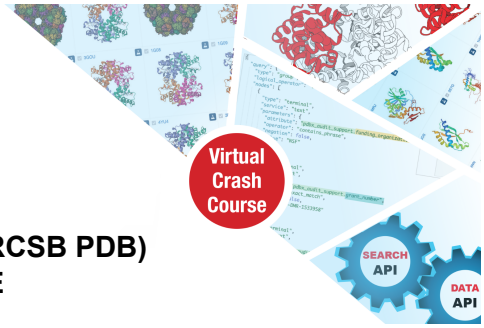

RCSB PROTEIN DATA BANK (RCSB PDB)
API CRASH COURSE

Leveraging RCSB PDB APIs for Bioinformatics Analyses and Machine Learning

Part 1: Introducing RCSB APIs

Thursday, October 12th, 2023
16:00-18:00 UTC

Announcement: https://www.rcsb.org/news/feature/64f9d026d78e004e766a9699

Agenda with quick links to sections in document (times in UTC):

16:10-16:40 - Introduction to RCSB PDB APIs and Data Schema

- Questions from the participants

16:40-17:10 - Data API

- Questions from the participants

17:20-17:50 - Search API

- Questions from the participants

17:50-18:00 - Search and Data API Hands-on Teaser

- Questions from the participants

General Questions from the participants (please find specific sections below):

Q: Will the recording of the session be shared afterwards?

A (Dennis): Yes! It will also be posted on YouTube. Also we will provide links at https://pdb101.rcsb.org/train/training-events

Q: What is API

A: An Application Programming Interface (API) is a set of predefined rules and protocols that allows different software applications to communicate with each other. (https://en.wikipedia.org/wiki/API)

16:10-16:40 - Introduction to RCSB PDB APIs and Data Schema

Speaker: Brinda Vallat, Ph.D. - RCSB PDB, Rutgers University

Synopsis: Introduction to RCSB PDB data and publicly available APIs. The focus will be on Data and Search APIs and the common underlying data schema that supports integration of PDB data with information from external resources to comprehensively deliver structural biology data to users worldwide.

Relevant material:

- RCSB.org web services: https://www.rcsb.org/docs/programmatic-access/web-services-overview
- Introduction RCSB.org APIs: https://pdb101.rcsb.org/learn/guide-to-understanding-pdb-data/introduction-to-rcsb-pdb-apis
- PDBx/mmCIF
    - Documentation: http://mmcif.wwpdb.org
    - Virtual crash course: https://pdb101.rcsb.org/train/training-events/mmcif
- External resources integrated in RCSB.org: https://www.rcsb.org/docs/general-help/data-from-external-resources-integrated-into-rcsb-pdb
- Organization of PDB data:
    - https://data.rcsb.org/index.html#data-organization
    - https://www.rcsb.org/docs/general-help/organization-of-3d-structures-in-the-protein-data-bank
- Identifiers in the PDB: https://www.rcsb.org/docs/general-help/identifiers-in-pdb
- Data schema: https://data.rcsb.org/index.html#data-schema
- Data attributes: https://data.rcsb.org/data-attributes.html

Questions from the participants:

Q: What is API?

A: An Application Programming Interface (API) is a set of predefined rules and protocols that allows different software applications to communicate with each other. (https://en.wikipedia.org/wiki/API)

Q: can you explain definition specific which is just presented

A: Can you please elaborate? Are you interested in mmCIF itself?

Q: yes mmcif

A: (Answered in webinar following Brinda's talk)

Q: how can we figure out which values are controlled vocabularies, and which are free text?

A: The schema language defines what are the controlled vocabularies and what files are defined as strings. The schemas definitions are based on the JSON schema specification.

A: You can obtain JSON schemas and parse it to look for attributes that have enumerations.

Q: For assemblies which have some sort of symmetry, I'm able to see a symmetry axis in the mol viewer on the rcsb website. Is there a way to access the coordinates/direction of this axis programmatically?

