

The Ribosome Network: A Decentralized RNA Inverse-Folding Subnet as a Synthetic Ribosome

Aqila

Independent Researcher

aqila@aqila.site

Abstract

RNA inverse folding — designing a nucleotide sequence that folds into a target 3D structure — is a fundamental problem in synthetic biology with direct applications in ribozyme engineering, riboswitch design, and mRNA therapeutics. The problem is computationally hard (NP-hard in the secondary-structure formulation), the data is scarce (RNA-only entries are under 1% of the PDB), and current state-of-the-art methods are tightly held by a small number of academic labs. We propose the **Ribosome Network**: a Bittensor-style decentralized subnet in which miners act as synthetases (proposing candidate sequences for a withheld target structure) and validators act as molecular chaperones (refolding each candidate via an independent structure predictor and scoring the structural match). The central contribution is conceptual: we show that the protocol design of such a subnet is *operationally isomorphic* to ribosomal biology, and that this isomorphism yields principled defaults for otherwise-arbitrary economic parameters — diversity bonuses map to evolutionary pressure, target-pool rotation maps to mRNA turnover, and commit-reveal schemes map to transient mRNA protection. We instantiate the protocol with a Phase 1 pipeline (BeeRNA generator + RhoFold refold oracle + TM-score/RMSD scoring) and describe the Phase 2 subnet layer as interface stubs. We are explicit about which components are implemented, which are proposed, which empirical parameters remain open, and where the biological analogy does or does not survive quantitative scrutiny.

1 Introduction

The RNA design problem asks: given a target three-dimensional RNA backbone structure, find a nucleotide sequence predicted to fold into it. Despite the field’s importance — designed ribozymes, riboswitches, and mRNA therapeutics all depend on it — current methods remain meaningfully behind the protein-design frontier. Three observations motivate this work.

First, RNA structural data is scarce. RNA-only entries constitute less than 1% of the Protein Data Bank (Berman et al., 2000). The RNASolo repository (Adamczyk et al., 2022) provides a curated, cleaned subset, but the underlying scarcity means that any single lab’s training set has high overlap with any other’s test set, and published benchmark numbers are sensitive to split choices (sequence similarity vs. structural similarity clustering). This makes honest benchmarking difficult and creates a structural advantage for whichever lab holds the largest private corpus.

Second, the evaluation metric itself is contested. Native sequence recovery — the percentage of positions where the designed sequence matches the native — was the dominant metric through 2024 (Joshi et al., 2025; Tan et al., 2025). But RNA sequence-to-structure mapping is degenerate: many distinct sequences fold to the same structure, so native-sequence recovery penalizes valid alternative solutions. Recent work (Mlaweh et al., 2025; Huang et al., 2024) has shifted to

a refold-and-compare protocol: take the designed sequence, refold it independently with RhoFold (Shen et al., 2024), and compare the predicted fold to the target via TM-score (Zhang and Skolnick, 2004) and RMSD. This is methodologically better but computationally expensive — each evaluation requires a forward pass through an independent structure predictor.

Third, the field is centralized. The leading academic methods — gRNAd (Joshi et al., 2025), RiboDiffusion (Huang et al., 2024), RISO-TTo (Bibekar et al., 2025), BeeRNA (Mlaweh et al., 2025) — are open-source, but the compute, data, and expertise required to reproduce or extend them are concentrated in a handful of institutions. A researcher without access to multi-day GPU training runs or curated structural datasets cannot meaningfully contribute.

We propose the **Ribosome Network**, a Bittensor-style decentralized subnet (Rao, 2020) that addresses all three problems. Miners receive a withheld target structure, submit candidate sequences, and are scored by validators using the refold-and-compare protocol. The target pool is rotated periodically to resist memorization; commit-reveal prevents within-round copying; a diversity bonus rewards structurally valid but meaningfully different submissions. The protocol is designed so that honest participation — submitting genuinely novel candidate sequences that fold correctly — is the profit-maximizing strategy.

The paper’s central claim is conceptual rather than empirical: **the protocol design of an RNA inverse-folding subnet is operationally isomorphic to ribosomal biology**¹ (Table 1). Miners are synthetases; validators are chaperones; the target pool is the transcriptome; diversity bonuses are evolutionary pressure; target rotation is mRNA turnover. This isomorphism is not merely metaphorical — it provides principled defaults for economic parameters that would otherwise be set arbitrarily. The diversity bonus weight, for instance, is not a free hyperparameter but an instantiation of the exploration-exploitation tradeoff that biological evolution has tuned over billions of years.

We are explicit about the paper’s scope. Phase 1 — the standalone ML pipeline (BeeRNA generator, RhoFold adapter, scoring, benchmark loaders) — is implemented and runs end-to-end with stub oracles. Phase 2 — the decentralized subnet layer — is described as interface stubs with a clear specification. We do not report benchmark numbers, because the only honest benchmark requires RhoFold to be installed and that integration is not yet validated. The contribution is the protocol design and the isomorphism that justifies it.

