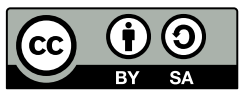
Session 3. Under the hood: rolling out your own scientific database

Saulius Gražulis
grazulis@ibt.lt

Institute of Biotechnology
Life Sciences Center
Vilnius University

USNC/Cr Database Workshop 2022



Id: slides.tex 1805 2022-04-14 13:56:53Z saulius April 14, 2022

Contents of this talk

1. Why do we need databases?
2. Principles of database construction
3. A specific example: let's construct a P1 cell database!

Contents of this talk

1. Why do we need databases?
2. Principles of database construction
3. A specific example: let's construct a P1 cell database!

Unreadable papers

Only in biomedical sciences, a paper is published about every 2 minutes!

https://www.nlm.nih.gov/bsd/medline_cit_counts_yr_pub.html

Home > MEDLINE/PubMed Resources

MEDLINE® Citation Counts by Year of Publication (as of January 2021)*

MEDLINE consists of completed citations indexed with MeSH® (Medical Subject Headings®).

Year of Publication	Total # Citations	# Citations Published in US	%s Citations Published in US
2020*	362,528	138,112	38%
2019	898,145	345,923	39%
2018	866,977	343,605	40%
2017	848,776	343,947	41%
2016	862,829	351,138	41%
2015	878,403	367,373	43%

PubMed.gov
Go to PubMed

$$898145/(365 \times 24 \times 60) = 1.7 \text{ papers/min}$$

New ways of publishing science

Suggesting a new way of communicating scientific results:

A peer-reviewed database record

New ways of publishing science

Suggesting a new way of communicating scientific results:

A peer-reviewed database record

Advantages:

- Machine-readable

New ways of publishing science

Suggesting a new way of communicating scientific results:

A peer-reviewed database record

Advantages:

- Machine-readable
- Allows comprehensive searches

New ways of publishing science

Suggesting a new way of communicating scientific results:

A peer-reviewed database record

Advantages:

- Machine-readable
- Allows comprehensive searches
- Allows insights from large data not available otherwise

New ways of publishing science

Suggesting a new way of communicating scientific results:

A peer-reviewed database record

Advantages:

- Machine-readable
- Allows comprehensive searches
- Allows insights from large data not available otherwise
- Faster to produce, more efficient to use

FAIR data exchange principles

Data should be [Wilkinson et al. (2016)]:

- **F**indable
- **A**cessible
- **I**nteroperable
- **R**eusable

Scientific databases

Definition

A ***scientific database*** is a an organized collection of *reliable scientific* data of *known provenance*, stored in a machine-readable form, with automated search and processing capabilities.

Scientific databases

Definition

A ***scientific database*** is a an organized collection of *reliable scientific* data of *known provenance*, stored in a machine-readable form, with automated search and processing capabilities.

- organized;

Scientific databases

Definition

A ***scientific database*** is a an organized collection of *reliable scientific* data of *known provenance*, stored in a machine-readable form, with automated search and processing capabilities.

- organized;
- machine-readable;

Scientific databases

Definition

A ***scientific database*** is a an organized collection of *reliable scientific* data of *known provenance*, stored in a machine-readable form, with automated search and processing capabilities.

- organized;
- machine-readable;
- reliable (known provenance);

Scientific databases

Definition

A ***scientific database*** is a an organized collection of *reliable scientific* data of *known provenance*, stored in a machine-readable form, with automated search and processing capabilities.

- organized;
- machine-readable;
- reliable (known provenance);
- suitable for scientific inferences;

Numerous databases have been created

The principles discussed here are suitable for various applications:

- AMCSD [Rajan et al. (2006)] (mineral data)
- COD [Gražulis et al. (2009)], [Gražulis et al. (2012)] (experimental crystal data)
- TCOD [Merkys et al. (2017)] (DFT calculated data)
- PCOD [Le Bail(2005)] (predicted crystal structures)
- ROD [Mendili et al. (2019)] (Raman spectra)

Confidential personal and medical data

A word of caution...

NB! Some modern uses of data require certain legal actions and compliance with local laws.

Confidential personal and medical data

A word of caution...

NB! Some modern uses of data require certain legal actions and compliance with local laws.

- GDPR (in Europe) for the use of personal data

Confidential personal and medical data

A word of caution...

NB! Some modern uses of data require certain legal actions and compliance with local laws.

- GDPR (in Europe) for the use of personal data
- Use of medical information may require authorization and/or approval of an Ethics committee.

Confidential personal and medical data

A word of caution...

