Annex. Creation of artificial data with 1% and 10% L-shaped genes

Alex sanchez and Berta Miro

8/28/2019

Classification analysis of genes following an L-shaped scatterplot (genes regulated by methylation, GRM) vs genes not following an L-shaped scatterplot (genes not regulated by methylation, GNRM).

The exercise parameters that we will apply are as follows:

1. Tests with a small number of samples (researcher type dataset) 1.1. Sample with 1% GRM, 99% GNRM 1.2. Sample with 10% GRM, 90% GNRM
2. Tests with a small number of samples (TCGA type dataset) 2.1. Sample with 1% GRM, 99% GNRM 2.2. Sample with 10% GRM, 90% GNRM

The GRM will follow 3 representative types of shapes; therefore, the distribution of the genes will be 0.33%, 0.34%, 0.33% in the 1% GRM scenario and 3.5%, 3.5%, 3% in the 10% scenario.

The GNRM will also follow 3 different types of shapes; the distribution of the genes will be 30% in the 90% GNRM scenario and 33% in the 99%. scenario.

```
require(knitr)
```

```
## Loading required package: knitr
```

```
opts_chunk$set(
concordance=FALSE, echo=TRUE, cache=FALSE, warning=FALSE, error=FALSE, message=FALSE)
```

```
#load packages
# install.packages("https://cran.r-project.org/src/contrib/Archive/mvtnorm/mvtnorm_1.0-8.tar.gz", repos
# install.packages("simstudy")
library(simstudy)
library(ggplot2)
```

CREATION OF SYNTHETIC DATASETS

Create simulation data by simstudy package.

First we create the "true" L-shaped genes with the splines method. The L-shaped genes will follow a pre-defined spline, in which the knots, limits and dispersion can be tuned. The splines are generated based on a fixed X variable (which represents the methylation). The package used is **R/simstudy**.

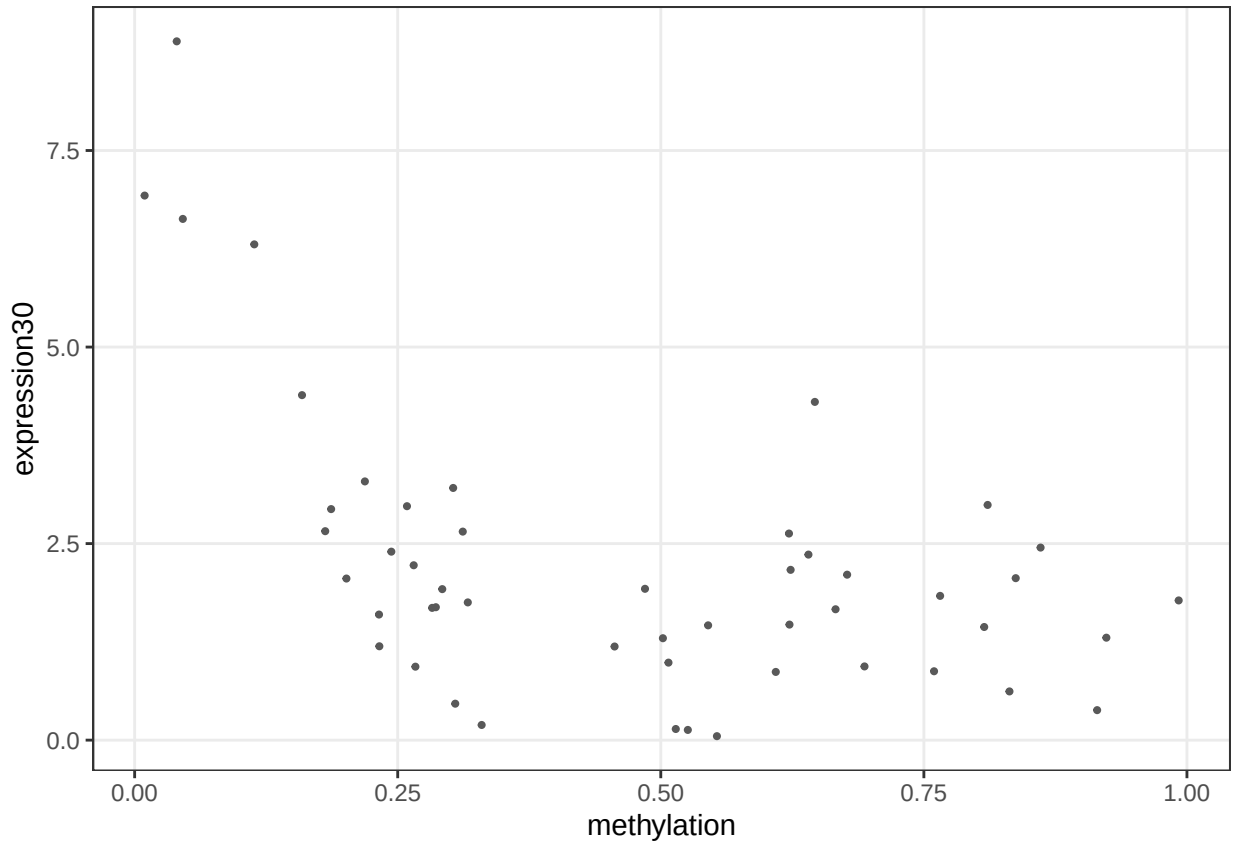
For the non-L genes, we also will use the spline generation function, in which we mirror an inverse L, with large noise in the variables generated.

