Zinc Binding Protein Molecular Mass Graphs

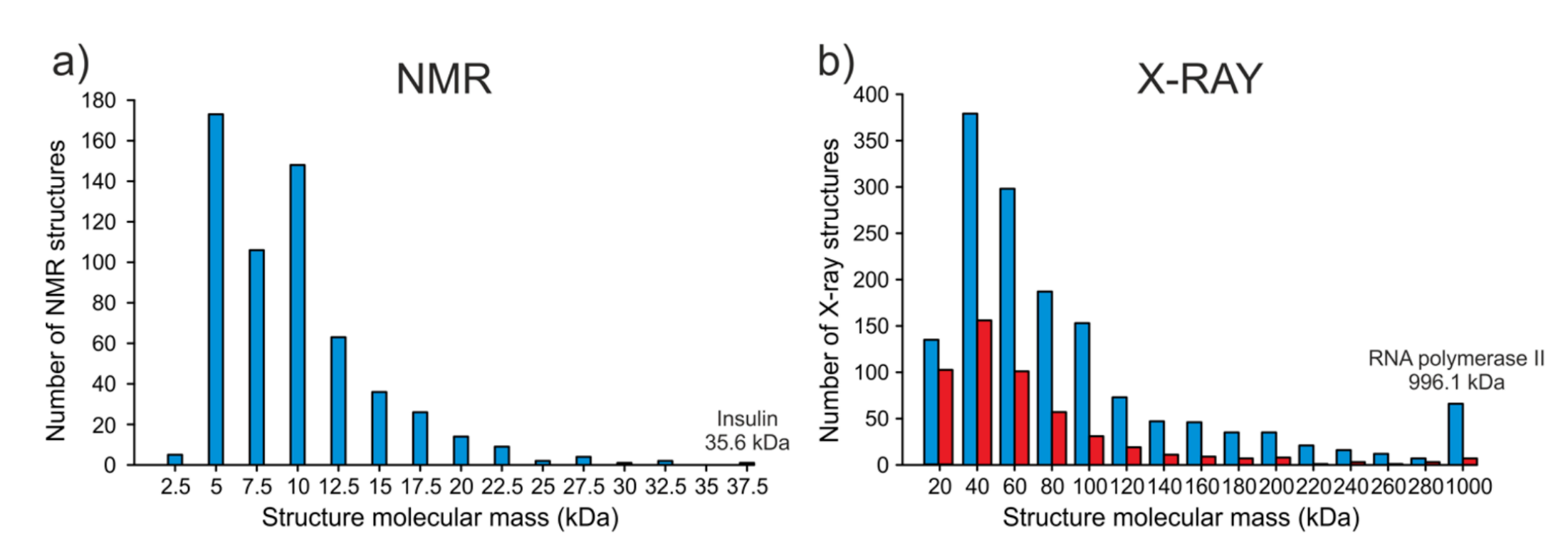
Simone Harrison & Maxine Bi

Background

- Laitaoja, et al. “Zinc Coordination Spheres in Protein Structures” (2013)
  - *“Zinc is one of the most abundant metals is biology, and it is estimated that about one-tenth of proteins may contain a zinc ion as a cofactor”*
- Protein coordination spheres = what amino acid atoms coordinate zinc
- This knowledge could predict zinc binding sites in novel proteins
- NMR structures = mostly zinc fingers, average molecular mass of ~8.6 kDa
- X-Ray structures = 80.6 kDa average for asymmetric unit (not biological assembly or functional unit). Enzymes have an average molecular mass = 92.6 kDa

Goal

- Reproduce Figure a and b:

