

User Manual

Computational Methods and Reproducible Workflows

Software version 4.02.06 | English edition

A practical guide for future iGEM teams

Contents

catmaOS | User Guide and Computational Methods

1. Scope and supported analyses 3

2. Software architecture 4

3. Installation and environment 5

4. Input preparation 7

5. G-CSF translation analysis 9

6. Analysis workflows 12

7. Computational methods 16

8. Results and exports 18

9. Applying the workflow to another project 20

10. Verification and troubleshooting 21

11. Availability and references 24

Appendix A. Parameters and reruns 25

Appendix B. Output fields and local API 27

Appendix C. Glossary and reporting checklist 28

About this guide

catmaOS provides a browser-based workspace for inspecting sequence-dependent computational analyses and retaining the evidence behind each result. This guide documents the Windows local launcher and its companion Wiki demonstration. Four supported tasks cover structural sensitivity with prepared evidence, conserved slow-codon annotation, relative translation traffic and exact synonymous comparison at one selected codon. A common five-step workflow connects input preparation, parameter selection, execution, interpretation and export.

A complete G-CSF example shows how to reproduce an analysis and inspect its numerical diagnostics. Additional workflows explain structural coverage, strict same-amino-acid evolutionary comparisons and the interpretation of synonymous candidates. Parameters, output fields, source records and troubleshooting procedures are provided so future iGEM teams can use the supported tasks with their own questions and extend the implementation with traceable assumptions.

This edition documents the implemented release. Translation quantities are relative to the supplied host proxy and parameterization. Structural scores and evolutionary annotations provide evidence for investigation; they do not establish experimental folding, pausing, aggregation or functional protein yield.

1. Scope and supported analyses

Begin by selecting the biological question and the required input type. The workspace exposes independent tasks; the five interface steps do not mean that all model layers are calculated in sequence. A completed translation task can legitimately contain skipped structural or yield modules.

Table 1. Supported analysis tasks in software version 4.02.06.

Task	Required input	Principal output
Structural sensitivity	Exact supplied 207-aa G-CSF protein and matching prepared 1RHG chain B evidence.	Reference-aligned TSSS, coverage, feature audit and tunnel-shift sensitivity.
Evolutionary pauses	Exact supplied G-CSF protein, codon alignment, species mapping and tRNA evidence.	Conservation, resistance, per-position coverage and species-level audit.
Relative translation	One valid DNA or RNA CDS; at most 1,800 nucleotides.	Relative flux, bound ribosome burden, exclusion proxy and codon profiles.
Exact synonymous search	A valid CDS and one selected sense-codon position.	All synonymous options at that site, objective values and Pareto membership.

Precomputed demonstration

The Wiki loads four saved real examples and their paired exports. Sequence and configuration controls are read-only. Selecting a case changes the displayed record; it does not invoke a model. The current Wiki includes embedded case data for direct opening after full extraction. Python is not required to inspect these records.

Local calculation

The Windows launcher starts a loopback HTTP service and executes the selected available task in a worker process. User inputs and run outputs remain on that computer. Python installation can require a download; scientific execution on the enabled paths uses supplied local evidence and explicit parameters.

The interface does not supply an arbitrary-protein structure predictor, a trained A2 correction, a calibrated functional-yield predictor or a portable promoter predictor. Inspect module states before attributing a result to any component.

2. Software architecture

The local workspace separates interaction, validation, scientific calculation and presentation. The browser prepares a typed request. The service checks the input and parameter contract before assigning an independent run directory. A worker dispatches to the relevant model adapter and writes normalized outputs alongside native artifacts. The result renderer reads the same metric, curve and table structures in Local and Precomputed modes.

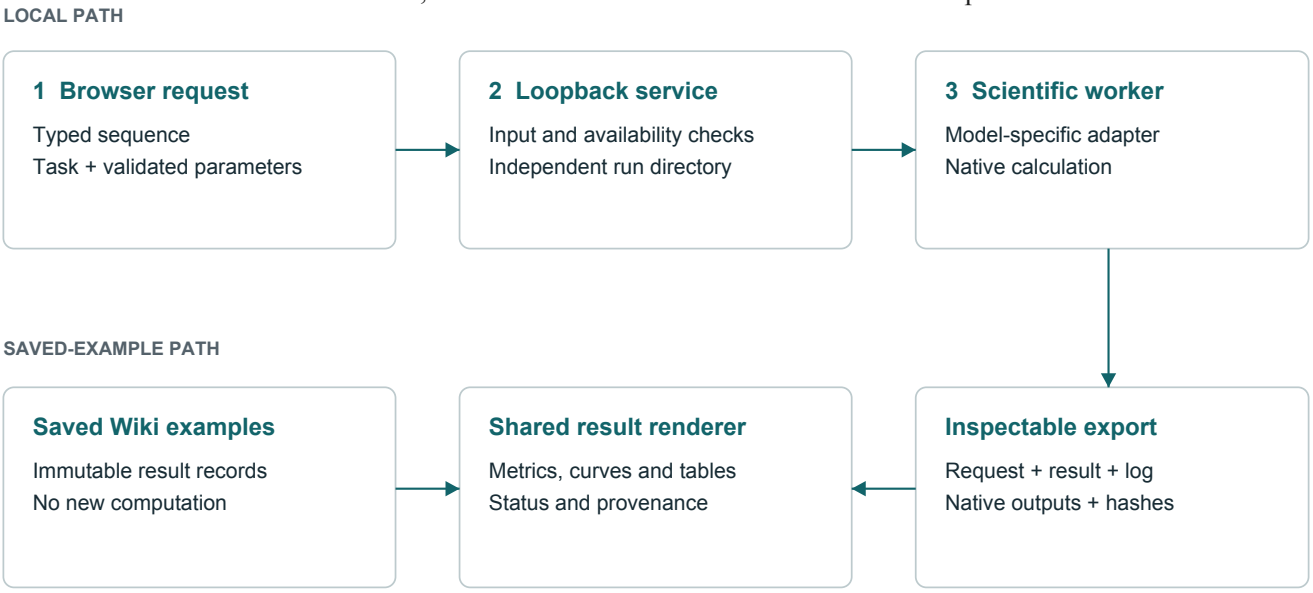


Figure 1. The local execution path and the saved-example path meet at a common result renderer. Validation precedes task execution. Native artifacts and normalized outputs are retained together for inspection.

Source organization

Table 2. Responsibilities of the main package components.

Component	Responsibility
app.py and dogma/server.py	Start the loopback service; expose availability, examples, run state and exports.
dogma/validation.py and runtime.py	Normalize requests, enforce boundaries, track processes and preserve run state.
worker.py and dogma/adapters/	Execute a real task, collect provenance and normalize model output.
web/ and dogma/exports.py	Render results and write readable CSV, JSON and ZIP artifacts.

The adapters preserve the native model identity. Software version 4.02.06, the individual model version and the recorded source revision are separate identifiers. Keep all three when reporting a calculation. Source identities are supplied in SOURCE_MANIFEST.json and the per-run provenance fields.

3. Installation and environment

3.1 Requirements

The distributed launcher targets Windows x64 with CPython 3.14. Its bundled dependency wheels are specific to that target. The saved example records identify Windows 11 and Python 3.14.3 as their original execution environment. The additional checks performed for this guide are described separately in Section 10; they are not a repeat of Windows installation.

3.2 Install Python

Obtain CPython from python.org/downloads/windows/. Select the Windows 64-bit installer for Python 3.14 and complete installation before running setup. Open a fresh terminal after installation so that the Python launcher is discoverable. If multiple Python installations exist, use DOGMA_PYTHON as described in Section 3.5.