Creation of L-shaped simulated data

Creation of simulated data for 50 data points First we define the variable methylation with the *defData* function, that variable will be generated with number ranging between 0 and 1 and will generate the number of points tht we want through the first argument of the *genData* function. Then, we use the *genSpline* function to create a variable based on the previous one that will represent the gene expression following a spline function defined to have an L-shape. The spline L-shape curving will be modified with the 'theta' and the 'knots' arguments.

creating 100 L shaped variables with 50 data points L-shaped genes: L

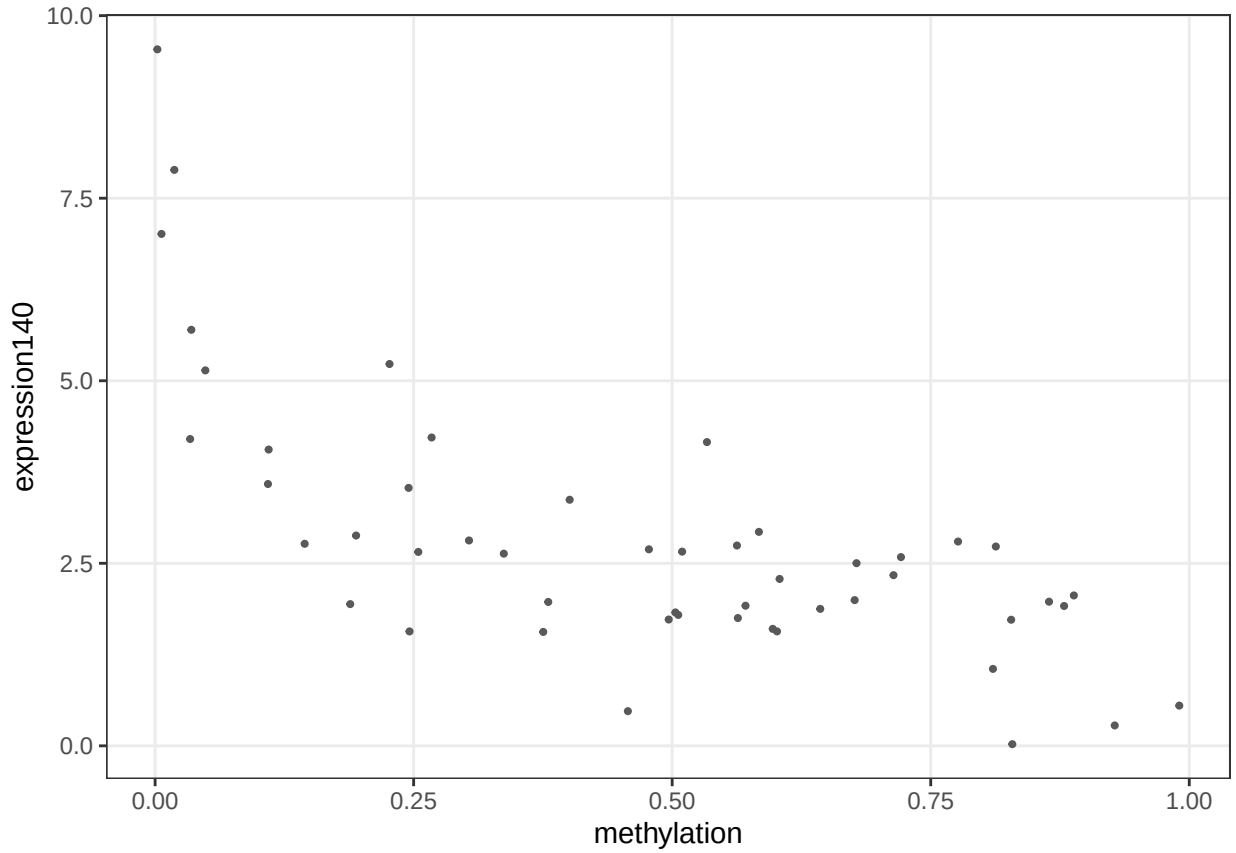
```
set.seed(1234)
#creating 100 L shaped variables with 50 data points

##L-shaped genes: L
ddef <- defData(varname = "methylation", formula = "0;1", dist = "uniform")
theta1 = c(0.9, 0.1,  0)
knots <- c(0.2)
dt <- genData(50, ddef)


for(i in 1:100){
dt1 <- genSpline(dt = dt, newvar = paste0("expression",i),
                 predictor = "methylation", theta = theta1,
                 knots = knots, degree = 1,
                 newrange = "1;9",
                 noise.var = 1)
}

dt1x <- abs(dt1)

#variable visualization
ggplot(data = dt1x, aes(x=methylation, y=expression30)) +
  geom_point(color = "grey35", size = 0.75) +
  #geom_smooth(se=FALSE, color="blue", size = 1, method = "auto") +
  theme_bw() + theme(panel.grid.minor = element_blank())
```




```
colNames1 <- names(dt1x)[3:102]
```

