ChEMBL REST API

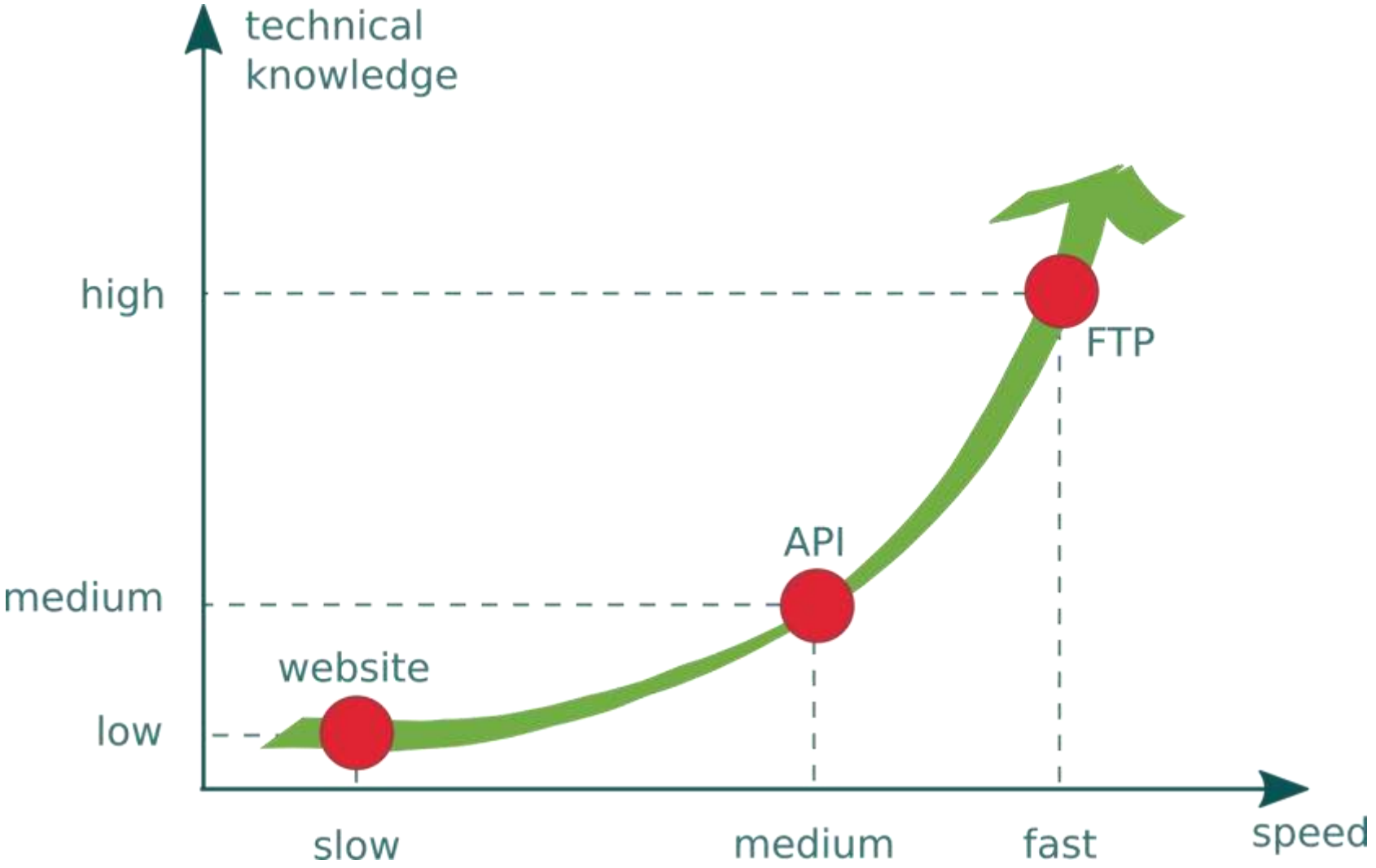
Practical introduction

Michał Nowotka
ChEMBL team

chembl-help@ebi.ac.uk

EMBL-EBI

ChEMBL data distribution channels



Web services - what's that?

"A Web service is a method of communication between two electronic devices over a network. [Wikipedia]"

"A software system designed to support interoperable machine-to-machine interaction over a network. [WC3]"

World Wide Web Consortium

Web services

- For machines
- But still human-readable
- Used in scripts, applications
- Can fetch data in bulk
- Web-friendly

ChEMBL API - where to start?

