National Czech Programme eosc

Mass Spectra Prediction and Analysis: Machine Learning and Quantum Computing Perspectives

Aleš Křenek[1], Adam Hájek[1], Michael Wagner[2]

[1]Institute of Computer Science, [2]IBM Czech Republic

ISGC 20205

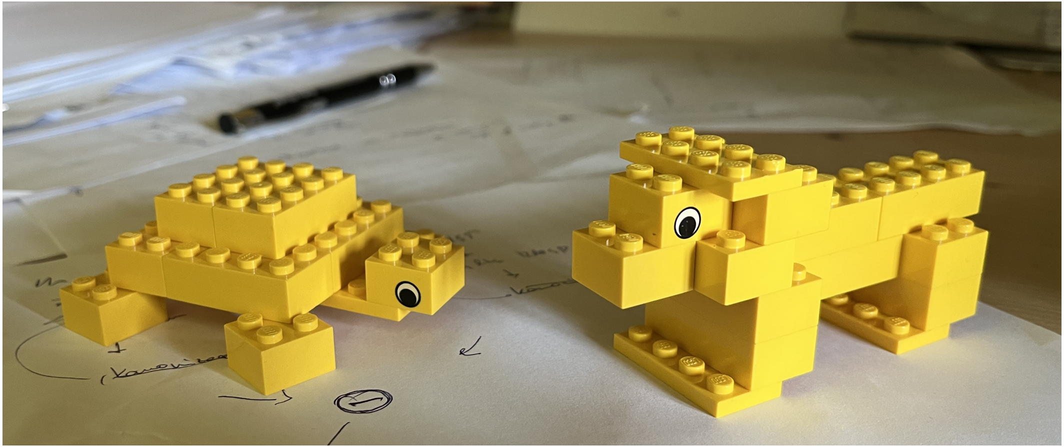
Co-funded by the European Union

MINISTRY OF EDUCATION, YOUTH AND SPORTS

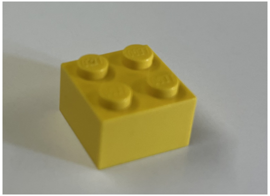
Overview

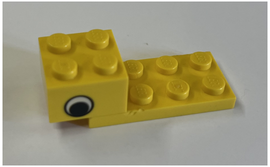
1. Mass spectrometry brief insights
2. Spectra simulation with quantum computers (targeted analysis)
3. Unknown spectra annotation with LLM-like models (untargeted analysis)