L-shaped genes: concave shape

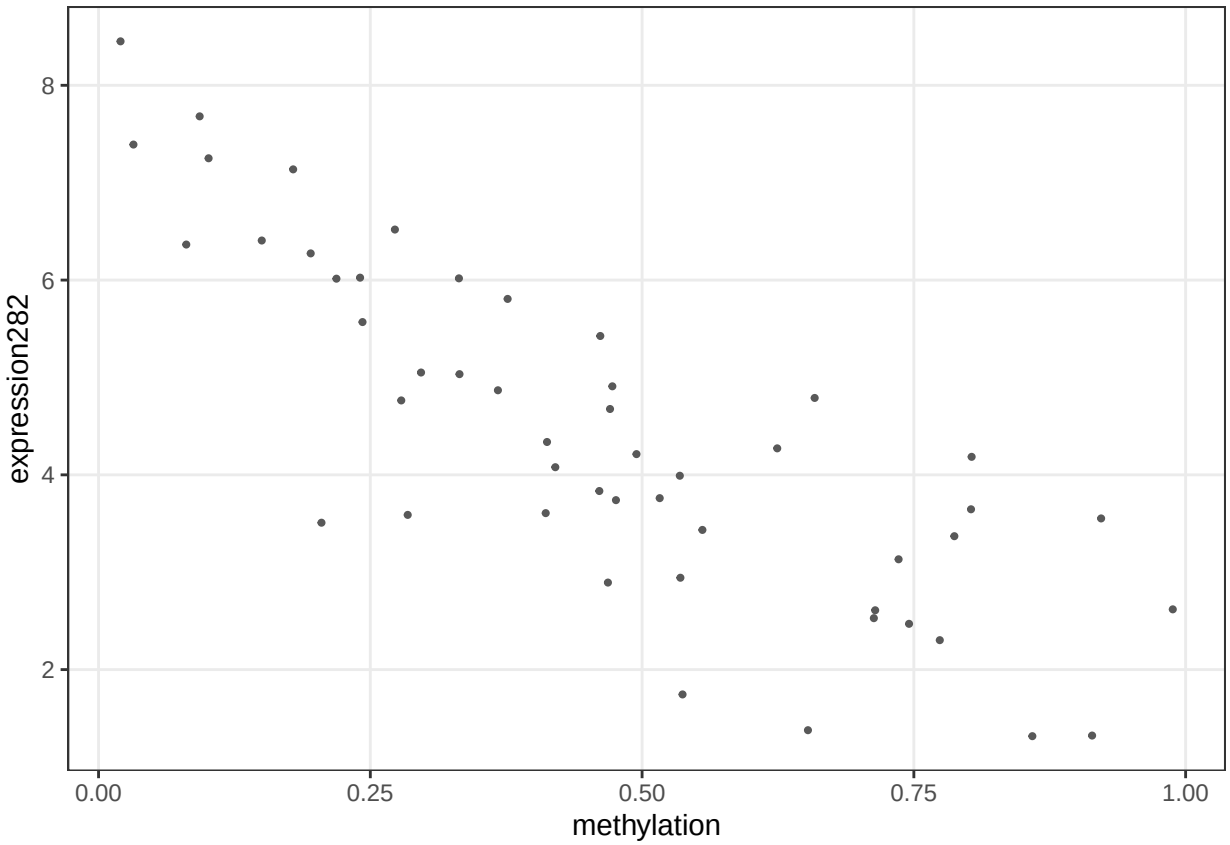
```
##L-shaped genes: concave shape
ddef <- defData(varname = "methylation", formula = "0;1", dist = "uniform")
theta2 = c(1, 0.4, 0.2, 0.1, 0)
knots <- c(0.10, 0.40, 0.60)
dt <- genData(50, ddef)


for(i in 101:200){
dt2 <- genSpline(dt = dt, newvar = paste0("expression",i),
                 predictor = "methylation", theta = theta2,
                 knots = knots, degree = 1,
                 newrange = "1;9",
                 noise.var = 1)
}

dt2x <- abs(dt2)

#variable visualization
ggplot(data = dt2x, aes(x=methylation, y=expression140)) +
  geom_point(color = "grey35", size = 0.75) +
  #geom_smooth(se=FALSE, color="blue", size = 1, method = "auto") +
  theme_bw() + theme(panel.grid.minor = element_blank())
```




```
colNames2 <- names(dt2x)[3:102]
```

L-shaped genes: linear shape (negative correlation)

```
##L-shaped genes: linear shape (negative correlation)
ddef <- defData(varname = "methylation", formula = "0;1", dist = "uniform")
theta3 = c(0.9, 0.7, 0.4, 0.1)
knots <- c(0.10, 0.40)
dt <- genData(50, ddef)


for(i in 201:300){
dt3 <- genSpline(dt = dt, newvar = paste0("expression",i),
                 predictor = "methylation", theta = theta3,
                 knots = knots, degree = 1,
                 newrange = "1;9",
                 noise.var = 1)
}

dt3x <- abs(dt3)

#variable visualization
ggplot(data = dt3x, aes(x=methylation, y=expression282)) +
  geom_point(color = "grey35", size = 0.75) +
  #geom_smooth(se=FALSE, color="blue", size = 1, method = "auto") +
  theme_bw() + theme(panel.grid.minor = element_blank())
```




```
colNames3 <- names(dt3x)[3:102]
```

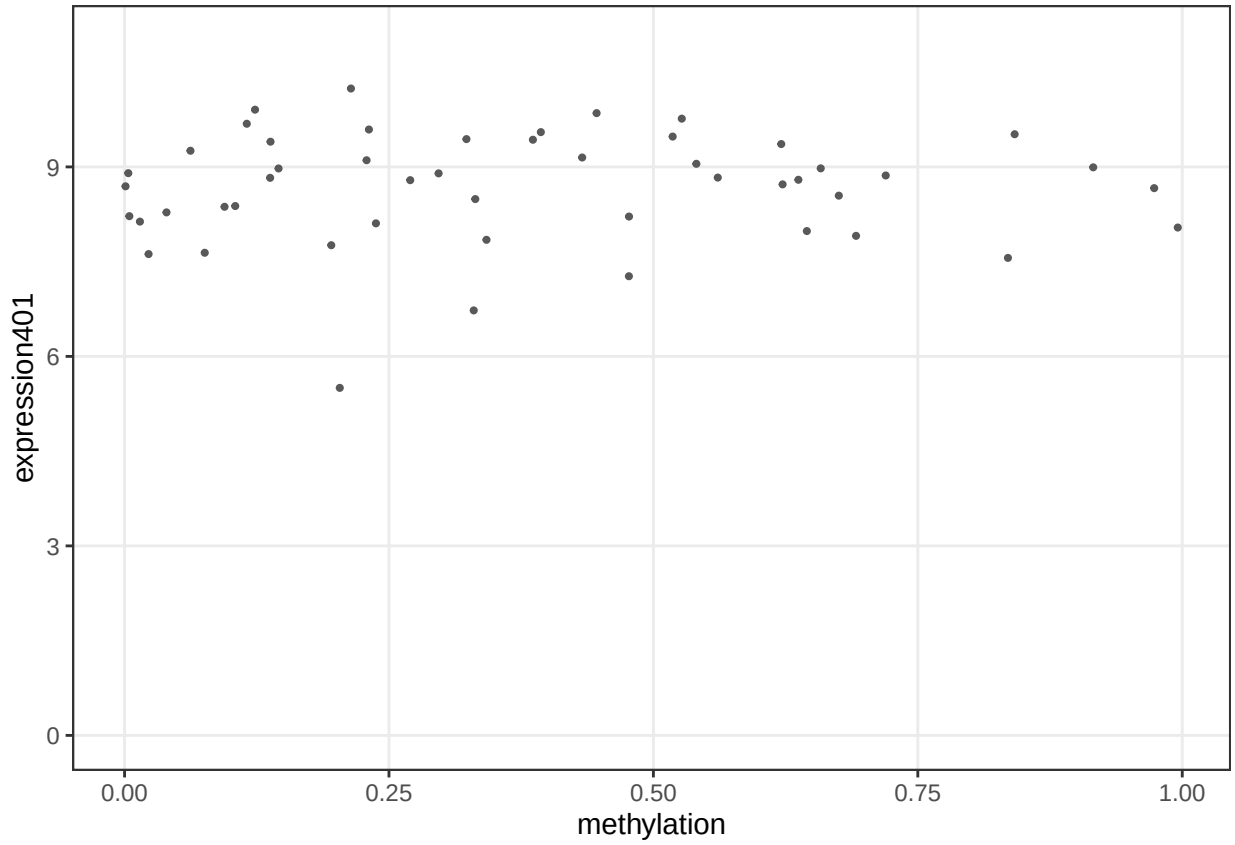
Cretion on non-L shaped simulation data for 50 data points