3.3 Extract the launcher

Use the Windows launcher ZIP supplied beside this guide on the Wiki Software page. Extract the entire archive into a writable folder, for example C:\catmaOS\4.02.06. Open the extracted release folder containing Setup.cmd and Start.cmd. Keep its setup scripts, source directories, wheelhouse and vendor evidence together. Running scripts while the archive is still open in a ZIP viewer is not supported.

```
release-folder/
 Setup.cmd Start.cmd
 app.py worker.py
 dogma/ web/
 examples/ vendor/
 wheelhouse/ requirements-lock.txt
 SOURCE_MANIFEST.json WHEEL_MANIFEST.json
```

3.4 Set up and launch

Open PowerShell in that extracted folder and run the commands below in order. Wait for setup to finish successfully before launching. Setup creates a private .venv directory; it does not replace global Python packages.

```
.\Setup.cmd
.\Start.cmd
```

Open the exact address printed in the terminal, normally http://127.0.0.1:8767/?mode=local. Confirm Local mode and read the task availability before submitting a calculation. Keep the terminal open while using the browser workspace.

3.5 Choose the intended Python executable

If Python cannot be found, set DOGMA_PYTHON to the actual CPython 3.14 executable in the current PowerShell session, then rerun setup. Replace the example path with the installed location on your computer. This variable selects the setup interpreter; it is not a sequence input or a model parameter.

```
$env:DOGMA_PYTHON = "C:\path\to\Python314\python.exe"
.\Setup.cmd
.\Start.cmd
```

3.6 Understand dependency installation

The launcher includes Windows dependency wheels and a locked requirements file. Normal setup detects wheelhouse and installs from local files using --no-index. WHEEL_MANIFEST.json records wheel versions, sizes, hashes and license locations. Keep those wheels intact if installation must work without package-index access. Installing Python itself remains a separate prerequisite.

The enabled scientific paths do not download trained model weights. Structural and evolutionary tasks reuse matching local upstream evidence. This differs from recomputing a protein structure or retrieving a new homolog alignment from a remote service.

3.7 Check availability without starting a calculation

```
.\.venv\Scripts\python.exe app.py --check
```

The output describes each task, accepted input types, parameter definitions and availability reasons. A missing dependency or evidence file should be corrected before running. Restoring the complete version-matched package is preferable to copying isolated files from another build.

3.8 Change the port or output location

```
.\Start.cmd --port 8770
.\Start.cmd --port 8770 --output-root C:\catmaOS\runs
```

Use a separate output root when another service owns the original runs directory. Open the new address printed by the launcher. If the folder path contains spaces, enclose it in double quotation marks.

3.9 End the session

Wait for completion, export the record, then press Ctrl+C in the server terminal. Closing the browser alone leaves the service running. Interrupting an active calculation does not establish a successful result. Use a new extracted folder for another release, and preserve the original software archive with the exported records.

4. Input preparation

Select the molecule type explicitly. The workspace accepts a pasted plain sequence or one FASTA record. Leading and trailing whitespace, line breaks and letter case are normalized; the biological molecule type remains part of the request. A DNA sequence is not silently accepted as RNA, and ambiguous bases are not replaced with guessed values.

Table 3. Input contracts enforced by the local release.

Input	Accepted form	Important checks
DNA CDS	A, C, G and T; up to 1,800 nt.	Complete codons, no internal stop; footprint must fit the sense-codon count.
RNA CDS	A, C, G and U; up to 1,800 nt.	Declare RNA explicitly; same coding-frame and task checks apply.
Protein	20 standard one-letter amino-acid symbols; general input cap 2,000 residues.	Current evidence tasks additionally require the exact supplied 207-aa G-CSF target.
FASTA	One header at the first nonempty line followed by one sequence.	Multiple records, gaps and unsupported symbols are rejected.

4.1 Check the coding region

Supply the intended coding region rather than an entire plasmid or a transcript containing untranslated regions. The length must be divisible by three. The native CDS parser rejects internal stop codons and preserves the amino-acid identity used by synonymous search. A terminal stop, when present, is excluded from sense-codon elongation rows. This interface does not impose a separate requirement that every input begin with a start codon or end with a terminal stop.

4.2 A small valid formatting example

```
>example_Ala_repeat_DNA
ATGGCTGCTGCTGCTGCTGCTGCTTAA
```

This short synthetic sequence is suitable for a formatting or small-domain exercise. Its nine sense codons require a footprint no greater than nine. To reproduce the scientific worked case in Section 5, select the supplied G-CSF CDS rather than substituting this sequence.

4.3 Common rejected inputs

Examples include ATGNNNTAA (ambiguous DNA letters), ATGGC (incomplete codon), two FASTA records in one input and a DNA string containing U. Correct the source sequence and declared type before retrying. Do not remove biological sequence segments merely to silence a validation error.

4.4 Three coordinate systems

Nucleotide positions, sense-codon positions and protein reference positions are related but distinct. Unless a field states otherwise, displayed positions are 1-based. For a CDS beginning at its first coding nucleotide, codon k spans nucleotides 3k-2 to 3k. A selected sense-codon site must lie within the translated sense-codon region, rather than at the terminal stop.

Table 4. Worked coordinate examples.

Quantity	Example	Interpretation
G-CSF input length	624 nucleotides	207 sense codons plus a terminal TGA.
Mutable codon 2	Nucleotides 4-6; baseline GCT	Alanine at reference residue 2 for the supplied CDS.
Structural position i	Regulatory position i + 35 by default	A model exit-tunnel mapping, not a change to the input sequence.
Terminal stop	Codon 208 in this CDS	Retained in the sequence; excluded from the 207 elongation rows.

4.5 Structure coverage is not sequence length

The supplied protein reference contains 207 residues. The prepared structural observation maps 1RHG chain B onto that reference and leaves missing coordinates explicit. The public structure entry and the full precursor reference have different coverage and numbering contexts [3,4]. Interpret the prepared mapping, observed flags and source-position fields together; do not index the PDB chain as though it directly covered every precursor residue.

4.6 Retain an informative label

Give custom inputs a label identifying the construct and revision. The validated request stores the normalized sequence, its identity hash, molecule type and source text. Distinguish the original database accession from a team-edited construct. A label is descriptive metadata; it does not prove sequence identity.

Keep the accession or construct identifier, full sequence, preparation date, edits relative to the reference, intended host and rationale for the comparison. Pair these details with the software version and parameter values.

For structural or evolutionary tasks, changing the target requires matching upstream evidence and a compatible adapter. The current user interface does not expose arbitrary evidence import. Changing only the protein text while retaining the G-CSF evidence is rejected.

5. G-CSF translation analysis

The first worked case uses the supplied G-CSF RefSeq CDS associated with NM_000759.4 and NP_000750.1. The packaged sequence has 624 nucleotides and translates to a 207-residue precursor plus a terminal stop. The versioned RefSeq protein record and CCDS entry provide the external sequence context [3]. Use the bundled copy when reproducing the recorded case.

5.1 Select the example

In Local mode, choose **G-CSF RefSeq CDS (NM_000759.4)** from the example selector. Verify that the molecule type is DNA, the label identifies G-CSF and the summary reports 624 nucleotides. Select the CDS example for translation analysis; the protein evidence example serves the structural and evolutionary tasks.

5.2 Validate the input summary

Confirm the expected length and provenance before proceeding. The complete coding sequence is already included in the package, so manual transcription from a PDF is unnecessary. If using a FASTA import, compare its normalized sequence with the selected example before expecting the same numerical output.

