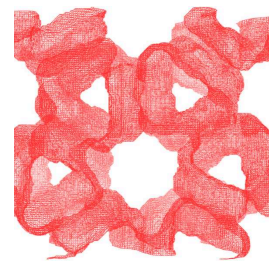


Sep 28, 2022

Calculation of MOF pore size distributions using PoreBlazer 4.0

DOI

dx.doi.org/10.17504/protocols.io.261ge671dl47/v1



Jonas Sundberg¹

¹Technical University of Denmark

Jonas Sundberg: Sensors & Functional Materials

Sensors and Functional Materials
Tech. support email: jonsun@dtu.dk



Jonas Sundberg

Technical University of Denmark

Create & collaborate more with a free account

Edit and publish protocols, collaborate in communities, share insights through comments, and track progress with run records.

Create free account

OPEN  ACCESS



DOI: <https://dx.doi.org/10.17504/protocols.io.261ge671dl47/v1>

Protocol Citation: Jonas Sundberg 2022. Calculation of MOF pore size distributions using PoreBlazer 4.0. **protocols.io**
<https://dx.doi.org/10.17504/protocols.io.261ge671dl47/v1>

License: This is an open access protocol distributed under the terms of the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: Working

We use this protocol and it's working

Created: June 17, 2019

Last Modified: September 28, 2022

Protocol Integer ID: 24606

Keywords: metal-organic frameworks, simulations, porosity, pore size distribution, MOF, pore size distribution, calculation of mof, using poreblazer, nice poreblazer software, csd mof database, mof, chem, materials informatics, organic framework, size distribution

Abstract

This is a brief step-by-step protocol for how to calculate and plot the pore-size distribution (PSD) of a metal-organic framework. The protocol uses the nice PoreBlazer software made by Sarkisov et. al., for in-depth information read their paper:

'Materials Informatics with PoreBlazer v4.0 and the CSD MOF Database'

Chem. Mater. 2020, 32, 23, 9849–9867

<https://doi.org/10.1021/acs.chemmater.0c03575>

Disclaimer

Our protocols are primarily for internal use by students, but shared openly as we believe they might be of broader interest. But remember, this is just how we do it. Not necessarily the best, or smartest, way (but hopefully somewhat correct). Code snippets are not meant to be definite, but used as learning tools and starting points to create your own scripts. If you have comments, please e-mail us freely and direct. :)

Troubleshooting

Safety warnings

- ⚠ Do not sit in front of the computer doing simulations for prolonged periods of time. Doing practical laboratory work is important for the progress of your project as well as mental health.

Before start

The protocol assumes that you have the following software packages installed and ready:

- [Mercury/Cambridge Structural Database software suite](#)
- [PoreBlazer Windows binary](#)
- [Aten Molecular modelling package](#)
- [Jupyter Notebooks/JupyterLab with functional Python environment](#)

JupyterLab with Python and Matplotlib are used for plotting the resulting data. For the less computer savvy, easiest way is to download Anaconda, and use conda to install the required packages.

Working folders

Create an appropriately (structure refcode, name or similar) named sub-folder under the directory where the '*poreblazer.exe*' executable is located. This is where you should store and save all files.

Preparation of input files

1 Crystallographic clean-up using Mercury

If your material has co-crystallized solvent molecule(s) it is important to remove before measuring the pore-size distribution. In this protocol we will work with HKUST-1 (CSD-refcode: FIQCEN) as an example. The crystal structure of HKUST-1 contains one co-crystallized water molecule per unit cell. Furthermore, each copper paddlewheel is coordinatively saturated by two water molecules.

1. Open the CIF-file (Ctrl+O) or use the '*Structure Navigator*' (right pane in default settings) to locate entry **FIQCEN**.
2. You will now see the asymmetric unit of the structure.
3. Select the free and bound water molecules (see figure below).
4. Select '*Edit>Edit Structure*'
5. Click '*Remove> Selected Atoms*'
6. Export the desolvated structure as cif by '*File>Save As*'; name the file '*FIQCEN_desolv.cif*'.

