RNA SECONDARY STRUCTURE PREDICTION BY LEARNING UNROLLED ALGORITHMS

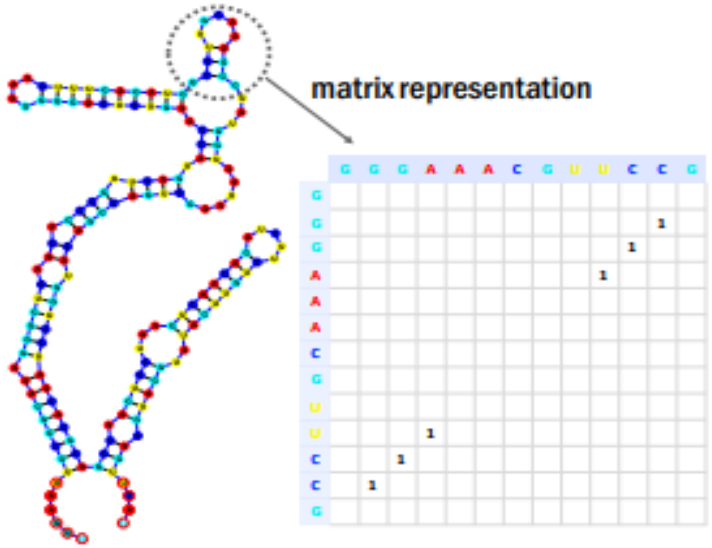
Xinshi Chen, Yu Li, Ramzan Umarov, Xin Gao, Le Song

Presented by

Monjura Afrin Rumi

Background

- Ribonucleic acid (RNA) - essential in numerous cellular processes and regulating gene expression
- Primary structure of RNA $x := (x_1, \ldots, x_L)$ where $x_i \in \{A, G, C, U\}$.
- Secondary structure

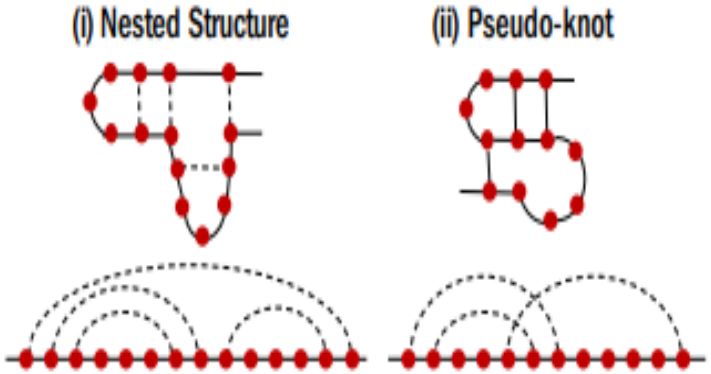


Motivation

- Assumption by existing methods
  - energy minimization

$$A^* = \arg\min_A E_{\boldsymbol{x}}(A).$$

- Exponentially large search space
  - Assumption – nested structure
  - Optimal substructure $O(L^3)$
- Pseudoknots
  - 1.4% of base-pairs
  - present in around 40% of the RNAs
  - Assist folding into 3D structures

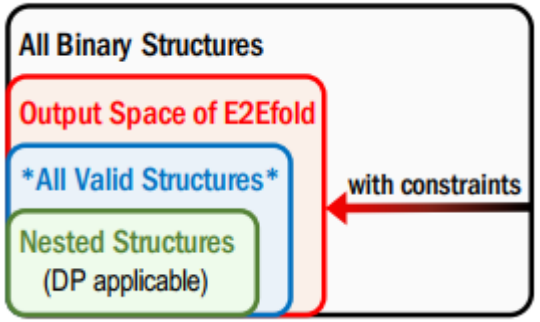


Proposed Method

- Secondary structure is the output of a feed-forward function

$$A^* = \mathcal{F}_\theta(x)$$

- Challenges
  - RNA secondary structure needs to obey certain hard constraints
  - number of RNA data points is limited
  - post processing to enforce constraints