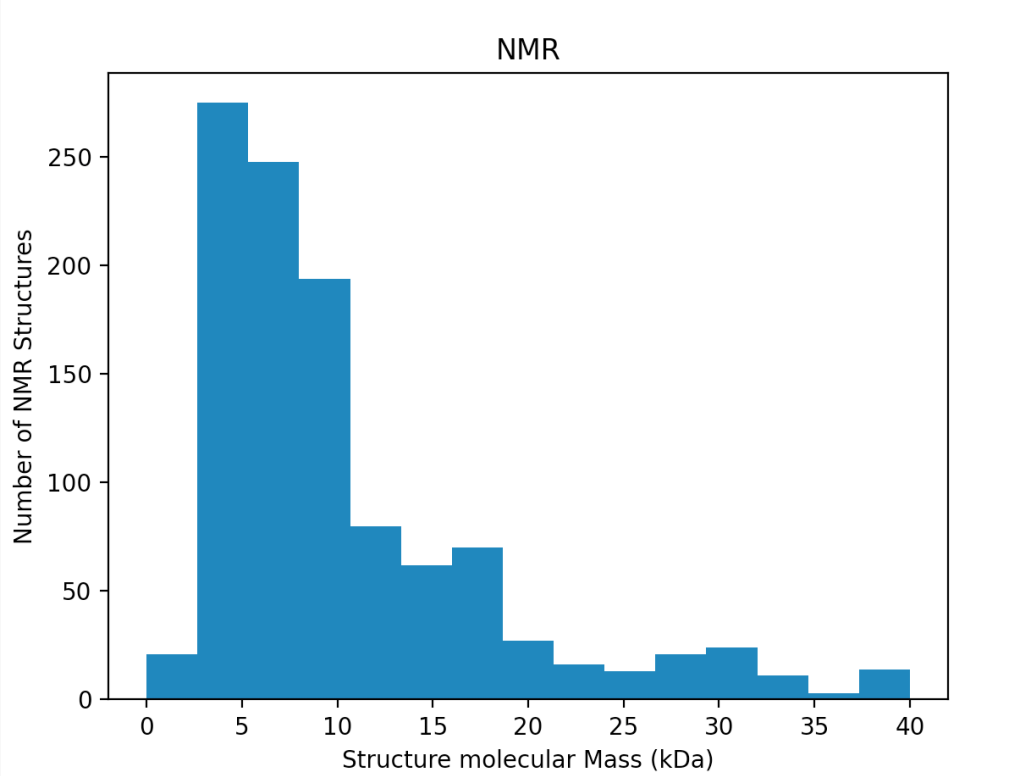


Molecular mass distributions of zinc proteins determined by (a) NMR and (b) X-ray crystallography. Red bars in the X-ray correspond to crystallization artifacts.

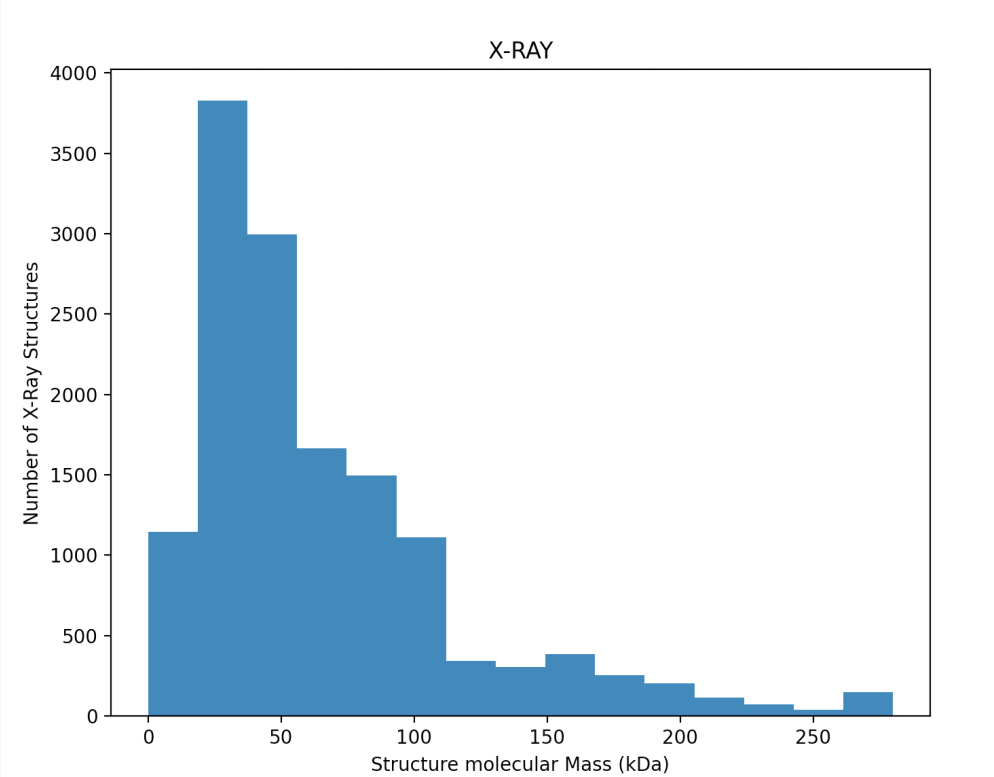
Product

- Reproduced Figure a and b:

a)



b)