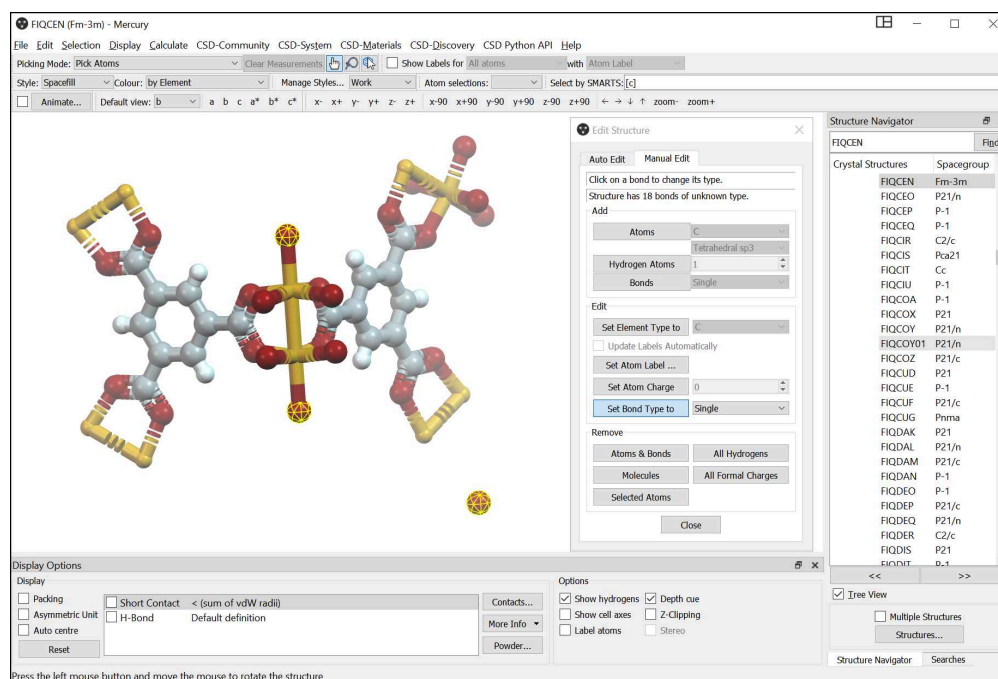



Figure 1. Screenshot of Mercury dialogue on atom removal.

 FIQCEN_desolv.cif

Software

Mercury

NAME

Windows 10

OS

CCDC

DEVELOPER

2 Use Aten to convert the CIF file to an PoreBlazer-compatible xyz input file.

Note

For to me unknown reasons, direct export of CIF to XYZ in Mercury yields an incorrect structure file and erroneous PSD data.

1. Start Aten and open the resulting CIF file by clicking the 'Open' icon.
2. Browse to your data directory, choose 'Crystallographic Information File (CIF)' from the 'Filter' dropdown menu, select your file and click 'Open'.
3. Confirm that the structure is fully desolvated.
4. Click 'Save As', ensure that 'Determine format of saved file by name' is selected, enter 'FIQCEN_desolv.xyz' as filename and click 'Save'.

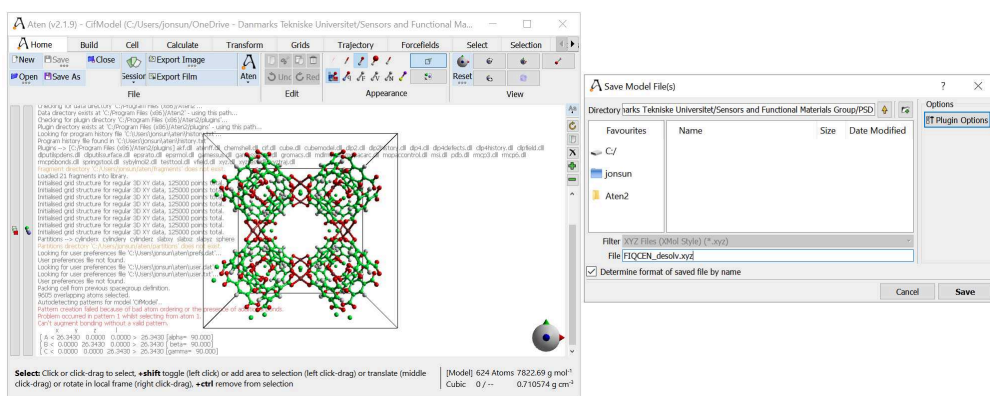



Figure 2. Interface of Aten after loading of CIF and 'Save as' dialogue.