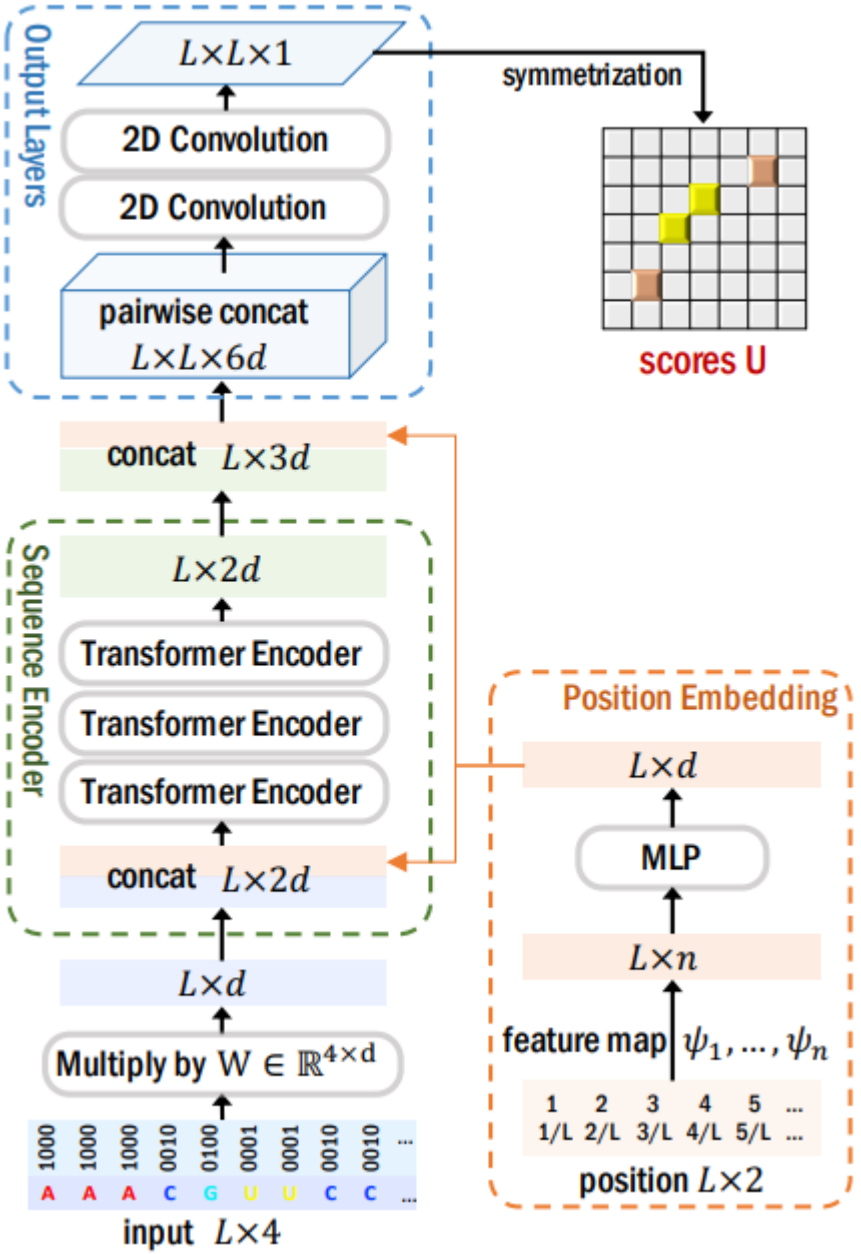
- End-to-end network - E2Efold
  - Deep Score Network $U_\theta(x)$: a transformer-based deep model which represents sequence information useful for structure prediction.
  - Post-Processing Network PP_ϕ : a multilayer network which gradually enforces the constraints and restrict the output space

Deep Score Network

- *L* × 4 dimensional one-hot embedding as input
- Position embedding matrix
  - distinguishes each x_i by exact and relative position

$$P_i = \mathrm{MLP}\big(\psi_1(i), \ldots, \psi_\ell(i), \psi_{\ell+1}(i/L), \ldots, \psi_n(i/L)\big).$$

- Transformer Encoders
  - encode the sequence information and the global dependency between nucleotides
- 2D Convolution layers
  - output the pairwise scores
- L × L symmetric matrix as output
  - X_{ij} denotes score of nucleotides x_i and x_j being paired



Post Processing Network

(i) Only three types of nucleotides combinations, $\mathcal{B} := \{AU, UA\} \cup \{GC, CG\} \cup \{GU, UG\}$, can form base-pairs.	$\forall i, j$, if $x_i x_j \notin \mathcal{B}$, then $A_{ij} = 0$.
(ii) No sharp loops are allowed.	$\forall \lvert i - j \rvert < 4, A_{ij} = 0$.
(iii) There is no overlap of pairs, i.e., it is a matching.	$\forall i, \sum_{j=1}^{L} A_{ij} \leq 1$.

