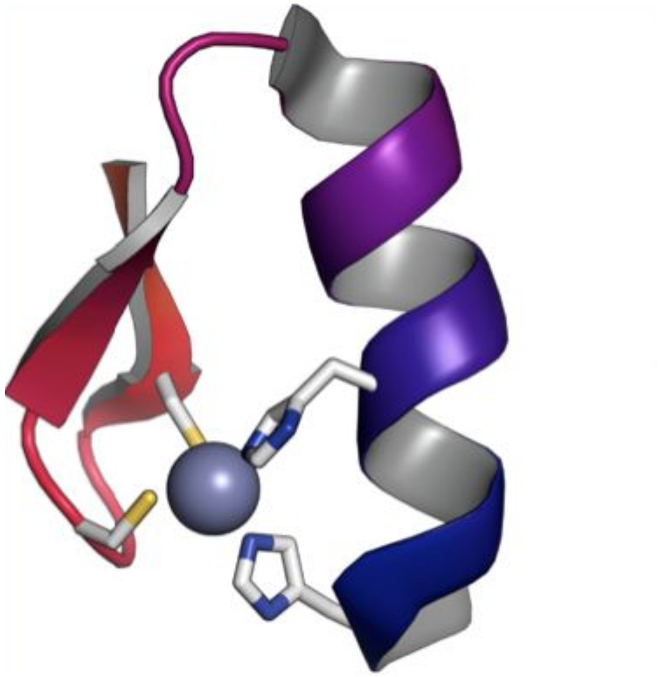
COMPUTATION IN BIOMEDICAL RESEARCH

Upneet Kaur and Alina Arzamassky

09/29/20

Zinc Coordination Spheres in Protein Structures

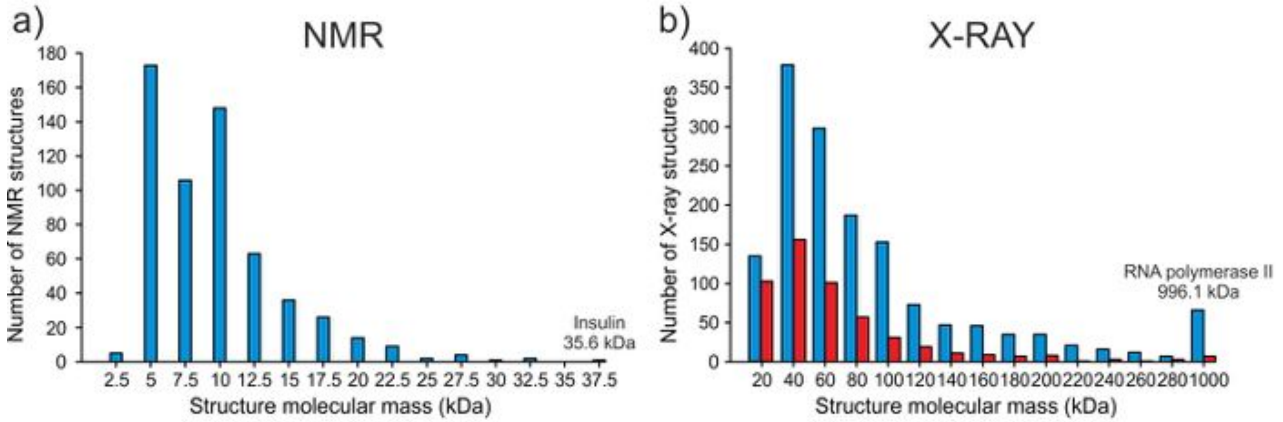


- Zinc metalloproteins are one of the most abundant and structurally diverse proteins
- A number of database surveys have been conducted on zinc proteins to gain broader insights into their rich coordination chemistry
- Zinc containing protein structures deposited in the Protein Data Bank (PDB) were analyzed in detail.

Mikko Laitaoja *.et al,* Inorganic Chem., 2013

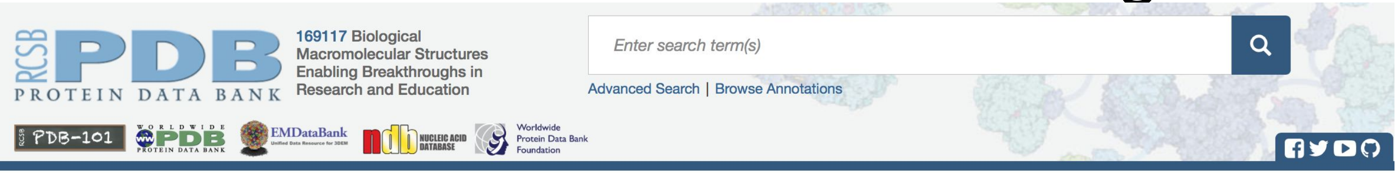
Molecular mass distributions of zinc proteins determined by NMR and X-ray crystallography

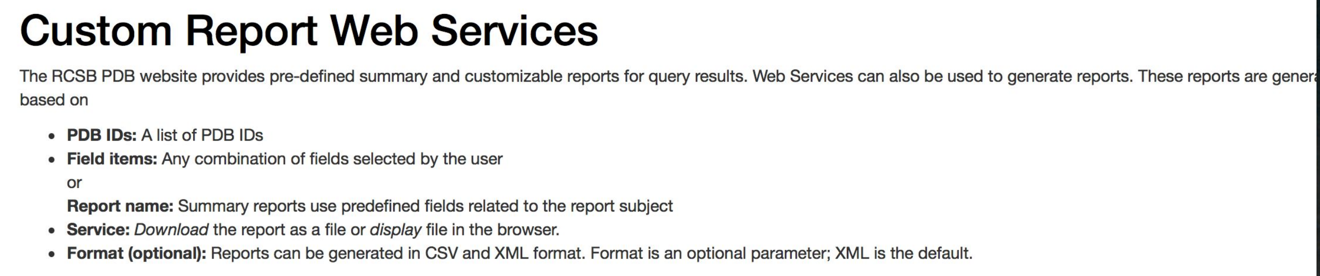
- Computational problem: Determine the molecular mass distribution of Zn-containing proteins solved by NMR and X-RAY crystallography.
- The input is the PDB file and output is the histogram displaying the molecular mass distribution.