A: Yes, the Mol* viewer pulls this information from Data API by providing identifiers for the corresponding assembly. The actual query looks like this:

```
query AssemblySymmetry($assembly_id: String!, $entry_id: String!) {
  assembly(assembly_id: $assembly_id, entry_id: $entry_id) {
    rcsb_struct_symmetry {
      clusters {
        avg_rmsd
        members {
          asym_id
          pdbx_struct_oper_list_ids
        }
      }
      kind
      oligomeric_state
      rotation_axes {
        order
        start
        end
      }
      stoichiometry
      symbol
      type
    }
  }
}
```

Q: Would there be changelogs for those APIs schemas ?

A: For Search API: https://search.rcsb.org/#changelog. We work on sharing a changelog for Data API.

Q: Is the polymer_entity_instance id the PDB chain id or the asym id?

A: Both programmatically assigned identifiers (label_asym_id) and author-assigned identifiers (auth_asym_id) are provided by most APIs as response. But in general, mmCIF-based label_asym_id are used when communicating with APIs, e.g. to request data.

A: Yes, the PDB provided chain identifier (label_asym_id) is used to assign the "rcsb_id" but the corresponding "auth_asym_id" can be easily obtained. The mapping is preserved.

Q: Hi the slide said the output file will be in .json (it's java based?) so I am wondering if it will be able to work with python notebooks?

A: JSON is a text-based data interchange format and is agnostic to programming languages. All programming languages have libraries to work and parse JSON data.

Q: Are you planning to rename chain IDs? Some chains are computationally hard to parse (question marks, semi-colons, etc…).

A: PDB assigns chain IDs automatically while processing an entry. These are provided in the attribute "label_asym_id". Author-provided chain ids are preserved in the attribute "auth_asym_id". "label_asym_id" is standardized and will not have question marks or semicolons. This is in PDBx/mmCIF and not in PDB format files. We recommend using PDBx/mmCIF files.

Q: What if an (old) entry has no Assembly ?

A: We remediate old entries and recently added assembly information for all entries.

Q: For NMR structures can I fetch only 1 out of many solutions?

A: Coordinates are available for all deposited models in the mmCIF file. APIs only provide information on the 1st/representative model of an NMR structure. The same is true for any type of similarity search that we will talk about later: Only the 1st model is considered.

Q: what is the asym id?

A: The asym ID is the chain ID ("asym" stands for "asymmetric")

16:40-17:10 - Data API

Speaker: Jose Duarte, Ph.D. - RCSB PDB, UCSD

Synopsis: Introduction to the Data API with its 2 interfaces: REST and GraphQL. How to make the most of GraphQL querying and finding your way through the schema.

Relevant material:

- Tutorial and many examples: https://data.rcsb.org
- The full list of data attributes: https://data.rcsb.org/data-attributes.html
- The schema browser in GraphiQL: https://data.rcsb.org/graphql/index.html ("Docs" link on top right)
- Reference for REST endpoints: https://data.rcsb.org/redoc/index.html
- Examples:
    - Title, experimental method and resolution and Rfree for some entries
    - Organisms and cluster membership of polymeric entities
    - Annotations at chain level (e.g. CATH or SCOP)
    - Data associated to a Computed Structure Model (e.g. pLDDT)
    - Interface properties for a certain assembly

Questions from the participants:

Q: Is there a limit on a number of entries that could be retrieved from graphql with a single request?

A: We recommend doing batches (of ~1000 IDs at once), as some large queries might be too heavy.

Q2: How fast would the API in responding to a request with 1000 IDs? Could it some cases these request timeout?

A: The speed depends on the amount of data you are requesting for each ID. Yes the request can time out if you are requesting too much information at once.

A: Data API mainly provides static data, which is cheap to do. However, certain queries can request a lot of information (e.g. mappings between label & auth identifiers for chains/residues). In that case it is helpful to reduce the batch size. Long-running queries will timeout after 1-2 minutes depending on the API.

