MMTF 2017 Hackathon

Summary Slides
June 28, 2017

Finding Structurally Similar Proteins (Kevin Savage)

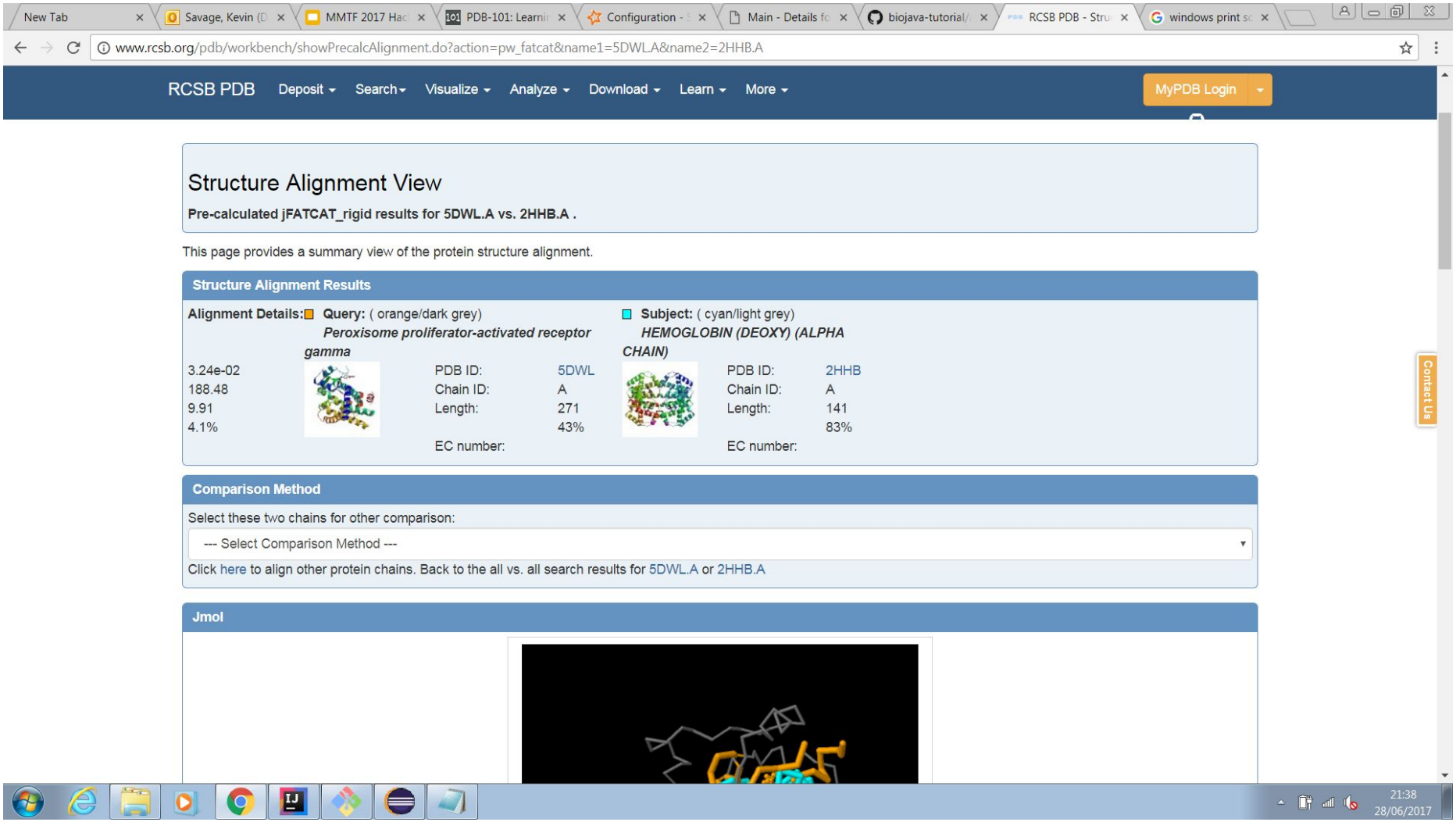
- Goal: Find something in the PDB that is structurally similar to a new protein (e.g. that has just come off the beamline) with 2HHB as example.
- Used Biojava to calculate similarity, Spark to run aver all PDB:
  - AFPChain chain = algorithm.align(atoms1, atoms2);
  - AFPChainScorer.getTMScore(chain, atoms1, atoms2)
- E.g. output:

4JYH.C was similar 0.980416892935969 2

1IRD.A was similar 0.980514190727562 5

5DWL.B was similar 0.998541151781364 5

5DWL vs 2HHB (one of 122 from 1/10th of the PDB)



PDBj Mine 2 filter feature (I)

- MMTF doesn’t contain the full metadata
- Sometimes you want to filter using stuff that isn’t in MMTF, but is in mmCIF
- PDBj’s Mine 2 RDB holds all mmCIF data and more (except atom data ⇒ use MMTF for that!), all accessible via a powerful SQL interface
- Issues:
  - The service is very flexible, so the column names are not predefined (cannot be hardcoded)
    - By default the `pdbid` column is used for MMTF filtering, can be overwritten by the user
    - If you want to filter by pdbid.chainid, you can define it on SQL level and then inform the API to use that column for filtering
  - Newlines aren’t handled properly by Spark’s CSV parser (fixed in 2.2)

PDBj Mine 2 filter feature (II)

```
20  public static void main( String[] args ) throws IOException
21  {
22      // goal: use an sql query to get a list of pdbids, then filter the MMTF-DB to only include those entries
23
24      SparkConf conf = new SparkConf().setMaster("local[*]").setAppName(App.class.getSimpleName());
25      JavaSparkContext sc = new JavaSparkContext(conf);
26
27      String path = System.getProperty("MMTF_FULL");
28      if (path == null) {
29          System.err.println("Environment variable for Hadoop sequence file has not been set");
30          System.exit(-1);
31      }
32
33      // read PDB in MMTF format
34      JavaPairRDD<String, StructureDataInterface> pdb = MmtfReader.readSequenceFile(path, sc);
35
36      String sql = "select pdbid from brief_summary where modification_date >= '2016-06-28' and resolution < 1.5";
37
38      MineSearch search = new MineSearch(sql);
39
40      System.out.println("Number of entries in MMTF 10% library matching query #1: "+pdb.filter(search).keys().count()+"/"+search.dataset.count());
41
42      // Retrieve PDB chain sequences matching to the Pfam accession "PF00046" (Homeobox)
43      // and having a resolution better than 2.0 Angstrom and a sequence length greater than or equal to 58 (residues)
44      // https://pdbj.org/mine/sql?query=SELECT+concat(s.pdbid%2C+%27.%27%2C+s.chain)+as+structureChainId%2C+s.*%2C+r.ls_d_res_high+as+reso%2C%0A+++++LENGTH(p.pdbx_seq_one_letter_code_can)+as+len%2
45      sql =   "SELECT concat(s.pdbid, '.', s.chain) as \"structureChainId\", s.*, r.ls_d_res_high as reso,"+
46              "    LENGTH(p.pdbx_seq_one_letter_code_can) as len, " +
47              "    ('>' || s.pdbid || s.chain) as header, " +
48              "    replace(p.pdbx_seq_one_letter_code_can,E'\n','') as aaseq " + // the replace here is required because spark's csv parser cannot handle newlines properly (should be fixed in 2.2)
49          "FROM sifts.pdb_chain_pfam s " +
50          "JOIN refine r on r.pdbid = s.pdbid " +
51          "JOIN entity_poly p on p.pdbid = s.pdbid " +
52          "AND s.chain = ANY(regexp_split_to_array(p.pdbx_strand_id,',')) " +
53          "WHERE pfam_id = 'PF00046' " +
54              "AND r.ls_d_res_high < 2.0 " +
55          "AND LENGTH(p.pdbx_seq_one_letter_code_can) >= 58 " +
56          "ORDER BY reso, len,s.chain ";
57      MineSearch search2 = new MineSearch(sql, "structureChainId", true);
58      search2.dataset.show(10);
59
60
61      System.out.println("Number of entries in MMTF 10% library matching query #2: "+pdb.flatMapToPair(new StructureToPolymerChains()).filter(search2).keys().count()+"/"+search2.dataset.count());
62
63      sc.close();
64
65  }
```