Mikko Laitaoja *.et al,* Inorganic Chem., 2013

Queried for all structures and their respective molecular weights

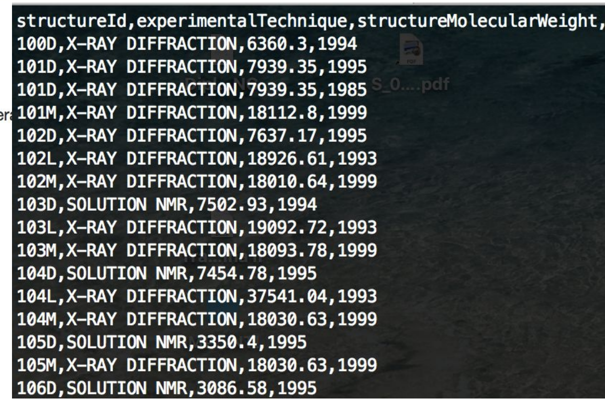

RCSB PDB PROTEIN DATA BANK
169117 Biological Macromolecular Structures Enabling Breakthroughs in Research and Education
Enter search term(s)
Advanced Search | Browse Annotations
PDB-101
WORLDWIDE PDB PROTEIN DATA BANK
EMDataBank
ndb NUCLEIC ACID DATABASE
Worldwide Protein Data Bank Foundation



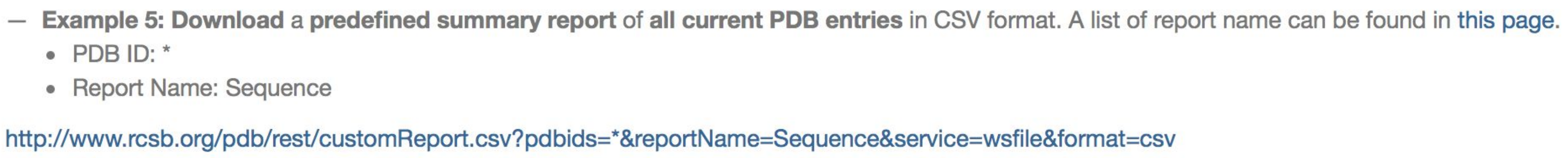
Custom Report Web Services

The RCSB PDB website provides pre-defined summary and customizable reports for query results. Web Services can also be used to generate reports. These reports are gener
based on

- **PDB IDs:** A list of PDB IDs
- **Field items:** Any combination of fields selected by the user
  or
  Report name: Summary reports use predefined fields related to the report subject
- **Service:** *Download* the report as a file or *display* file in the browser.
- **Format (optional):** Reports can be generated in CSV and XML format. Format is an optional parameter; XML is the default.



```
structureId,experimentalTechnique,structureMolecularWeight,
100D,X-RAY DIFFRACTION,6360.3,1994
101D,X-RAY DIFFRACTION,7939.35,1995
101D,X-RAY DIFFRACTION,7939.35,1985
101M,X-RAY DIFFRACTION,18112.8,1999
102D,X-RAY DIFFRACTION,7637.17,1995
102L,X-RAY DIFFRACTION,18926.61,1993
102M,X-RAY DIFFRACTION,18010.64,1999
103D,SOLUTION NMR,7502.93,1994
103L,X-RAY DIFFRACTION,19092.72,1993
103M,X-RAY DIFFRACTION,18093.78,1999
104D,SOLUTION NMR,7454.78,1995
104L,X-RAY DIFFRACTION,37541.04,1993
104M,X-RAY DIFFRACTION,18030.63,1999
105D,SOLUTION NMR,3350.4,1995
105M,X-RAY DIFFRACTION,18030.63,1999
106D,SOLUTION NMR,3086.58,1995
```

