

Hydrothermally processed 1D hydroxyapatite: mechanism of formation and biocompatibility studies

Zoran S. Stojanović¹, Nenad Ignjatović¹, Victoria Wu², Vojka Žunič³, Ljiljana Veselinović¹, Srečo Škapin³, Miroslav Miljković⁴, Vuk Uskoković², Dragan Uskoković¹

¹ Centre for Fine Particles Processing and Nanotechnologies, Institute of Technical Sciences of the Serbian Academy of Sciences and Arts, Knez Mihailova 35/4, 11000 Belgrade, Serbia

²Advanced Materials and Bionanotechnology Laboratory, Department of Bioengineering, University of Illinois, 851 South Morgan Street, Chicago, IL 60607-7052, USA

³ Advanced Materials Department, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia

⁴Laboratory for Electron Microscopy, Faculty of Medicine University of Niš, Dr. Zoran Đinđić Boulevard 81, 18 000 Niš, Serbia

Appendix A. Supplementary data

Table A1. Crystallite sizes inof samples with respect to corresponding reflection.

sample reflection	Sample1	Sample2	Sample3	Sample4	Sample5	Sample6	Sample7	Sample8
(002)	34	42	36	33	28	41	40	39
(121)	18	13	15	15	15	31	27	33
(211)	18	36	32	30	33	31	27	33
(112)	32	31	26	28	32	33	29	35
(300)	19	26	20	20	25	31	28	34

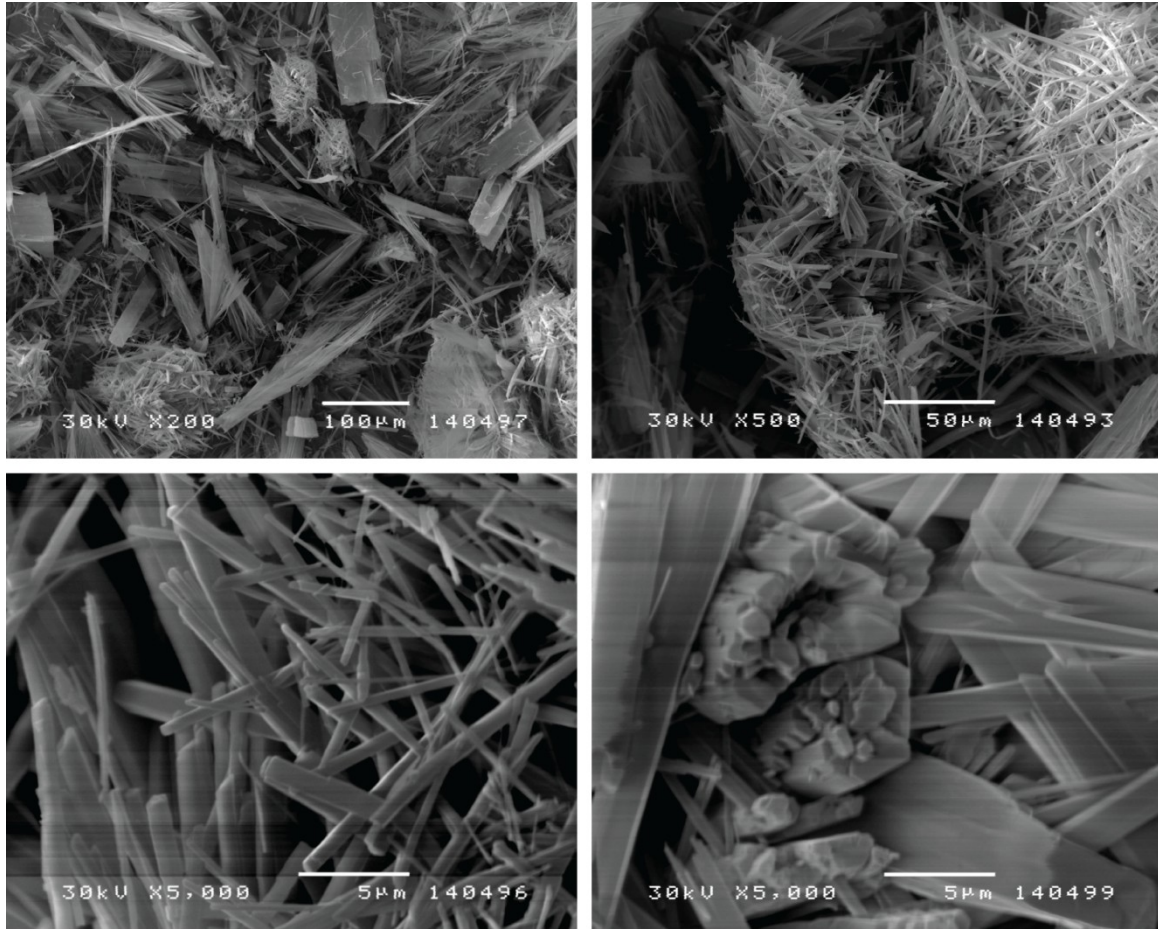


Figure A1. SEM images of Sample 2 showing bundles of HA wires at different magnification.