Q: Is there a date associated with annotations integrated from external resources? Are these integrated annotations updated?

A: Most external resources get updated on a weekly basis. But there's no explicit date associated with all annotations.

A: In some cases, "version" information is provided for external annotations.

Q: Will there be VS code integration for Graph/QL?

A: I haven't used it but there seems to be e.g. https://marketplace.visualstudio.com/items?itemName=GraphQL.vscode-graphql. In general, both GraphQL and VS Code tooling tends to be really good.

Q: Is the polymer_entity_instance id the PDB chain id or the asym id?

A: Both programmatically assigned identifiers (label_asym_id) and author-assigned identifiers (auth_asym_id) are provided by most APIs as response. But in general, label_asym_id are used when communicating with APIs, e.g. to request data.

A: Yes, the PDB provided chain identifier (label_asym_id) is used to assign the “rcsb_id” but the corresponding “auth_asym_id” can be easily obtained. The mapping is preserved.

Q: Is graphql also RESTful?

A: GraphQL is not RESTFul. It is a query language for the API. However, the API allows you to pass the GraphQL query as a URL parameter.

Q: Analysing the interface area you can retrieve the sequence or numbers of the residues involved ?

A: You can get the residues identifiers using this query

```
{
  polymer_entity_instances(instance_ids: ["2UZI.C"]) {
    rcsb_id
    rcsb_polymer_entity_instance_container_identifiers{
      auth_to_entity_poly_seq_mapping
    }
  }
}
```

Then, you can get the ASA of bound and unbound interface partners conformation with the request

```
query QueryInterfaceInstance{
  interfaces(
    interface_ids:["2UZI-1.3"]
  ){
    rcsb_interface_container_identifiers{
      rcsb_id
      interface_id
      assembly_id
      entry_id
    }
    rcsb_interface_partner{
      interface_partner_identifier{
        asym_id
        entity_id
      }
    }
    rcsb_interface_operator
                    rcsb_interface_partner {
     interface_partner_feature {
      name
      feature_positions {
       end_seq_id
       values
      }
     }
```

```
      }
    }
  }
```

You can derive the interface by changes on residue ASA between bound and unbound. To find the interface Ids for a given assembly use next query

```
query QueryAssemblyInterfaces {
  assemblies(
    assembly_ids: ["2UZI-1"]
  ){
    rcsb_id
    interfaces{
      rcsb_id
    }
  }
}
```

Q: How do I find what field has the data I want?

A: Query by example, GraphiQL schema browser and contextual help (e.g., click on “Data API” gear icon from a structure summary page—see next to “Download Files” on https://www.rcsb.org/structure/8IKR)

A: You can also look at the graphql schema explorer (https://data.rcsb.org/graphql/ under “Docs”, right hand top corner) to find which object and attributes contain the data that you want.

Q: Does the request allow for filtering?

A: No. Filtering must be done by consumer

Q: How do I get data for the whole archive?

A: Holdings REST endpoint and GraphQL queries by batches

Q: Can we get all the information in “Tabular report” by using GraphQL API?

A: Transformation into tabular reports should be done by the consumer. However, there is no easy and general way to transform documents into tables.

A: The JSON objects can be nested and therefore not easy to convert to tables.

Q: Is the organism information (_entity_src_gen.gene_src_common_name) available from API? For example, PDB ID 6VXX

A: Yes you can do that by connecting entries with entities. Query example:

```
query {
 entries(entry_ids: ["6VXX"]) {
  rcsb_id
  polymer_entities{
   rcsb_id
   rcsb_entity_source_organism {
    scientific_name
```

```
    }
   }
  }
}
```

Q: Do chemical components have an instance id? For example, 4HHB has HEM with asym_id E. So is HEM.E a valid instance id?

