



ENHANCE
YOUR
DATA.



dct:title

Breakout Session II: Hands on Data Annotation using Ontologies

Creating a prototype knowledge graph from NMR spectroscopy research data

dct:identifier

[DOI:10.5281/zenodo.7050763](https://doi.org/10.5281/zenodo.7050763)

dct:creator

[ORCID:0000-0002-1595-3213](https://orcid.org/0000-0002-1595-3213)

ro:'participates in' 1st Ontologies4Chem Workshop 2022 – Ontologies for chemistry (day 2)

dct:identifier

[DOI:10.25798/frnp-sn04](https://doi.org/10.25798/frnp-sn04)

sdo:duration

2022-09-08T16:50:00+02:00/PT10M

Creating knowledge graphs from research data

the preconditions - domain specific metadata



- it is hard to query & aggregate metadata from different sources
 - need to know provider specific metadata schemata
 - need to map it to a common ground (e.g. see <https://search.nfdi4chem.de/repository>)
 - no domain specific standard schema yet (possible candidate [IUPAC FAIRspec](#))
 - no clear border between metadata and actual data (e.g. NMR pulse sequence)

typeClass:	"compound"
▼ value:	
▼ 0:	
▶ topicClassValue:	{...}
▶ topicClassVocab:	{...}
▼ topicClassVocabURI:	
typeName:	"topicClassVocabURI"
multiple:	false
typeClass:	"primitive"
value:	"https://pubchem.ncbi.nlm.nih.gov/compound/5280805"

▼ tag:	
id:	574047
taggable_type:	"Molecule"
taggable_id:	3366
▼ taggable_data:	
▼ chemotion:	
doi:	"10.14272/QXXCRBSWGPRLJ-UHFFFAOYSA-N.1"
chemotion_first:	"2022-01-10T14:48:05.327+01:00"
last_published_at:	"2022-01-10T14:48:13.202+01:00"
pubchem_cid:	162394348

Creating knowledge graphs from research data

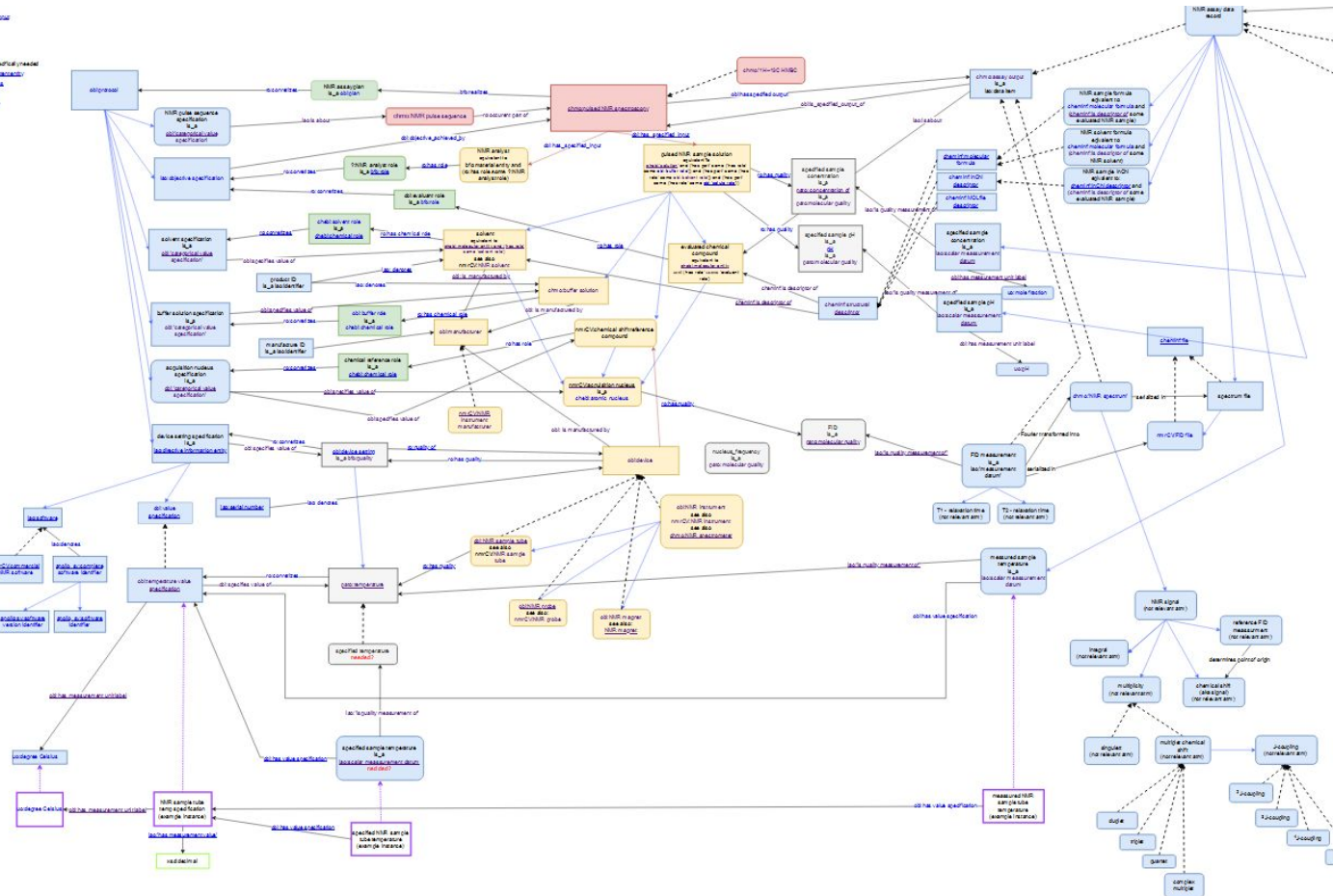
the preconditions - suitable ontologies already exist



