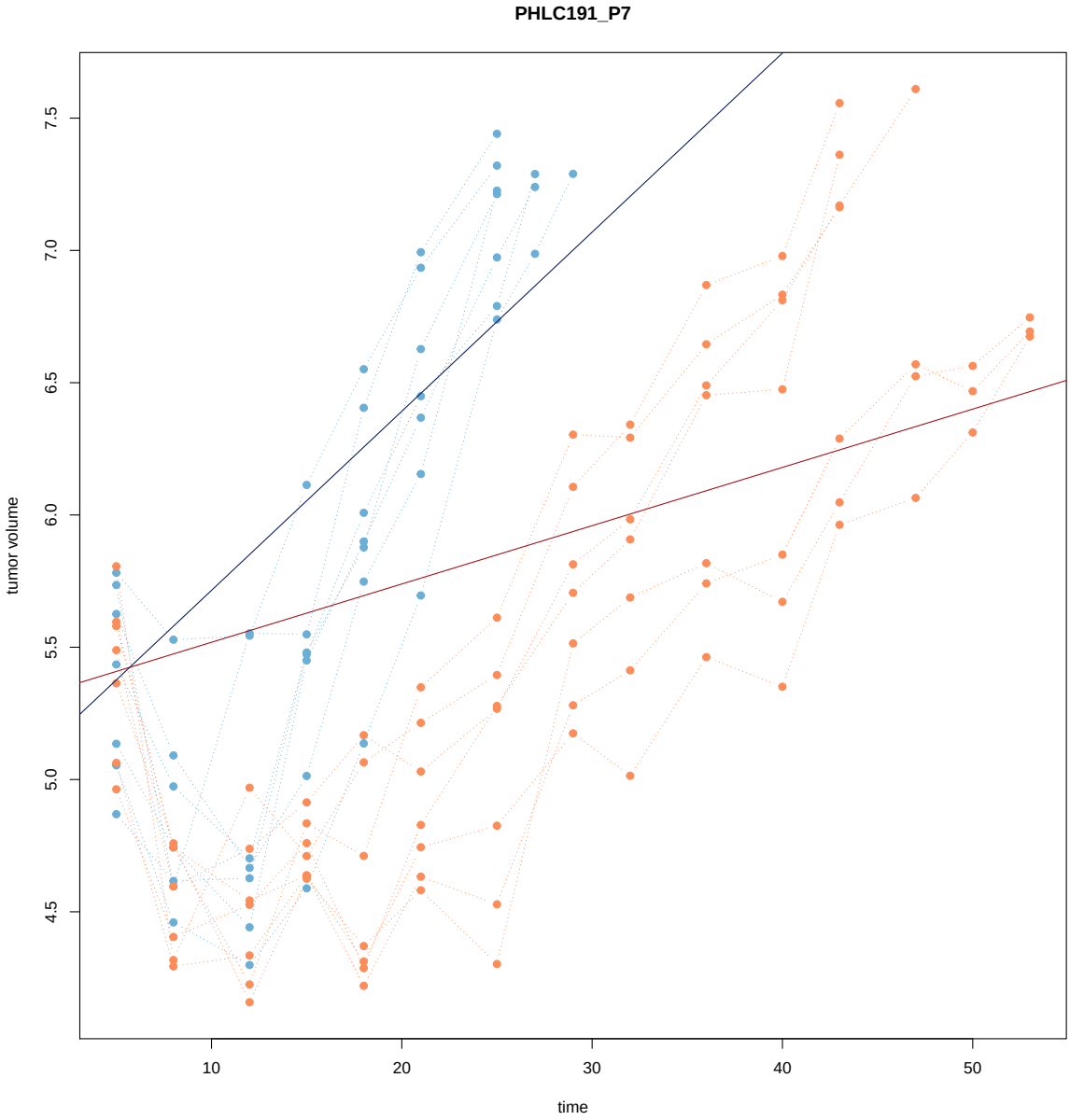

PHLC119_P5
tumor volume
time
0
2
4
6
40
50
60
70
80
90
100

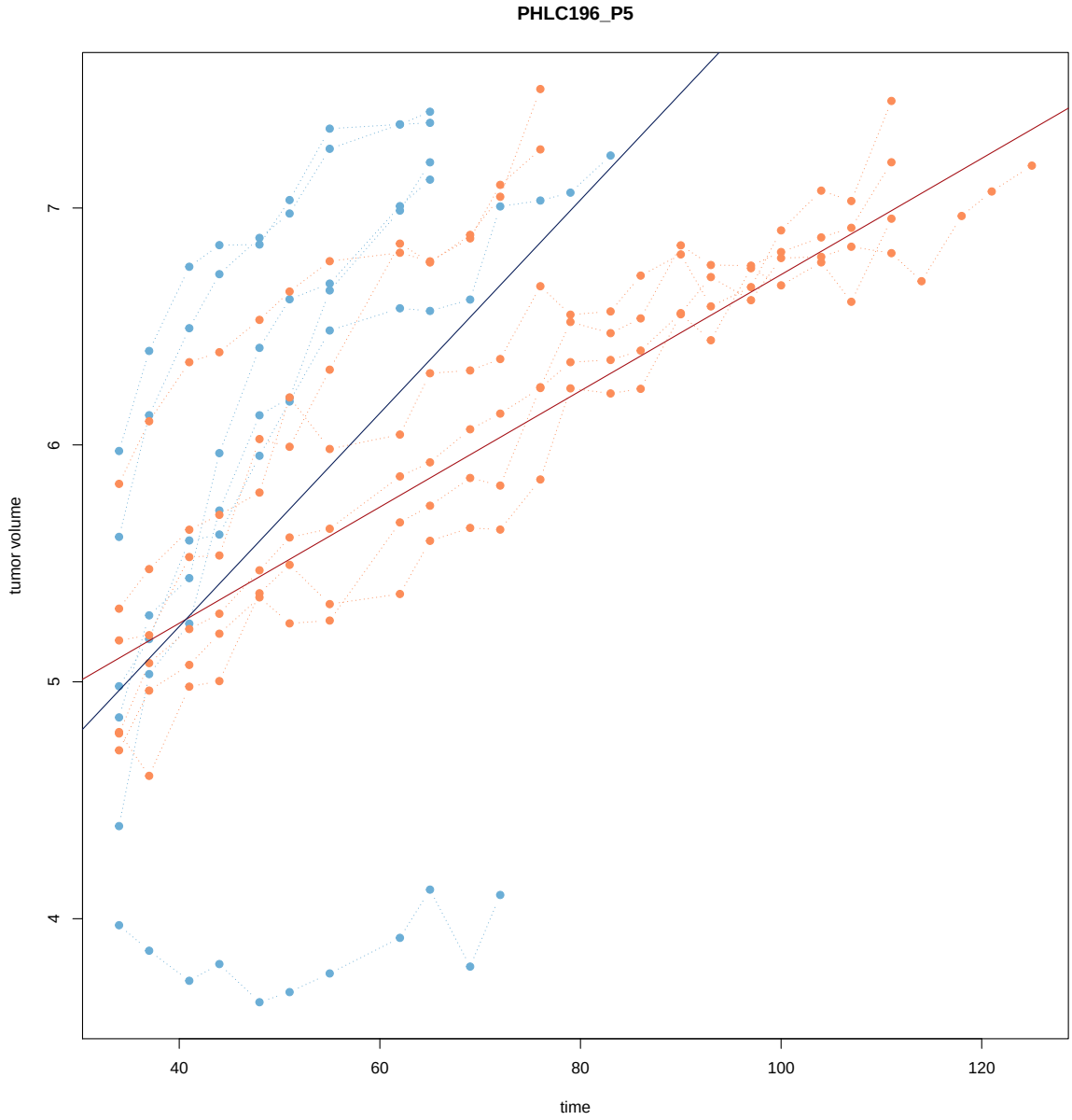

PHLC153_P6
tumor volume
time
5.0
5.5
6.0
6.5
7.0
100
120
140
160
180
200
220
240

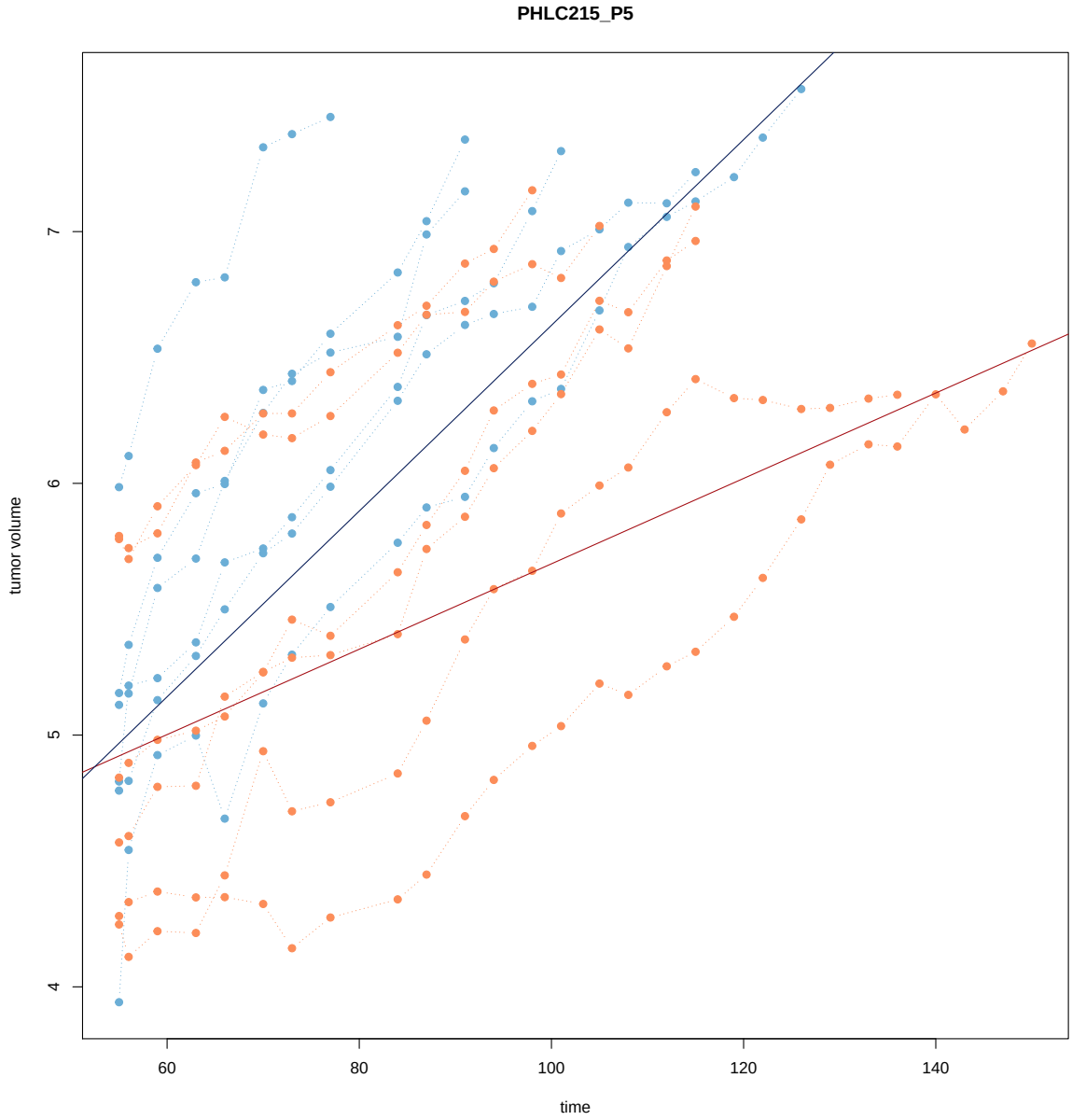

