SEARCHING IN CHEMBL

Michał Nowotka
ChEMBL Group
EMBL-EBI

WHAT'S CHEMBL?

WHAT'S CHEMBL?

"Our team develops and manages ChEMBL, a database of quantitative small molecule bioactivity data focused in the area of drug discovery."

WHAT'S CHEMBL?

"Our team develops and manages ChEMBL, a database of quantitative small molecule bioactivity data focused in the area of drug discovery."

"The majority of the ChEMBL data is derived by manual abstraction and curation from the primary scientific literature."

CHEMBL DATABASE

CHEMBL DATABASE

- Current version: 20 (21 coming #soon)

CHEMBL DATABASE

- Current version: 20 (21 coming #soon)
- 62 Tables

CHEMBL DATABASE

- Current version: 20 (21 coming #soon)
- 62 Tables
- Distinct compounds: 1,463,270

CHEMBL DATABASE

- Current version: 20 (21 coming #soon)
- 62 Tables
- Distinct compounds: 1,463,270
- Targets: 10,774

CHEMBL DATABASE

- Current version: 20 (21 coming #soon)
- 62 Tables
- Distinct compounds: 1,463,270
- Targets: 10,774
- Activities: 13,520,737

CHEMBL DATABASE

- Current version: 20 (21 coming #soon)
- 62 Tables
- Distinct compounds: 1,463,270
- Targets: 10,774
- Activities: 13,520,737
- Publications: 59,610

CHEMBL DJANGO ORM MODEL

CHEMBL DJANGO ORM MODEL

- No more raw SQL in Python code

CHEMBL DJANGO ORM MODEL

- No more raw SQL in Python code
- DB agnostic interface

CHEMBL DJANGO ORM MODEL

- No more raw SQL in Python code
- DB agnostic interface
- Less sensitive to schema changes

CHEMBL DJANGO ORM MODEL

- No more raw SQL in Python code
- DB agnostic interface
- Less sensitive to schema changes
- myChEMBL

CHEMBL WEB SERVICES

CHEMBL WEB SERVICES

- Written in Python

CHEMBL WEB SERVICES

- Written in Python
- RESTful design

CHEMBL WEB SERVICES

- Written in Python
- RESTful design
- Apache 2.0 Licensed

CHEMBL WEB SERVICES

- Written in Python
- RESTful design
- Apache 2.0 Licensed
- Available on **GitHub, PyPI**

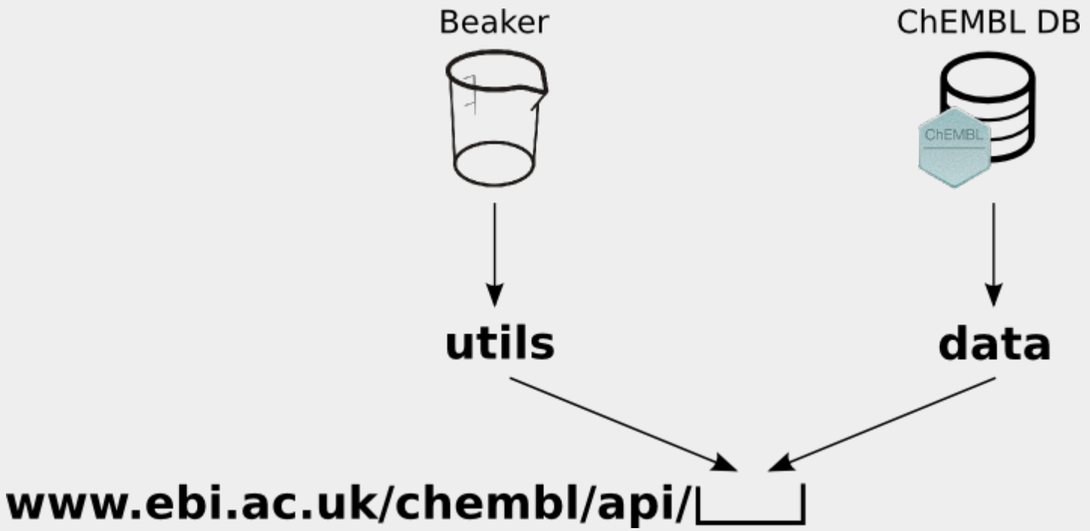
CHEMBL WEB SERVICES

- Written in Python
- RESTful design
- Apache 2.0 Licensed
- Available on **GitHub, PyPI**
- Providing data