- RDF knowledge graphs (KGs) as alternative or supplement
 - make federated queries with SPARQL
 - only need to know about the used ontology terms
 - fosters interdisciplinary work
- KGs from parsing research data? → proof of concept
 - make KGs from existing datasets
 - make federated queries on them with SPARQL

Creating knowledge graphs from research data

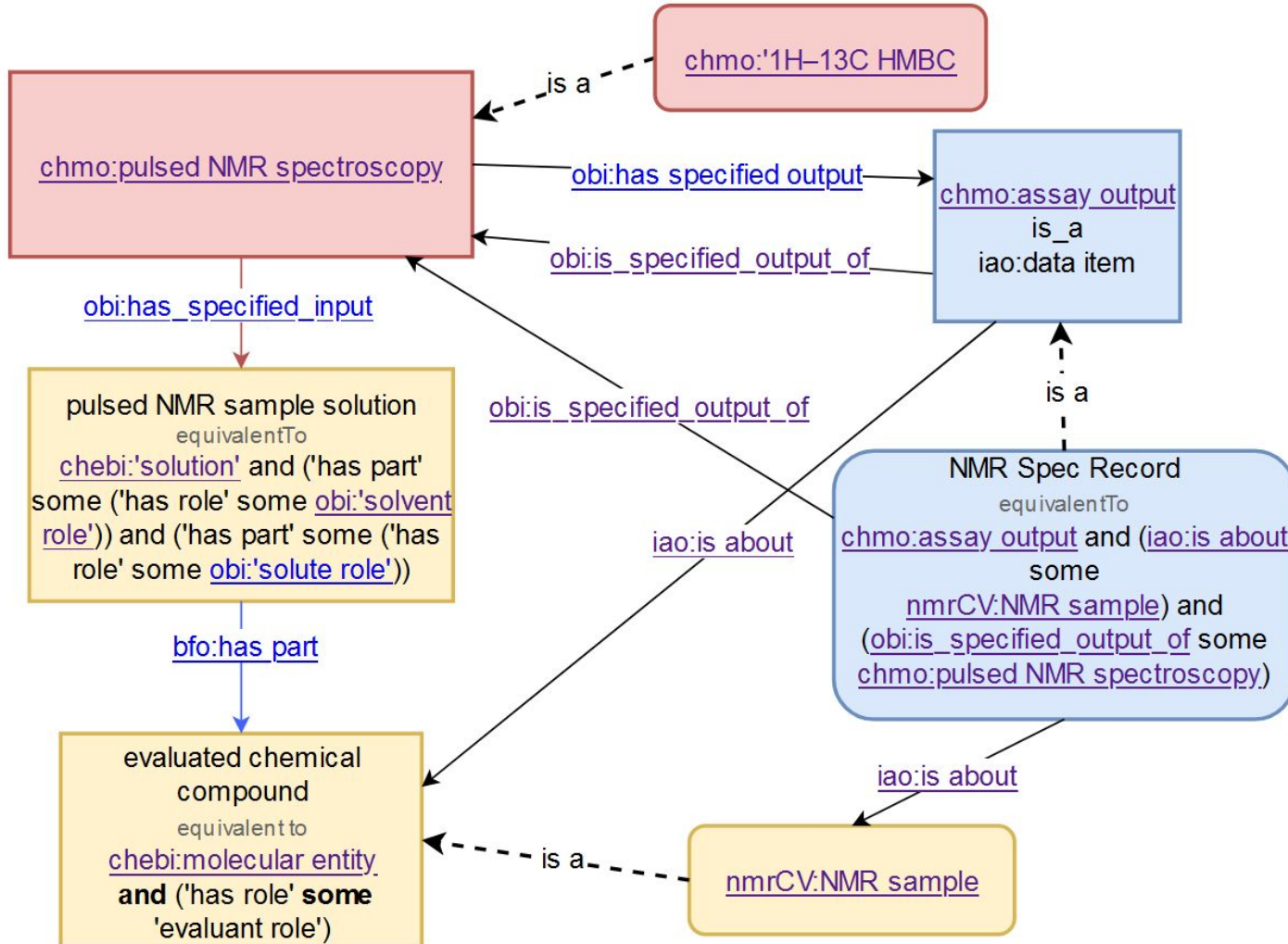
Step 1: define a terminology box (TBox)