PHLC181_P7
tumor volume
time
5
6
7
8
6
8
10
12
14
16
18
20

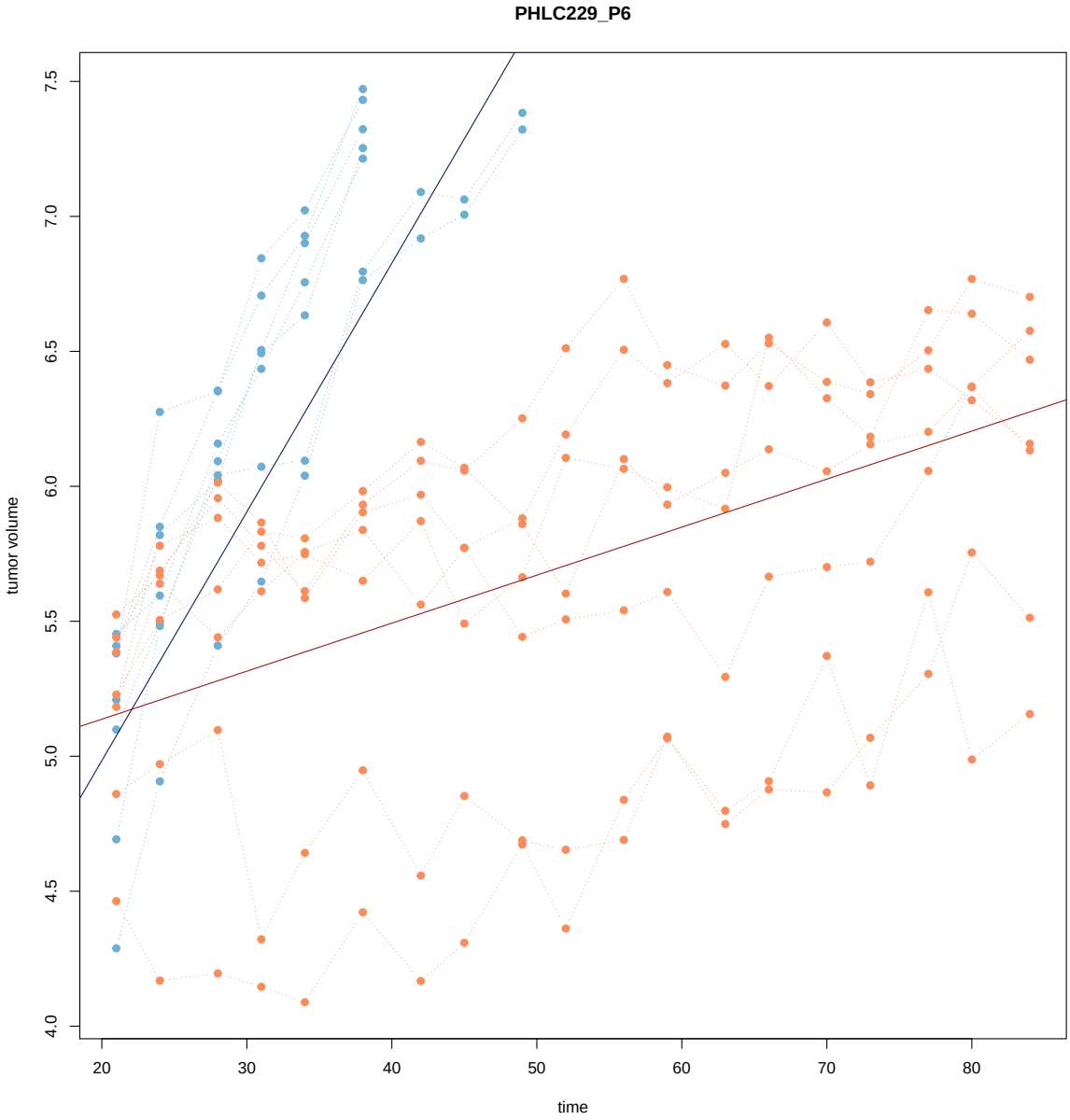

PHLC189_P5
tumor volume
time
4.0
4.5
5.0
5.5
6.0
6.5
7.0
100
150
200

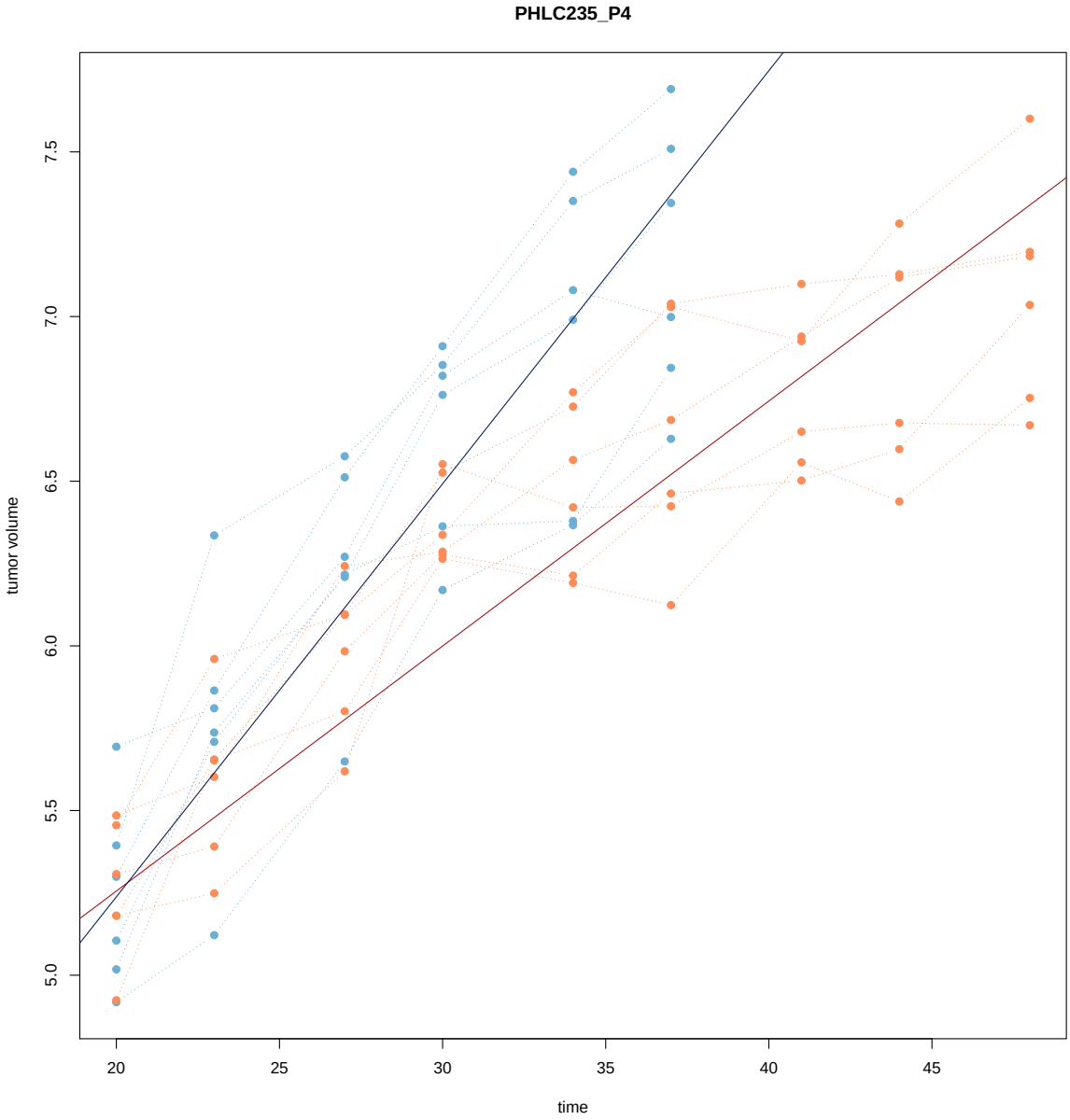

PHLC191_P5
tumor volume
time
5.0
5.5
6.0
6.5
7.0
7.5
8.0
20