CHEMBL WEB SERVICES

- Written in Python
- RESTful design
- Apache 2.0 Licensed
- Available on **GitHub, PyPI**
- Providing data
- Providing chemical utilities

UTILITIES AND DATA



FILTERING

FILTERING

Select all approved drugs:
molecule?max_phase=4

FILTERING

Select all approved drugs:
molecule?max_phase=4

Select all approved drugs with two or more aromatic rings: **molecule?max_phase=4&molecule_properties__aromatic_rings__gte=2**

FILTERING

Select all approved drugs:
molecule?max_phase=4

Select all approved drugs with two or more aromatic rings: **molecule?max_phase=4&molecule_properties__aromatic_rings__gte=2**

Select all targets with name starting from 'serotonin':
target?pref_name__istartswith=serotonin

ORDERING

ORDERING

Order molecules by weight, ascending:

molecule?order_by=molecule_properties__full_mwt

ORDERING

Order molecules by weight, ascending:
molecule?order_by=molecule_properties__full_mwt

Order molecules by weight, descending:
**molecule?
molecule_properties__isnull=false&order_by=-
molecule_properties__full_mwt**

ORDERING

Order molecules by weight, ascending:
molecule?order_by=molecule_properties__full_mwt

Order molecules by weight, descending:
**molecule?
molecule_properties__isnull=false&order_by=-
molecule_properties__full_mwt**

Order by aromatic rings ascending and then by weight descending: **molecule?
order_by=molecule_properties__aromatic_rings&order_by=-
molecule_properties__full_mwt**

WANT VIAGRA?

WANT VIAGRA?

- Search compounds by pref_name and synonyms

WANT VIAGRA?

- Search compounds by pref_name and synonyms
- Search targets by pref_name and synonyms

WANT VIAGRA?

- Search compounds by pref_name and synonyms
- Search targets by pref_name and synonyms
- Search in assay descriptions

WANT VIAGRA?

- Search compounds by pref_name and synonyms
- Search targets by pref_name and synonyms
- Search in assay descriptions
- Search in document abstracts

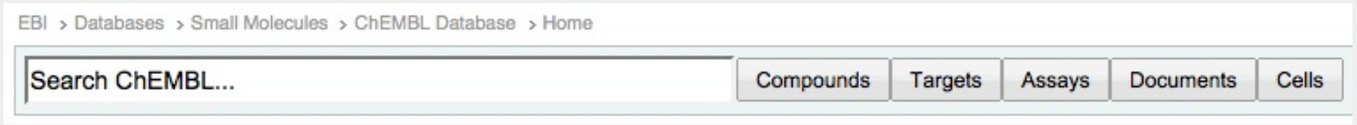
WHY DO WE NEED THIS?

WHY DO WE NEED THIS?

- User feedback (**#73**, **#79**)

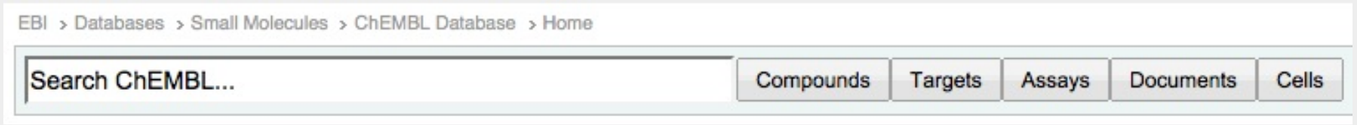
WHY DO WE NEED THIS?

