UGM RDkit 2025:
ConfScale: An Open-Source Python Package for Scalable Conformer Generation and Filtration

Etienne Reboul
PharmD,PhD candidate
Supervisor : Dr Antoine Taly
Date: 11/09/2025

I- introduction

II- Data parsing

III- Data parallelism

IV- Filtration

I.a- ConfScale aims

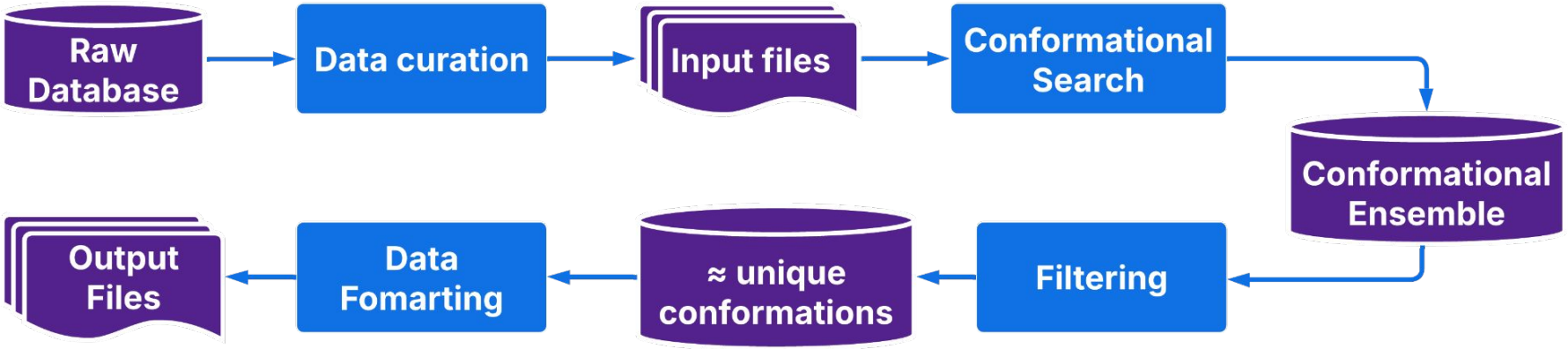
ConfScale is a python package (alpha) that aims at :

- Data formatting of molecular representations
- Data parallelism for conformers generation
- Conformers selections

ConfScale is a utility package to enable downstream Machine learning (ML) and Deep Learning task needing 3D data for *De novo* Drug Design

I.b- Pipeline Overview

ConfScale Workflow



I- introduction

II- Data curation

III- Data parallelism

IV- Filtration

II.a- Data curation problems

Working with very large library, you may encounter the following problems:

- Loosely define data file format
- Large uncompressed files with uneven distribution
- Little to no metadata and data type

II.a- Data curation problems

- Loosely define data file format

SMILES (.smi)

- Import and Export fully support the SMILES file format.

Background & Context

- MIME type: `chemical/x-daylight-smiles`
- SMILES chemical format.
- Commonly used to describe the structure of chemical molecules.
- SMILES is an acronym for Simplified Molecular Input Line Entry Specification.
- Used in cheminformatics applications and in chemistry databases to represent chemical formulas.
- ASCII format.
- Uses a linear notation to represent the connectivity graph of a molecule.
- Can store data for multiple molecules.
- Additional properties can be stored on the same line as the SMILES string.
- Developed in the 1980s by Arthur Weininger and David Weininger.

II.a- Data curation problems

- Large uncompressed files with uneven distribution

II.a- Data curation problems

- Little to no metadata and data type

Head of a file .smi file

```
smiles zinc_id
O=C(N[C@H](CO)C(=O)O)c1n[nH]c2ccccc12 19263719
CCC(=O)c1c(N)n(C)c(=O)n(C)c1=O 26421459
C[C@H](C(=O)NC1CC1)N1CCCNCC1 26513959
CC[C@@H](CO)NC(=O)[C@@H](NC(N)=O)C(C)C 35607305
Cn1cc(CNCC2CCC2)c(=O)n(C)c1=O 37179490
CN(C)S(=O)(=O)NCC[C@@H]1CCCNC1 37717456
CS(=O)(=O)CCNC(=O)CCNC1CC1 37740334
Cc1ccc(C(=O)NCCNCC(N)=O)cc1 37902923
O=C(Nc1ccon1)C(=O)N1CCOCC1 38725124
```

II.b- The parquet format

Apache Parquet is a free and open-source column-oriented format:

- (Fast) compression and encoding
- Load an arbitrary number of columns
- Explicitly defined data type and support metadata
- Interoperable with existing software

II.b- The parquet format

Apache Parquet is a free and open-source column-oriented format:

- (Fast) compression :
  - gzip
  - zstd
  - Snappy (default)
  - Lz4

- Efficient encoding of low cardinality data :
  - Dictionary encoding
  - Bytes packing
  - Delta encoding

II.b- The parquet format

Apache Parquet is a free and open-source column-oriented format:

- Each column is completely independent, and can be loaded without reading each row