We use the 'newrange' arugument of the *genSpline* function to direct the points toward a particular quadrant to mimic the vertical distribution of expression vs methylation of some genes. To regulate the horizontal distribution of some points, we use the 'formula' argument of the *defData* function.

Non L shaped genes: top horizontal distribution

```
#creating 250 non L shaped variables with 50 data points - top horizontal distribution


ddef <- defData(varname = "methylation", formula = "0;1", dist = "uniform")
theta4 = c(0.7, 0.9, 0.8, 1)
knots2 <- c(0.3, 0.9)

dta <- genData(50, ddef)

for(i in 301:550){
dta1 <- genSpline(dt = dta, newvar = paste0("expression",i),
                  predictor = "methylation", theta = theta4,
                  knots = knots2, degree = 1,
                  newrange = "8;9",
                  noise.var = 1)
}
```

```r
dta1x <- abs(dta1)

#variable visualization
ggplot(data = dta1x, aes(x=methylation, y=expression401)) +
  geom_point(color = "grey35", size = 0.75) + ylim(0,11)+
  #geom_smooth(se=FALSE, color="blue", size = 1, method = "auto") +
  theme_bw() + theme(panel.grid.minor = element_blank())
```

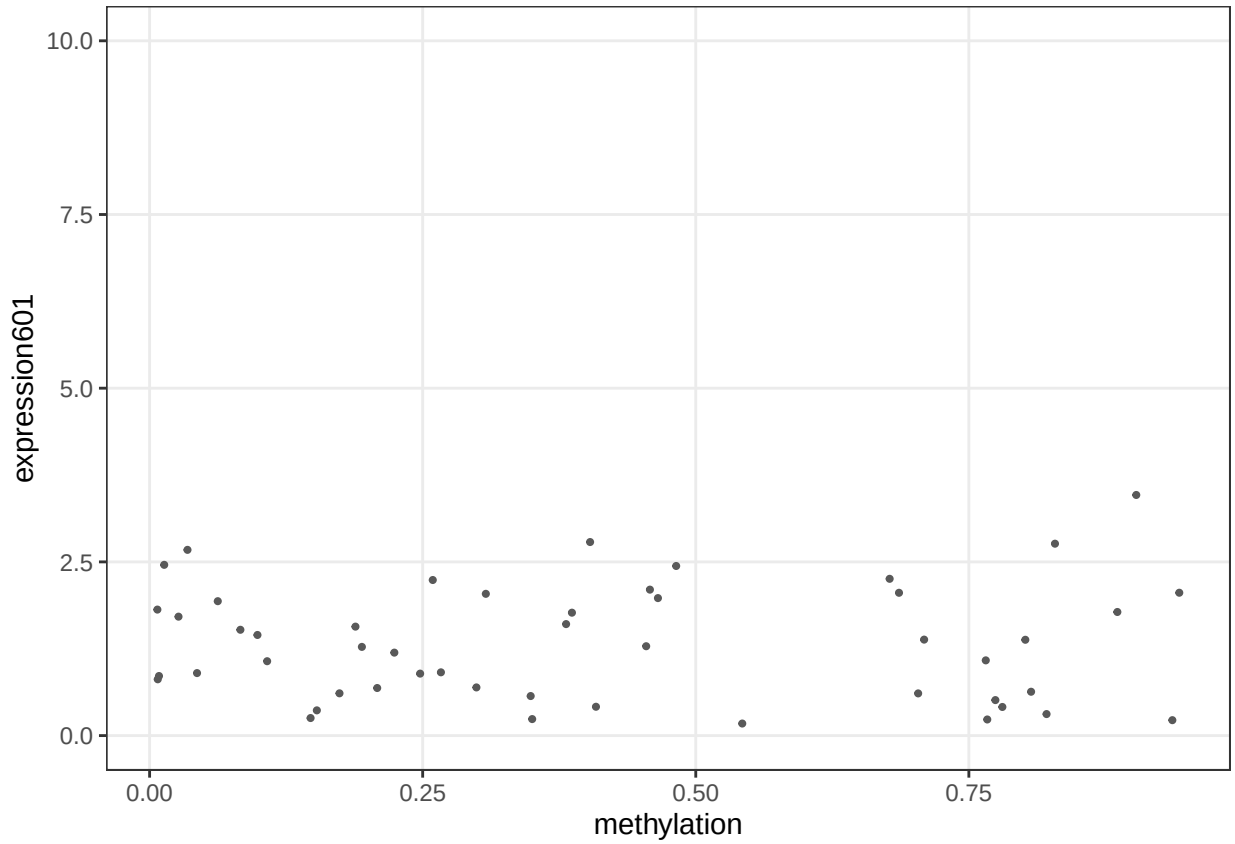



```r
colNames1a <- names(dta1x)[3:252]
```