Gert-Jan Bekker, Osaka-U

PDBj Mine 2 filter feature (III)

```
Number of entries in MMTF 10% library matching query #1: 188/1857
+----------------+-----+-----+----------+-------+----+---+------+--------------------+
|structureChainId|pdbid|chain|sp_primary|pfam_id|reso|len|header|              aaseq|
+----------------+-----+-----+----------+-------+----+---+------+--------------------+
|          4rdu.A| 4rdu|    A|    P56178|PF00046|1.85| 65|>4rduA|GHMVRKPRTIYSSFQLA...|
|          4rdu.D| 4rdu|    D|    P56178|PF00046|1.85| 65|>4rduD|GHMVRKPRTIYSSFQLA...|
|          2hdd.A| 2hdd|    A|    P02836|PF00046| 1.9| 61|>2hddA|MAEKRPRTAFSSEQLAR...|
|          2hdd.B| 2hdd|    B|    P02836|PF00046| 1.9| 61|>2hddB|MAEKRPRTAFSSEQLAR...|
|          2hos.A| 2hos|    A|    P02836|PF00046| 1.9| 63|>2hosA|GSDEKRPRTAFSSEQLA...|
|          2hos.B| 2hos|    B|    P02836|PF00046| 1.9| 63|>2hosB|GSDEKRPRTAFSSEQLA...|
|          1puf.B| 1puf|    B|    P40424|PF00046| 1.9| 73|>1pufB|ARRKRRNFNKQATEILN...|
|          1puf.A| 1puf|    A|    P09631|PF00046| 1.9| 77|>1pufA|NNPAANWLHARSTRKKR...|
|          1e3o.C| 1e3o|    C|    P14859|PF00046| 1.9|161|>1e3oC|EEPSDLEELEQFAKTFK...|
|          3cmy.A| 3cmy|    A|    P23760|PF00046|1.95| 61|>3cmyA|GQRRSRTTFTAEQLEEL...|
+----------------+-----+-----+----------+-------+----+---+------+--------------------+
only showing top 10 rows

Number of entries in MMTF 10% library matching query #2: 2/12
```

Importing validation reports (Przemek Porebski)

Not all residues in crystal structures are created equal. Data analysis should take into account that some residues may not be supported by electron density (erroneously modeled) or be disordered more than others.

The level of “experimental support” can be defined as a correlation between the density calculated from the model and experimental density. Currently this data is calculated by PDB during deposition of the structure and was calculated for all structures that have deposited structure factors. The values as part of validation reports which are available as XML files.

E.g.
http://files.rcsb.org/pub/pdb/validation_reports/qm/2qmo/2qmo_validation.xml.gz

The goal is to get this data available for filtering residues with poor fit to electron density using MMTF-Spark

Importing validation reports (Przemek Porebski)

Approach 1: Use XML files directly as data source

a. Use additional library: https://github.com/databricks/spark-xml
b. Problem: Structure of validation reports is not convenient for using as a data source
c. Resulting schema:

```
|-- _NatomsEDS: long (nullable = true)
|-- _altcode: string (nullable = true)
|-- _avgoccu: double (nullable = true)
|-- _chain: string (nullable = true)
|-- _cis_peptide: string (nullable = true)
|-- _corrupt_record: string (nullable = true)
|-- _ent: long (nullable = true)
|-- _icode: string (nullable = true)
|-- _model: long (nullable = true)
|-- _num-H-reduce: long (nullable = true)
|-- _owab: double (nullable = true)
|-- _phi: double (nullable = true)
|-- _psi: double (nullable = true)
|-- _rama: string (nullable = true)
|-- _resname: string (nullable = true)
|-- _resnum: long (nullable = true)
|-- _rota: string (nullable = true)
|-- _rscc: double (nullable = true)
|-- _rsr: double (nullable = true)
|-- _rsrz: double (nullable = true)
|-- _said: string (nullable = true)
|-- _seq: long (nullable = true)
|-- clash: array (nullable = true)
|    |-- element: struct (containsNull = true)
|    |    |-- _VALUE: string (nullable = true)
|    |    |-- _atom: string (nullable = true)
|    |    |-- _cid: long (nullable = true)
|    |    |-- _clashmag: double (nullable = true)
|    |    |-- _dist: double (nullable = true)
```

Importing validation reports (Przemek Porebski)

Approach 2: Convert XML files to more convenient, curated data structure and dump as parquet files.

```
|-- altcode: string (nullable = true)
|-- chain: string (nullable = true)
|-- clash: double (nullable = true)
|-- icode: string (nullable = true)
|-- model: long (nullable = true)
|-- resname: string (nullable = true)
|-- resnum: long (nullable = true)
|-- rscc: double (nullable = true)
|-- rsr: double (nullable = true)
|-- rsrz: double (nullable = true)
|-- structureId: string (nullable = true)
|-- ligRSRZ: double (nullable = true)
```

Conversion script:

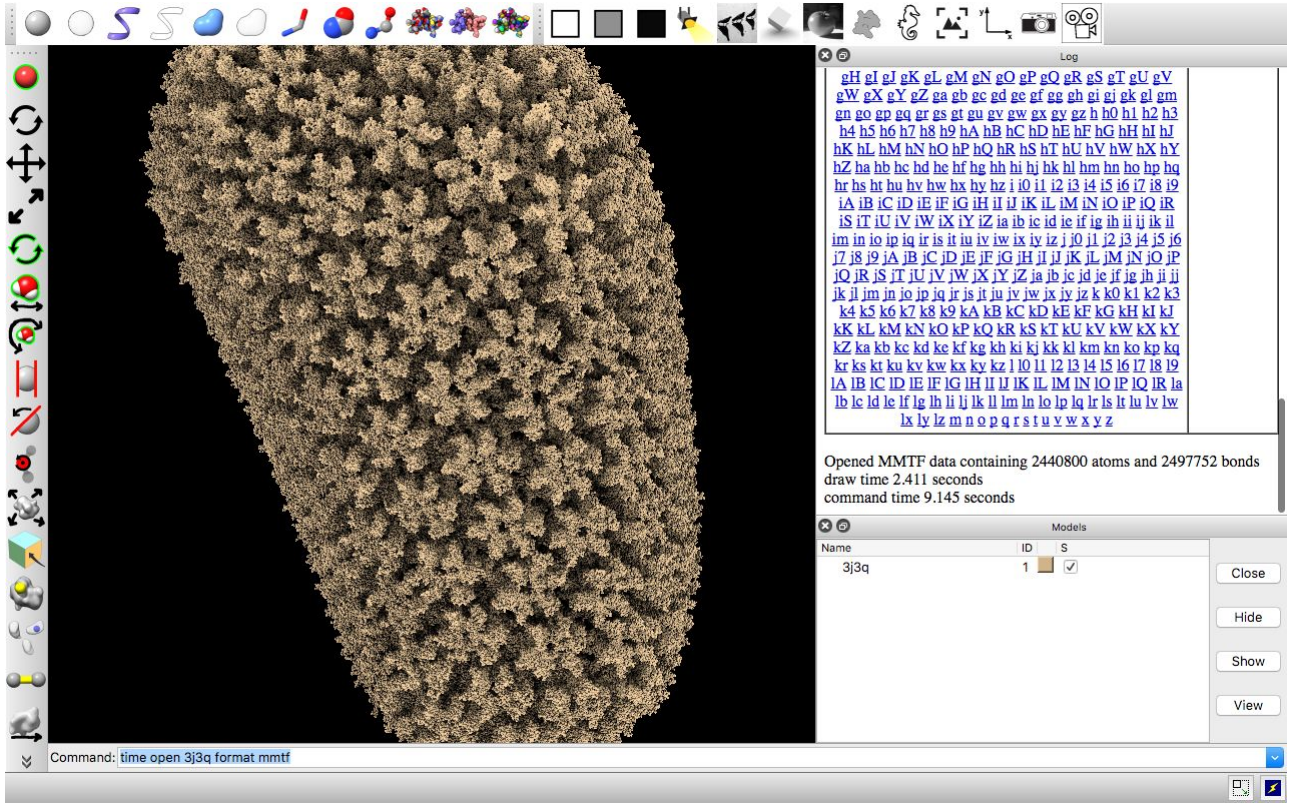
https://gist.github.com/hokinus/5c033fe8f8fcff539b595ae071da337d

For analysis filter out polymer residues with rsrz < 2 (poor fit to electron density or high disorder).