2 Background

2.1 RNA inverse folding

The RNA inverse folding problem is formally stated as follows. Given a target tertiary structure T_{3D} defined by 3D atomic coordinates from a PDB file, find a nucleotide sequence $S = \{s_1, s_2, \dots, s_n\}$ with $s_i \in \{A, U, G, C\}$ such that the structure $F(S)$ predicted by an independent folding model satisfies

$$\text{RMSD}(F(S), T_{3D}) \leq \epsilon \tag{1}$$

for some tolerance ϵ .

The secondary-structure variant of the problem was shown to be NP-hard (Lyngsø et al., 1999). The tertiary-structure variant is at least as hard, and is additionally bottlenecked by the inaccuracy

¹We use “isomorphic” in the operational sense common in systems biology and cybernetics — a structure-preserving correspondence between functional roles — not in the strict category-theoretic sense of a bijective, composition-preserving mapping. Section 3 states the correspondence; Section 7.2 is explicit about where it does not hold.

of available RNA structure predictors — the oracle F is itself a learned model with non-trivial error.

Four classes of methods have been explored:

1. **Deterministic / thermodynamic** — ViennaRNA (Lorenz et al., 2011), NUPACK (Zadeh et al., 2011). Dynamic programming over minimum-free-energy secondary structures. Fast but limited to secondary structure; cannot handle complex tertiary contacts.
2. **Stochastic / metaheuristic** — Monte Carlo inverse folding (Cazenave and Touzani, 2024), simulated annealing, genetic algorithms. Explore the sequence space broadly but converge slowly. BeeRNA (Mlaweh et al., 2025) is the most recent example, using an Artificial Bee Colony with a two-stage fitness (base-pair distance filter + RhoFold RMSD).
3. **Deep learning, GNN-based** — gRNAde (Joshi et al., 2025). E(3)-equivariant graph neural network with autoregressive decoding. Achieves 56.80% native sequence recovery on standard splits, with sub-second inference. Requires multi-day GPU training and large structural datasets.
4. **Deep learning, diffusion-based** — RiboDiffusion (Huang et al., 2024), RISOtTo (Bibekar et al., 2025). Generative diffusion models conditioned on tertiary structure. Currently the strongest benchmark numbers (RiboDiffusion: 58.96% recovery on Seq-0.8 splits) but the highest compute cost.

The published benchmark numbers move on a timescale of weeks. As of late 2025, representative numbers on the 14-structure Das et al. (2010) benchmark are: BeeRNA 12.02 Å RMSD, gRNAde 14.63 Å, RiboDiffusion 10.31 Å (Mlaweh et al., 2025, Table 4). On RNASolo short RNAs (<30 nt), BeeRNA reports 2.50 Å vs. gRNAde’s 9.33 Å — a substantial gap that reflects BeeRNA’s strength in the short-RNA regime rather than a general superiority.

2.2 Bittensor subnets

Bittensor (Rao, 2020) is a decentralized network of subnetworks (“subnets”), each coordinating a specific useful task. A subnet has three roles: **miners** produce outputs, **validators** score those outputs, and the **chain** (Yuma consensus) aggregates validator scores into a global ranking that determines how newly minted TAO tokens are distributed to miners and validators.

Subnet registration is gated by a dynamic TAO lock cost — historically in the thousands of TAO — which is a real capital requirement separate from the technical build. Once registered, a subnet operates continuously: miners submit outputs each block, validators score them, and incentives flow accordingly.

Existing subnets relevant to molecular design include SN25 (Mainframe) and SN68 (Nova), discussed further in Section 2.4. To our knowledge, as of mid-2026, no Bittensor subnet targets RNA inverse folding specifically. The closest analogues are protein-structure-prediction subnets, which use AlphaFold-style scoring rather than the refold-and-compare protocol we describe here.

2.3 The evaluation problem

A subtle but critical issue in RNA inverse folding is that the generator and the oracle cannot be the same model. If the generator is trained against RhoFold’s predicted structures, then using RhoFold as the oracle leaks the generator’s training signal into the evaluation. The BeeRNA and gRNAde papers address this by using RhoFold as an independent oracle that the generator did not directly

train against — but this is a social rather than formal guarantee. In a decentralized setting, the same problem recurs with economic stakes: a miner could in principle fine-tune a generator against the validator’s oracle and game the metric.

The protocol’s defense is twofold. First, the validator’s target pool is private and periodically rotated, so the miner cannot pre-compute oracle-fitted solutions. Second, the diversity bonus rewards structurally valid but *different* submissions, which an oracle-fitted generator would struggle to produce.

This distinction between *search-time* oracle use and *training-time* oracle leakage matters for reading the Phase 1 pipeline (Section 4.3) correctly. BeeRNA’s use of RhoFold RMSD as a search-time fitness function — scoring candidate sequences against one already-known target during optimization, then discarding that experience once the target changes — is not the leakage case described above. Leakage refers to training a parametric model (e.g., a GNN) on a corpus of RhoFold-predicted structures used as pseudo-ground-truth, so the model’s weights implicitly absorb RhoFold’s biases before it ever sees the evaluation target. A metaheuristic that queries the oracle once per target and retains no weights between targets cannot accumulate that kind of cross-target bias; it is exactly the workflow the protocol expects every miner to run locally, regardless of which generator they choose. The leakage risk applies to *trained* generators (gRNAd, RiboDiffusion, RISoTTo) if their training corpus was itself RhoFold-labeled, not to search-based ones like BeeRNA.