Non L shaped genes: bottom horizontal distribution

```r
#creating 250 non L shaped variables with 50 data points - bottom horizontal distribution


ddef <- defData(varname = "methylation", formula = "0;1", dist = "uniform")
theta5 = c(0.1, 0.2, 0.1, 0.1)
knots2 <- c(0.3, 0.9)

dta <- genData(50, ddef)

for(i in 551:800){
dta2 <- genSpline(dt = dta, newvar = paste0("expression",i),
                  predictor = "methylation", theta = theta5,
                  knots = knots2, degree = 1,
                  newrange = "1;2",
```

```
                noise.var = 1)
}

dta2x <- abs(dta2)

#variable visualization
ggplot(data = dta2x, aes(x=methylation, y=expression601)) +
  geom_point(color = "grey35", size = 0.75) + ylim(0,10)+
  #geom_smooth(se=FALSE, color="blue", size = 1, method = "auto") +
  theme_bw() + theme(panel.grid.minor = element_blank())
```

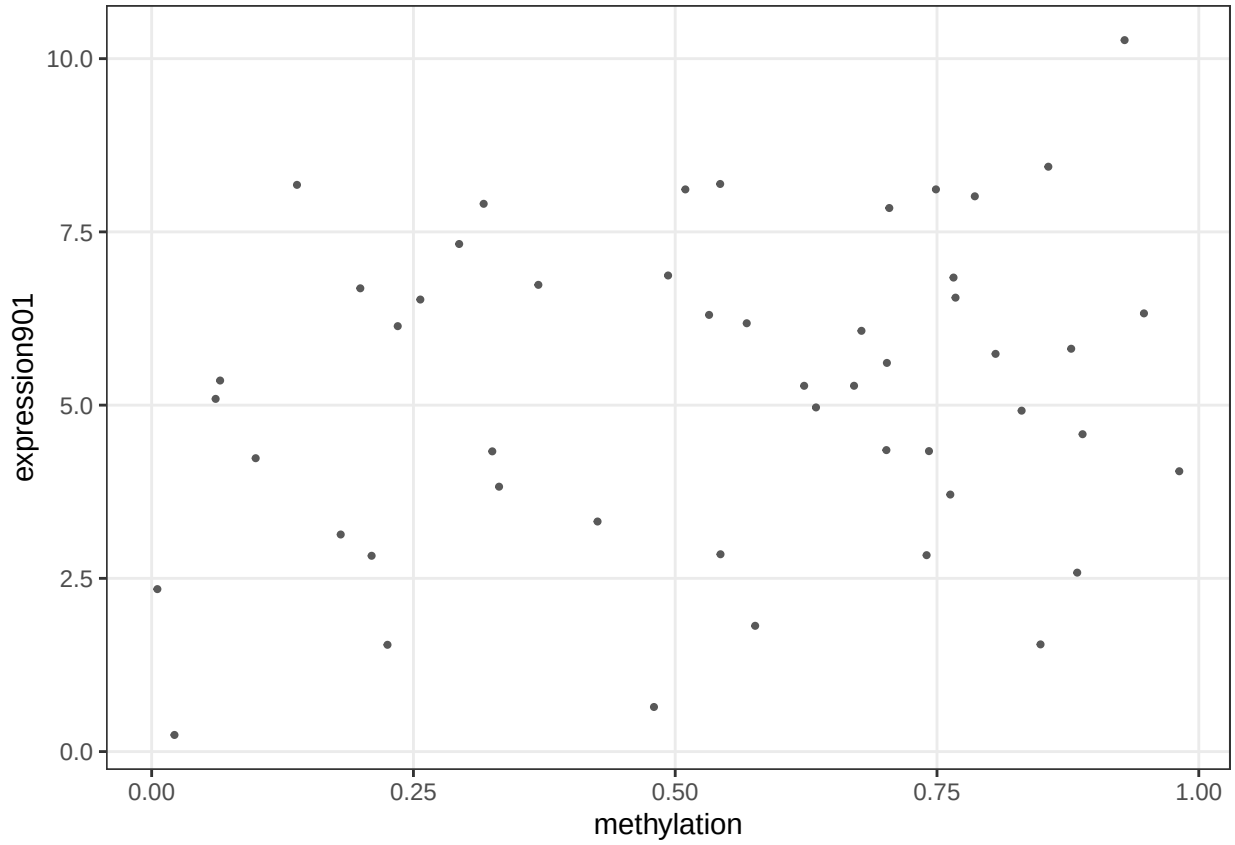



```
colNames2a <- names(dta2x)[3:252]
```

Non L shaped genes: random distribution

```
#creating 500 non L shaped variables with 50 data points - random distribution


ddef <- defData(varname = "methylation", formula = "0;1", dist = "uniform")
theta6 = c(0.3, 0.5, 0.4, 0.5, 0.6)
knots2 <- c(0.3,0.5,  0.9)
dta <- genData(50, ddef)

for(i in 801:1300){
dta3 <- genSpline(dt = dta, newvar = paste0("expression",i),
                predictor = "methylation", theta = theta6,
                knots = knots2, degree = 1,
```

```
                newrange = "1;9",
                noise.var = 7)
}

dta3x <- abs(dta3)

#variable visualization
ggplot(data = dta3x, aes(x=methylation, y=expression901)) +
  geom_point(color = "grey35", size = 0.75) +
  #geom_smooth(se=FALSE, color="blue", size = 1, method = "auto") +
  theme_bw() + theme(panel.grid.minor = element_blank())
```