 FIQCEN_desolv.xyz

Software

Aten

NAME

Windows 10

OS

TGA Youngs

DEVELOPER

<https://www.projectaten.com/aten>

SOURCE LINK

3 Create a PoreBlazer input files:

1. In your working directory, create and save a '*input.dat*' file in Notepad (or other text editor).
2. The first line should contain the xyz filename, e.g. '*FIQCEN_desolv.xyz*'
3. The second and third line contains unit cell dimensions (cell lengths and angles).
4. To obtain these, go back to Mercury. With your structure loaded, select '*Structure Information*' from the More Info' drop down menu (bottom pane).
5. The unit cell dimensions are provided under the '*Structure*' tab.
6. Enter the cell lengths (tab separated) on the second line.
7. Enter the cell angles (tab separated) on the third line.
8. In addition to the '*input.dat*' file, the PoreBlazer requires force field data. Default values are provided by PoreBlazer and should typically not be modified. Either copy the '*defaults.dat*' and '*UFF.dat*' files from the source distribution or use the values provided below.

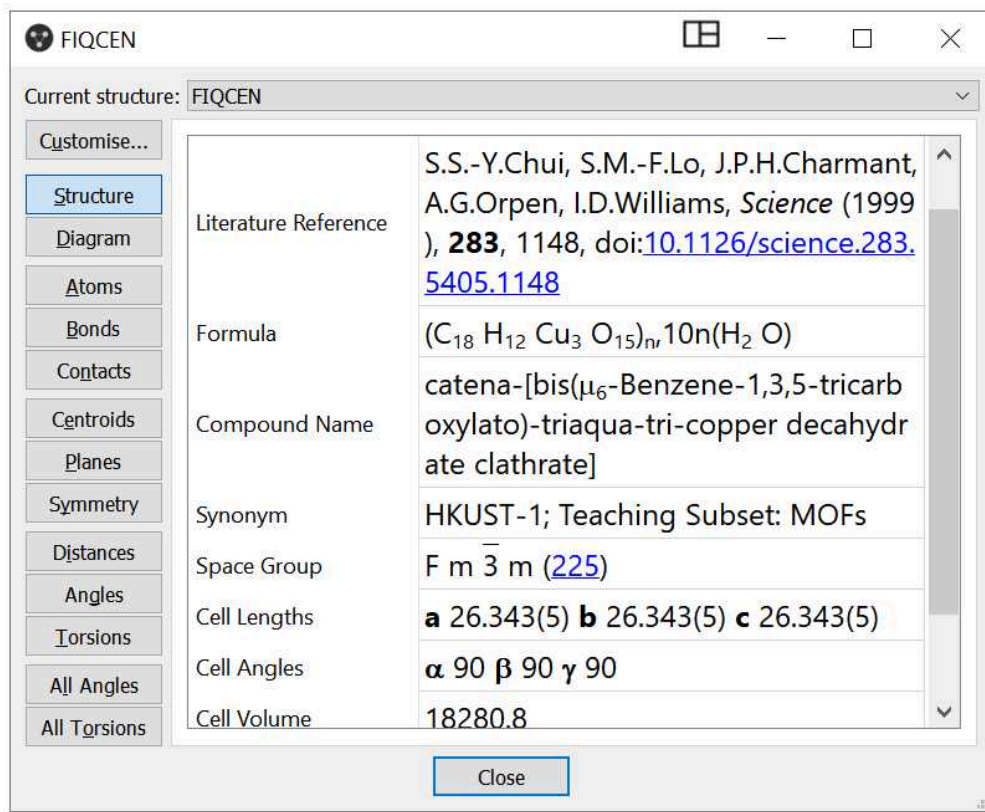


Figure 3. Unit cell dimensions of HKUST-1/FIQCEN as provided by Mercury.