- Target output $A(x) := \{\hat{A} \in [0,1]^{L \times L}$
- Maximize total score $\frac{1}{2}\sum_{i,j}(U_\theta(x)_{ij} - s)\hat{A}_{ij}$
- Nonlinear transformation $\mathcal{T}(\hat{A}) := \frac{1}{2}(\hat{A} \circ \hat{A} + (\hat{A} \circ \hat{A})^\top) \circ M(x)$
- L1 penalty term $\|\hat{A}\|_1 := \sum_{i,j} |\hat{A}_{ij}|$

$$\max_{\hat{A} \in \mathbb{R}^{L \times L}} \frac{1}{2}\left\langle U_\theta(x) - s, A := \mathcal{T}(\hat{A}) \right\rangle + \rho \|\hat{A}\|_1 \quad \text{s.t. } A1 \leq 1$$

- Lagrange multiplier λ

$$\min_{\lambda \geq 0} \max_{\hat{A} \in \mathbb{R}^{L \times L}} \underbrace{\frac{1}{2}\langle U_\theta(x) - s, A \rangle - \langle \lambda, \mathrm{relu}(A1 - 1) \rangle}_{f} - \rho \|\hat{A}\|_1.$$

Algorithm

- proximal gradient for maximization and gradient descent for minimization
- Final deep model

$$\textbf{E2Efold}: \quad \{A_t\}_{t=1}^T = \overbrace{\mathrm{PP}_\phi(\underbrace{U_\theta(\boldsymbol{x})}_{\text{Deep Score Network}}, M(\boldsymbol{x}))}^{\text{Post-Process Network}}.$$

- Continuous function to mimic TP, TN, FP, FN

$$\mathrm{TP} = \langle A, A^* \rangle,\ \mathrm{FP} = \langle A, 1 - A^* \rangle,\ \mathrm{FN} = \langle 1 - A, A^* \rangle,\ \mathrm{TN} = \langle 1 - A, 1 - A^* \rangle.$$

- Differentiable F1 loss

$$\mathcal{L}_{-\mathrm{F1}}(A, A^*) := -2\langle A, A^* \rangle / \left(2\langle A, A^* \rangle + \langle A, 1 - A^* \rangle + \langle 1 - A, A^* \rangle\right)$$

- Overall loss function

$$\min_{\theta,\phi} \frac{1}{|\mathcal{D}|} \sum_{(x,A^*)\in\mathcal{D}} \frac{1}{T} \sum_{t=1}^{T} \gamma^{T-t} \mathcal{L}_{-\mathrm{F1}}(A_t, A^*)$$

$$\{A_t\}_{t=1}^T = \mathrm{PP}_\phi(U_\theta(\boldsymbol{x}), M(\boldsymbol{x}))$$

Algorithm 1: Post-Processing Network $\mathrm{PP}_\phi(U, M)$

Parameters $\phi := \{w, s, \alpha, \beta, \gamma_\alpha, \gamma_\beta, \rho\}$
$U \leftarrow \mathrm{softsign}(U - s) \circ U$
$\hat{A}_0 \leftarrow \mathrm{softsign}(U - s) \circ \mathrm{sigmoid}(U)$
$A_0 \leftarrow \mathcal{T}(\hat{A}_0); \quad \boldsymbol{\lambda}_0 \leftarrow w \cdot \mathrm{relu}(A_0\mathbf{1} - \mathbf{1})$
For $t = 0, \ldots, T-1$ **do**
  $\boldsymbol{\lambda}_{t+1}, A_{t+1}, \hat{A}_{t+1} = \mathrm{PPcell}_\phi(U, M, \boldsymbol{\lambda}_t, A_t, \hat{A}_t, t)$
return $\{A_t\}_{t=1}^T$