30
40
50
60

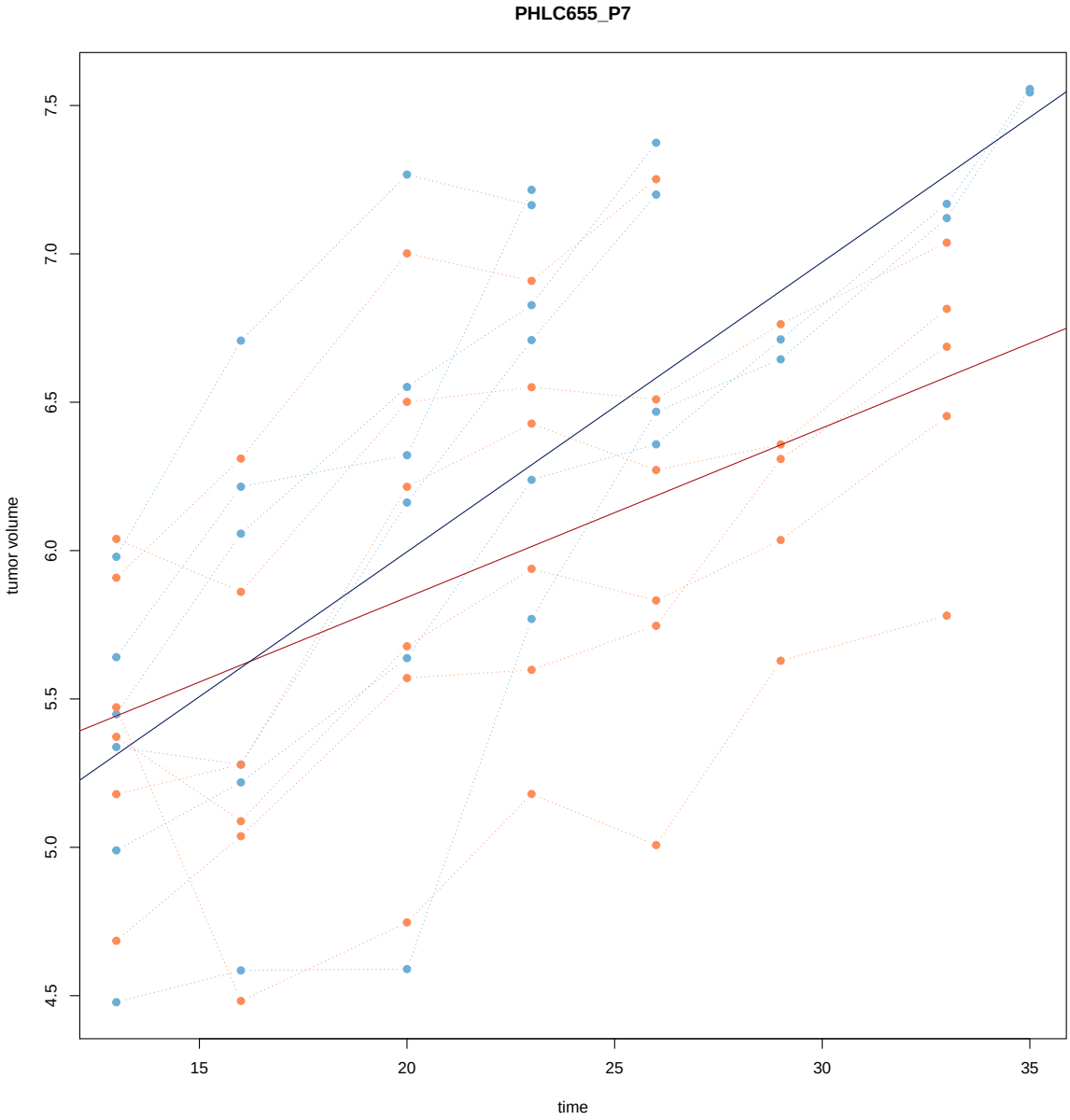

PHLC191_P7
tumor volume
time
4.5
5.0
5.5
6.0
6.5
7.0
7.5
10
20
30
40
50

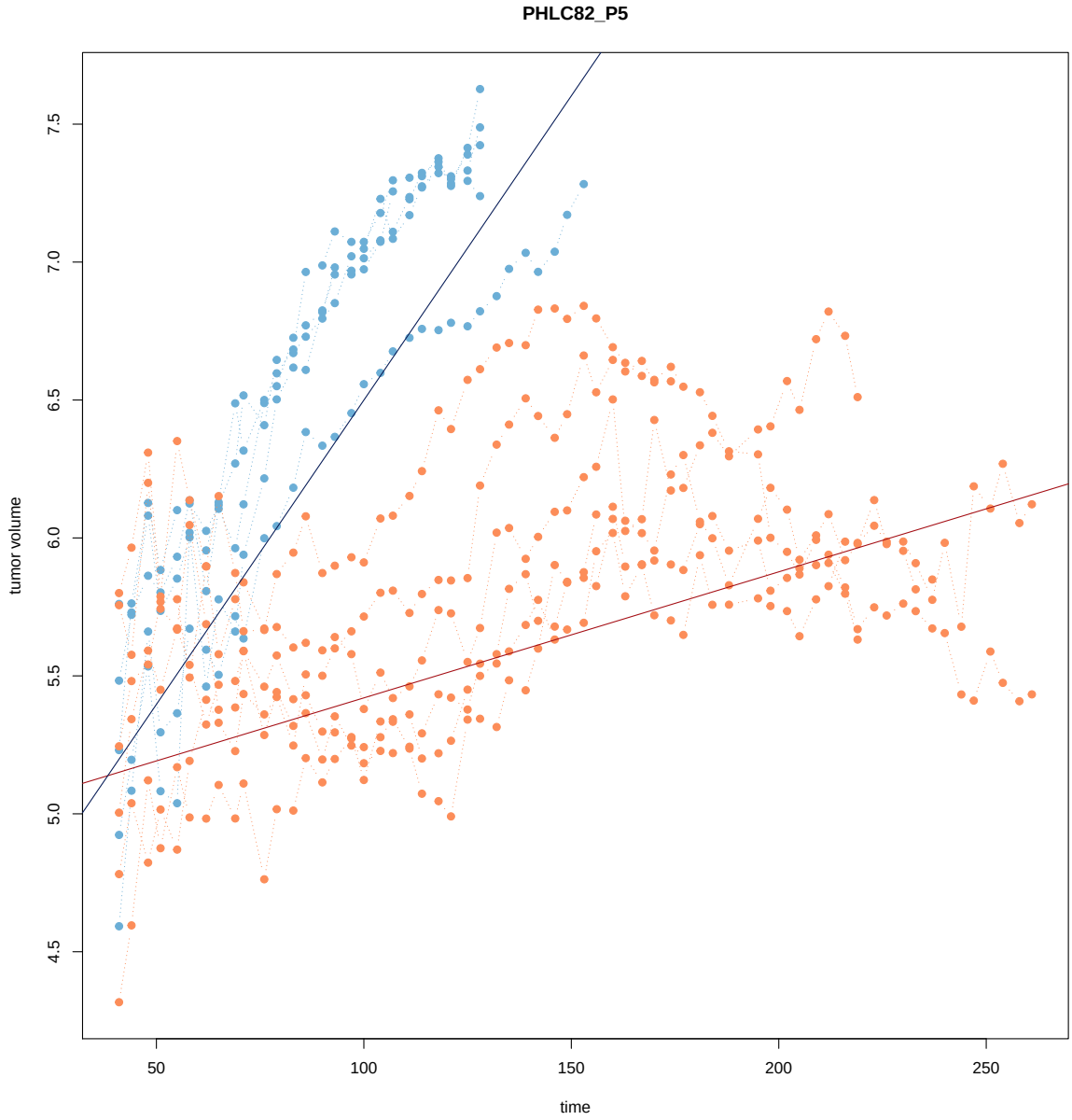

PHLC196_P5
tumor volume
time
4
5
6
7
40
60
80
100
120


PHLC215_P5
tumor volume
4
5
6
7
60
80
100
120
140
time


PHLC229_P6
tumor volume
time
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
20
30
40
50
60
70
80


PHLC235_P4
tumor volume
time
5.0
5.5
6.0
6.5
7.0
7.5
20
25
30
35
40
45


PHLC655_P7
tumor volume
time
4.5
5.0
5.5
6.0
6.5
7.0
7.5
15
20
25
30
35



Summarize Response of PDXs Get slop of each model and combine summarize all model slop which belongs to same patient by "mean"

```
lpdx_slop <- summarizeResponse(lpdx, response.measure = "slop",
                               group.by="patient.id", summary.stat = "mean")
```

