metals

Particle Size Distribution Models for Metallurgical Coke Grinding Products

Laura Colorado-Arango [1], Juan M. Menéndez-Aguado [2,*] and Adriana Osorio-Correa [1]

[1] Departamento de Ingeniería Química, Universidad de Antioquia, 050010 Medellín, Colombia; laura.coloradoa@udea.edu.co (L.C.-A.); adriana.osorio@udea.edu.co (A.O.-C.)
[2] Escuela Politécnica de Mieres, Universidad de Oviedo, 33600 Oviedo, Spain; maguado@uniovi.es
* Correspondence: maguado@uniovi.es; Tel.: +34 985458033

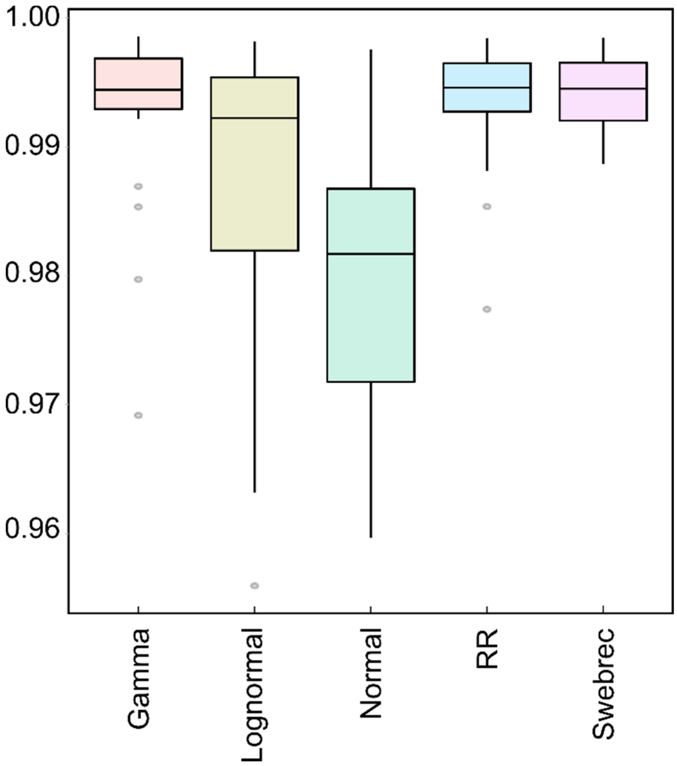

Figure S1. Box plot to compare the particle size models base on adjusted R² criterion, excluding Gates Gaudin Schuhmann distribution.

