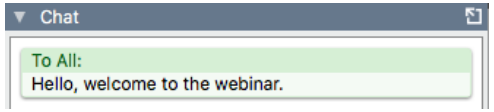
Welcome - webinar instructions

- GoToTraining works best in **Chrome** or on Linux, **Firefox**
- To access the full features of GoToTraining, use the desktop version by clicking **“switch to desktop version”**
- All **microphones will be muted** whilst the trainer is speaking
- If you have a question please use the **chat box** at the bottom of the GoToTraining box



- Please complete the **feedback survey** which will launch at the end of the webinar.
- This presentation is available in the materials section

PDBe API webinar series:
2) Searching with the PDBe API

PDBe.org/API

John Berrisford

pdbhelp@ebi.ac.uk
proteindatabank
@PDBeurope
pdbeurope
pdbart

PDBe
Protein Data Bank in Europe

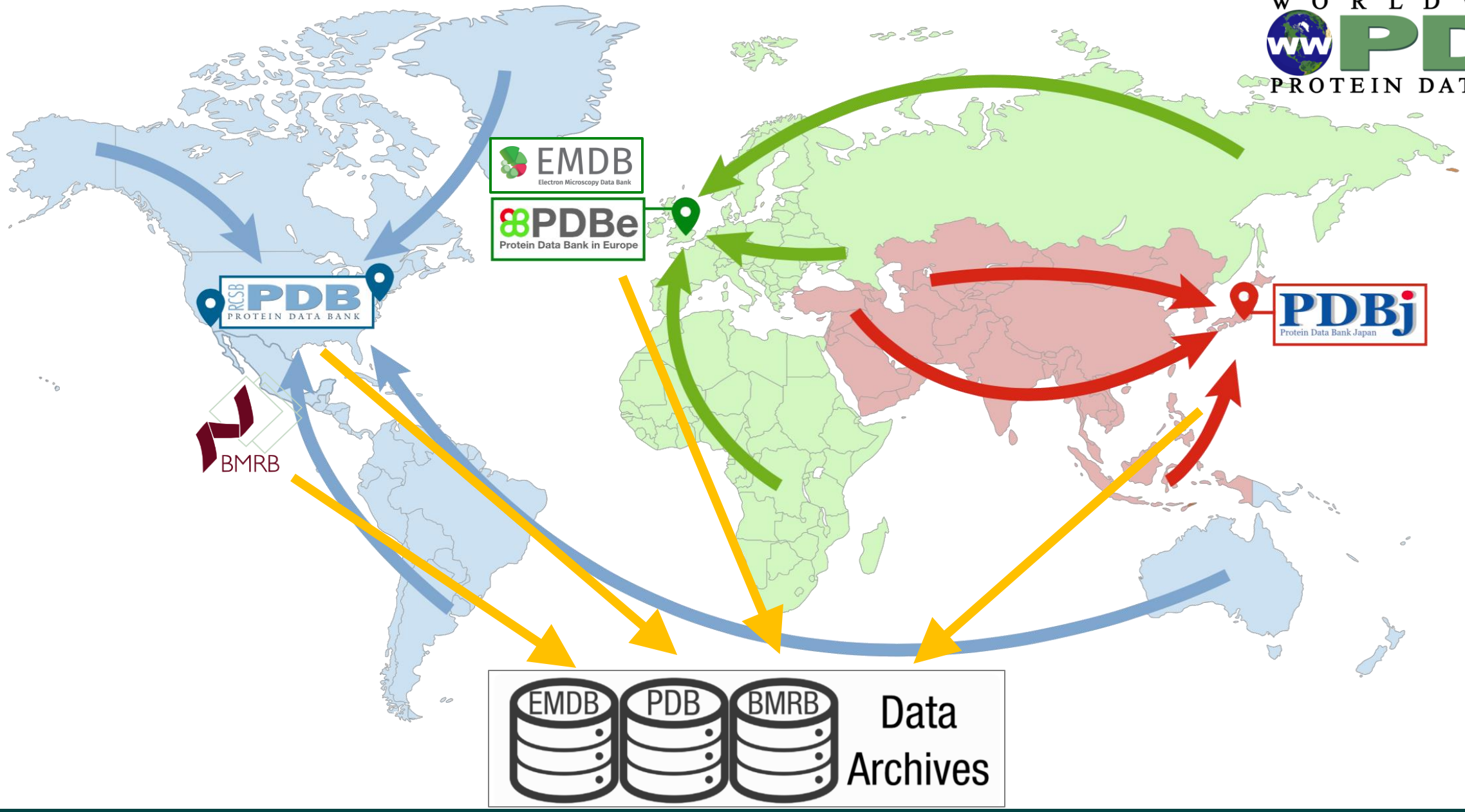
EMBL-EBI

PDBe API webinar series

Sep 15th	Introduction to PDBe programmatic access
Sep 22nd	Searching with the PDBe API
Sep 29th	Creating complex PDBe API queries
Oct 6th	Using the PDBe graph API
Oct 13th	PDBe tools in github
Oct 20th	Data visualisation at PDBe

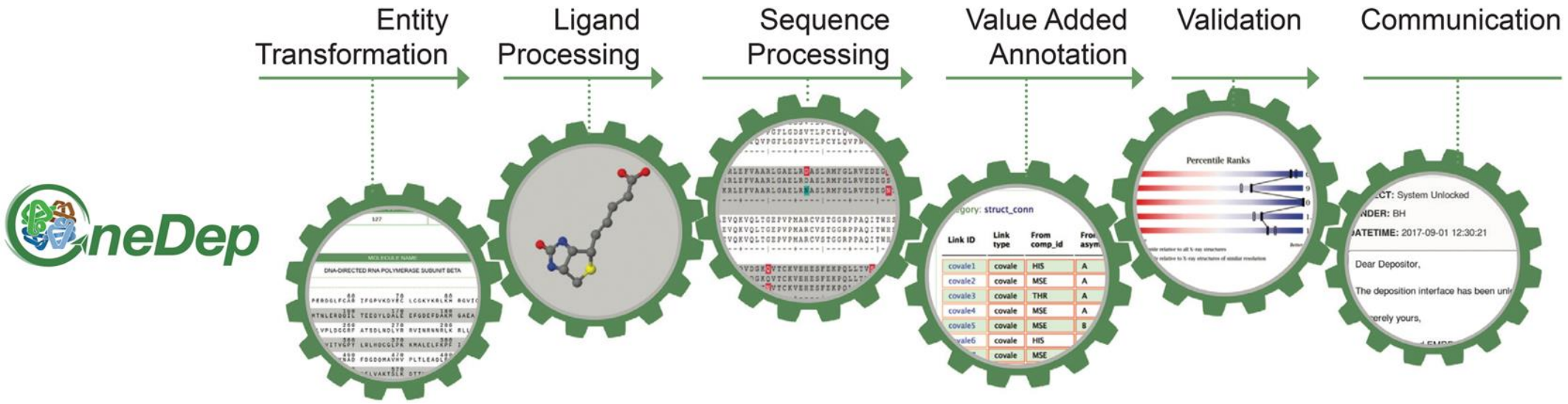
Summary of the webinar

- PDBe in the wwPDB
- PDBe search
- Advanced search
- Introduction to documentation page
- Usage examples

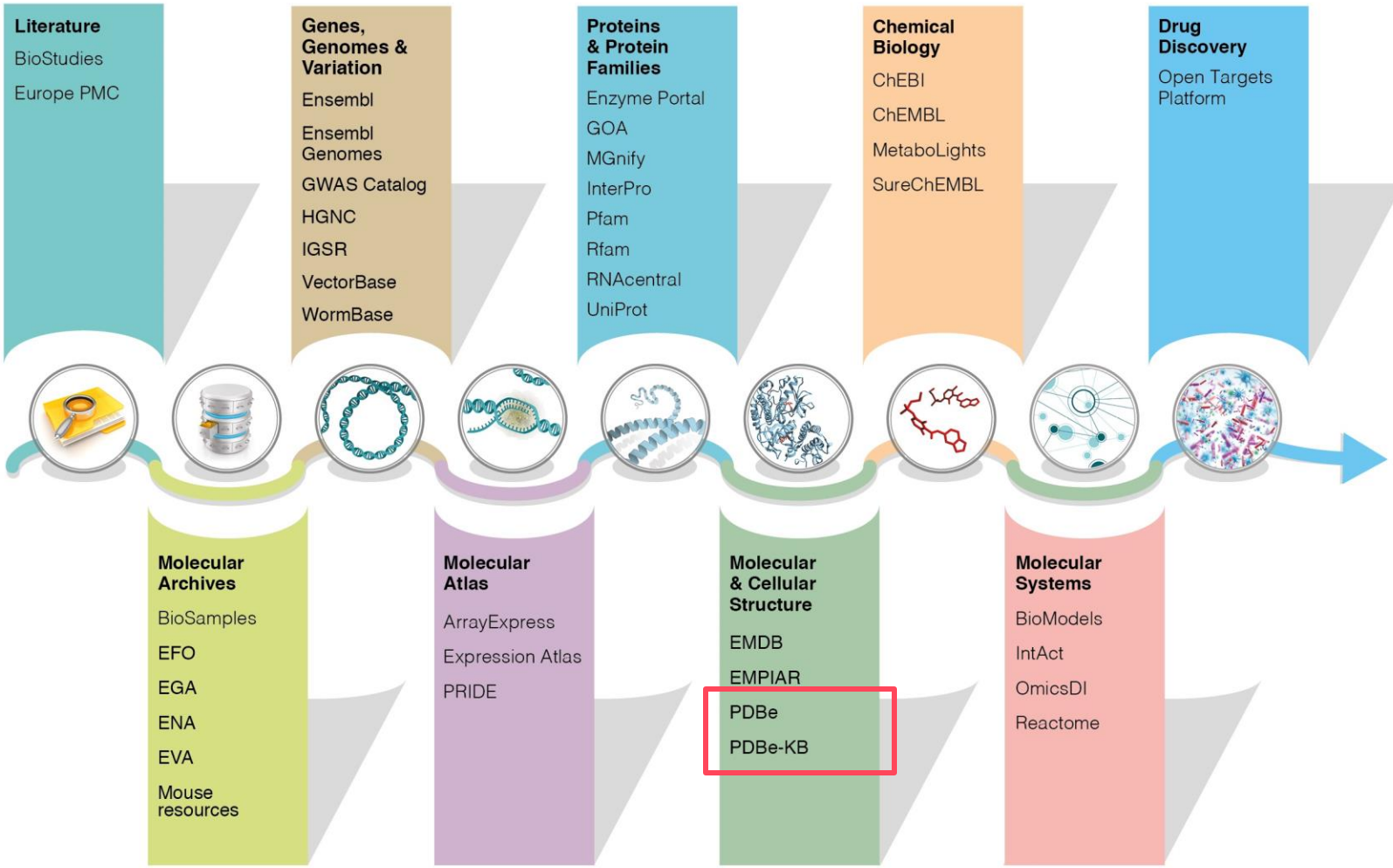

WORLDWIDE PDB
PROTEIN DATA BANK
EMDB
Electron Microscopy Data Bank
PDBe
Protein Data Bank in Europe
RCSB PDB
PROTEIN DATA BANK
PDBj
Protein Data Bank Japan
BMRB
EMDB
PDB
BMRB
Data Archives

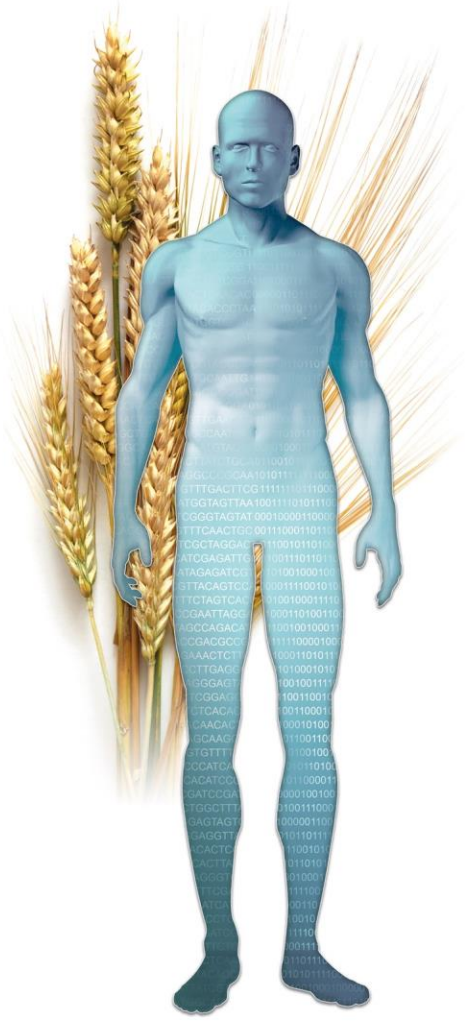
Data added through deposition

- Provided by depositor or added during curation
- Data added to define experimental methods, sequence information, connectivity, validation and more…
- Added to the archive mmCIF file or validation XML



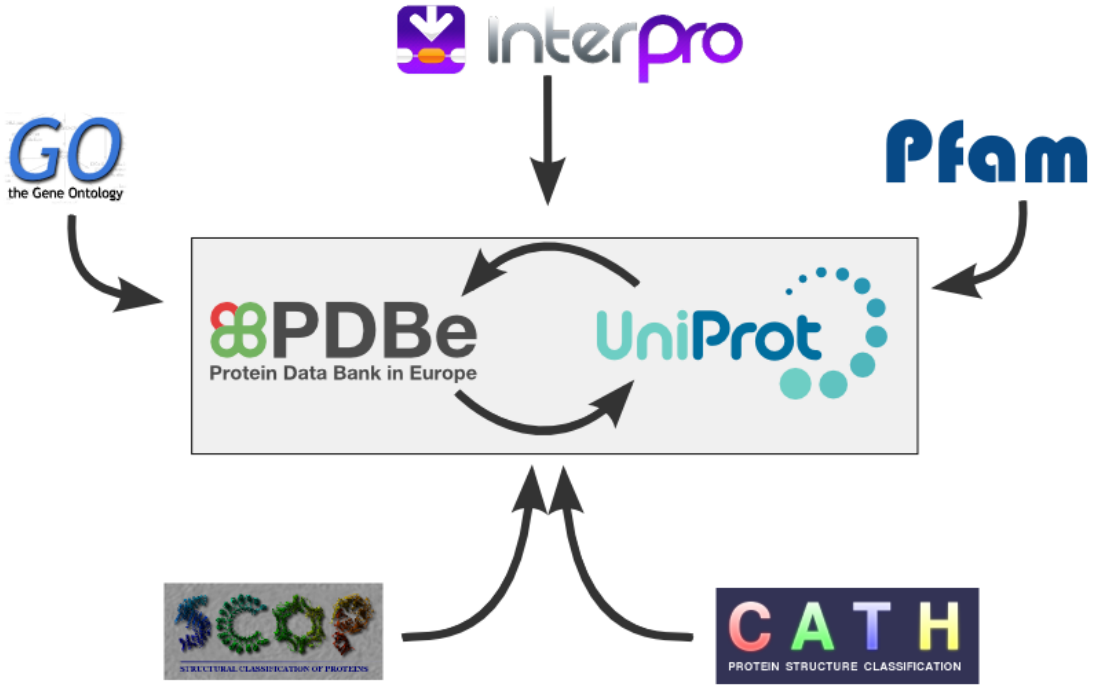
At the heart of EMBL-EBI resources





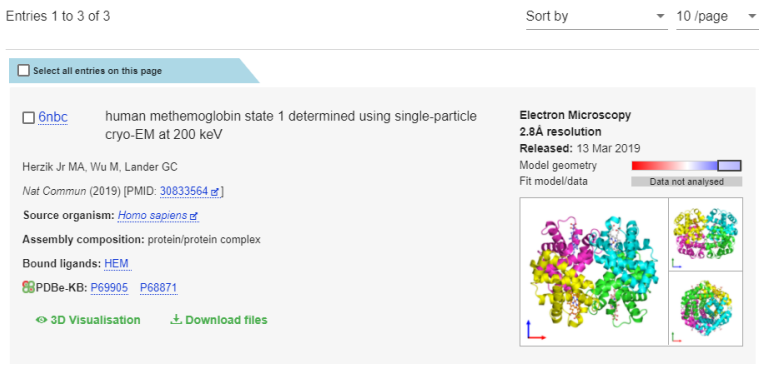
PDBe – add additional data

- At PDBe we integrate data from other data resources, including from within EBI
- An example
  - SIFTS - Structure Integration with Function, Taxonomy and Sequence