Get angle between treatment and control model ids. For each batch it will give one angle value

```
lpdx_angle <- summarizeResponse(lpdx, response.measure = "angle")
```

Get mutation expression profile

```
ldxe_mut <- getMolecularProfiles(lpdx, data.type="mutation")
print(ldxe_mut)
```

```
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 16116 features, 12 samples
##   element names: exprs
## protocolData: none
## phenoData
##   sampleNames: PHLC1106 PHLC111 ... PHLC82 (12 total)
##   varLabels: PHLC.ID X.ID
##   varMetadata: labelDescription
## featureData
##   featureNames: NOC2L ISG15 ... RNF128 (16116 total)
##   fvarLabels: probe.Id
##   fvarMetadata: labelDescription
## experimentData: use 'experimentData(object)'
## Annotation: MUT
```

The sample names in expression set are called biobase.id in model slot. Sample names from the expression set canb be be mapped to individual PDX model.ids as

```
# get sample names
library(Biobase)
```

```
## Loading required package: BiocGenerics
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     Filter, Find, Map, Position, Reduce, anyDuplicated, append,
##     as.data.frame, cbind, colnames, do.call, duplicated, eval,
##     evalq, get, grep, grepl, intersect, is.unsorted, lapply,
##     lengths, mapply, match, mget, order, paste, pmax, pmax.int,
##     pmin, pmin.int, rank, rbind, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which, which.max,
##     which.min
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
sn <- Biobase::sampleNames(ldxe_mut)
smap <- mapModelSlotIds(lpdx, id=sn, id.name = "biobase.id", map.to = "model.id")
head(smap)
```

```
##                       biobase.id             model.id
## PHLC1106_P5.501.A1.1   PHLC1106 PHLC1106_P5.501.A1.1
## PHLC1106_P5.504.A4.1   PHLC1106 PHLC1106_P5.504.A4.1
## PHLC1106_P5.506.B1.1   PHLC1106 PHLC1106_P5.506.B1.1
## PHLC1106_P5.507.B2.1   PHLC1106 PHLC1106_P5.507.B2.1
## PHLC1106_P5.508.B3.1   PHLC1106 PHLC1106_P5.508.B3.1
## PHLC1106_P5.511.C1.1   PHLC1106 PHLC1106_P5.511.C1.1
```

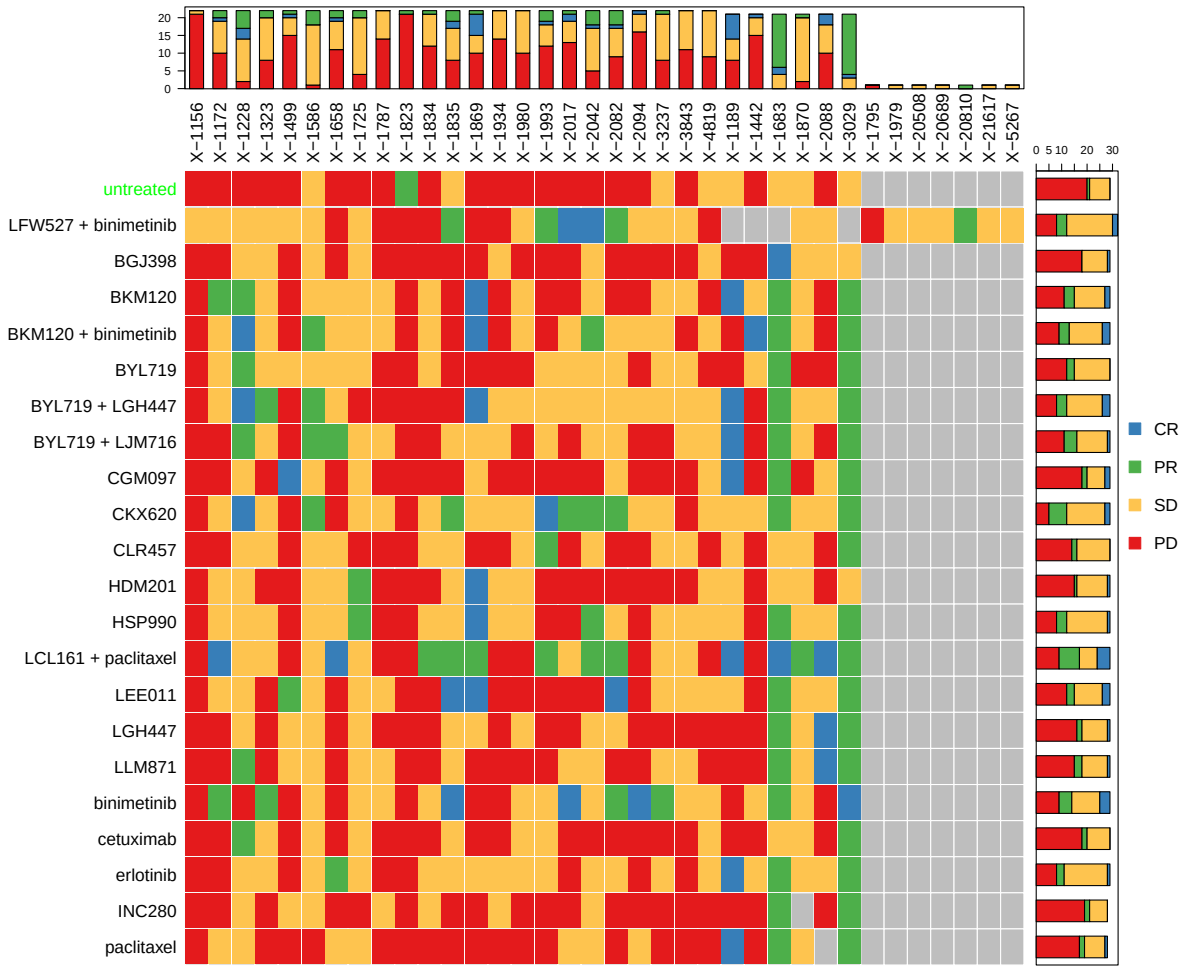
What should we do here

```
df = getExperiment(lpdx,"PHLC119_P5.506.B1.3")
print(df[df$time>85 & df$time<109, c("time", "width", "length", "volume", "comment", "dose")])
```