5.3 Select Relative translation

Choose **Model 3 · Relative translation**. Keep the default parameter values for the first run so that the result can be compared with the recorded reference. The host selector uses the supplied BL21 gene-copy proxy. It is not a measured tRNA abundance or charging profile for a particular culture.

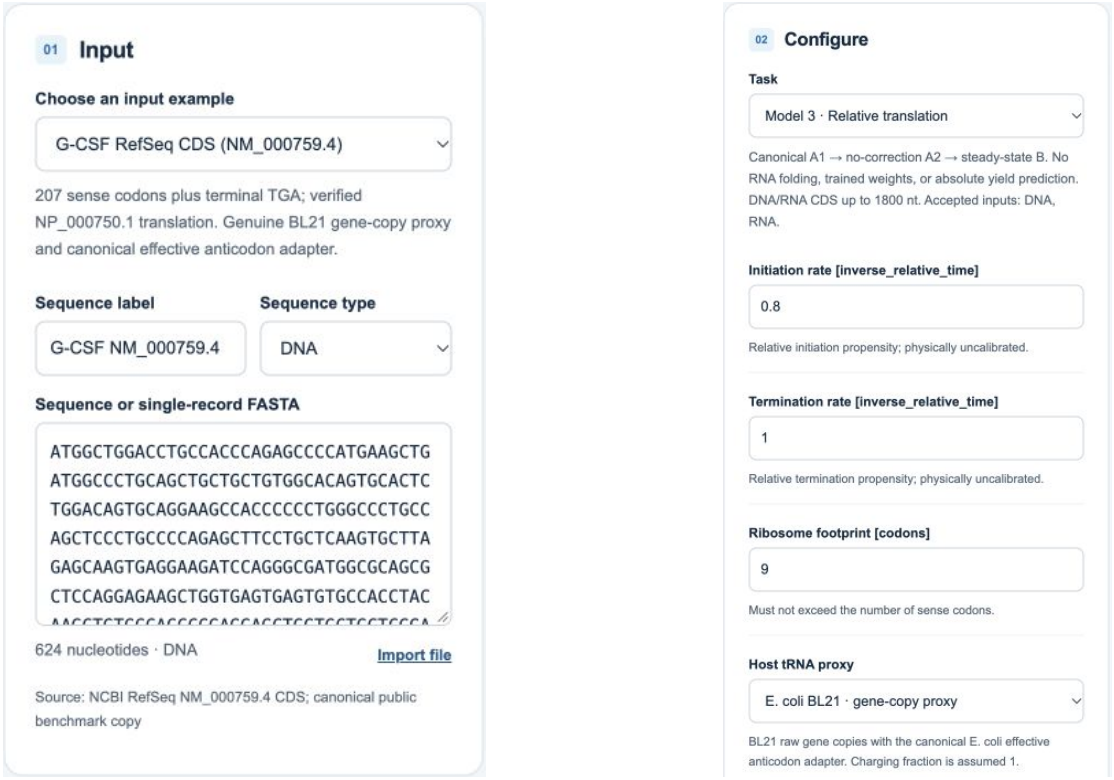


Figure 2. Input (left) and Configure (right) panels captured from the live Local workspace during guide verification. The supplied CDS has 624 nucleotides. Captures are from the macOS verification environment; Windows installation was not retested.

Table 5. Reference configuration for the G-CSF translation case.

Parameter	Reference value	Meaning
Initiation rate alpha	0.8	Relative initiation propensity.
Termination rate beta	1.0	Relative termination propensity.
Ribosome footprint	9 codons	Extended occupancy footprint in the traffic model.
Host tRNA proxy	E. coli BL21 gene-copy proxy	Standard-code supply assumptions; charging fraction is assumed 1.

5.4 Keep comparison conditions fixed

Changing alpha, beta, footprint or sequence changes the scientific request. First reproduce the reference configuration; then vary one justified factor at a time. Keep a short hypothesis beside each change, such as testing sensitivity to assumed initiation propensity. Parameter limits are interface bounds, not a statement that every allowed value is biologically calibrated.

5.5 Start one calculation

Click **Run analysis** once. The interface locks edits during an active run and shows the worker log. There is no percentage-complete estimate or in-page cancellation button. Wait for a terminal state before starting another task. If a calculation fails, inspect the error and export its record before changing the request.

5.6 Confirm what completed

Read the task label, input identity, Local run label, units and module states in Results. For the reference translation task, A1 and the steady-state B calculation complete. A2 uses the no-correction identity; the structural C and yield/stability D paths are skipped. E is skipped because this task does not request a search.

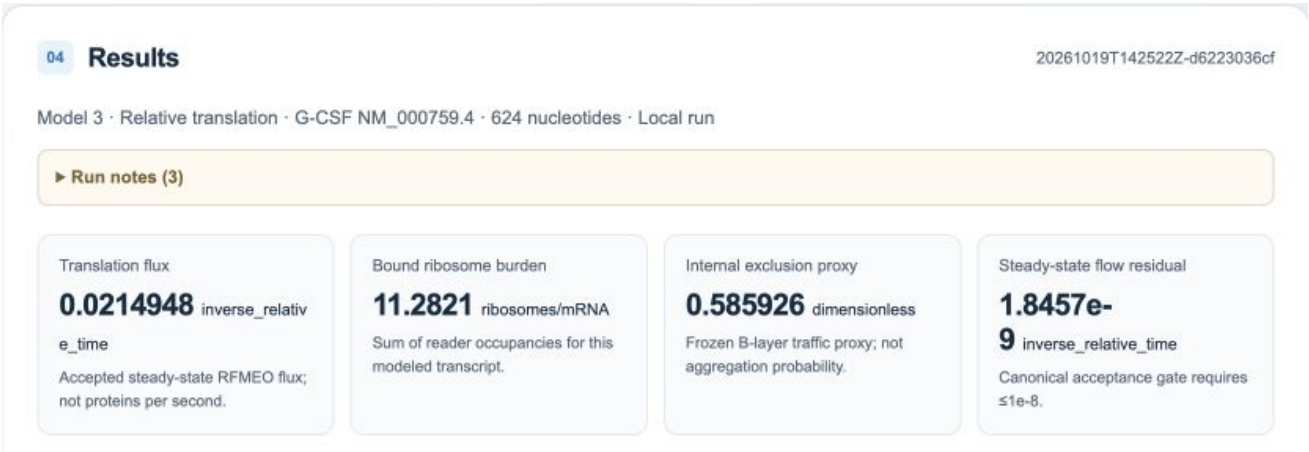


Figure 3. The expected result summary after the reference configuration completes. This panel was captured from a fresh Local run during verification. Section 5.6 explains the metrics and module checks; Table 6 retains the original Windows reference values.

Table 6. Recorded G-CSF translation reference (Windows, Python 3.14.3).

Metric	Recorded value	Unit
Translation flux	0.02149481967	inverse_relative_time
Bound ribosome burden	11.28212804	ribosomes/mRNA
Internal exclusion proxy	0.5859259347	dimensionless
Steady-state flow residual	1.84571879e-09	inverse_relative_time

Table 6 contains recorded reference values from the supplied Windows run, not experimental observations. Use full precision in result.json for a technical comparison. Last-digit equality across different dependency versions is not established by this guide. An accepted native solver state and compatible source/input provenance are stronger checks than a screenshot alone.

5.7 Interpret the residual

The flow residual measures consistency of the modeled steady-state flow. The native acceptance checks include a residual threshold of 10^{-8}, solver success, a physically feasible state and the native derivative diagnostic. Passing those checks supports numerical consistency within this implementation. It does not calibrate model time or verify a biological protein-production prediction.