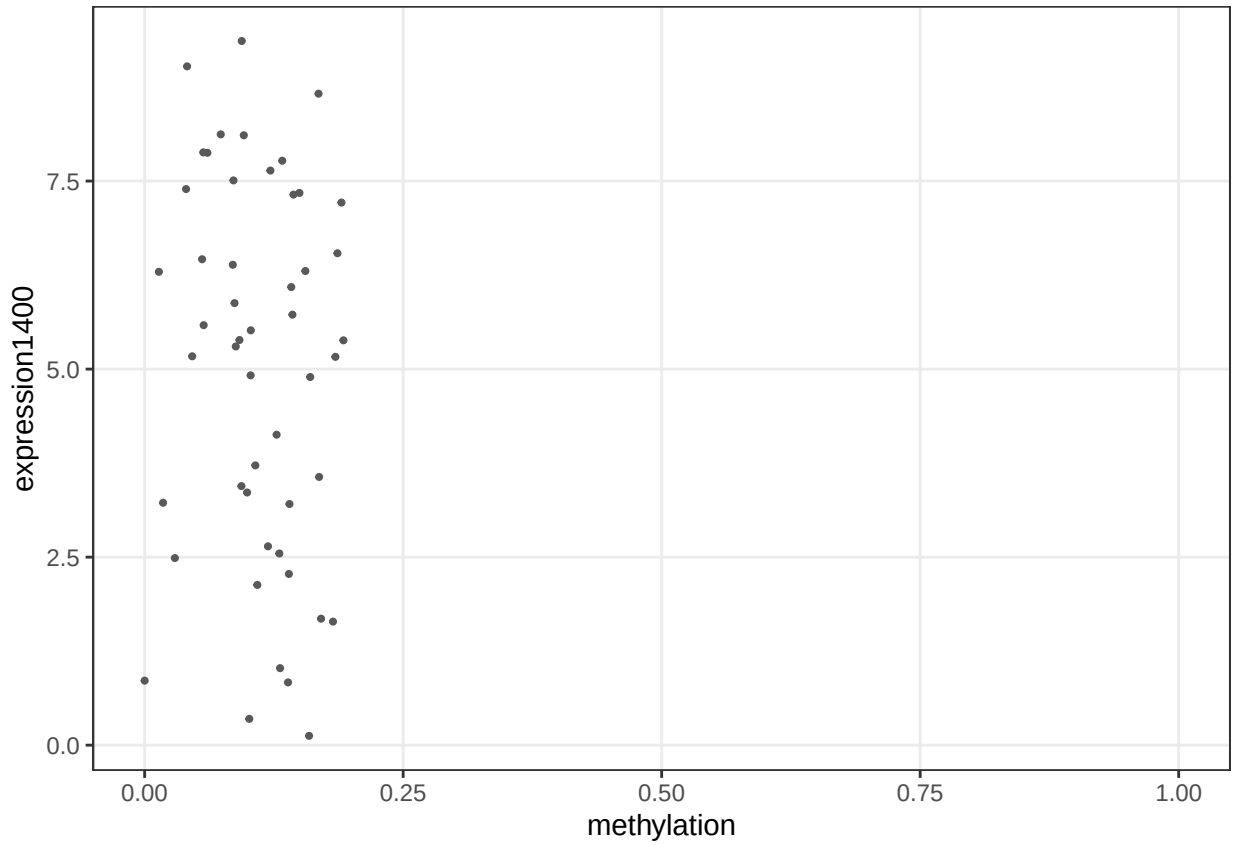



```
colNames3a <- names(dta3x)[3:502]
```

Non L shaped genes: vertical distribution left

```
#creating 250 non L shaped variables with 50 data points - vertical distribution left


ddef1 <- defData(varname = "methylation", formula = "0;0.2", dist = "uniform")
theta7 = c(0.1, 0.5, 1)
knots3 <- c(0.1)
dtac <- genData(50, ddef1)

for(i in 1301:1550){
dta4 <- genSpline(dt = dtac, newvar = paste0("expression",i),
                predictor = "methylation", theta = theta7,
```

```
                knots = knots3, degree = 1,
                newrange = "1;9",
                noise.var = 5)
}

dta4x <- abs(dta4)

#variable visualization
ggplot(data = dta4x, aes(x=methylation, y=expression1400)) +
  geom_point(color = "grey35", size = 0.75) + xlim(0,1)+
  #geom_smooth(se=FALSE, color="blue", size = 1, method = "auto") +
  theme_bw() + theme(panel.grid.minor = element_blank())
```

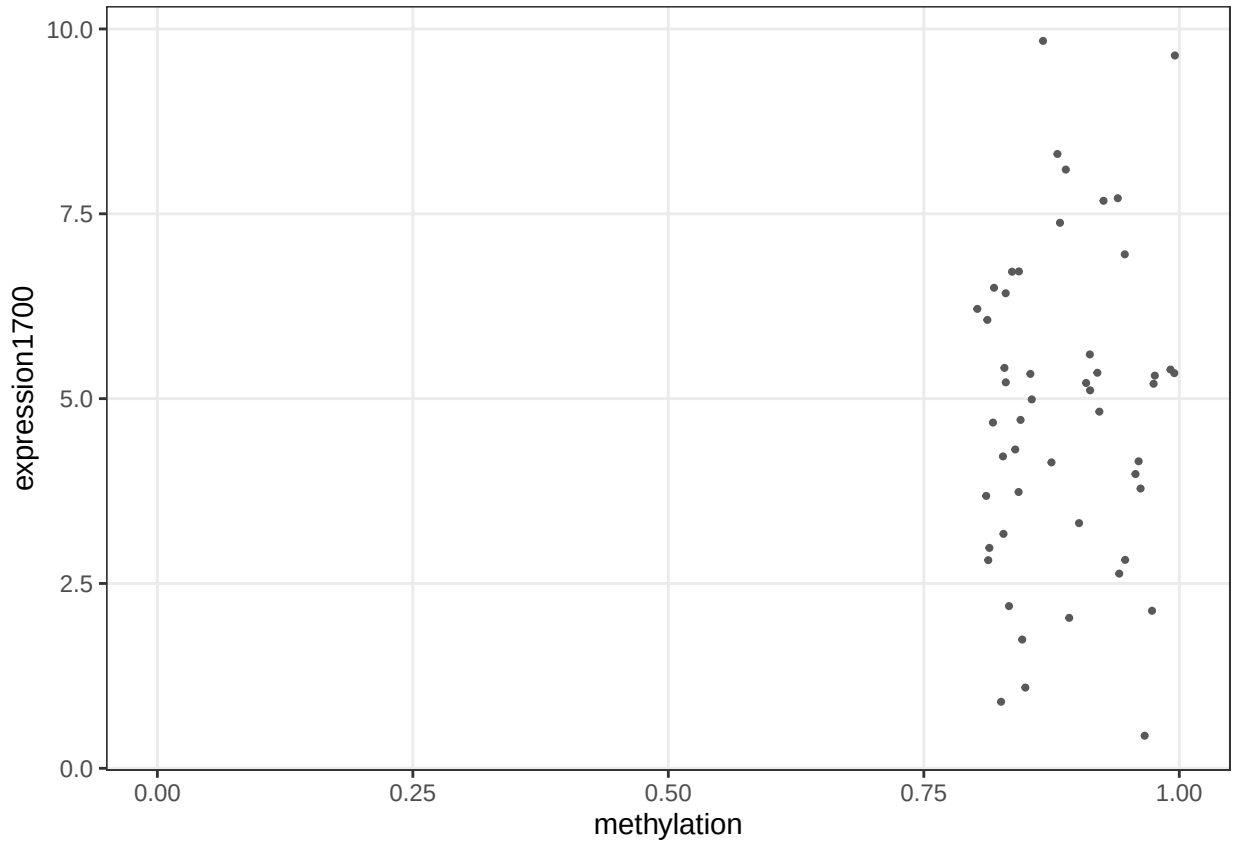



```
colNames4a <- names(dta4x)[3:252]
```