Small molecules with ligRSRZ > 5. (called LLDF) are suspicious, because their fit to electron density is significantly worse than surrounding residue. May indicate lack of evidence for a ligand

Port ChimeraX MMTF Reader to C++

- Greg Couch, UCSF RBVI, http://www.rbvi.ucsf.edu/chimerax/
- Opens 3j3q in ~8 seconds when using C++, didn't wait to find out how slow Python version was (mmCIF is ~14 seconds)
- Overhead in Python version was due to creating Python objects for every atom, bond, etc.



MMTF Feature Request (for Chimera)

- Separate data structure for
  - metal center bonds from mmCIF
  - split out metal bond in chemical components
  - H-bonds from mmCIF
- MMTF spec: add info about case of strings, e.g., entityType
- Add example code to traversal examples that shows how to map chain ids to entities (needed to find out which chains are polymers so a representation of gaps in the structure can be made)

Fragment structural similarity search

- **Goal**: find small protein fragments with similar conformation to a query fragment structure.

- **Solution**:
  a. Create a flatMap function to extract all possible fragments of a polypeptide chain.
  b. Create a similarity measure to compare the fragments with the query.

- **Problems**:
  a. BioJava objects are not serializable.

Sri, Kyle, Aleix

Fragment structural similarity search

```
// Quick hack, the user has to take care of providing that
Group[] query = (Group[]) StructureIO.getStructure("~/Downloads/5epc_fragment.pdb")
        .getChainByIndex(0).getAtomGroups().toArray(new Group[5]);

Double[] phi = new Double[query.length - 1];
Double[] psi = new Double[query.length - 1];

for (int i = 0; i < query.length - 1; i++) {

    double phi_q = Calc.getPhi((AminoAcid) query[i], (AminoAcid) query[i + 1]);
    double psi_q = Calc.getPsi((AminoAcid) query[i], (AminoAcid) query[i + 1]);

    phi[i] = phi_q;
    psi[i] = psi_q;
}
```

Sri, Kyle, Aleix

Fragment structural similarity search

```
JavaRDD<Row> rows = MmtfReader
        //.downloadMmtfFiles(Arrays.asList("5jn5"), sc)
        //.downloadMmtfFiles(Arrays.asList("5jn5", "5tr2", "5f9c", "5hsh"), sc)
        .readSequenceFile(path, sc)
        .filter(new ContainsLProteinChain()) // at least 1 protein chain required
        .flatMapToPair(new StructureToPolymerChains()) // split into polymer chains
        .filter(new ContainsLProteinChain()) // make sure this chain is a protein
        .mapValues(new StructureToBioJava()) // convert to a BioJava structure
        .flatMapToPair(new BioJavaStructureToFragments(query.length))
        .filter(new ConsecutiveFragment())
        .mapToPair(new SimilaryScorer(phi, psi))
        .map(new ResultsDataset());

Dataset<Row> ds = JavaRDDToDataset.getDataset(rows, "pdb","chain","resnum","score");

ds.sort(col("score").asc()).show(100);
```

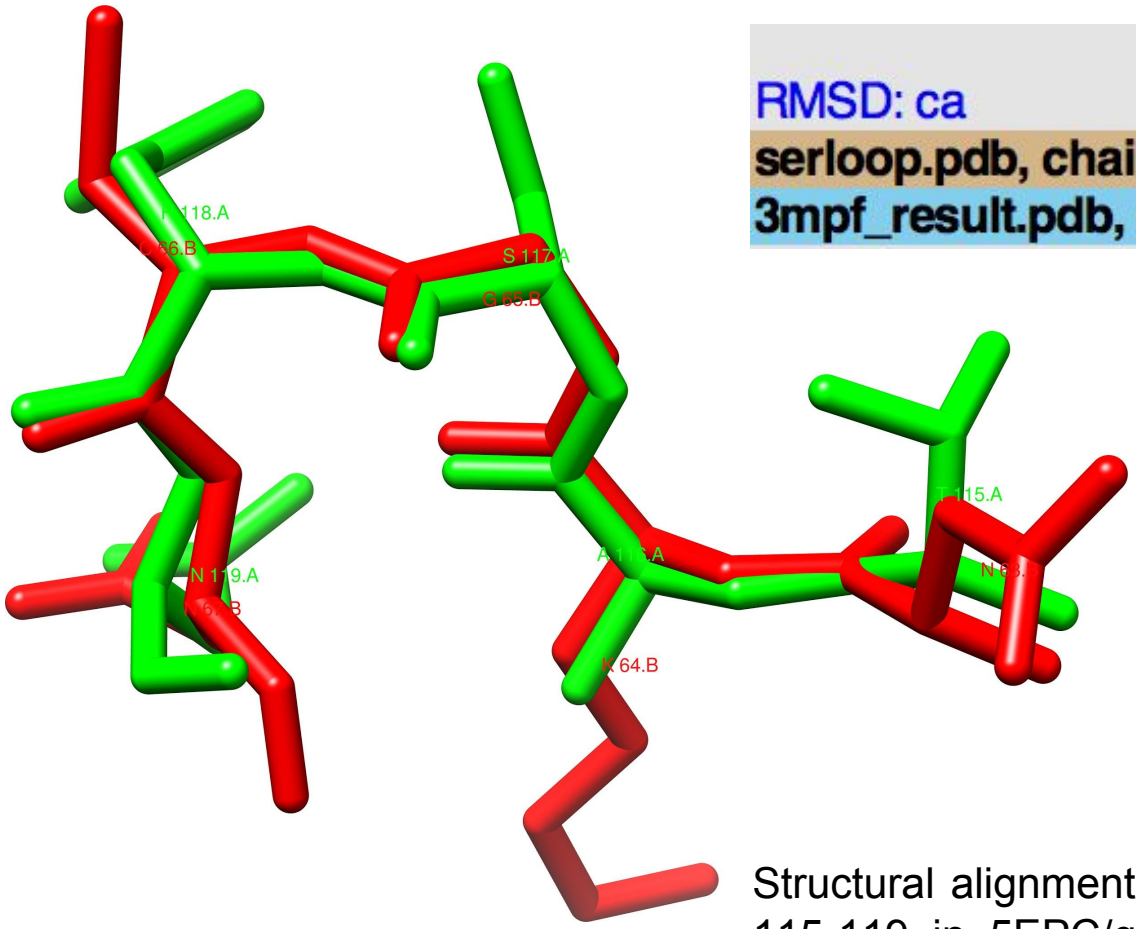
Sri, Kyle, Aleix

Fragment structural similarity search

pdb	chain	resnum	score
5EPC	A	115	6.279857419783 99E-7
5EPC	B	115	0.026708508240375108
3HM2	E	154	0.0271009978099 3716
4T01	A	462	0.027805021962101533
3I3W	A	99	0.029242464875566233
5HCD	A	1653	0.029461464080928607
5HCC	A	1653	0.030920538529235358
3MPF	B	63	0.031284202410822234
4BQV	B	186	0.031292878387673216
1ESE	A	255	0.03139615232496427 4