Scalar metrics summarize the run, while positional profiles show where the model assigns different rates and occupancies. Select a curve, inspect the axis labels and hover for individual values. Use the range controls to focus on a region, then reset before comparing the whole sequence. Chart data and exported CSV values are preferable to visually estimating a point.



Figure 4. Recorded G-CSF translation profiles. Local elongation propensity and reader occupancy are shown on separate axes because they represent different quantities. Positions refer to the 207 sense codons; the terminal stop is excluded. Data are taken directly from the supplied run record.

5.8 What a local peak means

A low intrinsic elongation propensity identifies a model-assigned local rate under the supplied assumptions. A high reader occupancy also depends on traffic from neighboring positions and initiation. Treat the rate and occupancy tracks as connected outputs, rather than interchangeable measurements of pause duration. Footprint coverage and reader occupancy use different definitions.

5.9 Export before changing the input

Choose **Run bundle · ZIP** in Export. Also use the JSON or CSV buttons when a lightweight copy is useful. Changing the input, molecule type, label, task, example or parameters clears the displayed result and its export links; run again to obtain a result for the revised request. Earlier run directories remain independent.

Keep the ZIP with its matching software version and a short interpretation note. Section 8 explains the exported file structure. If a task was unsuccessful, its record can still be useful for diagnosis; a failed calculation is not replaced with a saved successful demonstration.

6. Analysis workflows

6.1 Structural sensitivity

Choose the supplied G-CSF protein example and **Model 1 · Structural sensitivity**. This task recomputes the native V6 feature scores, mapping, fusion and tunnel sensitivity using matched structural evidence and an existing F1 free-energy profile. It does not rerun an upstream structure prediction service. The V5 presentation convention used for a regulatory plot does not change the recorded V6 engine identity.

Procedure and parameters

Confirm a 207-aa protein input. Keep exit-tunnel shift at 35 aa and stable peak support at 8 for the reference run. The shift maps a structural source position to a candidate regulatory position; the sensitivity analysis evaluates shifts from 30 to 40 aa. Stable support counts how often a position belongs to the top decile among the valid shifts.

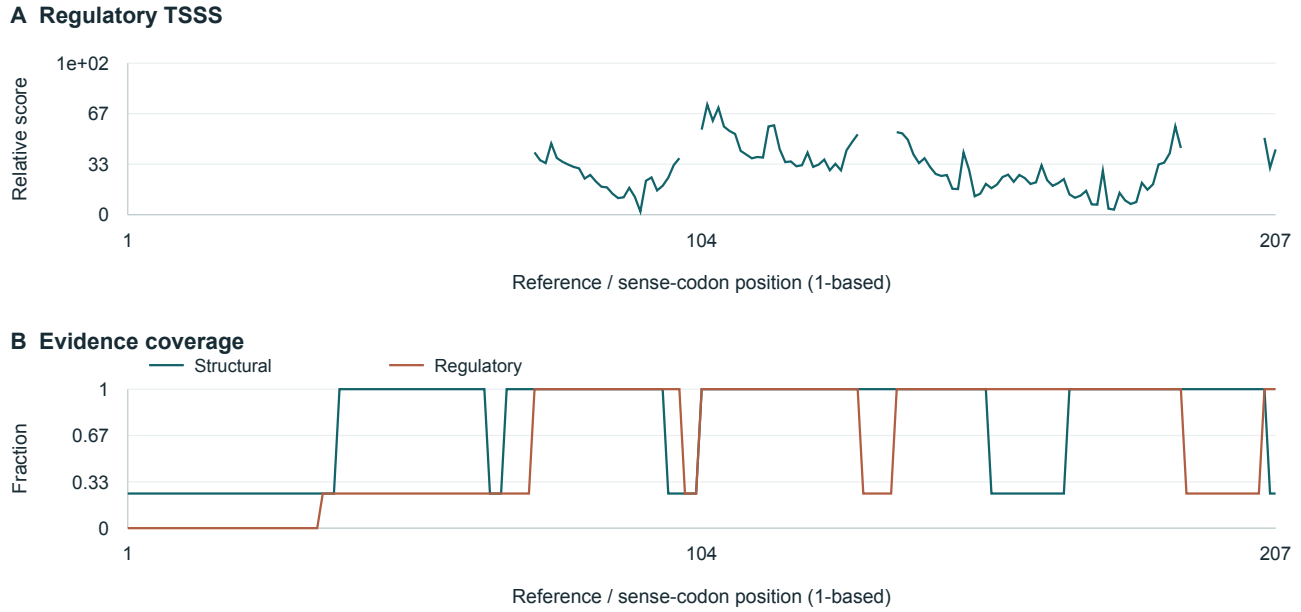


Figure 5. Recorded structural TSSS and evidence coverage. Missing coordinates remain missing rather than being plotted as zero sensitivity. The regulatory profile is shifted relative to structural source coordinates; use source-position and status columns when tracing a candidate.

Recorded structural reference checks.

Reference length	Observed coordinates	Regulatory observed	Stable peaks
207	144	111	0

The reference record has no position meeting the chosen stable-peak threshold. This is a valid result: it does not justify inventing a robust interval or lowering the threshold without reporting that sensitivity analysis. The feature audit records active components and the disabled F4 placeholder. Inspect coverage beside the score, especially near unobserved coordinates and tunnel-unreachable positions.

TSSS is a within-target structural priority score. A larger value is not a measured folding probability, and a missing score is not evidence that the position is unimportant.

6.2 Evolutionary pauses

Choose the supplied G-CSF protein example and **Model 2 · Evolutionary pauses**. The adapter uses the prepared 18-sequence codon alignment, species mapping and tRNA evidence. It computes a species resistance lookup and a target-aligned pause profile. Sequence weighting and explicit coverage accompany the scores so that uneven alignment support remains visible.

Strict mode as the primary analysis

Use **Same amino acid (strict)** with slow fraction 0.25 for the reference case. Slow classification occurs within synonymous families, retaining ties at the threshold. Strict scoring isolates codon variation among sequences encoding the same amino acid at a target position. Permissive mode includes valid codons with amino-acid substitutions and is a sensitivity analysis, not equivalent synonymous evidence.



Figure 6. Recorded G-CSF slow-codon conservation and scoring coverage. A conservation value is interpreted together with the number and coverage of participating homologs. The two tracks are displayed separately to preserve their meanings.

Recorded strict evolutionary reference checks.

Alignment sequences	Target residues	Mean conservation	Mean scored coverage
18	207	0.3738643	0.7903918

Trace a position back to species evidence

Inspect the pause-profile table, then the species-position details and codon-resistance audit. Compare alignment coverage, same-amino-acid coverage and scored coverage. High apparent conservation with little scored evidence warrants a different conclusion from a similarly high score with broad support. tAI-style supply/pairing concepts provide methodological background [2], while the packaged scorer defines the exact implementation.

Conservation and resistance are computed annotations. They do not establish experimentally measured ribosome pausing or a universal slow-codon label across organisms and conditions.

6.3 Exact synonymous search

Relative translation accepts other valid DNA or RNA coding sequences within the stated input contract. Exact synonymous search adds a controlled comparison at one selected sense-codon position. Every candidate is evaluated by the native forward model; the search is exhaustive only over that site, with all other codons fixed.

