

Project RNA Algorithm by Jurijus Pacalovas written

The code you shared (titled "Quantum Computer x Billion Qubits" by Jurijus Pacalovas) is **not a real algorithm for compressing RNA sequences** (or any data in a meaningful way). It has no connection to ribonucleic acid (RNA) biology, sequencing, secondary structure prediction, or legitimate RNA compression tools.

What the Code Actually Does

- It reads a binary file and treats its entire contents as one giant integer (converted to binary string).
- It generates pseudo-random parameters (Divide, Times_12, Multiply, Add_Numbers, SQUARE_OF_ROOT, etc.) by formatting a counter (k2) into a fixed-bit string and slicing it.
- In compression mode: It repeatedly applies the transformation `number = (number * mask) + add**mul` (where mask is roughly $2^{\text{Deep5}} - 1$), counting iterations until (by pure chance) it loops back to the original number — which almost never happens for files larger than a few bytes.
- If it "succeeds" (rarely), it saves minimal metadata (iteration count X1, transformation count, etc.).
- In decompression mode: It reapplyes the same transformations the stored number of times to recover the original.
- It creates unused Qiskit QuantumCircuit objects (no gates, no execution) purely for show.
- The "billion qubits" claim refers to bogus calculations like needing ~800 million qubits for 64 MB files — real quantum compression (Schumacher) only works for multiple identical quantum states, not classical files.

Why It Doesn't Work

- The transformation is **injective** (one-way): it rapidly increases bit length, so reversing by re-applying rarely returns to the start (only for trivial data like all-zero files).
- For real files, the loop runs forever or fails → no compression achieved (output is often larger or garbage).
- It violates information theory: lossless compression can't shrink all possible files smaller on average (pigeonhole principle). Super-compression claims are impossible without data patterns to exploit.
- No quantum advantage: everything runs classically; Qiskit parts are unused props.

Real RNA Compression Algorithms

Actual research on RNA data compression focuses on exploiting biological patterns (repeats, secondary structures, low entropy in sequences). Examples include:

- ****RNACOMPRESS**** (2008): A grammar-based lossless compressor specifically for RNA primary sequences + secondary structures (dot-bracket notation). It models structures with context-free grammars, achieving better ratios than gzip/WinRAR on RNA datasets by capturing pairing rules (A-U, G-C).
- ****GrASS**** (recent): Reference-free lossless compression leveraging grammatical rules, statistics, and substitutions in RNA/DNA — high ratios (up to ~20:1 on some RNA due to redundancy).
- ****Lossy methods****: For RNA-seq (fastq files), tools like lossy quality score compression (e.g., via Zstandard or specialized like Leon/Orcom) reduce size dramatically while preserving quantification accuracy.
- Quantum-related RNA work: Real papers use quantum optimization (e.g., QUBO on D-Wave/IBM hardware) for ****RNA secondary structure prediction/folding**** (minimizing free energy), not classical file compression. Examples: arXiv papers on mRNA folding with 100+ qubits.

The code you have is pseudoscience — overhyped classical math dressed up with "quantum" buzzwords. Jurijus Pacalovas has multiple GitHub repos with similar impossible claims (e.g., Black_Hole compressor, Pi-based transformations + PAQ), but nothing groundbreaking or peer-reviewed.

If you have an actual RNA sequence/file to compress or analyze, I recommend real tools like:

- Zstandard (as in our previous code) for general high-ratio compression.
- Specialized: RNACOMPRESS (if you have structures), or Beam/BSC for genomic/RNA-seq data.

Let me know if you want help with legitimate RNA bioinformatics!