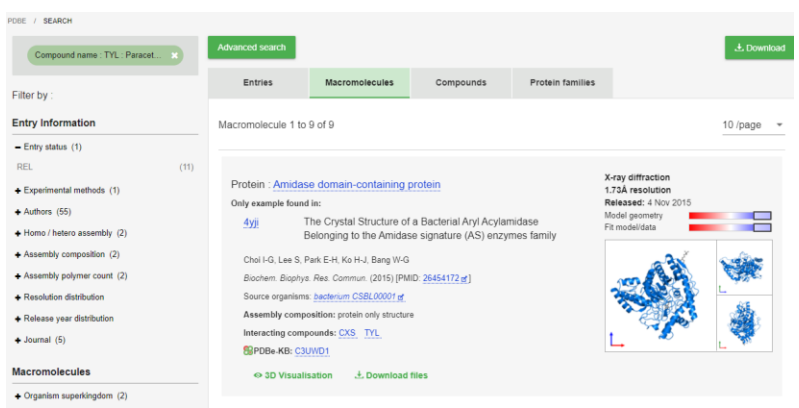


Why search the PDB?

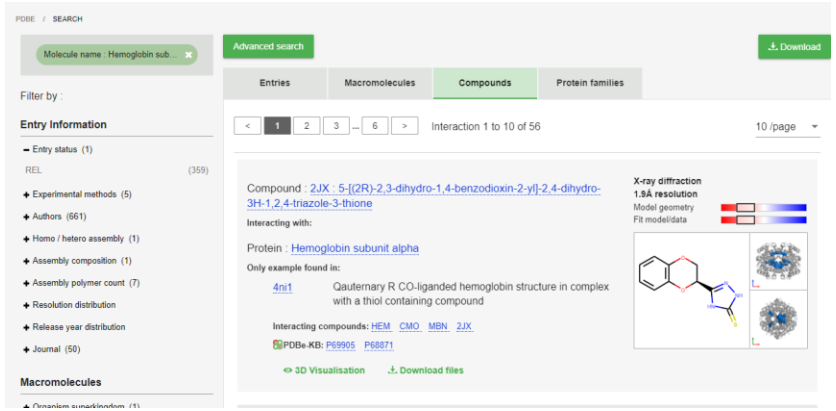
- Provides a convenient way of finding entries in the PDB



Hemoglobin solved by Electron Microscopy



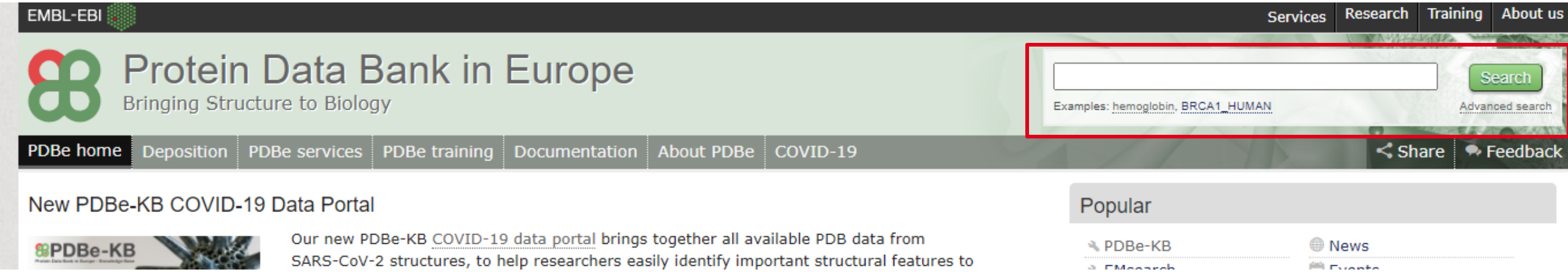
Which proteins bind to paracetamol



Which ligands bind to hemoglobin

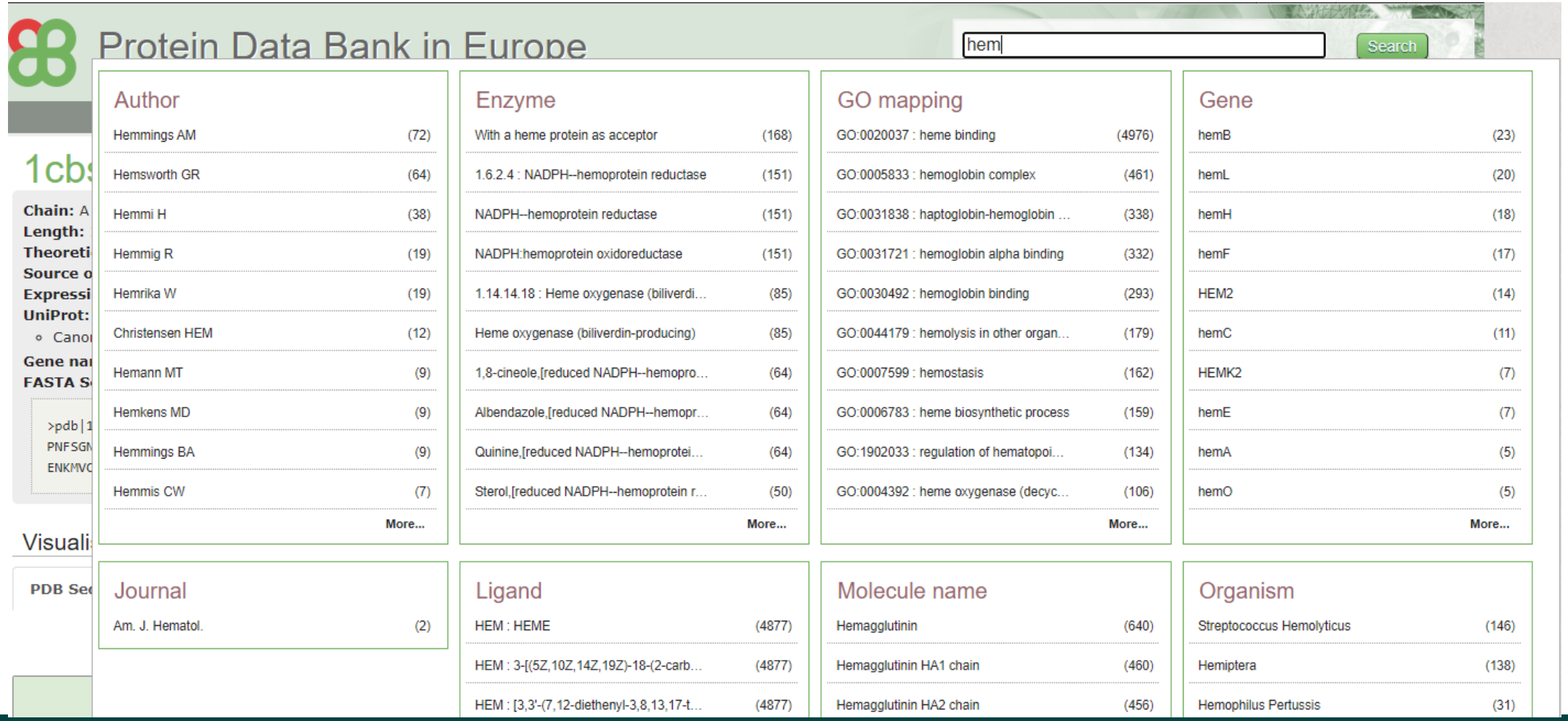
PDBe search - introduction

- Search is available directly from PDBe's home page – pdbe.org



PDBe's search - autocomplete

- PDBe search features an auto complete which provides results in categories

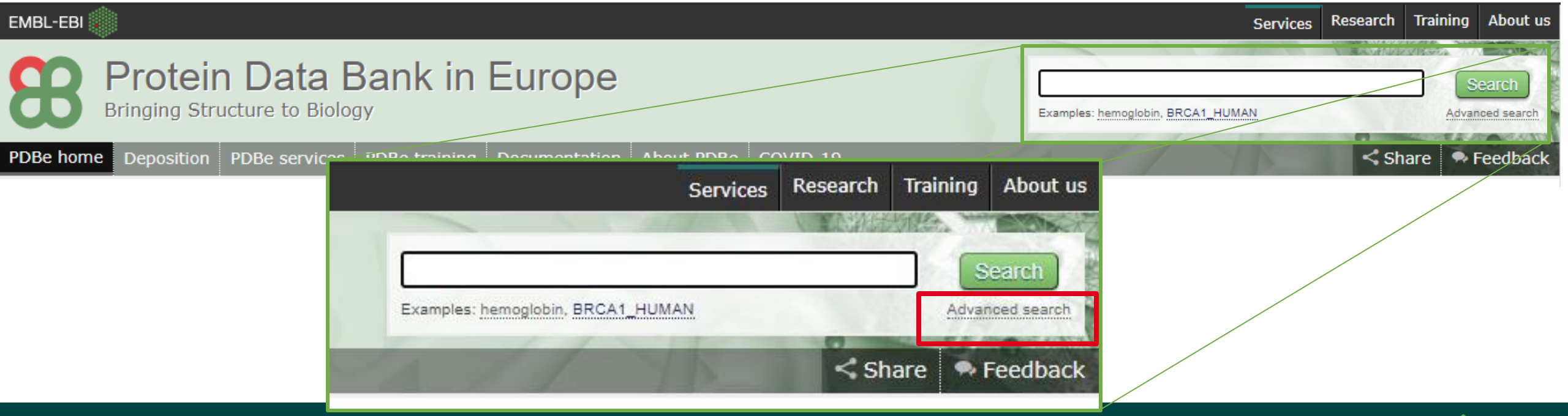


Summary of the webinar

- PDBe in the wwPDB
- PDBe search
- Advanced search
- Introduction to documentation page
- Usage examples