Configure the G-CSF search case

Choose **GCSF · small synonymous search**, retain the common reference parameters and inspect mutable site 2, evaluation budget 4 and the recorded seed. Position 2 is alanine in the supplied CDS. Its four synonymous DNA codons are GCA, GCG, GCC and GCT. The baseline GCT is included and the terminal stop is preserved.

Table 7. Recorded complete domain at alanine codon 2.

Codon	Flux (relative)	Bound burden	Pareto	Same protein
GCA	0.02149481970161	11.29389991	Yes	Yes
GCC	0.02149481969302	11.29049255	Yes	Yes
GCG	0.02149481968661	11.28801256	Yes	Yes
GCT	0.02149481967086	11.28212804	Yes	Yes

A budget must cover the complete site domain

A selected site can have one to six synonymous options under the standard genetic code. If the evaluation budget is smaller than the full domain, this task rejects the request rather than silently truncating the comparison. The configurable upper bound is 16, although this single-site task normally requires at most six evaluations. The selected site must exist among the sense codons.

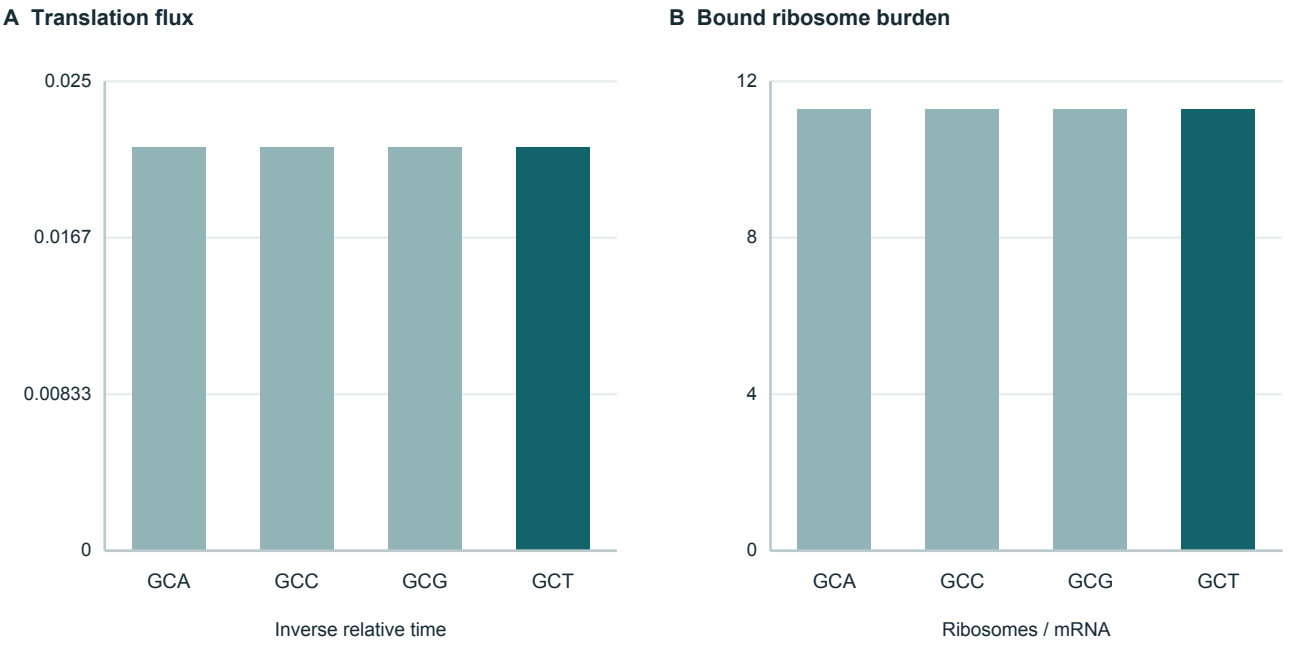
Read the candidate record

Check the chosen codon, edit count, preserved translation, candidate sequence hash, feasibility and Pareto membership. The baseline metrics shown above the candidate table remain those of the unedited sequence; they are not the score of a newly chosen winner. To retain a candidate, export the complete run so that sequence and objective values remain linked.

The exact backend enumerates the domain without random sampling. The seed is retained as provenance; changing it is not a substitute for a different biological input or an independent experimental replicate.

Synonymous candidate comparison

The search compares relative translation flux and bound ribosome burden. A candidate is dominated when another feasible candidate is at least as good on both active objectives and strictly better on one, according to the native comparison rules. Retaining several nondominated candidates communicates a tradeoff, rather than establishing a universally best sequence.



Diagnostic flux change vs GCT (ppb): GCA 1.431, GCC 1.031, GCG 0.733, GCT 0.000

Figure 7. Recorded single-site G-CSF comparison. Panel A uses a common flux scale beginning at zero; panel B shows modeled ribosome burden. A diagnostic line expresses tiny flux changes relative to baseline in parts per billion (ppb). These numerical differences are not experimental improvement.

All four saved candidates have Pareto membership in this example. Their flux values differ only at approximately the 10^{-11} absolute scale, while their modeled burdens differ slightly. Treat the minute flux ordering as a numerical result with solver sensitivity, not evidence that one construct will yield more protein in the laboratory.

Candidate selection

- Verify that every candidate encodes the intended amino-acid sequence and retains required construct features.
- Inspect magnitude as well as rank. Keep the unedited baseline and report the active objectives.
- Repeat a promising comparison under justified parameter sensitivity checks before treating its ordering as robust.
- Choose a manageable experimental set and measure the outcome relevant to the project, such as soluble amount or activity.

This release does not optimize structural folding or functional-yield objectives within the local search. The broader model framework may contain additional layers, but their presence in source material does not mean those outputs were computed for this task.

7. Computational methods

7.1 From a CDS to a relative rate profile

The translation adapter normalizes a typed CDS through the native CandidateCDS parser and builds the supplied BL21 tRNA pool. Gene-copy counts provide abundance proxies; the charging fraction is assumed to be one. The adapter retains 84 standard-amino-acid gene copies, excludes the Sec entry and records the effective anticodon transformations used by the canonical input mapping.

Arg ACG is represented as effective ICG and Ile CAU as effective IAU in the software input convention. These are computational representations, not measurements of a particular tRNA preparation. In particular, the Ile representation follows the canonical AUA-decoder proxy; the lysidine literature explains the biological distinction from a literal inosine substitution [5]. The transformation audit is exported in host-proxy-evidence.json.

Effective charged supply is represented conceptually as abundance multiplied by charging fraction. In this release the abundance source is gene copy number and charging is assumed. Replacing either with measured data requires a documented adapter change; the host selector itself exposes only the supplied proxy.

7.2 The enabled A-to-B path

The adapter uses A1 relative elongation, an identity A2 correction and a steady-state B traffic calculation. RNA-accessibility correction is disabled in this local configuration. The codon-aligned positive relative rate profile supplies the elongation propensities; alpha and beta specify relative initiation and termination.

The B calculation follows the finite-footprint RFMEO approach [1]. In this class of model, occupancy evolves by incoming flow minus outgoing flow. At an accepted steady state, site flows should agree. The extended footprint prevents incompatible occupancy in the modeled transport. Flux is reported in inverse relative time, and the summed reader occupancy is reported as ribosomes per mRNA.

```
site occupancy change = incoming flow - outgoing flow
steady state: incoming flow approximately equals outgoing flow
J = accepted completed-translation flow
R_bound = sum of reader occupancies along sense codons
```

The implementation records solver success, physical feasibility, flow residual and maximum absolute derivative in native-result.json. Section 5 gives the reference acceptance context. Stochastic KMC/TASEP outputs are not produced by this local task and should not be inferred from a traffic proxy.