Sri, Kyle, Aleix

Fragment structural similarity search



		1
RMSD: ca		
serloop.pdb, chain A	115	T A S H N
3mpf_result.pdb, chain B	63	N K G C N

Structural alignment between the query motif (residues 115-119 in 5EPC/green) and one of the search hits (residues 63-67 in 3MPF/red)

Sri, Kyle, Aleix

NGL viewer for matching multiprotein complexes

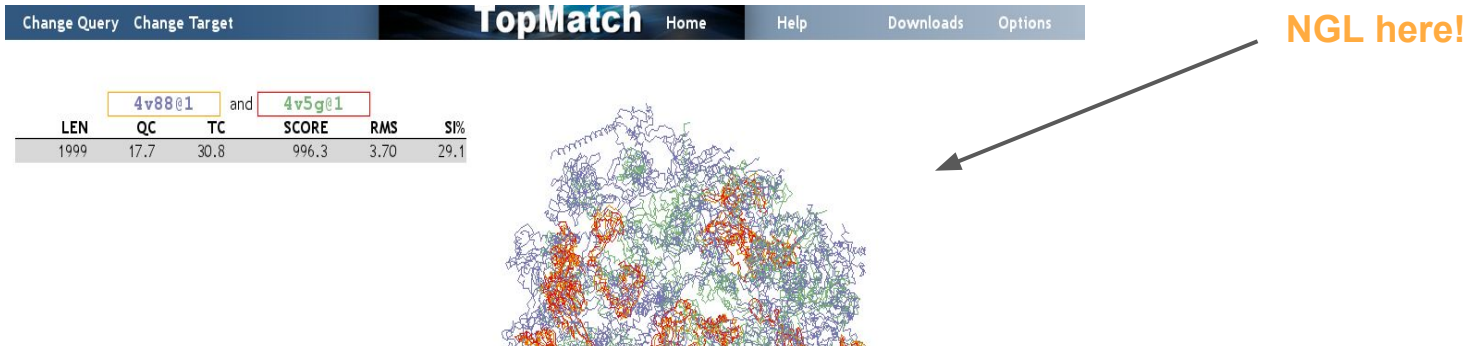
(Markus Wiederstein)

GOAL: Speed up visualization of superposition results from TopMatch/TopSearch for multiprotein complexes
(http://topsearch.services.came.sbg.ac.at)

1. Get two structures (“query” and “target”) into NGL stage.
   Status: *done*
2. Get structure alignment from TopMatch and apply transformation matrix to target.
   Status: *done*
3. Get list of equivalent residues from TopMatch and color respective parts.
   Status: *done* for single chain comparisons, but
   problem: *access to label_asym_id for selecting chains in multiprotein complexes*

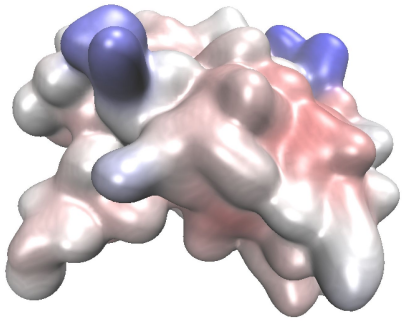
Next steps:

- Switch from `NGL.ColormakerRegistry.addSelectionScheme`
- MMTF writer (Python/C++) for uploaded structures
- Highlighting residues triggered by mouse-over events in sequence alignment widget

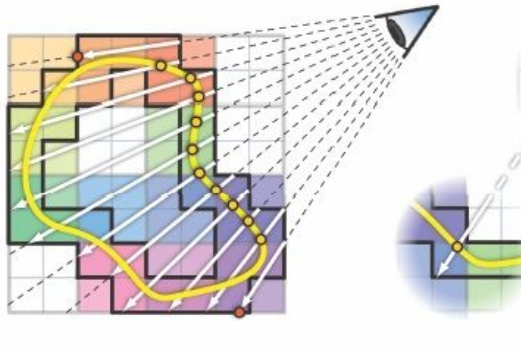


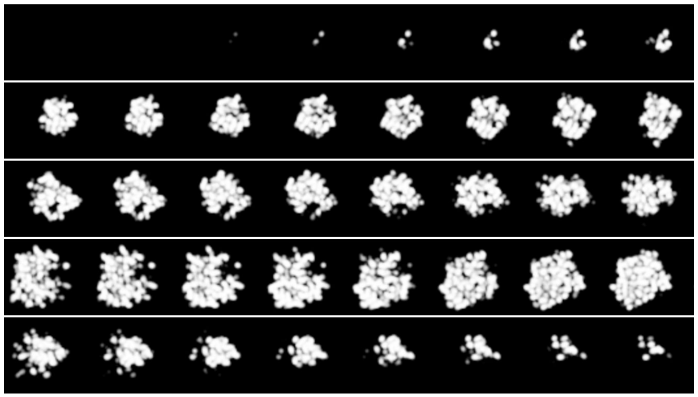
Fast Molecular Surfaces for NGL Viewer

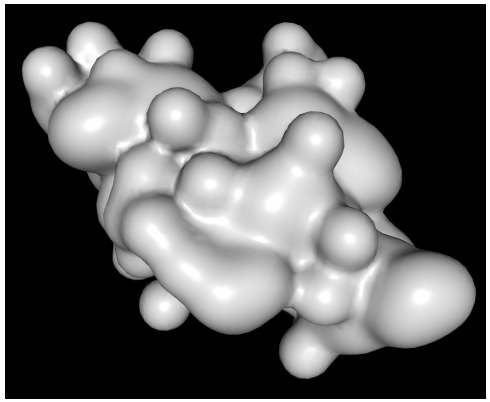
(Alexander Rose, Michael Krone)

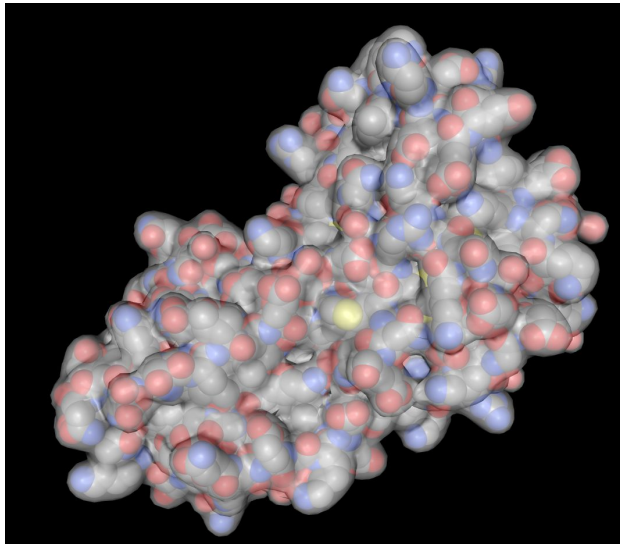


- Volumetric molecular surfaces for NGL Viewer (similar to QuickSurf in VMD)
  - Use WebGL / GPU for fast computation
  - Use REGL API (https://github.com/regl-project/regl)
- Try different rendering methods
  - Marching Cubes isosurfaces / Surface Nets
  - Direct Volume Rendering (isosurface ray marching)
- Missing features / Work in Progress
  - Coloring + volume-based Ambient Occlusion
  - Volume Ray Marching
  - Improve computation & rendering speed
  - Extend this to Solvent Excluded Surfaces?