- finding the right terms to describe NMR spectroscopy
- reusing as many terms as possible for interoperability
- we found most in:
 - CHMO
 - nmrCV
 - CHEBI
 - CHEMINF
 - OBI
 - IAO

Creating knowledge graphs from research data

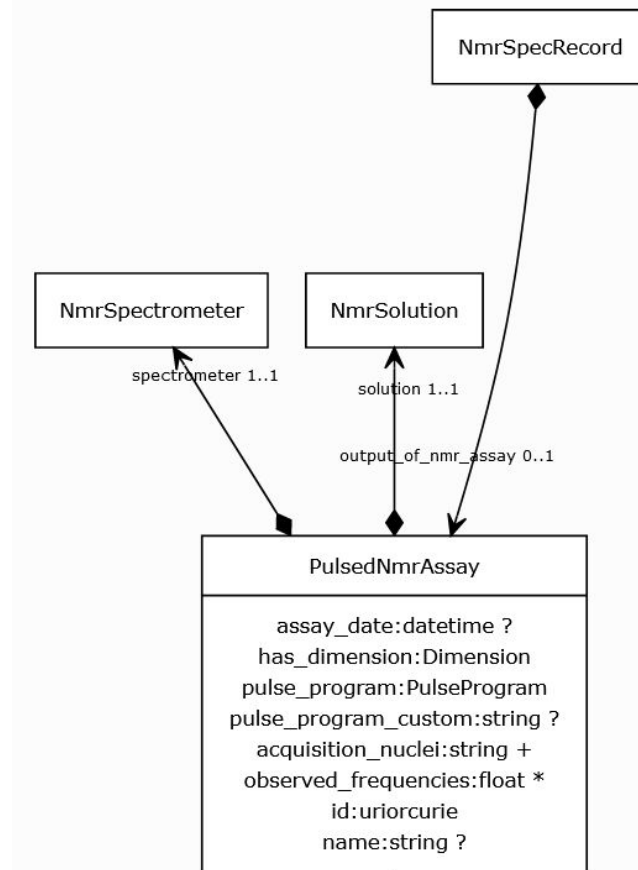
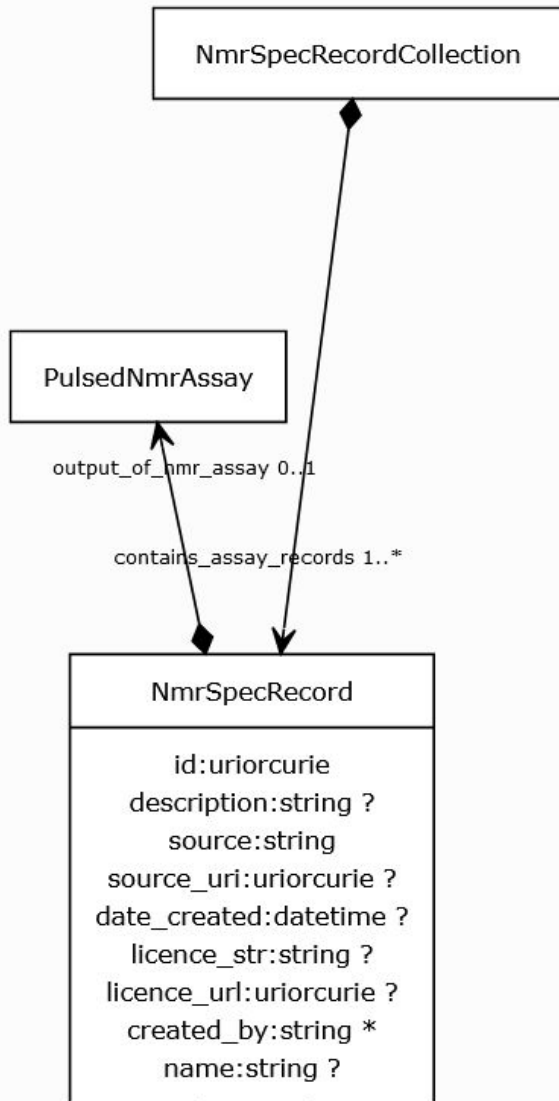
Step 1: define a terminology box (TBox)