Algorithm 2: Neural Cell PPcell_ϕ

Function $\mathtt{PPcell}_\phi(U, M, \boldsymbol{\lambda}, A, \hat{A}, t)$**:**
  $G \leftarrow \frac{1}{2}U - (\boldsymbol{\lambda} \circ \mathrm{softsign}(A\mathbf{1} - \mathbf{1}))\,\mathbf{1}^\top$
  $\dot{A} \leftarrow \hat{A} + \alpha \cdot {\gamma_\alpha}^t \cdot \hat{A} \circ M \circ (G + G^\top)$
  $\hat{A} \leftarrow \mathrm{relu}(|\dot{A}| - \rho \cdot \alpha \cdot {\gamma_\alpha}^t)$
  $\hat{A} \leftarrow 1 - \mathrm{relu}(1 - \hat{A})$ [i.e.,$\min(\hat{A}, 1)$]
  $A \leftarrow \mathcal{T}(\hat{A}); \boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \beta \cdot {\gamma_\beta}^t \cdot \mathrm{relu}(A\mathbf{1} - \mathbf{1})$
  return $\boldsymbol{\lambda}, A, \hat{A}$

Related Work

- Classical RNA folding methods
  - energy minimization through DP
  - Run time $O(L^3)$ and space $O(L^2)$
  - RNAstructure, Vienna RNAfold, UNAFold
  - LinearFold: run time O(L)
  - Heuristic algorithms: HotKnots, Probknots
- Learning-based RNA folding methods
  - rely on DP-based method
  - ContraFold, ContextFold, CDPfold
- Learning with differentiable algorithms
  - differentiable unrolled algorithms as a building block in neural architectures
  - OptNet : cubic complexity

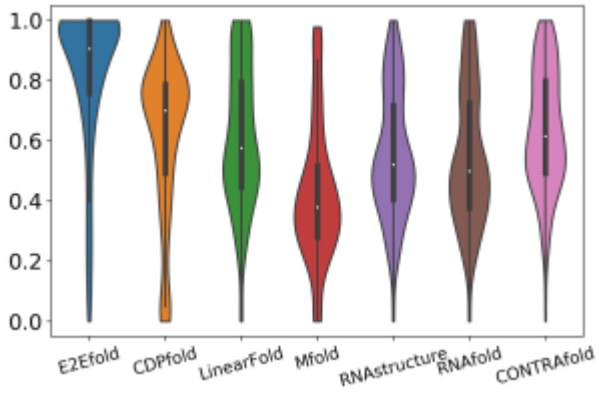
Result: dataset and training

Table 1: Dataset Statistics

Type	ArchiveII		RNAStralign	
	length	#samples	length	#samples
All	28~2968	3975	30~1851	30451
16SrRNA	73~1995	110	54~1851	11620
5SrRNA	102~135	1283	104~132	9385
tRNA	54~93	557	59~95	6443
grp1	210~736	98	163~615	1502
SRP	28~533	928	30~553	468
tmRNA	102~437	462	102~437	572
RNaseP	120~486	454	189~486	434
telomerase	382~559	37	382~559	37
23SrRNA	242~2968	35	-	-
grp2	619~780	11	-	-

Table 2: Results on RNAStralign test set. "(S)" indicates the results when one-position shift is allowed.

Method	Prec	Rec	F1	Prec(S)	Rec(S)	F1(S)
E2Efold	**0.866**	**0.788**	**0.821**	**0.880**	**0.798**	**0.833**
CDPfold	0.633	0.597	0.614	0.720	0.677	0.697
LinearFold	0.620	0.606	0.609	0.635	0.622	0.624
Mfold	0.450	0.398	0.420	0.463	0.409	0.433
RNAstructure	0.537	0.568	0.550	0.559	0.592	0.573
RNAfold	0.516	0.568	0.540	0.533	0.587	0.558
CONTRAfold	0.608	0.663	0.633	0.624	0.681	0.650



Result: no re-training and running time

Table 3: Performance comparison on ArchiveII

Method	Prec	Rec	F1	Prec(S)	Rec(S)	F1(S)
E2Efold	**0.734**	**0.66**	**0.686**	**0.758**	**0.676**	**0.704**
CDPfold	0.557	0.535	0.545	0.612	0.585	0.597
LinearFold	0.641	0.617	0.621	0.668	0.644	0.647
Mfold	0.428	0.383	0.401	0.450	0.403	0.421
RNAstructure	0.563	0.615	0.585	0.590	0.645	0.613
RNAfold	0.565	0.627	0.592	0.586	0.652	0.615
CONTRAfold	0.607	0.679	0.638	0.629	0.705	0.662