API main website (reference, examples, links):
https://www.ebi.ac.uk/chembl/ws

Live documentation
(execute simple API calls from your web browser):
https://www.ebi.ac.uk/chembl/api/data/docs

Source code:
https://github.com/chembl/chembl_webservices_2

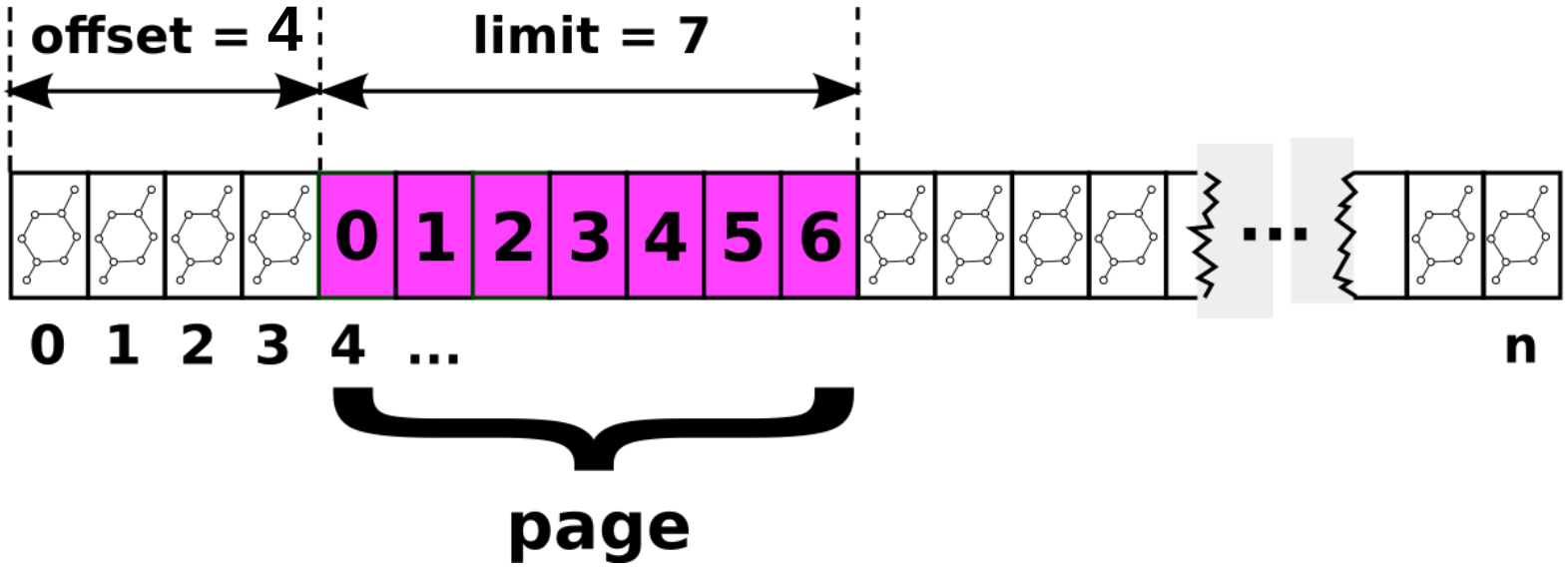
ChEMBL API basics - pagination

- ChEMBL contains 1.7M compounds and 14M activities
- You can’t get them all at once using the API
- Data is paginated
- Chunks are wrapped in envelopes

- Default page size is 20 but can be adjusted up to 1000.
- Pagination is controlled by two parameters:
  - limit
  - offset

Page anatomy

https://www.ebi.ac.uk/chembl/api/data/molecule/



Page anatomy

page.js

```
{
    "molecules":
[
{obj1},
{obj2},
...
{obj20}
],
"page_meta":
    {
        "limit": 20,
        "next": "/chembl/api/data/molecule.json?limit=20&offset=40",
        "offset": 20,
        "previous": "/chembl/api/data/molecule.json?limit=20&offset=0",
        "total_count": 1463270
    }
}
```

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

Filter types

- exact (iexact)
- contains (icontains)
- search
- startswith (istartswith)
- endswith (iendswith)
- regex (iregex)
- gt (gte)
- lt (lte)
- range
- in
- isnull