Non L shaped genes: vertical distribution right

```
#creating 250 non L shaped variables with 50 data points - vertical distribution right


ddef2 <- defData(varname = "methylation", formula = "0.8;1", dist = "uniform")
theta8 = c(0.1, 0.5, 1)
knots4 <- c(0.9)
dtav <- genData(50, ddef2)

for(i in 1551:1800){
dta5 <- genSpline(dt = dtav, newvar = paste0("expression",i),
```

```
                predictor = "methylation", theta = theta8,
                knots = knots4, degree = 1,
                newrange = "1;9",
                noise.var = 5)
}

dta5x <- abs(dta5)

#variable visualization
ggplot(data = dta5x, aes(x=methylation, y=expression1700)) +
  geom_point(color = "grey35", size = 0.75) + xlim(0,1)+
  #geom_smooth(se=FALSE, color="blue", size = 1, method = "auto") +
  theme_bw() + theme(panel.grid.minor = element_blank())
```




```
colNames5a <- names(dta5x)[3:252]
```

After generating the L and non-L genes, we format the dataset according to the requirements of the pipeline. We transpose the variables and create a "methylation dataset" by multiplying the variable X.

Creation of 2 datasets with the combinations: 1% GRM, 99% GNRM and 10% GRM, 90% GNRM

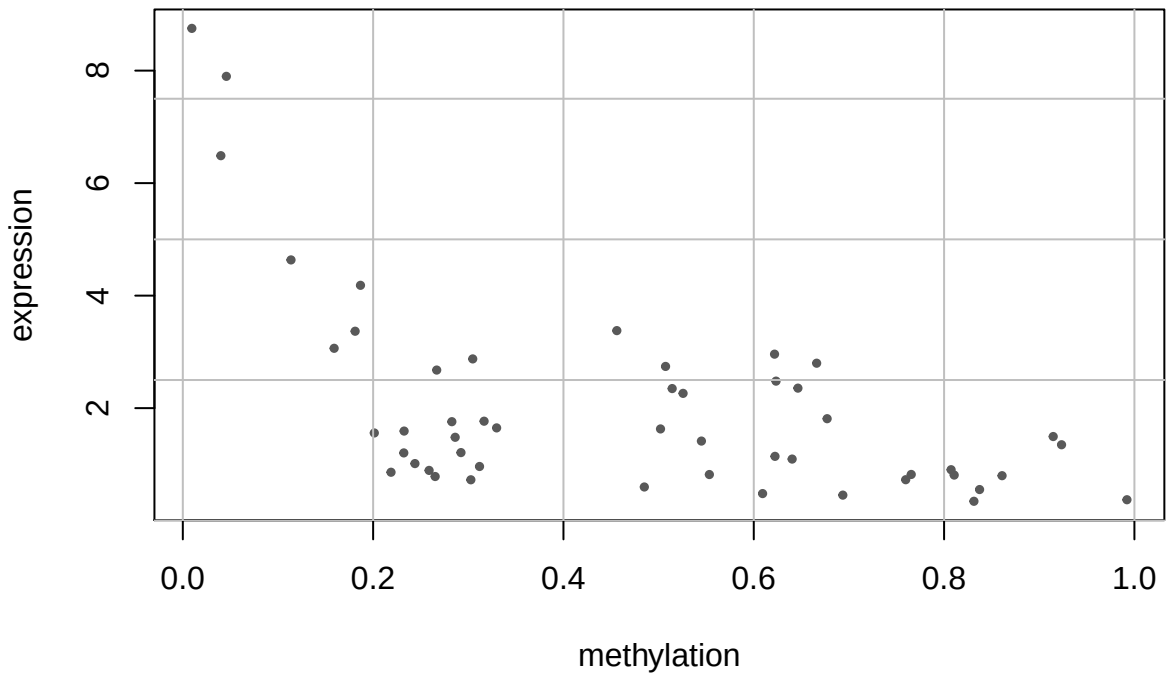
The 2 datasets will have 1000 genes in total.