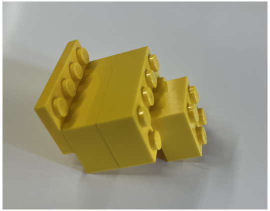
Mass spectrometry LEGO game



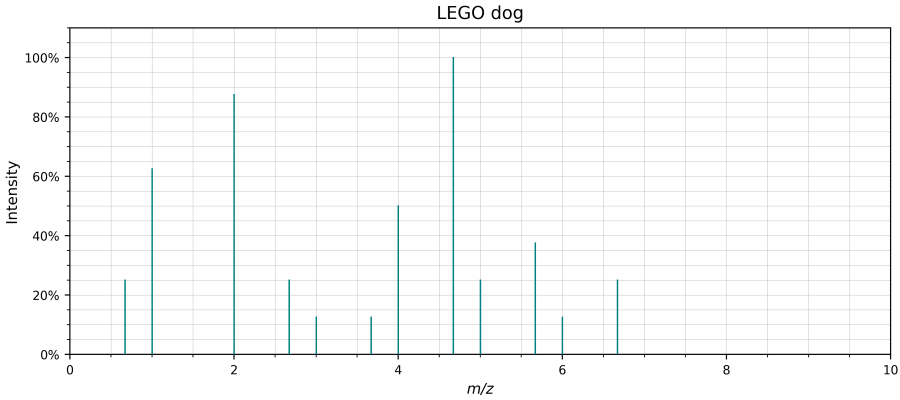
Mass spectrometry LEGO game





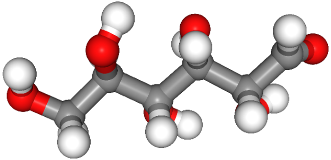


Mass spectrometry LEGO game



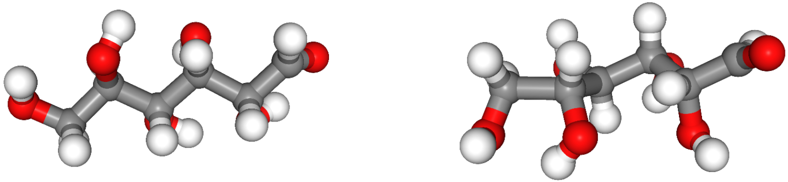
Real example

- Glucose molecule $C_6H_{12}O_6$



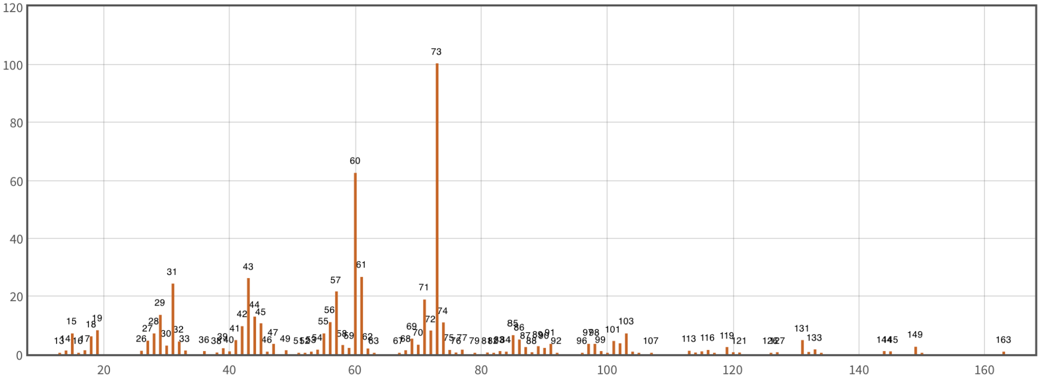
Real example

- Glucose molecule $C_6H_{12}O_6$
- Hit by electron beam of 70 eV (energy between UV and X-ray)



`https://www.youtube.com/embed/tEk_asS54Xg?autoplay=1&loop=1&playlist=tEk_asS54Xg`

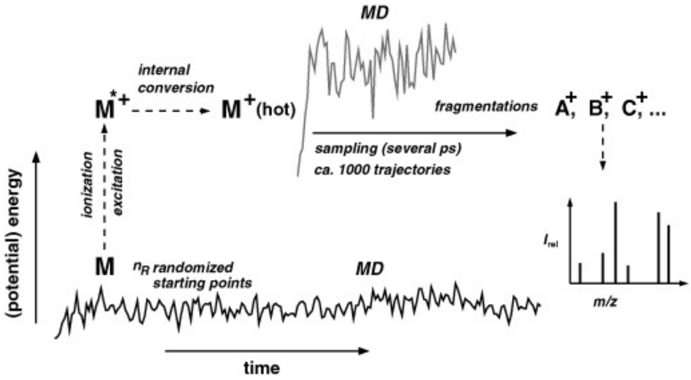
Real example



Targeted analysis

- There are “suspect(s)” to be present in the sample
- Their mass spectra are not known
- Accurate simulation is required

Spectra simulation



QCxMS (https://doi.org/10.1002/anie.201300158)

Ab-initio molecular dynamics

- Newtonian physics in a loop

$$x_{i+1} = x_i + v_i\,dt \qquad F = \partial E/\partial x \qquad v_{i+1} = v_i + F/m\,dt$$

Ab-initio molecular dynamics

- Newtonian physics in a loop

$$x_{i+1} = x_i + v_i\,dt \qquad F = \partial E/\partial x \qquad v_{i+1} = v_i + F/m\,dt$$

- Potential energy (accurate = quantum chemical)

$$\hat{H}\,\Psi = E\,\Psi$$

Ab-initio molecular dynamics

- Newtonian physics in a loop

$$x_{i+1} = x_i + v_i\, dt \qquad F = \partial E / \partial x \qquad v_{i+1} = v_i + F/m\, dt$$

- Potential energy (accurate = quantum chemical)

$$\hat{H}\,\Psi = E\,\Psi$$

- Only the smallest eigenvalue E_0 (ground state energy) is required
- Write down $\hat{H}$ and just solve the equation …