Ordering (ascending and descending)

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

Supported formats

- JSON
- XML
- YAML
- SDF
- PNG
- SVG

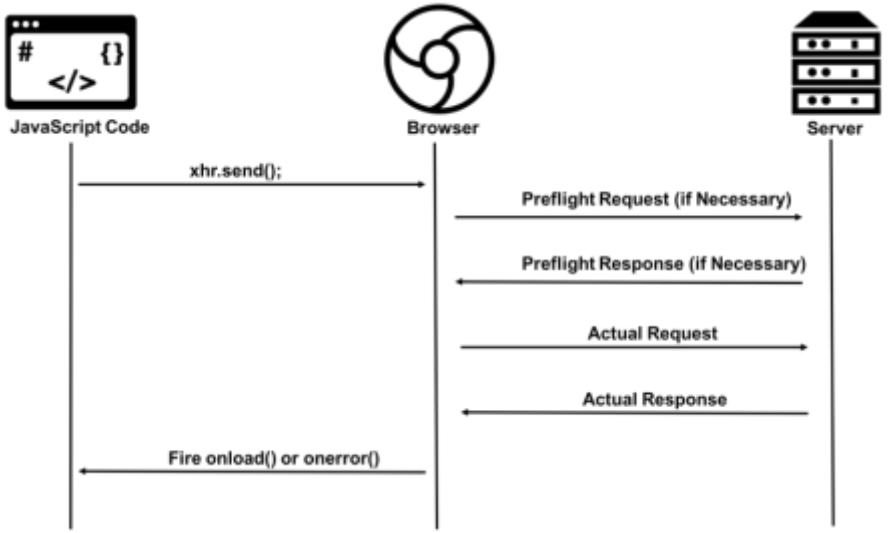
Supported HTTP methods

- GET
  - `curl -g "https://www.ebi.ac.uk/chembl/api/data/molecule/[Na+].CO[C@@H](CCC%23C\C=C/CCCC(C)CCCCC=C)C(=O)[O-]"`
  - `curl https://www.ebi.ac.uk/chembl/api/data/substructure/N%23CCc2ccc1ccccc1c2.json`

- POST (used as GET but with larger payload)
  - `curl -X POST -H "X-HTTP-Method-Override: GET" -H "Content-Type: application/json" -d '{"smiles":"N#CCc2ccc1ccccc1c2"}' https://www.ebi.ac.uk/chembl/api/data/substructure.json`
  - `curl -X POST -H "X-HTTP-Method-Override: GET" -d 'smiles=N#CCc2ccc1ccccc1c2' https://www.ebi.ac.uk/chembl/api/data/substructure.json`

In-browser cross-domain techniques supported

- CORS



- JSONP

JSON

```
{
 "roses": "red",
 "violets": "blue",
 "grass": "green"
}
```

JSONP

P for padding

```
grab({
 "roses": "red",
 "violets": "blue",
 "grass": "green"
})
```

SMILES support

Query SMILES:
[Na+].CO[C@@H](CCC#C\C=C/CCCC(C)CCCCC=C)C(=O)[O-]

- GET:
  - Wrong:
    https://www.ebi.ac.uk/chemblws/compounds/smiles/[Na+].CO[C@@H](CCC#C\C=C/CCCC(C)CCCCC=C)C(=O)[O-]
  - Right (for get use percent encoding):
    https://www.ebi.ac.uk/chembl/api/data/molecule/%5BNa+%5D.CO%5BC@@H%5D%28CCC%23C%5CC=C/CCCC%28C%29CCCCC=C%29C%28=O%29%5BO-%5D

- POST (no need to encode anything):
  - ```
    curl -X POST -H "X-HTTP-Method-Override: GET" -H "Content-Type: application/json" -d '{"smiles":"[Na+].CO[C@@H](CCC#C\\C=C/CCCC(C)CCCCC=C)C(=O)[O-]"}' https://www.ebi.ac.uk/chembl/api/data/molecule.json
    ```
    ```

Python ChEMBL client

- Handles pagination
- Takes care about HTTP
- Caches results locally
- Friendly syntax