- need to define new entities
 - in existing ontologies
 - one of our later TODOs
 - involves discussion with domain experts
 - semantically more precise

Creating knowledge graphs from research data

Step 2: define the shape of the knowledge graph



- <https://stroemphi.github.io/NMRspec/>
- using <https://linkml.io> framework 
- prototypical → not complete
- focus on pulsed NMR spectroscopy
 - only links to broader assay context

Creating knowledge graphs from research data

Step 2: define the shape of the knowledge graph



```
NMRspec.yaml
40 - semweb_context
41 imports:
42 - linkml:types
43 - ./Provenance
44 emit_prefixes:
45   nmrSPARQL
46 #####
47 classes:
48 # Mixins
49   ChemicalDescriptor: {}
82   NamedThing: {}
89
90 # processes
91 PulsedNmrAssay:
92   mixins:
93     - NamedThing
94   class_uri: chmo:0000613
95   slot_usage:
96     id:
97       ifabsent: uri(nmrSPARQL:Assay)
98   attributes:
99     assay_date: {}
105    solution: {}
112    spectrometer:
113      slot_uri: obi:0000293
114      inlined: true
115      required: true
116      domain: PulsedNmrAssay
117      range: NmrSpectrometer
118    has_dimension:
119      required: true
120      ifabsent: string(1D)
121      range: Dimension
122    pulse_program:
123      description: The pulse program of a PulsedNmrAssay is a required property and must be conform to one of the values defined in the PulseProgram enum class.
```

This model is to be used to semantify NMR spectroscopy research data.

Search docs

Home

Index

- Classes
- Mixins
- Slots
- Enums
- Subsets
- Types
- Credits

NMRspec

metamodel version: 1.7.0

version: 0.0.1

This model is to be used to semantify NMR spectroscopy research data.