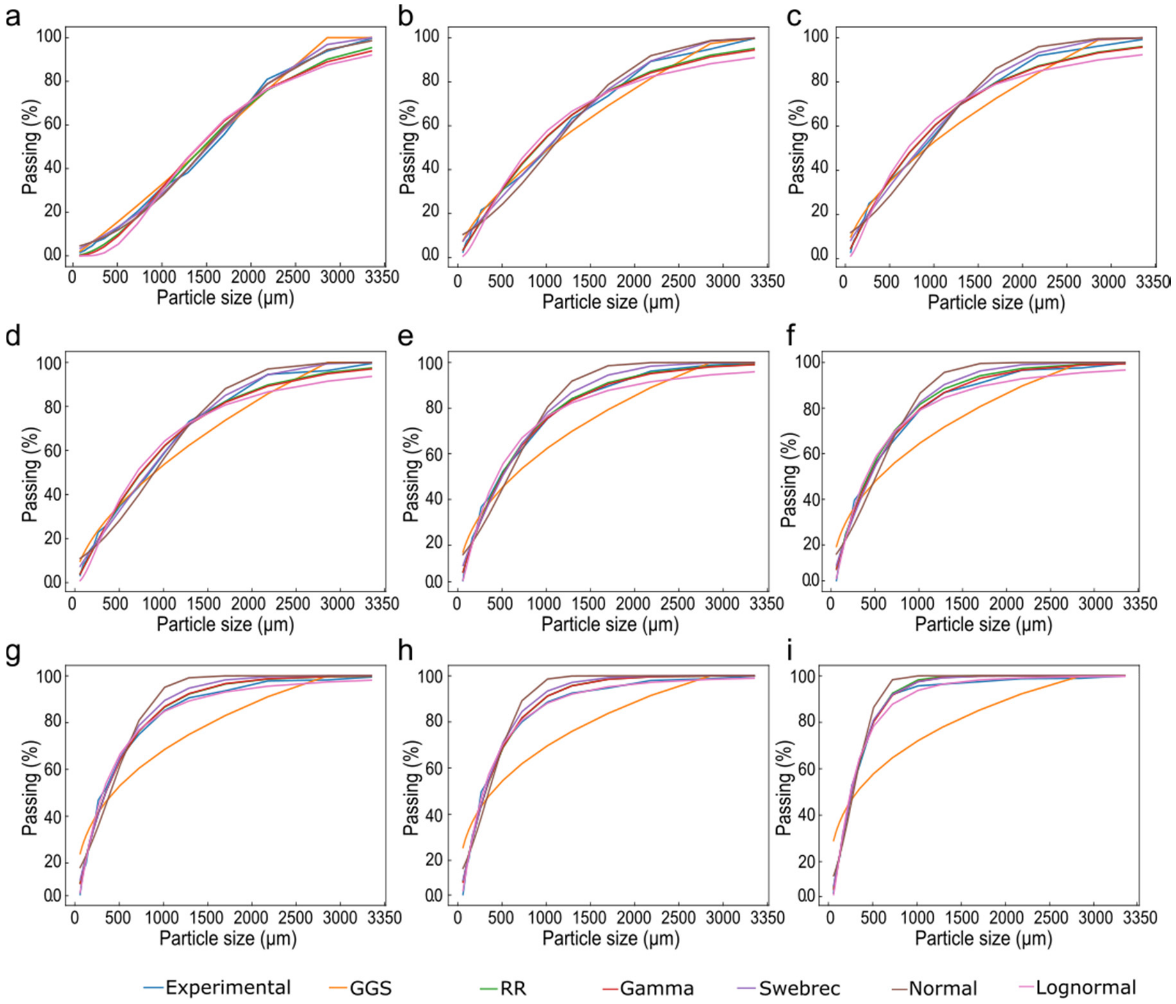

Figure S2. Cumulative distribution functions with 2.54 cm ball size, at different grinding times; **(a)** 0.0 min; **(b)** 0.5 min; **(c)** 1.0 min; **(d)** 2.0 min; **(e)** 3.0 min; **(f)** 4.0 min; **(g)** 5.0 min; **(h)** 6.0 min; **(i)**10 min.