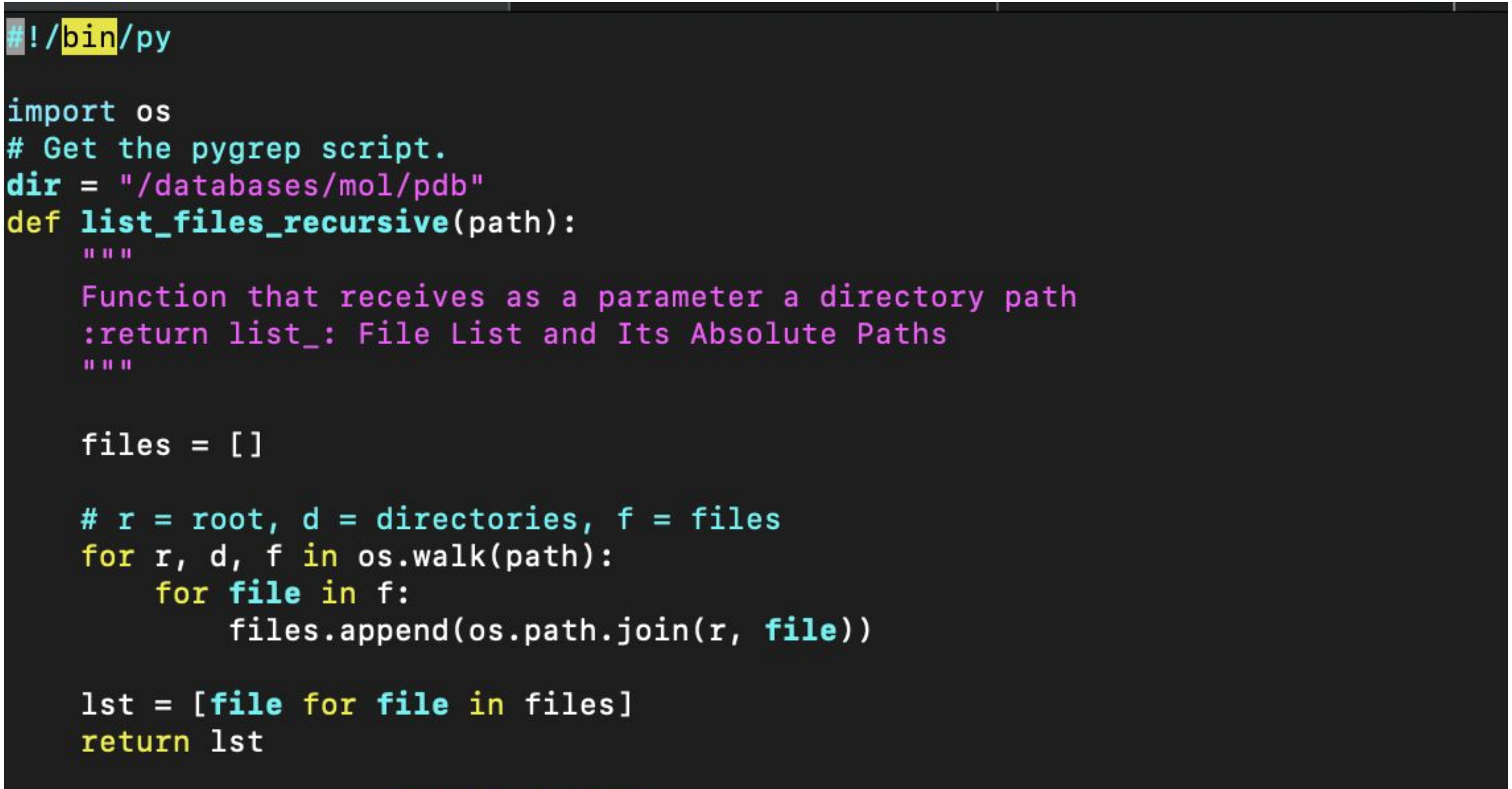


– **Example 5: Download** a **predefined summary report** of **all current PDB entries** in CSV format. A list of report name can be found in this page.
- PDB ID: *
- Report Name: Sequence

http://www.rcsb.org/pdb/rest/customReport.csv?pdbids=*&reportName=Sequence&service=wsfile&format=csv

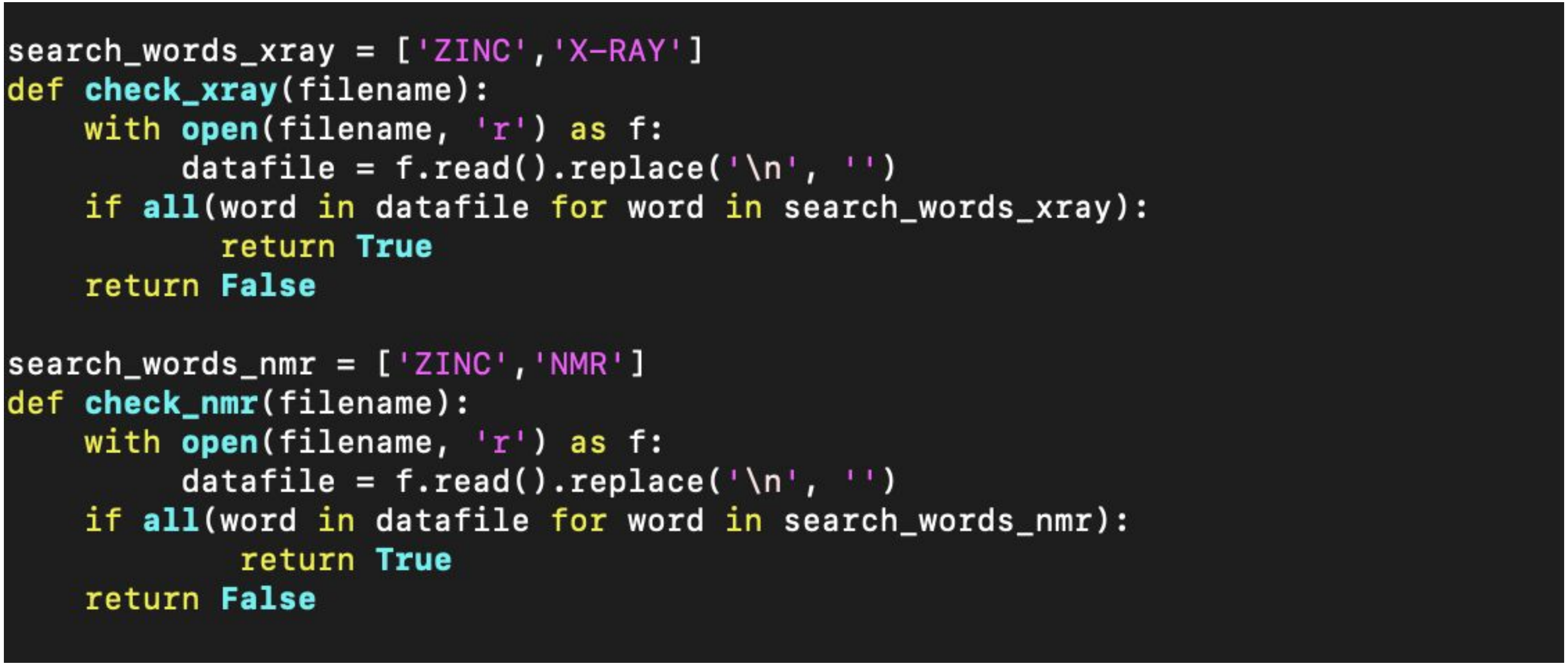
sed 's/<br \/>/\n/g' customReport.csv > customReport.sed.csv
or for mac
ed -e 's/<br \/>/\'$'\n/g' customReport2.csv > customReport2.sed.csv

