Demo

```r
library(rnaGinesis)
library(ggplot2)
library(reshape)
library(GGally)
library(NMF)
```

```
## Loading required package: pkgmaker
```

```
## Loading required package: registry
```

```
##
## Attaching package: 'pkgmaker'
```

```
## The following object is masked from 'package:base':
##
##     isNamespaceLoaded
```

```
## Loading required package: rngtools
```

```
## Loading required package: cluster
```

```
## NMF - BioConductor layer [OK] | Shared memory capabilities [NO: bigmemory] | Cores 23/24
```

```
##   To enable shared memory capabilities, try: install.extras('
## NMF
## ')
```

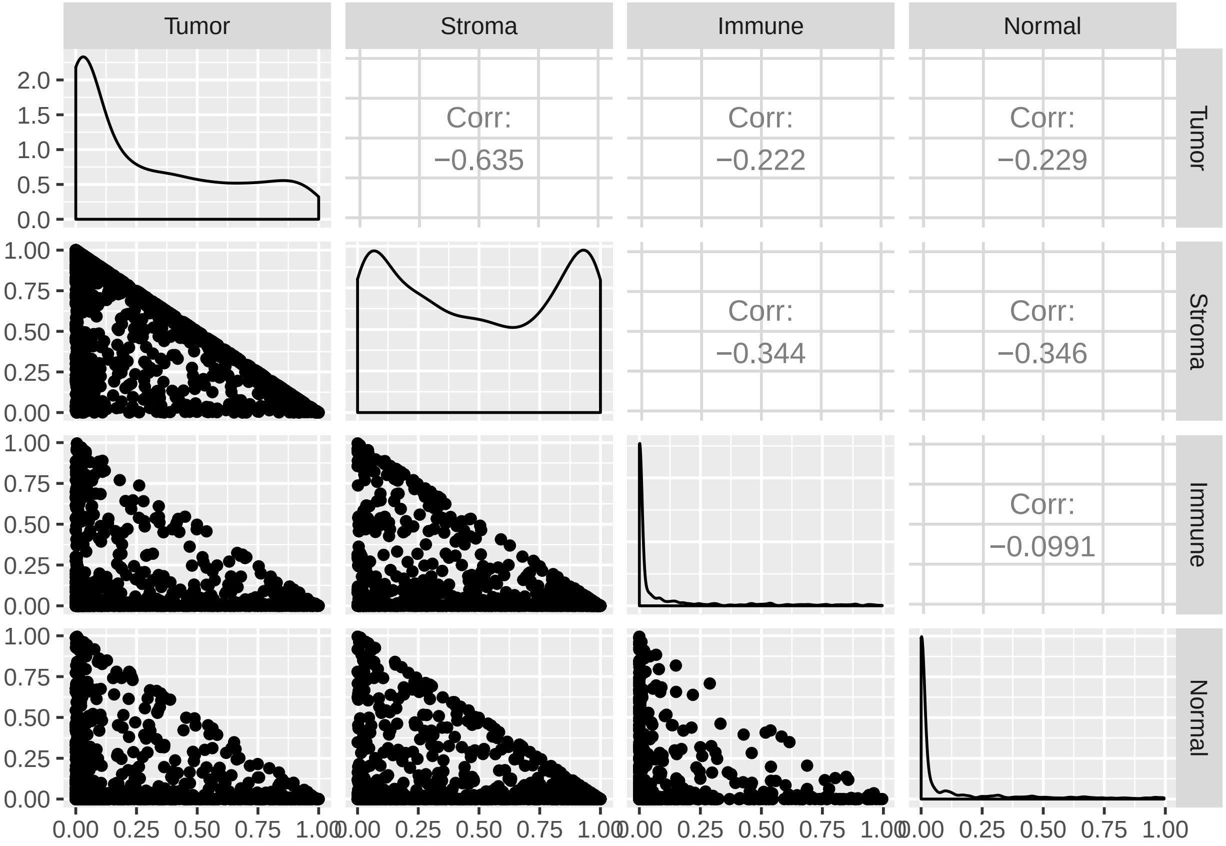
```r
library(nnls)
mu <- rnaGinesis::mu
A <- rnaGinesis::A
```