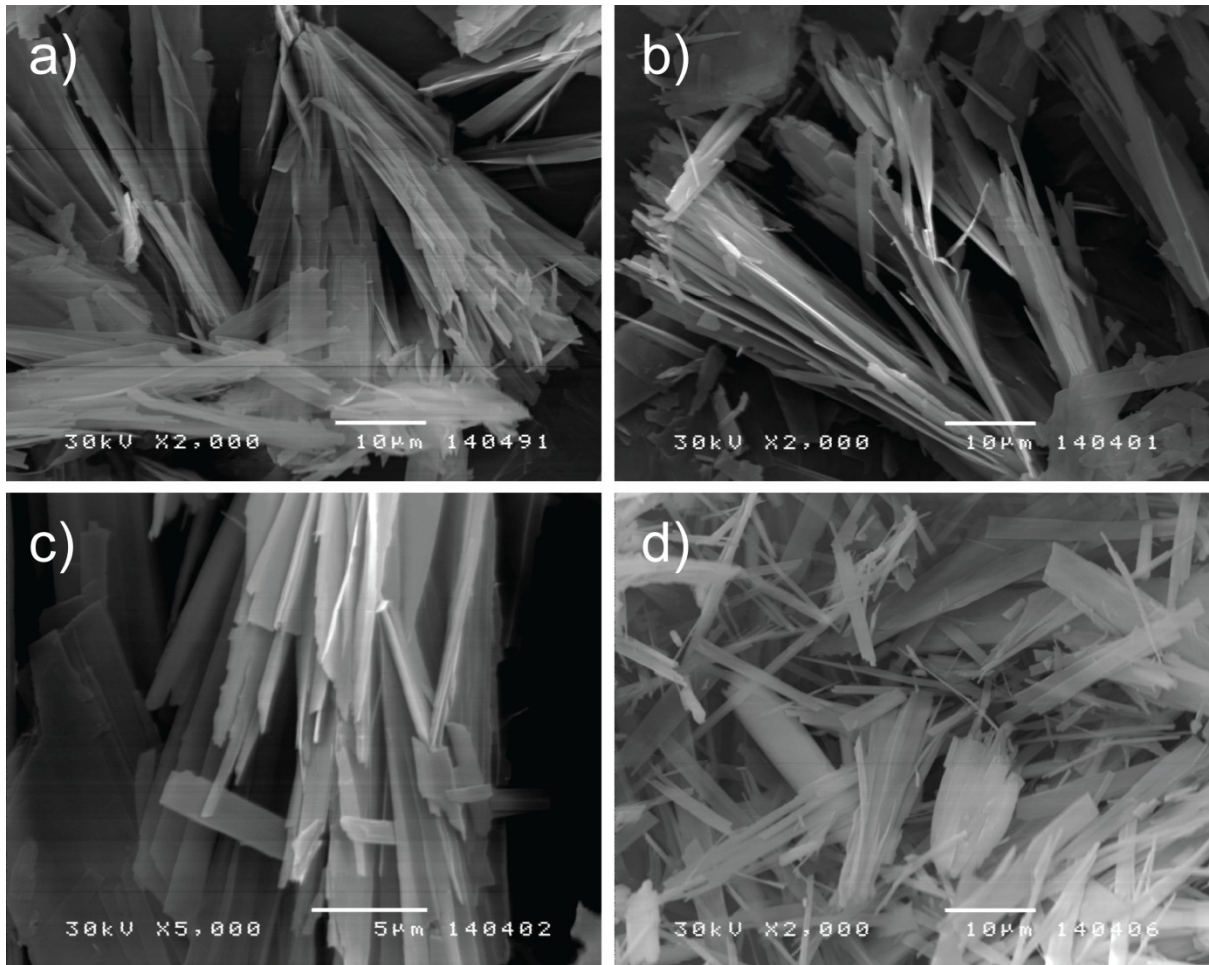


Figure A2. SEM images of samples 1 (a), 3 (b), 4 (c) and 5(d) showing bundles of HA platelets (ribbons) in all these samples. Sample 5 characterizes higher disintegration of bundles into single ribbon – like particles.

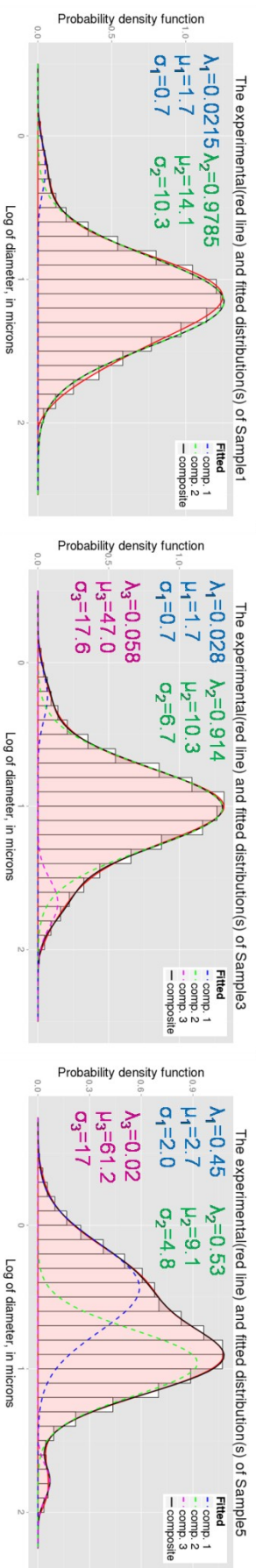
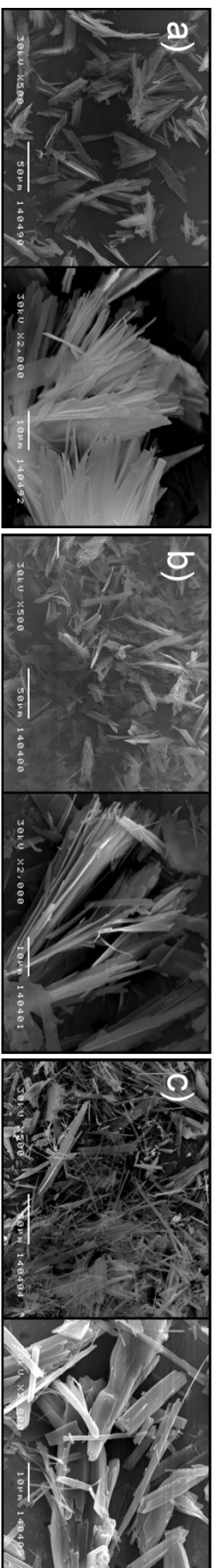


Figure A3. SEM images and size distributions of particles obtained by laser diffraction analysis in samples 1 (a), 3 (b) and 5 (c) produced using one-pot hydrothermal method.