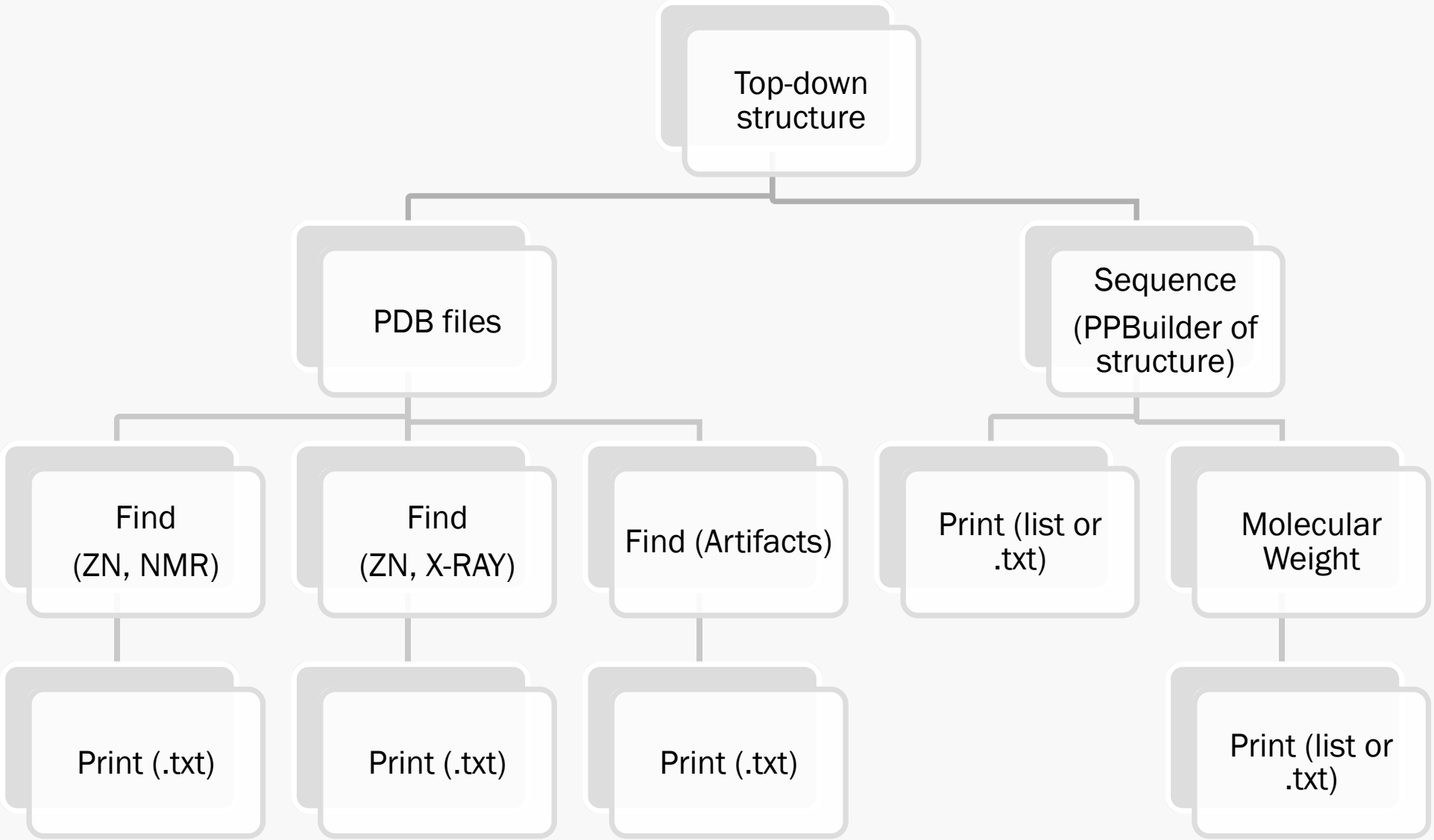


Molecular mass distributions of zinc proteins determined by (a) NMR and (b) X-ray crystallography. Red bars in the X-ray correspond to crystallization artifacts.