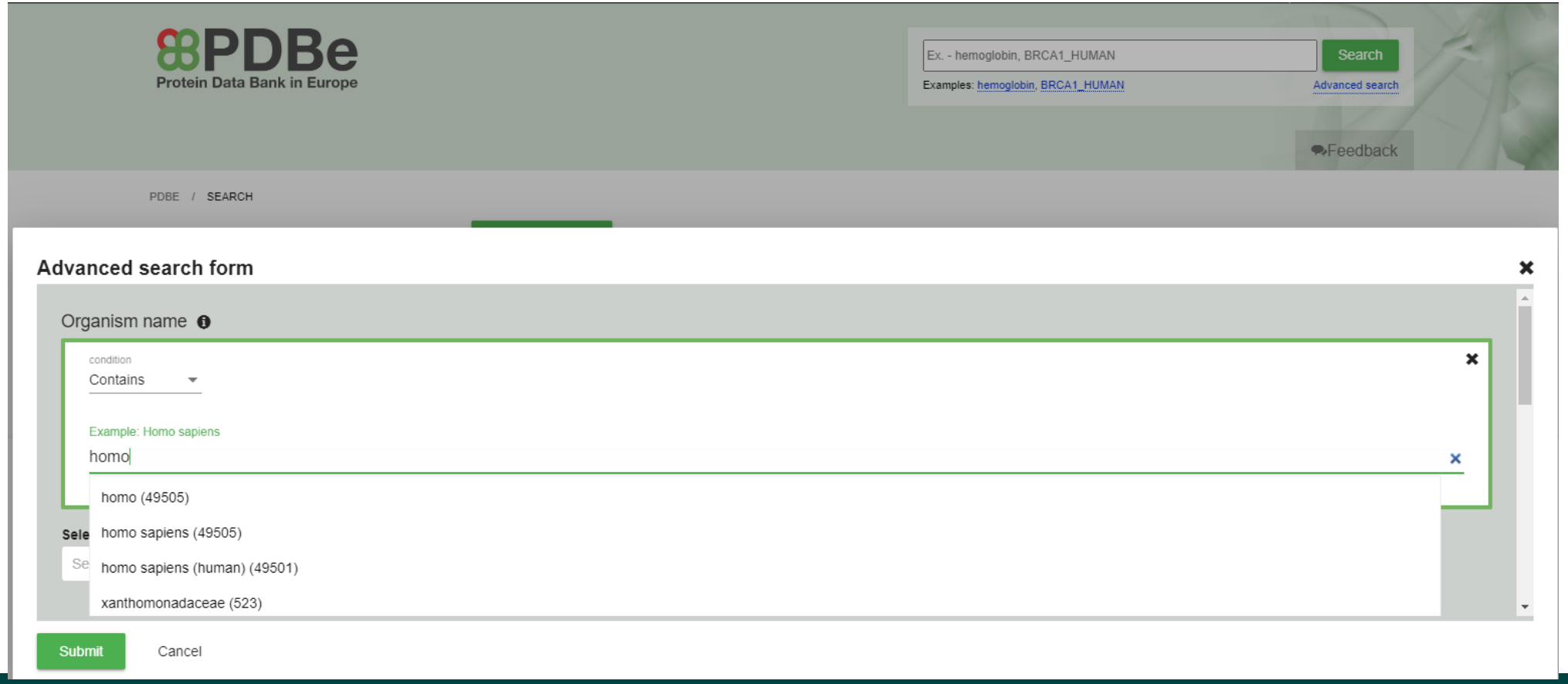
PDBe's search – advanced search

- PDBe's search provides an advanced search feature to build complex queries



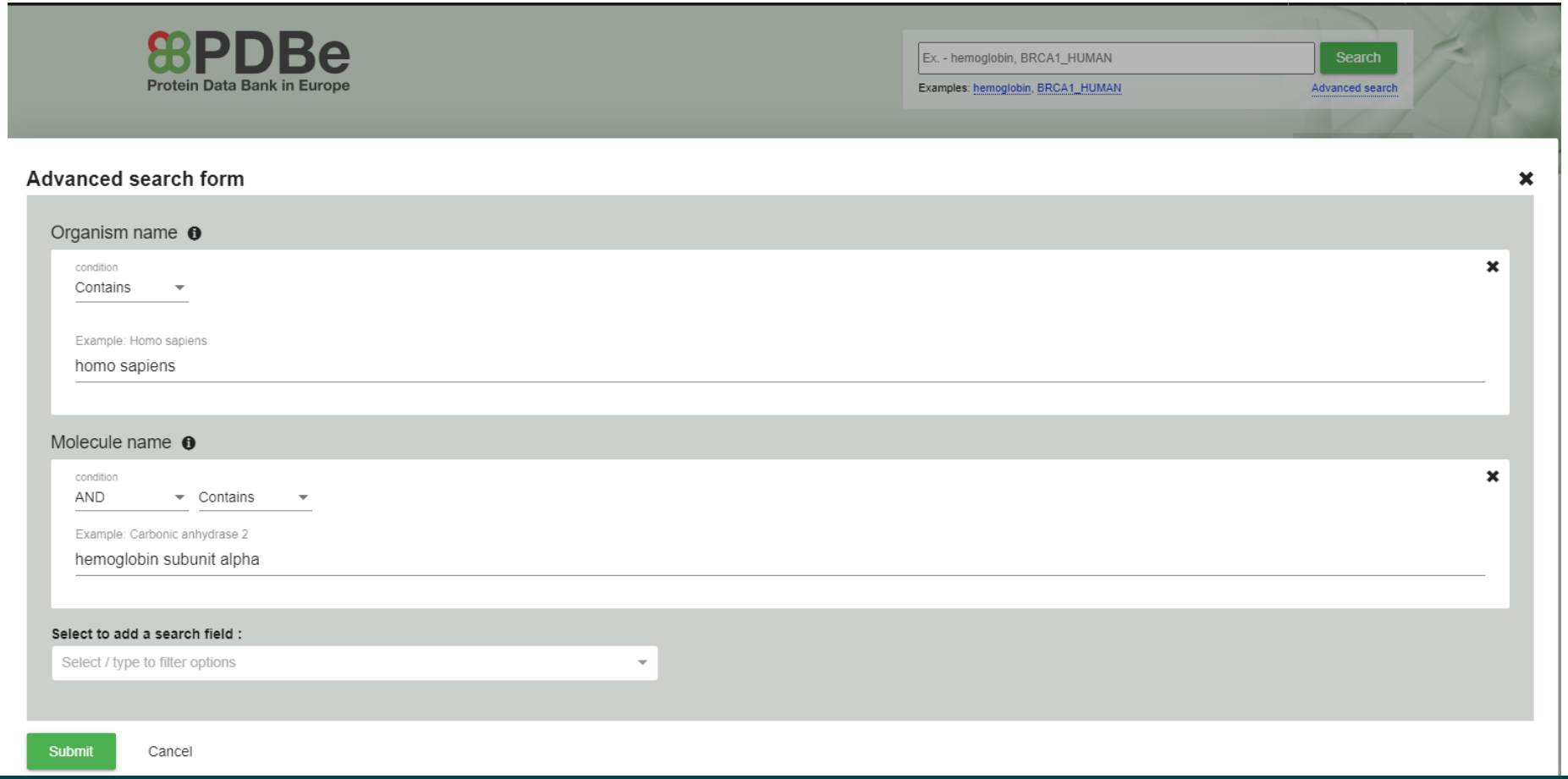
PDBe search – advanced search

- Advanced search contains the same auto complete functionality



PDBe search – advanced search

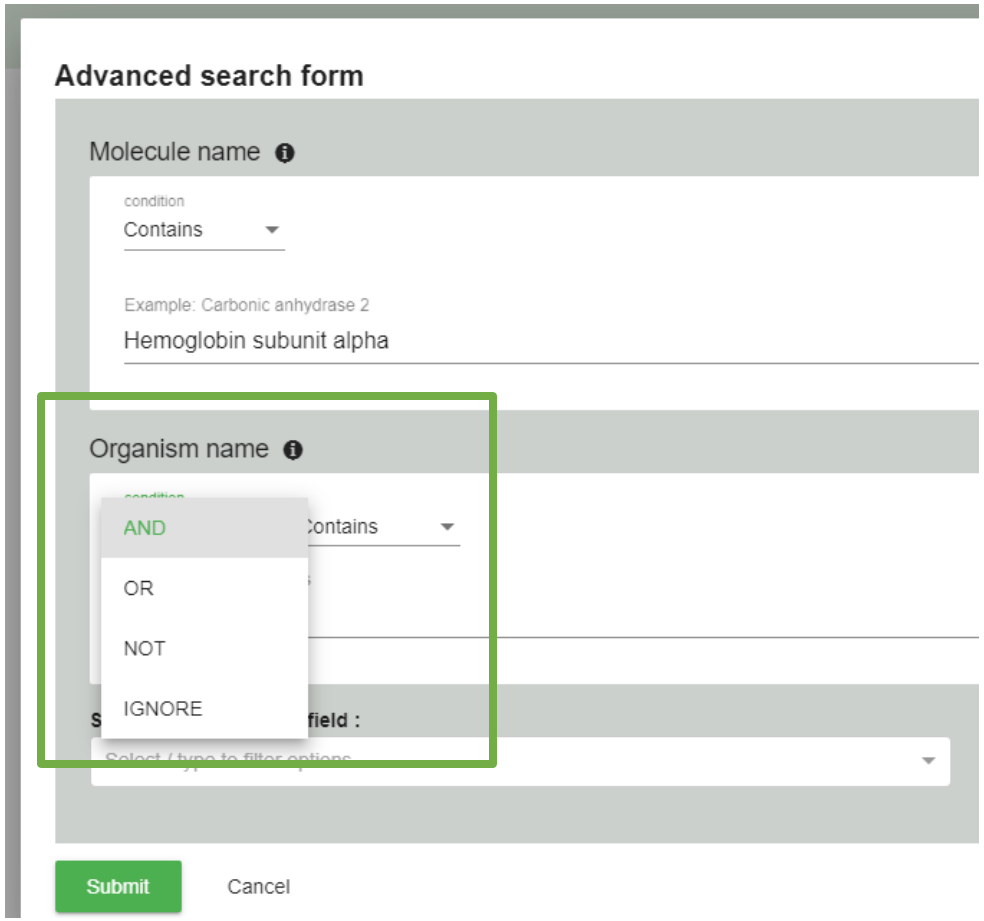
- Allowing complex queries to be made easily

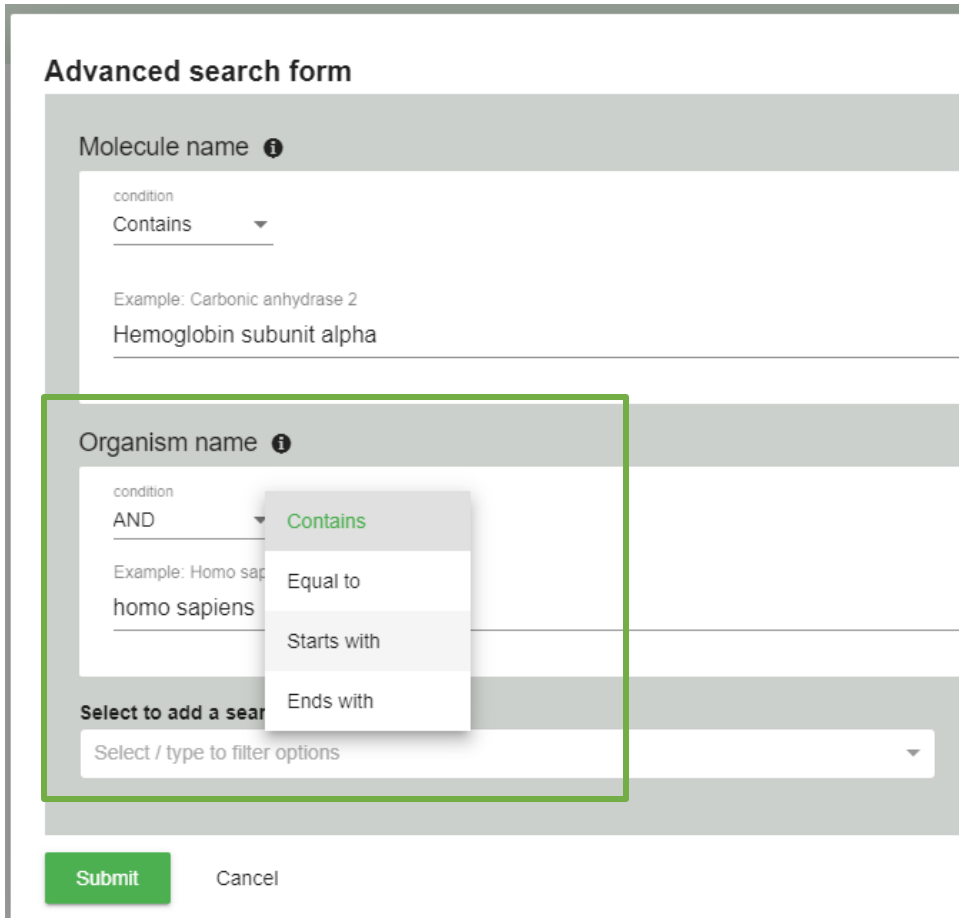


EMBL-EBI

PDBe search – advanced search

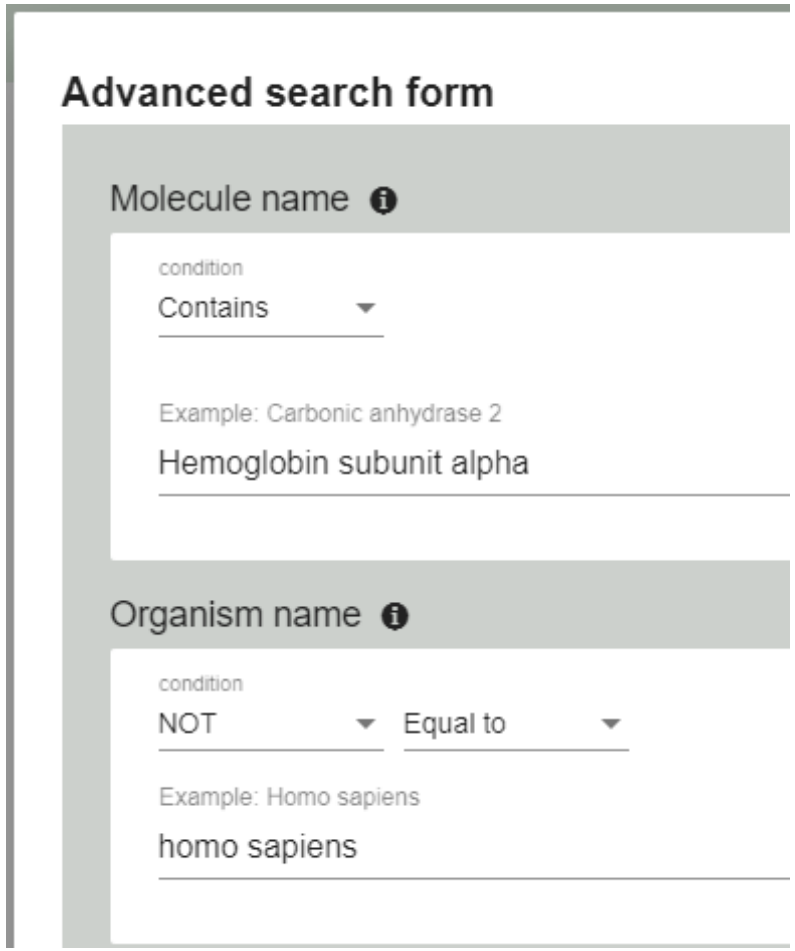
- Operator can be changed from AND and the regular expression can be