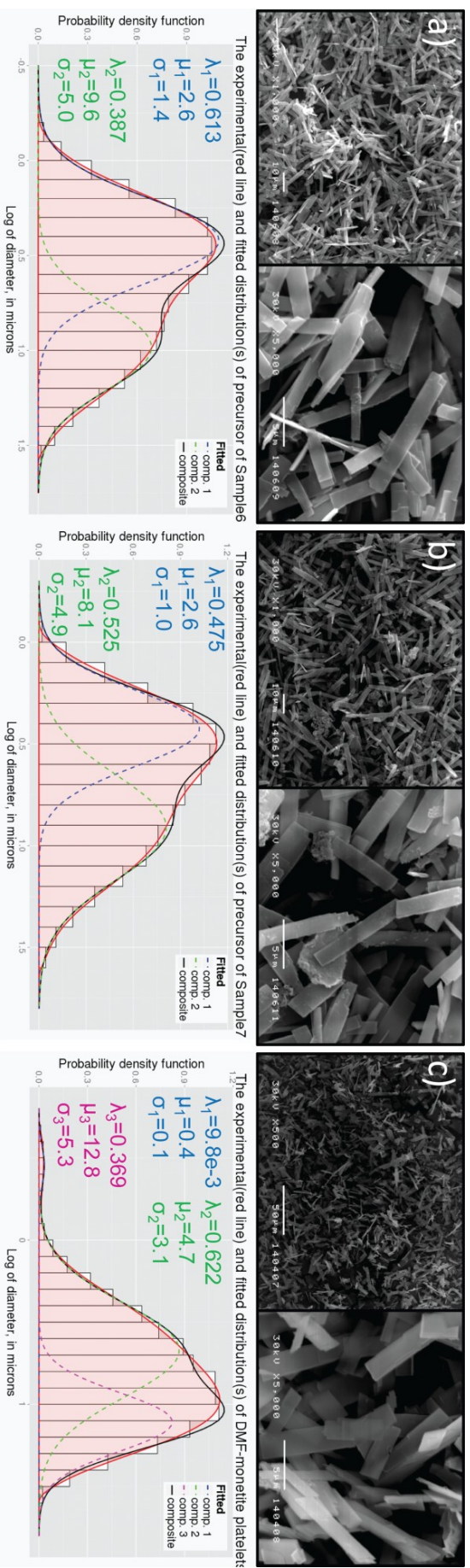


Figure A4. SEM images of platelets which corresponds to precursor of sample 6 (a), precursor of sample 7 (b), DMF-platelets (c), and summarized distribution parameters.

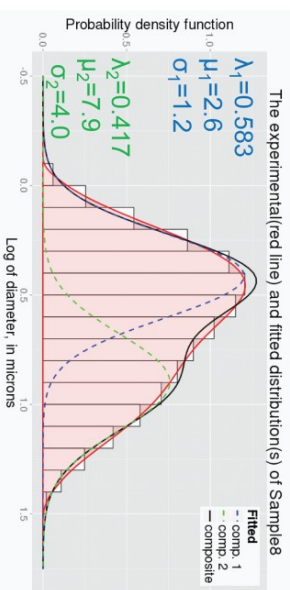
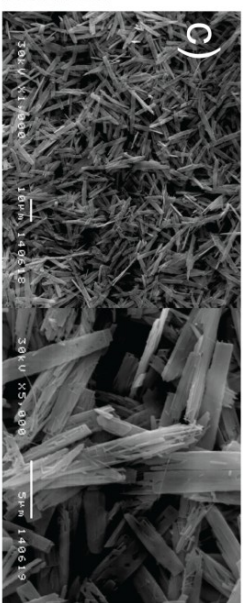
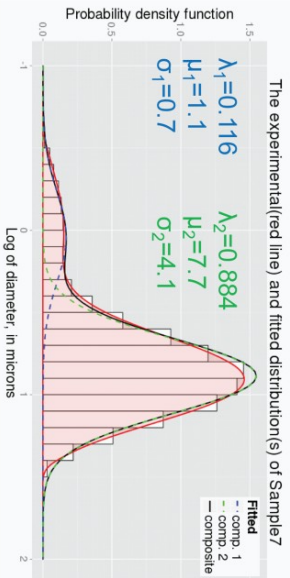
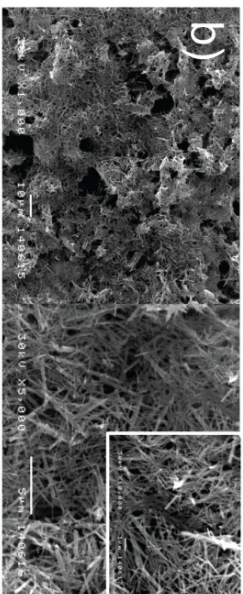
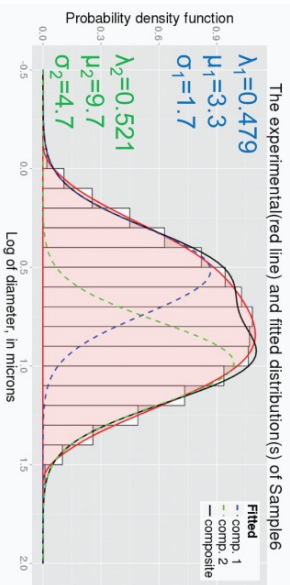
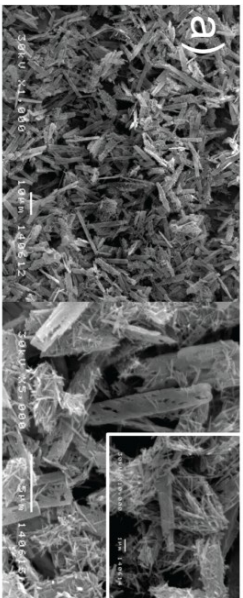


Figure A5. SEM images and particle size distributions which correspond to samples 6 (a), 7 (b) and 8 (c), and summarized distribution parameters.