Partition

- Code organization:
  - Functions to:
    - return files that contain zinc (determined by X-RAY and/or NMR) MB
    - Return sequence of each file MB & SH
    - Return MW of each file MB & SH
    - Plot Frequency and MW SH
- Started in bash, moved to python in order to use Biopython
- API
  - *Python*
  - *Biopython*
  - *NumPy*
  - *Matplotlib*

Flowchart


Top-down structure
PDB files
Sequence (PPBuilder of structure)
Find (ZN, NMR)
Find (ZN, X-RAY)
Find (Artifacts)
Print (list or .txt)
Molecular Weight
Print (.txt)
Print (.txt)
Print (.txt)
Print (list or .txt)

Documentation

- Wrote a ReadMe file
- Comments within the file to demonstrate tunability

master | Zinc-Project / pdb database sequence and MW program / readme.md | Go to file

simoneaharrison Add files via upload | Latest commit 932e2c2 1 hour ago | History

1 contributor

20 lines (12 sloc) | 1.32 KB | Raw | Blame

This program takes 3 arguments:

python script(either PDBsequencereader.py or MolecularWeightofPDB.py) searchword1 searchword2

In this case, search word 1 was a metal (ZN) which is why it was annotated as such. We used ' ZN ' to make sure it was found in the structure. "ZINC" or "ZN" was sometimes found in other places in the file, but in atom identifiers it always had the spacing around it.

Searchword2 in this case was the method used, 'X-RAY DIFFRACTION' or 'NMR'.

These terms can be substituted for any searchwords.

Within the files, one can change the file path, or change what the functions return (to print the filename/PDB identifier or just sequence or MW).

These work by first creating a structure object from the file with Biopython. Then, from the structure object, a sequence object is also created with Biopython. This object is translated in a string, and all strings (corresponding to all chains in the atom) are returned for each pdb file.

Then, to calculate the molecular weight from each structure, each chain is converted into a MW using Protein analysis from Bio.SeqUtils.ProtParam (also biopython) And the MW of all chains are cocatenated to a final molecular weight.

The 'find' function is used to iterate through all files with the two search words (in this case, metal and filetype), and run the functions on them.

Repository

MaxineBi / Zinc-Project Private

Watch 1 | Star 0 | Fork

Code | Issues | Pull requests | Actions | Projects | Security | Insights

master | Zinc-Project / pdb database sequence and MW program / | Go to file | Add file

MaxineBi Rename listreadergrapher.py to listreadergrapher_XRAY.py `93cfcb1` 36 minutes ago **History**

..

MolecularWeightofPDB.py	Add files via upload	41 minutes ago
PDBsequencereader.py	Add files via upload	41 minutes ago
listreadergrapher_XRAY.py	Rename listreadergrapher.py to listreadergrapher_XRAY.py	36 minutes ago
readme.md	Add files via upload	41 minutes ago

Implementation

- Tested on 2 subdirectories to begin
- Code Demo (write MW to a file, read file to a list and plot)

```
import sys
import numpy as np
from Bio.PDB import *
from Bio.SeqUtils.ProtParam import ProteinAnalysis
from sys import argv
import os,glob
folder_path = '/databases/mol/pdb/*/'
#use path above to direct to where your files are

script, metal, filetype = argv

#call structure from file
def structure(pdb):
 parser = PDBParser(PERMISSIVE=1, QUIET=True)
 try:
  structure = parser.get_structure("input", pdb)
  return structure
 except OSError as e:
  print('Could not locate PDB file')

#return sequence of all chains and pdb identifier
def sequence(filename):
 ppb = PPBuilder()
 return('>',filename[25:29])
 for pp in ppb.build_peptides(structure(filename)):
  seq = pp.get_sequence()
  seqstring = str(seq)
  return(seqstring)


for filename in glob.glob(os.path.join(folder_path, '*.ent')):
  with open(filename, 'r') as f:
   FileContents = f.read()
   if FileContents.find(metal) != -1 and FileContents.find(filetype) != -1:
     print(sequencefilename))
```

Conclusion

- This program allows you to search a database of pdb files for files containing two strings, then allows you to return the sequences and/or molecular weights of the structure represented in the PDB file

- A remarkably similar histogram, just different frequency scale

Future Directions

- Our histogram still looks different. These differences are due to:
  - *Different plotting style and slightly different binning*
  - *not being able to make blastclust work to remove redundant structures*
  - *database has grown exponentially*

- Future directions: use mmCIF instead, add error messages, remove redundant sequences, refine program to be able to exclude files with strings (as opposed to only include files with strings), figure out a way to reliably identify artifacts.