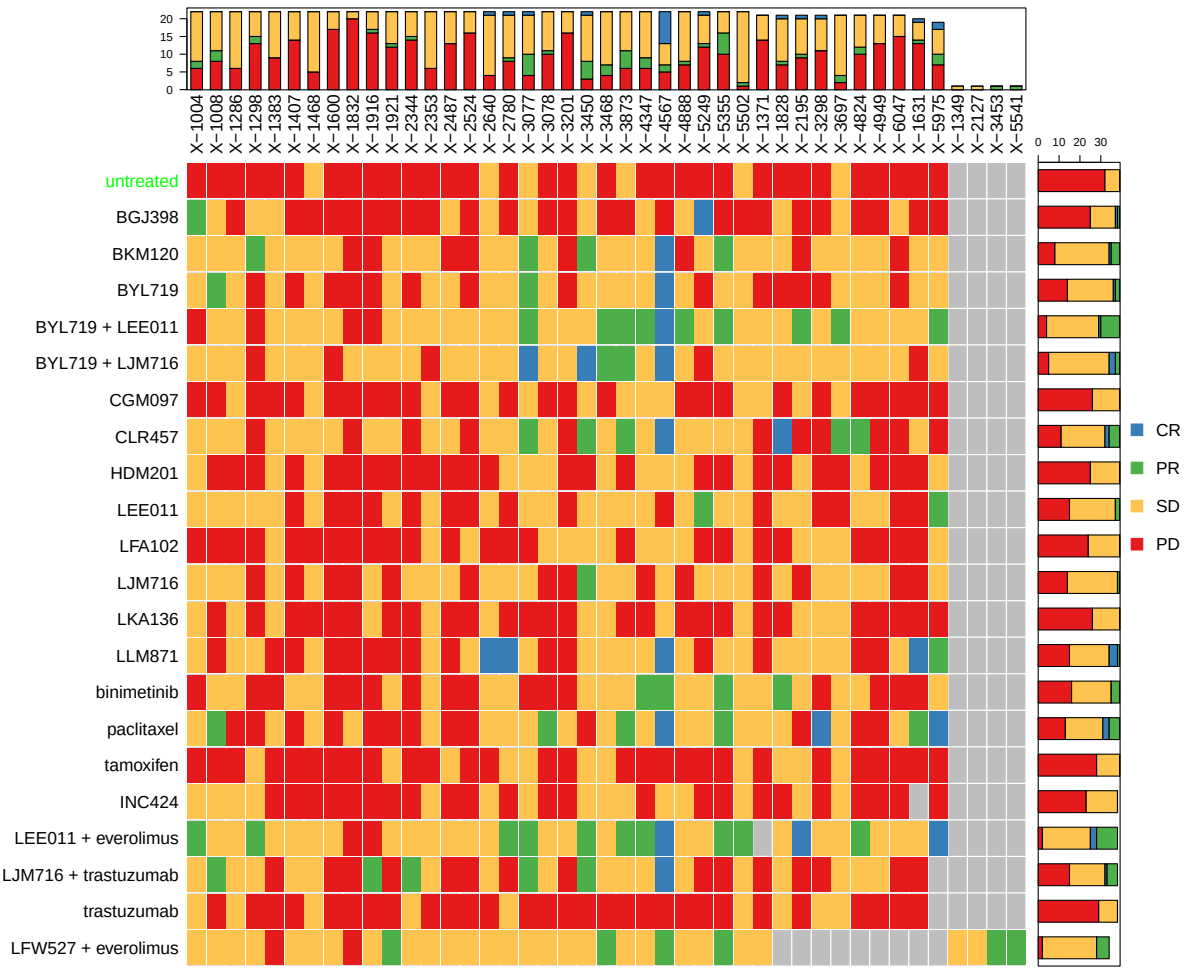
```
##    time width length   volume
## 22   88  3.20   4.01 21.35245
## 23   92  1.00   1.00  0.52000
## 24   95  3.71   5.19 37.14655
## 25   99  1.00   1.00  0.52000
## 26  102  4.13   4.98 44.17055
## 27  106  3.32   4.22 24.18755
##                                          comment     dose
## 22                                           <NA> 78.51786
## 23 small bud - put in arbritary "1" measurements 82.25357
## 24                                           <NA> 79.66071
## 25 small bud - put in arbritary "1" measurements 83.27500
## 26                                           <NA> 81.25357
## 27                                Stop treatment  0.00000
```

Create mRECIST plot for PDXE Lung Cancer data

```
data(pdxe)
df <- getmRECIST(pdxe)
## add tumor.type information
dfMap <- mapModelSlotIds(object=pdxe, id=df$model.id, id.name="model.id",
                         map.to="tumor.type", unique = FALSE)
if(all(df$model.id==dfMap$model.id)) {df$tumor.type = dfMap$tumor.type}
lungDf = df[df$tumor.type=="NSCLC", ]
#pdf(file="DATA-raw/mRECIST_plot_NSCLC.pdf", width=12, height=10)
plotmRECIST(lungDf, groupBy = "biobase.id", control.name = "untreated")
```




```
#pdf(file="DATA-raw/mRECIST_plot_BRCA.pdf", width=12, height=10)
brDF = df[df$tumor.type=="BRCA", ]
plotmRECIST(brDF, groupBy = "biobase.id", control.name = "untreated")
```

