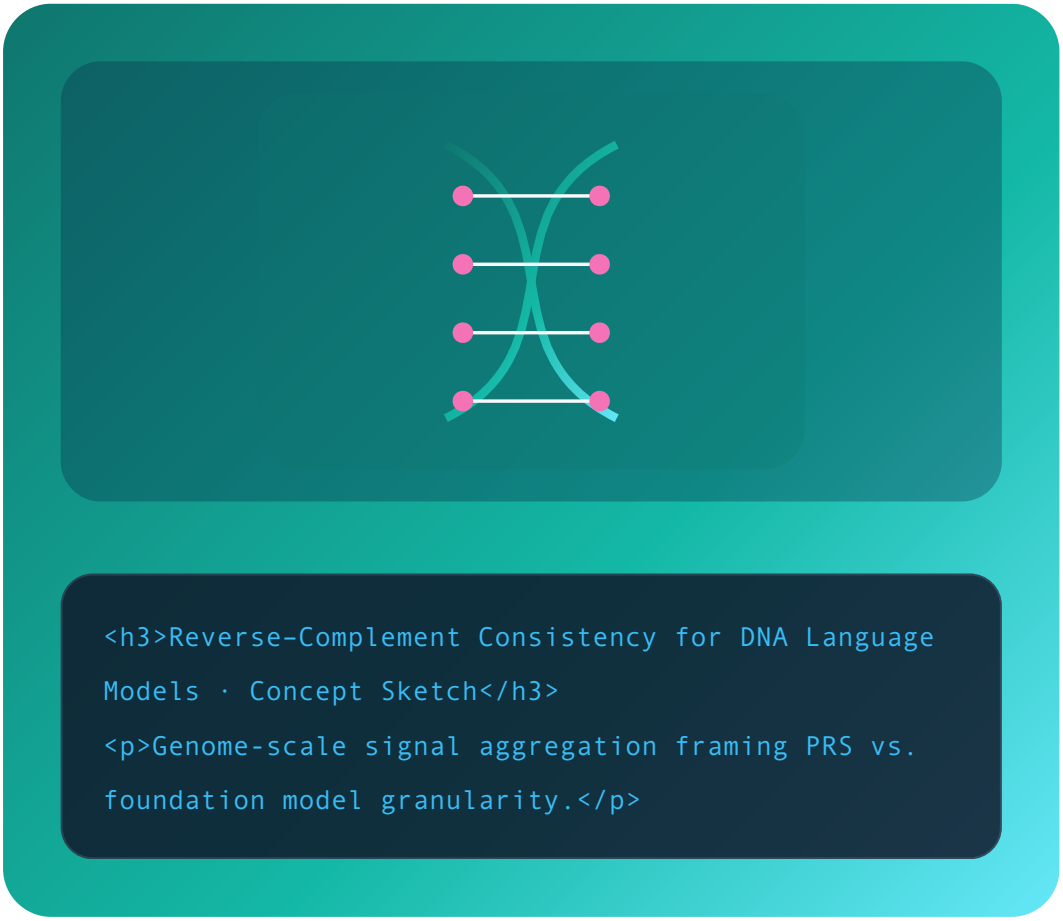

Reverse–Complement Consistency for DNA Language Models

Authors: Mingqian Ma

Year: 2025

Venue: arXiv preprint

1. Classification

- **Domain Category:**
  - **Genomics FM.** The paper addresses a critical failure mode in DNA language models (DNA LMs) where models produce inconsistent predictions for a sequence and its reverse complement, despite biological equivalence.
- **FM Usage Type:**
  - **Application of existing FM (fine-tuning method).** The paper proposes a fine-tuning objective (RCCR) that can be applied to any pretrained DNA LM backbone (Nucleotide Transformer, DNABERT-2, HyenaDNA) without modifying the architecture.
- **Key Modalities:**
  - Single-modality DNA sequence (nucleotide-level modeling with various tokenization schemes: BPE, k-mers, character-level).

---

2. Executive Summary

This paper addresses a pervasive but under-measured problem in DNA language models: sensitivity to input orientation. DNA sequences have a fundamental reverse-complement (RC) symmetry—a sequence and its RC carry identical biological meaning for many tasks—yet state-of-the-art DNA LMs frequently produce inconsistent predictions for x and RC(x), undermining reliability and interpretability. The authors introduce Reverse-Complement Consistency Regularization (RCCR), a simple, model-agnostic fine-tuning objective that directly penalizes divergence between a model's prediction on a sequence and the task-aligned prediction on its reverse complement. RCCR works across diverse task types (sequence-level classification, scalar regression, bin-wise profile prediction) via a task-aware alignment operator and appropriate divergence metrics (symmetric KL for classification, squared error/Poisson KL for regression). Theoretically, RCCR guarantees that symmetrization (test-time averaging)

is risk non-increasing, and with RC-symmetric labels, global minimizers are RC-consistent. Empirically, across three heterogeneous backbones (Nucleotide Transformer v2, DNABERT-2, HyenaDNA) and diverse genomic tasks, RCCR substantially improves RC robustness (lower flip rates, higher correlation) while maintaining or improving task accuracy compared to baselines like RC data augmentation and test-time averaging. Unlike test-time averaging, RCCR produces a single, intrinsically robust model without doubling inference cost. This work demonstrates how to encode biological priors directly into the learning objective, providing a practical recipe for improving DNA LM reliability without architectural changes.

