

## **v0\_6 → v0\_6.1 Script Inventory**

### **What each file contains and executes (oldest first)**

This document is a practical inventory of the v0\_6 / v0\_6.1 Python scripts, listed in the same prioritized order you provided (oldest first). Each entry summarizes (a) what the script is for, (b) what it executes, and (c) what outputs to expect. Descriptions are based on the project's established v0\_6/v0\_6.1 structure and the file naming conventions.

#### **Legend (terminology used below)**

- Seed: an RNG seed that produces one random Pauli-sum Hamiltonian instance.
- Near-degenerate pair: eigenvalue pair ( $E_i, E_j$ ) such that  $|E_i - E_j| < \epsilon$ .
- Signature key: an integer-binned descriptor of a candidate subspace (dominant basis count, entropy bins, leakage bin, etc.).
- Leakage: probability mass leaving the candidate subspace under lightweight time evolution; used as a stability proxy.

#### **Script-by-script overview**

1. **v0\_6.py (timestamp: 15-12-2025 12:42 • size: ~4.840 KB)**
  - Purpose: Core v0\_6 pipeline (baseline): systematic signature detection from many random Hamiltonians.
  - Executes: Scans N seeds; for each seed builds a random Pauli-sum Hamiltonian (3 qubits), diagonalizes it, detects near-degenerate eigenpairs, extracts eigenvector structure (dominant basis states + entropy), runs a lightweight dynamic leakage test, then aggregates results into signature families (keys) and prints a frequency summary.
  - Expected outputs: Console summary of signature counts; typically also writes v0\_6\_output.txt (or similarly named) with the same content.
2. **v0\_6\_next\_iteration\_50000\_seeds.py (timestamp: 15-12-2025 13:30 • size: ~5.845 KB)**
  - Purpose: Next-iteration scaling run tuned for 50,000 seeds (same scientific logic as v0\_6, different run configuration).
  - Executes: Runs the v0\_6 pipeline at high seed count. Typically includes micro-optimizations (less verbose printing, faster bookkeeping) and/or adjusted measurement resolution so

large runs complete in reasonable time. The goal is to see whether the signature frequency distribution stabilizes as the sample size grows.

- Expected outputs: Large-run summary: top signature families with occurrence counts; may write a dedicated output log for the 50,000-seed run.

3. **v0\_6\_next\_iteration\_500\_seeds.py (timestamp: 15-12-2025 13:34 • size: ~5.842 KB)**

- Purpose: Next-iteration mid-scale run tuned for ~500 seeds (bridges debugging → large-scale).
- Executes: Runs the same core v0\_6 pipeline at moderate sample size. Used to check that summaries and bins behave as expected before jumping to 50k+ seeds.
- Expected outputs: Console + optional output file with signature summary and basic run statistics.

4. **v0\_6\_sensitivity\_seeds\_60.py (timestamp: 15-12-2025 19:17 • size: ~9.038 KB)**

- Purpose: Sensitivity analysis run (low sample): test how results change under different ‘measurement resolutions’ or binning choices.
- Executes: Executes v0\_6-style scanning for 60 seeds while varying one or more discretization parameters ( $\epsilon$  threshold, entropy binning, leakage binning, dominant amplitude threshold, etc.) to see whether top signature families are stable or fragile to those choices.
- Expected outputs: A compact sensitivity summary for the 60-seed case, often including how many unique signatures appear and which families dominate.

5. **v0\_6\_sensitivity\_seeds\_500.py (timestamp: 15-12-2025 19:17 • size: ~9.040 KB)**

- Purpose: Sensitivity analysis run (medium sample): same as above but with 500 seeds to reduce noise.
- Executes: Repeats the sensitivity experiment with higher statistical power, making it easier to distinguish parameter sensitivity from sampling noise.
- Expected outputs: Sensitivity summary for 500 seeds; often compared directly to the 60-seed and 50k-seed runs.