Tails of dragon molecular descriptor list

4875	Depressant-50	Ghose-Viswanadhan-Wendoloski antidepressant-like index at 50%	Drug-like indices
4876	Psychotic-80	Ghose-Viswanadhan-Wendoloski antipsychotic-like index at 80%	Drug-like indices
4877	Psychotic-50	Ghose-Viswanadhan-Wendoloski antipsychotic-like index at 50%	Drug-like indices
4878	Hypertens-80	Ghose-Viswanadhan-Wendoloski antihypertensive-like index at 80%	Drug-like indices
4879	Hypertens-50	Ghose-Viswanadhan-Wendoloski antihypertensive-like index at 50%	Drug-like indices
4880	Hypnotic-80	Ghose-Viswanadhan-Wendoloski hypnotic-like index at 80%	Drug-like indices
4881	Hypnotic-50	Ghose-Viswanadhan-Wendoloski hypnotic-like index at 50%	Drug-like indices
4882	Neoplastic-80	Ghose-Viswanadhan-Wendoloski antineoplastic-like index at 80%	Drug-like indices
4883	Neoplastic-50	Ghose-Viswanadhan-Wendoloski antineoplastic-like index at 50%	Drug-like indices
4884	Infective-80	Ghose-Viswanadhan-Wendoloski antiinfective-like index at 80%	Drug-like indices
4885	Infective-50	Ghose-Viswanadhan-Wendoloski antiinfective-like index at 50%	Drug-like indices

Source : https://www.talete.mi.it/products/dragon_molecular_descriptor_list.pdf

II.b- The parquet format

Apache Parquet is a free and open-source column-oriented format:

- Explicitly defined data type and support metadata with schematics :

Field	Data Type	Description
`smiles`	String	Simplified Molecular Input Line Entry System (SMILES) notation
`zinc_id`	UInt32	Unique identifier for each molecule in the ZINC database
`tranches`	String	Data partitioning information based on logP and molecular weight

II.b- The parquet format

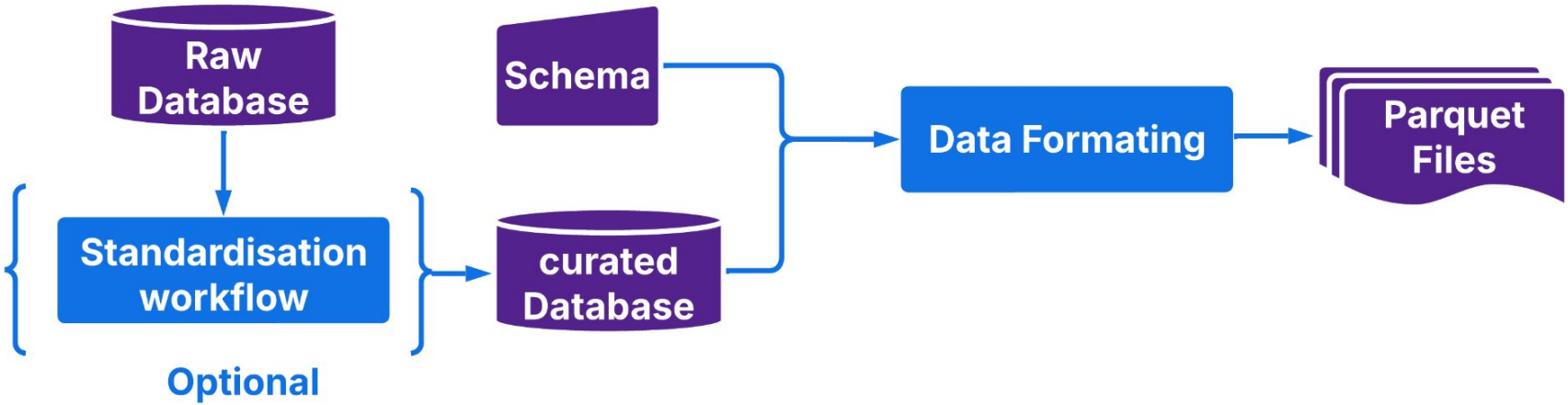
Apache Parquet is a free and open-source column-oriented format:

- Interoperable with existing software and python package like Pandas :

```
df= pd.read_parquet("zinc20_drug_like.parquet")
df.to_parquet("zinc20_drug_like.parquet")
```