2.4 Related decentralized-science models

The Ribosome Network is not the only attempt to decentralize biological research. Outside Bittensor, a separate ecosystem of “DeSci” (decentralized science) projects — Molecule, VitaDAO, and the broader BIO Protocol network of bioDAOs — decentralizes the *funding and IP ownership* of biotech research: contributors stake or donate tokens toward a specific research program (e.g., VitaDAO’s longevity work, or the VitaDAO-funded Vita-RNA program on mRNA therapeutics), and receive fractional rights to resulting intellectual property.² This is a governance and financing model, not a compute or evaluation protocol: it says nothing about how a specific computational task, such as RNA inverse folding, should be scored or incentivized on an ongoing basis.

The Ribosome Network is closer in kind to Bittensor’s existing science subnets — SN25 (Mainframe, protein docking and molecular dynamics) and SN68 (Nova, compound screening against therapeutic protein targets) — which decentralize *continuous computational competition* rather than funding. Both are active as of mid-2026, giving the Ribosome Network two adjacent subnets to plausibly interoperate with (Section 3, composability) and existing templates for miner/validator economics. Neither, to our knowledge, targets RNA design specifically.

3 The Ribosome Network: An Operational Isomorphism

We now state the paper’s central claim formally. The protocol design of an RNA inverse-folding subnet is *operationally isomorphic* to ribosomal biology, meaning that each protocol component corresponds to a biological component with the same functional role, and the parameters of the protocol component can be set by reference to the biological analogue.

The isomorphism has three consequences that we develop in the remainder of the paper:

1. **Parameter defaults.** Parameters that would otherwise be set arbitrarily — the diversity bonus

²See <https://www.bio.xyz> and <https://www.vitadiao.com> for current program listings; accessed August 2026.

Table 1: The ribosome-network isomorphism

Biological ribosome	Ribosome component	Network	Functional role
mRNA codons	Target 3D structure		The design specification
Ribosome (large + small subunit)	Miner		Synthesizes a candidate sequence from the template
Aminoacyl-tRNA synthetases	Generator (BeeRNA / etc.)	model / gRNAd	The mechanism by which the miner produces candidates
Molecular chaperones (GroEL/ES, etc.)	Validator		Independently folds and checks the candidate
Proteasome / quality control	Slashing, low scores		Degrades failed folds
Codon degeneracy	Sequence-structure degeneracy	de-	Many sequences map to the same structure
Immune self/non-self recognition	Sybil / duplicate detection		Distinguishes genuine from copy
Transient mRNA protection (5' cap, poly-A)	Commit-reveal scheme		Protects the message from copying during synthesis
mRNA turnover / decay	Target-pool rotation		Prevents staleness and overfitting to fixed templates
Evolutionary selection for diversity	Diversity bonus		Maintains exploration of the sequence space
Mutation rate	Generator stochasticity		Balances exploration and exploitation
Ribosomal stalling (ribosome quality control)	Timeout / failed-refold penalty		Handles sequences that cannot be evaluated
Translation fidelity	Validator scoring precision		The accuracy of the structural comparison

weight, the target-pool rotation cadence, the duplicate-detection threshold — acquire biological analogues that suggest principled defaults. We do not claim these defaults are optimal; we claim they are *non-arbitrary*, which is a meaningful improvement over free hyperparameter choice.

2. **Failure-mode analysis.** Biological ribosomes have evolved defenses against specific failure modes (ribosomal stalling, misfolding, viral mRNA competition). The corresponding protocol defenses (timeouts, slashing, target rotation) can be designed by analogy.
3. **Composability.** The ribosome does not operate in isolation — it interacts with the transcriptome, the chaperone machinery, and the degradation pathway. The Ribosome Network’s components are designed to be similarly composable, enabling integration with adjacent subnets (SN25 for docking, SN68 for ligand screening).

We emphasize that the isomorphism is *operational*, not *literal*. We are not claiming that the subnet is a ribosome, only that the protocol’s functional structure mirrors the ribosome’s functional structure. The metaphor’s value is heuristic: it tells us where to look when designing or debugging a component.

4 Architecture

4.1 Roles

The network has three roles.

Miners hold a generator model (any of the methods in Section 2.1). Each round, they receive a target structure, produce one or more candidate sequences, and submit them via the commit-reveal protocol (Section 5.1). Miners are scored by validators and earn TAO in proportion to their scores.

Validators hold a private pool of target structures (Section 5.4) and an independent structure predictor (RhoFold by default). Each round, they distribute one target from their pool to all miners, collect commits, collect reveals, refold each revealed sequence via their predictor, and score the structural match against the target. Validators are scored by the chain based on the agreement of their scores with the consensus validator set.

The chain (Yuma consensus) aggregates validator scores into miner rankings and emits token incentives. We do not modify the chain layer; the Ribosome Network is a subnet specification, not a chain modification.