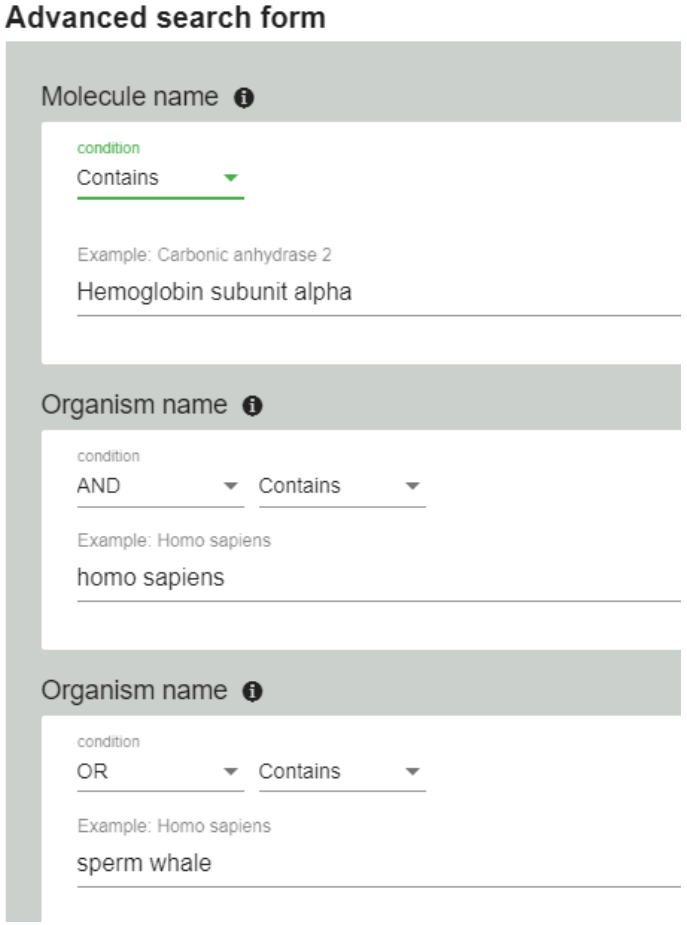


PDBe search – advanced search

Non human Hemoglobin subunit alpha



Hemoglobin subunit alpha from either human or sperm whale



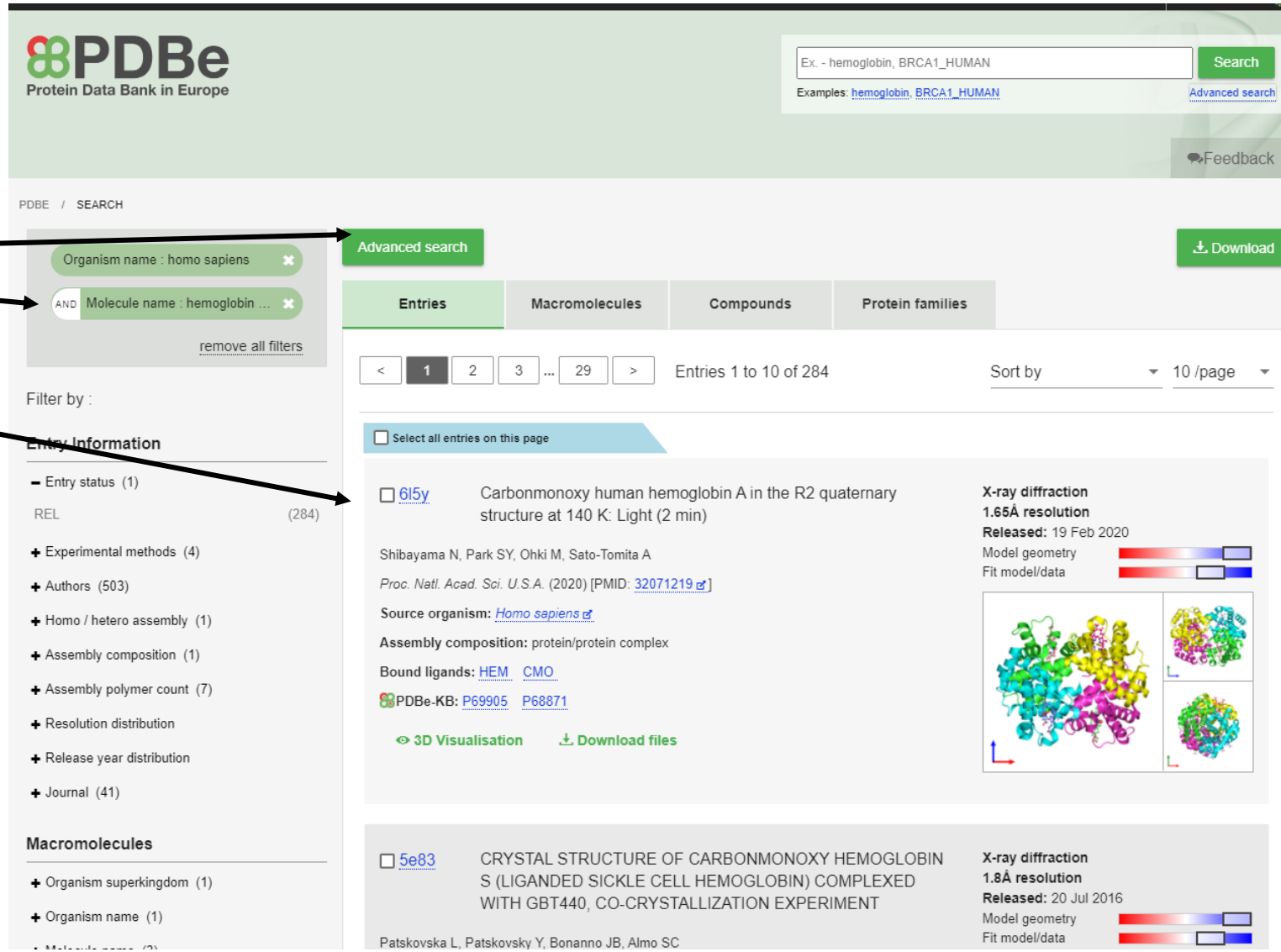
PDBe's search – wild card search

- Wild card searching works in the advanced search

- Single character (matches a single character) - ?
  - The search string te?t would match both test and text.
- Multiple characters (matches zero or more sequential characters) - *
  - The wildcard search:
    - tes* - would match test, testing, and tester.
    - You can also use wildcard characters in the middle of a term. For example: te*t - would match test and text.
    - *est - would match pest and test.

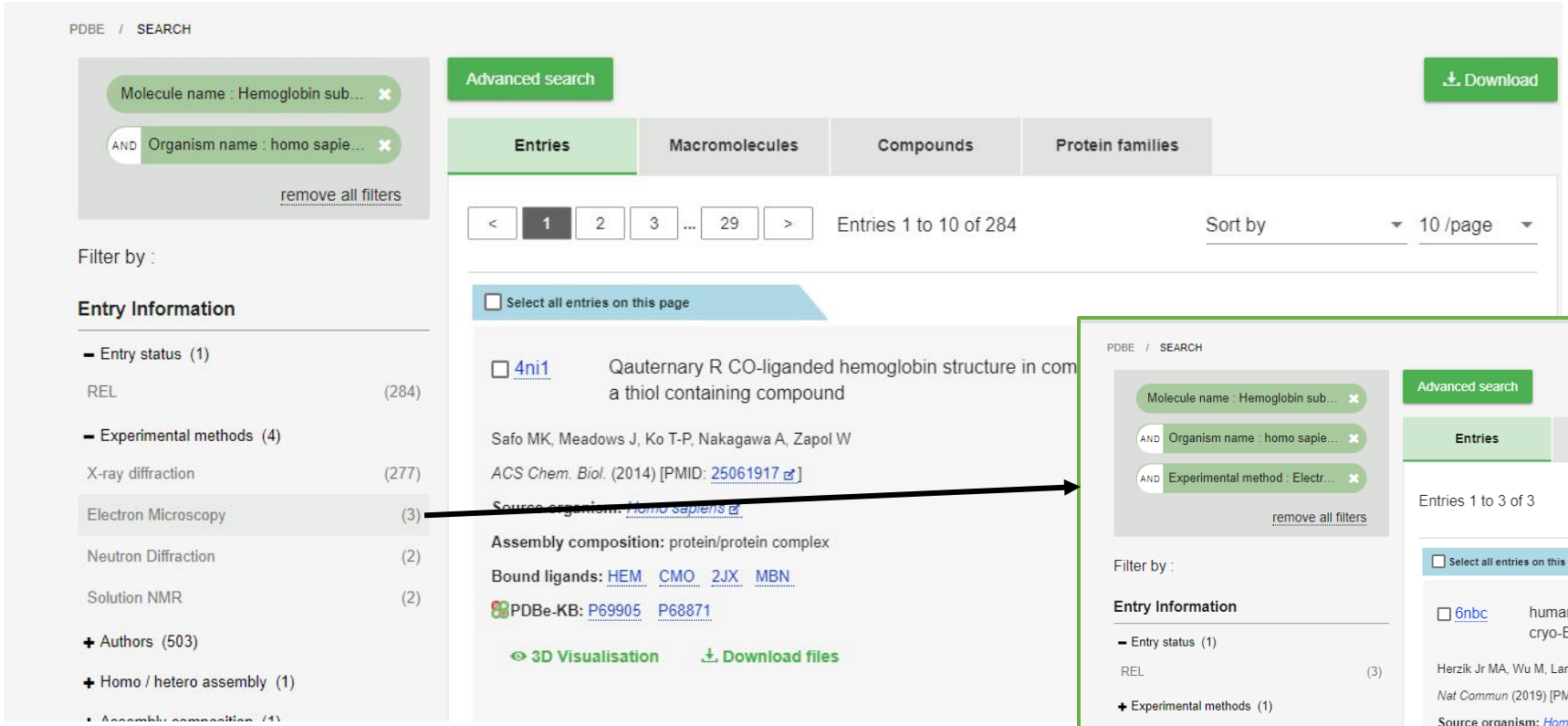
PDBe search - results

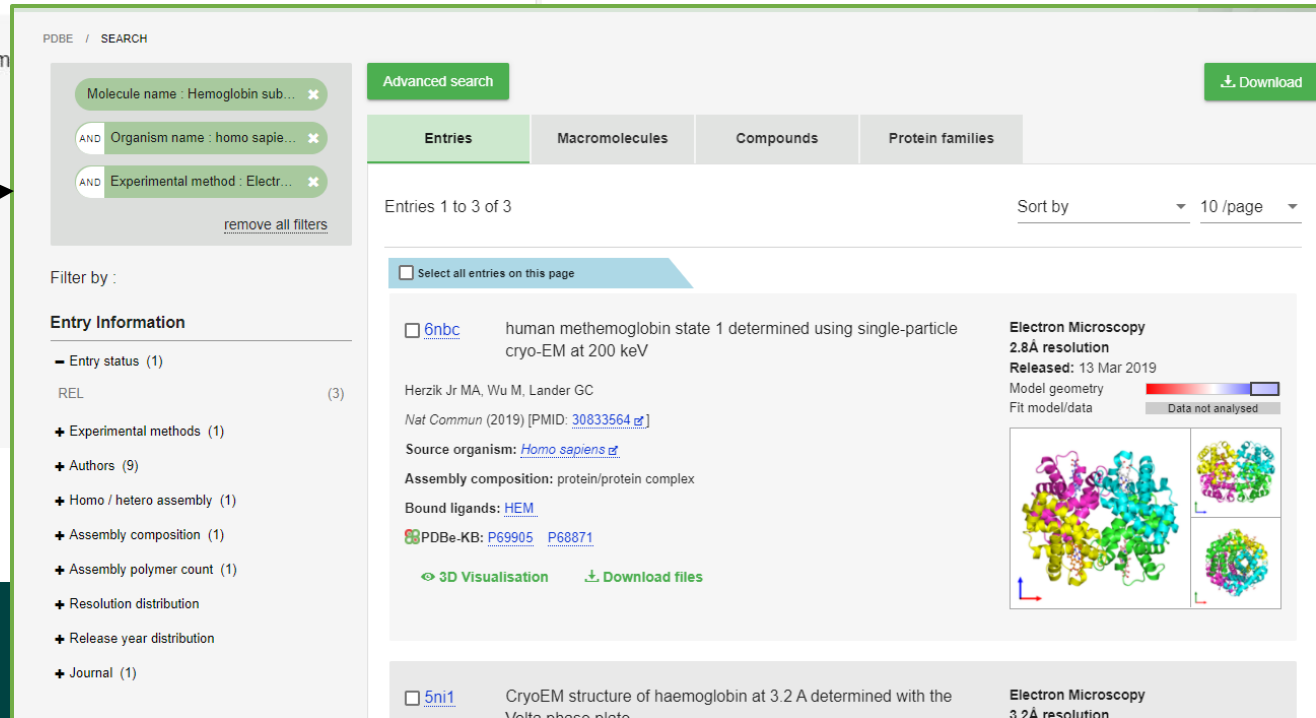
- The results page
  - Allows the search to be changed
  - Presents PDB search results with one panel per PDB entry



PDBe search – results – changing the query

- Additional filters can be easily added using the facets

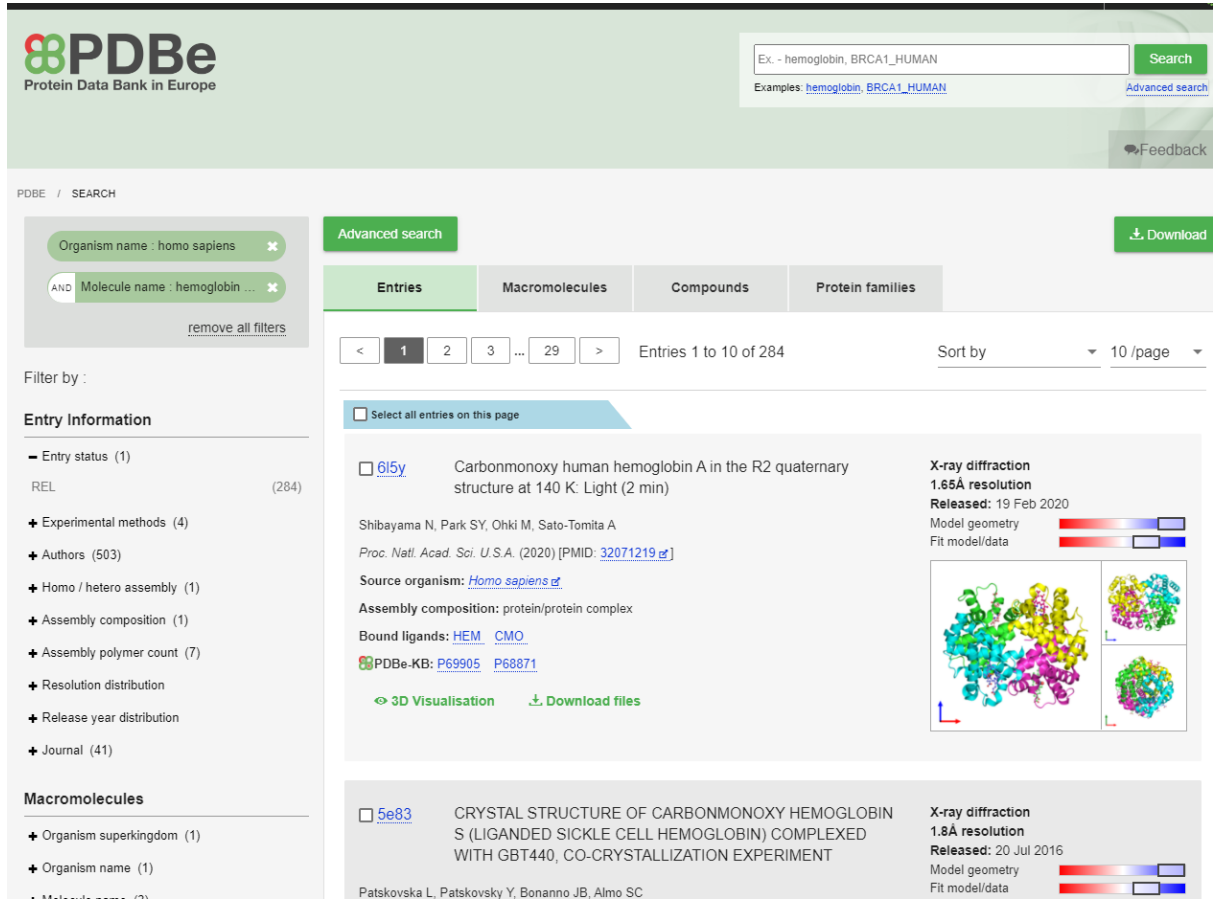




What data is available in the search?