6. **v0\_6\_sensitivity\_seeds\_50000.py (timestamp: 15-12-2025 19:21 • size: ~9.044 KB)**

- Purpose: Sensitivity analysis run (medium sample): same as above but with 500 seeds to reduce noise.

- Executes: Repeats the sensitivity experiment with higher statistical power, making it easier to distinguish parameter sensitivity from sampling noise.
- Expected outputs: Sensitivity summary for 500 seeds; often compared directly to the 60-seed and 50k-seed runs.

**7. v0\_6\_sensitivity\_fast2.py (timestamp: 15-12-2025 19:48 • size: ~9.865 KB)**

- Purpose: Performance-optimized sensitivity runner: neighbor-scan + integer signature keys (fast large-scale).
- Executes: Same scientific pipeline as v0\_6 sensitivity runs, but engineered for speed: (1) neighbor-scan logic that reduces redundant work and limits the search to ‘nearby’ spectral neighbors; (2) integer-coded signature keys to avoid slow float formatting; and (3) reduced per-seed verbosity. Designed to make 50,000+ seeds practical.
- Expected outputs: Fast summary of signature-family frequencies and the stable-set distribution under the chosen resolution settings.

**8. v0\_6\_1\_interestingness.py (timestamp: 16-12-2025 00:31 • size: ~12.994 KB)**

- Purpose: v0\_6.1 concept prototype: introduce an ‘interestingness’ score to prioritize candidate families.
- Executes: Builds on v0\_6 by defining a composite score that ranks candidates/families by stability and structure (e.g., low leakage + low dominant-basis count, or baseline-normalized leakage). Typically maintains a top-K list of the most ‘interesting’ candidates and reports which signature families dominate the top set.
- Expected outputs: Ranked outputs (top candidates / top families) plus diagnostics. This script is often an intermediate step before the finalized evidence-layer version.

**9. v0\_6\_1\_Final.py (timestamp: 16-12-2025 07:55 • size: ~14.548 KB)**

- Purpose: v0\_6.1 evidence layer (first ‘final’): robustness sweep + enrichment test + hold-out replication.
- Executes: Runs v0\_6.1’s three evidence checks: (1) robustness sweep across  $\epsilon$ /threshold/binning settings; (2) enrichment of signature families in a ‘stable set’ (e.g., top 1% / 5% by stability score), often with bootstrap confidence intervals; and (3) hold-out replication by splitting seeds into two halves and testing whether top families discovered in one half remain top families in the other.

- Expected outputs: Evidence tables: Jaccard overlaps for robustness/hold-out, enrichment factors with CIs, and lists of top signature families. Usually written both to console and to an output text file.

**10. v0\_6\_1\_Final\_patched.py (timestamp: 16-12-2025 09:14 • size: ~22.427 KB)**

- Purpose: v0\_6.1 evidence layer (patched): improved readability + more stable hold-out behavior + output hygiene.
- Executes: Same logic as v0\_6\_1\_Final.py, but with practical patches: more readable tables, safer integer keys, UTF-8-safe logging, and parameter choices that reduce edge cases where hold-out overlap collapses toward zero. This is the recommended script to run for publishable summaries.
- Expected outputs: Clean, 'scientifically readable' output: top families, enrichment factors (with CI), robustness overlaps, and hold-out replication statistics, saved to a dedicated v0\_6.1 output .txt file.

**Recommended default run path**

- If you want a clean, defensible summary without running every variant:
11. 1) Run v0\_6.py (or v0\_6\_sensitivity\_fast2.py if you want a fast large-seed scan).
  12. 2) Run v0\_6\_1\_Final\_patched.py to generate the evidence tables (robustness + enrichment + hold-out).

Note: If you paste the actual console headers for each script (the first ~20 lines of each output), this inventory can be tightened to include the exact parameter values ( $\epsilon$ , thresholds, bins, time steps) used in your runs.