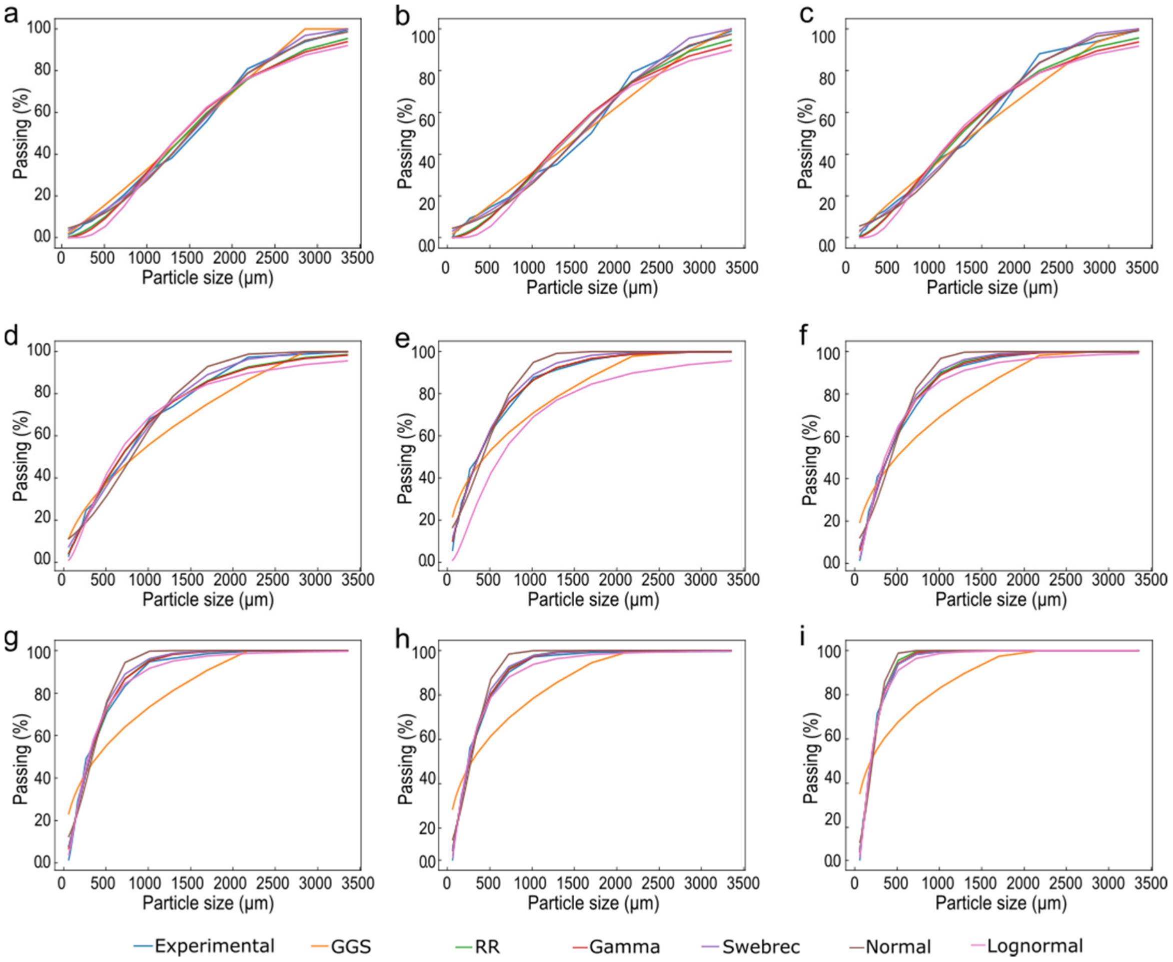

Figure S3. Cumulative distribution functions with 4.0 cm ball size, at different grinding times; **(a)** 0.0 min; **(b)** 0.5 min; **(c)** 1.0 min; **(d)** 2.0 min; **(e)** 3.0 min; **(f)** 4.0 min; **(g)** 5.0 min; **(h)** 6.0 min; **(i)**10min.

Table S1. Statistical descriptors for three criteria.

Models	**Adjusted R²**				**RMSE**				**AIC**			
	Min	**Max**	**Mean**	**S.d**	**Min**	**Max**	**Mean**	**S.d**	**Min**	**Max**	**Mean**	**S.d**
Gamma	0.969	0.999	0.993	0.00671	0.0130	0.0626	0.0294	0.0115	−127.256	−88.53	−112.33	10.82
Lognormal	0.956	0.998	0.987	0.01976	0.0160	0.0604	0.0326	0.0120	−132.787	−83.22	−104.81	15.13
Normal	0.960	0.998	0.982	0.01090	0.0253	0.0907	0.0576	0.0173	−119.028	−83.76	−94.88	9.78
RR	0.977	0.998	0.994	0.00469	0.0132	0.0444	0.0285	0.0099	−126.300	−93.51	−112.96	8.77
GGS	0.724	0.981	0.905	0.07360	0.0417	0.1779	0.1074	0.0423	−96.040	−51.89	−71.94	13.91
Swebrec	0.989	0.999	0.996	0.00246	0.0162	0.0557	0.0331	0.0116	−124.090	−99.12	−111.0	7.07

Table S2. Custom script to determine fitting parameter by least square method.

```
#Call libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

#Import dataset
xl=pd.ExcelFile('Balls_3cm.xlsx') #dataset employing ball sizes of 3cm
#
#Name x,y values
xdata=df['Diameter'] #Particle size diameter (mm)
ydata=df['M2.0']#Cumulative undersize at 2 min of grinding time(-)

#Chose a model to evaluate (Rosin Rammler case)
def model(xdata,coeffs):
        return 1-np.exp(-np.power((xdata/coeffs[0]),coeffs[1]))

#Initial guess
x0=np.array([246.73165778,4.8654162],dtype=float)

#Define the objective function
Def residuals(coeffs, y, t):
        Return (y - model(xdata, coeffs))**2

#Least squares method
from scipy.optimize import leastsq

xdata=df['Diameter']
x,flag=leastsq(residuals,x0,args=(ydata, xdata))
print(x)
```