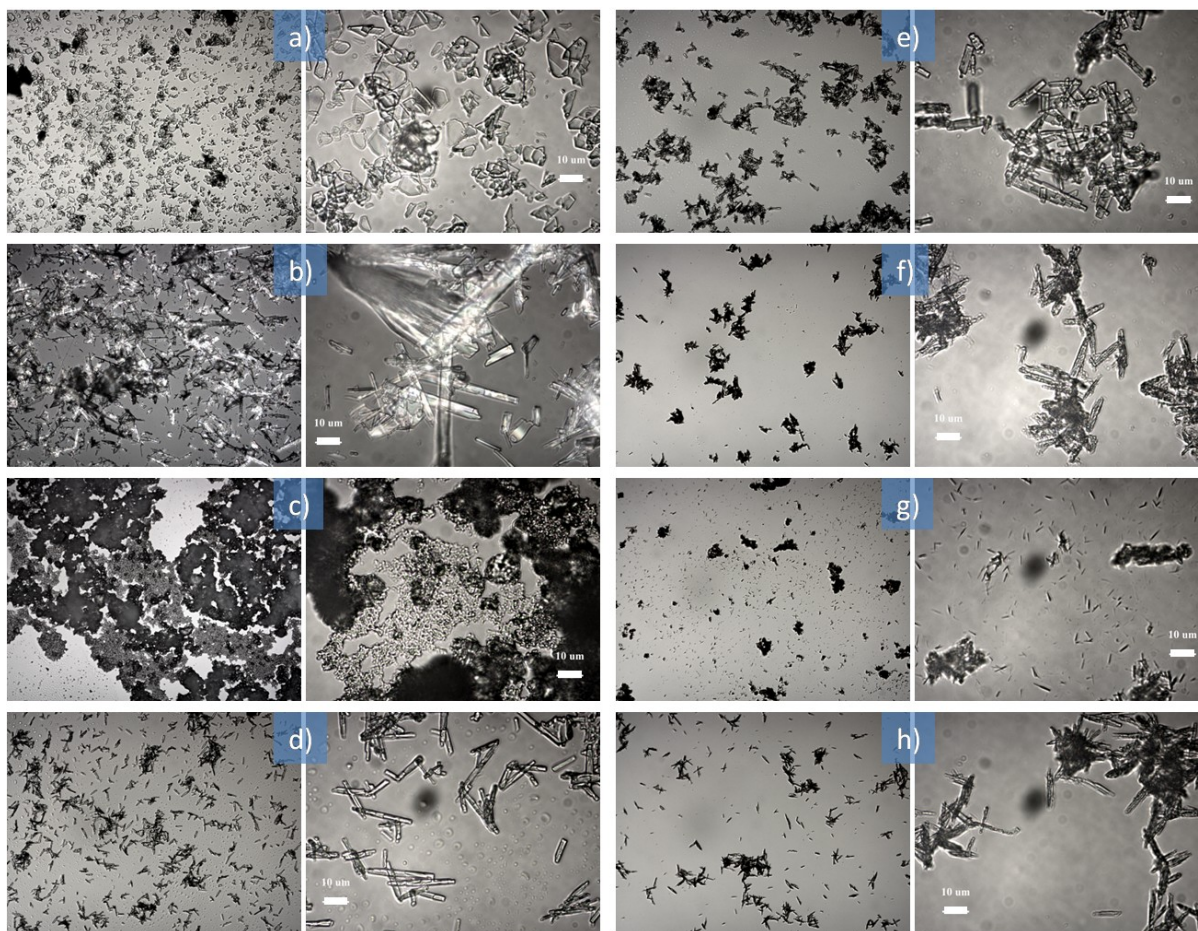


Figure A6. Images from optical microscope given in pairs – magnification 200x (without scale bar) and 1000x (scale bars 10 µm): precursor of sample 2 (a), sample 2 (b), Ca oleate complex (c), precursor of sample 6 (d), precursor of sample 7 and 8 (e), sample 6 (f), sample 7 (g) and sample 8 (h).

The code for multimodal distribution decomposition using package mixtools

The following links and documentation are used in analysis:

<https://cran.r-project.org/web/packages/mixtools/mixtools.pdf>

<http://exploringdatablog.blogspot.rs/2011/08/fitting-mixture-distributions-with-r.html>

http://rstudio-pubs-static.s3.amazonaws.com/3355_d3f08cb2f71f44f2bbec8b52f0e5b5e7.html

<http://ocw.mit.edu/courses/mathematics/18-443-statistics-for-applications-fall-2006/lecture-notes/lecture14.pdf>

The following code was used for decomposition of composite X-ray diffraction peaks:

```
setwd("~/Desktop/za_rad_HAp1D/XRD")

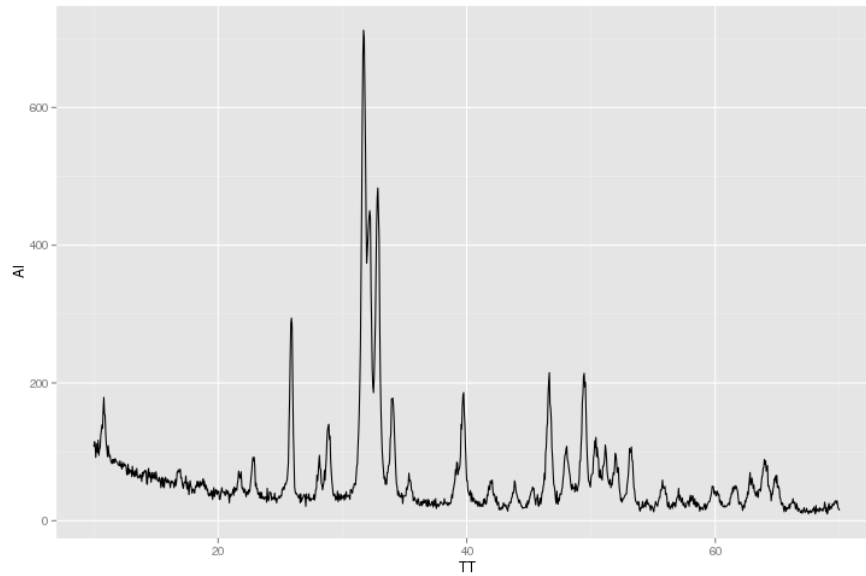
# parsing data file in .txt format and store it as data.frame object a
a <- read.delim("Sample_1.txt", header = FALSE, sep = "")
```

```
# assign names for columns TT (2 theta) and AI (arbitrary intensity)
names(a) <- c('TT','AI')
```