---

3. Problem Setup and Motivation

- **Scientific / practical problem**
  - DNA language models are increasingly used for genomic prediction tasks, but they often fail to respect the fundamental RC symmetry of DNA.
  - Many tasks are RC-invariant (e.g., promoter classification: a sequence and its RC should have the same label) or RC-equivariant (e.g., profile prediction: outputs should be aligned by reversing and complementing).
  - Empirically, reversing and complementing a sequence can alter a model's output even when ground truth is unchanged or predictably transformed, degrading reliability and complicating interpretation.
- **Why this is hard**
  - **Standard fine-tuning doesn't enforce consistency:**
    - Models are trained to minimize task loss but not explicitly penalized for orientation sensitivity.

        - Even with RC data augmentation (training on both x and RC(x) with same labels), models can learn orientation-dependent features that don't generalize.
    - **Test-time averaging is inefficient:**
        - Averaging predictions on x and RC(x) at inference doubles compute cost and doesn't fix the underlying model.
    - **Architectural approaches have limitations:**
        - RC-equivariant architectures (e.g., Caduceus) hardcode symmetry but may reduce flexibility and aren't applicable to widely used pretrained backbones.
        - Some tasks (e.g., strand-specific prediction) explicitly violate RC symmetry, so hardcoding it is inappropriate.
    - **Lack of standardized evaluation:**
        - Previous work didn't systematically measure RC consistency, making it hard to compare methods and track progress.

---

4. Data and Modalities

- **Datasets used**
    - **Nucleotide Transformer Benchmark:**
        - 18 sequence-level classification tasks (enhancers, promoters, histone modifications, splice sites).
        - Sequence lengths: 200-1000 bp.
    - **Genomics Long-Range Benchmark (LRB):**
        - Bulk RNA expression prediction (sequence-level regression): 4,096 bp sequences, 218 cell types.
        - CAGE profile prediction (bin-wise regression): 4,096 bp sequences, 128-bp bins.
    - **Strand classification (negative control):**
        - 1,024 bp sequences centered on transcription start sites, predicting "+" vs. "-" strand (explicitly RC-dependent task).

- **Modalities**
  - Single modality: **DNA sequence** at nucleotide resolution.
  - Outputs vary by task:
    - Binary/multi-class labels (classification).
    - Scalar values (regression).
    - Position-wise profiles (bin-wise regression).
- **Preprocessing / representation**
  - **Backbone-specific tokenization:**
    - Nucleotide Transformer: 6-mer tokenization.
    - DNABERT-2: BPE tokenization (4,096 tokens).
    - HyenaDNA: Character-level (single nucleotide tokens).
  - **Task-specific alignment:**
    - Sequence-level: identity alignment (RC-invariant).
    - Profile-level: reverse and swap strand channels (RC-equivariant).

---

5. Model / Foundation Model

- **Model Type**
  - **Not a new foundation model.** RCCR is a fine-tuning objective applicable to any pretrained DNA LM backbone.
  - Tested on three backbones:
    - **Nucleotide Transformer v2:** BERT-style encoder with 6-mer tokenization.
    - **DNABERT-2:** BERT-style encoder with BPE tokenization.
    - **HyenaDNA:** Hyena operator-based decoder with character-level tokens.

- **Is it a new FM or an existing one?**
  - **Fine-tuning method for existing FMs.** RCCR modifies the training objective but does not change the backbone architecture or pretraining procedure.
- **Key components and innovations**
  - **Reverse-Complement Consistency Regularization (RCCR):**
    - Augments task loss with a consistency term: $L_{RCCR} = E[\ell(Y, f(X))] + \lambda E[D(\phi(f(X)), \phi(\Pi f(RC(X))))]$
    - Where:
      - $f(X)$ is model output on sequence X.
      - Π is task-aware alignment operator (identity for classification, reverse+swap for profiles).
      - ϕ is link function (softmax for classification, identity for regression).
      - D is divergence (symmetric KL for classification, squared error/Poisson KL for regression).
      - λ is regularization strength.
  - **Task-aware alignment operator Π:**
    - For sequence-level tasks: identity (RC-invariant).
    - For profile tasks: reverses positional axis and swaps strand channels if present.
  - **Theoretical guarantees:**
    - Symmetrization (test-time averaging) is risk non-increasing under RCCR.
    - With RC-symmetric labels and strictly convex loss, global minimizers are RC-consistent.
    - Symmetric KL penalty controls Jensen-Shannon divergence and is locally quadratic in logit space (stable gradients).
- **Training setup**
  - **Fine-tuning:** Apply RCCR during task-specific fine-tuning of pretrained backbones.