- User feedback (**#73**, **#79**)
- New version of web site will be based on API



WHY DO WE NEED THIS?

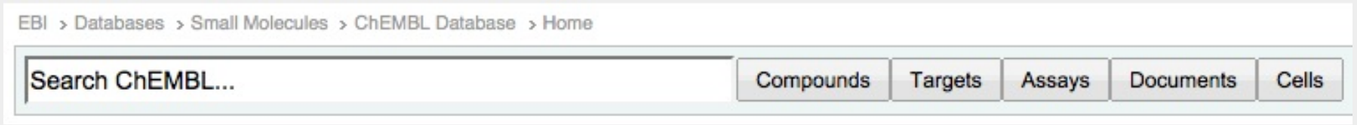
- User feedback (**#73**, **#79**)
- New version of web site will be based on API



- myChEMBL

WHY DO WE NEED THIS?

- User feedback (**#73**, **#79**)
- New version of web site will be based on API



- myChEMBL
- web widgets

IMPLMENTING SEARCH

IMPLMENTING SEARCH

- SQL would be too slow to execute

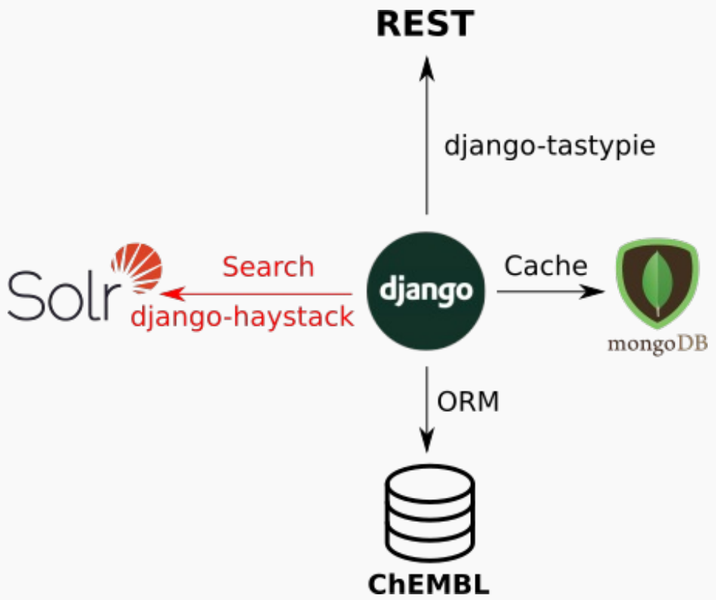
IMPLMENTING SEARCH

- SQL would be too slow to execute
- Speed depends on DB engine

IMPLMENTING SEARCH

- SQL would be too slow to execute
- Speed depends on DB engine
- Filtering simultaneously on many columns results in a long filter

SOFTWARE STACK



DJANGO-HAYSTACK

DJANGO-HAYSTACK

- Like ORM but for search

DJANGO-HAYSTACK

- Like ORM but for search
- Provides uniform API

DJANGO-HAYSTACK

- Like ORM but for search
- Provides uniform API
- Model-oriented

DJANGO-HAYSTACK

- Like ORM but for search
- Provides uniform API
- Model-oriented
- Many search backends supported:

DJANGO-HAYSTACK

- Like ORM but for search
- Provides uniform API
- Model-oriented
- Many search backends supported:
  - Solr

DJANGO-HAYSTACK

- Like ORM but for search
- Provides uniform API
- Model-oriented
- Many search backends supported:
  - Solr
  - Elastic