Implementation



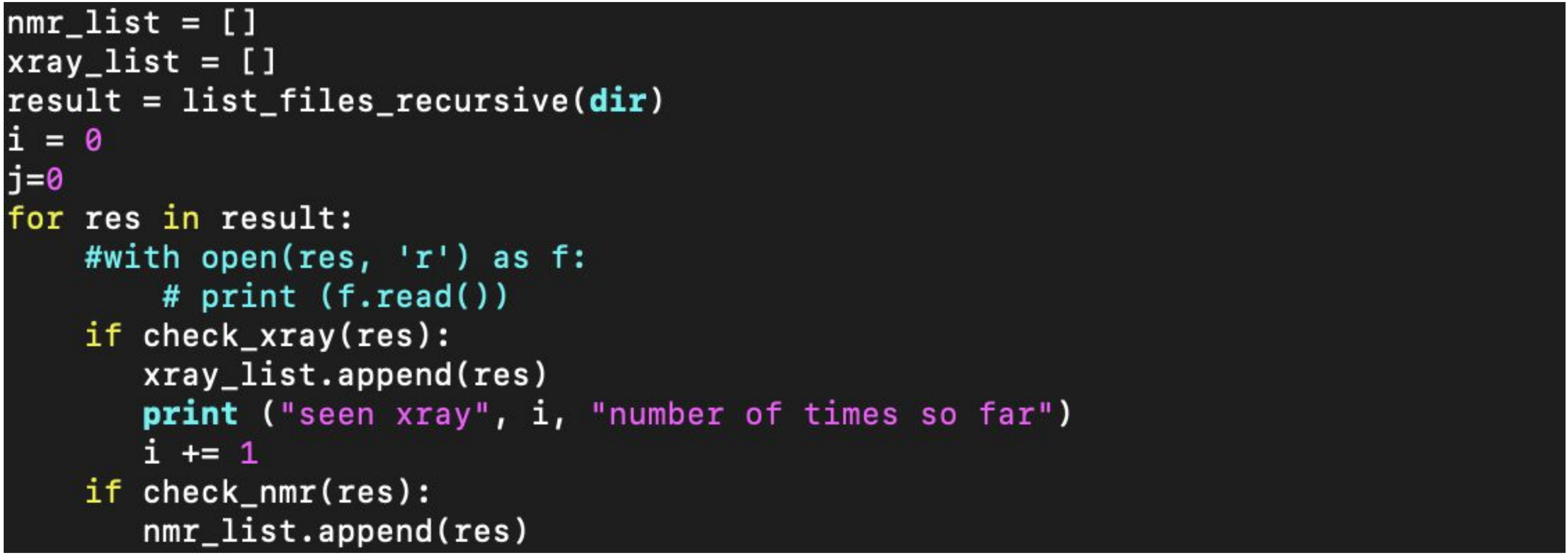
```
#!/bin/py

import os
# Get the pygrep script.
dir = "/databases/mol/pdb"
def list_files_recursive(path):
    """
    Function that receives as a parameter a directory path
    :return list_: File List and Its Absolute Paths
    """

    files = []

    # r = root, d = directories, f = files
    for r, d, f in os.walk(path):
        for file in f:
            files.append(os.path.join(r, file))

    lst = [file for file in files]
    return lst
```

Implementation



```python
search_words_xray = ['ZINC','X-RAY']
def check_xray(filename):
    with open(filename, 'r') as f:
         datafile = f.read().replace('\n', '')
    if all(word in datafile for word in search_words_xray):
           return True
    return False

search_words_nmr = ['ZINC','NMR']
def check_nmr(filename):
    with open(filename, 'r') as f:
         datafile = f.read().replace('\n', '')
    if all(word in datafile for word in search_words_nmr):
            return True
    return False
```