generate data

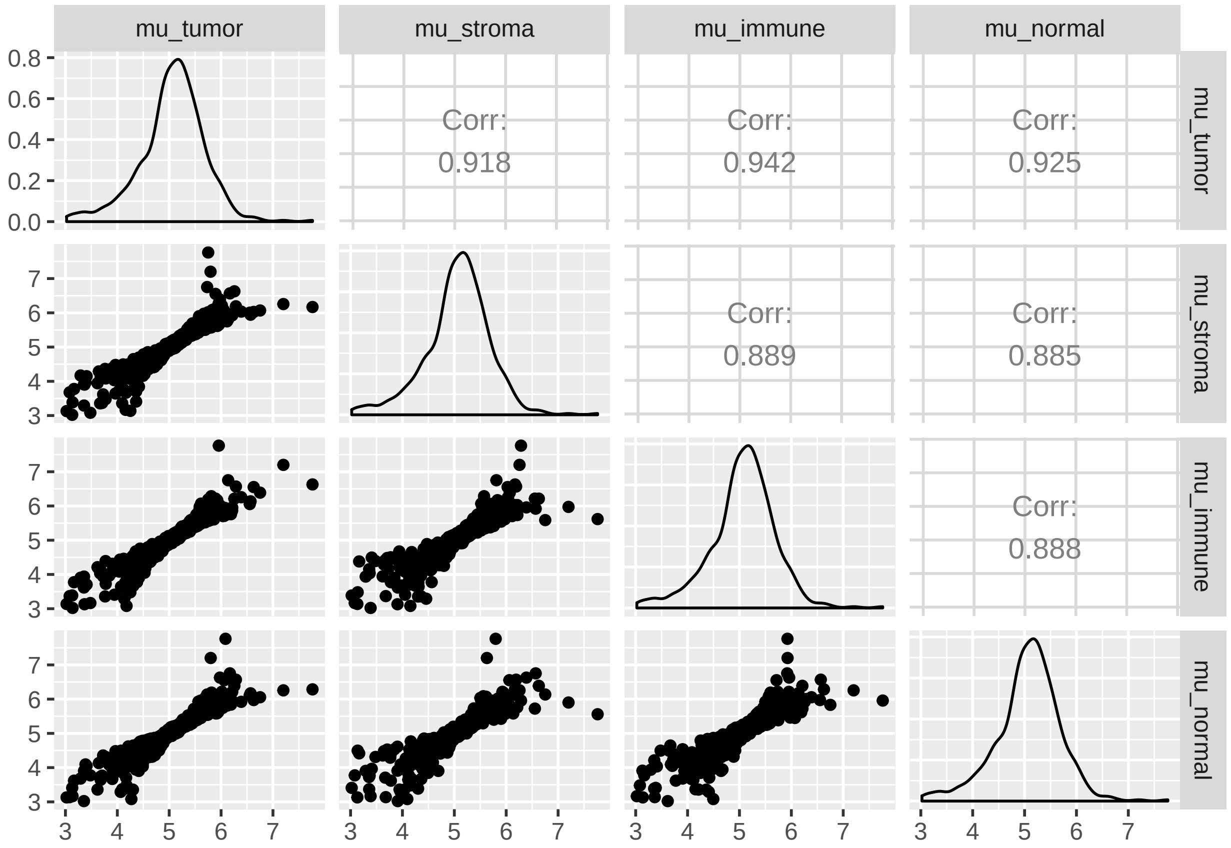
```r
ngenes <- 500;
nsamples = 1000;
simresult <- Complete_simulation(A_tumor   = A[1:ngenes,1:ngenes],
                                 mu_tumor = mu[1:ngenes],
                                 Samplesize     = nsamples,
                                 scaleFactor    = 100,
                                 d.params       = c("Tumor"    = .3,
                                                   "Stroma" = .5,
                                                   "Immune"  = .1,
                                                   "Normal"  = .1),
                                 noise_setting = 1.5,
                                 seed           = 1234)
data   <- simresult[[1]]
W      <- simresult[[2]]
H      <- simresult[[3]]
rm("simresult")
rm("mu")
rm("A")
```

Display data

```r
ggpairs(data = data.frame(t(H)))
```



```r
ggpairs(data = as.data.frame(lapply(data.frame(W),log10)))
```