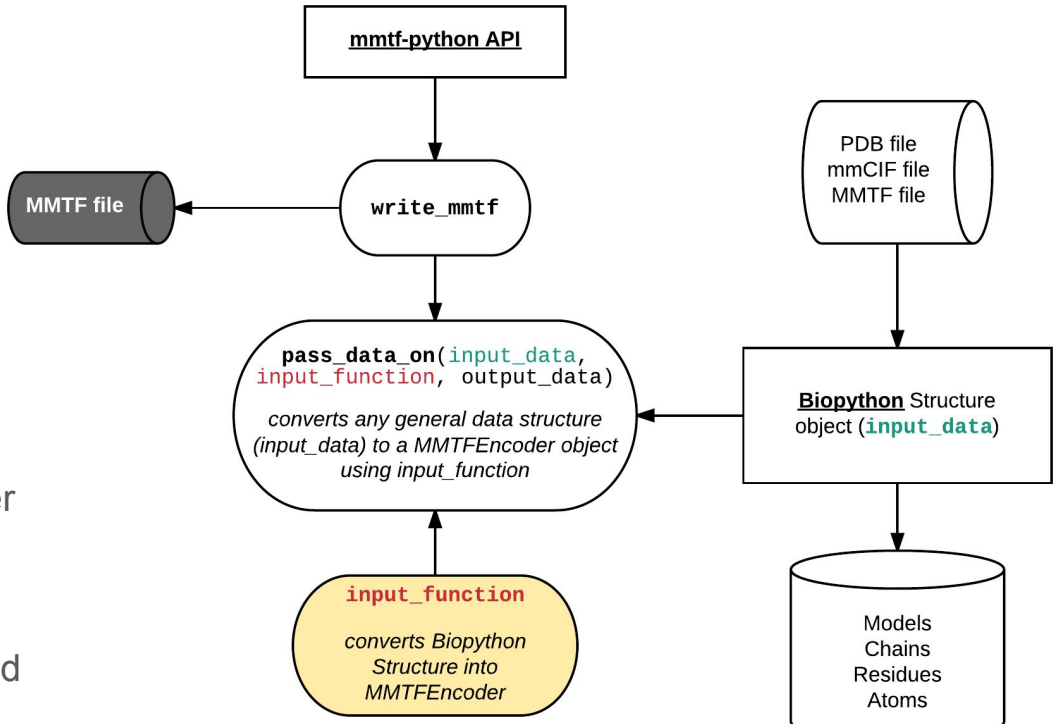




Biopython Structure → MMTF file

(Nathan, Yiling)

- Problem
  - Biopython can read MMTFs but not write them
- Solution
  - Existing mmtf-python API can accept custom write solutions for any data type
  - Write custom function to convert Biopython Structure to MMTFEncoder object
- Problems
  - mmtf-python API incomplete, untested
  - Able to write but not load completed MMTF file
- Missing features
  - Biopython does not parse bioassembly, header, bond counts, etc - to do!

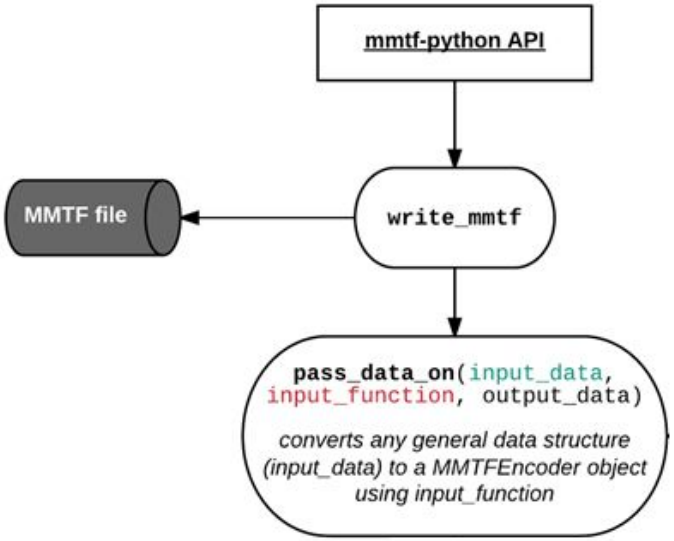



```
def biopython_input_function(bm_struct, biop_encoder):
    """Take in a Biopython Structure (bm_struct) and transform
    it into an MMTFEncoder object (biop_encoder).
    """
```

Write MMTF file

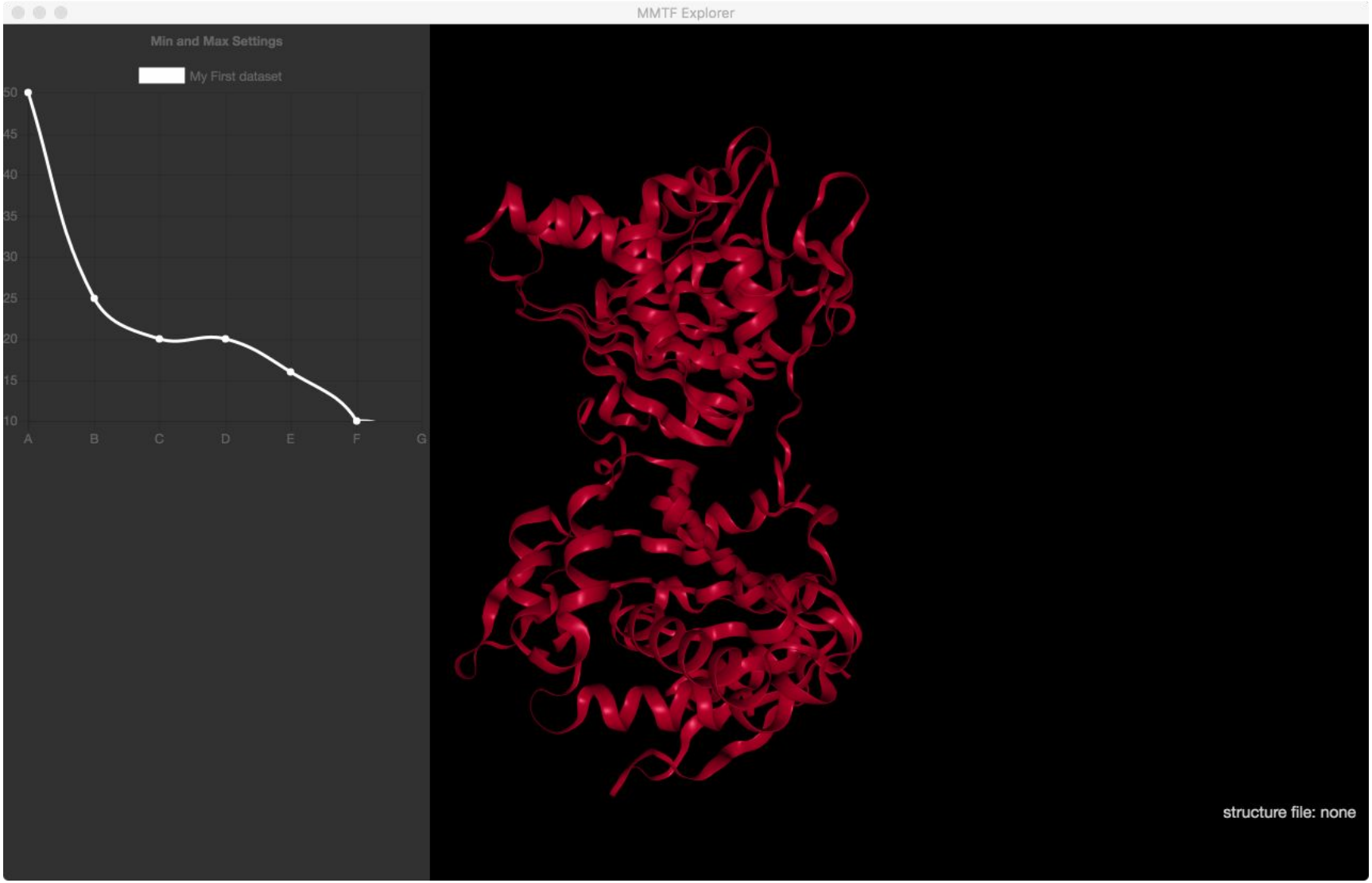
(Nathan, Yiling)

- Problem
  - MMTF-python does not write files
- Solution
  - Existing mmtf-python API can accept custom write solutions for any data type
  - Write custom function to write MMTF files
- Problems
  - mmtf-python API incomplete, untested
- Missing features
  - Atoms and groups




```
#add general
mmtf_encoder.init_structure(input_data.num_bonds, input_data.num_atoms,input_data.num_groups,
    input_data.num_chains,input_data.num_models,input_data.structure_id)

mmtf_encoder.set_xtal_info(input_data.space_group, input_data.unit_cell)

mmtf_encoder.set_header_info(input_data.r_free, input_data.r_work, input_data.resolution,
    input_data.title, input_data.deposition_date, input_data.release_date, input_data.experimental_methods)

mmtf_encoder.set_model_info(input_data.chains_per_model)

#add chain
for i in range(0, len(input_data.chain_name_list)):
    mmtf_encoder.set_chain_info(
        input_data.chain_id_list[i],
        input_data.chain_name_list[i],
        input_data.groups_per_chain)
```

SAXS Client to Spark Server (S. Doutreligne)



A ‘diff’ utility for macromolecular coordinates

Wolfram Tempel

- problem: even insignificant changes in coordinates would confound text-based comparison

- general use case: highlight *significant* differences within pairs of structures

- specific use case: automate annotation of model revisions (Thanks to P. Porebski)

- simplest case: aligned models only differ in coordinates

- added complexity: unaligned models, inconsistent residue ids, types, etc.