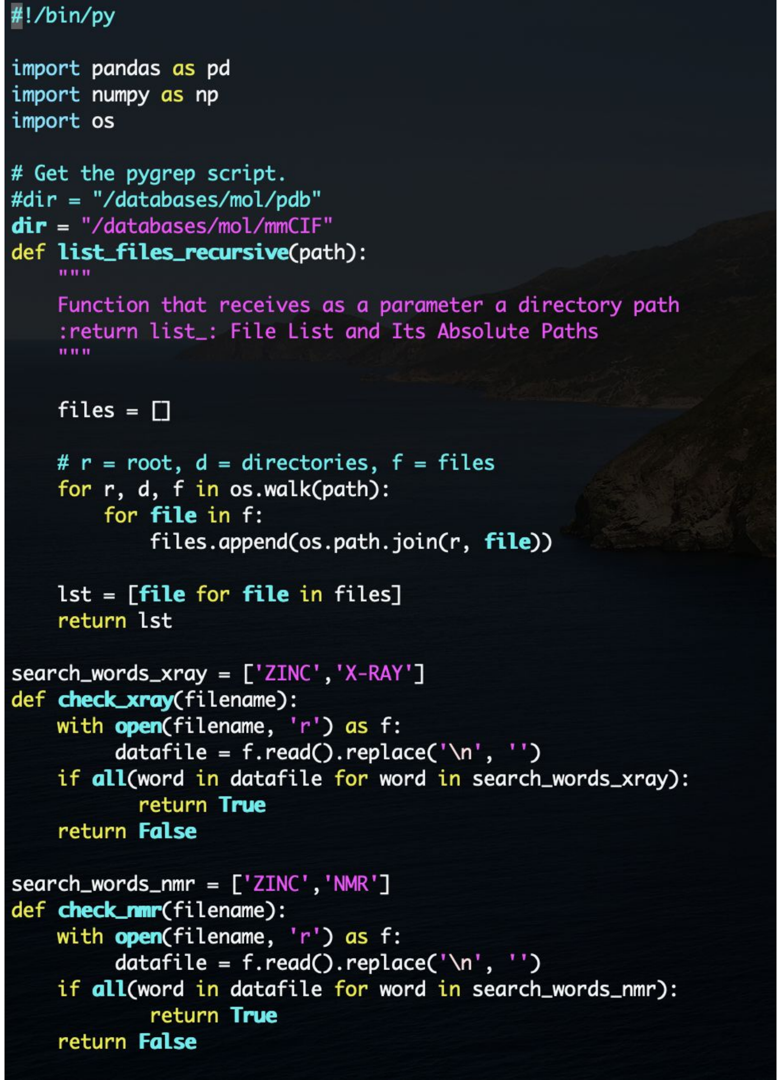
Implementation



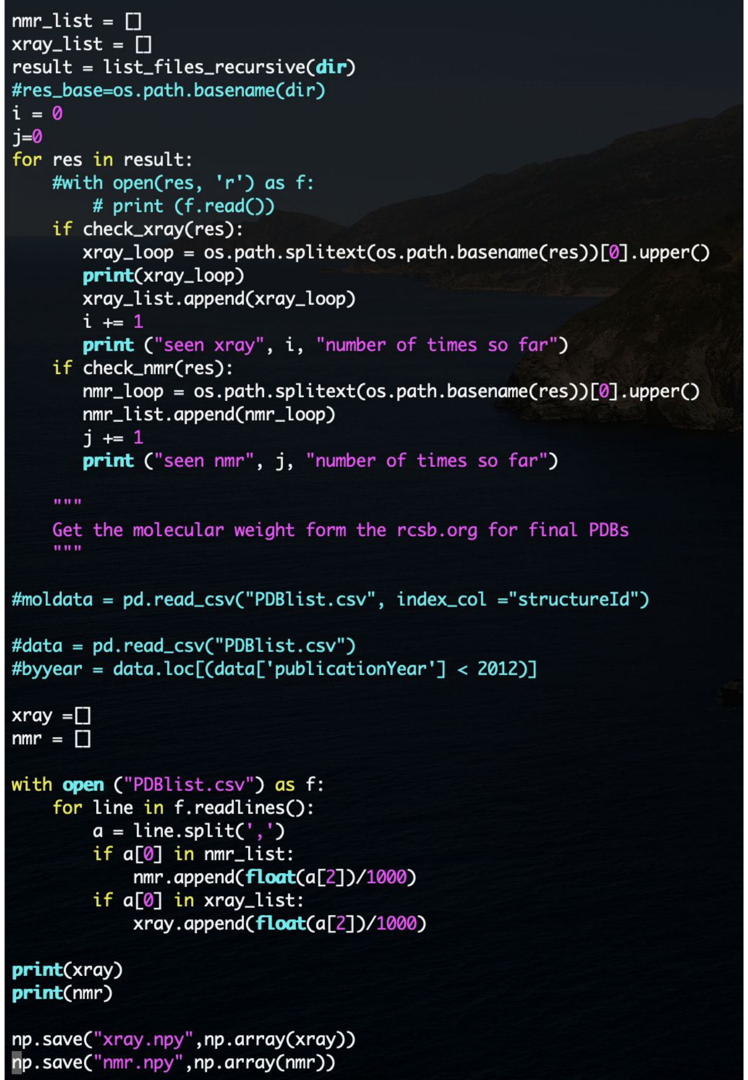
```python
nmr_list = []
xray_list = []
result = list_files_recursive(dir)
i = 0
j=0
for res in result:
    #with open(res, 'r') as f:
        # print (f.read())
    if check_xray(res):
       xray_list.append(res)
       print ("seen xray", i, "number of times so far")
       i += 1
    if check_nmr(res):
       nmr_list.append(res)
```