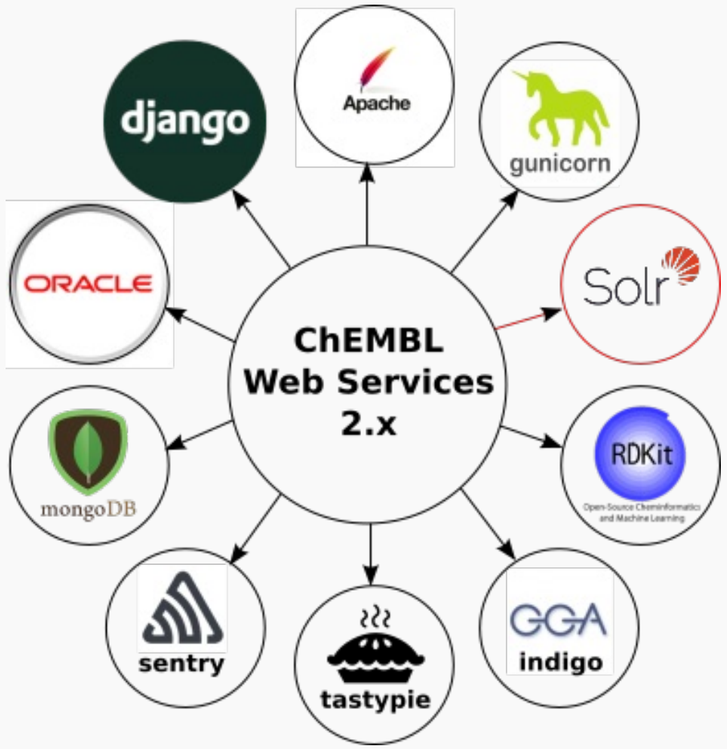
DJANGO-HAYSTACK

- Like ORM but for search
- Provides uniform API
- Model-oriented
- Many search backends supported:
  - Solr
  - Elastic
  - Whoosh

DJANGO-HAYSTACK

- Like ORM but for search
- Provides uniform API
- Model-oriented
- Many search backends supported:
  - Solr
  - Elastic
  - Whoosh
  - Xapian

TECHNOLOGIES USED



WHY SORL?

WHY SORL?

- Mature and trusted

WHY SORL?

- Mature and trusted
- Already used at EBI

WHY SORL?

- Mature and trusted
- Already used at EBI
- Apache license

WHY SORL?

- Mature and trusted
- Already used at EBI
- Apache license
- BioSolr, compound similarity

WHAT ABOUT ELASTIC?

WHAT ABOUT ELASTIC?

- Plans to use Kibi

WHAT ABOUT ELASTIC?

- Plans to use Kibi
- Put Kibi on myChEMBL first

WHAT ABOUT ELASTIC?

- Plans to use Kibi
- Put Kibi on myChEMBL first
- Make it public and evaluate

WHAT ABOUT ELASTIC?

- Plans to use Kibi
- Put Kibi on myChEMBL first
- Make it public and evaluate
- Kibi + sureChEMBL?

MONGODB

MONGODB

- Used as a cache backend

MONGODB

- Used as a cache backend
- Experiments with similarity search

MONGODB

- Used as a cache backend
- Experiments with similarity search
- LSH

MONGODB

- Used as a cache backend
- Experiments with similarity search
- LSH
- Good performance, good results

CACHE CHARACTERISTICS

CACHE CHARACTERISTICS

- Once cached, request won't change until next ChEMBL release

CACHE CHARACTERISTICS

- Once cached, request won't change until next ChEMBL release
- Cache should be shared across many production machines