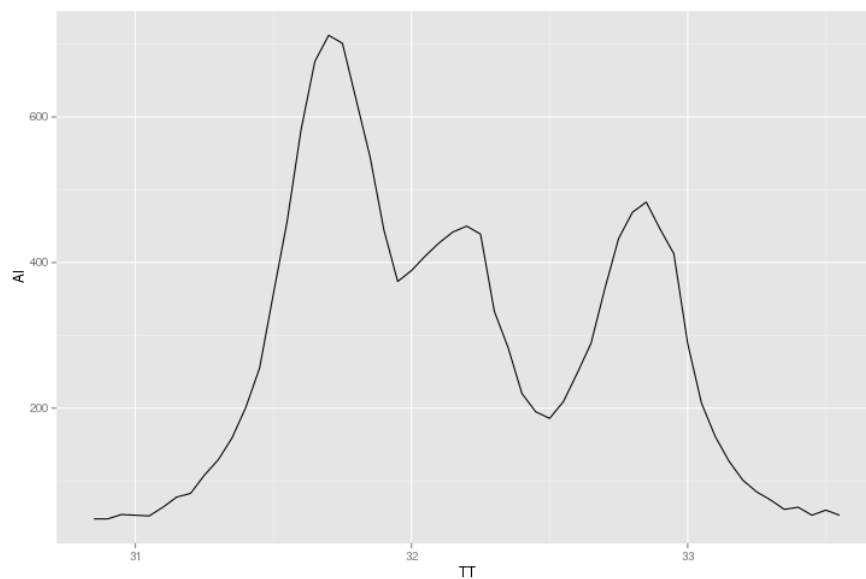
```
# loading ggplot2 package into the environment
library(ggplot2)
```

```
# creating ggplot object for whole 2 theta interval
p <- ggplot(data = a, mapping = aes(x = TT, y = AI))
```

```
# plotting with line geometry
p+layer(geom = "line")
```



```
# plotting in range of interest for distribution decomposition
p <- ggplot(data = a[a[,1]>30.8 & a[,1]<33.6,], mapping = aes(x = TT, y = AI))
p+layer(geom = "line")
```



```
# assigning an interval of values from object a to a variable copy
copy <- a[a[,1]>30.8 & a[,1]<33.6,]
names(copy) <- c('TT','AI')
```

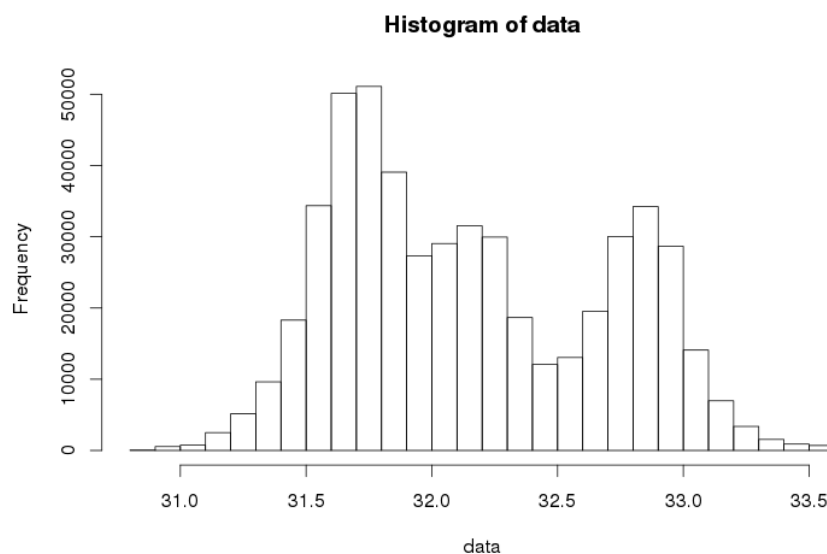


```
# do a little spline interpolation to increase the numbers of points in TT and AI
copy <- spline(copy$TT,copy$AI,n=110)
```

```
# subrtactig the background of AI vector
copy <- cbind(copy$x,copy$y-min(copy$y))
```

```
# replicate each point of 2 theta by corresponding value of intensity multiplied
# by 20 to get arbitrary vector of values corresponding
# to radiation intensity distribution
data <- rep(copy[, 1], times = copy[, 2]*20)
```

```
#plot these values as histogram
hist(data)
```



```
# loading package mixtools
library(mixtools)
```