4.2 Round structure

A round proceeds in five phases (Figure 1):

1. **Target distribution** ($t = 0$). Each validator selects a target from their pool and broadcasts it to all miners. The target’s identity is withheld from non-participating parties.
2. **Commit window** ($0 < t \leq t_c$). Miners compute candidate sequences and submit commit hashes (Section 5.1).
3. **Reveal window** ($t_c < t \leq t_r$). Miners reveal their sequences and salts. The validator verifies each reveal against its commit hash.

4. **Scoring** ($t_r < t \leq t_s$). The validator refolds each revealed sequence via RhoFold, computes TM-score and RMSD against the target, applies duplicate detection (Section 5.2) and the diversity bonus (Section 5.3), and produces a final score per submission.
5. **Attestation** ($t_s < t \leq t_a$). The validator submits scores to the chain. Scores are revealed in delayed batches (Section 5.5) to prevent oracle attacks on the private target pool.

Round length is configurable. A reasonable starting point is a 15-minute round, which gives miners enough time to run a generator (BeeRNA’s CPU runtime is ~ 3 minutes for short RNAs) without making the network sluggish.

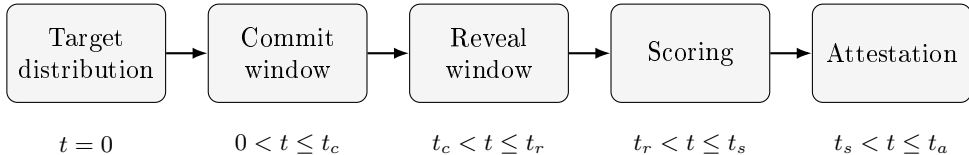


Figure 1: The five phases of a Ribosome Network round.

4.3 The Phase 1 pipeline

The Phase 1 pipeline — the standalone ML implementation that miners would run locally — is implemented and runs end-to-end with stub oracles. Its components:

- **Generator:** BeeRNA (Mlaweh et al., 2025), reproduced in full. The Artificial Bee Colony maintains a population of 40 candidate sequences for 40 iterations, with a two-stage fitness (ViennaRNA base-pair distance filter, then RhoFold RMSD). Watson–Crick-aware initialization, GC-content balancing, annealed mutation schedule.
- **Predictor:** RhoFold adapter (shells out to a local RhoFold checkout) with a deterministic stub fallback for plumbing tests.
- **Metrics:** RMSD over backbone P/C4'/N1/N9 atoms (BeeRNA’s definition), TM-score via US-align (Zhang et al., 2022) (with a pure-Python Kabsch-based fallback), GDT-TS (Zemla, 2003), base-pair distance, native sequence recovery (secondary), GC content.
- **Data:** 14-structure Das et al. (2010) benchmark loader (with RCSB download), RNASolo loader, RFAM loader, gRNAde split mirror.

The pipeline produces JSON/CSV/Markdown reports with per-target and aggregate metrics. We have verified that it runs end-to-end on real PDB structures (e.g., 1F27) with the stub oracle, but we do not report benchmark numbers because the only honest benchmark requires RhoFold integration that we have not yet validated.

4.4 Generators are pluggable

A key design property is that the generator is pluggable. A miner may run any generator they choose — BeeRNA (training-free, CPU), gRNAde (GNN, GPU), RiboDiffusion (diffusion, GPU), or a novel method of their own. The protocol scores the *output*, not the method.

This is a deliberate departure from the academic benchmarking convention, where each method is evaluated separately on a fixed test set. In the Ribosome Network, methods compete directly: a miner running a weak generator on cheap hardware can be outscored by a miner running a strong generator on expensive hardware, and the token incentive flows accordingly. The network thus performs continuous selection on generator quality, much as biological evolution performs continuous selection on synthetase efficiency.

The downside is that comparison with published benchmark numbers becomes indirect. A miner’s score reflects both their generator’s quality and their hardware’s speed; disentangling the two requires auxiliary data (which the protocol could expose via metadata).

5 Anti-gaming

The protocol’s economic security rests on making honest participation — submitting genuinely novel candidate sequences that fold correctly — the profit-maximizing strategy. We describe five anti-gaming mechanisms, each with a biological analogue.

5.1 Commit-reveal (transient mRNA protection)

Mechanism. Miners submit a hash $h = \text{SHA256}(\text{miner_uid} \parallel S \parallel \text{salt})$ during the commit window. The salt is secret until reveal. After the commit window closes, miners reveal (S, salt) ; the validator verifies $h = \text{SHA256}(\text{miner_uid} \parallel S \parallel \text{salt})$.

Defense. Prevents a miner from copying another miner’s sequence within a round. To reproduce another miner’s commit hash, the copier would need to know the salt, which is secret until reveal — by which point the commit window has closed.

Biological analogue. Transient mRNA protection mechanisms (5’ cap, poly-A tail, RNA-binding proteins) protect the mRNA from degradation or interference during translation. The commit window is the “translation” phase; the reveal is the “completed protein” phase. Without protection, a competing ribosome (or in our case, a Sybil miner) could hijack the partial work.

Limitation. Commit-reveal does not prevent *cross-round* copying. A miner could observe another miner’s revealed sequence in round n and submit a slight variant in round $n + 1$. This is handled by the duplicate detector (Section 5.2) and the target-pool rotation (Section 5.4).