/!\ Need to install dependencies : pyarrow or fastparquet

II.c- Data curation pipeline



II.d - Use case ZINC database

ZINC20 database drug-like subset is roughly **47,4 GB**

Dask Performance Report

Select different tabs on the top for additional information

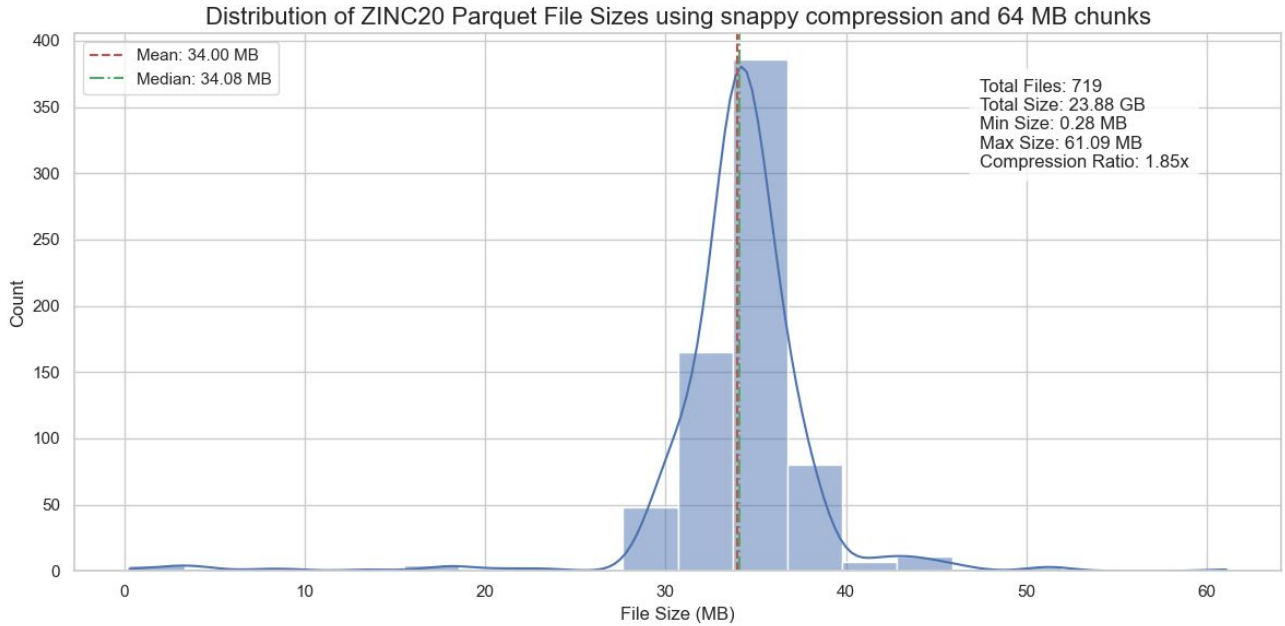
Duration: 140.71 s

Tasks Information

- number of tasks: 3596
- compute time: 73m 29s
- disk-read time: 123.62 s
- disk-write time: 1.69 s
- transfer time: 15.93 s

Scheduler Information

- Address: tcp://127.0.0.1:34061
- Workers: 18
- Threads: 36
- Memory: 25.15 GiB
- Dask Version: 2025.4.1
- Dask.Distributed Version: 2025.4.1



II.d - Use case : Freedom space

Freedom space 3.0 439 millions enumerated subset
original compressed file: **13,8 GB** vs compressed file **62,6 GB**

Dask Performance Report

Select different tabs on the top for additional information

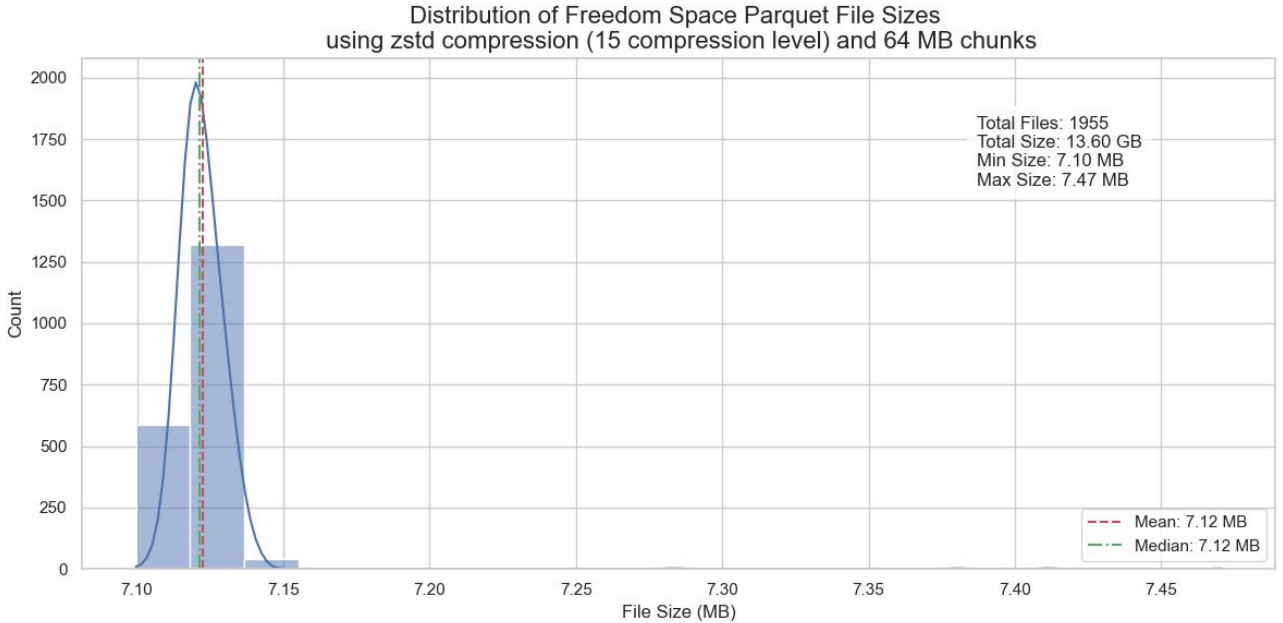
Duration: 20m 43s

Tasks Information

- number of tasks: 9776
- compute time: 12hr 18m
- disk-read time: 17.99 s
- disk-write time: 885.93 ms
- transfer time: 112.68 s

Scheduler Information

- Address: tcp://127.0.0.1:37047
- Workers: 18
- Threads: 36
- Memory: 25.15 GiB
- Dask Version: 2025.4.1
- Dask.Distributed Version: 2025.4.1