Example code to create a dataset with 10% genes following an L-shape and 90% a non-L shape, from the artificial data generated above. We selected a most equal number of genes from each of the gene distribution

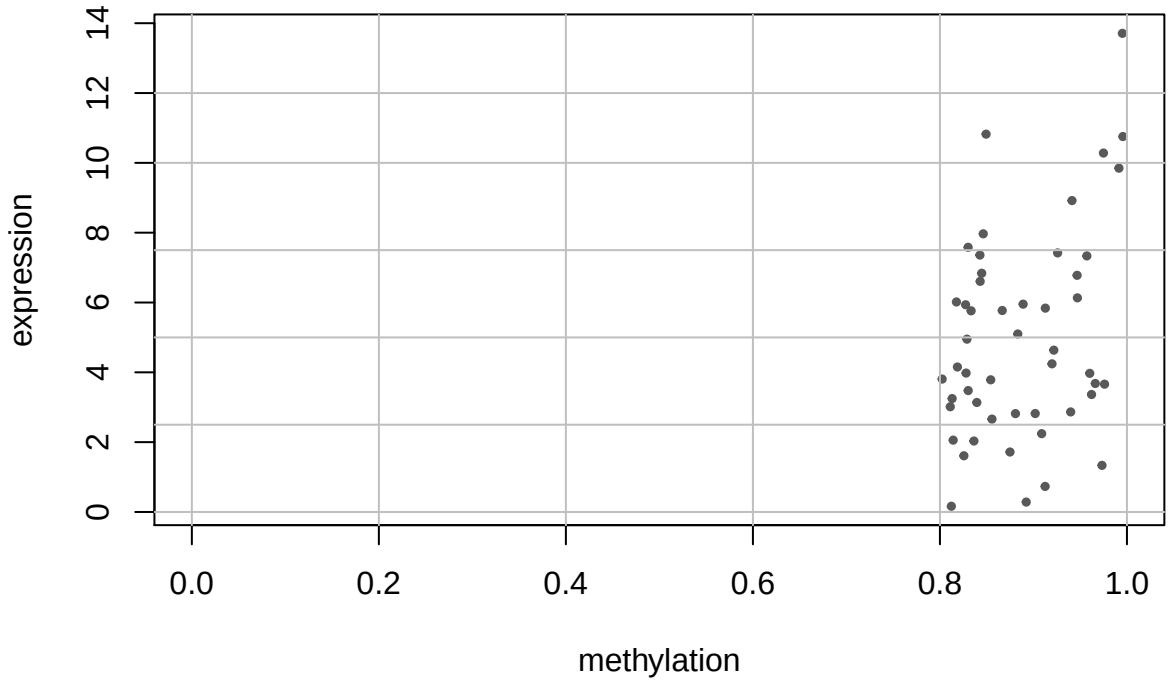
categories. For example, to obtain a 10% of GR M with an L-shape in a sample of 1000, we selected 35 random genes from the L distribution group, 35 random genes from the concave distribution and 30 from the negative correlation group to add up to 100 (the 10% corresponding to a total sample of 1000 genes). From the non L-shape distribution, we selected 150 genes from each category, except for the randomly distributed genes, for which we selected 300.

```
rep.col<-function(x,n){
   matrix(rep(x,each=n), ncol=n, byrow=TRUE)
}

#first prepare 2 datasets one for methylation and one for expression according to the genes format- lon
### 10% of GRM
dt1xtx <- t(rep.col(dt1x$methylation, 35))
dtt1x <- t(dt1x[,3:102])
dx1x <- dtt1x[sample(nrow(dtt1x), 35),]

dt2xtx <- t(rep.col(dt2x$methylation, 35))
dtt2x <- t(dt2x[,3:102])
dx2x <- dtt2x[sample(nrow(dtt2x), 35),]

dt3xtx <- t(rep.col(dt3x$methylation, 30))
dtt3x <- t(dt3x[,3:102])
dx3x <- dtt3x[sample(nrow(dtt3x), 30),]

metL10 <- rbind(rbind(dt1xtx, dt2xtx), dt3xtx)
exprL10 <- rbind(rbind(dx1x, dx2x), dx3x)

### 90% GNRM

dt1axtx <- t(rep.col(dta1x$methylation, 150))
dtt1ax <- t(dta1x[,3:252])
dx1ax <- dtt1ax[sample(nrow(dtt1ax), 150),]

dt2axtx <- t(rep.col(dta2x$methylation, 150))
dtt2ax <- t(dta2x[,3:252])
dx2ax <- dtt2ax[sample(nrow(dtt2ax), 150),]

dt3axtx <- t(rep.col(dta3x$methylation, 300))
dtt3ax <- t(dta3x[,3:502])
dx3ax <- dtt3ax[sample(nrow(dtt3ax), 300),]

dt4axtx <- t(rep.col(dta4x$methylation, 150))
dtt4ax <- t(dta4x[,3:252])
dx4ax <- dtt4ax[sample(nrow(dtt4ax), 150),]

dt5axtx <- t(rep.col(dta5x$methylation, 150))
dtt5ax <- t(dta5x[,3:252])
dx5ax <- dtt5ax[sample(nrow(dtt5ax), 150),]

metN10 <- rbind(rbind(rbind(rbind(dt1axtx, dt2axtx), dt3axtx),dt4axtx), dt5axtx)
exprN10 <- rbind(rbind(rbind(rbind(dx1ax, dx2ax), dx3ax), dx4ax), dx5ax)


#create one uniform dataset with the combination of L-shaped and non L-shaped genes.
```

```r
# met1 <- rbind(metL1, metN1)
# expr1 <- rbind(exprL1, exprN1)

met10 <- rbind(metL10, metN10)
expr10 <- rbind(exprL10, exprN10)


#visualization examples

# plot(met1[2,], expr1[2,])
# plot(met1[820,], expr1[820,])



plot(met10[2,], expr10[2,], pch= 20, col= "gray35", cex= 0.75, xlab= "methylation", ylab= "expression")
abline(v=(c(0,0.2,0.4, 0.6, 0.8,1)), col="gray75")
abline(h=(c(0,2.5, 5,7.5,10)), col="gray75")
```




```r
plot(met10[920,], expr10[920,], pch= 20, col= "gray35", cex= 0.75, xlab= "methylation", ylab= "expressi
abline(v=(c(0,0.2,0.4, 0.6, 0.8,1)), col="gray75")
abline(h=(c(0,2.5, 5,7.5,10, 12)), col="gray75")
```




```r
# rownames(met1) <- rownames(expr1)
# colnames(expr1) <- paste0("s", 1:50)
# colnames(met1) <- colnames(expr1)

rownames(met10) <- rownames(expr10)
colnames(expr10) <- paste0("s", 1:50)
colnames(met10) <- colnames(expr10)


# write.csv(met10, "methylation10_simul.csv")
# write.csv(expr10, "expression10_simul.csv")
```