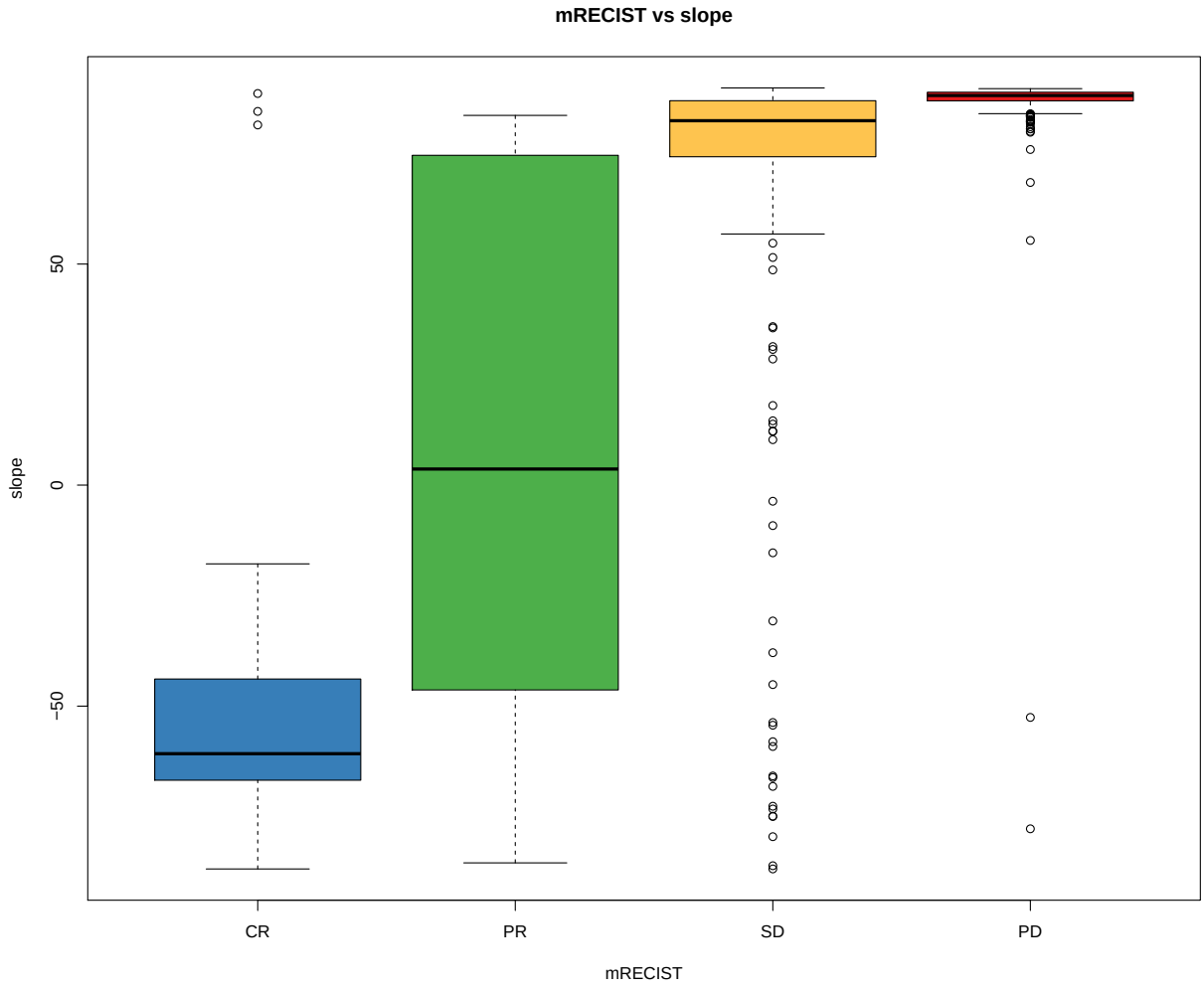



```
#dev.off()
```

Creat mR vs slop bar-plot

```
data(pdxe)

pm = modelInfo(pdxe)
lungPID = unique(pm[pm$tumor.type=="NSCLC", "patient.id"])

pdxe_slop <- summarizeResponse(pdxe, response.measure = "slop",
                               group.by="patient.id", summary.stat = "mean")

lung_pdxe_slope <- pdxe_slop[, lungPID]

##-------
pdxe_mR <- summarizeResponse(pdxe, response.measure = "mRECIST_recomputed",
                               group.by="patient.id")

lung_pdxe_mR = pdxe_mR[, lungPID]
```

```
slope=c(); mR=c()
for(dn in rownames(lung_pdxe_slope))
{
  for(pi in colnames(lung_pdxe_slope))
  {
    v = c(lung_pdxe_slope[dn,pi], lung_pdxe_mR[dn,pi])
    if(!is.na(v[1]) & !is.na(v[1]))
    { slope = c(slope,v[1]); mR=c(mR,v[2]) }
  }
}

df = data.frame(mR= mR, slope= as.numeric(slope), stringsAsFactors = FALSE)
df$mR= factor(df$mR, c("CR", "PR", "SD", "PD"))

colPalette = c("#377eb8", "#4daf4a", "#fec44f", "#e41a1c")
#pdf(file="DATA-raw/boxplot_lungCancer.pdf", width=12, height=10)
boxplot(slope~mR, data=df, col=colPalette,
  main="mRECIST vs slope", xlab="mRECIST", ylab="slope")
```



```
#dev.off()
```