- natural potential for parallelization: for each atom in structure 1, find corresponding atom in structure 2, based on distance and atom type/context after alignment

- desired output: list of inter-model differences, sorted by “significance”: missing/additional atoms, mutated residue types, translated coordinates

Identify structural Variation based on SNPs

Alexander seitz

Mapping of IDs:
ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/idmapping/idmapping_selected.tab.gz

Genes (Output Format: gtf)

http://genome.ucsc.edu/cgi-bin/hgTables

(switch “Track” to “TransMapEnsembl”)

```
chr20 43534683 . A G . .
chr20 43534803 . T C . .
```

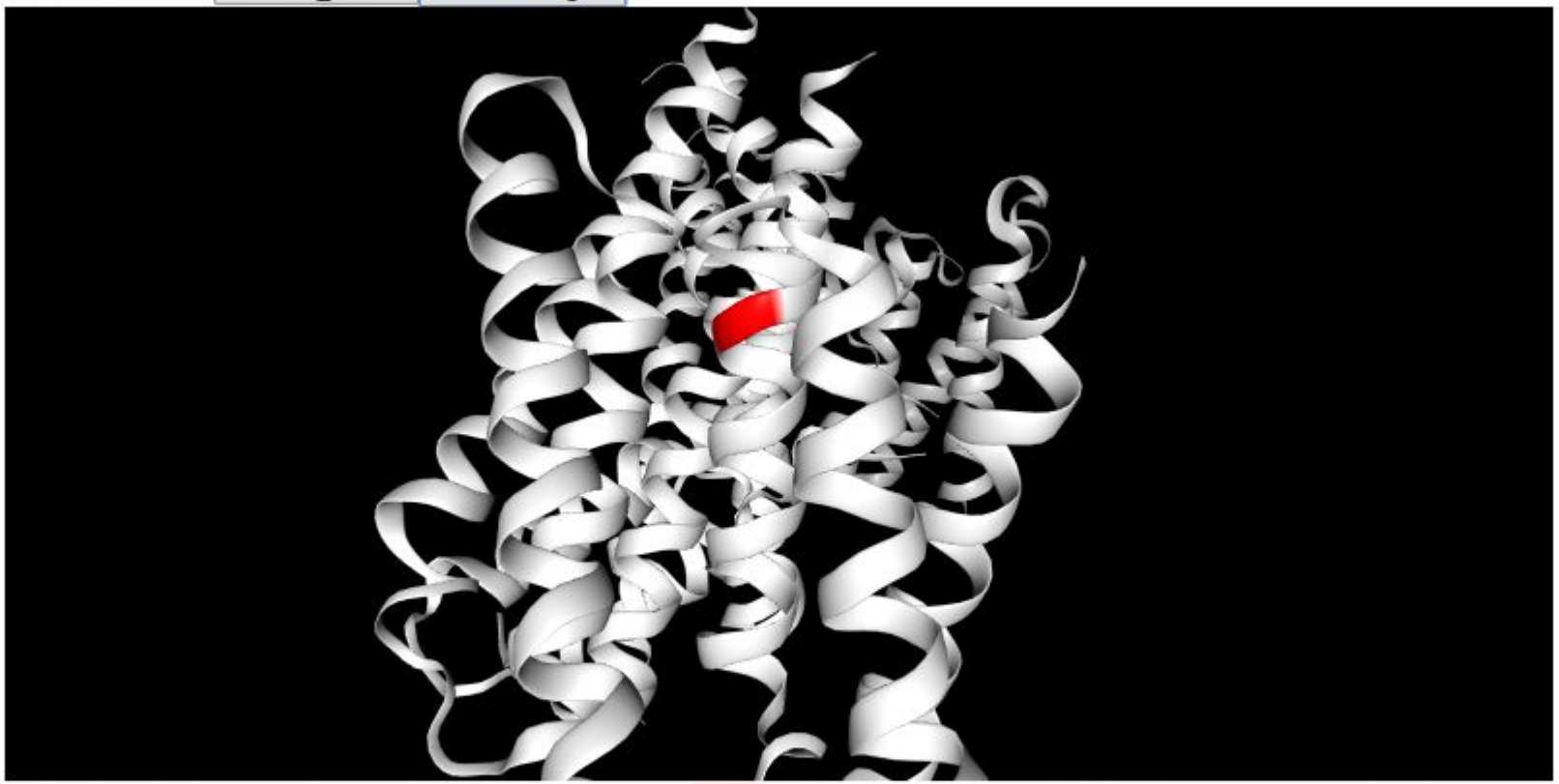
```
chr20 43534683 . A G . . PDB=2C23:A_180!GLY,4DNK:B_187!GLY,2BQ0:
    ↪ A_185!GLY,2BQ0:B_185!GLY,4DNK:A_186!GLY
chr20 43534803 . T C . .
```

Choose Files testInput.vcf_modified.vcf

available IDs: 2C23:A_180 colorChanges

Choose Files testInput.vcf_modified.vcf

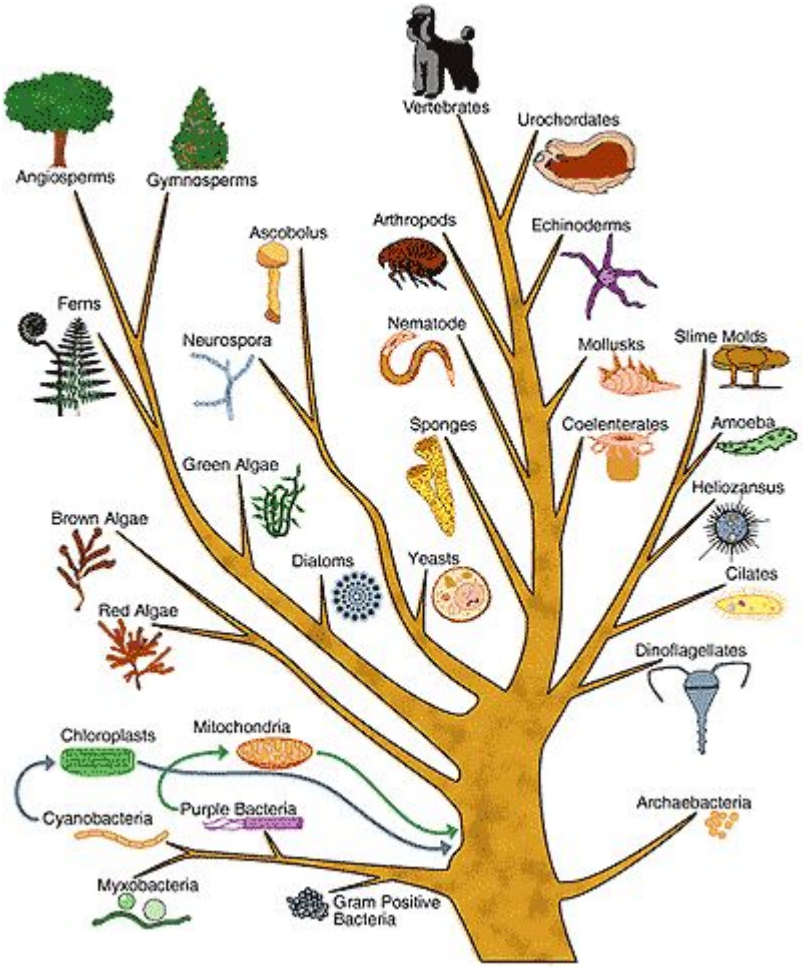
available IDs: 2C23:A_180 ▾ colorChanges