NB! Some modern uses of data require certain legal actions and compliance with local laws.

- GDPR (in Europe) for the use of personal data
- Use of medical information may require authorization and/or approval of an Ethics committee.

We will not discuss these topics since we will only used open data in this tutorial!

Building a scientific database

... is not as difficult as it may seem :)

But what if crystallographers work together to establish a public domain database with all relevant crystallographic data? This would not only overcome the current situation with 'fragmented' databases, it would also prevent for becoming dependent from monopolists.

What would be needed?

1. A small team of engaged scientists with some experience in database and software design to coordinate the project.

2. The authors (i.e. the scientific community = YOU) who provides the project with database entries (note, that if you have'nt sold your experimental results exclusively, you are free to distribute the data to such a database, even if they have already been part of a publication - and a lot of good data have never been published).

3. Free software a) for maintaining the database, b) for data evaluation and calculation of derived data (e.g. calculated powder pattern from crystal structures for search-match purposes), c) for browsing and retrieval.

gemstonede (Dr. Michael BERNDT) Fri Feb 14, 2003 1:26 pm

Contents of this talk (cont.)

1. Why do we need databases?
2. Principles of database construction
3. A specific example: let's construct a P1 cell database!

Stable identifiers

Identifiers for data records:

Stable identifiers

Identifiers for data records:

- UNIQUE

Stable identifiers

Identifiers for data records:

- UNIQUE
- NOT NULL

Stable identifiers

Identifiers for data records:

- UNIQUE
- NOT NULL
- stable

Stable identifiers

Identifiers for data records:

- UNIQUE
- NOT NULL
- stable
- Semantically neutral (no promised meaning!)

Stable identifiers

Identifiers for data records:

- UNIQUE
- NOT NULL
- stable
- Semantically neutral (no promised meaning!)

Modify old record?

```
formula: C6 H10 O6
unit cell: 10 10 10 90 90 90
```

```
formula: C6 H12 O6
unit cell: 10 10 10 90 102 90
```

Stable identifiers

Identifiers for data records:

- UNIQUE
- NOT NULL
- stable
- Semantically neutral (no promised meaning!)

Modify old record?

```
DB:23721
formula:  C6 H10 O6
unit cell:  10 10 10 90 90 90
```

```
DB:23721
formula:  C6 H12 O6
unit cell:  10 10 10 90 102 90
```

Stable identifiers

Identifiers for data records:

- UNIQUE
- NOT NULL
- stable
- Semantically neutral (no promised meaning!)

Add a new record?

```
DB:23721
formula:  C6 H10 O6
unit cell:  10 10 10 90 90 90
```

```
DB:24786
formula:  C6 H12 O6
unit cell:  10 10 10 90 102 90
```

Kinds of identifiers

1. Locally assigned (by your database):

Kinds of identifiers

1. Locally assigned (by your database):
   - Database primary keys (ids) (e.g. cod/2000000)

Kinds of identifiers

1. Locally assigned (by your database):
   - Database primary keys (ids) (e.g. cod/2000000)
   - Unique names

Kinds of identifiers

1. Locally assigned (by your database):
   - Database primary keys (ids) (e.g. cod/2000000)
   - Unique names
2. Globally assigned (by an “authority”):

Kinds of identifiers

1. Locally assigned (by your database):
   - Database primary keys (ids) (e.g. cod/2000000)
   - Unique names
2. Globally assigned (by an "authority"):
   - ARK (e.g. ark:/53355/cl010066723)

Kinds of identifiers

1. Locally assigned (by your database):
   - Database primary keys (ids) (e.g. cod/2000000)
   - Unique names
2. Globally assigned (by an "authority"):
   - ARK (e.g. ark:/53355/cl010066723)
   - DOI (e.g. doi:10.1107/s1600576715021871)

Kinds of identifiers

1. Locally assigned (by your database):
   - Database primary keys (ids) (e.g. cod/2000000)
   - Unique names
2. Globally assigned (by an "authority"):
   - ARK (e.g. ark:/53355/cl010066723)
   - DOI (e.g. doi:10.1107/s1600576715021871)
3. Distributed:
   - UUID (e.g. `ef3ecf0c-bbf1-11ec-a420-1fe8a65f4a22`)

Kinds of identifiers

1. Locally assigned (by your database):
   - Database primary keys (ids) (e.g. cod/2000000)
   - Unique names
2. Globally assigned (by an "authority"):
   - ARK (e.g. ark:/53355/cl010066723)
   - DOI (e.g. doi:10.1107/s1600576715021871)
3. Distributed:
   - UUID (e.g. `ef3ecf0c-bbf1-11ec-a420-1fe8a65f4a22`)
4. Distributed:

Kinds of identifiers

1. Locally assigned (by your database):
   - Database primary keys (ids) (e.g. cod/2000000)
   - Unique names
2. Globally assigned (by an "authority"):
   - ARK (e.g. ark:/53355/cl010066723)
   - DOI (e.g. doi:10.1107/s1600576715021871)
3. Distributed:
   - UUID (e.g. `ef3ecf0c-bbf1-11ec-a420-1fe8a65f4a22`)
4. Distributed:
   - Checksums (MD5, SHA1, SHA256...)

Kinds of identifiers

1. Locally assigned (by your database):
   - Database primary keys (ids) (e.g. cod/2000000)
   - Unique names
2. Globally assigned (by an "authority"):
   - ARK (e.g. ark:/53355/cl010066723)
   - DOI (e.g. doi:10.1107/s1600576715021871)
3. Distributed:
   - UUID (e.g. `ef3ecf0c-bbf1-11ec-a420-1fe8a65f4a22`)
4. Distributed:
   - Checksums (MD5, SHA1, SHA256...)
   - "Synthetic" (business) keys (a set of unique properties of your data record)

Version control systems ideally suitable for tracing file provenance!

Version control

CVS, Subversion Git

Version control systems ideally suitable for tracing file provenance!

- Subversion

Version control

CVS, Subversion Git

Version control systems ideally suitable for tracing file provenance!

- Subversion
- Git

Version control

CVS, Subversion Git

Version control systems ideally suitable for tracing file provenance!

- Subversion
- Git
- CVS

Database engines

Relational (or NoSQL) database as a fast search cache:

Database engines

Relational (or NoSQL) database as a fast search cache:

- MariaDB (MySQL)

Database engines

Relational (or NoSQL) database as a fast search cache:

- MariaDB (MySQL)
- PostgreSQL

Database engines

Relational (or NoSQL) database as a fast search cache:

- MariaDB (MySQL)
- PostgreSQL
- SQLite :)

Database engines

Relational (or NoSQL) database as a fast search cache:

- MariaDB (MySQL)
- PostgreSQL
- SQLite :)
- Mongo, Coach, ...

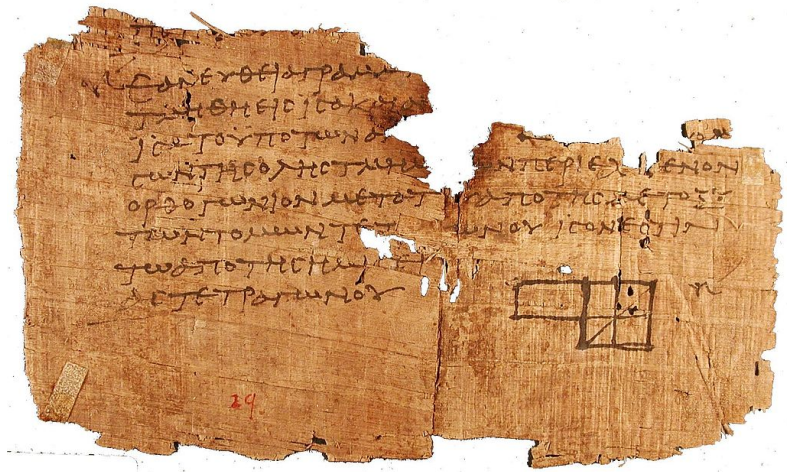
Data representation

Data representation

Most of the knowledge has reached us as *texts*...

Example:

Even though written about 2000 years ago, the Greek text with Euclid's "Elements" can still be read today:



Euclid, Public domain, via Wikimedia Commons

Data representation

- Text files are readable over *decades* on various hardware and software architectures

Data representation

- Text files are readable over *decades* on various hardware and software architectures
- Binary formats (especially proprietary ones) can become obsolete in under 10 years. (Try to read an MS Word 1 or MS Word 6 file today ...)

Data representation

- Text files are readable over *decades* on various hardware and software architectures
- Binary formats (especially proprietary ones) can become obsolete in under 10 years. (Try to read an MS Word 1 or MS Word 6 file today ...)
- Standard text (character) encodings are available: ASCII, Unicode.