```
molecule.get(['CHEMBL6498', 'CHEMBL6499', 'CHEMBL6505'])

molecule.filter(molecule_properties__acd_logp__gte=1.9)
.filter(molecule_properties__aromatic_rings__lte=3)
.filter(chirality=(-1))
.exists()

mol = utils.smiles2ctab("[Na]OC(=O)c1ccccc1")
br = utils.breakbonds(mol)
fingerprints = utils.sdf2fps(mol)
```

Examples

- Scripting
  - Bash (curl)
  - Python
  - R
- Web widgets
- Slack
- KNIME
- Jupyter (IPython)

Bash/Python/R scripts

The same task:
Read a CSV file with molecule ids and create the output CSV file with mapping between each compound and related target, optionally filtered by organism.

- Bash: ./mol2tar.sh
- R: ./mol2tar.r [-O "Homo sapiens"]
- Python: python mol2tar.py [-O "Homo sapiens"]

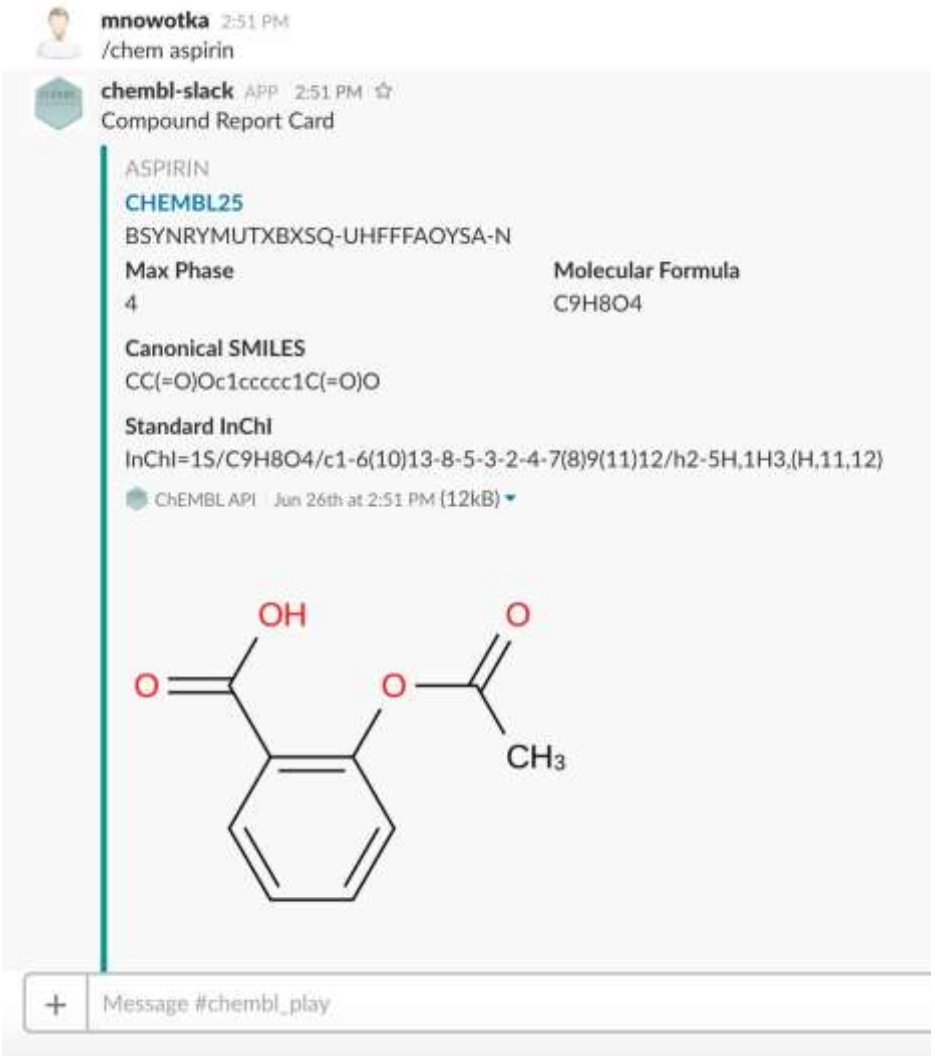
Bonus: bash script to fetch images for compounds in CSV file: ./mol2img.sh

Web widgets