X-RAY: 16570

NMR: 1479

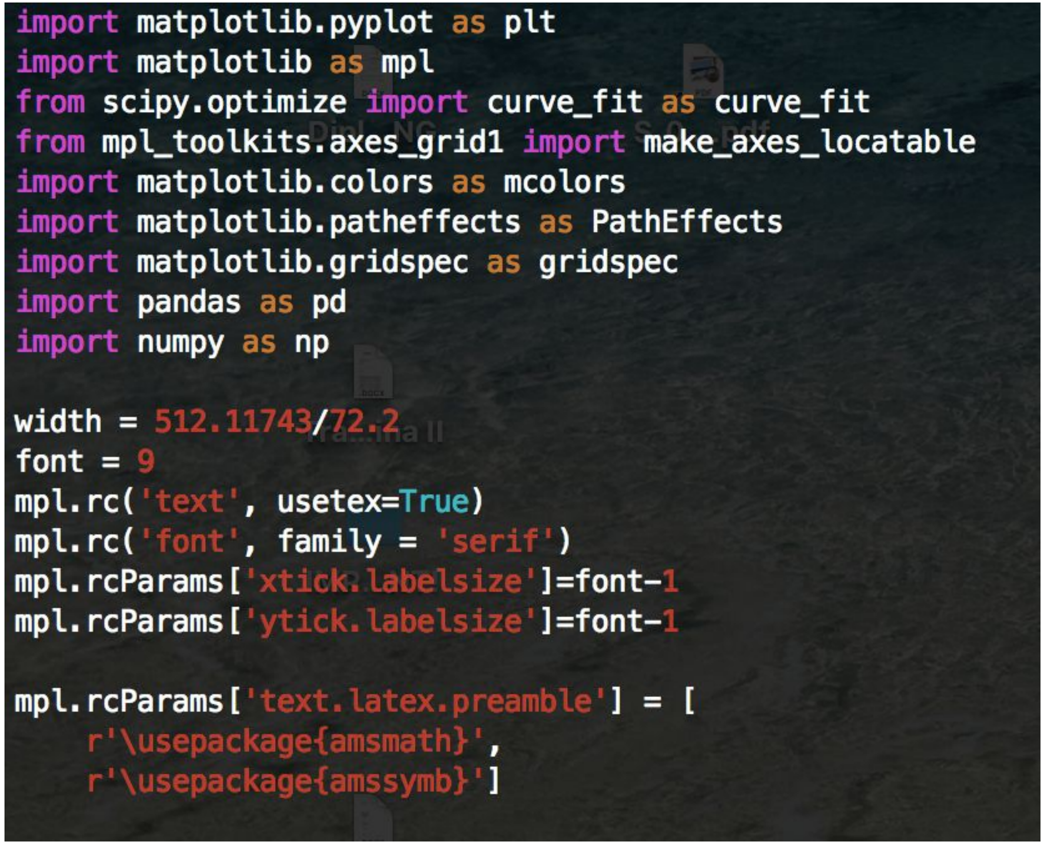


```python
#!/bin/py

import pandas as pd
import numpy as np
import os

# Get the pygrep script.
#dir = "/databases/mol/pdb"
dir = "/databases/mol/mmCIF"
def list_files_recursive(path):
    """
    Function that receives as a parameter a directory path
    :return list_: File List and Its Absolute Paths
    """

    files = []

    # r = root, d = directories, f = files
    for r, d, f in os.walk(path):
        for file in f:
            files.append(os.path.join(r, file))

    lst = [file for file in files]
    return lst

search_words_xray = ['ZINC','X-RAY']
def check_xray(filename):
    with open(filename, 'r') as f:
         datafile = f.read().replace('\n', '')
    if all(word in datafile for word in search_words_xray):
           return True
    return False

search_words_nmr = ['ZINC','NMR']
def check_nmr(filename):
    with open(filename, 'r') as f:
         datafile = f.read().replace('\n', '')
    if all(word in datafile for word in search_words_nmr):
            return True
    return False
```



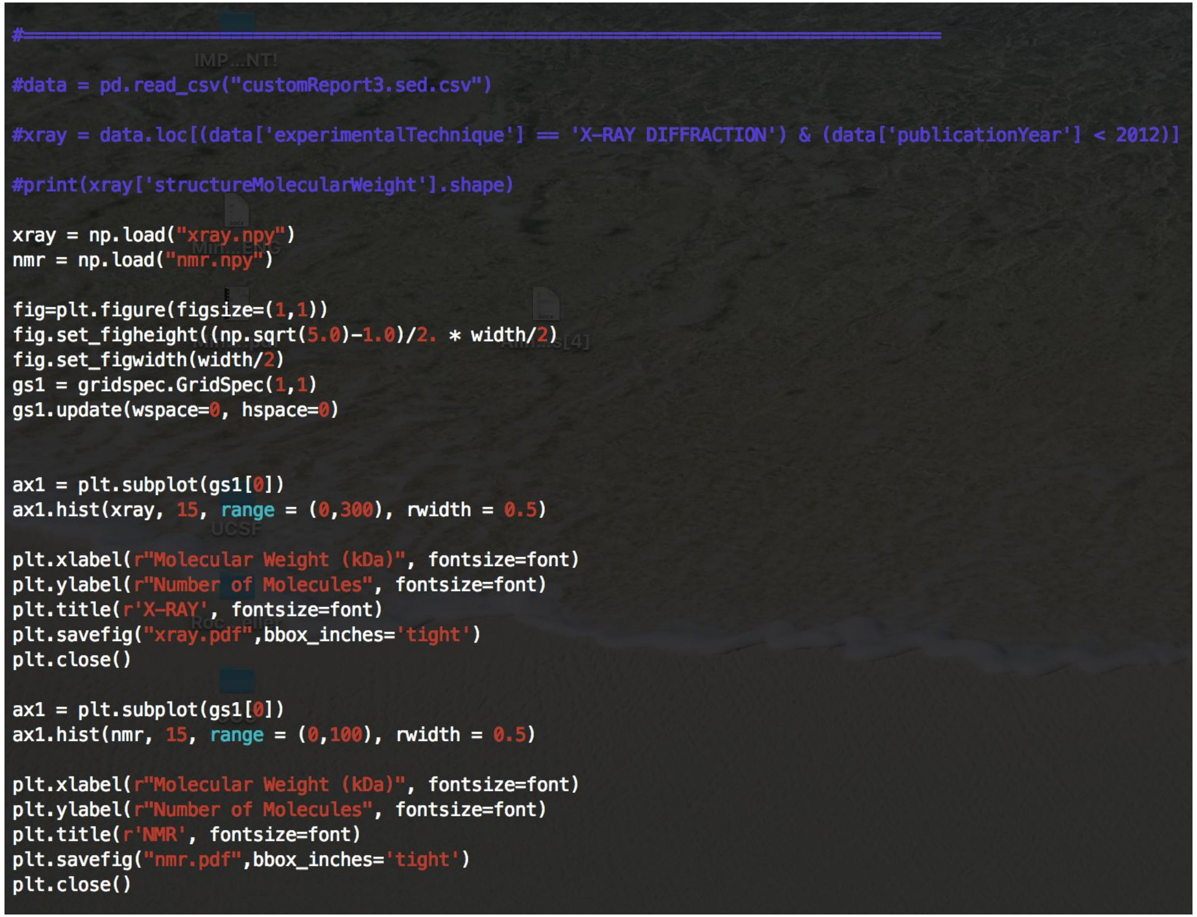
```python
nmr_list = []
xray_list = []
result = list_files_recursive(dir)
#res_base=os.path.basename(dir)
i = 0
j=0
for res in result:
    #with open(res, 'r') as f:
        # print (f.read())
    if check_xray(res):
       xray_loop = os.path.splitext(os.path.basename(res))[0].upper()
       print(xray_loop)
       xray_list.append(xray_loop)
       i += 1
       print ("seen xray", i, "number of times so far")
    if check_nmr(res):
       nmr_loop = os.path.splitext(os.path.basename(res))[0].upper()
       nmr_list.append(nmr_loop)
       j += 1
       print ("seen nmr", j, "number of times so far")

    """
    Get the molecular weight form the rcsb.org for final PDBs
    """

#moldata = pd.read_csv("PDBlist.csv", index_col ="structureId")

#data = pd.read_csv("PDBlist.csv")
#byyear = data.loc[(data['publicationYear'] < 2012)]

xray =[]
nmr = []

with open ("PDBlist.csv") as f:
    for line in f.readlines():
        a = line.split(',')
        if a[0] in nmr_list:
            nmr.append(float(a[2])/1000)
        if a[0] in xray_list:
            xray.append(float(a[2])/1000)

print(xray)
print(nmr)

np.save("xray.npy",np.array(xray))
np.save("nmr.npy",np.array(nmr))
```