7.3 Structural evidence integration

Model 1 computes enabled features from the prepared observation and cached upstream evidence, fuses their contributions and records a feature audit. Coverage indicates how much enabled evidence is available at a position. The exit-tunnel mapping places source-position evidence downstream in reference coordinates. The fixed 30-40 aa sweep summarizes how that mapping changes the regulatory profile.

A stable-peak call uses the configured support threshold across available shifts. It does not convert the score into a folding probability. Missing coordinates retain null values and explicit status. Treat differences in evidence availability separately from differences in computed sensitivity.

7.4 Evolutionary slow-state evidence

Model 2 derives species-level resistance from the supplied tRNA evidence, classifies slow codons within synonymous families and aggregates the scored aligned positions using sequence weights. The strict mode excludes amino-acid substitutions from its scored comparison. Alignment gaps, invalid codons and lack of compatible evidence affect which sequences can contribute.

```
slow evidence at position k
 = weighted slow-state contribution / scored sequence weight
coverage and scored-sequence counts accompany that ratio
```

The displayed conservation fraction is a weighted annotation, not a probability estimated from repeated ribosome-pause experiments. Weighted resistance and its standard deviation provide a distinct track. Retain the per-species details when using a high-scoring region to motivate a design.

7.5 Design sensitivity checks around a question

Table 8. Sensitivity checks supported by current controls.

Question	Controlled change	What to retain
Does an annotation depend on tunnel mapping?	Vary tunnel shift within 30-40 aa; inspect the native sweep.	Coverage, source positions and stable support.
Does evolutionary support depend on scoring mode?	Compare strict with permissive as a labeled sensitivity run.	Mode, scored coverage and species details.
Does translation ordering depend on initiation?	Vary alpha while fixing sequence, beta and footprint.	Flux, burden, diagnostics and all run parameters.

A parameter sweep is an analysis of model assumptions. Its spread is not a confidence interval for biological uncertainty unless an explicit calibration and uncertainty model justify that interpretation.

8. Results and exports

Each new local request receives its own directory under runs or the chosen output root. The export ZIP contains the run contents rather than a screenshot of the browser. Files depend on the requested task and its terminal state. Keep the complete bundle when sharing a result with teammates or rerunning a calculation.

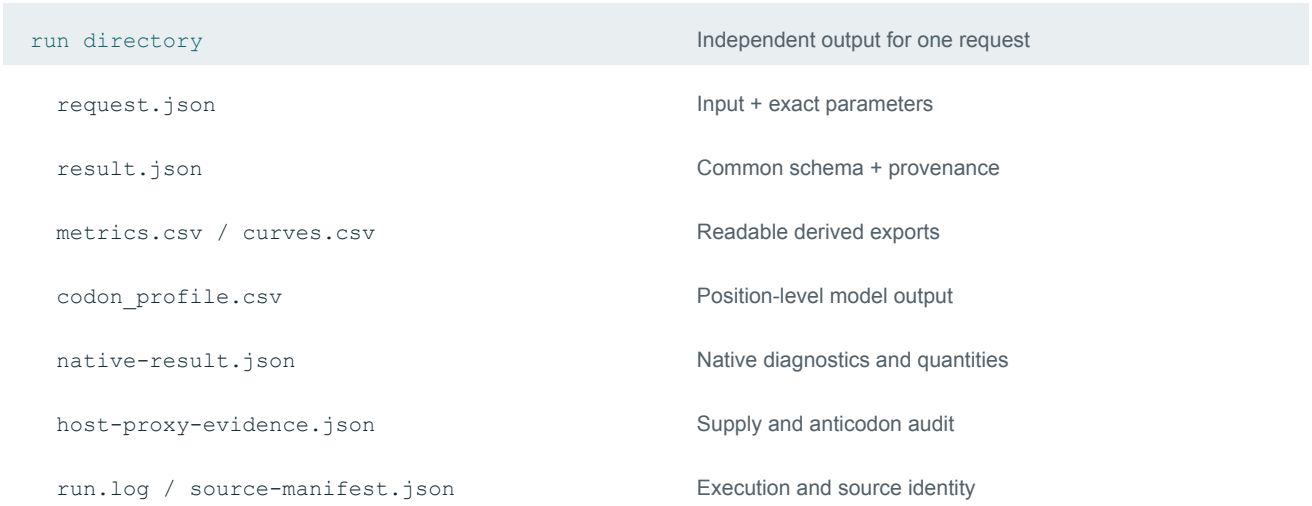


Figure 8. A typical translation run bundle. Structural and evolutionary runs also retain their matching evidence and native-output directories. File presence depends on the executed task and whether calculation reached the relevant stage.

Table 9. Exported files and their roles.

File or directory	Use
request.json	Normalized input, task identifier, validated parameters and request identity.
result.json	Common result schema, metrics, curves, tables, candidates, warnings and reproducibility.
metrics.csv and curves.csv	Readable scalar metrics and long-form positional data, with units.
codon_profile.csv / task tables	Position-level evidence or candidates matching the renderer.
native-result.json or native/	Unmodified native values and diagnostics for technical inspection.
run.log and source-manifest.json	Actual execution log and copied source identity record.

Missing JSON values remain null. In CSV exports, missing values are empty cells rather than zeros. Do not fill them with zero before plotting without a documented reason. Text cells beginning with spreadsheet formula characters are escaped by the CSV writer; use typed numeric fields when doing numerical comparisons.

8.1 Common schema

The normalized result schema is version 1.0.0. A metric has an identifier, label, value, unit and description. A curve specifies coordinate labels, units and one or more series of x/y points. A table declares columns with keys, labels and units, followed by rows. Candidate records retain the sequence and its relationship to the baseline.

```
{
 "schema_version": "1.0.0",
 "task": {"id": "model3_evaluate", ...},
 "status": "completed",
 "metrics": [{"id": "translation_flux",
 "value": 0.021494819670858416,
 "unit": "inverse_relative_time", ...}],
 "curves": [...], "tables": [...],
 "modules": [...], "warnings": [...],
 "reproducibility": {...}
}
```

This excerpt illustrates field organization; ellipses are explanatory and should not be copied as a valid request. For machine reading, use the complete exported JSON. Do not compare values solely because their labels look similar: check metric identifiers, units, input hashes and parameterization.

8.2 Status and availability

Table 10. Status meanings in the workspace.

State	Interpretation
queued / running	A request is active or awaiting execution; wait before export.
completed	The selected task reached a successful terminal state. Inspect its modules.
failed	Execution encountered an error; inspect the log and preserved request.
blocked	A required dependency, source or evidence item is unavailable.
skipped	A component was not requested or not computed in this task.

8.3 Preserve the interpretation

Warnings, provenance and module state form part of the scientific result. Keep them alongside a figure or spreadsheet. An empty candidate list in a forward-only task is expected. A blocked or skipped module is not a zero-valued prediction. Appendix B lists fields useful for reproducible reporting.

9. Applying the workflow to another project

Future teams can use the translation task to investigate how their valid CDS behaves under the supplied relative host proxy, then use the selected-site search for a bounded synonymous comparison. The useful transfer is a repeatable analysis procedure with inspectable assumptions; a universal prediction of expression improvement is not established.

9.1 Prepare a comparison record

- Identify the target protein, intended construct and biological outcome of interest.
- Provide a complete valid CDS and verify its translated amino-acid sequence.
- Document the host assumptions. This interface exposes the supplied BL21 proxy rather than an arbitrary host upload.
- Choose a baseline and a small set of biologically justified variants; hold other conditions fixed.