A: You would want to address instances of a non-polymer by entry_id and its asyn_id. A Data API query could e.g. look like this:
https://data.rcsb.org/graphql/index.html?query=%7B%0A%20%20nonpolymer_entity_instance(entry_id%3A%20%224HHB%22%2C%20asym_id%3A%22E%22)%20%7B%0A%20%20%20%20rcsb_id%0A%20%20%20%20rcsb_nonpolymer_instance_feature_summary%20%7B%0A%20%20%20%20%20%20comp_id%0A%20%20%20%20%20%20count%0A%20%20%20%20%20%20maximum_length%0A%20%20%20%20%20%20maximum_value%0A%20%20%20%20%20%20minimum_length%0A%20%20%20%20%20%20minimum_value%0A%20%20%20%20%20%20type%0A%20%20%20%20%7D%0A%20%20%7D%0A%7D%0A

Q: related question… Is it possible to retrieve chains that are bound to a specific ligand (say I'm interested in a trimeric protein, and want to know how many monomers have an Mg ion bound)?

A: You could query for all entries that contain Mg like this. However, for full control it's probably best to run some local script on this pre-filtered data set.

Q: When is the whole AlphaFold database going to be integrated?

A: We are working on scaling our services to integrate more Computed Structure Models that are of interest to the community.

Q: On the website, I can get the search output in the table format even if the data is nested. Is there information available on how that table is generated from the json output of data api?

A: In case you're talking about the Tabular Reports feature: It's best to fetch your data of interest directly from Data API because the tabular representation is opinionated especially for nested properties. Better to transform the data yourself to fit your actual needs (in contrast to the general assumptions made by the Tabular Reports representation).

Q: I was asking from a visualization point of view, the json output is harder to visualise but the table is more suitable for this purpose. But, as you said, the nested json is really hard to convert to the table format, which is the exact problem I'm facing. But, on the website, the tabular report option seems to convert the output to table. I wanted to know how, so I can use the similar method to convert my json output to a tabular format (just for the visualisation). (Sorry for the long explanation)

A:

Q: Have DATA RCSB annotations a date of when where associated?

A: No, but some include version information from external (or internal) resources.

Q: Using the Data API, how can I get the FASTA sequence (i.e.: protein) from the PDB ID? Example: PDB ID 1WQ5. What is the format of the endpoint?

A: https://www.rcsb.org/fasta/entry/4HHB/display

A: You can use the “pdbx_seq_one_letter_code” and “pdbx_seq_one_letter_code_can” attributes in the “entity_poly” object to get the one letter sequence. See query below.

```
query structure ($id: String!) {
  entry(entry_id:$id){
        rcsb_id
        polymer_entities {
        entity_poly {
        pdbx_seq_one_letter_code
        pdbx_seq_one_letter_code_can
        }
      }
  }
}
```

If you need it specifically in FASTA format, it can be downloaded from the entry pages.

17:20-17:50 - Search API

Speaker: Yana Rose, Ph.D. - RCSB PDB, UCSD (yana.rose@rcsb.org)

Synopsis: We will dive into the RCSB PDB Search API's capabilities. You’ll learn how to utilize advanced query options tailored to the needs of structural bioinformaticians.

Relevant material:

- Tutorial: https://search.rcsb.org/
- API reference documentation (UI): https://search.rcsb.org/redoc/index.html
- API reference in OpenAPI Specification (https://swagger.io/specification): https://search.rcsb.org/openapi.json
- API endpoint for data schema: https://search.rcsb.org/rcsbsearch/v2/metadata/schema (‘searchable’ attributes will have `` `rcsb_search_context` `` metadata)
- Search Attributes tutorial (https://search.rcsb.org/#search-attributes):
    - https://search.rcsb.org/structure-search-attributes.html
    - https://search.rcsb.org/chemical-search-attributes.html
- Examples: https://search.rcsb.org/#examples

Questions from the participants:

Q: How is the score calculated?

A: It depends on the type of search performed. Attribute searches are true or false (so 1 or 0). Other searches follow a complicated algorithm. We use Elasticsearch as the tool for ranking search

results. See our documentation for more information:
https://www.rcsb.org/docs/search-and-browse/basic-search#relevancy-scoring

Q: Search by motif: does the score directly relate to how many times the motif was found?