O(N!) complexity of classical approaches

- Pauli exclusion for fermions:

$$\Psi(r_1, r_2) = -\Psi(r_2, r_1)$$

  yields more complex combinations of per-particle wave functions:

$$\Psi(r_1, r_2) = \frac{1}{\sqrt{2}} \begin{vmatrix} \chi_1(r_1) & \chi_2(r_1) \\ \chi_1(r_2) & \chi_2(r_2) \end{vmatrix}$$

- In general, $N!$ terms for N particles
- 96 electrons in glucose, $96! \doteq 10^{150}$

Map to quantum hardware

- Capture all “quantum magic” in qubits, keeping $O(N)$ size

Map to quantum hardware

- Capture all “quantum magic” in qubits, keeping $O(N)$ size
- Occupancy number Fock space basis $|n_1, n_2, \ldots, n_M\rangle$ (where $n_i = 0$ or 1)
- Mapping to qubits
  - straightforward Jordan-Wigner
  - more hardware friendly Bravyi-Kitaev

Map to quantum hardware

- Capture all “quantum magic” in qubits, keeping $O(N)$ size
- Occupancy number Fock space basis $|n_1, n_2, \ldots, n_M\rangle$ (where $n_i = 0$ or 1)
- Mapping to qubits
  - straightforward Jordan-Wigner
  - more hardware friendly Bravyi-Kitaev
- Creation $\hat{a}^\dagger$ and annihilation $\hat{a}$ operators map to QC gates
- Hamiltonian in the 2nd quantization form

$$\hat{H} = \sum_{ij} h_{ij} \hat{a}_i^\dagger \hat{a}_j + \frac{1}{2} \sum_{ijkl} h_{ijkl} \hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_l \hat{a}_k$$

where h_{ij} and h_{ijkl} can be computed classically

Variational Quantum Eigensolver

- Choose initial parameters θ (classical)
- Prepare trial wavefunction $|\psi(\theta)\rangle$ (quantum)
- Evaluate $E(\theta) = \langle\Psi(\theta)|H|\Psi(\theta)\rangle$ (quantum)
- Change θ slightly and repeat (classical)

- Arriving at $E_0 \approx E(\theta^*)$ eventually

Are we there yet?

- *“It’s far far away, Donkey!”* (Shrek, 2001)

Are we there yet?

- *"It's far far away, Donkey!"* (Shrek, 2001)

- Qiskit code to compute energies with VQE ✓
  - offloads the $O(N!)$ problem to quantum hardware

Are we there yet?

- *"It's far far away, Donkey!"* (Shrek, 2001)
- Qiskit code to compute energies with VQE ✓
  - offloads the $O(N!)$ problem to quantum hardware
- QCxMS calls our Qiskit code ✓
  - Fortran → Python . . .

Are we there yet?

- *“It’s far far away, Donkey!”* (Shrek, 2001)
- Qiskit code to compute energies with VQE ✓
  - offloads the $O(N!)$ problem to quantum hardware
- QCxMS calls our Qiskit code ✓
  - Fortran $\rightarrow$ Python …
- Usable guinea pig – boron hydride (BH_3) ✓
  - the smallest molecule working with QCxMS
  - the biggest molecule for our Qiskit code (on simulator)

Are we there yet?

- *"It's far far away, Donkey!"* (Shrek, 2001)
- Qiskit code to compute energies with VQE ✓
  - offloads the $O(N!)$ problem to quantum hardware
- QCxMS calls our Qiskit code ✓
  - Fortran → Python . . .
- Usable guinea pig – boron hydride (BH_3) ✓
  - the smallest molecule working with QCxMS
  - the biggest molecule for our Qiskit code (on simulator)
- Issues with $\partial E/\partial x$ gradients ??

Are we there yet?

- *“It’s far far away, Donkey!”* (Shrek, 2001)
- Qiskit code to compute energies with VQE ✓
  - offloads the $O(N!)$ problem to quantum hardware
- QCxMS calls our Qiskit code ✓
  - Fortran → Python . . .
- Usable guinea pig – boron hydride (BH_3) ✓
  - the smallest molecule working with QCxMS
  - the biggest molecule for our Qiskit code (on simulator)
- Issues with $\partial E/\partial x$ gradients ??
- Still on simulator, reaching quantum hardware is the next step !!