Implementation

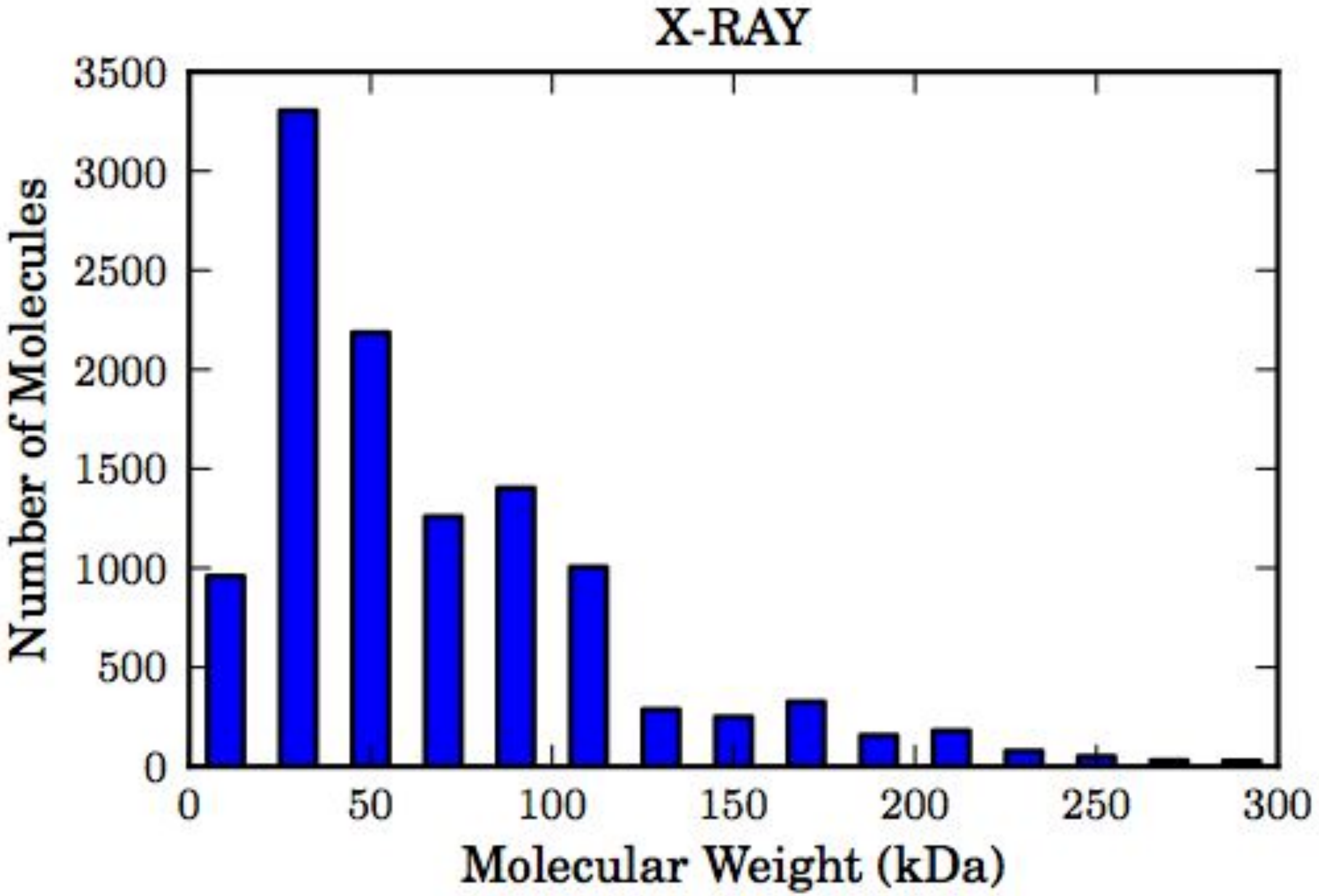


```python
import matplotlib.pyplot as plt
import matplotlib as mpl
from scipy.optimize import curve_fit as curve_fit
from mpl_toolkits.axes_grid1 import make_axes_locatable
import matplotlib.colors as mcolors
import matplotlib.patheffects as PathEffects
import matplotlib.gridspec as gridspec
import pandas as pd
import numpy as np

width = 512.11743/72.2
font = 9
mpl.rc('text', usetex=True)
mpl.rc('font', family = 'serif')
mpl.rcParams['xtick.labelsize']=font-1
mpl.rcParams['ytick.labelsize']=font-1

mpl.rcParams['text.latex.preamble'] = [
    r'\usepackage{amsmath}',
    r'\usepackage{amssymb}']
```