Plain Residuals

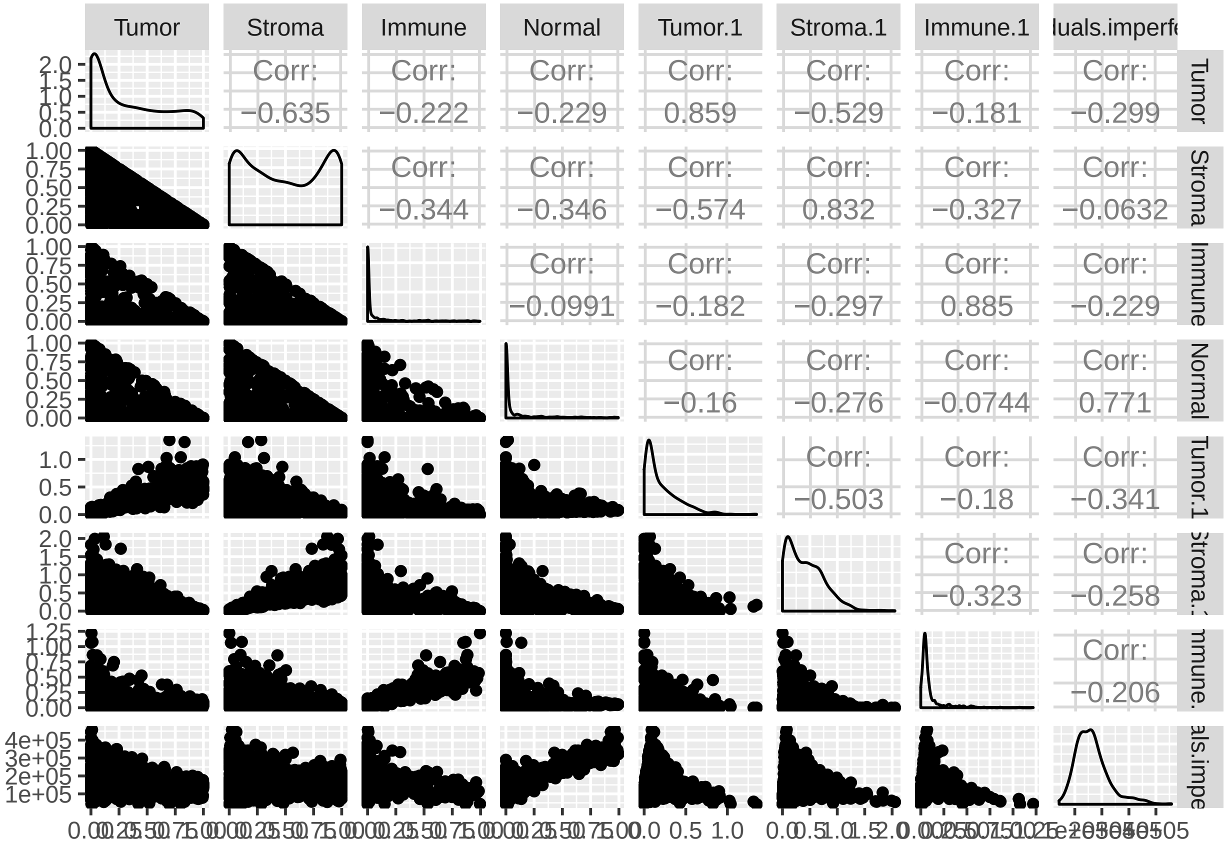
perfect knowledge

```
known.set <- 1:4
nnls.H <- H[known.set,]
nnls.resids.1 <- numeric(nsamples)
nnls.resids.2 <- numeric(nsamples)

for(i in 1:nsamples){
  this_sample <- data[,i]
  nnls.result <- nnls(A = W[,known.set],b = this_sample)
  nnls.H[known.set,i] <- nnls.result$x
  nnls.resids.1[i] <- sum((nnls.result$residuals[nnls.result$residuals>0]))

}
residuals.perfect.1 <- nnls.resids.1/ngenes
```