II.e- Take Home Message

Parquet format is good at :

- Data type and Metadata
- Interoperability
- Storing Data as independent column

Parquet format is okay-ish at :

- Compressing High cardinality data
- Readability

II.e- Perspectives

Apache parquet support nested data structure :

- One hot encoding of molecular string representation
- Saving binary array like fingerprints
- store scipy sparse matrix

I- introduction

II- Data curation

III- Data parallelism

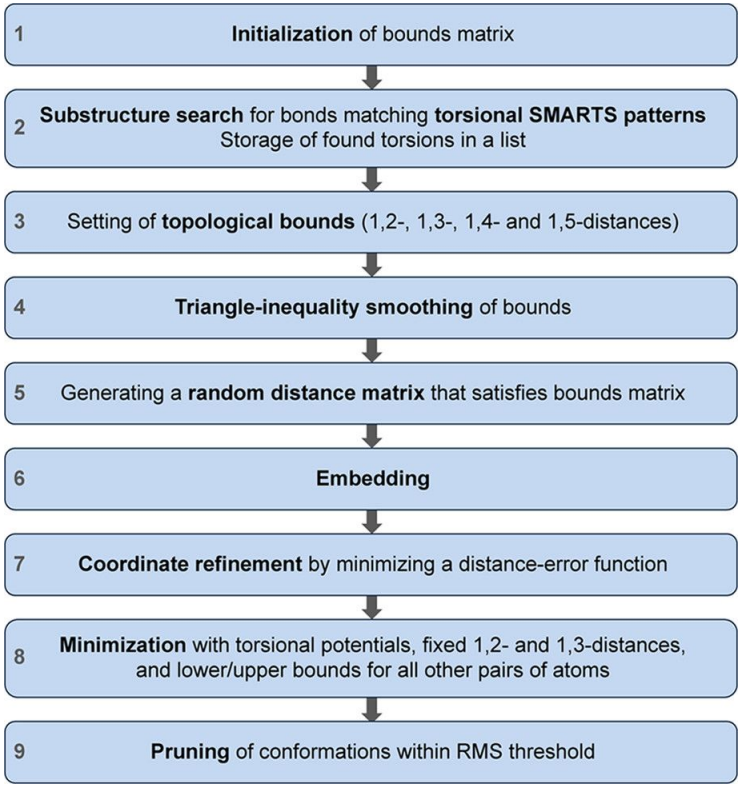
IV- Filtration

III.a - Conformer generation with ETKDGv3

Using RDKit ETKDG algorithm to generate conformers with bias toward crystal-like structures.

Is decently fast with :

- multithreading
- Optimize forcetol = 0.0135



Wang, S.; Witek, J.; Landrum, G. A.; Riniker, S. Improving Conformer Generation for Small Rings and Macrocycles Based on Distance Geometry and Experimental Torsional-Angle Preferences. *J. Chem. Inf. Model.* **2020**, *60* (4), 2044–2058. https://doi.org/10.1021/acs.jcim.0c00025

III.b - Data Parallelism with Dask

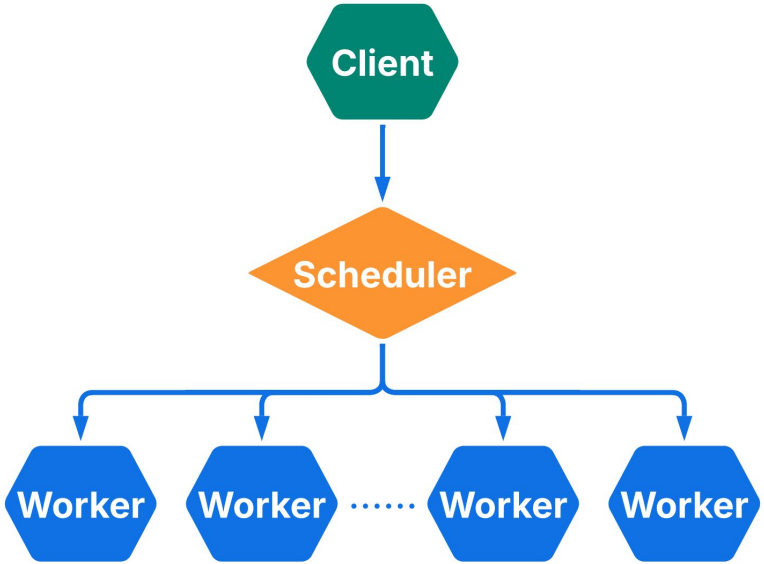
Overview

- Open-source **parallel computing library** for Python powered by joblib (like scikit-learn)
- ~~(Easy)~~ integration with existing NumPy, pandas codes
- Supposedly Scales from **desktop computer => clusters :**
  - Interface with HPC cluster via Dask-Jobqueue
  - Support commercial architecture with Kubernetes and Dask Gateway

III.b - Data Parallelism with Dask

Worker

- A **Dask Worker** is a **process** that executes tasks given by the Dask Scheduler
- Runs computations on **data partitions** (chunks of your dataset)
- Worker can be configure to have n **threads**



