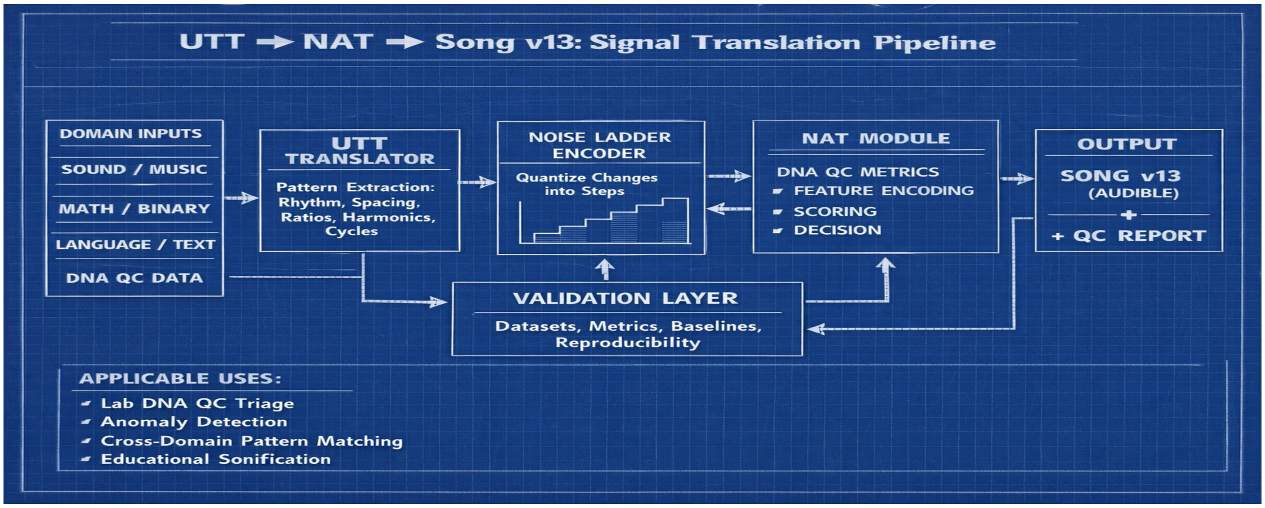
NAT Song v13.0 - FastQC Inputs

Frozen Encoding Spec + Blinded Validation Kit

Version: v13.0 (frozen)   **Date:** January 18, 2026

Purpose: Lock the NAT-to-Song mapping (no moving targets) and prove it works on unseen public data. NAT uses **FastQC** module outputs as inputs, then produces: (1) a numeric QC score, (2) PASS/WARN/FAIL, and (3) a deterministic audio render (the NAT song snippet) that encodes the QC state.



1) NAT Inputs (FastQC modules) and weights

FastQC module (input)	What we extract	Penalty (WARN / FAIL	Song control (deterministic)
Per base sequence quality	Mean Q + low-tail fraction	10 / 25	Pitch center (down) + detune (up)
Per sequence quality scores	Low-Q mode + spread	2 / 5	Dynamics range: flatter = worse
Adapter content	Adapter % (max)	8 / 20	Click/hi-hat density: more = worse
Overrepresented sequences	Count + top fraction	6 / 15	Alarm motif repetition: more = worse
Per sequence GC content	Delta from expected + skew	4 / 10	Timbre brightness: drift = darker
Sequence duplication levels	High-duplication %	4 / 10	Loop tightness: higher dup = tighter
Sequence length distribution	Spread + truncation	2 / 5	Note duration jitter: more = worse
Per base N content	Max N %	4 / 10	Noise floor amplitude: more = worse

2) NAT Score + decision rule

Initialize score = 100.
Subtract penalties per module based on FastQC status:
- If module is WARN, subtract WARN penalty.
- If module is FAIL, subtract FAIL penalty.

Hard overrides: If *Per base sequence quality* is FAIL or *Adapter content* is FAIL, decision is FAIL regardless of score.
Decision thresholds: PASS if score >= 85. WARN if 70-84. FAIL if < 70.

3) Deterministic audio render rules (Song v13)

Tempo: 120 BPM. **Length:** 16 bars per sample. **Seed:** seed = hash(sample_id + "v13.0") to guarantee identical renders.
Arrangement: 4 tracks (Bed, Pulse, Perc, Noise). Each FastQC module controls one parameter only (no double-dipping).
PASS: stable harmony + low noise. **WARN:** mild detune + extra clicks. **FAIL:** strong detune + alarm motif + high noise floor.

Song parameter	Range	Driven by	Rule
Pitch center	MIDI 48-72	Per base quality	Higher mean Q -> higher center
Detune amount	0-35 cents	Low-tail fraction	More low-tail -> more detune
Click density	0-8 hits/bar	Adapter content	More adapters -> more hits
Alarm motif	0-4 repeats	Overrep sequences	More overrep -> more repeats
Timbre brightness	Lowpass 800-6k Hz	GC drift	More drift -> lower cutoff
Loop tightness	1x-4x loop	Duplication	Higher duplication -> tighter loop
Duration jitter	0-25% swing	Length spread	More spread -> more jitter
Noise floor	-24 to -6 dB	N content	More N -> higher noise

Blinded Validation Protocol (GIAB + Option D)

This is the "no cheating" step: frozen mapping, unseen runs, scored after unblinding.