- **Hyperparameters:**
  - Regularization strength λ: tuned per task (typically 0.1-0.3).
  - Temperature T=2.0 for softmax in symmetric KL.
  - Standard AdamW optimizer with learning rate 2×10^-4.
- **Evaluation metrics:**
  - Task metrics: AUPRC, MCC, RMSE, Spearman correlation.
  - RC consistency metrics: SFR (sequence flip rate), RC-Corr (correlation between x and RC(x) predictions).

---

6. Multimodal / Integration Aspects (If Applicable)

- **Not applicable.** RCCR is designed for unimodal DNA sequence modeling. The consistency principle could potentially extend to other biological symmetries or multimodal settings, but this is not explored in the paper.

---

7. Experiments and Results

Main findings

- **RCCR improves RC robustness across all backbones:**
  - Consistently reduces SFR (fewer prediction flips) and increases RC-Corr (higher alignment) compared to RC-Aug baseline.
  - Improvements are substantial: e.g., SFR drops from 0.154 to 0.156 (NT-v2) to 0.106-0.156 range, RC-Corr increases from 0.924 to 0.930-0.980 range.

- **Task performance maintained or improved:**
  - RCCR matches or outperforms RC-Aug and TTA on task metrics (AUPRC, MCC, RMSE, Spearman) across most tasks.
  - On NT benchmark: RCCR achieves best or second-best performance in nearly every category.
  - On bulk RNA regression: RCCR achieves best RMSE, R^2, and Spearman correlation.
  - On CAGE profiles: RCCR significantly outperforms baselines (RMSE: 0.2454 vs. 0.2619 for NT-v2).
- **Comparison to baselines:**
  - **RC-Aug:** RCCR achieves similar or better task performance with substantially better RC consistency (lower SFR, higher RC-Corr).
  - **TTA:** RCCR matches TTA's robustness without doubling inference cost and produces a single, intrinsically consistent model.
- **Negative control (strand classification):**
  - As expected, RCCR hurts performance on strand-specific task (AUPRC drops from 0.9054 to 0.8930 for NT-v2), confirming it should only be applied to RC-symmetric tasks.
  - RC consistency metrics show RCCR is working (reducing orientation dependence) even when it's inappropriate for the task.

Ablation studies

- **Regularization strength λ:**
  - Optimal λ is task-dependent but moderate values (0.1-0.3) consistently provide good trade-offs.
  - Too high λ can hurt task performance; too low λ provides minimal consistency improvement.

Key insights

- **RCCR encodes explicit biological prior:**
  - Unlike RC-Aug (which only exposes model to both orientations), RCCR directly penalizes disagreement, leading to better consistency.
- **Single robust model vs. inference-time fixes:**
  - RCCR produces a model that is consistent by design, unlike TTA which masks inconsistency at inference time.
- **Backbone-agnostic:**
  - Works across diverse architectures (Transformer, Hyena) and tokenization schemes (BPE, k-mers, character-level), demonstrating generality.

---

8. Strengths and Limitations

Strengths

- **Simple and practical:**
  - Drop-in fine-tuning objective that doesn't require architectural changes.
  - Works with any pretrained DNA LM backbone.
- **Theoretically grounded:**
  - Proves that symmetrization is risk non-increasing and that global minimizers are RC-consistent under appropriate conditions.
  - Symmetric KL penalty has desirable properties (controls JS divergence, locally quadratic).
- **Comprehensive evaluation:**
  - Tests across three diverse backbones, multiple task types (classification, regression, profiles), and 20+ datasets.

- Introduces standardized RC consistency metrics (SFR, RC-Corr) for future work.

- **Maintains or improves task performance:**
  - Unlike some regularization methods, RCCR doesn't trade accuracy for consistency; it often improves both.
- **Efficient:**
  - Single model inference (no 2× cost like TTA).
  - Better explainability than TTA (model is consistent by design, not via post-processing).