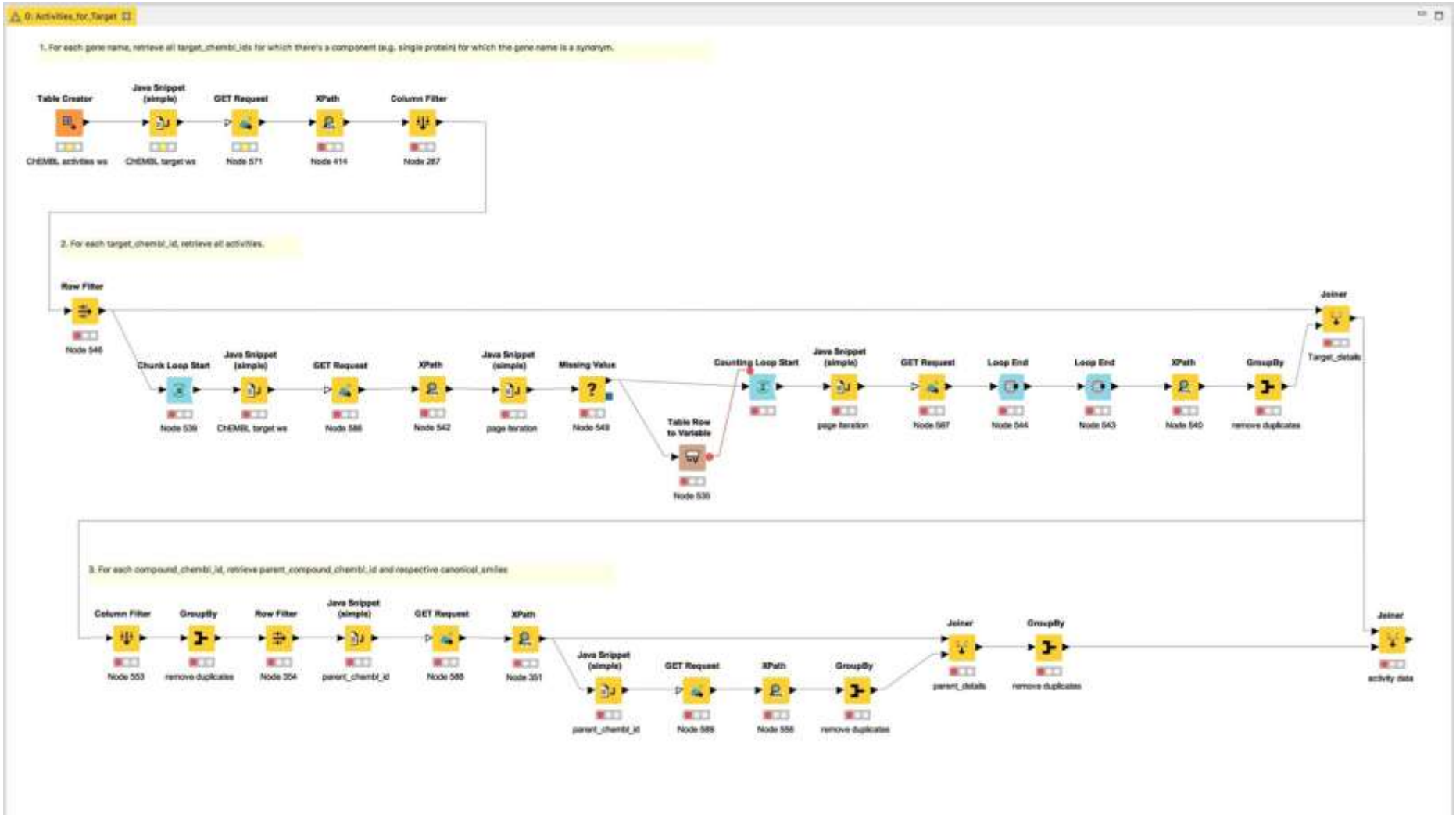
Compound Name and Classification

[fiddle 1](#) | [fiddle 2](#) | [fiddle 3](#)

Slack bot


mnowotka 2:51 PM
/chem aspirin
chembl-slack APP 2:51 PM
Compound Report Card
ASPIRIN
CHEMBL25
BSYNRYMUTXBXSQ-UHFFFAOYSA-N
Max Phase
4
Molecular Formula
C9H8O4
Canonical SMILES
CC(=O)Oc1ccccc1C(=O)O
Standard InChI
InChI=1S/C9H8O4/c1-6(10)13-8-5-3-2-4-7(8)9(11)12/h2-5H,1H3,(H,11,12)
ChEMBL API Jun 26th at 2:51 PM (12kB)
OH
O
O
O
CH3
Message #chembl_play

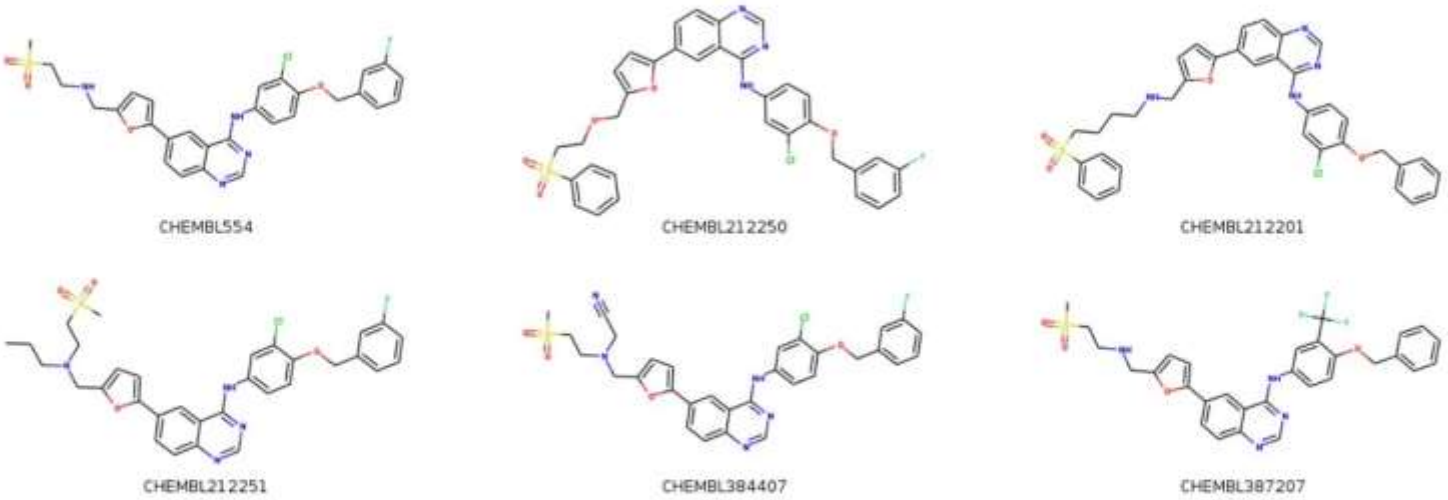
KNIME


0: Activities_for_Target
1. For each gene name, retrieve all target_chembl_ids for which there's a component (e.g. single protein) for which the gene name is a synonym.
Table Creator
Java Snippet (simple)
GET Request
XPath
Column Filter
CHEMBL activities ws
CHEMBL target ws
Node 571
Node 414
Node 287
2. For each target_chembl_id, retrieve all activities.
Row Filter
Node 546
Chunk Loop Start
Java Snippet (simple)
GET Request
XPath
Java Snippet (simple)
Missing Value
Node 539
CHEMBL target ws
Node 586
Node 542
page iteration
Node 548
Table Row to Variable
Node 535
Counting Loop Start
Java Snippet (simple)
GET Request
Loop End
Loop End
XPath
GroupBy
Joiner
page iteration
Node 587
Node 544
Node 543
Node 540
remove duplicates
Target_details
3. For each compound_chembl_id, retrieve parent_compound_chembl_id and respective canonical_smiles
Column Filter
GroupBy
Row Filter
Java Snippet (simple)
GET Request
XPath
Node 553
remove duplicates
Node 354
parent_chembl_id
Node 588
Node 351
Java Snippet (simple)
GET Request
XPath
GroupBy
parent_chembl_id
Node 589
Node 556
remove duplicates
Joiner
GroupBy
Joiner
parent_details
remove duplicates
activity data

Jupyter/IPython