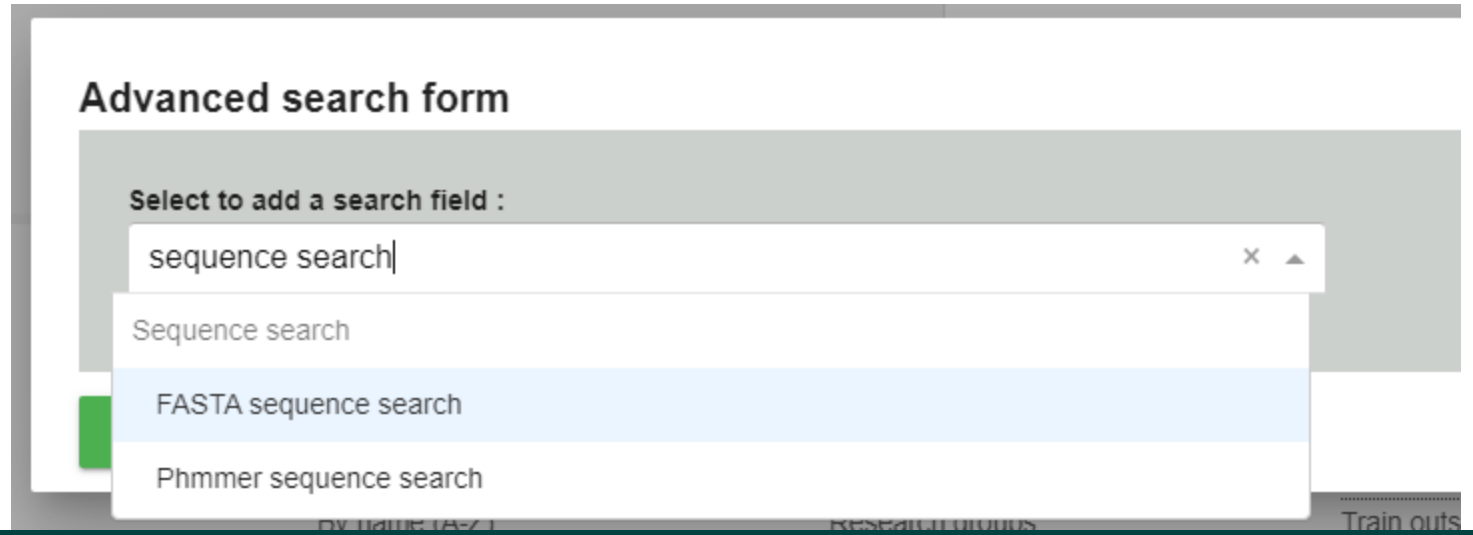
- Information about the entry
  - PDB ID
  - Date of release
  - Experimental method
- Related database identifiers
  - EMDB, BMRB
- Source and taxonomy information
- Experimental details for the entry
- Crystallographic cell parameters
- Simplified quality information
- Primary citation
- Likely biological assembly
- Chemical components details
- Cross references
  - UniProt, Pfam, SCOP, CATH, InterPro, GO, Rfam

- Any information you see on the search results page



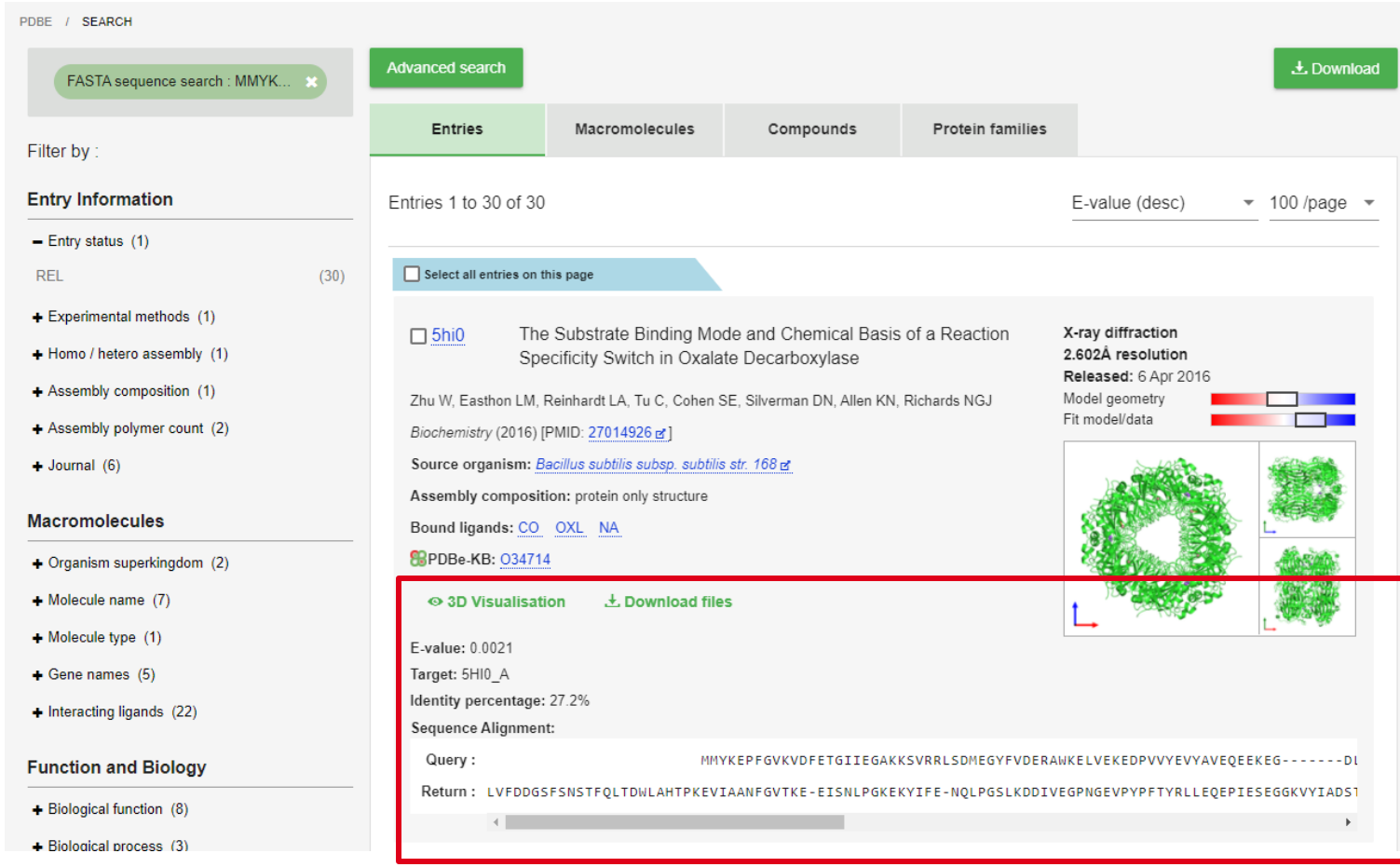
PDBe search – sequence search

- PDBe's search offers two protein sequence searches to search against the PDB in the advanced search
- Both use EBI's sequence search programs
  - FASTA – https://www.ebi.ac.uk/Tools/sss/fasta/ - Heuristic approach
  - Phmmer - https://www.ebi.ac.uk/Tools/hmmer/search/phmmer - hidden Markov model approach



PDBe search – sequence search

- Search results from sequence search provide the
  - Alignment
  - E-value
  - Percentage identity
  - Target chain ID in the entry



Summary of the webinar

- PDBe in the wwPDB
- PDBe search
- Advanced search
- Introduction to documentation page
- Usage examples

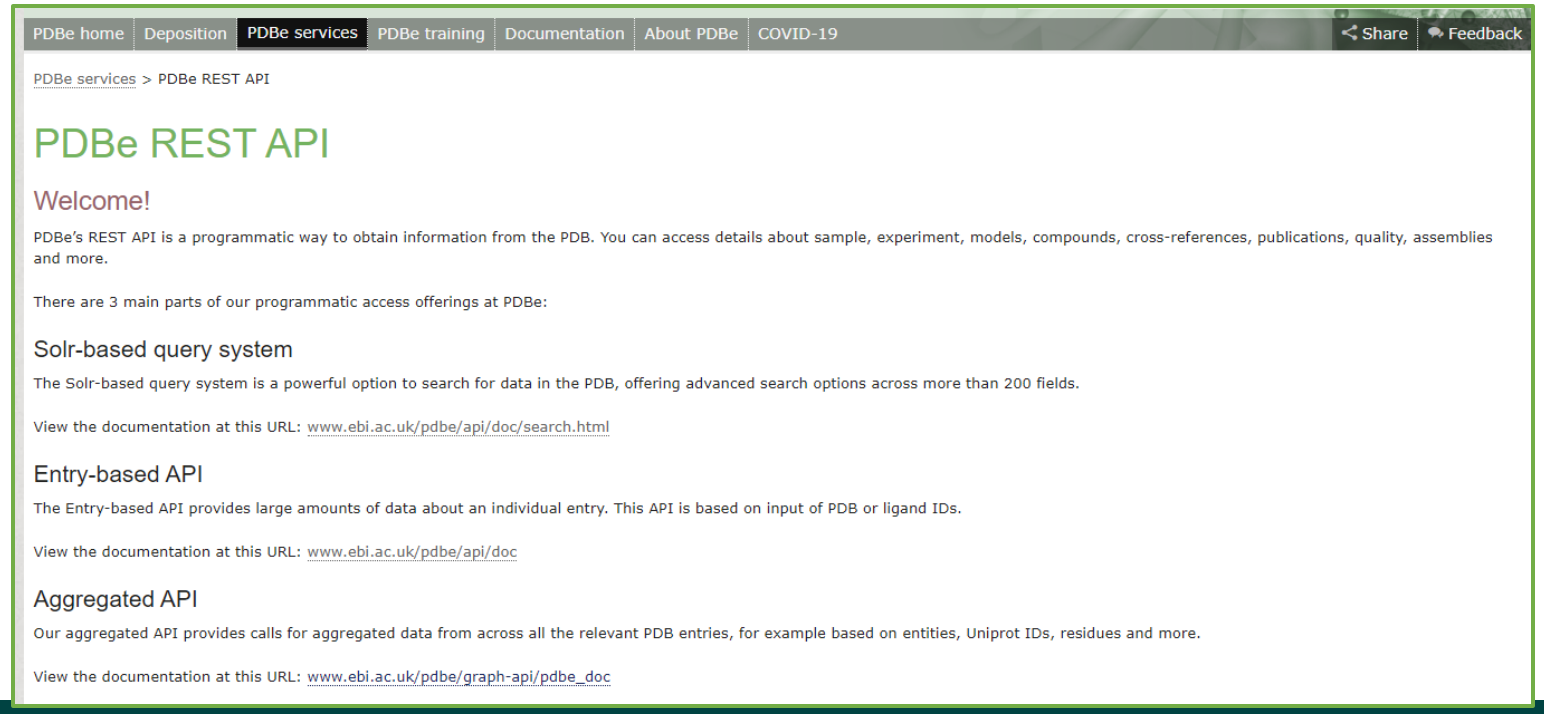
PDBe search – Solr powered

- PDBe's search is powered by Apache Solr
- 214 data items indexed
- 126 are available through advanced search
- 76 are available through the auto complete functionality
- 37 are available through facets in search results

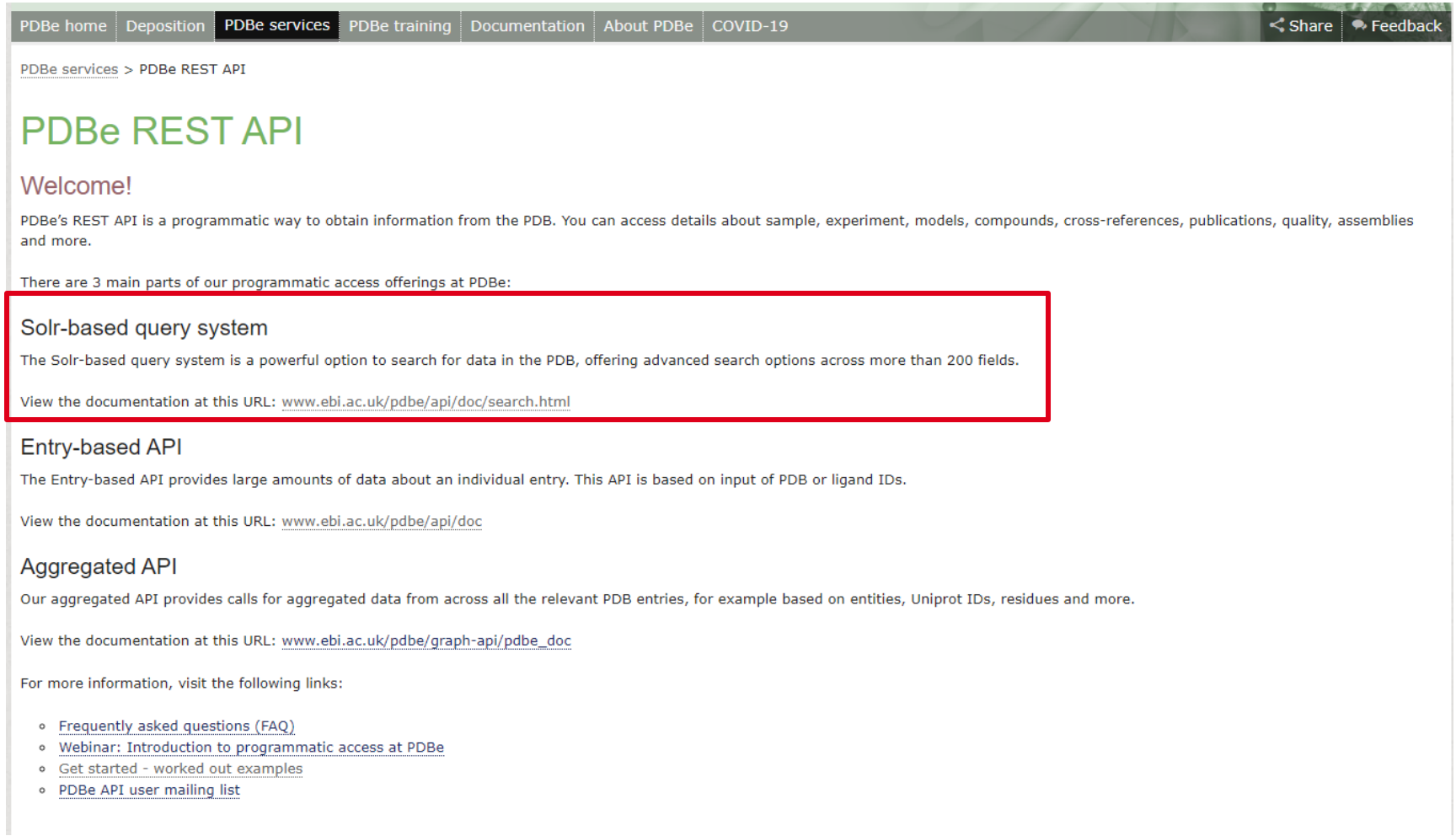
PDBe's APIs

- All data for PDBe's webpages are served through APIs
- Search is no exception to this

- PDBe's API are available at https://pdbe.org/api



PDBe's search - API



PDBe home | Deposition | PDBe services | PDBe training | Documentation | About PDBe | COVID-19 | Share | Feedback

PDBe services > PDBe REST API

PDBe REST API

Welcome!

PDBe's REST API is a programmatic way to obtain information from the PDB. You can access details about sample, experiment, models, compounds, cross-references, publications, quality, assemblies and more.

There are 3 main parts of our programmatic access offerings at PDBe:

Solr-based query system

The Solr-based query system is a powerful option to search for data in the PDB, offering advanced search options across more than 200 fields.

View the documentation at this URL: www.ebi.ac.uk/pdbe/api/doc/search.html

Entry-based API

The Entry-based API provides large amounts of data about an individual entry. This API is based on input of PDB or ligand IDs.

View the documentation at this URL: www.ebi.ac.uk/pdbe/api/doc

Aggregated API

Our aggregated API provides calls for aggregated data from across all the relevant PDB entries, for example based on entities, Uniprot IDs, residues and more.