Classes

- MOLfile
- Manufacturer
- MolarConcentration - A quality inhering in a substance by virtue of the amount of the bearer's there is mixed with another substance. [Wikipedia:<http://en.wikipedia.org/wiki/concentration>]
- MolarityMeasurementDatum - A scalar measurement datum that is the result of measuring the molarity (aka volume concentration) of a chemical solution.
- NmrBuffer
- NmrSample
- NmrSampleTube
- NmrSolution - A NMR solution is the solution made up of the solvent in which the evaluated sample is dissolved in as well as possibly the buffer needed to adjust the pH value.
- NmrSolvent

```
dict2yaml.py x NMRspec.py x Pipfile x jdx.py x
Project
  NMR-schema C:\Users\stroem
  .github
  docs
  make-venv
  model
  NMRspec
    jdx_files
    jsonld
    jsonschema
    model
    _init_.py
    dict2yaml.py
    jdx.py
    NMRspec.py
    Provenance.py
    test_nmrsec.py
    yaml2rdf.bat
  target
  tests
  venv
  .gitignore
  ABOUT.md
  MakeConfig
  Makefile

@dataclass
class PulsedNmrAssay(YAMLRoot):
    _inherited_slots: ClassVar[List[str]] = []

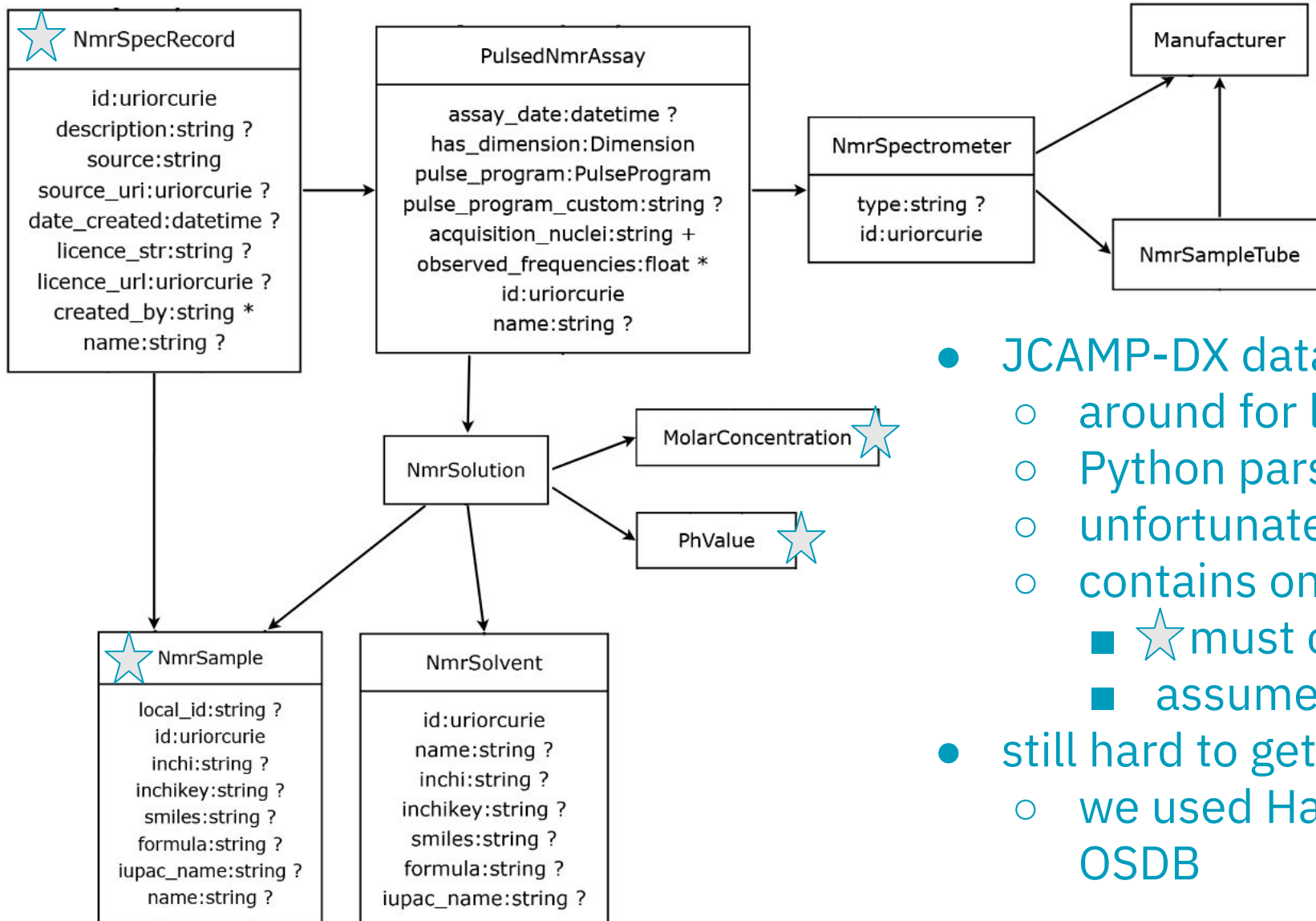
    class_class_uri: ClassVar[URIRef] = CHMO["0000613"]
    class_class_curie: ClassVar[str] = "chmo:0000613"
    class_name: ClassVar[str] = "PulsedNmrAssay"
    class_model_uri: ClassVar[URIRef] = NMRSPEC.PulsedNmrAssay

    solution: Union[dict, "NmrSolution"] = None
    spectrometer: Union[dict, "NmrSpectrometer"] = None
    acquisition_nuclei: Union[str, List[str]] = None
    has_dimension: Union[str, "Dimension"] = "1D"
    pulse_program: Union[str, "PulseProgram"] = "not provided"
    id: Union[str, PulsedNmrAssayId] = NMRSPARQL.Assay
    assay_date: Optional[Union[str, XSDDateTime]] = None
    pulse_program_custom: Optional[str] = None
    observed_frequencies: Optional[Union[float, List[float]]] = empty_list()
    name: Optional[str] = None
```

hema. It is a data item that serves as a
: spectroscopy assay record. The properties
MR assay, such as: ' what kind of assay was
are used for that (e.g. spectrometer,
of the assayed sample (name, formula,
ollection of NMR spectroscopy records. It is
order to analyse a given sample.
n that is the result of measuring the pH
nent datum that is the result of measuring