input.dat

```
FIQCEN_desolv.xyz
26.343 26.343 26.343
90      90      90
```

run.bat

```
..\poreblazer.exe < input.dat > results.txt
```

defaults.dat



```
UFF.atoms
2.58, 10.22, 298, 12.8
3.314
500
0.2
20.0, 0.25
21908391
0

! Default forcefield: UFF
! Helium atom sigma (A), helium atom epsilon (K), temperature (K),
cutoff distance (A)
! Nitrogen atom sigma (A)
! Number of samples per atom for the surface area calculation
! 0.2: Cubelet size (A)
! Largest anticipated pore diameter (A), size of the bin for PSD
(A)
! Random number seed
! Visualization options: 1 -xyz, 2 - grd, 3 - both; 0 - none

! Do not change these values unless you know what you are doing
```

UFF.atoms

98			
C	3.431	52.8	12.0
O	3.118	30.2	16.0
H	2.571	22.14	1.0
N	3.261	34.70	14.007
F	2.997	25.14	18.998
Na	2.658	15.09	22.99
Mg	2.691	55.82	24.305
Al	4.008	253.94	26.982
Si	3.826	202.15	28.085
P	3.695	153.37	30.974
S	3.595	137.78	32.06
Cl	3.516	114.15	35.45
K	3.396	17.60	39.098
Ca	3.028	119.68	40.078
Sc	2.936	9.55	44.956
Ti	2.829	8.55	47.867
V	2.801	8.05	50.942
Cr	2.693	7.54	51.996
Mn	2.638	6.54	54.938
Fe	2.594	6.54	55.845
Co	2.559	7.04	58.933
Ni	2.525	7.54	58.693
Cu	3.114	2.51	63.546
Zn	2.462	62.38	65.39
Zr	2.783	34.70	91.224
Mo	2.719	28.16	95.96
Re	2.632	33.19	186.21
Be	2.446	42.74	9.0121831
B	3.638	90.51	10.811
Zn	2.462	62.35	65.39
Ga	3.905	208.69	69.723
Ge	3.813	190.58	72.64
As	3.769	155.38	74.9216
Se	3.746	146.33	78.96
Br	3.732	126.22	79.904
Rb	3.665	20.11	85.4678
Sr	3.244	118.17	87.62
Y	2.980	36.21	88.9059
Nb	2.820	29.67	92.9064
Tc	2.671	24.14	98
Ru	2.640	28.16	101.07
Rh	2.609	26.65	102.9055
Pd	2.583	24.14	106.42

Ag	2.805	18.10	107.8682
Cd	2.537	114.65	112.411
In	3.976	301.21	114.818
Sn	3.913	285.12	118.71
Sb	3.938	225.78	121.76
Te	3.982	200.14	127.6
In	4.009	170.47	126.9045
Cs	4.024	22.63	132.9055
Ba	3.299	183.04	137.327
La	3.138	8.55	138.9055
Ce	3.168	6.54	140.116
Pr	3.213	5.03	140.9077
Nd	3.185	5.03	144.24
Pm	3.160	4.53	145
Sm	3.136	4.02	150.36
Eu	3.112	4.02	151.964
Gd	3.001	4.53	157.25
Tb	3.074	3.52	158.9253
Dy	3.054	3.52	162.5
Ho	3.037	3.52	164.9303
Er	3.021	3.52	167.259
Tm	3.006	3.02	168.9342
Yb	2.989	114.65	173.04
Lu	3.243	20.62	174.967
Hf	2.798	36.21	178.49
Ta	2.824	40.73	180.9479
W	2.734	33.69	183.84
Re	2.632	33.19	186.207
Os	2.780	18.61	190.23
Ir	2.530	36.71	192.217
Pt	2.454	40.23	195.078
Au	2.934	19.61	196.9665
Hg	2.410	193.60	200.59
Tl	3.873	341.94	204.3833
Pb	3.828	333.39	207.2
Bi	3.893	260.48	208.9804
Po	4.195	163.43	209
At	4.232	142.81	210
Rn	4.245	124.71	222
Ra	3.276	203.15	226
Ac	3.099	16.59	227
Th	3.025	13.07	232
Pa	3.050	11.06	231
U	3.025	11.06	238
Np	3.050	9.55	237