9.2 Run, compare and explain

Evaluate the baseline first. Export the full run, then repeat with each variant using the same alpha, beta and footprint. Compare flux, bound burden and relevant positional profiles. If variants have different lengths or encode different proteins, state that they answer a different question from a synonymous comparison. Retain the baseline even when the predicted change is small or unfavorable.

9.3 Synthetic sequence example

The synthetic alanine-repeat CDS printed in Section 4 was freshly evaluated with alpha 0.8, beta 1.0, footprint 2 and mutable site 2 during verification. The four candidates below are a software exercise, not an experimentally optimized construct.

Fresh synthetic-case outputs; environment stated in Section 10.

Codon	Flux (relative)	Bound burden	Same protein
GCA	0.1028862	2.761312	Yes
GCC	0.1017939	2.696592	Yes
GCG	0.1009457	2.651033	Yes
GCT	0.09878143	2.548846	Yes

9.4 Plan experimental measurements

Select follow-up experiments that match the claim: total protein, soluble protein and functional activity are different outcomes. Include the original construct and appropriate controls, and document culture and induction conditions. If the objective is a local translation pause, use an appropriate measurement of translation behavior before inferring a pause mechanism from final protein abundance.

For another protein, translation can run on its valid CDS. Structural or evolutionary analysis requires newly prepared matching evidence and adapter support; the present interface does not implement that import workflow.

10. Verification and troubleshooting

Verification is reported at three levels: the provenance of existing examples, checks performed during verification and the biological meaning of model outputs. The packaged Windows records retain their original software, Python, platform and dependency metadata. Additional checks below identify their actual environment rather than extending Windows support claims.

Verification summary.

Evidence level	Observed status
Recorded examples	Four Windows 11 / Python 3.14.3 runs; provenance retained.
Fresh scientific checks	5 completed runs on macOS / Python 3.12.14; four reference tasks and one synthetic exercise.
Python test suites	30 tests: 27 passed, 1 byte-hash mismatch, 2 Windows-only skips.
JavaScript checks	2 test files passed; 23 result-utility assertions plus workspace state checks.
Windows installation	Not repeated during this guide revision.

The structural test mismatch concerns the frozen native CSV byte hash. The largest absolute difference among its exported scalar metrics in the fresh reference rerun was 0. The recorded file uses Windows CRLF line endings and the fresh file uses LF. Their bytes are identical after newline normalization, and all parsed fields match. The original unmodified assertion is retained in the audit record.

10.1 What the supplied tests examine

The Python suites cover typed input and parameter validation, export integrity, loopback service behavior, run lifecycle, structural/evolutionary adapters and the native Model 3 adapter. Tests include missing-value preservation, same-origin submission checks, evidence mismatch rejection, complete synonymous enumeration, translation preservation and numerical acceptance. Windows-specific checks are conditional on the execution platform.

The JavaScript checks exercise result utilities and a lightweight DOM harness. Saved examples serve as fixtures for rendering and export pairing; that harness is not proof of a fresh scientific calculation. Browser inspection is used separately to check the visible interface.

10.2 A user-level reproduction check

- Match software and model source identifiers, input sequence and validated parameters.
- Read native solver acceptance and module states; retain any warnings.
- Check exported numbers against the result renderer and preserve null values.
- Describe numerical deviations under a changed environment rather than hiding them behind rounded screenshots.

Software correctness tests and numerical consistency do not establish biological predictive accuracy. No newly measured protein yield, folding probability, aggregation rate or culture-specific tRNA abundance is introduced by this guide.

10.3 Installation and session problems

Table 11. Operational troubleshooting.

Symptom	Check and corrective action
Python not found	Install CPython 3.14 x64, open a fresh terminal and rerun setup. If necessary set DOGMA_PYTHON to the real executable path.
Missing dependency	Run Setup.cmd in the fully extracted folder. Keep wheelhouse intact and preserve the exact setup error. Windows wheels do not provide macOS/Linux dependencies.
Browser cannot connect	Keep the Start.cmd terminal open and use its printed loopback address. Read terminal errors before retrying.
Port already in use	Stop the prior instance with Ctrl+C or start with --port 8770. Use the corresponding printed address.
Runs folder already owned	Stop the earlier service or provide a separate --output-root directory. Do not remove ownership records while a service is active.
Repeated clicks / active task	Wait for the current task. The service permits one active calculation and keeps request identity to avoid duplicate work.
Old result disappears after editing	This is expected. Revised inputs/configuration require a new run; old exports must not be attributed to a changed request.
Wiki example does not load	Extract the whole Wiki, keeping software-demo, demo-data.js and the case artifacts together. The current embedded demo can open directly; a local HTTP preview is also possible.

Collect a useful problem report

Retain the software version, task, operating system, Python version, exact message and run ZIP. For an installation issue, preserve the setup terminal output. For a scientific failure, preserve request.json, result.json if present, and run.log. Review any sequence or project information before sharing it outside the team.

Use a fresh output directory for a retry. A failed record can explain the problem; overwriting it removes evidence. A file merely being present in a directory does not imply that its task completed successfully.

10.4 Input and evidence problems

Table 12. Scientific-task troubleshooting.

Symptom	Interpretation and action
Invalid coding sequence	Check the declared type, alphabet, frame and internal stops. Import one record and remain within 1,800 nt.
Footprint exceeds sequence length	Count sense codons, excluding any terminal stop. Use a footprint that fits the modeled CDS.
Exact search budget too small	Allow the full synonymous domain at the selected site. The software rejects a partial domain instead of calling it exhaustive.
G-CSF evidence mismatch	Select the exact supplied protein example. A different sequence requires its own matching evidence and adapter.
Evidence hash mismatch	Restore the version-matched files. Changing a prepared source/evidence file invalidates its recorded identity.
No cognate supply / native error	Read the exact codon and supplied pairing assumptions in the log. The software does not insert a guessed replacement score.
Missing structural values	Inspect coordinate coverage and status. Missing coordinates or tunnel-unreachable mappings are not zero sensitivity.
No stable structural interval	Report the negative threshold result. Explore a labeled sensitivity analysis if scientifically justified.
Tiny candidate flux differences	Inspect native precision and parameter sensitivity. Do not interpret a minuscule numerical rank as a demonstrated expression improvement.

Documenting unsuccessful runs

A contribution includes a clear account of boundaries and failure modes. Record the request that failed, the diagnostic and the corrective step. Distinguish unavailable evidence from a valid low score, and a numerical failure from a biological observation. If a feature is unsupported in the current release, document that gap before proposing an extension.

11. Availability and references

11.1 Resource access

The Wiki Software page distributes this manual, the Windows launcher and the four-case example bundle. Use the three resources together. In an extracted Wiki, open drylab-tool.html and its Downloads section. Paths below are relative to the downloads directory; filenames preserve release capitalization.

Table 13. Version-matched resources.

Resource	Relative file
Expanded English guide	catmaOS-4.02.06-User-Manual-EN.pdf
Windows launcher	catmaOS-4.02.06-windows-launcher.zip
Recorded worked examples	catmaOS-4.02.06-gcsf-examples.zip
Editable guide source and audit record	catmaOS-4.02.06-Manual-Source.zip

11.2 Redistribution and maintenance

The supplied package describes a local review build and records that a complete project release license has not been established. Original source and third-party notices remain relevant. This manual documents the available resource; it does not grant additional rights over code, datasets or dependencies. Confirm the applicable project and evidence permissions before public redistribution. Cite original methods and the exact catmaOS software version when reporting results.