Phase	What you do	Outputs
A. Build truth set (30)	10 GIAB human WGS + 10 FDA-ARGOS microbial + 10 Zymo mock community. Assign IDs S01-S30 and hide sources/labels.	Blinded ID list (S01-S30)
B. Run FastQC	Run FastQC on each FASTQ (or each pair). Record module status + key numeric summaries.	FastQC reports + extracted metric
C. NAT scoring + song	Apply frozen v13.0 penalties and render the 16-bar deterministic snippet.	Score + decision + song file
D. Unblind + score	Reveal true source and expected QC label. Compute confusion matrix + sensitivity/specificity.	Metrics table + plots
E. Repeatability	Pick 5 samples. Re-run 3 times. Confirm identical numeric scores and identical audio renders.	Repeatability evidence

Scoring rubric (success criteria)

Minimum pass bar:
- Overall agreement with the expected QC category: >= 80% on the 30-sample set.
- FAIL sensitivity (catch true bad runs): >= 85% (priority is not missing failures).
- Repeatability: same input yields the same score and the same rendered audio (byte-identical if possible).

Report: include confusion matrix and list every misclassified sample with the FastQC modules responsible.

How to create "known bad" controls from public data

To guarantee WARN/FAIL cases (without relying on subjective labels), create transformed copies of reads:
- Add adapter sequences to a fraction of reads (adapter FAIL).
- Trim reads harshly (length distribution WARN/FAIL).
- Corrupt tail quality (per-base quality FAIL).
- Introduce Ns (N-content WARN/FAIL).

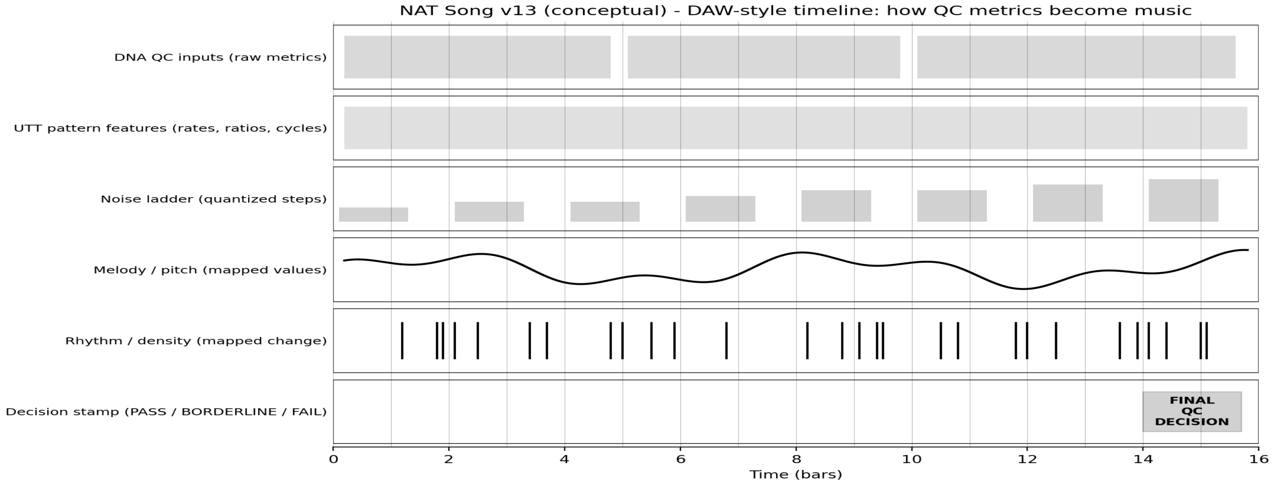
These synthetic controls should always score worse, and the audio should become more noisy/dissonant in a predictable way.

Blinded Worksheet Template (what to fill)

Use the CSV template that ships with this PDF. Keep source and truth labels hidden until scoring.

Column	Meaning
sample_id	Blinded ID (S01-S30)
source_hidden	GIAB / FDA-ARGOS / Zymo (leave blank until unblinding)
run_or_file_id	Run accession or filename
fastq_1	Path/URL to R1
fastq_2	Path/URL to R2 (optional)
fastqc_status_*	PASS/WARN/FAIL for each selected module
fastqc_metric_*	Numeric summaries extracted from the FastQC report
nat_score_v13	Score computed from penalties (0-100)
nat_decision_v13	PASS / WARN / FAIL after overrides
song_render_path	Audio output path for the 16-bar snippet
notes	Any observations
truth_label_hidden	Expected QC label (filled only after unblinding)
correct	TRUE/FALSE after scoring

Song v13 visual timeline (what the render represents)


NAT Song v13 (conceptual) - DAW-style timeline: how QC metrics become music
DNA QC inputs (raw metrics)
UTT pattern features (rates, ratios, cycles)
Noise ladder (quantized steps)
Melody / pitch (mapped values)
Rhythm / density (mapped change)
Decision stamp (PASS / BORDERLINE / FAIL)
FINAL QC DECISION
0
2
4
6
8
10
12
14
16
Time (bars)

Interpretation: more clicks/noise/repeats and lower pitch center indicate poorer QC; PASS stays clean and stable.

Lock rule: do not change the mapping or weights until the blinded scoring is complete. Any changes become v13.1 and must be revalidated.