QINTELchem v0

Quantum-Integrated Intelligence Engine for Molecular Filtering

QINTELchem is a hybrid classical-quantum logic engine for filtering chemical compounds based on their molecular or simulation-derived properties. It simulates a quantum decision engine that evaluates whether compounds satisfy all user-defined conditions — offering a quantum-inspired way to screen molecules for research or drug discovery.

🔬 What Does It Do?

- Loads a .csv dataset of molecular compounds with associated properties (e.g., dipole moment, polarizability, toxicity).
- Lets users pick **key molecular descriptors** to filter on.
- Accepts flexible rule inputs (e.g., > 2.1, == True, < 0.8).
- Converts these rules into **binary vectors**.
- Simulates a quantum circuit using X-gates on qubits representing matching conditions.
- Marks compounds as matches only if **all selected conditions are met**.
- Exports a list of matching compounds to a new .csv file.

⚛️ The Quantum Circuit Logic

For every compound:

- A **binary vector** is created indicating if each property passed the filter (1) or not (0).
- A quantum circuit is initialized with N qubits (where N = number of selected properties).
- For each binary 1, the corresponding qubit is flipped using an X-gate — simulating a match.
- The circuit then checks if all qubits are $|1\rangle$, indicating a total match.
- Matched compounds are logged and exported.

This mimics a **Grover-style oracle logic**, but operates in a classical simulation of qubit behavior.

⚙️ How to Use

1. Prepare your molecular dataset:
    - File format: .csv
    - Recommended to include a Compound Name column
    - All property values should be **numeric or boolean**
2. Run the script:

    python QINTELchem_v0.py your_dataset.csv
3. Follow prompts to:
    - Select property columns by their index
    - Define condition for each (like == True, > 2.5, < 0.6)
4. Output:
    - Filtered results saved to: qintelchem_output.csv
    - Printed summary of matched compounds in terminal

□ Dataset Format Rules

- Must be a **.csv file** with headers.
- Boolean values must be written as: True or False (capitalized).
- Only numeric (int/float) or boolean columns are accepted.
- String columns are ignored during filtering in this version.

🚀 Future Plans

In upcoming versions, QINTELchem will:

- Integrate **molecule simulation** capabilities for all matched compounds using quantum chemistry tools (e.g., ORCA, Qiskit Nature).
- Add support for:
    - **Partial matches** (e.g., ≥ 2 out of 3 filters passed)
    - **Multi-threshold filters** (like ranges or bounds)
    - **Fuzzy logic and descriptor weights**

- Transition toward a **fully quantum filtering backend** with hardware execution support.
- Optimize performance to handle larger datasets and multi-qubit logic faster.

⚠ Disclaimer

If you encounter any errors, bugs, or unexpected results:

- Please note this was built by a 19-year-old quantum enthusiast with no direct access to a quantum lab or hardware simulators.
- All bugs will be fixed in future versions — your patience is appreciated.
- QINTELchem is built for experimental research and will continue to evolve into a more accurate, faster, and deeply quantum-powered chemical engine.