Data representation

- Text files are readable over *decades* on various hardware and software architectures
- Binary formats (especially proprietary ones) can become obsolete in under 10 years. (Try to read an MS Word 1 or MS Word 6 file today ...)
- Standard text (character) encodings are available: ASCII, Unicode.
- For images and video, of course, you'll have to use binary – but make sure the format is well documented (endianness, number radix, floating point formats, data, color encoding, layout, etc.)

File formats

General formats

Let's pick a well-documented, possibly standardized file format.

File formats

General formats

Let's pick a well-documented, possibly standardized file format.
General structured data formats:

File formats

General formats

Let's pick a well-documented, possibly standardized file format. General structured data formats:

- STAR

File formats

General formats

Let's pick a well-documented, possibly standardized file format.
General structured data formats:

- STAR
- XML

File formats

General formats

Let's pick a well-documented, possibly standardized file format.
General structured data formats:

- STAR
- XML
- JSON

File formats

General formats

Let's pick a well-documented, possibly standardized file format.
General structured data formats:

- STAR
- XML
- JSON
- YAML

File formats

General formats

Let's pick a well-documented, possibly standardized file format.
General structured data formats:

- STAR
- XML
- JSON
- YAML
- HDF5, TIFF, PNG, JPEG, MPEG (binary...)

File formats

General formats

Let's pick a well-documented, possibly standardized file format. General structured data formats:

- STAR
- XML
- JSON
- YAML
- HDF5, TIFF, PNG, JPEG, MPEG (binary...)

Powerful and extensible; require complex parsers but parser libraries are available.

File formats

Specialized data formats

Specialized data formats for chemistry and crystallography:

Specialized data formats for chemistry and crystallography:

- CIF (based on STAR)

File formats

Specialized data formats

Specialized data formats for chemistry and crystallography:

- CIF (based on STAR)
- CML (based in XML)

File formats

Specialized data formats

Specialized data formats for chemistry and crystallography:

- CIF (based on STAR)
- CML (based in XML)
- Chemical JSON (based on JSON)

File formats

Specialized data formats

Specialized data formats for chemistry and crystallography:

- CIF (based on STAR)
- CML (based in XML)
- Chemical JSON (based on JSON)
- SDF, MOL

File formats

Specialized data formats

Specialized data formats for chemistry and crystallography:

- CIF (based on STAR)
- CML (based in XML)
- Chemical JSON (based on JSON)
- SDF, MOL

Same as generic structured formats + semantics descriptions (CIF dictionaries, XML schemata, CML dictionaries...)

File formats

Exchange formats

Generic data exchange formats:

File formats

Exchange formats

Generic data exchange formats:

- CSV, TSV

File formats

Exchange formats

Generic data exchange formats:

- CSV, TSV
- XYZ (for chemistry)

File formats

Exchange formats

Generic data exchange formats:

- CSV, TSV
- XYZ (for chemistry)
- SMILES, InChi

File formats

Exchange formats

Generic data exchange formats:

- CSV, TSV
- XYZ (for chemistry)
- SMILES, InChi

Very easy to parse, very simple to describe, but require additional documentation and are not easy to extend.

Data flow

What is your primary: DB or files?

Data flow

What is your primary: DB or files?

- Files $\rightarrow$ database (the COD way)

Data flow

What is your primary: DB or files?

- Files $\rightarrow$ database (the COD way)
- Database $\rightarrow$ files (we are experimenting with it...)

Data flow

What is your primary: DB or files?

- Files $\rightarrow$ database (the COD way)
- Database $\rightarrow$ files (we are experimenting with it...)
- Database $\leftrightarrow$ files (not recommended...)

Data flow

What is your primary: DB or files?

- Files $\rightarrow$ database (the COD way)
- Database $\rightarrow$ files (we are experimenting with it...)
- Database $\leftrightarrow$ files (not recommended...)

Data transfer from files in a repository to a database can be implemented as a **post-commit hook** (supported by Subversion, Git, CVS).

Data flow

What is your primary: DB or files?

- Files $\rightarrow$ database (the COD way)
- Database $\rightarrow$ files (we are experimenting with it...)
- Database $\leftrightarrow$ files (not recommended...)

Data transfer from files in a repository to a database can be implemented as a **post-commit hook** (supported by Subversion, Git, CVS).

A *post-commit hook* is a small script (program) that is run automatically by a version-control system server immediately after the commit was registered.

Quality controls

Syntax checks, semantic checks

As a minimum:

Quality controls

Syntax checks, semantic checks

As a minimum:

- Check the formal correctness of **syntax** of the incoming files

Quality controls

Syntax checks, semantic checks

As a minimum:

- Check the formal correctness of **syntax** of the incoming files
- Check the **required data items**

Quality controls

Syntax checks, semantic checks

As a minimum:

- Check the formal correctness of **syntax** of the incoming files
- Check the **required data items**

Syntax of the submitted files can be checked by a **pre-commit hook** (supported by Subversion, Git, CVS).