Pu	3.050	8.05	244
Am	3.012	7.04	243
Cm	2.963	6.54	247
Bk	2.975	6.54	247
Cf	2.952	6.54	251
Es	2.939	6.03	252
Fm	2.927	6.03	257
Md	2.917	5.53	258
No	2.894	5.53	259
Lw	2.883	5.53	262

! name of framework atom, diameter (LJ sigma) in A, epsilon in K, mol weight

Simulation and plotting/presentation of results

4 Execute PoreBlazer:

1. Start the Windows command prompt by pressing Windows+R and entering '*cmd*' followed by clicking '*OK*'.
2. Change directory to enter the working folder containing the input files.
3. Execute '*run.bat*', wait a few seconds and voila!

```

Command Prompt
Microsoft Windows [Version 10.0.19044.1766]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jonsun>m:

M:\>cd Software\PoreBlazer-main\Windows\FIQEN

M:\Software\PoreBlazer-main\Windows\FIQEN>run

M:\Software\PoreBlazer-main\Windows\FIQEN>.\poreblazer.exe 0

```

Figure 4. Command prompt, showing execution of the PB binary called from *run.bat*

The software should generate the following files:

- '*results.txt*';



- '*summary.dat*'; Contains a brief summary of the results, including pore-limiting and largest cavity diameter, total and network accessible surface area.
- '*Total_psd.txt*' + '*Total_psd_cumulative.txt*'; Total pore-size distribution and
- '*Network-accessible_psd.txt*' + '*Network-accessible_psd_cumulative.txt*'; network accessible pore-size distribution

An archive containing both input and output files as described in this protocol is attached below. A more detailed description of the output can be found at <https://github.com/SarkisovGitHub/PoreBlazer>

Software

PoreBlazer

NAME

Windows 10

OS

Lev Sarkisov

DEVELOPER

<https://github.com/SarkisovGitHub/PoreBlazer>

SOURCE LINK



PSD_FIQCEN_input_output.zip

5 Plotting of single PSD data

Note

This section assumes that you are working with the Anaconda platform for accessing Jupyter Notebooks. Adjust accordingly if not. :)

1. Start JupyterLab by initiating an 'Anaconda prompt' and entering the command '*jupyter lab*'
2. Use the left-pane browser to locate your working directory containing the output from PoreBlazer.
3. Create new launcher by clicking the '+' symbol and choose a Python 3 Notebook.
4. Rename the '*untitled.ipynb*' to '*DATE-REFCODE_PSD_Plots.ipynb*', e.g. '*220927-FIQCEN_PSD_Plots.ipynb*'

5. Add the following code snippets to cells of the notebook (**example notebook is attached below**).
6. Press Ctrl+Enter to execute each cell in order.

Import the required packages

```
# Load pandas for data import/processing + Matplotlib for plotting
import pandas as pd
import matplotlib.pyplot as plt
#Import scipy to do peak fitting
from scipy.signal import find_peaks
```

Define and load data

```
# Load NA PSD data
data1 = "./Network-accessible_psd.txt"
na_psd = pd.read_csv(data1, header=None, skiprows=1, sep='\s+')
na_psd.columns = ['Derivative distribution function', 'd (Å)']

# Load total PSD data
data2 = "./Total_psd.txt"
tot_psd = pd.read_csv(data2, header=None, skiprows=1, sep='\s+')
tot_psd.columns = ['Derivative distribution function', 'd (Å)']
```

Plotting of network accessible pore size distribution

```
def cm2inch(value):  
    return value/2.54  
  
# Initiate plot and define classic style, and 16:9 ratio.  
plt.figure()  
plt.style.use('classic')  
plt.rcParams["figure.figsize"] = (cm2inch(19.2),cm2inch(10.8))  
plt.minorticks_on()  
plt.grid()  
  
# Label axes  
plt.xlabel('d (Å)')  
plt.ylabel('Derivate distribution function')  
  
#Plot NA PSD  
plt.plot(na_psd['Derivative distribution function'],na_psd['d  
(Å)'],color="#8b0000")  
  
# Export as png, change to svg if necessary.  
plt.savefig('FIQCEN_desolv_NA_PSD.png', dpi=300,  
bbox_inches='tight',transparent="True", pad_inches=0)
```