III.b - Parallelism : Process vs Thread

Process

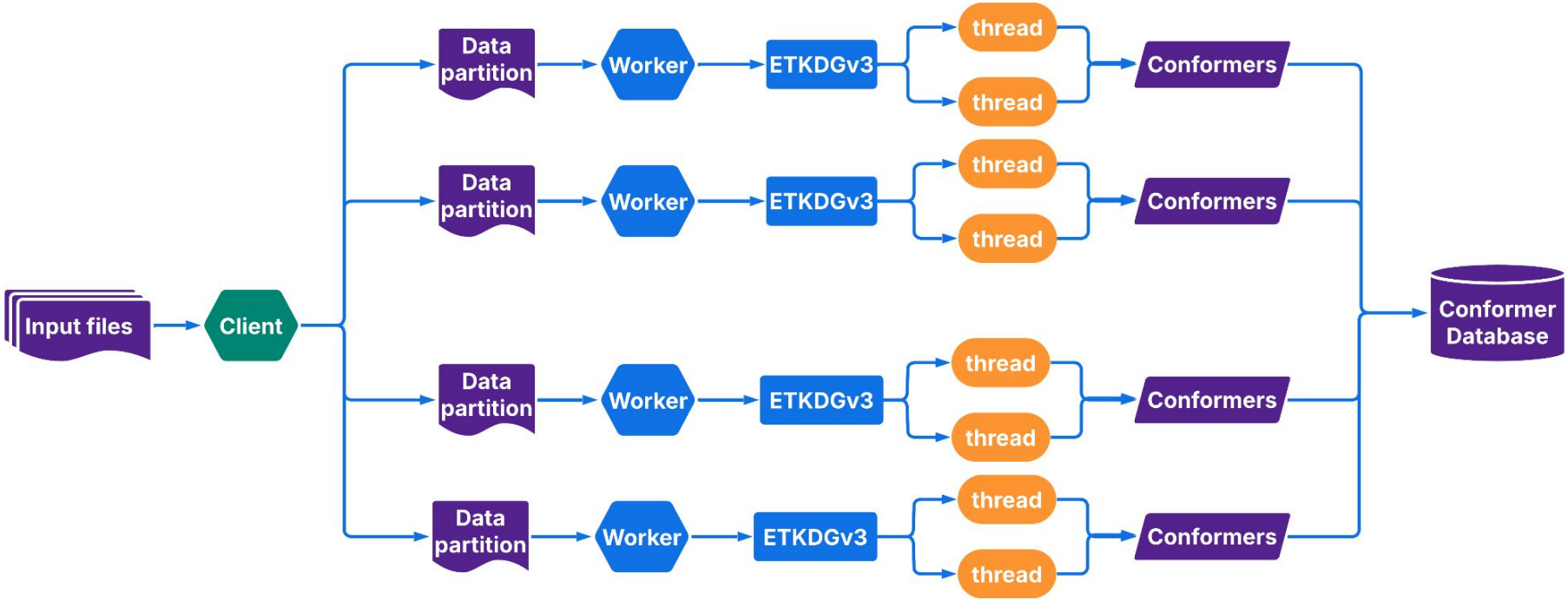
- Independent execution unit
- Has its **own memory space** and **code**
- Require time to set up
- Crashes should **not affect other processes**

Thread

- Lightweight execution unit typically **within a process**
- **Shares memory** and **code** with other threads of the same process
- Faster to create
- A crash can affect the **entire process**

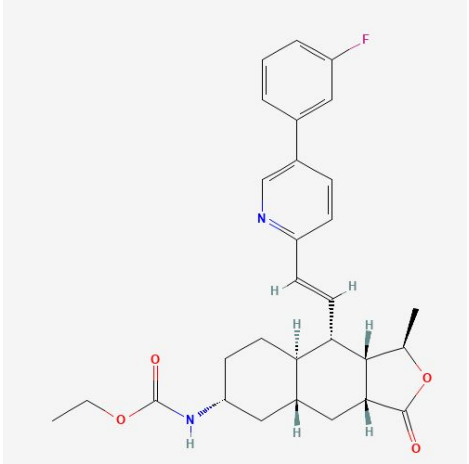
Process are **safer** but more **resource intensive** (python) , thread are **cheaper** but more **risky** (C++)