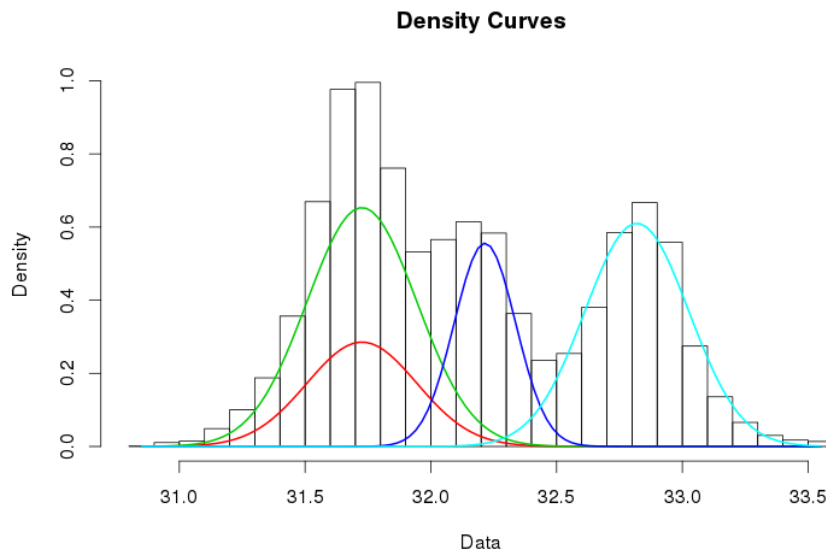
```
## Loading required package: boot
## Loading required package: MASS
## Loading required package: segmented
## mixtools package, version 1.0.3, Released 2015-04-18
## This package is based upon work supported by the National Science Foundation under Grant No. SES-0518772.
```

```
#set an initial values for reflections of HA (121), (211), (112) i (300)
mu_init <- c(31.65, 31.65, 32.15, 32.92)
sigma_init <- c(.2, .2, .2, .2)
lambda_init <- c(30.4, 69.6, 48.86, 61.32)
lambda_init <- lambda_init/sum(lambda_init)
```

```
# run normalmixEM function
mixmdl<-normalmixEM(data, lambda=lambda_init, mu=mu_init, sigma=sigma_init,
k=4, epsilon = 10^-3, verb = TRUE)
```

```
...
##      [,1] [,2] [,3] [,4]
## lambda 0.1561377 0.3574732 0.1704146 0.3159745
## mu     31.7259493 31.7259407 32.2151360 32.8190774
## sigma  0.2182761 0.2182955 0.1222912 0.2065501
## number of iterations= 82
```

```
#plotting results
plot(mixmdl,which=2)#,xlim=c(-.5,2))
```



```
#printing medians and std
mixmdl$mu
```

```
## [1] 31.72595 31.72594 32.21514 32.81908
```

```
mixmdl$sigma
```

```
## [1] 0.2182761 0.2182955 0.1222912 0.2065501
```

```
# nice plot of results with a little help of again ggplot2 using
# layer by layer plotting
```

```
p<-ggplot(data.frame())
```

```
fun1<-function(x) mixmdl$lambda[1]*dnorm(x,mixmdl$mu[1],mixmdl$sigma[1])
fun2<-function(x) mixmdl$lambda[2]*dnorm(x,mixmdl$mu[2],mixmdl$sigma[2])
fun3<-function(x) mixmdl$lambda[3]*dnorm(x,mixmdl$mu[3],mixmdl$sigma[3])
fun4<-function(x) mixmdl$lambda[4]*dnorm(x,mixmdl$mu[4],mixmdl$sigma[4])
```

```
# composite function
```

```
fun5<- function(x) mixmdl$lambda[1]*dnorm(x,mixmdl$mu[1],mixmdl$sigma[1])+
  mixmdl$lambda[2]*dnorm(x,mixmdl$mu[2],mixmdl$sigma[2])+
  mixmdl$lambda[3]*dnorm(x,mixmdl$mu[3],mixmdl$sigma[3])+
  mixmdl$lambda[4]*dnorm(x,mixmdl$mu[4],mixmdl$sigma[4])
```

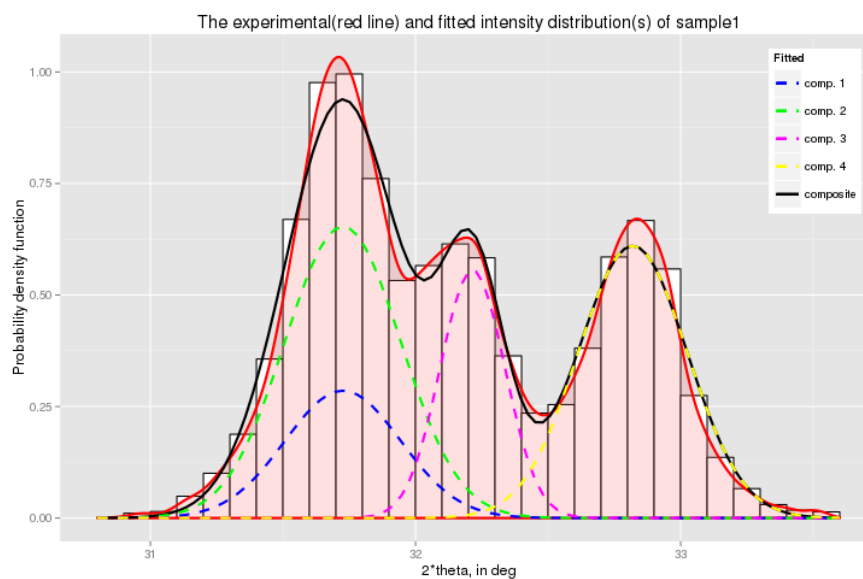
```
p+
```

```
  layer(stat="bin",geom="histogram",
    geom_params=list(fill="white",color="black"),
    stat_params=list(binwidth=.1),
    mapping=aes(x=data,y=..density..))+
  geom_density(size=1,alpha=.2, colour="red",
    fill="#FF6666",aes(x=data))+
  layer(stat="function",fun=fun5,size=1,
    mapping=aes(color="composite",linetype="composite"))+
  layer(stat="function",fun=fun1,size=1,
    mapping=aes(color="comp. 1",linetype="comp. 1"))+
  layer(stat="function",fun=fun2,size=1,
    mapping=aes(color="comp. 2",linetype="comp. 2"))+
```

```

layer(stat="function",fun=fun3,size=1,
      mapping=aes(color="comp. 3",linetype="comp. 3"))+
layer(stat="function",fun=fun4,size=1,
      mapping=aes(color="comp. 4",linetype="comp. 4"))+
scale_x_continuous(limits=c(30.8,33.6))+
scale_color_manual(name="Fitted",
                  values=c("comp. 1"="blue",
                           "comp. 2"="green", "comp. 3"="magenta",
                           "comp. 4"="yellow", "composite"="black"))+
scale_linetype_manual(name="Fitted",
                    values=c("comp. 1"=2,"comp. 2"=2,"comp. 3"=2,
                              "comp. 4"=2,"composite"=1))+
theme(legend.justification = c(1, 1), legend.position = c(1, 1))+
labs(x="2*theta, in deg",y="Probability density function",
     title="The experimental(red line) and fitted intensity distribution(s) of sample1")