```
http://localhost/chemblws/substructure/c4ccc%28Nc2ncnc3ccc%28c1ccco1%29cc23%29cc4.json
82
```

```
mols = [Chem.MolFromSmiles(x['molecule_structures']['canonical_smiles']) for x in records[:6]]
legends=[str(x["molecule_chembl_id"]) for x in records]
Draw.MolsToGridImage(mols, legends=legends, subImgSize=(400, 200), useSVG=False)
```



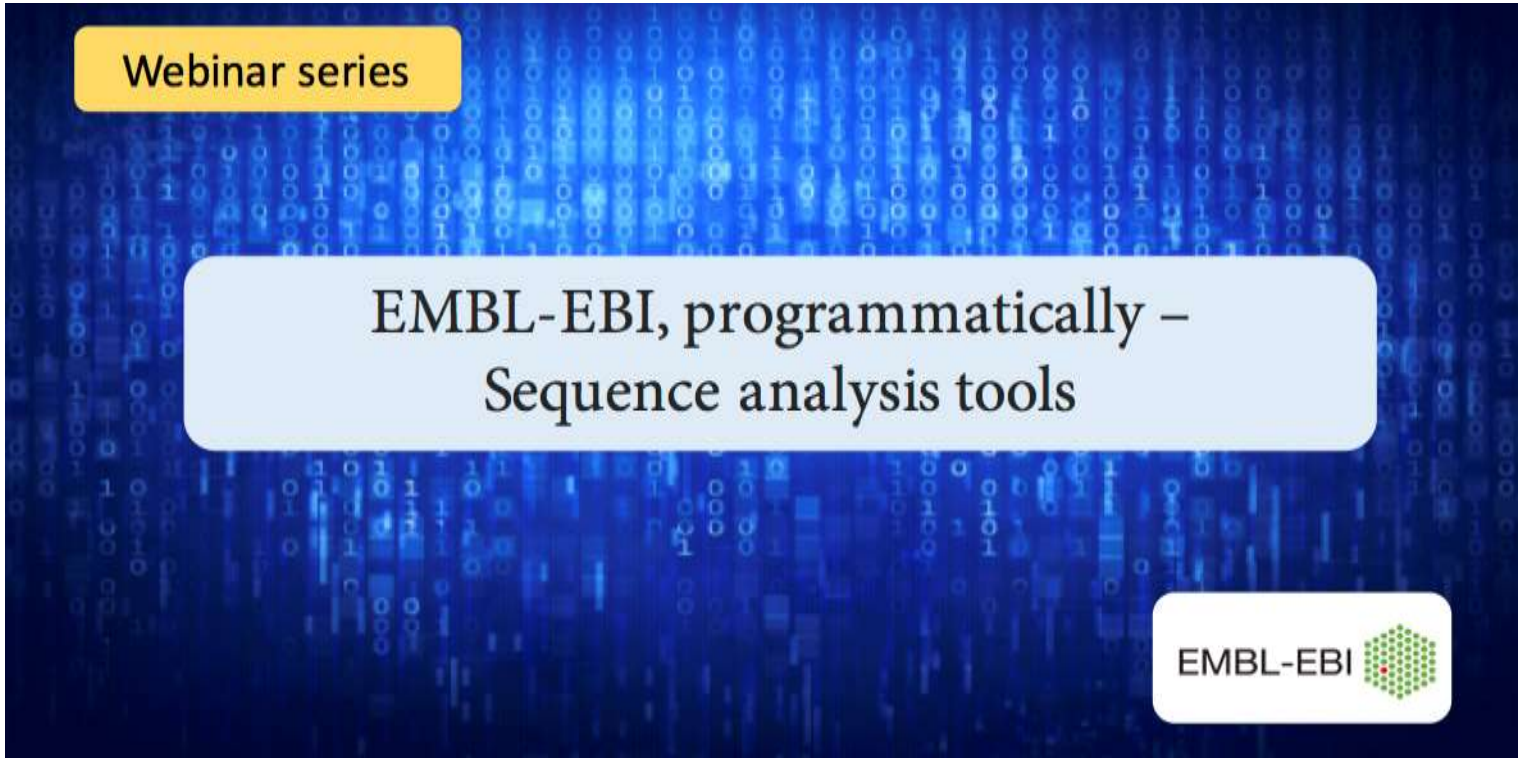
Similarity searching

The web services may also be used to perform SMILES-based similarity searches.

```
# Lapatinib
smiles = "CS(=O)(=O)CCNCc1oc(cc1)c2ccc3ncnc(Nc4ccc(OCc5cccc(F)c5)c(Cl)c4)c3c2"
```

Next in the series

See the full list of upcoming webinars at
http://www.ebi.ac.uk/training/webinars



Feedback

Tell us what you think