LEU	GLY	LEU	ALA	LEU	ASN	PHE	SER	VAL	PHE	TYR	TYR	GLU	ILE	LEU	ASN	SER	PRO	GLU	LYS	ALA
h	h	h	h	h	h	h	h	h	h	h	h	h	t	s	l	l	h	h	h	h
LEU	GLY	LEU	ALA	LEU	ASN	PHE	SER	VAL	PHE	GLY	TYR	GLU	ILE	LEU	ASN	SER	PRO	GLU	LYS	ALA

Evolution Explorer

- Goal: Analytics friendly Data Frame organization of PDB data with initial emphasis on evolution
- First, organize structures by domains of life
- Next, organize by enzyme classification, transporter classification, GO ontologies, etc.
- Filter structures to only those that have a minimum number of samples from each domain



Chad Bernier

Evolution Perspective

- Scientists interested in early evolution need samples from archaea, bacteria, and eukarya.
- It would be nice to have at least two of each (certainly at least one)
- How many proteins and nucleic acid structures have that kind of diversity?
- What kinds of properties do these samples have in common?
- Is there a statistically significant difference for some properties?

Chad Bernier

Code Snippet

```
List<String> domains = Arrays.asList("Archaea", "Bacteria","Eukaryota");
for (int i = 0; i < domains.size(); i++){
    String domain = domains.get(i);
    String query = "<orgPdbQuery>" + "<queryType>org.pdb.query.simple.TreeEntityQuery</queryType>" +
            "<t>1</t>" + "<n>2157</n>" +
            "<nodeDesc>" + domains.get(i) + "</nodeDesc>" + "</orgPdbQuery>";

    JavaRDD<Tuple3<String,String,StructureDataInterface>> data = MmtfReader
            .readSequenceFile(path, sc)
            .filter(new AdvancedQuery(query))
            .map(t -> new Tuple3<String, String, StructureDataInterface>(t._1, domain, t._2));

    System.out.println(data.count());
    Dataset<Row> ds = PdbDataSet.getDataset(data);
    ds.printSchema();
        ds.show(10);
```

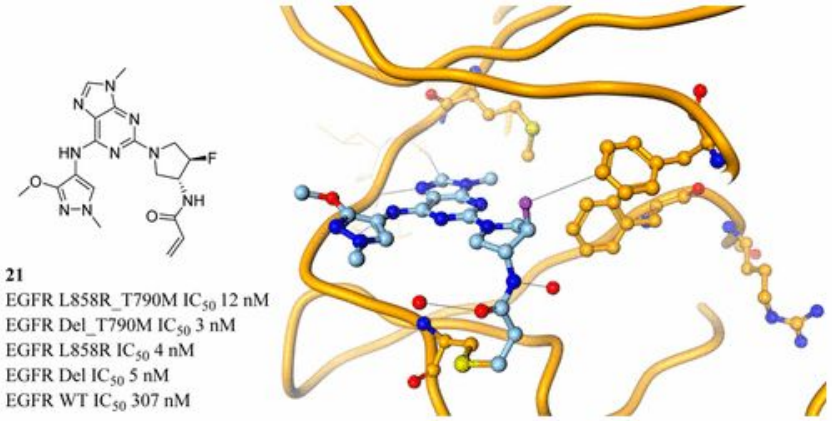
- Major hurdle was just set up. Learning Java, Spark, and everything else.
- Successfully have three datasets in “Data Frame” format, that can be unioned, reduced, aggregated, etc.
- Next step would be to group structures by identity
- Then, collect additional data of interest

Oncogenic mutant structural comparison

1. E.g. EGFR L858R vs EGFR Mutant

   http://www.rcsb.org/pdb/explore/explore.do?structureId=5U8L
   http://www.rcsb.org/pdb/explore/explore.do?structureId=5UG8

2. https://github.com/jjgao/mutantpdb
3. https://jjgao.github.io/mutantpdb/



EGFR	P00533	S	645	C	S645C	Likely Oncogenic	Likely Gain-of-function
EGFR	P00533	H	773	L	H773L	Oncogenic	Gain-of-function
EGFR	P00533	G	810	S	G810S	Likely Oncogenic	Gain-of-function
EGFR	P00533	L	858	Q	L858Q	Likely Oncogenic	Likely Gain-of-function
EGFR	P00533	N	826	S	N826S	Oncogenic	Gain-of-function
EGFR	P00533	L	858	M	L858M	Likely Oncogenic	Likely Gain-of-function
EGFR	P00533	H	773	Y	H773Y	Likely Oncogenic	Likely Gain-of-function
EGFR	P00533	P	596	L	P596L	Likely Oncogenic	Likely Gain-of-function
EGFR	P00533	K	467	T	K467T	Likely Oncogenic	Likely Gain-of-function
EGFR	P00533	G	724	S	G724S	Oncogenic	Gain-of-function
EGFR	P00533	R	836	C	R836C	Likely Oncogenic	Likely Gain-of-function
EGFR	P00533	V	774	A	V774A	Likely Oncogenic	Likely Gain-of-function
EGFR	P00533	L	861	Q	L861Q	Oncogenic	Gain-of-function
EGFR	P00533	R	776	H	R776H	Likely Oncogenic	Gain-of-function
EGFR	P00533	G	719	S	G719S	Oncogenic	Gain-of-function
EGFR	P00533	G	735	S	G735S	Oncogenic	Gain-of-function
EGFR	P00533	N	826	Y	N826Y	Likely Oncogenic	Likely Gain-of-function
EGFR	P00533	L	703	P	L703P	Likely Oncogenic	Likely Gain-of-function
EGFR	P00533	L	747	V	L747V	Likely Oncogenic	Likely Gain-of-function
EGFR	P00533	G	598	V	G598V	Oncogenic	Gain-of-function
EGFR	P00533	L	861	P	L861P	Likely Oncogenic	Likely Gain-of-function
EGFR	P00533	E	734	Q	E734Q	Likely Oncogenic	Gain-of-function
EGFR	P00533	E	868	G	E868G	Likely Oncogenic	Gain-of-function
EGFR	P00533	E	709	G	E709G	Likely Oncogenic	Likely Gain-of-function
EGFR	P00533	E	746	V	E746V	Likely Oncogenic	Likely Gain-of-function

Matt Hudson: https://github.com/mhudson-compbio/
Alexey Strokach: https://github.com/ostrokach/
Daniel Kool: https://github.com/kool7d/
Yana Valasatava: https://github.com/valasatava
Jamaine Davis: https://github.com/jdavislab/
JJ Gao: https://github.com/jjgao

Oncogenic mutant structural comparison

- Goal: Compare structure of single AA mutants
- Steps involved
  - Get a list of oncogenic mutations
  - Map to PDB positions
  - Extract residues from PDB
  - NGL view of mutant to WT structures
- Problems encountered
  - Acquiring the amino acids in the structures based on pdb id, chain and residue number
    - Would be great if MMTF interface can support
  - Error handling when failed to load a structure in Spark

Matt Hudson: https://github.com/mhudson-compbio/
Alexey Strokach: https://github.com/ostrokach/
Daniel Kool: https://github.com/kool7d/
Yana Valasatava: https://github.com/valasatava
Jamaine Davis: https://github.com/jdavislab/
JJ Gao: https://github.com/jjgao