5.2 Duplicate detection (immune self/non-self recognition)

Mechanism. For each pair of submissions (S_i, S_j) in a round, compute the sequence identity $\text{id}(S_i, S_j) \in [0, 1]$. If $\text{id}(S_i, S_j) \geq \theta_{\text{dup}}$, flag the lower-scoring submission as a duplicate and exclude it from the diversity bonus.

We provide two identity functions: per-position Hamming identity (fast, length-sensitive) and Needleman–Wunsch global alignment identity (slower, handles insertions/deletions). The threshold θ_{dup} is a tunable parameter.

Defense. Detects Sybil miners who submit near-identical sequences from multiple identities. Also detects miners who copy another miner’s previous-round submission with minor mutations.

Biological analogue. The immune system distinguishes self from non-self via Major Histocompatibility Complex (MHC) presentation. A peptide that is too similar to a self-peptide is tolerated; a sufficiently different peptide triggers a response. The duplicate threshold θ_{dup} plays the role of

the tolerance threshold.

Open parameter. The threshold $\theta_{\text{dup}} = 0.85$ is our default, but this is a guess. Too low and the protocol kills legitimate sequence diversity (many valid sequences are 85% identical); too high and Sybils slip through. Empirical tuning against real submission data is required. We do not invent an answer.

5.3 Diversity bonus (evolutionary pressure)

Mechanism. For each non-duplicate submission S_i in a round, compute the diversity bonus:

$$\text{bonus}_i = w_{\text{div}} \cdot \frac{1}{|R| - 1} \sum_{j \neq i} (1 - \text{id}(S_i, S_j)) \quad (2)$$

where R is the set of non-duplicate submissions in the round and w_{div} is the diversity weight. The final score is:

$$\text{score}_i = (1 - w_{\text{div}}) \cdot \text{structural_score}_i + \text{bonus}_i \quad (3)$$

where $\text{structural_score}_i$ is the TM-score (or 1 minus the normalized RMSD).

Defense. Incentivizes miners to explore the sequence space rather than converge on a single “best” sequence. Without this bonus, the equilibrium strategy is for all miners to submit the same high-scoring sequence (which the duplicate detector would flag, but the bonus makes exploration directly profitable).

Biological analogue. Evolutionary selection maintains genetic diversity as a hedge against environmental change. A population that converges on a single genotype is fragile; a diverse population is robust. The diversity bonus instantiates this pressure economically.

Open parameter. The weight $w_{\text{div}} = 0.1$ is our default. Too low and the bonus is meaningless; too high and miners are incentivized to submit diverse *junk* (sequences that are different but do not fold correctly). The structural-score term should dominate, but the exact balance needs empirical tuning.

Subtle failure mode. The diversity bonus is itself gameable. A miner could submit many structurally plausible but non-functional sequences to collect diversity bonuses without contributing to the network’s design goal. The defense is that the structural-score term filters out non-folding sequences — but a sophisticated miner could submit sequences that *just barely* pass the structural threshold while being maximally diverse. We do not have a clean solution; this is an open problem.

5.4 Target-pool rotation (mRNA turnover)

Mechanism. Each validator holds a private pool of N_{pool} target structures (default 32). Every T_{rot} rounds (default 16), the validator re-samples the active subset from a larger base pool. The active subset is never revealed to miners; only the per-round target is revealed, and only at round start.

Defense. Prevents miners from pre-computing oracle-fitted solutions. Even if a miner extracts the active pool from a compromised validator, the pool rotates before the miner can exploit it.

Biological analogue. mRNA turnover — the continuous degradation and replacement of mRNA transcripts — prevents the cell from being locked into a fixed translation program. The rotation period T_{rot} plays the role of the mRNA half-life.

Trade-off. Faster rotation increases security but reduces miner learning. A miner that observes a target for multiple rounds can iteratively improve their generator for that target’s structural class.

Faster rotation forecloses this. The default $T_{\text{rot}} = 16$ rounds (about 4 hours at 15-minute round length) is a starting point; the optimal value depends on miner behavior, which we do not yet have data for.

A quantitative check. Taking the mRNA-turnover analogy literally, rather than qualitatively, does not obviously support this default. Genome-wide measurements in *E. coli* put the median bulk mRNA half-life at roughly 5 minutes (Bernstein et al., 2002), against a proposed 15-minute round and a $T_{\text{rot}} = 16$ -round (~ 4 -hour) rotation period. Scaled to round-units, real bacterial mRNA turnover is closer to one-third of a *single* round, not sixteen of them — the opposite direction from what we chose. We flag this rather than paper over it: the biological analogue supports the *existence* of a rotation mechanism and names the tradeoff it navigates (faster turnover trades miner-learning time for security), but the specific cadence measured in *E. coli* does not transfer numerically to a security parameter with different stakes and different failure costs. $T_{\text{rot}} = 16$ remains an engineering default pending real tuning against submission data, not a biologically derived one, and we no longer imply otherwise.

5.5 Delayed score reveals (ribosome stalling defense)

Mechanism. Validators submit scores to the chain in delayed batches of B rounds (default 4). The scores for round n are revealed only at round $n + B$.

Defense. Prevents oracle attacks on the private target pool. If scores were revealed immediately, a miner could infer which targets produce high scores and bias their future submissions toward those targets, effectively reconstructing the private pool. With delayed reveals, the miner’s information lags the validator’s pool by B rounds, by which point the pool has rotated.

Biological analogue. Ribosome stalling — the pausing of translation on problematic mRNAs — is a quality-control mechanism that prevents the production of malformed proteins. Delayed score reveals are a softer analogue: the network pauses before committing to a score, allowing time for anomalies (failed refolds, duplicate clusters, oracle failures) to be detected.

5.6 A note on incentive alignment

The claim that honest participation is profit-maximizing (Section 5) is asserted throughout this section but never formally argued. We give a first-order argument here, with the explicit caveat that it is a single-round, non-adversarial sketch, not a full mechanism-design proof.

Consider a miner choosing between two strategies in a given round: (a) submit a genuine candidate produced by running a generator against the round’s target, or (b) submit a cheap default (a fixed, previously-successful sequence, a random sequence, or a copy of another miner’s prior submission perturbed just below the duplicate threshold θ_{dup}). Under the scoring rule of Section 5.3,

$$\text{score}_i = (1 - w_{\text{div}}) \cdot \text{structural_score}_i + \text{bonus}_i,$$

a random or reused sequence scores near zero on $\text{structural_score}_i$ against a freshly rotated, privately-held target (Section 5.4) unless it happens to resemble the target’s fold class by chance — which is exactly what target-pool privacy and rotation are designed to make unlikely. Because $w_{\text{div}} = 0.1$ by default, the diversity bonus cannot compensate for a near-zero structural score: even a maximally diverse but structurally worthless submission earns at most $w_{\text{div}} \approx 0.1$, against a genuine, well-folded submission earning close to 1. Strategy (b) is therefore dominated in expectation whenever the miner’s generator does better than chance at the refold-and-compare metric — precisely the condition under which running a generator is worthwhile in the first place.

This argument does not cover more sophisticated strategies. A miner who optimizes specifically to game the diversity bonus (Section 5.3, “subtle failure mode”), or a coordinated Sybil ring that splits submissions just under θ_{dup} , is not addressed by it, and neither is validator collusion or repeated-game effects across rounds (Section 6.3). We surface this as a gap in the paper’s own argument rather than let the biological framing stand in for a security proof it has not earned.

6 Evaluation protocol

6.1 The refold-and-compare pipeline

The evaluation protocol follows BeeRNA and RiboDiffusion (Mlaweh et al., 2025; Huang et al., 2024):

1. Take the designed sequence S .
2. Refold it independently using RhoFold (a model the generator did not train against, or at least was not fine-tuned against on the validator’s target pool).
3. Compare the predicted fold $F(S)$ to the target T_{3D} via TM-score and RMSD.
4. Report native sequence recovery as a secondary metric, for comparability with older papers.

This protocol is more expensive than native-sequence-recovery-only evaluation (each scoring requires a RhoFold forward pass) but is methodologically correct: it measures what we actually care about (does the sequence fold to the target structure?) rather than a proxy that the field has outgrown.

6.2 What we would measure

A full empirical evaluation of the Ribosome Network would measure:

- **Generator quality:** TM-score and RMSD distributions for each generator class (BeeRNA, gRNade, RiboDiffusion) on a fixed target pool.
- **Network-level diversity:** the effective number of distinct structural solutions the network produces per round, measured as the entropy of the structural-similarity distribution of revealed sequences.
- **Sybil resistance:** the score distribution under adversarial conditions (a fraction of miners running Sybil strategies).
- **Diversity-bonus sensitivity:** network-level diversity as a function of w_{div} , sweeping from 0 to 0.5.
- **Rotation-cadence sensitivity:** Sybil-reconstruction success rate as a function of T_{rot} .

We do not report these numbers here. The Phase 1 pipeline is implemented and runs end-to-end with stub oracles, but the only honest benchmark requires RhoFold to be installed and that integration is not yet validated. We flag this as a limitation rather than fabricating preliminary numbers.

6.3 Honest gaps

We enumerate the open questions that we have not resolved, with explicit acknowledgment that we will not invent answers:

1. **RhoFold VRAM requirements.** Not independently verified. The BeeRNA paper reports running RhoFold on CPU; the adapter here will use whatever device RhoFold itself picks up. Check the RhoFold repo before planning a GPU run.
2. **Sybil-detection threshold.** $\theta_{\text{dup}} = 0.85$ is a default, not a tuned value. Empirical tuning against real submission data is required.
3. **Diversity-bonus weight.** $w_{\text{div}} = 0.1$ is a default. The optimal value depends on the structural-score distribution, which depends on the generator mix, which depends on miner participation — a circular dependency that can only be resolved by running the network.
4. **Target-pool rotation cadence.** $T_{\text{rot}} = 16$ rounds is a default. The optimal value depends on miner learning rates, which we do not have data for. A literal reading of the mRNA-turnover analogue in fact suggests a much faster cadence (Section 5.4); we flag rather than resolve this tension.
5. **The incentive-alignment argument is informal.** Section 5.6 shows that one cheap strategy is dominated under a single-round, non-adversarial reading of the scoring rule, but this is not a full game-theoretic equilibrium analysis. Adaptive Sybil strategies, validator collusion, and repeated-game effects across rounds remain open.
6. **Cross-subnet integration.** The plausibility of integrating with SN25 (Mainframe — protein docking) for RNA-protein complex modeling, and SN68 (Nova — compound screening) for riboswitch-ligand pairing, is suggested by the composability of the underlying tasks but has not been validated.
7. **Niche focus.** A narrower, genuinely underexplored niche — e.g., antimicrobial-resistance-relevant regulatory RNA design — may hold up under a full literature review as a higher-value target than general RNA inverse folding. Our initial search found no direct prior art, but this was not an exhaustive review.