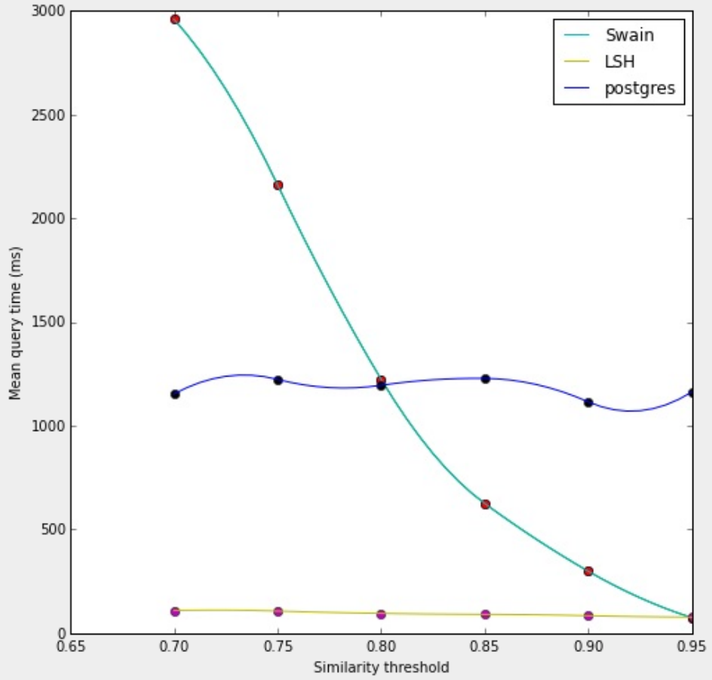
CACHE CHARACTERISTICS

- Once cached, request won't change until next ChEMBL release
- Cache should be shared across many production machines
- Available from python, supported by EBI

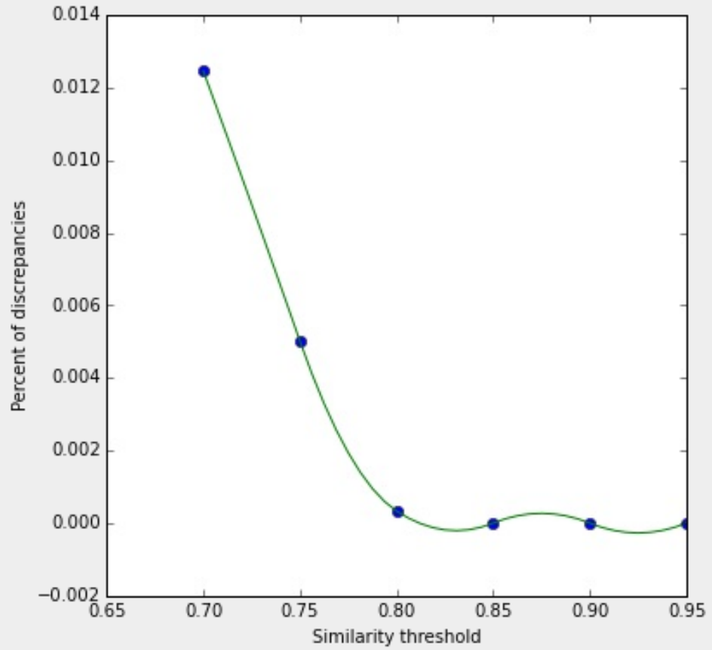
CACHE CHARACTERISTICS

- Once cached, request won't change until next ChEMBL release
- Cache should be shared across many production machines
- Available from python, supported by EBI
- Failproof, timeout

LSH-BASED SIMILARITY SEARCH



LSH-BASED SIMILARITY SEARCH



MYCHEMBL

MYCHEMBL

- special NoSQL issue

MYCHEMBL

- special NoSQL issue
- Kibi

MYCHEMBL

- special NoSQL issue
- Kibi
- Mining relations using Neo4j

MYCHEMBL

- special NoSQL issue
- Kibi
- Mining relations using Neo4j
- LSH-search on MongoDB

MYCHEMBL

- special NoSQL issue
- Kibi
- Mining relations using Neo4j
- LSH-search on MongoDB
- Ipython Notebooks

GRAPHQL

GRAPHQL

- New way of structuring client applications

GRAPHQL

- New way of structuring client applications
- User decides what API returns

GRAPHQL

- New way of structuring client applications
- User decides what API returns
- Fits perfectly into **Django stack**

GRAPHQL

- New way of structuring client applications
- User decides what API returns
- Fits perfectly into **Django stack**
- Very new idea (to us)

THANK YOU!

Questions?