Table 4: Inference time on RNAStralign

Method		total run time	time per seq
E2Efold (Pytorch)		**19m (GPU)**	**0.40s**
CDPfold (Pytorch)		440m*32 threads	300.107s
LinearFold	(C)	20m	0.43s
Mfold	(C)	360m	7.65s
RNAstructure	(C)	3 days	142.02s
RNAfold	(C)	26m	0.55s
CONTRAfold	(C)	1 day	30.58s

Table 5: Evaluation of pseudoknot prediction

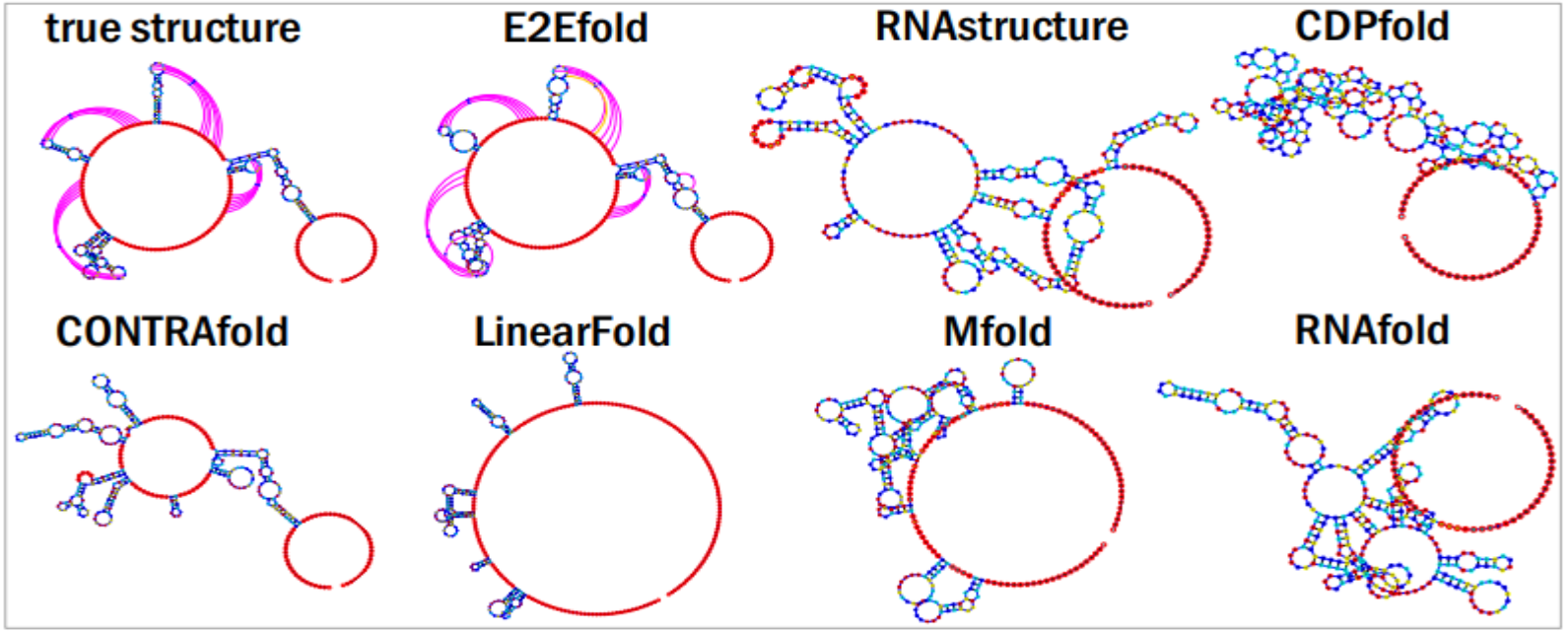
Method	Set F1	TP	FP	TN	FN
E2Efold	0.710	1312	242	1271	0
RNAstructure	0.472	1248	307	983	286

Result: visualization and ablation study

- Naive post processing
  - Choose offset s and set A_{ij} = 1 if $U_\theta(\boldsymbol{x})_{ij} > s$.

Table 6: Ablation study

Method	Prec	Rec	F1	Prec(S)	Rec(S)	F1(S)
E2Efold	**0.866**	**0.788**	**0.821**	**0.880**	**0.798**	**0.833**
U_θ+PP	0.755	0.712	0.621	0.782	0.737	0.752



Comments

- Cross validation
- performance per RNA category
- Ablation study
  - Variation in deep score network