```



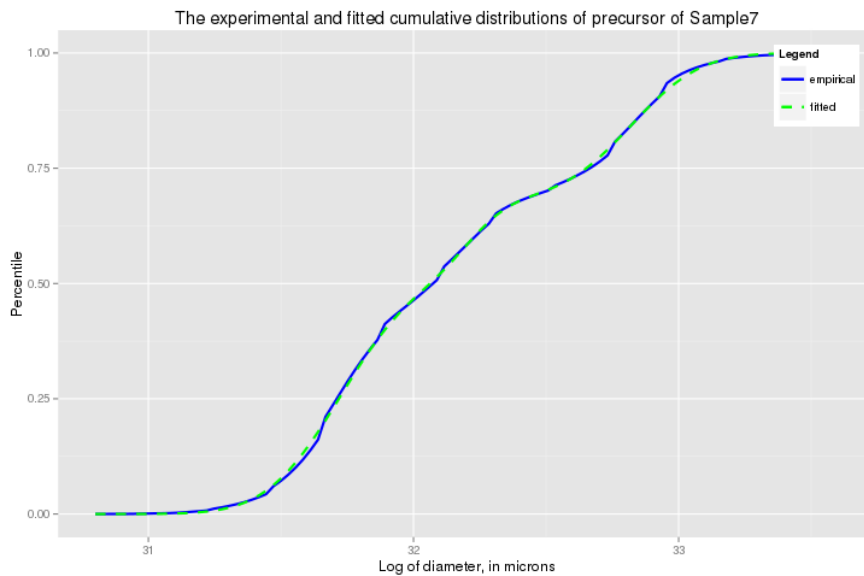
```

# creating function for cumulative distribution using pnorm()
# which uses calculated values of median, std and mixing proportions
cum_model<-function(x) mixmdl$lambda[1]*pnorm(x,mixmdl$mu[1],mixmdl$sigma[1])+
  mixmdl$lambda[2]*pnorm(x,mixmdl$mu[2],mixmdl$sigma[2])+
  mixmdl$lambda[3]*pnorm(x,mixmdl$mu[3],mixmdl$sigma[3])+
  mixmdl$lambda[4]*pnorm(x,mixmdl$mu[4],mixmdl$sigma[4])

# creating an ecdf object which computes a empiric cumulative distribution
# function
cum_exp<-ecdf(data)

# plotting empirical and model cumulative curve
np<-ggplot(data.frame())
np+
  layer(stat="function",fun=cum_exp,size=1,
        mapping=aes(color="empirical",linetype="empirical"))+
  layer(stat="function",fun=cum_model,size=1,
        mapping=aes(color="fitted",linetype="fitted"))+
scale_x_continuous(limits=c(30.8,33.6))+
scale_color_manual(name="Legend",values=c("empirical"="blue", "fitted"="green"))+
scale_linetype_manual(name="Legend",values=c("empirical"=1, "fitted"=2))+
theme(legend.justification = c(1, 1), legend.position = c(1, 1))+
labs(x="Log of diameter, in microns",y="Percentile",
     title="The experimental and fitted cumulative distributions of Sample1")

```



```
# Generating 100 random values from model distribution using sample f-on
mu<-mixmdl$mu
sigma<-mixmdl$sigma
N <- 100
components <- sample(1:length(mu),prob=mixmdl$lambda,size=N,replace=TRUE)

# creating a vector of length 100 with a values drawn from model distribution
# for reproducibility we can use set.seed() method to get always the same values
set.seed(1)
hypo<-rnorm(n=N,mean=mu[components],sd=sigma[components])

# Performing Kolmogorov-Smirnov test of random samples on hypothesis
sample<-sample(data,100)
ks.test(sample,hypo)

## Warning in ks.test(sample, hypo): p-value will be approximate in the
## presence of ties

##
## Two-sample Kolmogorov-Smirnov test
##
## data: sample and hypo
## D = 0.07, p-value = 0.9671
## alternative hypothesis: two-sided

# computing the crystallite size
K <- 1
lam <- 1.54178
beta <- mixmdl$sigma*2.355*pi/180.
mu_rad <- mixmdl$mu*pi/360.
D <- K*.154178/(beta*cos(mu_rad))
D

## [1] 17.86528 17.86369 31.92653 18.93168
```


