Computational Framework for Quantum Coherence Analysis

Methods

Author: Martin Doina, Independent Researcher
ORCID: [0009-0002-3855-2268]
GitHub: [gatanegro] | Zenodo: [Zenodo]

1. Protein Structure Selection and Preprocessing

I analyzed high-resolution protein structures from the RCSB Protein Data Bank (PDB) [cite PDB paper]. Structures were selected based on:
- Resolution ≤ 3.0 Å
- Completeness of key functional residues
- Representative proteins from major neural signaling families

Selected proteins and their biological roles:
- Ion channels: NMDA receptor (7EU7), potassium channel (1BL8), sodium channel (6UER), GABA-A receptor (6HUP)
- G protein-coupled receptors: Dopamine receptor (7JVP), serotonin receptor (4IAQ)
- Sensory protein: Rhodopsin (1U19)
- Pathological protein: Amyloid-β (2BEG)

Structures were processed using BioPython PDB parser, extracting atomic coordinates and residue information.

2. Molecular Component Classification

Molecules were categorized using a neural-system-specific database:

```python
 Example classification schema
MOLECULE_TYPES = {
    'neurotransmitter': ['GLU', 'GABA', 'GLY', 'ASP', ...],
    'ion_channel': ['K', 'NA', 'CA', 'CL', ...],
    'signaling': ['ATP', 'GTP', 'CAMP', ...],
```

```
    'structural': ['HOH', 'PO4', ...]
}
```

3. Quantum Coherence Calculation Framework

3.1 Mass-Energy Bridge Formulation

We employed a first-principles approach relating molecular mass to quantum state parameters:

```python
def compute_n_mass(mass, system_type):
    """Bridge Formula connecting mass to quantum recursive depth"""
    ref = SYSTEM_REFERENCES[system_type]
    return π  (log10(mass / ref['mass_ref']) - log10(QDF)) / log10(LZ)
```

Where:
- LZ = 1.234883696... (fundamental recursive constant)
- QDF = 0.809792860... (quantum distribution factor)
- mass_ref = System-specific reference mass (neural: 2.8e-25 kg)

3.2 Quantum State Mapping

Molecular quantum states were mapped to a 9-phase circular Hilbert space:

```python
def octave_position(n):
    """Map n-value to quantum phase position"""
    reduced_n = int(round(n)) % 9
    angle = 2  π  reduced_n / 9
    return cos(angle), sin(angle), reduced_n, angle
```

3.3 Coherence Metrics Calculation

Superposition Ratio:
```python
```

superposition_ratio = molecules_in_coherent_phases / total_molecules
```

Optimal Spatial Scaling:
```python
radius = ref['radius_ref']  (LZ  n)  resonance_factor
```

Mass Coherence:
```python
mass_coherence = (std(masses) / mean(masses))  100
```

4. Validation and Robustness

- Parameter sensitivity: Tested across physiological ranges
- Structure dependence: Verified consistency across similar proteins
- Null models: Compared against randomized molecular distributions

5. Software Implementation

The analysis was implemented in Python 3.8 using:
- NumPy for numerical computations
- BioPython for PDB/CIF parsing
- Matplotlib for quantum state visualization
- **Custom quantum coherence algorithms**

Code availability: [GitHub]
```

- "Ion channels regulate electrical signaling"
- "GPCRs mediate chemical neurotransmission"
- "Rhodopsin is the visual pigment"

- "Neural proteins were selected based on established functional classifications [RCSB PBD]"

"My focus is developing computational methods to detect quantum patterns in protein structures. The biological interpretations are hypotheses generated by our quantitative findings, which we hope will inspire experimental validation by neuroscience laboratories."

"I provide a novel computational framework for quantifying quantum properties in neural proteins. While biological mechanism elucidation requires experimental work, my method offers testable predictions and quantitative metrics for future investigation."

Software: github