7 Discussion

7.1 What the isomorphism buys

The ribosome-network isomorphism is not necessary for the protocol to function — the same protocol could be specified without any biological reference. The isomorphism’s value is heuristic: it tells us where to look when designing or debugging a component.

When designing the diversity bonus, for instance, the question “what weight should we use?” is hard to answer from first principles. The biological analogue — “what mutation rate does evolution settle on?” — does not directly answer the question, but it reframes it: the diversity bonus is not a free parameter but an exploration-exploitation tradeoff, and the relevant comparison is not “0.1 vs. 0.2” but “how does the structural-score distribution shape the optimal exploration pressure?” This reframing does not substitute for empirical tuning, but it makes the tuning target clearer.

Similarly, when debugging a target-rotation failure (e.g., miners somehow predicting the next round’s target), the biological analogue — “how does the cell detect mRNA leakage?” — suggests looking at the reveal mechanism rather than the rotation mechanism. The bug is likely in the target distribution, not the pool sampling.

7.2 Where the isomorphism breaks

The isomorphism is not perfect. Three points of breakdown are worth flagging.

First, biological ribosomes do not compete. A ribosome does not earn more resources for translating more proteins; it simply translates until it is degraded. Miners in the Ribosome Network *do* compete, and competition creates incentives that have no biological analogue (e.g., the incentive to copy another miner’s sequence, which is what commit-reveal defends against). The isomorphism maps the *functional structure* of ribosomal biology onto the protocol, but the *economic structure* is novel.

Second, biological chaperones are not adversarial. A chaperone protein assists folding; it does not score the result. Validators in the Ribosome Network *do* score the result, and a malicious validator could under-report scores to suppress a competitor. The biological analogue does not help here; the defense is the standard Bittensor mechanism of validator-set consensus.

Third, biological evolution is not goal-directed. Evolution selects for fitness, but fitness is a moving target. The Ribosome Network has a fixed goal (correctly folded structures), which makes the optimization problem cleaner but also makes the diversity bonus less clearly aligned with the objective. In biology, diversity is valuable because the future environment is unknown; in the Ribosome Network, the objective is known, and diversity is only valuable insofar as it improves the network’s collective ability to find correct folds. This is a weaker justification, and the diversity bonus should be tuned more conservatively than biological mutation rates would suggest.

7.3 Comparison to centralized benchmarks

A natural objection: if the goal is to advance RNA inverse folding, why not just run a centralized benchmark server? The Ribosome Network’s advantages over a centralized server are threefold.

First, continuous operation. A centralized benchmark server runs when its operators choose to run it; the Ribosome Network runs continuously, with incentives for miners to participate even when no academic paper is being written.

Second, methodological pluralism. A centralized benchmark server typically evaluates one method at a time, on a fixed test set, with a fixed metric. The Ribosome Network evaluates multiple methods simultaneously, on a rotating target pool, with a composite score (structural + diversity). This is closer to how the field actually progresses.

Third, anti-monopoly. A centralized benchmark server is controlled by whoever runs it; the Ribosome Network is governed by the chain. This matters because benchmark servers have historically been used to privilege the methods of their operators (intentionally or not — the choice of test set and metric is itself a methodological choice).

The disadvantage is cost. A centralized benchmark server is cheap to operate; a Bittensor subnet requires continuous token issuance to incentivize participation. Whether the continuous-operation and methodological-pluralism benefits justify the cost is an empirical question that depends on the subnet’s actual research output.

7.4 Limitations

We are explicit about the paper’s limitations.

The Phase 1 pipeline is implemented but not benchmarked. We have verified that the pipeline runs end-to-end with stub oracles, but we have not validated RhoFold integration. The numbers in Section 2.1 are from the cited papers, not from our own runs.

The Phase 2 subnet layer is interface-only. The protocol is specified, the anti-gaming mechanisms are described, and the implementation stubs are in place, but the subnet is not registered on Bittensor. Registration is a separate problem with a real capital cost (thousands of TAO at current prices) that we do not address.

The isomorphism is illustrative, not provably optimal. We argue that the ribosome-network isomorphism provides principled defaults, but we do not prove that those defaults are optimal. They are starting points for empirical tuning, not final answers.

The anti-gaming mechanisms are individually sound but their composition is untested. Each mechanism (commit-reveal, duplicate detection, diversity bonus, target rotation, delayed reveals) has a clear defense and a clear biological analogue, but their *interaction* — e.g., a Sybil miner adapting to the duplicate detector by submitting sequences just below the threshold — has not been validated. We expect that real deployment will reveal attack vectors we have not anticipated.