Quality controls

Syntax checks, semantic checks

As a minimum:

- Check the formal correctness of **syntax** of the incoming files
- Check the **required data items**

Syntax of the submitted files can be checked by a **pre-commit hook** (supported by Subversion, Git, CVS).

A *pre-commit hook* is a small script (program) that is run automatically by a version-control system server *before* the revision is committed. A non-zero exit status of the program aborts the commit.

Further developments

Things to think about

Further developments

Things to think about

1. Data curation policies

Further developments

Things to think about

1. Data curation policies
   - What is the *meaning* of your data?

Further developments

Things to think about

1. Data curation policies
   - What is the *meaning* of your data?
   - What does a single record represent?

Further developments

Things to think about

1. Data curation policies
   - What is the *meaning* of your data?
   - What does a single record represent?
   - How are data curated (under what circumstances are database records changed and how)?

Further developments

Things to think about

1. Data curation policies
   - What is the *meaning* of your data?
   - What does a single record represent?
   - How are data curated (under what circumstances are database records changed and how)?
2. Public deposition system?

Further developments

Things to think about

1. Data curation policies
   - What is the *meaning* of your data?
   - What does a single record represent?
   - How are data curated (under what circumstances are database records changed and how)?
2. Public deposition system?
3. Data privacy policies (accounts, GDPR, etc.)?

Further developments

Things to think about

1. Data curation policies
   - What is the *meaning* of your data?
   - What does a single record represent?
   - How are data curated (under what circumstances are database records changed and how)?
2. Public deposition system?
3. Data privacy policies (accounts, GDPR, etc.)?
4. Connections with other databases?

Further developments

Things to think about

1. Data curation policies
   - What is the *meaning* of your data?
   - What does a single record represent?
   - How are data curated (under what circumstances are database records changed and how)?
2. Public deposition system?
3. Data privacy policies (accounts, GDPR, etc.)?
4. Connections with other databases?
5. Longevity of your database?

Software quality issues

- **Do** use automated (unit) testing

Software quality issues

- **Do** use automated (unit) testing
- **Do** use version control

Software quality issues

- **Do** use automated (unit) testing
- **Do** use version control
- **Do** use released versions (tags); manage your software release cycle

Software quality issues

- **Do** use automated (unit) testing
- **Do** use version control
- **Do** use released versions (tags); manage your software release cycle
- Prefer semantic versioning where possible

Contents of this talk (cont.)

