Supplementary Information

Genomic codon usage is structurally consistent with First-Classness across the tree of life

Douglas J. Huntington Moore

April 2026

S1 Family-count sensitivity analysis

The main text reports the family-balanced FC ratio using all 21 concrete coding families (20 sense families plus the stop-codon family). To verify that the null result is not driven by the inclusion of singleton families (Met, Trp) or the stop-codon family, we repeated the permutation test under four family-count configurations (Table S1).

In each configuration, the excluded families' codons are removed from both the observed and null computations. The null model preserves the family-size multiset appropriate to each configuration:

- **21 families** (full): $\{1^2, 2^9, 3^2, 4^5, 6^3\}$, 64 codons.
- **20 families** (excluding Stop3 only): $\{1^2, 2^9, 3^1, 4^5, 6^3\}$, 61 codons.
- **19 families** (excluding Met1 and Trp1 only): $\{2^9, 3^2, 4^5, 6^3\}$, 62 codons.
- **18 sense families** (excluding Met1, Trp1, and Stop3): $\{2^9, 3^1, 4^5, 6^3\}$, 59 codons.

All four configurations were run on a random sample of 1,000 species (seed 2025) with 1,000 null trials each.

Table S1: Family-count sensitivity analysis. The FC ratio ($R = \sigma_{\text{inter}}/\sigma_{\text{intra}}$) is computed under four family-count configurations. No configuration produces a significant result.

Configuration	Families	Codons	Z	p	% > null
Full (21 families)	21	64	0.06	0.43	51.0%
No Stop3 (20 families)	20	61	−0.82	0.79	38.1%
No singletons (19 families)	19	62	0.86	0.19	63.3%
Sense only (18 families)	18	59	−0.19	0.56	45.2%

The Z-scores range from −0.82 to +0.86, all well within chance variation. No configuration approaches conventional significance. The null result reported in the main text is not an artefact of including singletons (which contribute zero within-family dispersion) or stop codons (which have atypically low frequencies). The result is stable across all four partitions of the codon space.

The 19-family configuration (excluding singletons, retaining stops) shows the highest Z of 0.86 with 63.3% of species above the null mean, while the 20-family configuration (excluding stops, retaining singletons) shows the lowest at $Z = -0.82$. This pattern indicates that the stop-codon family, with its very low and uniform frequencies, slightly favours the biological partition, while this effect is insufficient to produce a significant result in any configuration.

S2 GenBank robustness analysis

To assess whether the structural findings reported in the main text are specific to the curated RefSeq dataset, we replicated all three analyses on the GenBank codon usage database, which contains 1,447,521 entries spanning nuclear, mitochondrial, and plastid sequences.

S2.1 Dataset composition and aggregation level

The RefSeq and GenBank datasets differ fundamentally in their unit of aggregation (Table S2).

Table S2: Comparison of entry-level codon count distributions between RefSeq and GenBank (nuclear entries only). RefSeq provides species-level aggregates; GenBank contains individual gene submissions, strain records, and partial sequences.

Statistic	RefSeq nuclear	GenBank nuclear
Entries	70,425	472,190
Median codons/entry	1,149,923	585
Mean codons/entry	4,266,982	90,813
1st percentile	608	41
25th percentile	647,309	247
75th percentile	1,576,101	2,424
99th percentile	29,086,023	1,365,035
Entry size classes		
<100 codons	9 (0.0%)	31,231 (6.6%)
100–1,000	1,438 (2.0%)	255,273 (54.1%)
1,000–10,000	5,533 (7.9%)	151,585 (32.1%)
10,000–100,000	4,392 (6.2%)	21,981 (4.7%)
100,000–1,000,000	19,042 (27.0%)	5,279 (1.1%)
>1,000,000	40,011 (56.8%)	6,841 (1.4%)

RefSeq entries are species-level aggregates with a median of approximately 1.15 million codons per entry; 57% of entries exceed one million codons. GenBank nuclear entries have a median of 585 codons; 54% contain between 100 and 1,000 codons. This nearly 2,000-fold difference in median entry size reflects the different purposes of the two databases: RefSeq provides curated, non-redundant reference sequences aggregated per species, while GenBank accepts individual gene submissions, partial sequences, strain-level records, and other fragmentary data.

S2.2 RefSeq composition

The RefSeq dataset used in the main text contains 70,950 entries, of which 70,425 (99.3%) are labelled `genomic` (nuclear). The remaining 525 entries are mitochondrial or plastid sequences. Filtering to nuclear-only entries produces results identical to the unfiltered analysis (median $R = 0.861$ vs. 0.862; $Z = 0.05$ vs. 0.06), confirming that the main text results are effectively nuclear-only.

S2.3 GenBank results

Table S3 summarises the FC null model results across three GenBank configurations alongside RefSeq.

Table S3: FC null model results across datasets. The GenBank nuclear subset shows an apparently significant result ($Z = 3.15$) that is explained by the small entry sizes in GenBank (Section S2.4).