Creating knowledge graphs from research data

Step 3: parsing values from JCAMP-DX



- JCAMP-DX datasets as source for NMR
 - around for long and open
 - Python parser already exists
 - unfortunately many dialects :(
 - contains only limited metadata
 - ★ must come from other sources
 - assumed to be auto-generated in ELNs
- still hard to get many .jdx files
 - we used Havard Dataverse, Chemotion, OSDB

Creating knowledge graphs from research data

Step 3: parsing values from JCAMP-DX



```
dataset_info.yaml
Typ: YAML-Datei

Limonene_7020ug200uL_CDC13_13CNMR_400MHz_JDX.jdx
Typ: JDX-Datei

Limonene_7020ug200uL_CDC13_COSY_400MHz_JDX.jdx
Typ: JDX-Datei

Limonene_7020ug200uL_CDC13_HMBC_400MHz_JDX.jdx
Typ: JDX-Datei

dataset_info.yaml
9 licence_url: https://creativecommons.org/share-your-work/public-domain/cc0
10 date_created: "2019-10-28"
11 description: "NMR data of D-Limonene in DMSOd6. The dataset contains 1D 1H 13C as
well as 2D COSY, HSQC, HMBC, all acquired at 400 MHz (Jeol 400 MHz spectrometer
with SuperCOOL Probe) (2019-10-07). Related Publication Can Invalid Bioactives
Undermine Natural Product-Based Drug Discovery? J. Med. Chem. doi: 10.1021/
acs.jmedchem.5b01009 https://doi.org/10.1021/acs.jmedchem.5b01009. Related
Material Indofine 5989-27-5 Lot #025082s 7.02 mg 3mm Tube Cambridge Isotope
DML-10-10X1 Lot #10E-645 Solvent volume 200 µL."
12 id: doi:10.7910/DVN/2UEA9M
13 name: D-Limonene 400 MHz in DMSOd6 NMR data
14 created_by:
15   - "Kim, Seon Beom (University of Illinois at Chicago) - ORCID: 0000-0001-8015-1404"
16   - "Simmler, Charlotte (University of Illinois at Chicago) - ORCID:
0000-0002-6923-2630"
17   - "Bisson, Jonathan (University of Illinois at Chicago) - ORCID: 0000-0003-1640-9989"
18   - "Pauli, Guido (University of Illinois at Chicago) - ORCID: 0000-0003-1022-4326"
19
20 #####
21 # sample metadata #
22 #####
23 assays_sample:
24   # you MUST provide at least one of the following metadata on the assayed sample
25   name: D-Limonene
26   id: pubchemCID:440917
27   formula: C10H16
28   iupac_name: (4R)-1-methyl-4-prop-1-en-2-ylcyclohexene
29   inchi: InChI=1S/C10H16/c1-8(2)10-6-4-9(3)5-7-10/h4,10H,1,5-7H2,2-3H3/t10-m/s1
30   inchikey: XMGQYMMWDOXHJM-JTQLQIEISA-N
31   smiles: CC1=CCC(CC1)C=C
32   mol_file:
33     id: doi:10.7910/DVN/2UEA9M/JOCQYF
34     name: D-Limonene.mol
35
36 #####
37 # records in dataset #
38 #####
39 contains_assay_records:
40   # you MUST provide at least an URI as the id of each assay record (.jdx) in this
dataset. If the records do not have an URI, you must use "nmrSPARQL:" as prefix and
the filename (e.g. "id: nmrSPARQL:filename1.jdx")
41   - id: nmrSPARQL:Limonene_7020ug200uL_CDC13_13CNMR_400MHz_JDX
42     source: Limonene_7020ug200uL_CDC13_13CNMR_400MHz_JDX.jdx
```

- dataset_info.yaml with metadata not in JDX files
 - dataset provenance (source, creators, PID, ...)
 - metadata of the assayed sample (InChI, InChIKey, SMILES, ...)
- dict2yaml.py
 - parse JDX files in batch
 - create one NMR record collection as YAML
 - each JDX is one NMRSpecRecord
- yaml2rdf.bat
 - convert the YAML to TTL