Plotting of total accessible pore size distribution

```
def cm2inch(value):  
    return value/2.54  
  
# Initiate plot and define classic style, and 16:9 ratio.  
plt.figure()  
plt.style.use('classic')  
plt.rcParams["figure.figsize"] = (cm2inch(19.2),cm2inch(10.8))  
plt.minorticks_on()  
plt.grid()  
  
# Label axes  
plt.xlabel('d (Å)')  
plt.ylabel('Derivate distribution function')  
  
#Plot total PSD  
plt.plot(tot_psd['Derivative distribution function'],tot_psd['d  
(Å)'],color="#457ca4")  
  
# Export as png, change to svg if necessary.  
plt.savefig('FIQCEN_desolv_tot_PSD.png', dpi=300,  
bbox_inches='tight',transparent="True", pad_inches=0)
```

Note

Adjust file names, resolution and size after needs, e.g. change to .svg to generate vector graphics for publication.

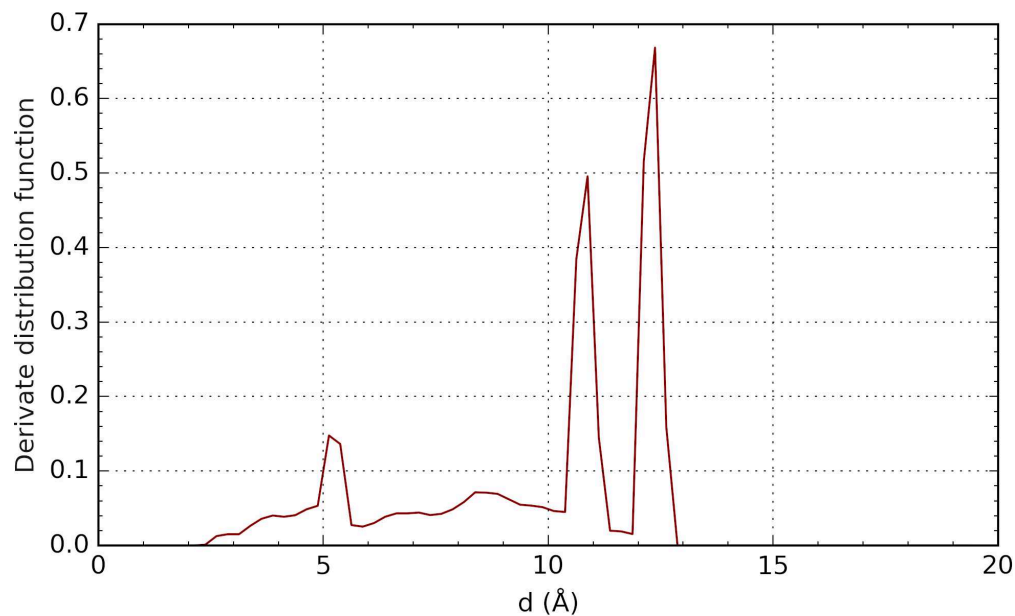


Figure 5. Example network accessible PSD plot of FIQCEN.



220927-FIQCEN_PSD_plots.ipynb

6 Plotting of multiple PSD series as overlay

1. Create a working directory for storage of all PSD data, e.g. '*PSDs*'
2. Make a copy of each single PSD datafile and rename it to 'REFCODE_NA_PSD.txt' (or 'REFCODE_tot_PSD.txt' if plotting total PSD). Note:
3. Move all uniquely named data files into the working directory.
4. Create a new Python 3 notebook in Jupyter Lab as described previously.
5. Add the following code snippets to cells, and execute (**example notebook is attached below**).

Import required packages

```
# Load pandas for data processing, matplotlib for plotting
import pandas as pd
import matplotlib.pyplot as plt
```




Note

This snippet basically imports *all* data from a specific subfolder, plots and labels them incrementally. Very basic, but effective. The legend is labelled based on filename, export as SVG and modify text to more appropriate names (or modify the code).

Multi-data import and overlay plotting

```
#This is a multtiplot script, taking all files from a specific
directory and plotting as overlay.
# Get list of datafiles, modify data_path accordingly if needed
import os
data_path = r'./PSDs/'
data = []
for path in os.listdir(data_path):
    # check if current path is a file
    if os.path.isfile(os.path.join(data_path, path)):
        data.append(path)
print(data)

# Setup plot
def cm2inch(value):
    return value/2.54

plt.figure()
plt.style.use('classic')
plt.rcParams["figure.figsize"] = (cm2inch(19.2),cm2inch(10.8))
plt.minorticks_on()
plt.grid()

plt.xlabel('d (Å)')
plt.ylabel('Derivative distribution function')

#Setup colorscheme, colorblind-friendly palette taken from
https://yoshke.org/blog/colorblind-friendly-diagrams. Modify as
wanted.
colors =
['#000000', '#E69F00', '#56B4E9', '#009E73', '#F0E442', '#0072B2', '#D55
E00', '#CC79A7']

#MinMax
#plt.xlim(500, 2000)

# Do the actual plotting
int = 1
for fname in data:
    df = pd.read_csv(data_path+fname, index_col=False, header=None,
skiprows=1, sep='\s+')
    df.columns = ['Derivative distribution function', 'd (Å)']

    X=df['d (Å)']
    Y=df['Derivative distribution function']
```

```
# Prepare plot with incremental coloring
plt.plot(Y,X,color=colors[int],label=fname)
plt.legend(fontsize=8)
int = int+1

# Export figure, modify filename and settings if needed.
plt.savefig('220927-PSD_NA_Overlay.png', dpi=300,
bbox_inches='tight',transparent="True", pad_inches=0)
```

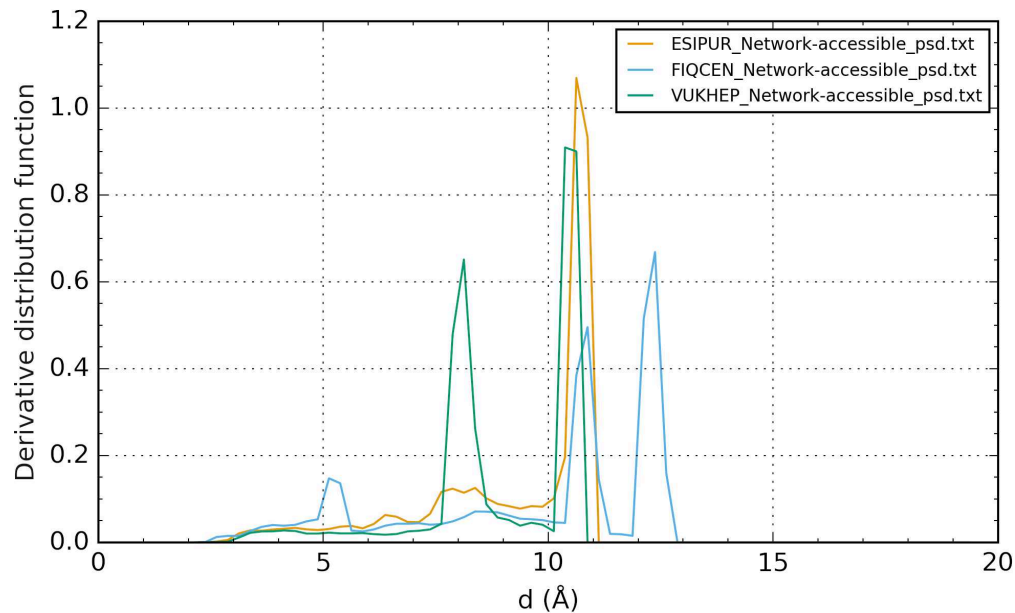


Figure 6. Example of overlay plot of multiple PSDs.

7 **Bonus: Summary clean-up and Excel export**

This is just a short code-snippet to reformat the multiple 'summary.dat' output file into a slightly more user friendly edition. :)

1. Create a secondary working subfolder called '*Summaries*'
2. Make a copy of each 'summary.dat' named '*REFCODE_summary.dat*' and copy into the '*Summaries*' subfolder.
3. In the parent directory, create a new Jupyter Notebook and add the following code to a cell:

```
# Load pandas for data processing
import pandas as pd

# Load single summary.dat
data1 = "./summary.dat"
# Summary.dat is a bit funky, so read using regex with min 2
spaces as separator.
summary = pd.read_csv(data1, header=None, index_col=None,
skiprows=1, sep='\s{2,}', names=
['Parameter', 'Value'], engine='python')
#na_psd.columns = ['Derivative distribution function', 'd (Å)']

# Get original xyz source from summary.dat
with open(data1) as f:
    source_xyz = f.readline()
    # Strip newline character
    source_xyz = source_xyz.strip()

summary.rename(columns = {'Value':source_xyz}, inplace = True)

# Generate list for renaming of indeces
parameters = ['Volume (Å)',
'Mol. Weight (g/mol)',
'Density (g/cm3)',
'Pore-limiting diameter (Å)',
'Largest cavity diameter (Å)',
'Dimensionality', 'Total properties',
'Total accessible surface area (Å2)',
'Total accessible surface area (m2/cm3)',
'Total accessible surface area (m2/g)',
'Total properties',
'Total helium pore volume (Å3)',
'Total helium pore volume (cm3/g)',
'Total geometric pore volume (Å3)',
'Total geometric pore volume (cm3/g)',
'Total probe-occupiable pore volume (Å3)',
'Total probe-occupiable pore volume (cm3/g)',
'Total volume fraction (%)',
'Network accessible',
'Network accessible surface area (Å2)',
'Network accessible surface area (m2/cm3)',
'Network accessible surface area (m2/g)',
'Network accessible',
'Network accessible helium pore volume (Å3)',
```

```
'Network accessible helium pore volume (cm3/g)',
'Network accessible geometric pore volume (Å3)',
'Network accessible geometric pore volume (cm3/g)',
'Network accessible probe-occupiable pore volume (Å3)',
'Network accessible probe-occupiable pore volume (cm3/g)',
'Network accessible volume fraction (%)']

# Update parameter list with slightly more userfriendly text
int = 0
for index, row in summary.iterrows():
    summary.loc[int, ['Parameter']] = [parameters[int]]
    int = int+1

# Drop rows with NaN values
summary = summary.dropna()

# Export as Excel file.
summary.to_excel('220927-Multi_Summary.xlsx', index=False)
```

Parameter	ESIPUR.xyz	FIQCEN_desolv.xyz	VUKHEP.xyz
Volume (Å)	2666.065	18280.821	1764.214
Mol. Weight (g/mol)	1696.786	7818.552	135136.972
Density (g/cm3)	1.057	0.71	127.196
Pore-limiting diameter (Å)	10.16	6.38	2.96
Largest cavity diameter (Å)	10.86	12.57	4.33
Dimensionality	1	3	1
Total accessible surface area (Å2)	330.39	3089.92	2.79
Total accessible surface area (m2/cm3)	1239.23	1690.25	15.83
Total accessible surface area (m2/g)	1172.59	2379.97	0.12
Total geometric pore volume (Å3)	1704.147	12888.62	361.44
Total geometric pore volume (cm3/g)	0.605	0.993	0.002
Total probe-occupiable pore volume (Å3)	1385.131	12401.43	142.118
Total probe-occupiable pore volume (cm3/g)	0.492	0.955	0.001
Total volume fraction (%)	0.51954	0.67838	0.08056
Network accessible surface area (Å2)	349.48	3068.63	0
Network accessible surface area (m2/cm3)	1310.85	1678.6	0
Network accessible surface area (m2/g)	1240.36	2363.57	0
Network accessible geometric pore volume (Å3)	1703.981	12887.481	359.409
Network accessible geometric pore volume (cm3/g)	0.605	0.993	0.002
Network accessible probe-occupiable pore volume (Å3)	1378.521	12431.265	0
Network accessible probe-occupiable pore volume (cm3/g)	0.489	0.958	0
Network accessible volume fraction (%)	0.51706	0.68002	0

Figure 7. Example of multi-summary Excel output.