Dataset	Entries	Median R	Null mean	Z	p
RefSeq (all)	70,950	0.862	0.855	0.06	0.44
RefSeq (nuclear)	70,425	0.861	0.855	0.05	0.44
GenBank (all)	1,447,521	0.926	0.902	0.19	0.38
GenBank (nuclear)	472,190	1.122	0.846	3.15	0.006

The full GenBank dataset (mixing nuclear, mitochondrial, and plastid entries) produces a null result ($Z = 0.19$), consistent with RefSeq. However, the nuclear-only GenBank subset produces an apparently significant result ($Z = 3.15$, $p = 0.006$), with an observed median $R = 1.12$ substantially higher than the null mean of 0.85.

S2.4 Interpretation: entry size as confound

The apparent significance of the GenBank nuclear result is explained by the difference in entry sizes between the two databases. GenBank nuclear entries have a median of 585 codons, compared to RefSeq's median of 1,149,923.

When an entry contains only a few hundred codons, the within-family codon frequencies are dominated by sampling noise rather than genuine codon usage preferences. A family with two codons and a total count of 15 will show high variance simply from integer-count granularity, regardless of any biological preference. The biological partition groups codons that share tRNA recognition machinery, and this grouping happens to reduce within-family variance slightly better than random partitions at small sample sizes—enough to produce a modest but significant effect across 472,000 entries.

When entries contain over a million codons, the within-family frequencies are precisely estimated. Codon usage preferences are fully expressed, and random partitions of similar architecture can achieve comparable inter-to-intra ratios because the frequencies are stable enough that any partition with the same size profile will find similar structure.

This interpretation is supported by the observation that the GenBank nuclear observed median R (1.12) is substantially higher than RefSeq's (0.86), while the null means are comparable (0.85 in both cases). The elevated observed R in GenBank reflects the greater sampling noise in small entries, which inflates within-family dispersion for random partitions more than for the biological partition, because the biological grouping benefits from shared tRNA pools that impose a floor on within-family similarity even at small sample sizes.

The GenBank nuclear result is therefore consistent with a small-sample explanation rather than a robust structural FC signal. On this reading, the RefSeq comparison is primary, because species-level aggregation provides sufficient counts for codon usage preferences to stabilise.

S2.5 CV ordering in GenBank

Despite the noise inflation, the broad CV ordering observed in RefSeq (2-fold < 4-fold < 3-fold < 6-fold) is recognisable in the GenBank nuclear data, though with higher absolute CVs and greater overlap between family-size classes. The full GenBank dataset (mixing compartments) shows further degradation of the ordering, consistent with the known compositional differences between nuclear and mitochondrial genomes.

S2.6 Fine structure in GenBank

The split-family fine-structure ratios on GenBank nuclear data ($AG^*/UC^* = 0.49$ for serine; $UU^*/CU^* = 0.43$ for leucine) are broadly consistent with RefSeq values (0.56 and 0.50 respectively), confirming four-codon sub-block dominance. The full GenBank dataset, which includes mitochondrial sequences with very different codon preferences, produces inverted or distorted ratios ($AG^*/UC^* = 0.25$; $UU^*/CU^* = 1.09$), further supporting the interpretation that compartmental mixing, not the FC metric, drives the discrepancy.

S3 Assignment-level Rosetta concordance (hypothesis-generating)

The main text focuses on structural results that are directly measured in codon-usage data. Here we record an assignment-level concordance observation suggested by the Rosetta correspondence used in the FC alignment map ($i \to \mathrm{G}$, $j \to \mathrm{A}$, $\varepsilon \to \mathrm{U}$, $\eta \to \mathrm{C}$).

In the FC algebraic reading, the codon GGG corresponds to the ordered triplet $\langle i, i, i \rangle$, i.e. a quaternionic rotor form representing pure rotational symmetry. GGG belongs to the glycine family (GGN), and glycine is uniquely achiral among the 20 standard amino acids. Complementarily, UGG encodes tryptophan, the largest standard amino acid by side-chain bulk, giving a second endpoint for structure–property comparison within the same correspondence.

We treat this as hypothesis-generating rather than evidentiary: it is not used in any statistical test in the main text, and it does not enter the derivation of the FC family partition. Its role is to note a possible assignment-level echo that could motivate future targeted analysis.

S4 Reproducibility

Minimal reproduction recipe. From the Zenodo archive for *Genomic codon usage is structurally consistent with First-Classness across the tree of life* (DOI: 10.5281/zenodo.18102019), run:

```
bash scripts/run_verify_setup.sh
bash scripts/run_verify_key_results.sh
bash scripts/run_verify_figures.sh
```

A successful run reproduces the reported key statistics and regenerates Table 1 and Figures 1–3 in publication-ready formats.

All analyses are designed to be reproducible from the archived code and data package. Random number generation uses NumPy's `default_rng` with fixed seed 2025 throughout. The canonical FC metric is implemented once in `fc_metric_canonical.py`; all analysis scripts import this module.