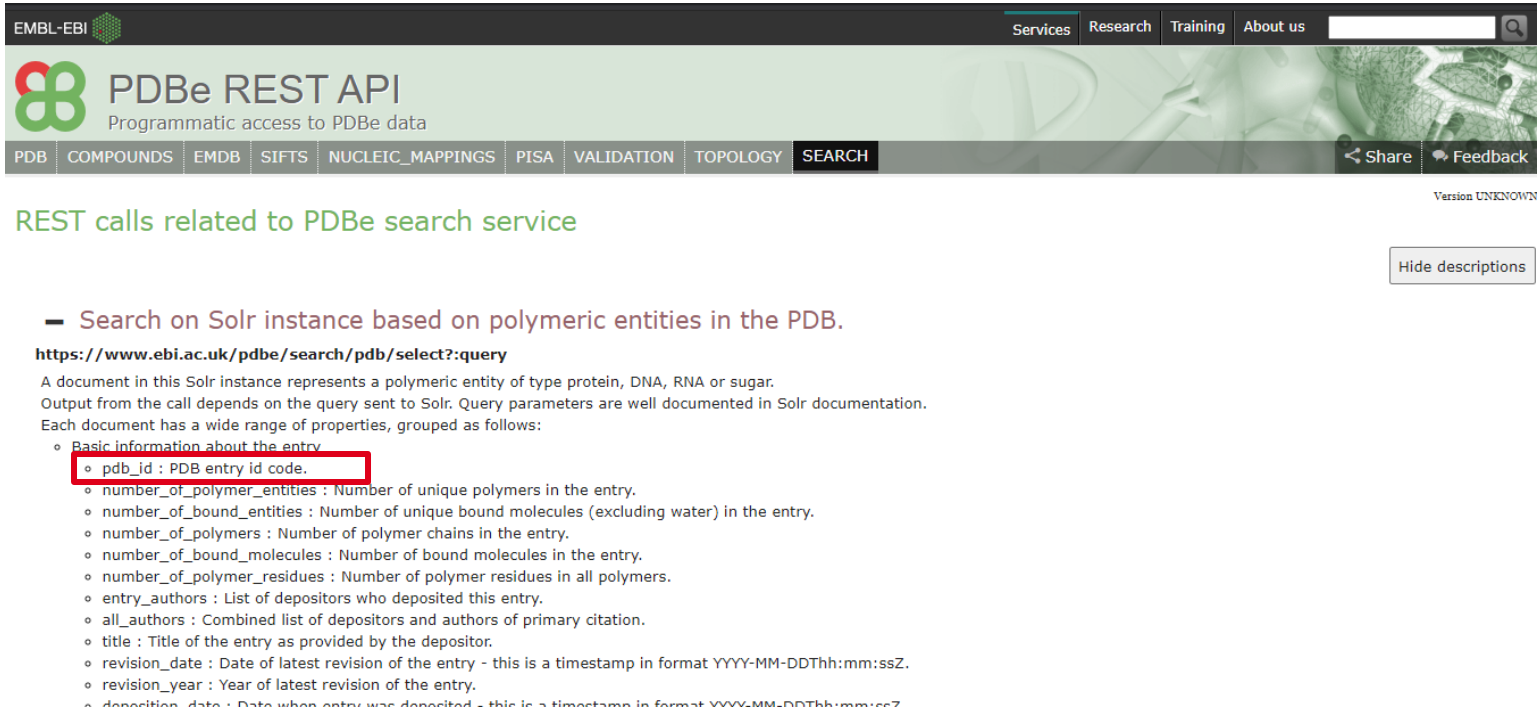
View the documentation at this URL: www.ebi.ac.uk/pdbe/graph-api/pdbe_doc

For more information, visit the following links:

- Frequently asked questions (FAQ)
- Webinar: Introduction to programmatic access at PDBe
- Get started - worked out examples
- PDBe API user mailing list

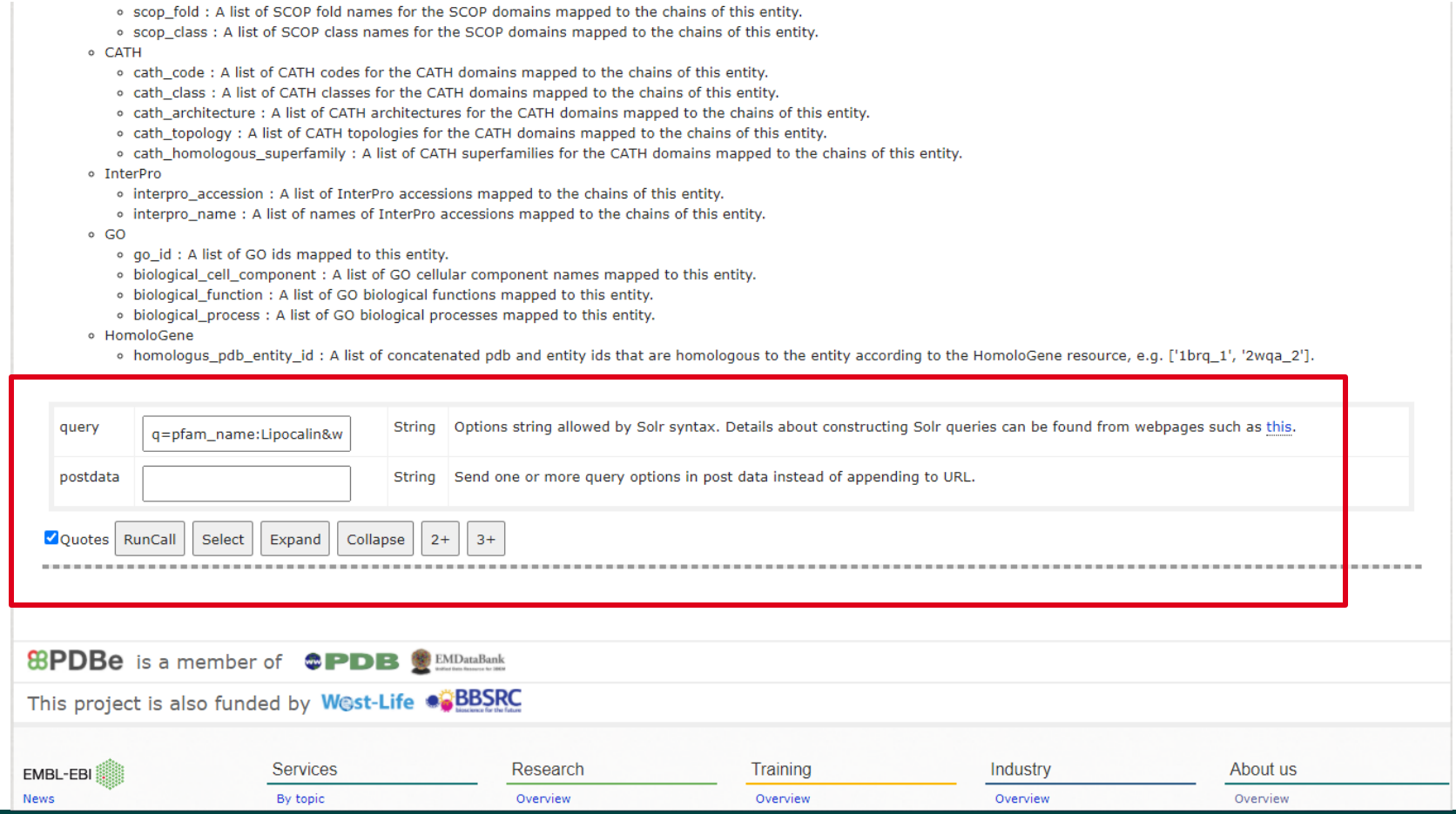
PDBe's search – API documentation

- The documentation provides a list of fields which are available for querying
- For example
  - pdb_id : PDB entry id code
- Field name is before the colon
- A description on what data is in the field is provided after the colon



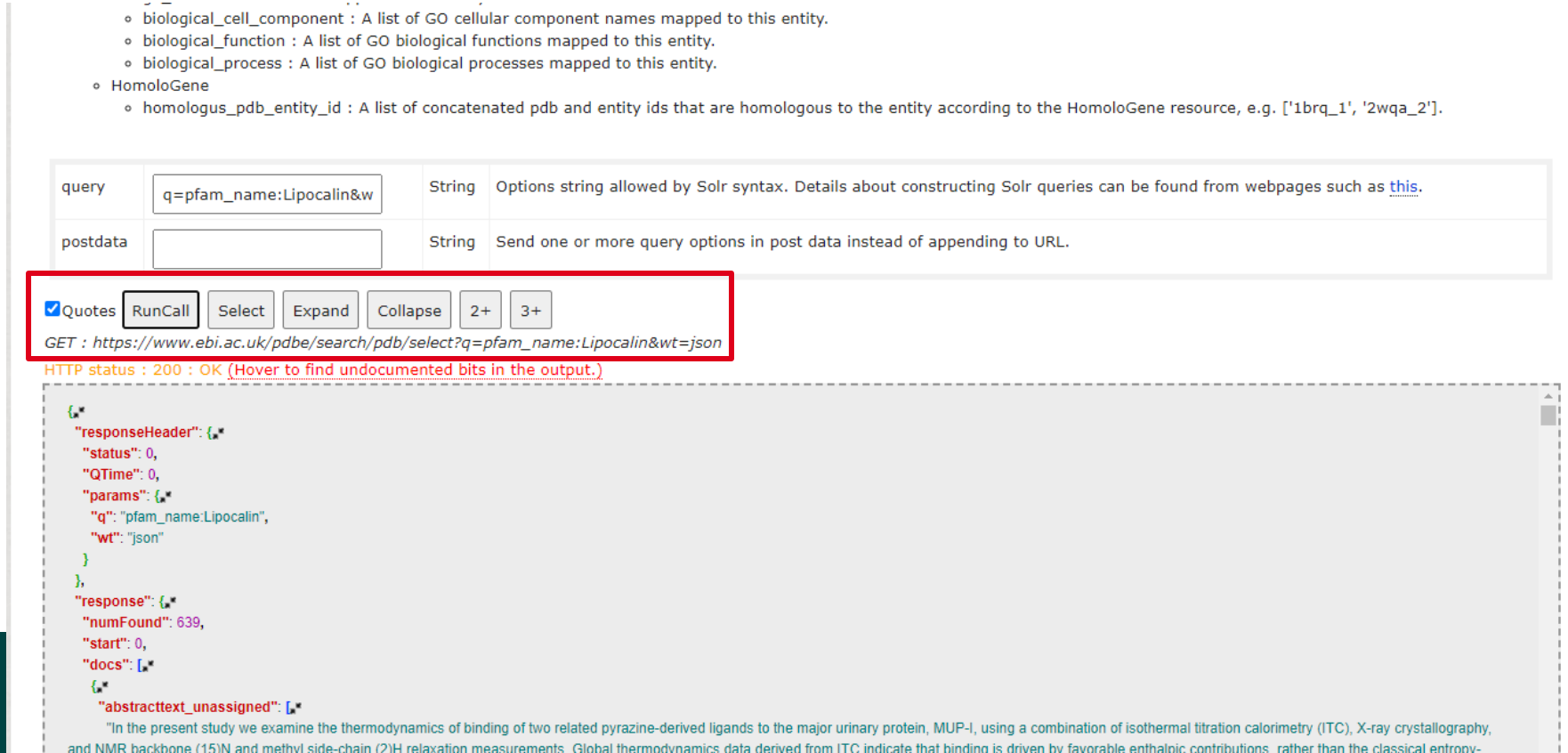
PDBe's search – API documentation

- And a input box for an interactive query tool



PDBe's search – API documentation

- Running the call gives
  - the URL which was used
  - The results in the same page



PDBe's search – API documentation

- The search can be changed directly in the query box
- Any field in red in the "docs" section can be used in the search – these fields are the same as those in the documentation section above
- Here the search has been changed to "all_assembly_type:tetramer"

query	q=all_assembly_type:tetra	String	Options string allowed by Solr syntax. Details about constructing Solr queries can be found from webpages such as this.
postdata		String	Send one or more query options in post data instead of appending to URL.

Quotes | RunCall | Select | Expand | Collapse | 2+ | 3+

GET : https://www.ebi.ac.uk/pdbe/search/pdb/select?q=all_assembly_type:tetramer&wt=json

HTTP status : 200 : OK (Hover to find undocumented bits in the output.)


```
{
 "responseHeader": {
  "status": 0,
  "QTime": 0,
  "params": {
   "q": "all_assembly_type:tetramer",
   "wt": "json"
  }
 },
 "response": {
  "numFound": 30550,
  "start": 0,
```

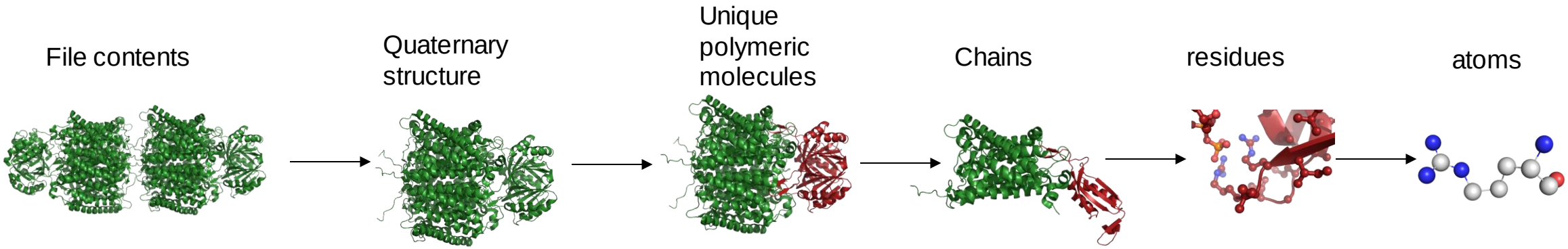
PDBe search – Solr powered

- A “document” is the unit of search and index
- Within a document there are fields – properties of the document

- Comparing Solr to a database
  - Document = A table row
  - Fields = columns

- So for PDB entries what is the optimal “document”?