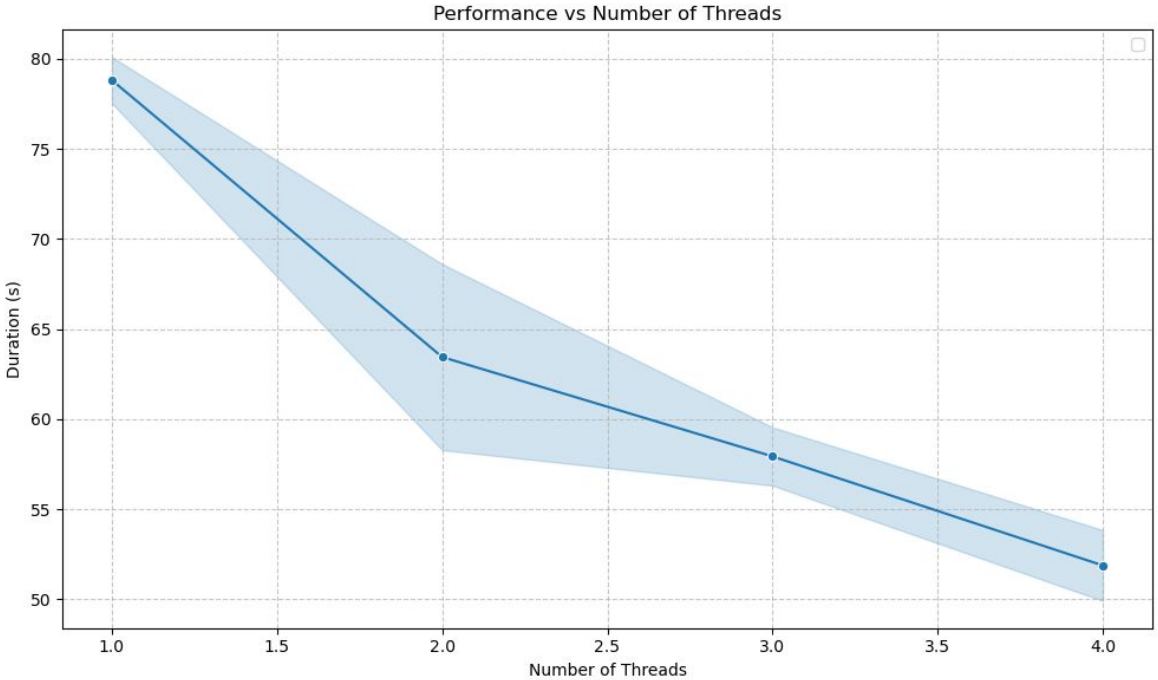
III.b - Combining Multiprocessing and Multithreading



III.c - test case vorapaxar

Experience set up: We setup **5** workers with a dummy set of **5** vorapaxar mol from which generate 1000 conformers. We change the number of thread going from 1 to 4 per worker





III.d- Take Home Message

Dask is good at and with :

- At Horizontal scaling (add lot small workers)
- Data that is well partitioned with well defined type

Dask is difficult with :

- Vertical scaling (big worker)
- Task that require inter-worker communication
- Pre-existing code needs to be carefully adapted

III.e- Perspectives

Dask Dataframe should enable the following fixture:

- GPU accelerated computing with cuda dask worker
- Full integration with existing RDKit PandasTools => distributed pattern search
- Works with python binding of software like Vina

I- introduction

II- Data curation

III- Data parallelism

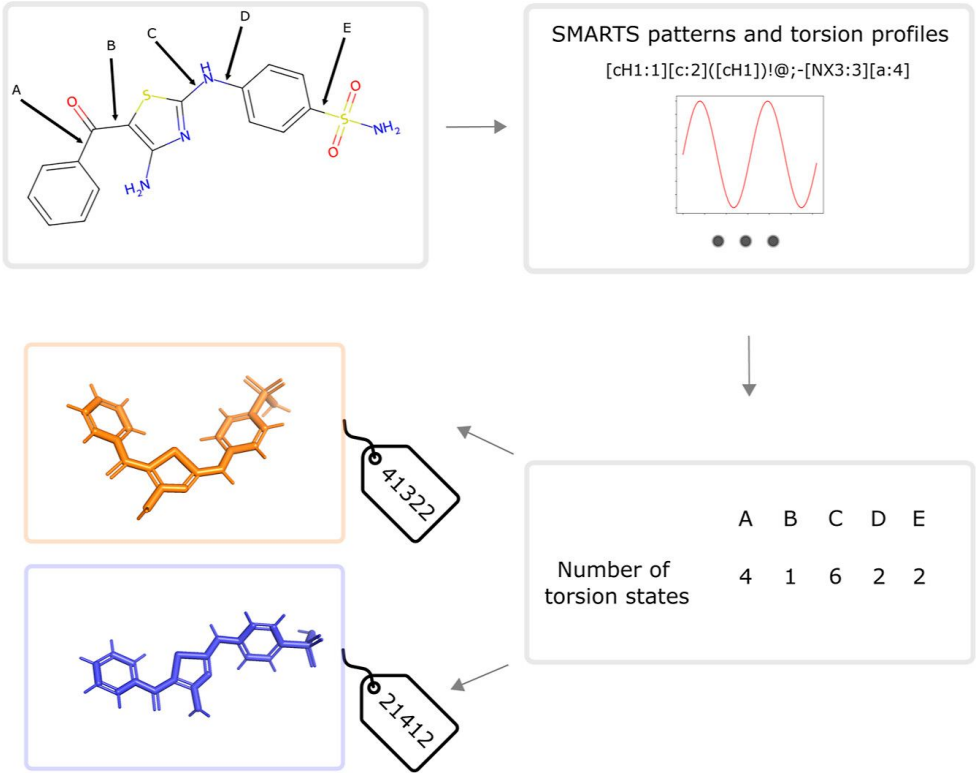
IV- Filtration

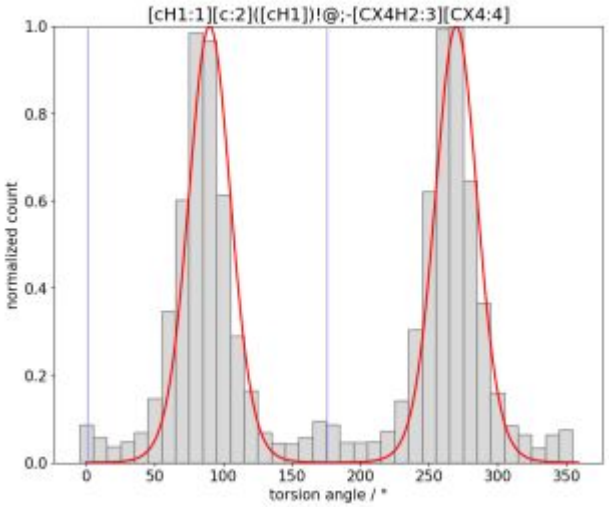
IV.a- The RMSD Filtration problem

RMSD suffers from the following drawbacks:

- What is a good reference conformation for alignment
- Pairwise RMSD is resource intensive
- What is an acceptable threshold to consider two conformations different ?