S4.1 Constructive verification of the FC family partition

The family partition stated in Theorem 1 is realised and verified by two standalone Python scripts that implement a strict separation of concerns. The constructive script contains no biological labels or amino-acid mappings; those appear only in the alignment checker, which performs the biological comparison. Both scripts are archived with the Zenodo software verification package for *Genomic codon usage is structurally consistent with First-Classness across the tree of life* (DOI: 10.5281/zenodo.18102019).

Constructive script. The script `fc_redundancy_family_core.py` realises the theorem in executable form. The four-letter alphabet $\{\mathrm{A}, \mathrm{U}, \mathrm{G}, \mathrm{C}\}$ encodes the four dispositional generators $\{j, \varepsilon, i, \eta\}$ via the Rosetta correspondence defined in the main text ($i \to \mathrm{G}$, $j \to \mathrm{A}$, $\varepsilon \to \mathrm{U}$, $\eta \to \mathrm{C}$). The two-two partition $\{\mathrm{U}, \mathrm{C}\}$ and $\{\mathrm{A}, \mathrm{G}\}$ — equivalently $\{\varepsilon, \eta\}$ and $\{i, j\}$ — realises the unique fixed-point-free involution on the generators induced by the κ polarity swap (Theorem 1, clause C). Starting from this alphabet and the FC closure rules stated in Theorem 1 clauses (A)–(D), the script computes the 21 codon families. The script contains no amino acid names, biological labels, or interpretive commentary. On import, the script executes four self-verification assertions corresponding to the constructive verification paragraph in the main text: (i) the 20 sense families are pairwise disjoint; (ii) their union covers exactly 61 codons; (iii) the residue equals exactly $\{\mathrm{UAA}, \mathrm{UAG}, \mathrm{UGA}\}$; (iv) sense codons and residue together exhaust all 64 triplets. Any assertion failure halts execution with a diagnostic message.

Alignment checker. The script `fc_redundancy_check_alignment.py` reads the canonical family dump produced by the constructive script and tests, codon by codon, whether each FC-derived family corresponds to exactly one amino acid family in the universal genetic code (NCBI translation table 1). All biological knowledge — amino acid names, codon-to-amino-acid mappings, FC object classifications — resides exclusively in this checker and nowhere in the constructive script. The report shows a per-family verdict (MATCH, PARTIAL, or FAIL), confirms coverage of all 64 codons and pairwise disjointness, and issues an overall PASS/FAIL verdict. Exit code 0 indicates all checks passed; exit code 1 indicates one or more alignment failures; exit code 2 indicates an input format error.

Verification procedure. The two scripts are run in sequence. First, the constructive script writes the canonical family dump to a file:

```
python3 fc_redundancy_family_core.py --dump families.txt
```

This writes the dump to `families.txt`, displays all 21 families on screen, and prints an instruction for the next step. Then, separately, the alignment checker reads that file:

```
python3 fc_redundancy_check_alignment.py families.txt
```

Alternatively, the two steps may be piped directly:

```
python3 fc_redundancy_family_core.py --dump | \
    python3 fc_redundancy_check_alignment.py -
```

In the piped form, `--dump` without a filename writes the canonical dump to stdout, and the checker reads from stdin when given `-` as its argument. A passing run confirms that the FC closure rules produce a partition that matches the universal genetic code family-for-family under the checker's mapping.

Interactive exploration. The constructive script provides an interactive mode, invoked with no arguments:

```
python3 fc_redundancy_family_core.py
```

This first displays all 21 families with their canonical labels and size distribution, then enters a lookup loop. The user may type any triad over $\{A, U, G, C\}$ (e.g. `AUG`, `UCG`, `GGG`) to see which FC family it belongs to: its canonical label, size, all fellow members, and whether it is a sense-coding family or structural residue. The command `dump` emits the full canonical family listing; `quit` (or `exit`, `q`, Ctrl-D) exits. No biological names appear in any output — the interactive mode presents the theorem's purely algebraic content.

S4.2 Analysis pipeline

The complete analysis pipeline is executed by bash wrapper scripts run in sequence from the package root:

- `bash scripts/run_verify_setup.sh` — prepares the pinned runtime environment and installs locked dependencies. Run once (or follow the script steps in an existing environment).
- `bash scripts/run_verify_key_results.sh` — reproduces the numerical results reported in the paper from 70,950 RefSeq genomes (orbit-by-orbit analysis, diagnostic null test). Expected runtime: 5–10 minutes.
- `bash scripts/run_verify_figures.sh` — regenerates Table 1 and Figures 1–3 in publication-ready formats (PDF, TIFF, SVG). Expected runtime: 3–5 minutes.
- `bash scripts/run_verify_genbank.sh` — replicates all three analyses on 1,447,521 GenBank genomes, verifying that the structural findings hold independently of RefSeq curation. Expected runtime: approximately 3 hours.

Numerical results and logs are written to `scripts/fc_out/`; figures are written to `scripts/fc_plots/`.

Code, curated data, and the software verification package are archived at Zenodo under *Genomic codon usage is structurally consistent with First-Classness across the tree of life* (DOI: 10.5281/zenodo.18102019).