imperfect knowledge

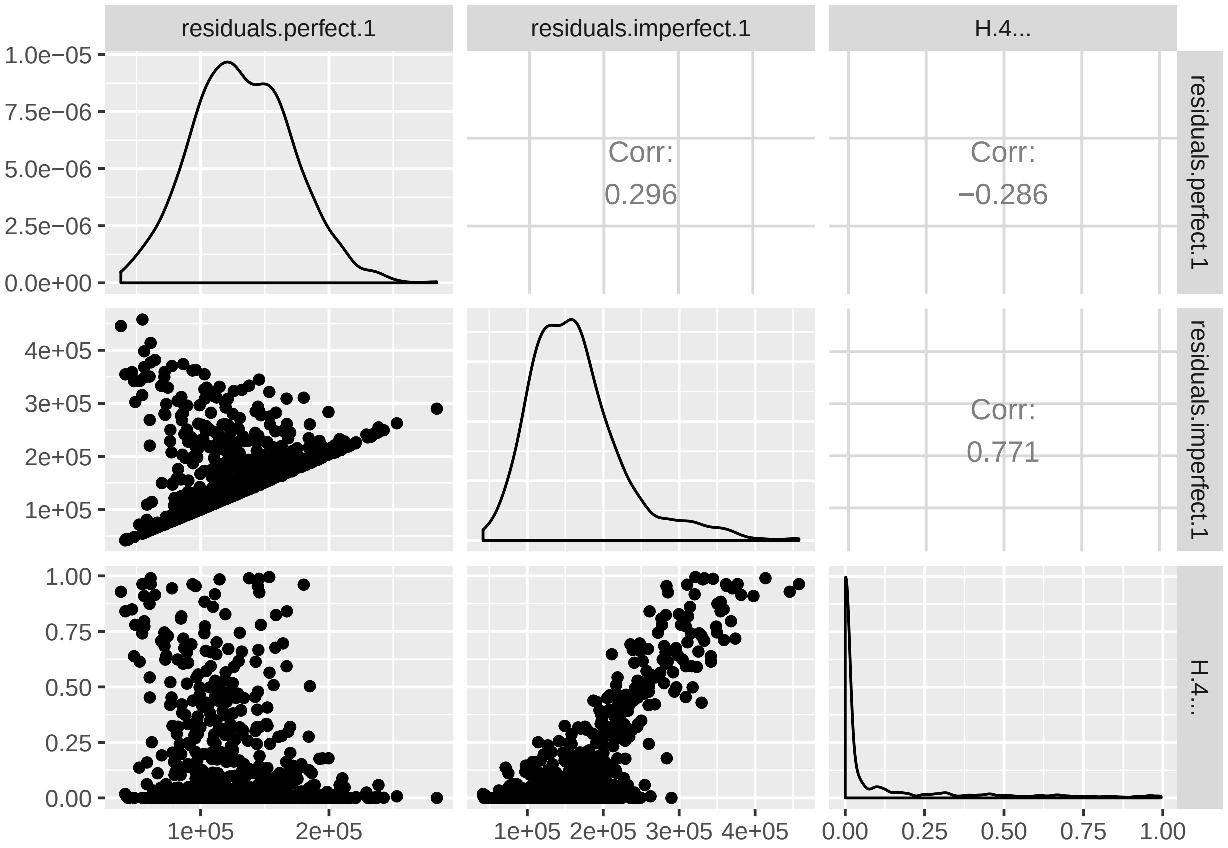
```
known.set <- 1:3
nnls.H <- H[known.set,]
nnls.resids.1 <- numeric(nsamples)
nnls.resids.2 <- numeric(nsamples)

for(i in 1:nsamples){
  this_sample <- data[,i]
  nnls.result <- nnls(A = W[,known.set],b = this_sample)
  nnls.H[known.set,i] <- nnls.result$x
  nnls.resids.1[i] <- sum((nnls.result$residuals[nnls.result$residuals>0]))
}
residuals.imperfect.1 <- nnls.resids.1/ngenes
```

```
ggpairs(data = data.frame(t(rbind(H,nnls.H,residuals.imperfect.1))))
```



```
ggpairs(data = data.frame(residuals.perfect.1,
                          residuals.imperfect.1,
                          H[4,]),alpha = 0.2)
```

```
## Warning in warn_if_args_exist(list(...)): Extra arguments: 'alpha' are
## being ignored. If these are meant to be aesthetics, submit them using the
## 'mapping' variable within ggpairs with ggplot2::aes or ggplot2::aes_string.
```


residuals.perfect.1
residuals.imperfect.1
H.4...
1.0e−05
7.5e−06
5.0e−06
2.5e−06
0.0e+00
Corr:
0.296
Corr:
−0.286
residuals.perfect.1
4e+05
3e+05
2e+05
1e+05
Corr:
0.771
residuals.imperfect.1
1.00
0.75
0.50
0.25
0.00
H.4...
1e+05
2e+05
1e+05
2e+05
3e+05
4e+05
0.00
0.25
0.50
0.75
1.00