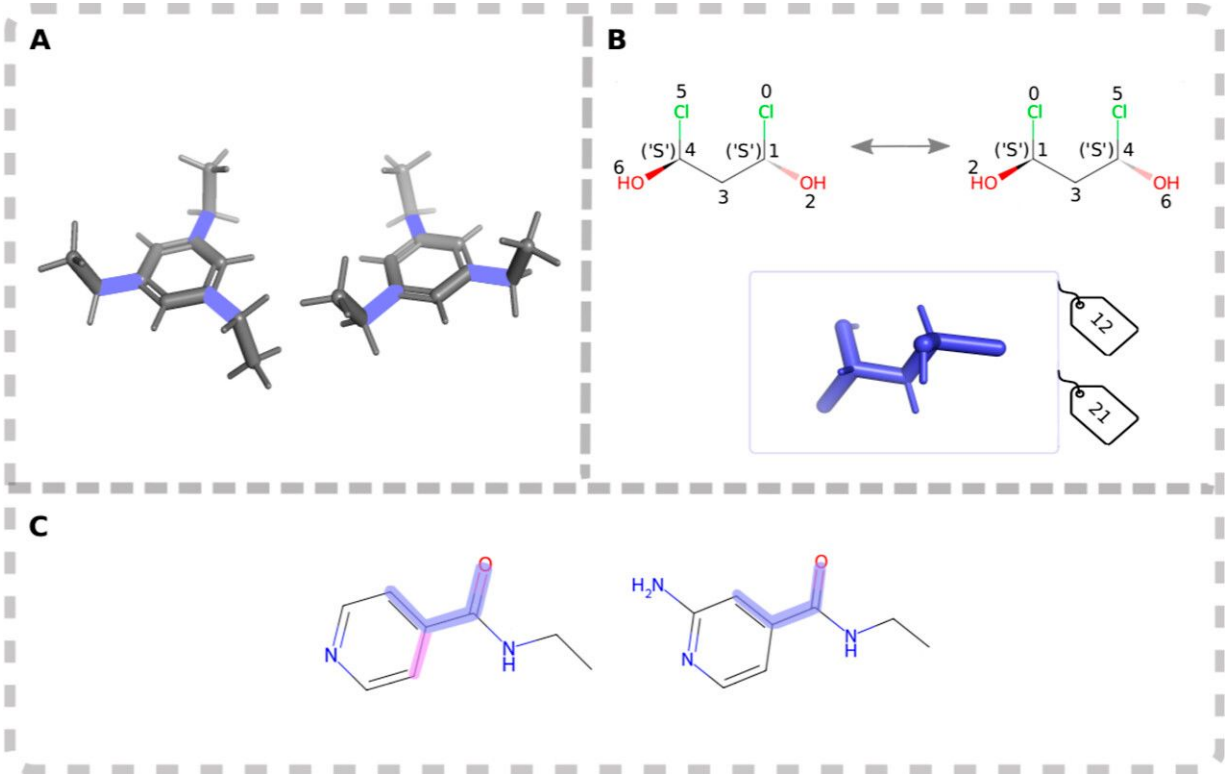
IV.b- Torsion Angular Bin String (TABS)





Braun, J.; Katzberger, P.; Landrum, G. A.; Riniker, S. Understanding and Quantifying Molecular Flexibility: Torsion Angular Bin Strings. *J. Chem. Inf. Model.* **2024**, *64* (20), 7917–7924. https://doi.org/10.1021/acs.jcim.4c01513

IV.b- Global and local symmetry Problems



Braun, J.; Katzberger, P.; Landrum, G. A.; Riniker, S. Understanding and Quantifying Molecular Flexibility: Torsion Angular Bin Strings. *J. Chem. Inf. Model.* **2024**, *64* (20), 7917–7924. https://doi.org/10.1021/acs.jcim.4c01513

IV.b- Proposed alternative for symmetry

TABS maps the molecule using SMARTS querying. If number of mapping >1 => global symmetry

<u>Alternative</u> :

Computing atom canonical ranking without breaking ties :

- If set of canonical indices < nb atoms => global symmetry

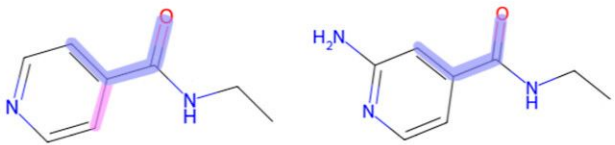
- Local symmetry yield identical canonical indices => aggregation and deduplication

IV.c- Conformational Fingerprint

Categorical labels derived from tabs are transformed into (sim)count fingerprint :