Hierarchy of a PDB entry

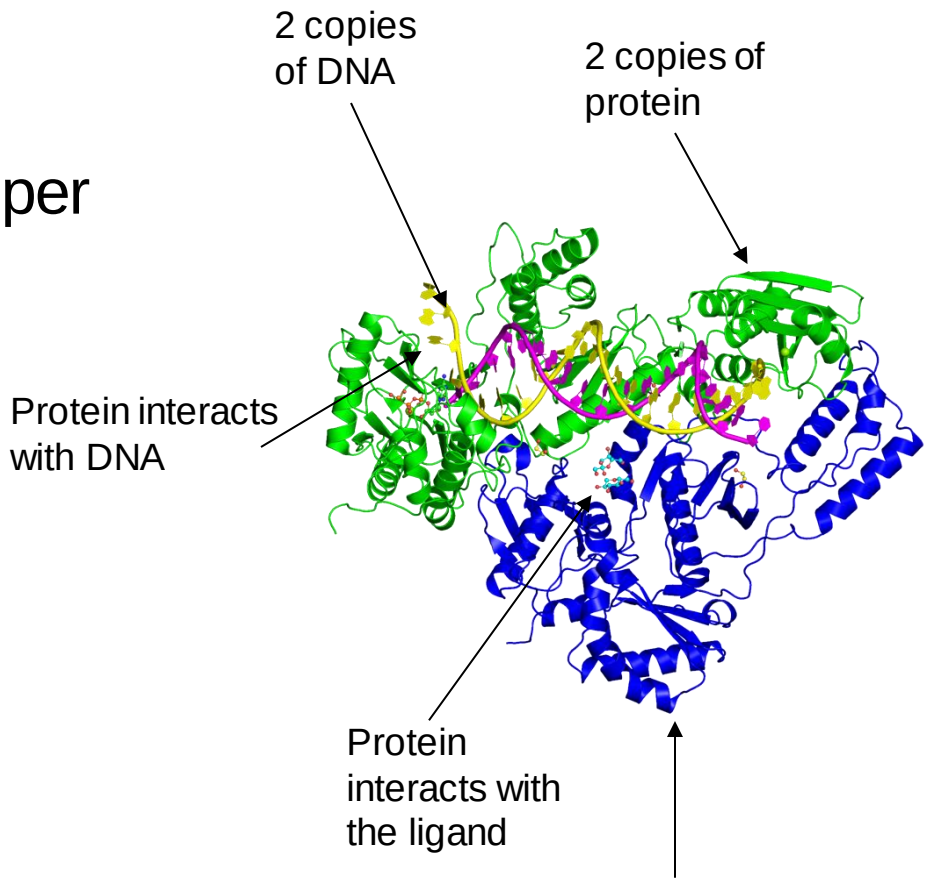


- Document must allow users to
  - Find all molecules in an entry
  - Access properties of a molecule – i.e. sequence
  - Access properties of an entry – i.e. quaternary structure
  - Not create too large an index so that the search is slow and returns unnecessary documents
- A PDBe we use “Unique polymeric molecules”

PDBe's search – API structure

- Advantages of using unique polymeric molecule per document
  - Type of molecule
  - The sequence
  - Number of copies in the entry and their chain IDs
  - If the sequence has modified residues

  - And:
    - Which other polymers interact with this molecule
    - Which ligands interact with this molecule




```
>pdb|5txl|A C
MVPISPIETVPVKLKPGMDGPKVKQWPLTEEKIKALVEICTEMEKEGKISKIGPENPYNTPVFAIKKKDSTKWRKLVDFRELNKRT
QDFWEVQLGIPHPAGLKKKKSVTVLDVGDAYFSVPLDEDFRKYTAFTIPSINNETPGIRYQYNVLPQGWKGSPAIFQSSMTKILEP
FKKQNPDIVIYQYMDDLYVGSDLEIGQHRTKIEELRQHLLRWGLTTPDKKHQKEPPFLWMGYELHPDKWTVQPIVLPEKDSWTVND
```

PDBe's search – API structure

- So if we change the search to pdb_id:6l5y – haemoglobin with 2 unique polymer molecules
- Then we get 2 results

query	q=pdb_id:6l5y&wt=json	String	Options string allowed by Solr syntax. Details about constructing Solr queries can be found from webpages such as this.
postdata		String	Send one or more query options in post data instead of appending to URL.

Quotes | RunCall | Select | Expand | Collapse | 2+ | 3+

GET : https://www.ebi.ac.uk/pdbe/search/pdb/select?q=pdb_id:6l5y&wt=json

HTTP status : 200 : OK (Hover to find undocumented bits in the output.)


```
{
  "responseHeader": {
    "status": 0,
    "QTime": 0,
    "params": {
      "q": "pdb_id:6l5y",
      "wt": "json"
    }
  },
  "response": {
    "numFound": 2,
    "start": 0,
    "docs": [
```

Summary of the webinar

- PDBe in the wwPDB
- PDBe search
- Advanced search
- Introduction to documentation page
- Usage examples

PDBe's search – URL construction

- PDBe's search is a Solr search

- https://www.ebi.ac.uk/pdbe/search/pdb/select?q=all_assembly_type:tetramer&wt=json
- Parts of this URL:
  - Search URL: https://www.ebi.ac.uk/pdbe/search/pdb/select?
  - Query: q=all_assembly_type:tetramer
  - Return type: wt=json – options are json, xml – default is json
- By default this returns 100 documents
  - Rows: rows=100

- More details about Solr syntax can be found in the Solr help pages
  - https://lucene.apache.org/solr/guide/common-query-parameters

PDBe's search – accessing data programmatically

- GET and POST calls supported
  - GET - https://www.ebi.ac.uk/pdbe/search/pdb/select?q=all_assembly_type:tetramer&fl=pdb_id,experimental_method&rows=100&wt=json
    - With "&" separating parameters
  - POST
    - URL = https://www.ebi.ac.uk/pdbe/search/pdb/select?
    - data is posted as a JSON with parameters a key value pairs

```
{'fl': 'pdb_id,experimental_method',
 'q': 'all_assembly_type:tetramer',
 'rows': 100,
 'wt': 'json'}
```

PDBe's search – accessing data programmatically

- By default each result / "document" contains every field
- This is useful for developing queries but usually you only need a few fields
- The Solr parameter Field List - "**fl**" allows only selected fields to be returned – speeding up return of results
- E.g.
  - q=all_assembly_type:tetramer
  - fl=pdb_id,experimental_method

https://www.ebi.ac.uk/pdbe/search/pdb/select?

EMBL-EBI

PDBe's search – exclusive searches - prosite patterns

- More advanced queries are possible in the API than the advanced search

- It's possible to look for prosite (https://prosite.expasy.org/) patterns in sequences
- For example
  - [ST]-x(2)-[DE] à Casein kinase II phosphorylation site
    - q=molecule_sequence:/.*?[ST]..[DE].*/

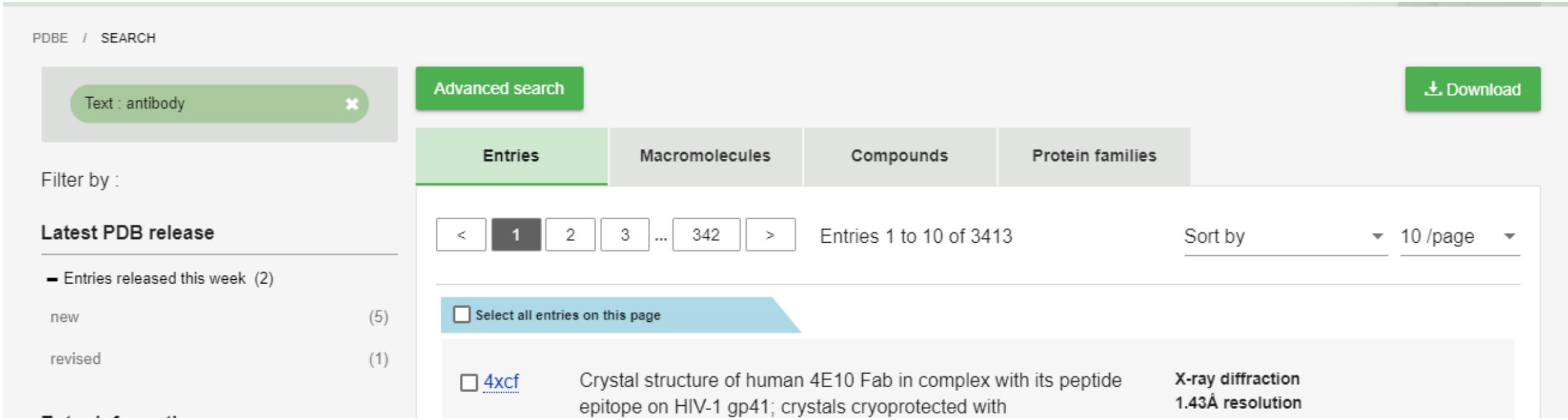
  - [RK]-x(2,3)-[DE]-x(2,3)-Y à Tyrosine kinase phosphorylation site
    - q=molecule_sequence:/.*?[RK].{2,3}[DE].{2,3}Y.*/

  - G-{EDRKHPFYW}-x(2)-[STAGCN]-{P} à N-myristoylation site
    - q=molecule_sequence:/.*?G[^EDRKHPFYW].{2}[STAGCN][^P].*/

- Prosite rules changed into regular expression rules
  - Prosite x(2,3) = any 2 or 3 residues - which becomes .{2,3}
  - Prosite {} = not any of these residues - which becomes [^]
  - Prosite [] = any one of the residues - which remains []

PDBe's search – exclusive searches - antibodies

- Antibody data is currently only available in the search API
- Text search for "antibody" gives 3414 PDB entries

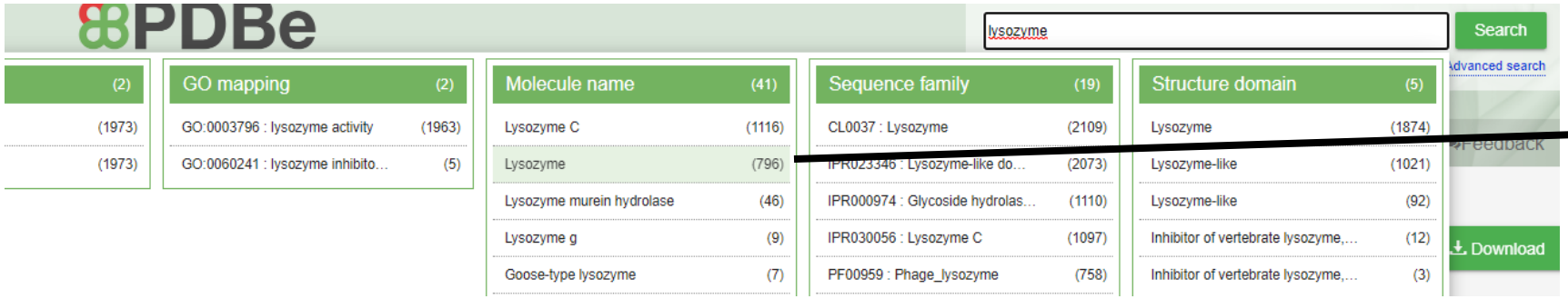


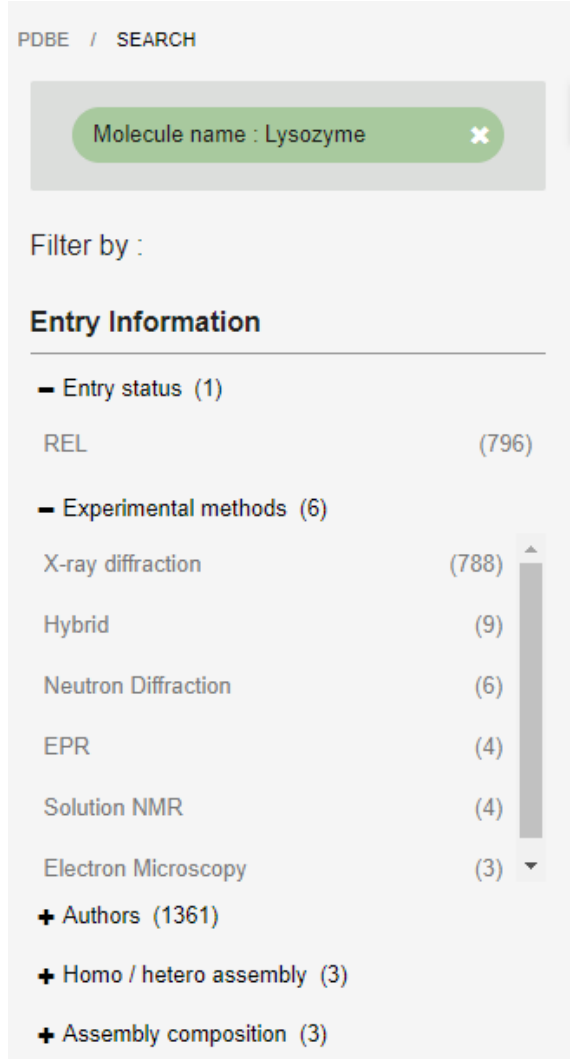
PDBe's search – worked example

- There is a flag for antibodies in the search API
  - "antibody_flag" – if this is set to Y then an entity is an antibody
  - So this can be searched for using
    - q=antibody_flag:Y
  - However, each PDB entry can contain multibody antibodies so it is necessary to group the results by "pdb_id" to get the number of PDB entries
  - This can be done using the following
    - q=antibody_flag:Y
    - group=true
    - group.field=pdb_id
    - group.mincount=1
    - group.ngroups=true
    - fl=pdb_id
  - Then "ngroups" is the number of PDB IDs
  - This 4931 is significantly more than the 3414 from a free text search

PDBe's search – worked example of faceting

- At PDBe we use faceting to display how many entries fit into each class
- For example
  - After searching for "Lysozyme" we can find the number of entries solved by each experimental method

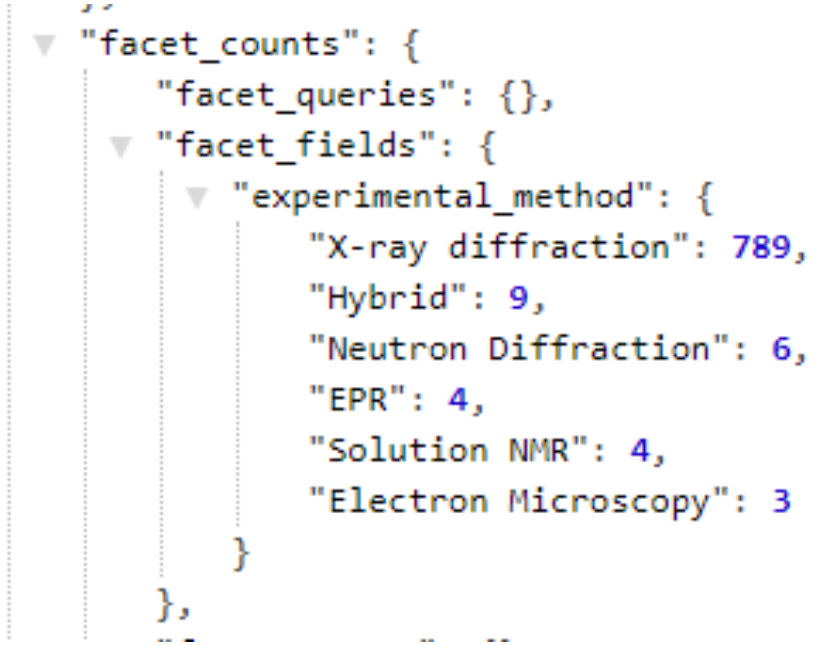




PDBe's search – worked example of faceting

- We can repeat the same query using the search API

  - q=all_molecule_names:Lysozyme
  - group=true
  - group.field=pdb_id
  - group.ngroups=true
  - json.nl=map
  - facet=true
  - facet.mincount=1
  - facet.field=experimental_method
  - group.facets=true



PDBe's search – worked example of pivoting

- We can pivot the results so we get
  - We use the query q=all_molecule_names:Lysozyme
  - group=true
  - group.field=pdb_id
  - group.ngroups=true
  - json.nl=map
  - facet=true
  - facet.mincount=1
  - group.facets=true
  - facet.pivot=experimental_method

https://www.ebi.ac.uk/pdbe/search/pdb/select?