Creating knowledge graphs from research data

Step 4: load the knowledge graphs into a triple store



- NMR research metadata expressed in triple statements
- example: [.../output/\(%2B\)-Tetrandrine_400_600_MHz_in_DMSOd6_NMR_data.ttl](#)

```
<https://doi.org/10.7910/DVN/F34GVS/KZOQZU/1HNMR_jdx> a ns1:NmrSpecRecord ;
  ns2:OBI_0000312 <https://doi.org/10.7910/DVN/F34GVS/KZOQZU/1HNMR_jdx/Assay> ;
  dc:source "1HNMR_jdx.jdx" .

<https://doi.org/10.7910/DVN/F34GVS/KZOQZU/1HNMR_jdx/Assay> a ns2:CHMO_0000613 ;
  ns2:OBI_0000293 <https://doi.org/10.7910/DVN/F34GVS/KZOQZU/1HNMR_jdx/Solution>,
    <https://doi.org/10.7910/DVN/F34GVS/KZOQZU/1HNMR_jdx/Spectrometer> ;
  ns1:acquisition_nuclei "1H" ;
  ns1:has_dimension "1D" ;
  ns1:observed_frequencies "399.78219837825"^^xsd:float ;
  ns1:pulse_program "NMR" .
```

SPARQL Queries



Show me only 1H NMR Spectra with a resolution higher 400 MHz

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX OBO: <http://purl.obolibrary.org/obo/>

PREFIX NMR: <https://raw.githubusercontent.com/StroemPhi/NMRspec/main/model/schema/>

SELECT ?dataset ?frequency

WHERE {

?dataset NMR:pulse_program "NMR".

?dataset NMR:acquisition_nuclei "1H".

?dataset NMR:observed_frequencies ?frequency

FILTER(?frequency >= 400)

}

The screenshot shows a SPARQL query interface with the following content:

SPARQL Query
To try out some SPARQL queries against the selected dataset, enter your query here.

Example Queries: Selection of triples, Selection of classes

Prefixes: rdf, rdfs, owl, xsd

Content Type (SELECT): JSON, Content Type (GRAPH): Turtle

```
4 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
5 PREFIX OBO: <http://purl.obolibrary.org/obo/>
6 PREFIX NMR: <https://raw.githubusercontent.com/StroemPhi/NMRspec/main/model/schema/>
7
8 SELECT ?dataset ?frequency
9 WHERE {
10 ?dataset NMR:pulse_program "NMR".
11 ?dataset NMR:acquisition_nuclei "1H".
12 ?dataset NMR:observed_frequencies ?frequency
13 FILTER( ?frequency >= 400)
14 }
15
16
17
```

6 results in 0.039 seconds

dataset	frequency
<https://doi.org/10.7910/DVN/KUEDTG/Linalool_10000ug200uL_CDCI3_1HNMR_900MHz_JDX/Assay>	"900.077600295" <http://www.w3.org/2001/XMLSchema#float>
<https://doi.org/10.7910/DVN/M8QR20/VEIWX8/Lupeol_LUPE01_q1H/Assay>	"900.077600295" <http://www.w3.org/2001/XMLSchema#float>
<https://doi.org/10.7910/DVN/SAC0TQ/SNCCAF/Tetrandrine_3090ug200uL_DMSOd6_1HNMR_600MHz_JDX/Assay>	"600.1536009" <http://www.w3.org/2001/XMLSchema#float>
<https://doi.org/10.14272/QXXCRBSWGPRILJ-UHFFFAOYSA-N/CHMO0000595/Assay>	"400.13160052" <http://www.w3.org/2001/XMLSchema#float>
<https://doi.org/10.14272/QXXCRBSWGPRILJ-UHFFFAOYSA-N/CHMO0000593/Assay>	"400.132470802" <http://www.w3.org/2001/XMLSchema#float>

Federated SPARQL Queries



Federated query across Research Data Triplestore and Wikidata based on PubChemId

PREFIX wdt: <http://www.wikidata.org/prop/direct/>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX NMR: <https://raw.githubusercontent.com/StroemPhi/NMRspec/main/m

SELECT ?NMRSample ?pubchemId ?item WHERE {

 ?NMRSample rdf:type NMR:NmrSample.

 BIND(REPLACE(str(?NMRSample),

 "https://pubchem.ncbi.nlm.nih.gov/compound/", ""))

 AS ?pubchemId).

 SERVICE <<https://query.wikidata.org/bigdata/namespace/wdq/sparql>>

 {

 ?item wdt:P662 ?pubchemId.