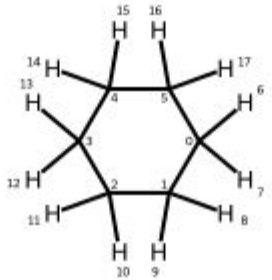
- All torsion angles with same index are aggregated
- Torsion angle with same middle bond are discarded

C



Braun, J.; Katzberger, P.; Landrum, G. A.; Riniker, S. Understanding and Quantifying Molecular Flexibility: Torsion Angular Bin Strings. *J. Chem. Inf. Model.* **2024**, *64* (20), 7917–7924. https://doi.org/10.1021/acs.jcim.4c01513

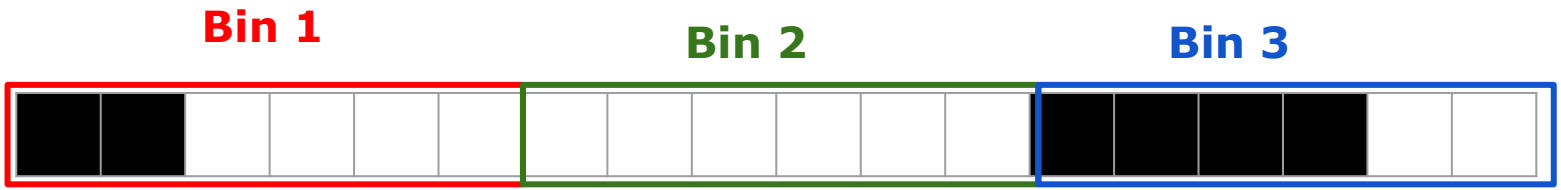
IV.b- Confp: cyclohexane



Cyclohexane analysis with ETKDGv3:

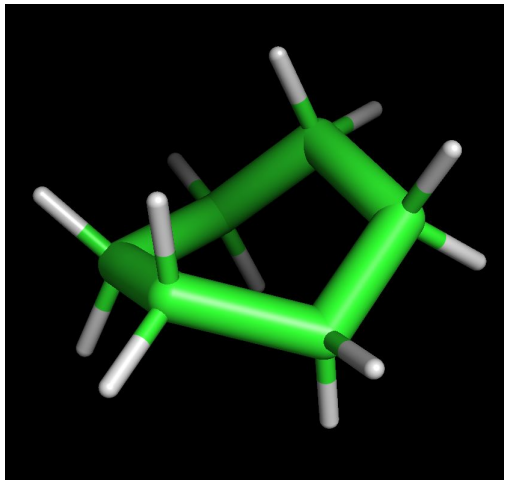
- Torsion smarts :
  [!#1;r{5-8}:1]@[CX4;r{5-8}:2]@;-[CX4;r{5-8}:3]@[!#1;r{5-8}:4]
- Number of angular Bin derived from literature : 3 [0, 120, 240]
- Number of equivalent torsion angle using canonical atom ranking : 6

Confp

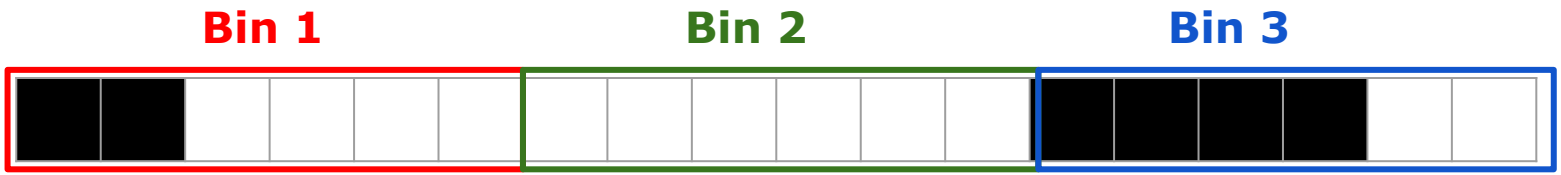


Braun, J.; Katzberger, P.; Landrum, G. A.; Riniker, S. Understanding and Quantifying Molecular Flexibility: Torsion Angular Bin Strings. *J. Chem. Inf. Model.* **2024**, *64* (20), 7917–7924. https://doi.org/10.1021/acs.jcim.4c01513

IV.b- Confp: cyclohexane



Confp



Braun, J.; Katzberger, P.; Landrum, G. A.; Riniker, S. Understanding and Quantifying Molecular Flexibility: Torsion Angular Bin Strings. *J. Chem. Inf. Model.* **2024**, *64* (20), 7917–7924. https://doi.org/10.1021/acs.jcim.4c01513

III.d- Take Home Message

RMSD is not a very good solution because :

- Need to define reference pose
- Pairwise comparison are expensive
- What defines a good threshold to differentiate two conformers

TABS and confp are :

- Doesn’t require alignment
- Can take into account global symmetry (TABS and confp)
- Can take into account some local symmetry (confp)

Conclusion

COnfScale provides some utilities to :

- Format data file to prepare for data parallelism
- Combine multithreading and multiprocessing to accelerate conformer generation
- Provide filtration of conformer that differs from classical RMSD

Perspectives

COnfScale should also provide in the future :

- Support to 3D/2D structural datafile like sdf
- Integration with chemoinformatics projects such as:
  - Scikit-fingerprint
  - EasyDock

Thank You