Untargeted analysis

- Acquired data from real-world sample
- No particular idea what the chemicals can be
- Transform the set of spectra to formulae

Database search

- Traditional approach
- Spectral databases built over decades
  - 12k in 1970, . . . , 900k in 2023

Database search

- Traditional approach
- Spectral databases built over decades
  - 12k in 1970, . . . , 900k in 2023
- 10^9 known “small” molecules, 10^{60} possibly existing

Database search

- Traditional approach
- Spectral databases built over decades
  - 12k in 1970, . . . , 900k in 2023
- 10^9 known “small” molecules, 10^{60} possibly existing
- Extended methods to retrieve “something similar”

Machine language translation analogy

- (27.0, 103.91), (28.0, 84.92), (29.0, 120.89), (30.0, 79.93), (31.0, 745.33), (32.0, 86.92), (41.0, 149.86), (42.0, 162.85), (43.0, 572.48), (44.0, 275.75), (45.0, 190.83), (49.0, 75.93), (55.0, 180.84), (56.0, 188.83), (57.0, 945.15), (58.0, 116.89), (60.0, 527.52), (61.0, 366.67), (68.0, 56.95), (69.0, 146.87), (71.0, 226.8), (72.0, 121.89), (73.0, 999.0), (74.0, 140.87), (77.0, 158.86), (85.0, 156.86), (86.0, 211.81), (97.0, 50.95), (101.0, 51.95), (102.0, 61.94), (103.0, 106.9), (113.0, 34.97), (115.0, 27.97), (119.0, 19.98), (126.0, 19.98), (127.0, 21.98), (131.0, 43.96), (132.0, 34.97), (133.0, 35.97), (144.0, 12.99), (145.0, 18.98), (149.0, 23.98), (163.0, 9.99)
- O1[C@H](CO)[C@@H](O)[C@H](O)[C@@H](O)[C@H]1O[C@@]2(O[C@@H]([C@@H](O) [C@@H]2O)CO)CO

Machine language translation analogy

- (27.0, 103.91), (28.0, 84.92), (29.0, 120.89), (30.0, 79.93), (31.0, 745.33), (32.0, 86.92), (41.0, 149.86), (42.0, 162.85), (43.0, 572.48), (44.0, 275.75), (45.0, 190.83), (49.0, 75.93), (55.0, 180.84), (56.0, 188.83), (57.0, 945.15), (58.0, 116.89), (60.0, 527.52), (61.0, 366.67), (68.0, 56.95), (69.0, 146.87), (71.0, 226.8), (72.0, 121.89), (73.0, 999.0), (74.0, 140.87), (77.0, 158.86), (85.0, 156.86), (86.0, 211.81), (97.0, 50.95), (101.0, 51.95), (102.0, 61.94), (103.0, 106.9), (113.0, 34.97), (115.0, 27.97), (119.0, 19.98), (126.0, 19.98), (127.0, 21.98), (131.0, 43.96), (132.0, 34.97), (133.0, 35.97), (144.0, 12.99), (145.0, 18.98), (149.0, 23.98), (163.0, 9.99)
- O1[C@H](CO)[C@@H](O)[C@H](O)[C@@H](O)[C@H]1O[C@@]2(O[C@@H]([C@@H](O) [C@@H]2O)CO)CO
- Good language models can generalize to not-seen-before “sentences”

Naïve approach

- Pick a suitable language model
  - BART encoder-decoder transformer, 354M trainable parameters
- Elaborate on spectra and SMILES tokenization
- Further minor technicalities
  - e.g. use the orthogonal “position” input channel for intensities

Naïve approach

- Pick a suitable language model
  - BART encoder-decoder transformer, 354M trainable parameters
- Elaborate on spectra and SMILES tokenization
- Further minor technicalities
  - e.g. use the orthogonal “position” input channel for intensities
- Accuracy 28/51%, similarity 0.56/0.72 on 1 and 10 candidates
  - cf. extended database search 0% accuracy, 0.45/0.57 similarity