PDBe's search – worked example of faceting

- If we want more complex examples we can find the number of components in the assembly per experimental method in entries solved between 2014 to 2020
  - q=release_year:[2014 TO 2019]
  - group=true
  - group.field=pdb_id
  - group.ngroups=true
  - json.nl=map
  - facet=true
  - facet.mincount=1
  - group.facets=true
  - facet.pivot=release_year,experimental_method,assembly_num_component
  - facet.sort=release_year asc
- Pivots are given as a comma separated list and in the results you get a hierarchy of pivots

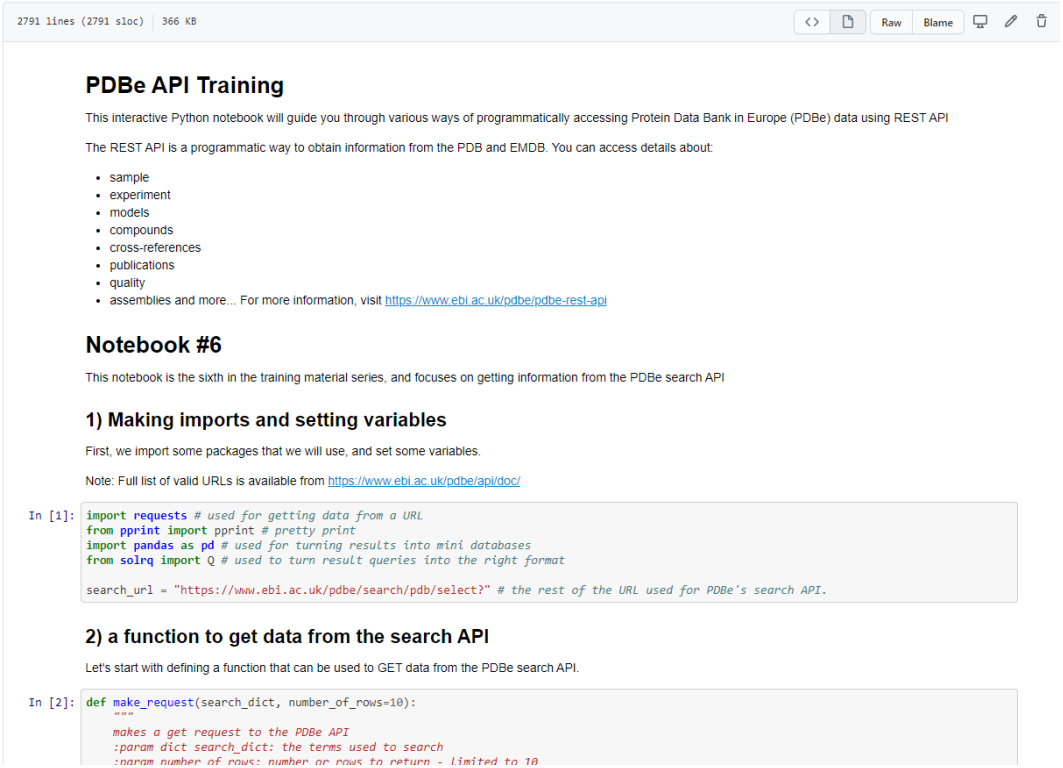

```
"grouped": {
  "pdb_id": {
    "matches": 147867,
    "ngroups": 63413,
    "groups": []
  }
},
"facet_counts": {
  "facet_queries": {},
  "facet_fields": {},
  "facet_ranges": {},
  "facet_intervals": {},
  "facet_heatmaps": {},
  "facet_pivot": {
    "release_year,experimental_method,assembly_num_component": [
      {
        "field": "release_year",
        "value": 2014,
        "count": 27242,
        "pivot": [
          {
            "field": "experimental_method",
            "value": "Electron Microscopy",
            "count": 5185,
            "pivot": [
              {
                "field": "assembly_num_component",
                "value": 1,
                "count": 4
              },
              {
                "field": "assembly_num_component",
                "value": 2,
                "count": 15
```


https://www.ebi.ac.uk/pdbe/search/pdb/select?

PDBe's search – teaching resources

- API tutorials are available in Python on PDBe's GitHub page
- https://github.com/PDBeurope/pdbe-api-training

- A specific search API tutorial is available
- https://github.com/PDBeurope/pdbe-api-training/blob/master/api_tutorials/6_PDB_search.ipynb



2791 lines (2791 sloc) | 366 KB

Raw Blame

PDBe API Training

This interactive Python notebook will guide you through various ways of programmatically accessing Protein Data Bank in Europe (PDBe) data using REST API

The REST API is a programmatic way to obtain information from the PDB and EMDB. You can access details about:

- sample
- experiment
- models
- compounds
- cross-references
- publications
- quality
- assemblies and more... For more information, visit https://www.ebi.ac.uk/pdbe/pdbe-rest-api

Notebook #6

This notebook is the sixth in the training material series, and focuses on getting information from the PDBe search API

1) Making imports and setting variables

First, we import some packages that we will use, and set some variables.

Note: Full list of valid URLs is available from https://www.ebi.ac.uk/pdbe/api/doc/

In [1]:
```
import requests # used for getting data from a URL
from pprint import pprint # pretty print
import pandas as pd # used for turning results into mini databases
from solrq import Q # used to turn result queries into the right format

search_url = "https://www.ebi.ac.uk/pdbe/search/pdb/select?" # the rest of the URL used for PDBe's search API.
```

2) a function to get data from the search API

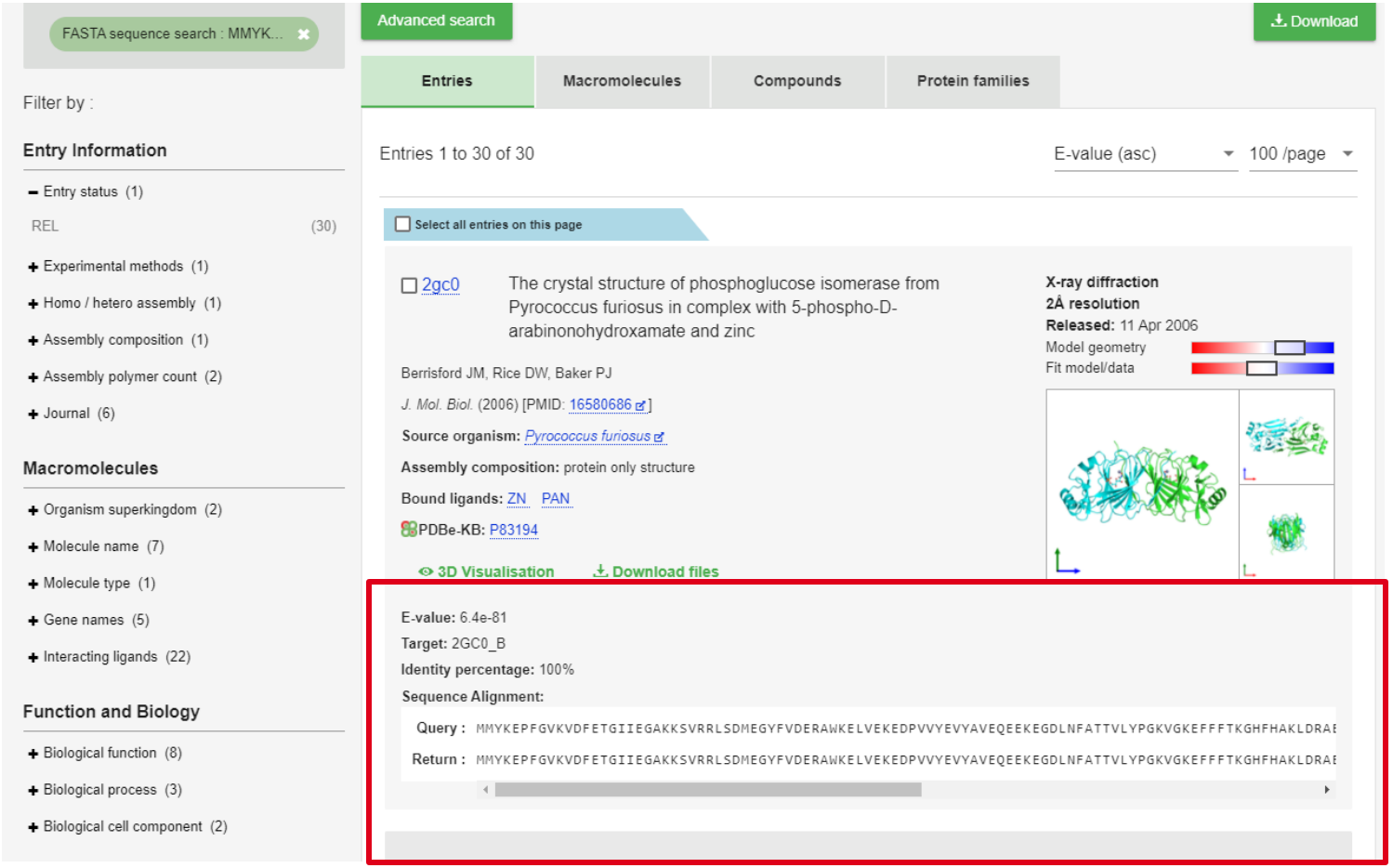
Let's start with defining a function that can be used to GET data from the PDBe search API.

In [2]:
```
def make_request(search_dict, number_of_rows=10):
    """
    makes a get request to the PDBe API
    :param dict search_dict: the terms used to search
    :param number_of_rows: number or rows to return - limited to 10
```

EMBL-EBI

PDBe search – sequence search

- Search results from sequence search is also available through the API



PDBe's search – sequence search

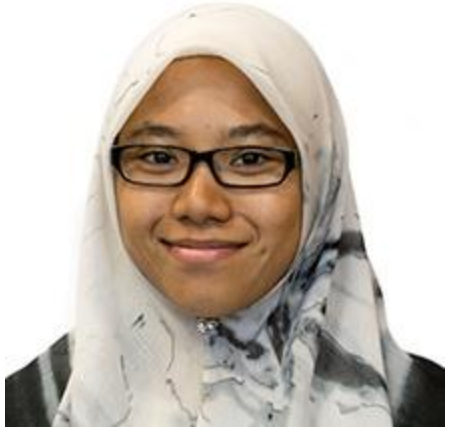
- PDBe's FASTA and pHMMER searches are available through the search API
- The sequence search results are joined to the usual results and a bit of post processing is required
- This is covered in
  - https://github.com/PDBeurope/pdbe-api-training/blob/master/api_tutorials/7_PDB_sequence_search.ipynb

PDBe's search

- PDBe provides an advanced search powered by Solr
- PDBe provides 3 distinct APIs
  - Search
  - Entry based
  - Aggregated API
- PDBe's search system is accessible directly via the search API
- Complex queries are possible, including FASTA and HMMER searches
- Tutorials are available at https://github.com/PDBeurope/pdbe-api-training

PDBe API webinar series

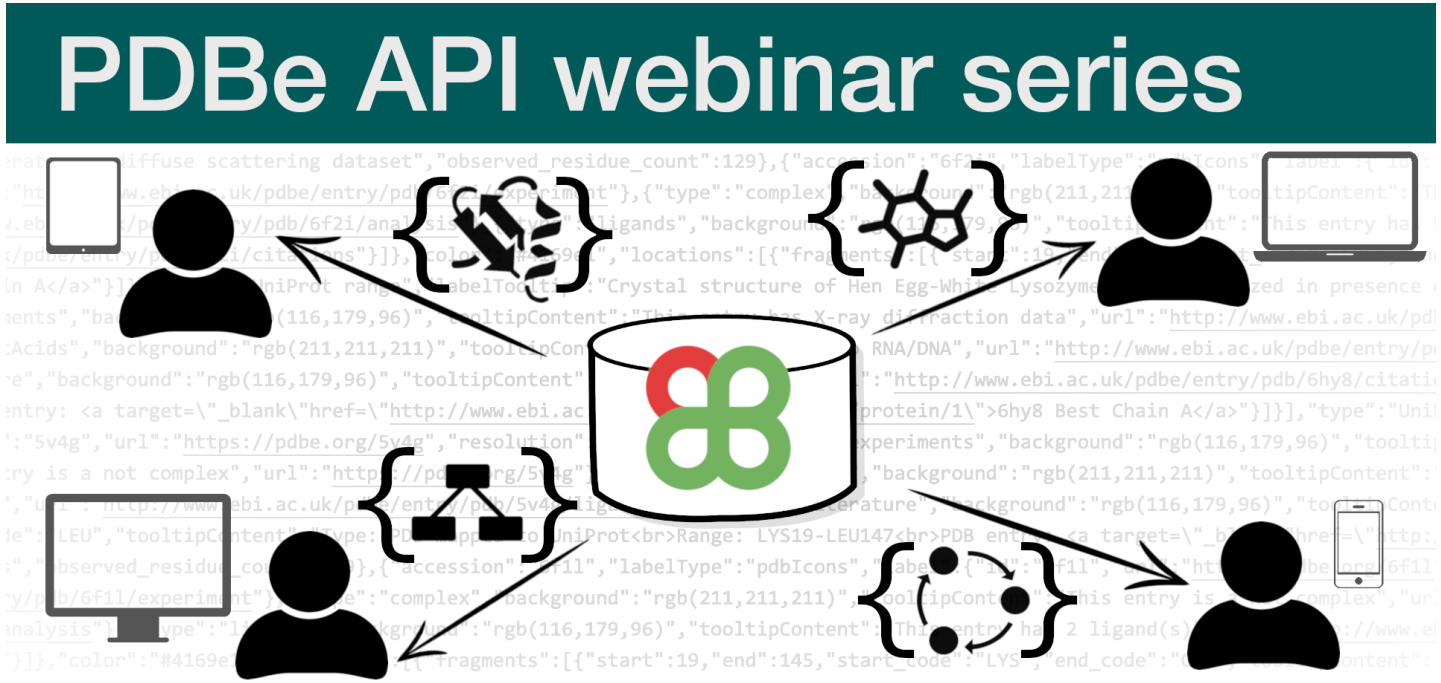
Sep 15th	Introduction to PDBe programmatic access
Sep 22nd	Searching with the PDBe API
Sep 29th	**Creating complex PDBe API queries**
Oct 6th	Using the PDBe graph API
Oct 13th	PDBe tools in github
Oct 20th	Data visualisation at PDBe



Nurul Nadzirin

To register for future webinars in the series

- See the full list of upcoming webinars at bit.ly/PDBeAPIwebinars
- Or visit the PDBe events pages at PDBe.org/events
- Remember to register for each webinar individually!!!



Thank you for your attention!
Any questions?

PDBe.org/API

John Berrisford

pdbhelp@ebi.ac.uk

proteindatabank

@PDBeurope

pdbeurope

pdbart

PDBe
Protein Data Bank in Europe

EMBL-EBI