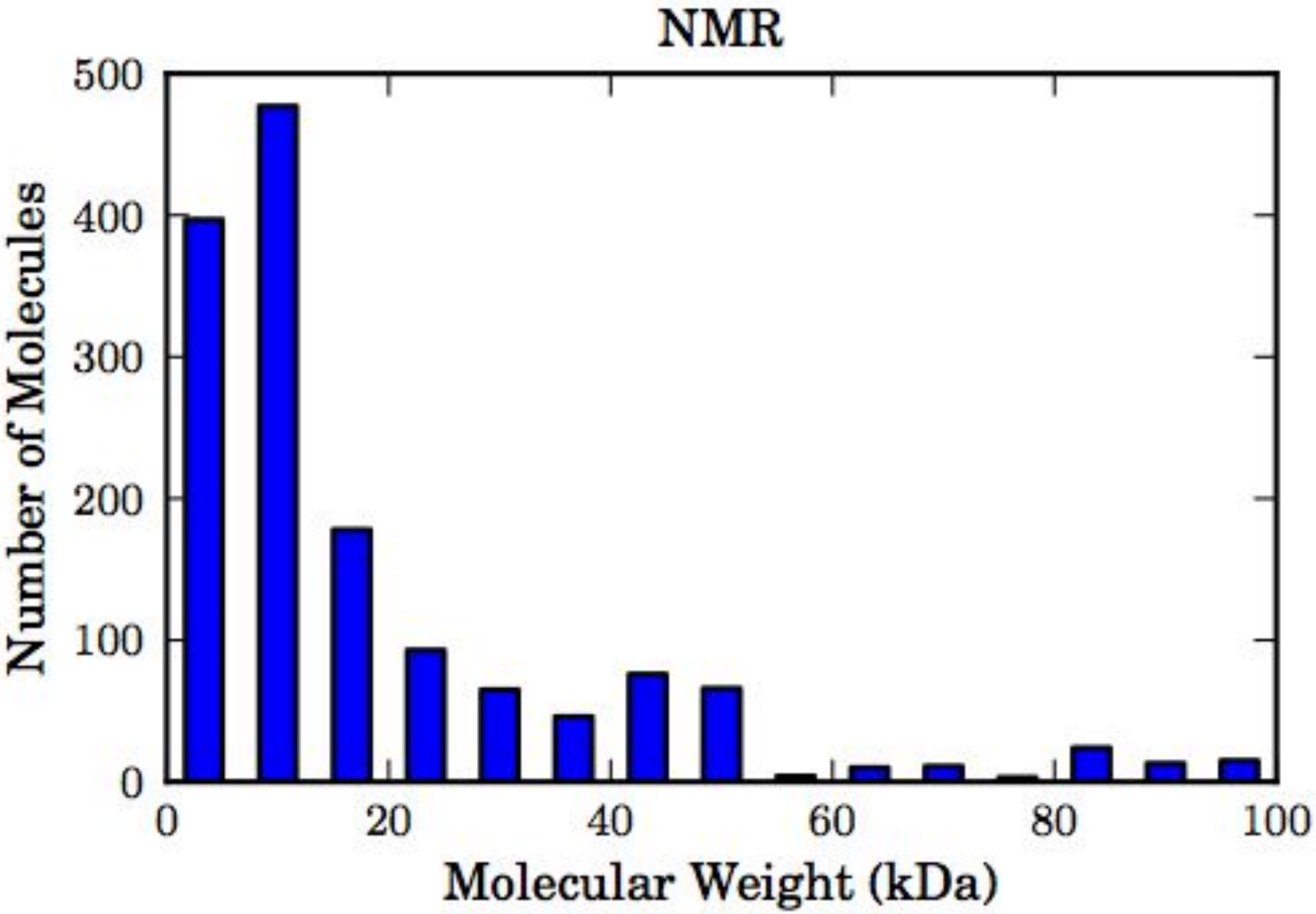
Implementation



```
#=================================================================================================

#data = pd.read_csv("customReport3.sed.csv")

#xray = data.loc[(data['experimentalTechnique'] == 'X-RAY DIFFRACTION') & (data['publicationYear'] < 2012)]

#print(xray['structureMolecularWeight'].shape)

xray = np.load("xray.npy")
nmr = np.load("nmr.npy")

fig=plt.figure(figsize=(1,1))
fig.set_figheight((np.sqrt(5.0)-1.0)/2. * width/2)
fig.set_figwidth(width/2)
gs1 = gridspec.GridSpec(1,1)
gs1.update(wspace=0, hspace=0)


ax1 = plt.subplot(gs1[0])
ax1.hist(xray, 15, range = (0,300), rwidth = 0.5)

plt.xlabel(r"Molecular Weight (kDa)", fontsize=font)
plt.ylabel(r"Number of Molecules", fontsize=font)
plt.title(r'X-RAY', fontsize=font)
plt.savefig("xray.pdf",bbox_inches='tight')
plt.close()

ax1 = plt.subplot(gs1[0])
ax1.hist(nmr, 15, range = (0,100), rwidth = 0.5)

plt.xlabel(r"Molecular Weight (kDa)", fontsize=font)
plt.ylabel(r"Number of Molecules", fontsize=font)
plt.title(r'NMR', fontsize=font)
plt.savefig("nmr.pdf",bbox_inches='tight')
plt.close()
```



X-RAY
Number of Molecules
Molecular Weight (kDa)
0
500
1000
1500
2000
2500
3000
3500
0
50
100
150
200
250
300


NMR
Number of Molecules
Molecular Weight (kDa)
0
100
200
300
400
500
0
20
40
60
80
100