The following code was used for decomposition of LD particle size distributions:

```
setwd("~/Desktop/RStudio_files")

# parsing data file in .txt format and store it as data.frame object a
a<-read.delim("monetit_sa_DMfom_disperzija_ORPM.txt")

# removing 0 values and NA value
# for frequency to get interval of interest
kls<-a[2,]==0|is.na(a[2,])
x<-log10(a[1,!kls]) #make log scale to be more Gaussian
y<-a[2,!kls]

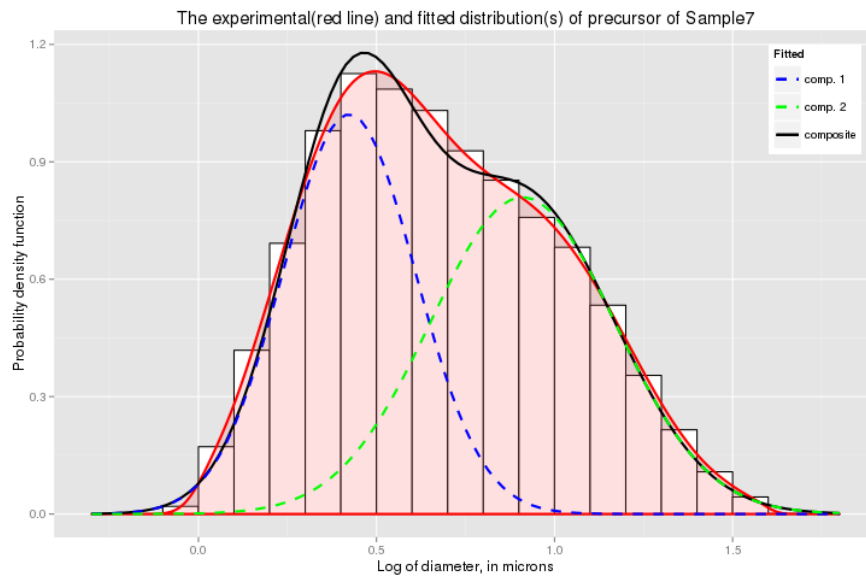
# do a spline interpolation
k<-spline(x,y,n=500)

# creating vector of values which correspond to samples from distribution
t<-round(k$y*1e2)
data<-rep(k$x,times=t)

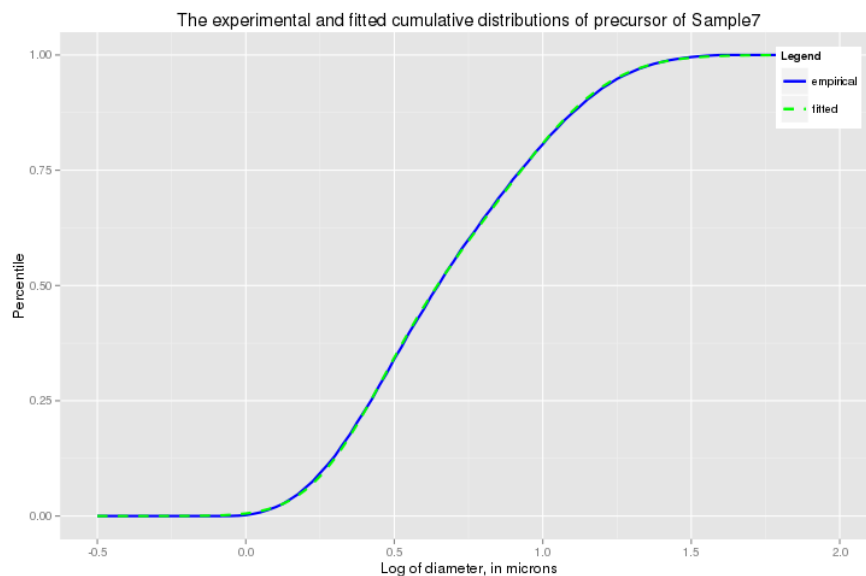
# run normalmixEM function to compute components
library(mixtools)
mixmdl<-normalmixEM(data,mu=c(.5,1),sigma=c(.25,.25),k=2)

## number of iterations= 571

#plotting results
library(ggplot2)
p<-ggplot(data=data.frame(x=data),mapping=aes(x=x))
fun1<-function(x) mixmdl$lambda[1]*dnorm(x,mixmdl$mu[1],mixmdl$sigma[1])
fun2<-function(x) mixmdl$lambda[2]*dnorm(x,mixmdl$mu[2],mixmdl$sigma[2])
fun3<-function(x) mixmdl$lambda[1]*dnorm(x,mixmdl$mu[1],mixmdl$sigma[1])+
  mixmdl$lambda[2]*dnorm(x,mixmdl$mu[2],mixmdl$sigma[2])
library(ggplot2)
p+
  layer(stat="bin",geom="histogram",
        geom_params=list(fill="white",color="black"),
        stat_params=list(binwidth=.1),
        mapping=aes(y=..density..))+
  geom_density(size=1,alpha=.2, colour="red",
              fill="#FF6666")+
  layer(stat="function",fun=fun3,size=1,mapping=aes(color="composite",linetype="composite"))+
  layer(stat="function",fun=fun1,size=1,mapping=aes(color="comp. 1",linetype="comp. 1"))+
  layer(stat="function",fun=fun2,size=1,mapping=aes(color="comp. 2",linetype="comp. 2"))+
  scale_x_continuous(limits=c(-.3,1.8))+
  scale_color_manual(name="Fitted",values=c("composite"="black","comp. 1"="blue","comp. 2"="green"))+
  scale_linetype_manual(name="Fitted",values=c("composite"=1,"comp. 1"=2,"comp. 2"=2))+
  theme(legend.justification = c(1, 1), legend.position = c(1, 1))+
  labs(x="Log of diameter, in microns",y="Probability density function",
       title="The experimental(red line) and fitted distribution(s) of precursor of Sample7")
```



```
cum_model<-function(x) mixmdl$lambda[1]*pnorm(x,mixmdl$mu[1],mixmdl$sigma[1])+
  mixmdl$lambda[2]*pnorm(x,mixmdl$mu[2],mixmdl$sigma[2])
cum_exp<-ecdf(data)
np<-ggplot(data.frame())
np+
  layer(stat="function",fun=cum_exp,size=1,
        mapping=aes(color="empirical",linetype="empirical"))+
  layer(stat="function",fun=cum_model,size=1,
        mapping=aes(color="fitted",linetype="fitted"))+
  scale_x_continuous(limits=c(-.5,2))+
  scale_color_manual(name="Legend",values=c("empirical"="blue","fitted"="green"))+
  scale_linetype_manual(name="Legend",values=c("empirical"=1,"fitted"=2))+
  theme(legend.justification = c(1, 1), legend.position = c(1, 1))+
  labs(x="Log of diameter, in microns",y="Percentile",
       title="The experimental and fitted cumulative distributions of precursor of Sample7")
```



```
#Generating 100 random values from model distribution
mu<-mixmdl$mu
sigma<-mixmdl$sigma
```

```
N <- 100
components <- sample(1:length(mu),prob=mixmdl$lambda,size=N,replace=TRUE)

#set.seed(1)
hypo<-rnorm(n=N,mean=mu[components],sd=sigma[components])

#Performing Kolmogorov-Smirnov test of random samples on hypothesis
sample<-sample(data,100)
ks.test(sample,hypo)

## Warning in ks.test(sample, hypo): p-value will be approximate in the
## presence of ties

##
## Two-sample Kolmogorov-Smirnov test
##
## data: sample and hypo
## D = 0.15, p-value = 0.2106
## alternative hypothesis: two-sided
```