Oncogenic mutant structural comparison

- Use information from mutations.parquet to select required structures
- Create pdb_id and pdb_chain_id HashSets to filter the structures and chains in MMTF_FULL
  - downloadMmtfFiles missing many structures (e.g. 4ZWH, 4ZWK, 5BPR, 5BNU)
  - HashSet works much faster than list (as expected)
- Get 3621 MMTF chains in ~1 second

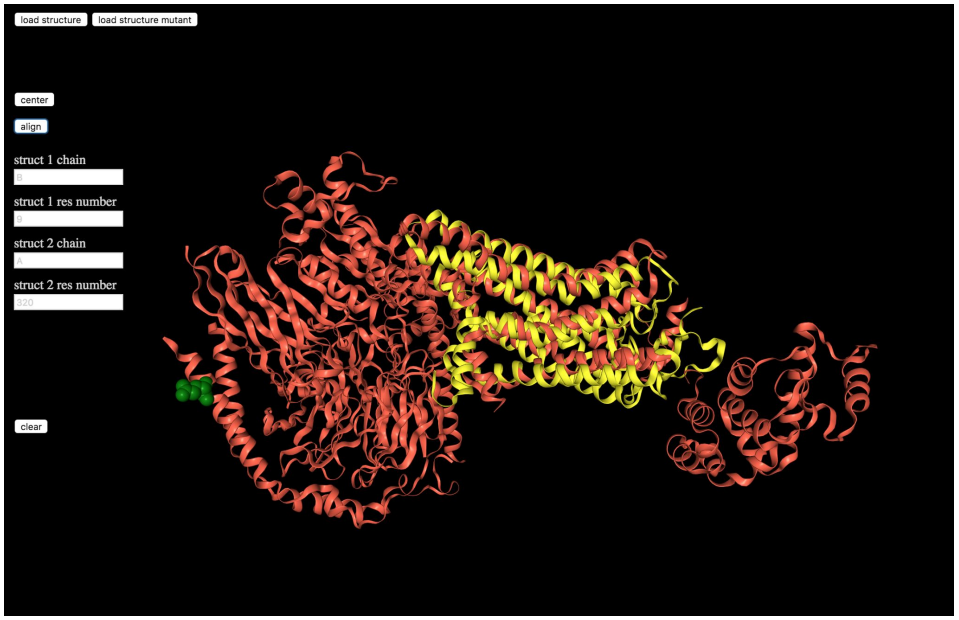
```
+----+-------+---+--------+-------+-------+-------+----------+-----+
|Gene|Uniprot|Ref|Position|Varaint|chainId|insCode|pdbAtomPos|pdbId|
+----+-------+---+--------+-------+-------+-------+----------+-----+
|TP53| P04637|  E|     326|      L|      A|   null|        26| 1A1U|
|TP53| P04637|  F|     328|      V|      A|   null|        28| 1A1U|
|TP53| P04637|  T|     329|      I|      A|   null|        29| 1A1U|
|TP53| P04637|  L|     330|      R|      A|   null|        30| 1A1U|
|TP53| P04637|  L|     330|      H|      A|   null|        30| 1A1U|
|TP53| P04637|  L|     330|      P|      A|   null|        30| 1A1U|
|TP53| P04637|  Q|     331|      H|      A|   null|        31| 1A1U|
|TP53| P04637|  Q|     331|      R|      A|   null|        31| 1A1U|
```

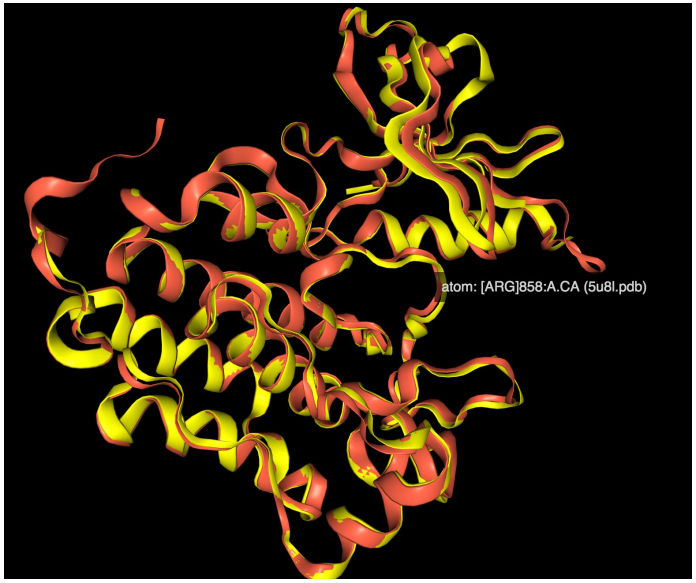
```
JavaPairRDD<String, StructureDataInterface> chains = MmtfReader
        .readSequenceFile(path, sc)
        .filter(t -> pdb_ids.contains(t._1))
        .flatMapToPair(new StructureToPolymerChains())
        .filter(t -> pdb_chains_ids.contains(t._1));
```

Matt Hudson: https://github.com/mhudson-compbio/
Alexey Strokach: https://github.com/ostrokach/
Daniel Kool: https://github.com/kool7d/
Yana Valasatava: https://github.com/valasatava
Jamaine Davis: https://github.com/jdavislab/
JJ Gao: https://github.com/jjgao

Oncogenic mutant structural comparison

https://jjgao.github.io/mutantpdb/





Matt Hudson: https://github.com/mhudson-compbio/
Alexey Strokach: https://github.com/ostrokach/
Daniel Kool: https://github.com/kool7d/
Yana Valasatava: https://github.com/valasatava
Jamaine Davis: https://github.com/jdavislab/
JJ Gao: https://github.com/jjgao

Hydrogen Bonds Dataset (Luiz Borro)

Create a dataset with all hydrogen bonds for a given list of PDB structures

- Three types of Hydrogen Bonds
  - Main Chain - Main Chain
  - Main Chain - Side Chain
  - Side Chain - Side Chain

structureId	chain1	residue1	atom1	element1	index1	chain2	residue2	atom2	element2	index2	distance	InteractionType
10GS	A	TYR	N	N	1	A	PRO	O	O	0	2.2522461	HBond_MM
10GS	A	THR	N	N	2	A	TYR	O	O	1	2.254425	HBond_MM
10GS	A	THR	N	N	2	A	GLN	O	O	54	3.0278215	HBond_MM
10GS	A	VAL	N	N	3	A	THR	O	O	2	2.2466044	HBond_MM
10GS	A	VAL	N	N	3	A	LYS	O	O	27	2.927684	HBond_MM
10GS	A	VAL	N	N	4	A	VAL	O	O	3	2.2475886	HBond_MM
10GS	A	ARG	N	N	11	A	TYR	OH	O	5	3.0740633	HBond_MS

Amino Acids Contacts (Luiz Borro)

Creates a dataset with all hydrogen bonds for a set of protein structures

- Internal and Interface Contacts
- Hydrogen Bonds
    1. Main Chain - Main Chain
    2. Main Chain - Side Chain
    3. Side Chain - Side Chain
    - Aromatic Contacts
    - Charged Interactions
    - Hydrophobic Interactions
    - Disulfide bonds

Water-mediated hydrogen bond?

Amino Acids Contacts (Luiz Borro)

structureId	chain1	residue1	atom1	element1	index1	chain2	residue2	atom2	element2	index2	distance	InteractionType
10GS	A	TYR	N	N	1	A	PRO	O	O	0	2.2522461	HBond_MM
10GS	A	THR	N	N	2	A	TYR	O	O	1	2.254425	HBond_MM
10GS	A	THR	N	N	2	A	GLN	O	O	54	3.0278215	HBond_MM
10GS	A	VAL	N	N	3	A	THR	O	O	2	2.2466044	HBond_MM
10GS	A	VAL	N	N	3	A	LYS	O	O	27	2.927684	HBond_MM
10GS	A	VAL	N	N	4	A	VAL	O	O	3	2.2475886	HBond_MM
10GS	A	ARG	N	N	11	A	TYR	OH	O	5	3.0740633	HBond_MS