Limitations

- **Only for RC-symmetric tasks:**
  - Not applicable to strand-specific tasks (e.g., replication origin prediction, strand-specific transcription).
  - Requires careful task analysis to determine if RC symmetry holds.
- **Hyperparameter tuning needed:**
  - Optimal λ varies by task and backbone, requiring validation set tuning.
- **Doesn't address pretraining:**
  - Only applies to fine-tuning; doesn't modify pretraining objectives to learn RC-consistent representations from the start.
- **Limited to DNA:**
  - Focuses on RC symmetry; doesn't address other biological symmetries (e.g., codon translation, RNA secondary structure).
- **No architectural improvements:**
  - Doesn't propose new architectures; only modifies training objective.
  - May be less parameter-efficient than architectural equivariance (e.g., Caduceus).

Open questions / future directions

- **Pretraining integration:**
  - Can RCCR principles be applied during pretraining to learn RC-consistent representations from the start?
- **Other biological symmetries:**
  - Can similar consistency regularization handle other symmetries (e.g., codon translation, RNA folding)?
- **Generative models:**
  - How to enforce RC consistency in generative DNA models (e.g., Evo 2, GENERator)?
- **Interpretability:**
  - Does RC consistency improve model interpretability? What features do RC-consistent models learn?
- **Combination with architectural methods:**
  - Can RCCR complement architectural equivariance (e.g., in Caduceus) for even better performance?

---

9. Context and Broader Impact

Relation to other work

- **Compared to RC data augmentation (RC-Aug):**
  - RC-Aug exposes model to both orientations during training but doesn't enforce agreement.
  - RCCR directly penalizes disagreement, leading to better consistency while maintaining task performance.
- **Compared to test-time averaging (TTA):**
  - TTA averages predictions on x and RC(x) at inference, guaranteeing consistency but doubling cost.

    - RCCR produces a single, consistent model without inference-time overhead.
- **Compared to architectural equivariance (Caduceus, RC-equivariant CNNs):**
    - Architectural methods hardcode symmetry but may reduce flexibility and aren't applicable to existing pretrained backbones.
    - RCCR is a flexible fine-tuning method that works with any backbone.
- **Connection to consistency regularization:**
    - RCCR is a form of consistency regularization (common in semi-supervised learning) applied to biological symmetry.
    - Similar principles could apply to other data augmentations or symmetries.

Broader scientific and practical impact

- **Improves DNA LM reliability:**
    - Addresses a critical failure mode that undermines trust in DNA LMs for clinical and research applications.
- **Standardizes evaluation:**
    - Introduces RC consistency metrics (SFR, RC-Corr) that should be reported alongside task metrics.
- **Practical recipe:**
    - Provides a simple, effective method that can be immediately applied to improve existing DNA LMs.
- **Theoretical contribution:**
    - Proves that enforcing consistency doesn't sacrifice accuracy under appropriate conditions, providing theoretical justification for the approach.

Open questions for future research

- **Pretraining integration:**
  - Can consistency principles be incorporated into pretraining objectives (e.g., masked language modeling)?
- **Other symmetries:**
  - What other biological symmetries should be enforced (e.g., codon translation, RNA secondary structure)?
- **Generative models:**
  - How to ensure RC consistency in sequence generation models?
- **Combination strategies:**
  - Can RCCR be combined with architectural equivariance for even better performance?

---

10. Key Takeaways

1. **Biological priors should be encoded in learning objectives:**
   Rather than hoping models learn symmetries from data, explicitly penalize violations of known biological priors (like RC symmetry).

2. **Consistency regularization is powerful:**
   The principle of enforcing agreement between semantically equivalent inputs (e.g., x and RC(x)) is a general technique applicable beyond DNA.

3. **Theoretical guarantees matter:**
   Proving that symmetrization is risk non-increasing and that minimizers are consistent provides confidence that the method won't hurt performance.

4. **Evaluation should measure what matters:**
   Don't just report task accuracy; measure consistency metrics (SFR, RC-Corr) to ensure models are robust to orientation.

5. **Fine-tuning can fix pretraining issues:**
   Even if pretrained models don't respect symmetries, fine-tuning with appropriate objectives can enforce them without architectural changes.

6. **Not all tasks are symmetric:**
   Some tasks (e.g., strand classification) explicitly violate RC symmetry; don't apply RCCR blindly.

7. **Single robust model > inference-time fixes:**
   Producing a model that is consistent by design is better than masking inconsistency at inference time (TTA).

8. **Backbone-agnostic methods are valuable:**
   Methods that work across diverse architectures (Transformers, Hyena) and tokenization schemes are more practical than architecture-specific solutions.

9. **Hyperparameter tuning is necessary:**
   Regularization strength λ needs to be tuned per task, but moderate values (0.1-0.3) generally work well.

10. **This is a practical contribution:**
    RCCR is a simple, effective method that can be immediately applied to improve DNA LM reliability, making it a valuable tool for practitioners working with genomic foundation models.