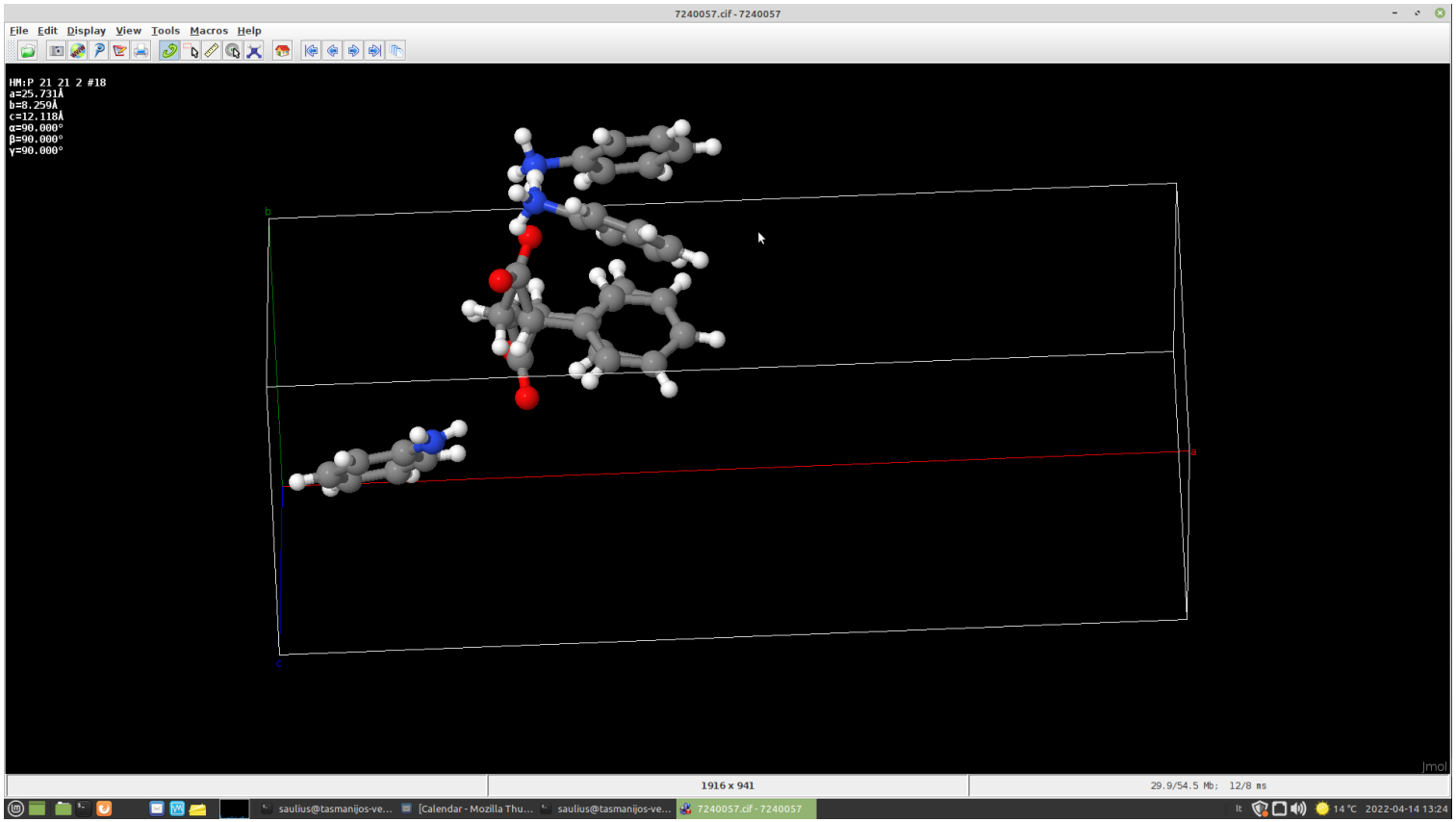
1. Why do we need databases?
2. Principles of database construction
3. A specific example: let's construct a P1 cell database!

Let's build a P1 cell database

Each record will contain all molecules in a crystal unit cell:

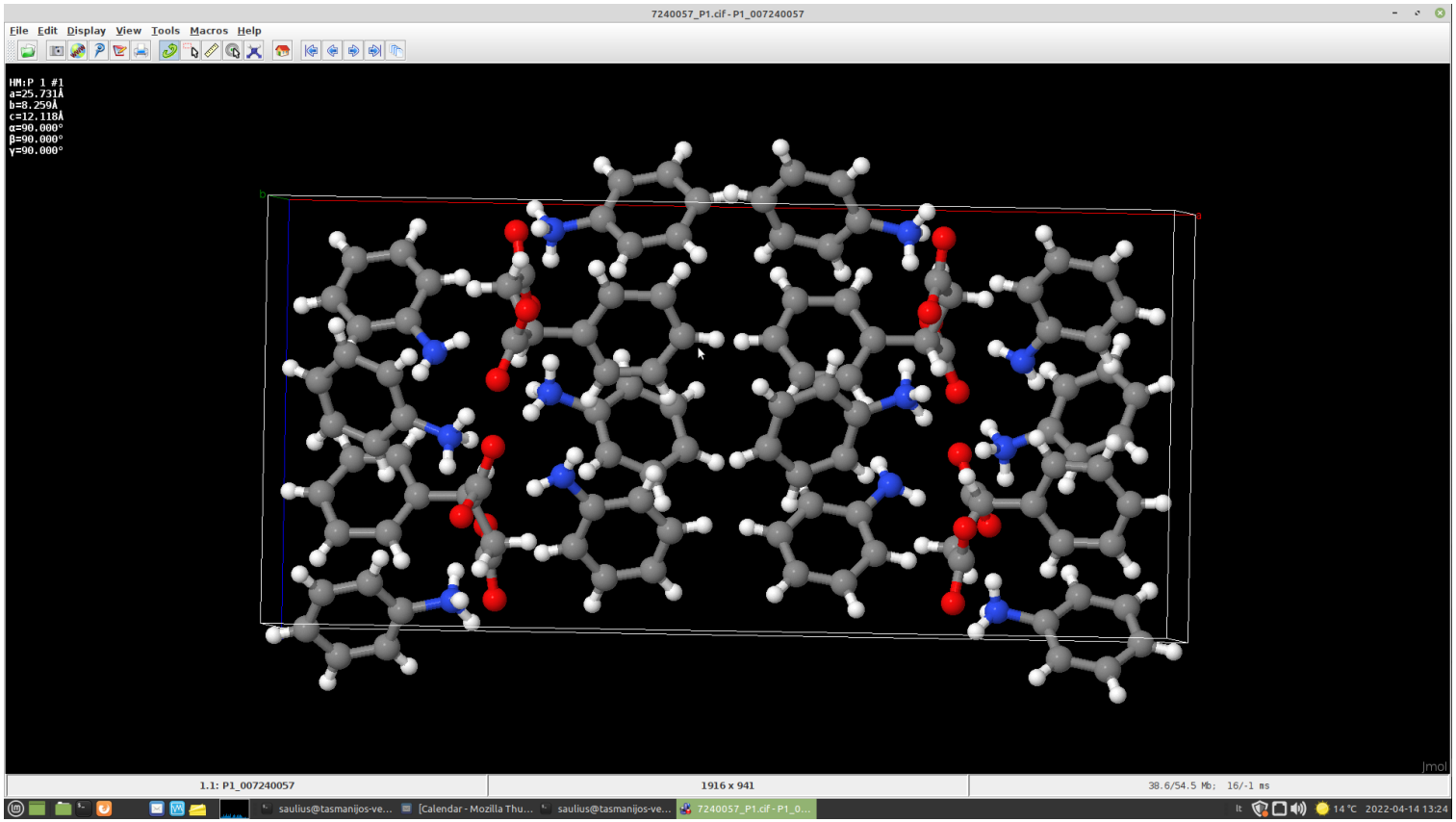
Let's build a P1 cell database

Each record will contain all molecules in a crystal unit cell:
Asymmetric unit:



Let's build a P1 cell database

Each record will contain all molecules in a crystal unit cell:
Full unit cell (aka "P1 cell"):



What this database is good for?

What scientific questions can we ask with this database?

Even if the entries are computed, we can view it as cache of computation results.

What this database is good for?

What scientific questions can we ask with this database?

- What enantiomers are there in the database?

Even if the entries are computed, we can view it as cache of computation results.

What this database is good for?

What scientific questions can we ask with this database?

- What enantiomers are there in the database?
- What contacts do molecules make in a crystal?

Even if the entries are computed, we can view it as cache of computation results.

What this database is good for?

What scientific questions can we ask with this database?

- What enantiomers are there in the database?
- What contacts do molecules make in a crystal?
- Generate a super-cell easily – only lattice translations need to be applied

Even if the entries are computed, we can view it as cache of computation results.

What this database is good for?

What scientific questions can we ask with this database?

- What enantiomers are there in the database?
- What contacts do molecules make in a crystal?
- Generate a super-cell easily – only lattice translations need to be applied
- What chemical species are there in a database?

Even if the entries are computed, we can view it as cache of computation results.

What this database is good for?

What scientific questions can we ask with this database?

- What enantiomers are there in the database?
- What contacts do molecules make in a crystal?
- Generate a super-cell easily – only lattice translations need to be applied
- What chemical species are there in a database?
- ... :)

Even if the entries are computed, we can view it as cache of computation results.

What this database is good for?

What scientific questions can we ask with this database?

- What enantiomers are there in the database?
- What contacts do molecules make in a crystal?
- Generate a super-cell easily – only lattice translations need to be applied
- What chemical species are there in a database?
- ... :)

Even if the entries are computed, we can view it as cache of computation results.
A word of caution: do *not* use these entries for crystallographic refinement!

Select implementations

- File format: CIF

Select implementations

- File format: CIF
- Database: SQLite (or MariaDB)

Select implementations

- File format: CIF
- Database: SQLite (or MariaDB)
- Version control: Subversion

Select implementations

- File format: CIF
- Database: SQLite (or MariaDB)
- Version control: Subversion
- Platform: Debian Linux

Select implementations

- File format: CIF
- Database: SQLite (or MariaDB)
- Version control: Subversion
- Platform: Debian Linux
- Web server: Apache2

Select implementations

- File format: CIF
- Database: SQLite (or MariaDB)
- Version control: Subversion
- Platform: Debian Linux
- Web server: Apache2
- Web display: CGI + “Web scriptlets”