8 Conclusion

We have proposed the Ribosome Network, a decentralized RNA inverse-folding subnet whose protocol design is operationally isomorphic to ribosomal biology. The isomorphism is not merely metaphorical: it provides principled defaults for economic parameters (diversity bonus, target rotation, duplicate threshold) that would otherwise be set arbitrarily, and it suggests failure-mode defenses by analogy with biological ribosomes.

The Phase 1 pipeline — a standalone ML implementation of BeeRNA with RhoFold refold-and-compare scoring — is implemented and runs end-to-end with stub oracles. The Phase 2 subnet layer is specified as interface stubs with clear anti-gaming mechanisms. We are explicit about which components are implemented, which are proposed, and which empirical parameters remain open.

The paper’s contribution is conceptual (the isomorphism), methodological (the protocol specification), and infrastructural (the Phase 1 pipeline). It is not empirical — we do not report benchmark numbers, because the only honest benchmark requires RhoFold integration that we have not yet validated. We flag this as a limitation rather than fabricating preliminary results.

The broader hope is that the ribosome-network framing generalizes. Other decentralized-science subnets — protein design, small-molecule screening, materials discovery — may have analogous biological isomorphisms (ribosome for design, immune system for screening, metabolic network for materials) that provide principled defaults for their own protocol parameters. We do not develop this generalization here, but we note it as a direction for future work.

References

Adamczyk, B., Antczak, M., and Szachniuk, M. (2022). RNAsolo: a repository of cleaned PDB-derived RNA 3D structures. *Bioinformatics*, 38(14):3668–3670.

- Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242.
- Bernstein, J. A., Khodursky, A. B., Lin, P.-H., Lin-Chao, S., and Cohen, S. N. (2002). Global analysis of mRNA decay and abundance in *Escherichia coli* at single-gene resolution using two-color fluorescent DNA microarrays. *Proceedings of the National Academy of Sciences*, 99(15):9697–9702.
- Bibekar, P., Krapp, L. F., and Dal Peraro, M. (2025). Context-aware geometric deep learning for RNA sequence design. *bioRxiv*.
- Cazenave, T. and Touzani, H. (2024). Monte Carlo Inverse RNA Folding. In *Methods in Molecular Biology*, volume 2847, pages 205–215. Humana Press.
- Das, R., Karanicolas, J., and Baker, D. (2010). Atomic accuracy in predicting and designing noncanonical RNA structure. *Nature Methods*, 7:291–294.
- Huang, H., Lin, Z., He, D., Hong, L., and Li, Y. (2024). RiboDiffusion: tertiary structure-based RNA inverse folding with generative diffusion models. *Bioinformatics*, 40(Supplement_1):i347–i356.
- Joshi, C. K., Jamasb, A. R., Viñas, R., Harris, C., Mathis, S. V., Morehead, A., Anand, R., and Liò, P. (2025). gRNade: Geometric deep learning for 3D RNA inverse design. *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Lorenz, R., Bernhart, S. H., Höner zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P. F., and Hofacker, I. L. (2011). ViennaRNA Package 2.0. *Algorithms for Molecular Biology*, 6(1):26.
- Lyngsø, R. B., Zuker, M., and Pedersen, C. N. S. (1999). Fast evaluation of internal loops in RNA secondary structure prediction. *Bioinformatics*, 15(6):440–445.
- Mlaweh, M., Cazenave, T., and Alaya, I. (2025). BeeRNA: tertiary structure-based RNA inverse folding using Artificial Bee Colony. *arXiv:2511.21781*.
- Rao, Y. (2020). Bittensor: A Peer-to-Peer Intelligence Market. *Bittensor whitepaper*. <https://www.bittensor.com/whitepaper>.
- Shen, T., Hu, Z., Sun, S., Liu, D., Wong, F., Wang, J., Chen, J., Wang, Y., Hong, L., Xiao, J., et al. (2024). Accurate RNA 3D structure prediction using a language model-based deep learning approach. *Nature Methods*, 21:2287–2298.
- Tan, C., Zhang, Y., Gao, Z., Cao, H., Li, S., Ma, S., and Blanchette, M. (2025). R3Design: Deep tertiary structure-based RNA sequence design and beyond. *Briefings in Bioinformatics*, 26(1):bbae682.
- Zadeh, J. N., Steenberg, C. D., Bois, J. S., Wolfe, B. R., Pierce, M. B., Khan, A. R., Dirks, R. M., and Pierce, N. A. (2011). NUPACK: Analysis and design of nucleic acid systems. *Journal of Computational Chemistry*, 32(1):170–173.
- Zemla, A. (2003). LGA: A method for finding 3D similarities in protein structures. *Nucleic Acids Research*, 31(13):3370–3374.
- Zhang, Y. and Skolnick, J. (2004). Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710.

Zhang, C., Shine, M., Pyle, A. M., and Zhang, Y. (2022). US-align: Universal structure alignments of proteins, nucleic acids, and macromolecular complexes. *bioRxiv*.