A team extending the software should retain a new version identifier, tests for the changed behavior, updated example records and clear data provenance. A custom host input, arbitrary structural evidence import or additional biological objective requires a matching implementation and validation record. Describe proposed extensions separately from shipped capabilities.

References

[1] Zarai Y, Margaliot M, Tuller T. Ribosome flow model with extended objects. J R Soc Interface. 2017;14:20170128. Source.

[2] dos Reis M, Savva R, Wernisch L. Solving the riddle of codon usage preferences: a test for translational selection. Nucleic Acids Res. 2004;32:5036-5044. Source.

[3] NCBI RefSeq and CCDS. CSF3 precursor NP_000750.1, reference mRNA NM_000759.4 and CCDS11357.1. Source.

[4] Hill CP, Osslund TD, Eisenberg D. The structure of granulocyte-colony-stimulating factor and its relationship to those of other growth factors. Proc Natl Acad Sci USA. 1993;90:5167-5171. PDB 1RHG. Source.

[5] Muramatsu T, et al. A novel lysine-substituted nucleoside in the first position of the anticodon of minor isoleucine tRNA from Escherichia coli. J Biol Chem. 1988;263:9261-9267. Source.

[6] catmaOS supplied software 4.02.06. README, model adapters, source manifests, result schema, tests and recorded G-CSF examples. Local technical sources retained with this guide.

Appendix A. Parameters and reruns

The values below come from the current adapter definitions. The interface rejects unknown parameters, nonfinite numbers, Boolean values in numerical fields and values outside declared bounds. Numeric bounds specify the supported interface domain; they do not provide experimental calibration.

Table A1. Translation parameters.

Key	Default / bounds	Meaning and use
alpha	0.8; 0.01-2.0	Relative initiation propensity in inverse relative time. Hold fixed for a sequence comparison.
beta	1.0; 0.01-2.0	Relative termination propensity in inverse relative time.
footprint_codons	9; integer 1-15	Number of codons covered by one modeled ribosome; must fit the sense-codon count.
host_proxy	BL21 supplied proxy	Only the supplied effective-anticodon gene-copy pool is exposed. Charging fraction is assumed 1.

Table A2. Additional exact-search parameters.

Key	Default / bounds	Meaning and use
mutable_site	2; integer 1-600	1-based sense-codon position; must exist in the submitted sequence.
max_candidates	4; integer 1-16	Evaluation budget; must cover all synonyms at the selected site.
seed	20261008; 0-2147483647	Recorded provenance for exact enumeration; no random sampling in this backend.

Example configuration

```
task_id: model3_optimize
sequence_type: DNA
alpha: 0.8
beta: 1.0
footprint_codons: 9
mutable_site: 2
max_candidates: 4
seed: 20261008
```

This compact configuration documents the controls, rather than a complete API payload. Use the UI for normal operation or retain a validated request.json when reproducing a run. Source-level host changes must preserve a traceable supply model and the resulting source revision.

Table A3. Structural-task parameters.

Key	Default / bounds	Meaning
tunnel_shift	35; integer 30-40 aa	Regulatory position equals structural source position plus the selected shift.
stable_min_count	8; integer 1-11 shifts	Required top-decile support in the fixed 30-40 aa sensitivity sweep.

Table A4. Evolutionary-task parameters.

Key	Default / bounds	Meaning
slow_fraction	0.25; 0.1-1.0	Top resistance fraction within each synonymous family; retain ties at the cutoff.
amino_acid_mode	strict or permissive	Strict requires same amino acid; permissive includes valid amino-acid substitutions as sensitivity analysis.

Advanced rerun from an exported request

Keep the matching launcher version. Extract the contents of a run export into a new directory under runs, ensuring that request.json is directly inside that directory. Use a fresh directory for execution; do not overwrite the original result bundle. The worker reads its validated request and writes new outputs in the provided location.

```
.\.venv\Scripts\python.exe worker.py --run-dir runs\NEW_DIRECTORY
```

For normal use, repeat through the interface so that a new request identity and independent directory are generated. An advanced rerun assumes the matching source and evidence are available. Validate the new result, its parameters and native diagnostics before comparing it with the original.

Suggested naming practice

Name exports using a construct identifier, software version, task and run identifier. Record why a parameter was changed, rather than relying on a folder name to encode all scientific context. Preserve original request/result files and derived figures separately so that downstream analysis does not change the underlying record.

The per-run code_sha256 field is the SHA-256 of SOURCE_MANIFEST.json in this implementation. It is not a new hash of every source file at run time. Use the manifest, retained archive and individual model source revision together.

Appendix B. Output fields and local API

Table B1. Selected normalized-result fields.

Field	Meaning
schema_version	Format contract for the normalized result; currently 1.0.0.
run_id / mode / status	Run identity, presentation/execution mode and active or terminal state.
input	Normalized sequence identity, molecule type, length, label and source.
task / parameters	Selected analysis and exact validated controls.
model	Model identifier, version and source revision.
modules	Component status and reason; completed, skipped and blocked remain distinct.
metrics / curves	Scalar values and coordinate-aware series with identifiers, descriptions and units.
tables / candidates	Typed evidence rows and synonymous sequences/objectives when produced.
warnings / errors	Interpretation limits and execution diagnostics.
artifacts	Produced files with paths, sizes, media types and file hashes.
reproducibility	Software/Python/platform/dependency versions, command, timing and source-manifest hash.

CSV exports

metrics.csv contains Metric, Value, Unit and Description. curves.csv contains Curve, Series, X, Y, X unit and Y unit. Named table CSVs follow their declared columns. The codon profile excludes the terminal stop.

Reporting a figure

Report construct, task, versions, host proxy, parameters, coordinates and units. Identify recorded versus fresh runs. Retain evidence coverage and the search baseline/objective directions.

Optional local HTTP integration

Table B2. Existing loopback API routes.

Route	Purpose
GET /api/health	Software version and per-session token.
GET /api/capabilities; /api/examples	Task definitions, availability and supplied input examples.
POST /api/runs	Submit a validated JSON task; returns a run identifier.
GET /api/runs/{run_id}	Read status, result and log for that run.
GET /api/runs/{run_id}/export	Download the completed or unsuccessful run bundle; active runs cannot export.

Submission requires application/json and X-Dogma-Token. Use the printed loopback address. Host and Origin checks apply; these routes support local integration.

Appendix C. Glossary and reporting checklist

Table C1. Working terminology.

Term	Meaning in this guide
CDS	Coding sequence supplied for a translation task.
TSSS	Translation-Speed Sensitivity Score; a relative structural-priority annotation.
Gene-copy proxy	tRNA gene counts used as an abundance assumption.
Charging fraction	Fraction of a tRNA pool carrying amino acid; assumed 1 in the exposed host configuration.
RFMEO	Deterministic ribosome-flow model with extended occupancy objects.
Reader occupancy	Modeled occupancy associated with the ribosome reader position.
Footprint coverage	Modeled coverage accounting for an extended ribosome footprint.
Pareto candidate	A feasible candidate retained under the active multi-objective comparison.
Provenance	Traceable input, method, version and source information associated with a record.

Reporting and citation

Retain the sequence identity, task, host assumptions, parameters, versions and module states with the full run ZIP. Check diagnostics and warnings before reporting a result. Figures should identify units, coordinates, missing evidence and the comparison baseline. State which conclusions are relative model outputs and which are measured biology, and retain applicable code, data and third-party notices.

Software citation: catmaOS team. catmaOS, software version 4.02.06. User Guide and Computational Methods. Include the team Wiki or repository URL when available.