 }

}

The screenshot shows a SPARQL query interface with the following elements:

- Buttons: query, add data, edit, info
- Section: SPARQL Query
- Text: To try out some SPARQL queries against the selected dataset, enter your query here.
- Example Queries: Selection of triples, Selection of classes
- Prefixes: rdf, rdfs, owl, xsd
- Content Type (SELECT): JSON
- Content Type (GRAPH): Content Type (GRAPH)
- Query text (lines 1-14):

```
1 PREFIX wdt: <http://www.wikidata.org/prop/direct/>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX owl: <http://www.w3.org/2002/07/owl#>
4 PREFIX wdt: <http://www.wikidata.org/prop/direct/>
5 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
6 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
7 PREFIX NMR: <https://raw.githubusercontent.com/StroemPhi/NMRspec/main/model/schema/>
8 SELECT ?NMRSample ?pubchemId ?item WHERE {
9     ?NMRSample rdf:type NMR:NmrSample.
10    BIND(REPLACE( str(?NMRSample), "https://pubchem.ncbi.nlm.nih.gov/compound/", ""))
11        SERVICE <https://query.wikidata.org/bigdata/namespace/wdq/sparql> {
12            ?item wdt:P662 ?pubchemId.
13        }
14 }
```
- Results: 9 results in 2.932 seconds
- Table view with columns: NMRSample, pubchemId, item
- Table data (rows 1-5):

NMRSample	pubchemId	item
<https://pubchem.ncbi.nlm.nih.gov/compound/2758>	2758	<http://www.wikidata.org/entity/Q161572>
<https://pubchem.ncbi.nlm.nih.gov/compound/1548943>	1548943	<http://www.wikidata.org/entity/Q273169>
<https://pubchem.ncbi.nlm.nih.gov/compound/5280805>	5280805	<http://www.wikidata.org/entity/Q407857>
<https://pubchem.ncbi.nlm.nih.gov/compound/64945>	64945	<http://www.wikidata.org/entity/Q416260>
<https://pubchem.ncbi.nlm.nih.gov/compound/222284>	222284	<http://www.wikidata.org/entity/Q121802>

Federated SPARQL Queries



Find matching entries in further data repository: Like finding a corresponding Mass Spectra in MassBank

```
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX NMR: <https://raw.githubusercontent.com/StroemPhi/NMRspec/main/
SELECT ?NMRSample ?pubchemId ?item WHERE {
    ?NMRSample rdf:type NMR:NmrSample.
    BIND(REPLACE( str(?NMRSample),
        "https://pubchem.ncbi.nlm.nih.gov/compound/", ""))
        AS ?pubchemId).
    SERVICE <https://query.wikidata.org/bigdata/namespace/wdq/sparql>
    {
        ?item wdt:P662 ?pubchemId.
        ?item wdt:P6689 ?massbank.
    }
}
```

The screenshot shows a SPARQL query interface with the following elements:

- Buttons: query, add data, edit, info
- Section: SPARQL Query
- Text: To try out some SPARQL queries against the selected dataset, enter your query here.
- Example Queries: Selection of triples, Selection of classes
- Prefixes: rdf, rdfs, owl, xsd
- Content Type (SELECT): JSON
- Content Type (GRAPH):
- Query text (lines 1-14):

```
1 PREFIX wdt: <http://www.wikidata.org/prop/direct/>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX owl: <http://www.w3.org/2002/07/owl#>
4 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
5 PREFIX OBO: <http://purl.obolibrary.org/obo/>
6 PREFIX NMR: <https://raw.githubusercontent.com/StroemPhi/NMRspec/main/model/schema/>
7 SELECT ?NMRSample ?massbank WHERE {
8   ?NMRSample rdf:type NMR:NmrSample.
9   BIND(REPLACE( str(?NMRSample), "https://pubchem.ncbi.nlm.nih.gov/compound/", "")) AS ?pubchemId.
10  SERVICE <https://query.wikidata.org/bigdata/namespace/wdq/sparql> {
11    ?item wdt:P662 ?pubchemId.
12    ?item wdt:P6689 ?massbank.
13  }
14 }
```
- Response: 54 results in 3.451 seconds
- Table with columns: NMRSample, massbank

NMRSample	massbank
<https://pubchem.ncbi.nlm.nih.gov/compound/1548943>	PR100259
<https://pubchem.ncbi.nlm.nih.gov/compound/1548943>	PR100683
<https://pubchem.ncbi.nlm.nih.gov/compound/1548943>	TY000093
<https://pubchem.ncbi.nlm.nih.gov/compound/1548943>	TY000245
<https://pubchem.ncbi.nlm.nih.gov/compound/1548943>	WA001601



Thank you