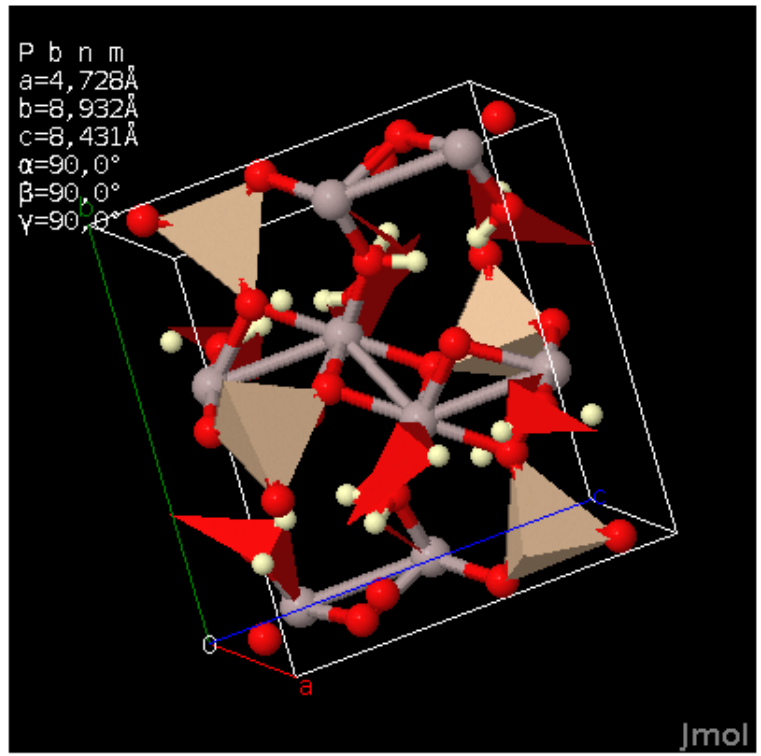
Acknowledgments

COD Advisory Board

Saulius Gražulis
Andrius Merkys
Daniel Chateigner
Robert T. Downs
Werner Kaminsky
Armel Le Bail
Luca Lutterotti
Peter Moeck
Peter Murray-Rust
Miguel Quirós

Thanks you!

http://en.wikipedia.org/wiki/Topaz



Coordinates	2207377.cif
Original IUCr paper	HTML

http://www.crystallography.net/2207377.html

References I

Gražulis S, Chateigner D, Downs RT, Yokochi AFT, Quirós M, Lutterotti L, et al. (2009) Crystallography Open Database – an open-access collection of crystal structures. Journal of Applied Crystallography 42:726–729, DOI 10.1107/S0021889809016690, URL `http://dx.doi.org/10.1107/S0021889809016690`

Gražulis S, Daškevič A, Merkys A, Chateigner D, Lutterotti L, Quirós M, et al. (2012) Crystallography Open Database (COD): an open-access collection of crystal structures and platform for world-wide collaboration. Nucleic Acids Research 40:D420–D427, DOI 10.1093/nar/gkr900, URL `http://nar.oxfordjournals.org/content/40/D1/D420.abstract`

Le Bail A (2005) Inorganic structure prediction with *grinsp*. Journal of Applied Crystallography 38:389–395, DOI 10.1107/S0021889805002384, URL `http://dx.doi.org/10.1107/S0021889805002384`

Mendili YE, Vaitkus A, Merkys A, Gražulis S, Chateigner D, Mathevet F, et al. (2019) Raman Open Database: first interconnected Raman–X-ray diffraction open-access resource for material identification. Journal of Applied Crystallography 52(3):618–625, DOI 10.1107/s1600576719004229

References II


Merkys A, Mounet N, Cepellotti A, Marzari N, Gražulis S, Pizzi G (2017) A posteriori metadata from automated provenance tracking: Integration of AiiDA and TCOD. Journal of Cheminformatics 9(1):56, DOI 10.1186/s13321-017-0242-y, URL `https://jcheminf.springeropen.com/articles/10.1186/s13321-017-0242-y`, `1706.08704v3`


Rajan H, Uchida H, Bryan D, Swaminathan R, Downs R, Hall-Wallace M (2006) Building the american mineralogist crystal structure database: A recipe for construction of a small internet database. In: Sinha A (ed) Geoinformatics: Data to Knowledge, Geological Society of America Special Papers, vol 397, Geological Society of America, Boulder, CO, United States, pp 73–80, DOI 10.1130/2006.2397(06)


Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, et al. (2016) The FAIR guiding principles for scientific data management and stewardship. Scientific Data 3(1), DOI 10.1038/sdata.2016.18, URL `https://doi.org/10.1038/sdata.2016.18`