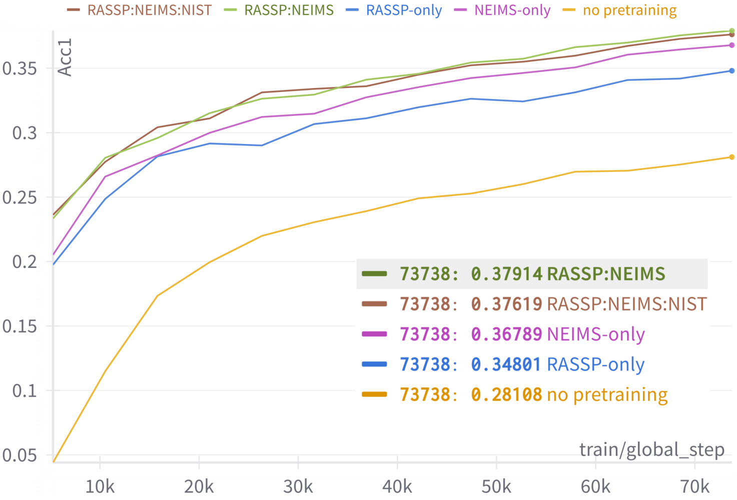
Naïve approach

- Pick a suitable language model
  - BART encoder-decoder transformer, 354M trainable parameters
- Elaborate on spectra and SMILES tokenization
- Further minor technicalities
  - e.g. use the orthogonal “position” input channel for intensities
- Accuracy 28/51%, similarity 0.56/0.72 on 1 and 10 candidates
  - cf. extended database search 0% accuracy, 0.45/0.57 similarity
- Not so bad but starting to overfit
  - cf. 354M parameters vs. 225k training spectra à 100 peaks

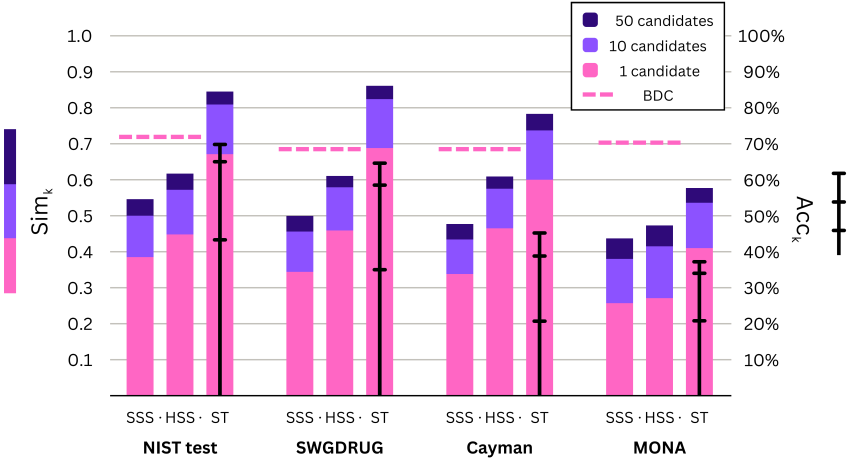
Pretraining with synthetic data

- Large-scale experimental data are not available
- Formula $\rightarrow$ spectrum “translation” is easier
  - several models are available, we pick NEIMS and RASSP
  - use the same training set to avoid information leak
- Harvested 30M random “standard-annotated-druglike” formulae from ZINC
- Filter to 9.5M according to RASSP restrictions
- Generate 2 $\times$ 9.5 spectra to pretrain the main model

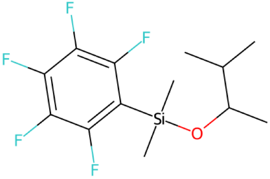
Experiment: Pretraining dataset mixing

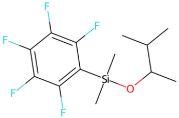


Main results

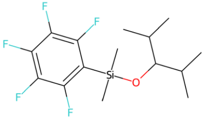


Prediction example

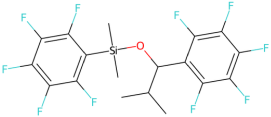




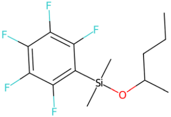
1.0



0.72



0.62



0.55

Availability

- Preprint `https://doi.org/10.48550/arXiv.2502.05114`
- Blog `https://blog.cerit.io/blog/spectus/`
- Demo Binder notebook `https://github.com/ljocha/spectus-demo`

Summary

- Mass spectra predicton/analysis seen from two perspectives

- Accurate spectra prediction of non-toy molecules
  - (nearly) unfeasible with classical computing
  - quantum computers could go further

- Identification of unknown spectra
  - gap between database sizes and chemical space
  - LLM-based models are promising
  - superior accuracy over previous models