A: For sequence motif search, the score is “binary”, it either matches or the query doesn’t match, meaning all scores are uniformly 1.0. For structure motif search, it’s the RMSD of the alignment between query and detected motif.

Q: How does the search by structure differ from Foldseek?

A: RCSB.org structure search is based on Biozernike descriptors (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7371193/). This representation allows us to compare volumes derived from structures. One of the advantages of this method is enabling the search for assemblies and not only for single chains (PDB Instances).

git repo: https://github.com/biocryst/biozernike

A: Note that an important difference with Foldseek is that it will do global shape matching only. It will not find structures that are only similar locally (i.e. for some subdomain only)

Q: Is there any email, RSS or Atom feed on both APIs so the community can be informed of changes or additions?

A: api@rcsb.org - APIs mailing list

A: to subscribe go to: https://groups.google.com/a/rcsb.org/g/api/about?pli=1

Q: Can we have a functionality in search API as to search same ligand with various available proteins? (kind of like kinome scanning)

A: You can look into Chemical Similarity Search:
https://www.rcsb.org/docs/search-and-browse/advanced-search/chemical-similarity-search

Q: Can all the links from the presenters’ slides be copied to this document?

A: Sure thing! Please, find them in **Relevant material** section above

Q: How can i obtain the FASTA sequence (i.e. protein) based in PDBID ?

A: https://www.rcsb.org/fasta/entry/4HHB/display

A: You can use the “pdbx_seq_one_letter_code” and “pdbx_seq_one_letter_code_can” attributes in the “entity_poly” object to get the one letter sequence. See query below.

```
query structure ($id: String!) {
  entry(entry_id:$id){
        rcsb_id
        polymer_entities {
        entity_poly {
        pdbx_seq_one_letter_code
        pdbx_seq_one_letter_code_can
        }
      }
  }
```

}

If you need it specifically in FASTA format, it can be downloaded from the entry pages.

Q: Might be outside scope, but are there coding examples or published research integrating the RCSB PDH Search API with Graph Neural Network for biomedical research and clinical analysis? …. I have just heard that AI models will be covered in Part 2 :)

A: One possible example is https://onlinelibrary.wiley.com/doi/full/10.1002/pro.3730

17:50-18:00 - Search and Data API Hands-on Teaser

Speaker: Dennis Piehl, Ph.D., RCSB PDB, Rutgers University

Synopsis: Quick walkthrough of an example use case/pipeline that ties together everything presented above—schemas, search API, and data API. The specific goal demonstrated here is to fetch all the citation information associated with structures of insulin.

Relevant material:

- Example pipeline:
  - Goal: Get the citation information for structures of insulin
  - Search API query
  - Data API query
    - Pro tip on GraphiQL shortcuts:
      - Auto Complete: Ctrl-Space (or Option-Space)
      - Run query: Ctrl-Enter
      - Format query: Ctrl-Shift-P
- RCSB PDB Search API Python package: https://rcsbsearchapi.readthedocs.io/en/latest/
  - Offers Pythonic interface to RCSB PDB Search API
  - Supports all types of searches
  - Quickstart tutorial: https://rcsbsearchapi.readthedocs.io/en/latest/quickstart.html
- References and documentation:
  - List of possible attributes: https://www.rcsb.org/docs/search-and-browse/advanced-search/attribute-details
  - Search API query schema: https://search.rcsb.org/redoc/index.html#tag/Search-Service
    - After navigating to the page, click on “json” to open up the list of query parameters, and continue clicking on sub-items to expand further.
- Requirements for next crash course (Part II):
  - Registration! (Links will be sent after today’s event)
    - If you don’t receive an invite after a few days, contact info@rcsb.org
  - Familiarity with Python basics
  - Google account (for accessing a Google Colab notebook)
  - Questions and real use cases that you wish to investigate
    - Provide these in the Exit Survey after today’s course, if possible!

Questions from the participants: