中國人民大學
RENMIN UNIVERSITY OF CHINA

高瓴人工智能学院
Gaoling School of Artificial Intelligence

Deep Generative Design of RNA Family Sequences

Shunsuke Sumi, Michiaki Hamada, Hirohide Saito

Nature Method

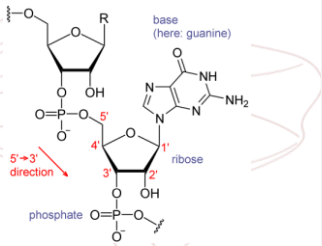
Fanmeng Wang

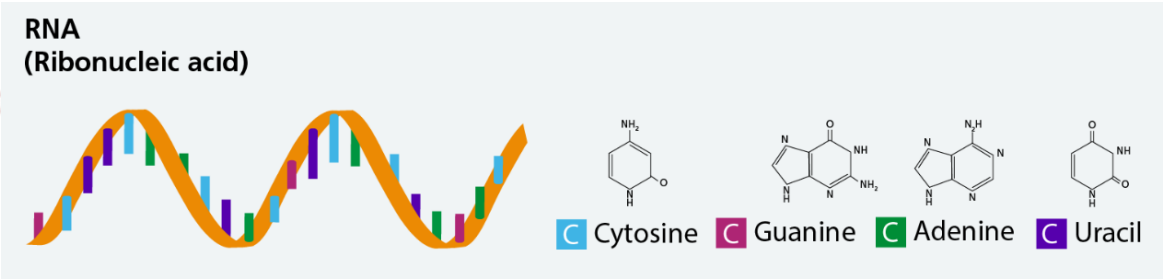
2025-6-13

Introduction

■ RNA Engineering

- Over the past decades, RNA function exploitation has drived the invention of various synthetic molecular systems, leading to a substantial impact on numerous fields such as basic research, biomanufacturing and medical applications.
- However, the experimental search for desired RNA sequences remains costly and inefficient due to the vast sequence space of RNA, which exponentially increases in complexity with sequence length.
- Therefore, a versatile computational platform to understand the sequence space and to efficiently design functional RNA is in great demand.





Introduction

■ Existing Methods

➢ RNA Inverse Folding:

- It aims to find sequences that fold into a given secondary structure, guided by RNA secondary structure prediction and a discrete optimization algorithm.
- Howevrer, the functionality of RNA is not characterized by structure alone. Besides, RNA inverse folding lacks flexibility and generalizability, and its accuracy is inherently limited by the accuracy of RNA secondary structure prediction and the optimization algorithm.

Introduction

■ Existing Methods

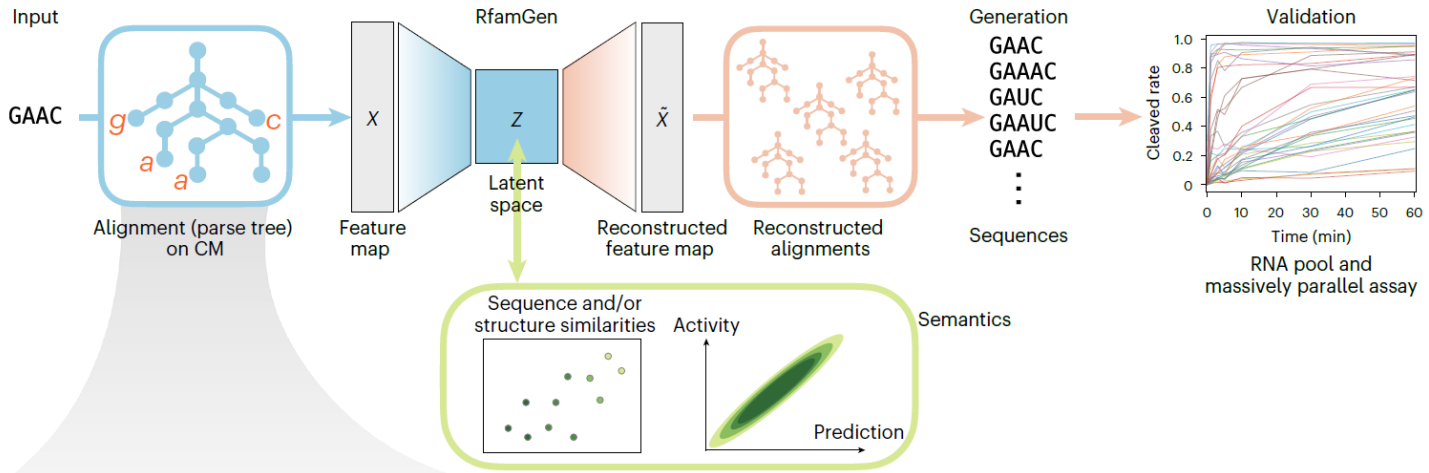
➢ Covariance Models (CM):

- It's a statistical framework for RNA alignment and consensus secondary structure, quantitatively evaluates variations of sequence and structure without relying on RNA secondary structure prediction.
- It has been the gold standard in RNA homology search for decades to categorize most functional RNA species into thousands of RNA families.
- However, it has not been used in functional RNA design previously.

Introduction

■ RfamGen

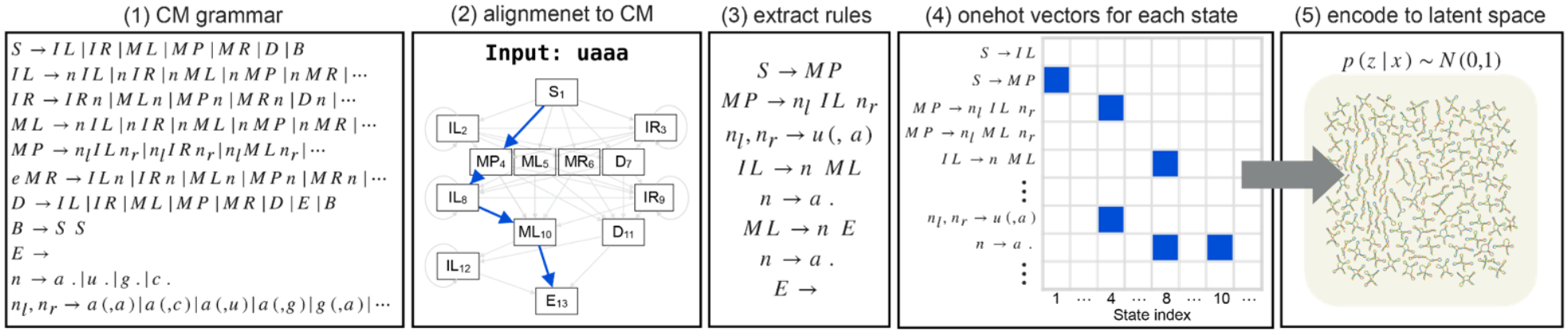
- In this study, we propose the RNA family sequence generator (RfamGen), which is a deep generative model for functional RNA design.
- In particular, RfamGen leverages Covariance Models (CM) with Variational Autoencoders (VAE) to generate artificial sequences of an RNA family, while also providing semantically meaningful representations of sequences.



Method

■ Data Preprocessing

1. Given a sequence, it can be aligned to a CM and the alignment is transformed to a one-hot expression.



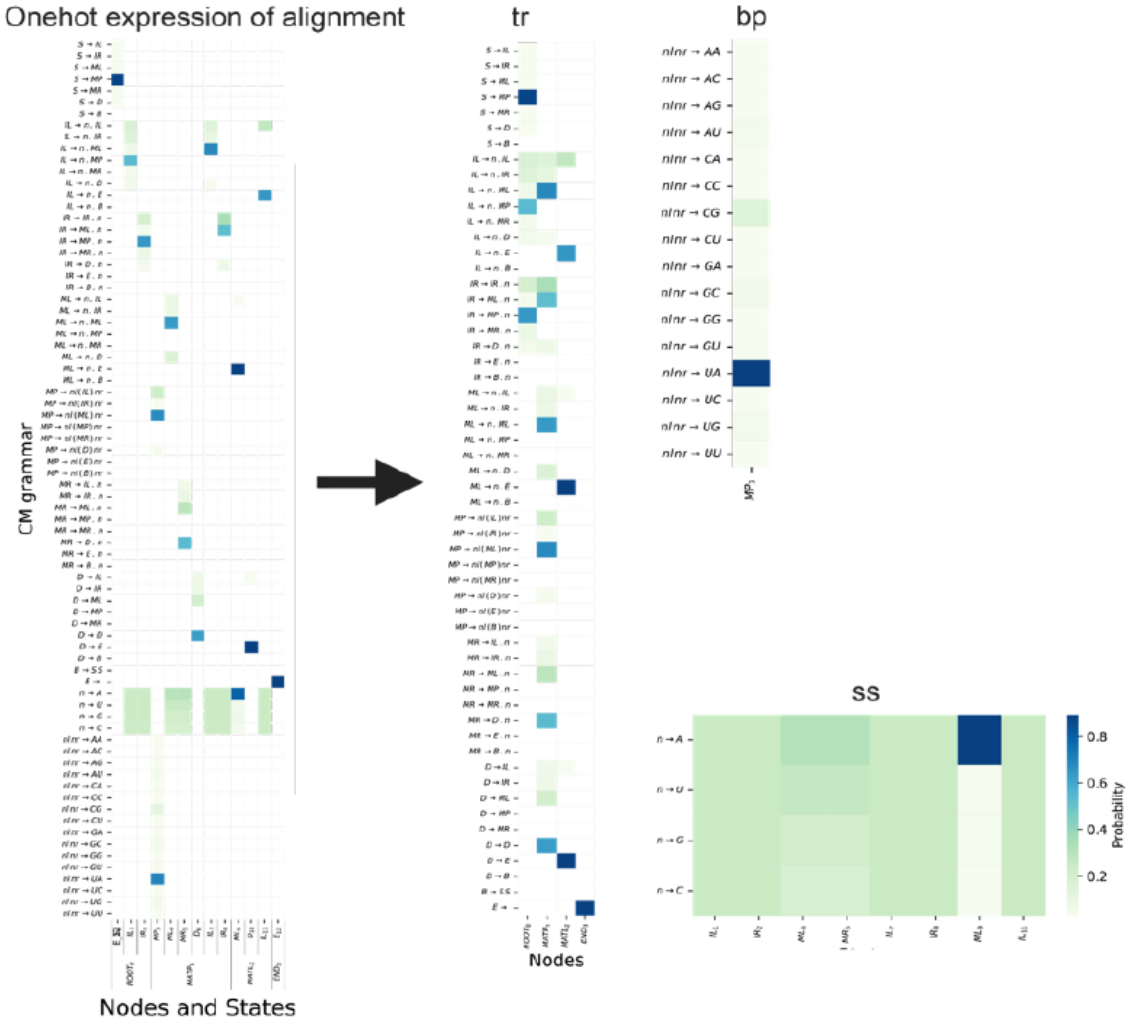
CM grammar contains 76 rules: 56 rules of transition, 4 rules of single strand emission and 16 rules of base pair emission.

$$S \to IL|IR|ML|MP|MR|D|B$$
$$IL \to n'.'IL|n'.'IR|n'.'ML|n'.'MP|n'.'MR|n'.'D|n'.'E|n'.'B$$
$$IR \to IR'.'n|ML'.'n|MP'.'n|MR'.'n|D'.'n|E'.'n|B'.'n$$
$$ML \to n'.'IL|n'.'IR|n'.'ML|n'.'MP|n'.'MR|n'.'D|n'.'E|n'.'B$$
$$MP \to n_l'('IL')'n_r|n_l'('IR')'n_r|n_l'('ML')'n_r|n_l'('MP')'n_r|n_l'('MR')'n_r|n_l'('D')'n_r|n_l'('E')'n_r|n_l'('B')'n_r$$
$$MR \to IL'.'n|IR'.'n|ML'.'n|MP'.'n|MR'.'n|D'.'n|E'.'n|B'.'n$$
$$D \to IL|IR|ML|MP|MR|D|E|B$$
$$B \to SS$$
$$E \to$$
$$n \to 'A'|'U'|'G'|'C'$$
$$n_l n_r \to 'AA'|'AC'|'AG'|'AU'|'CA'|'CC'|'CG'|'CU'|'GA'|'GC'|'GG'|'GU'|'UA'|'UC'|'UG'|'UU'$$

Method

■ Data Preprocessing

2. The one-hot expression is further decomposed into a triplet of three matrices (tr, ss, bp), where each matrix represents the transition, single strand emission and base pair emission probability.




```
import preprocess
from infernal_tools import TracebackFileReader, make_trsp_from_deriv_dict

preprocess.cmalign(
    cmfile = args.cmfile,
    seqfile = args.fasta,
    log = True,
    trunc = False,
    suffix = "_notrunc",
    cpu = args.cpu)
basename, _  = os.path.splitext(args.fasta)
path_to_traceback = basename + "_notrunc_traceback.txt.gz"

id_all = []
with gzip.open(path_to_traceback, "rb") as tb:
    for line in tb:
        if line.startswith(b'>'):
            id_all.append(line.replace(b">", b"").decode('utf-8'))

n_size = len(id_all)
print(f"Start reading {path_to_traceback}...")
tbreader       = TracebackFileReader(args.cmfile, path_to_traceback)
tbtext_all     = [list(tbreader.traceback_iter())[0] for i in range(n_size)]
print(f"Start making tbtext")
tbdf_all       = [tbreader._make_tbdf_from_tbtext(tbtext) for tbtext in tbtext_all]
print(f"Start making tbdict")
deriv_dict_all = [tbreader.make_aligned_tbdict_from_tbdf_ELinitCM(tbdf) for tbdf in tbdf_all]

tr, ss, bp = make_trsp_from_deriv_dict(args.cmfile, deriv_dict_all[0])
```

Method

■ Data Preprocessing

3. Furthermore, to reduce the observation bias of the Rfam database, each sequence is reweighted by the number of sequences outside a neighborhood of a sequence as proposed in previous studies.

$$D_{\mathrm{H}}(s,t) = \sum_{M\in\{\mathrm{tr,ss,bp}\}} \mathrm{Hamming\ Distance}(M_s, M_t)/\mathrm{length}(M)$$

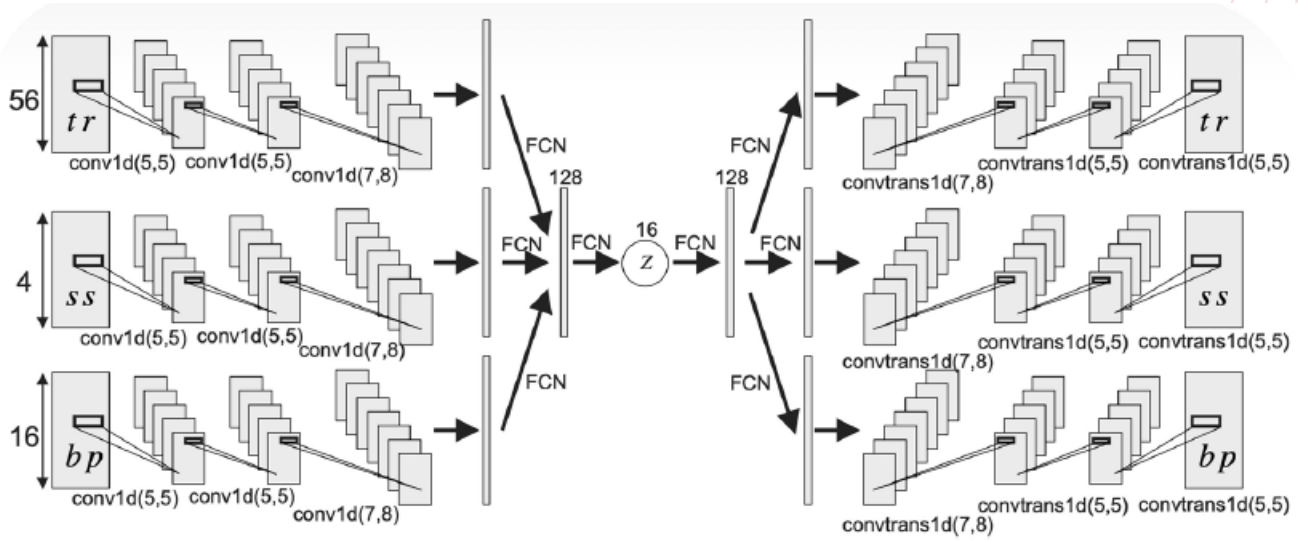
$$\pi_s = \left(\sum_{t}^{N} I\left[D_{\mathrm{H}}(s,t) < \theta\right]\right)^{-1}$$

4. Finally, each sequence can be represented by a triplet X (tr, ss, bp) along with the corresponding weight π_s.

Method

■ RfamGen Architecture and Training

➢ RfamGen takes the triplet *X* (tr, ss, bp) as input of VAE and produces the reconstructed triplet as output.



$$L(X;\theta,\varphi) = \text{Reconst}_{\theta,\varphi}(X,\tilde{X}) - \beta\text{KL}(q_\varphi(Z|X)|p(Z))$$

$$\text{Reconst}_{\theta,\varphi}(X,\tilde{X}) = \pi_X \sum_{M\in\{\text{tr,ss,bp}\}} \text{cross entropy}(M,\tilde{M}_{\theta,\varphi})$$

```
from infernal_tools import CovarianceModel, make_deriv_dict_from_trsp
cmreader = CMReader(args.cmfile)
print("Start loading cm dict. This process may take much time for long sequences.")
cm_deriv_dict = cmreader.load_derivation_dict_from_cmfile()
out_dict = make_deriv_dict_from_trsp(cm_deriv_dict, (softmax(tr), softmax(s), softmax(p)))
cm = CovarianceModel(out_dict)
seq, _ = cm.cmemit(sample = False)[0]
```

Experiment

Dataset

- RfamGen takes various RNA families from the Rfam database as dataset.

Baseline

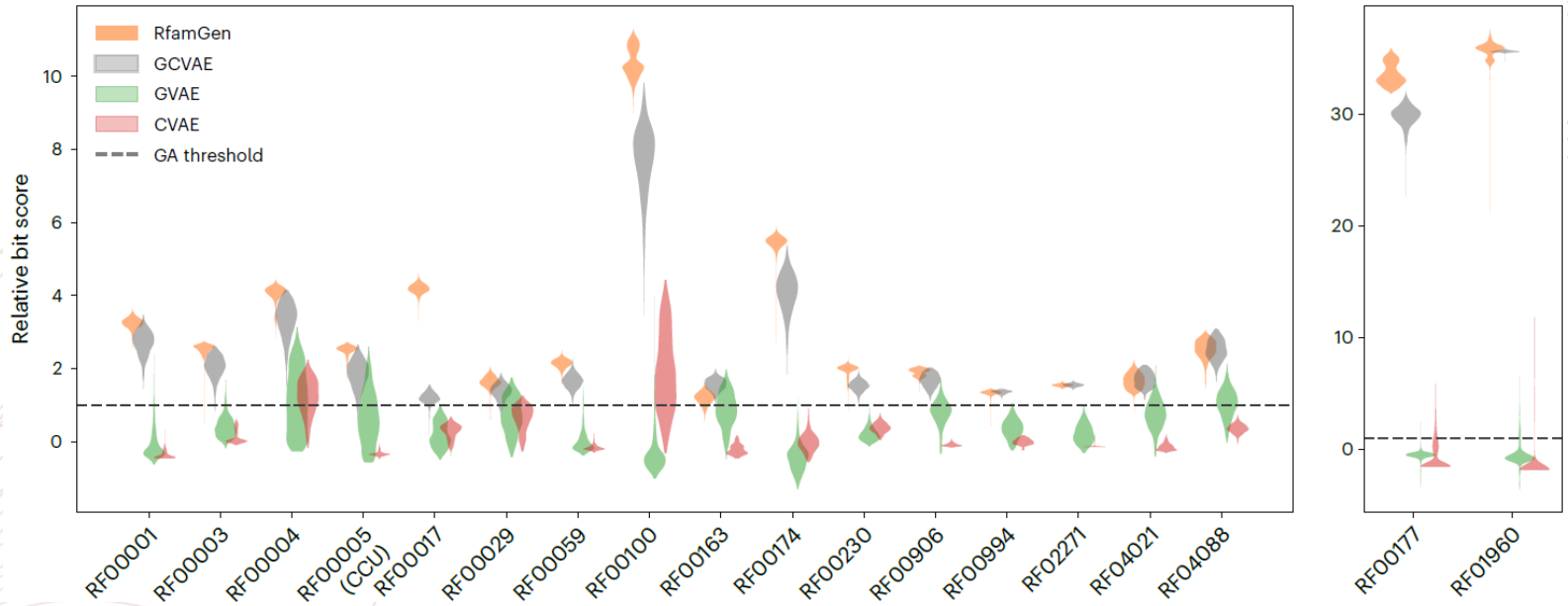
- GCVAE (gapped character VAE), which includes only alignment
- GVAE (grammar VAE), which includes only secondary structure
- CVAE (character VAE), which includes neither

Evaluation

- We assessed the quality of the generated sequences by likelihood as the RNA family ('bit score') calculated based on alignment to the 'ground truth' CM.

Experiment

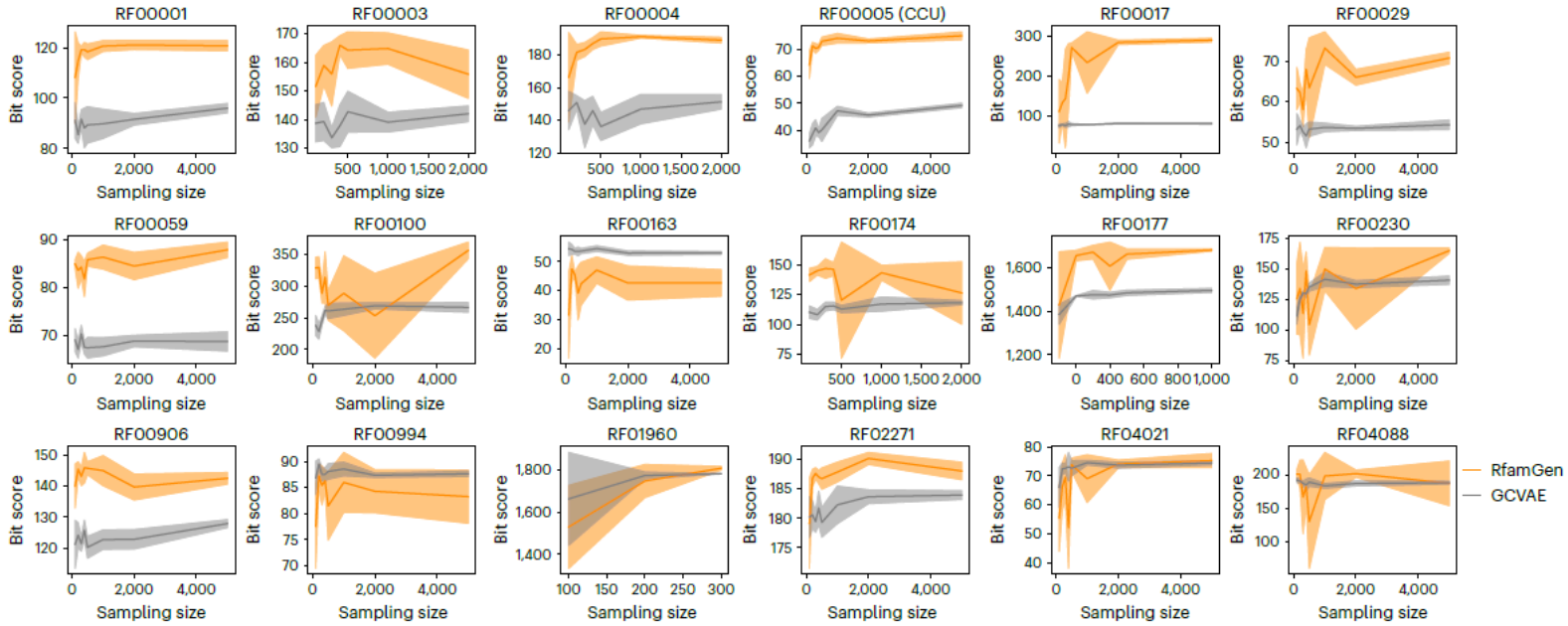
- **RfamGen is a data-efficient generative model**
  - We firsly use 18 RNA families with full alignments composed of at least 10,000 sequences in the Rfam database as dataset, and quantify the average bit score of 1,000 randomly generated sequences from the models for each RNA family.



RfamGen performs best across most RNA families

Experiment

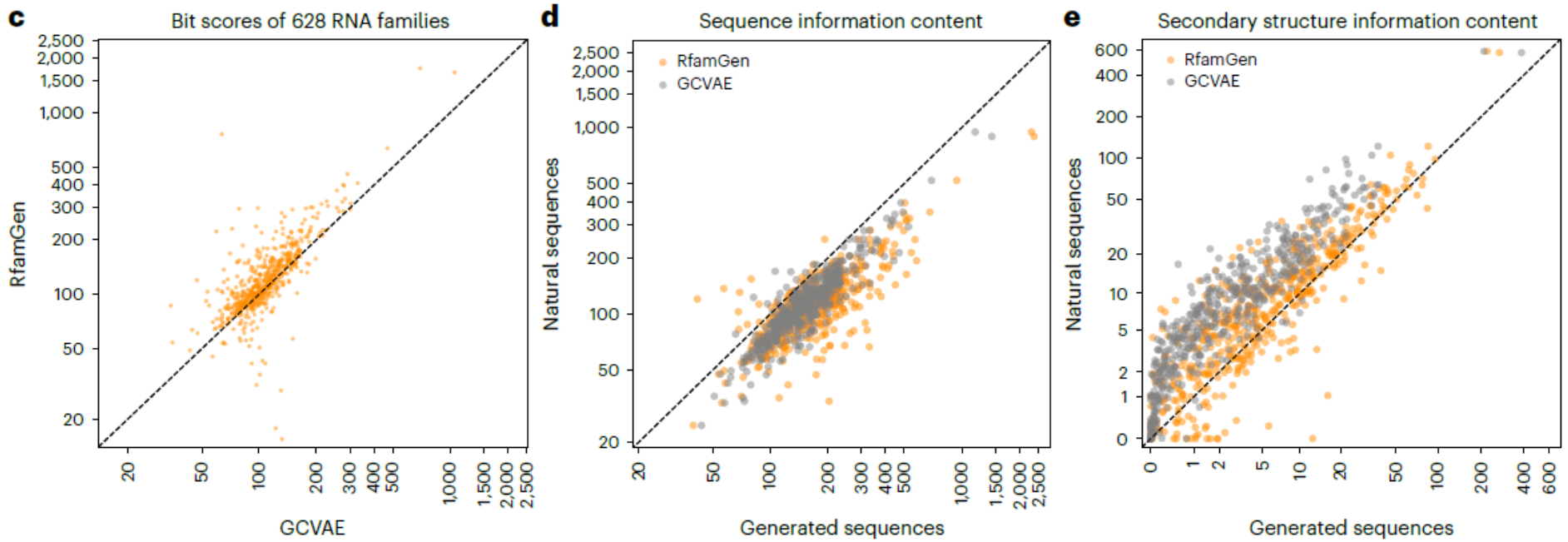
- **RfamGen is a data-efficient generative model**
  - We next assesse the effectiveness and robustness of RfamGen and GCVAE against various data sizes by undersampling.



Only ~500 input sequences needed to reach near-peak performance for most RNA families
Maintains high generation capability on small datasets

Experiment

- **RfamGen is a data-efficient generative model**
  - Finally, we train RfamGen using 628 RNA families whose full alignments consisted of at least 100 sequences in the Rfam database, and compare it to GCVAE to confirm its breadth of application.



Successfully learns nested base pair information

Experiment

- **RfamGen learns a semantically rich latent space**
  - We examine whether the latent space of RfamGen contains meaningful representations of sequences.

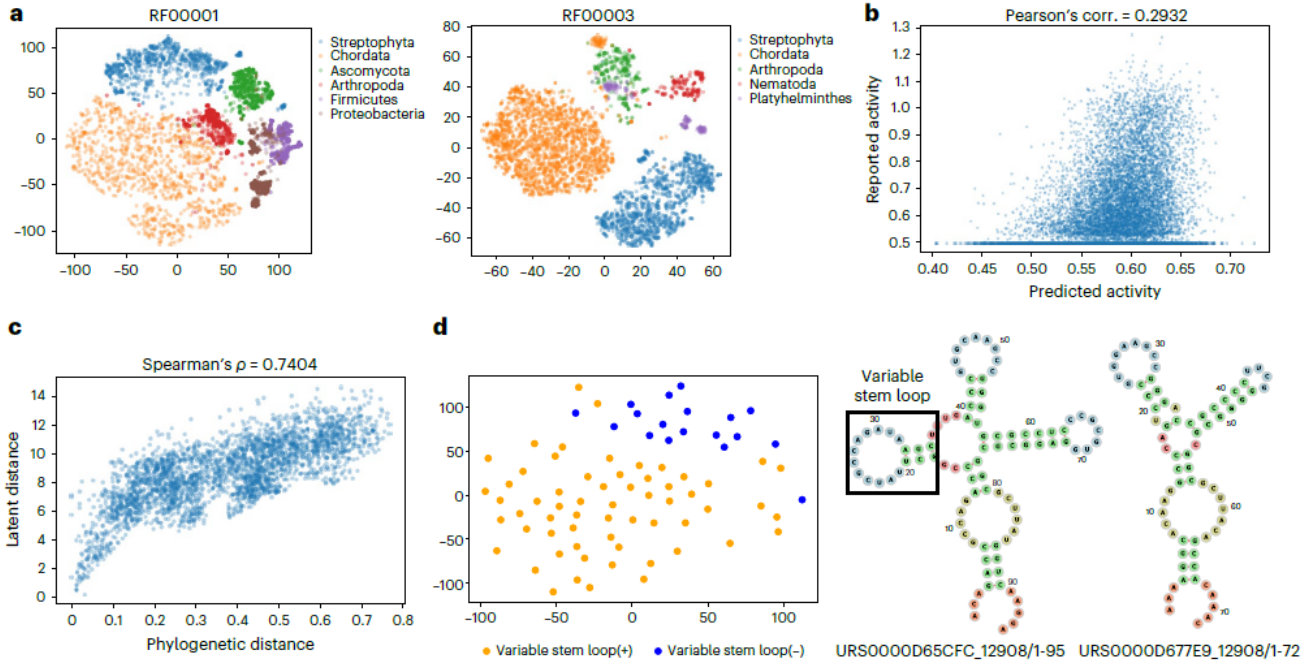


Fig. 3 | Latent space has semantically rich representation of sequences. **a**, *t*-SNE visualization of the latent space trained with 5S ribosomal RNA (RF00001) and U1 spliceosomal RNA (RF00003), colored with phylogenetic information. **b**, Prediction of activity of tRNA variants reported by Li et al.[31] (*y* axis) and predicted activity by linear regression of latent space (*x* axis). **c**, Comparison of phylogenetic distance (*x* axis) and Euclidean distance in latent space (latent distance, *y* axis) trained by twister-sister ribozymes (RF02681). **d**, *t*-SNE visualization of the latent space trained by twister-sister ribozyme (RF02681) colored with structural variants (left). Examples of the structural variants of a twister-sister ribozyme that are defined by the presence or absence of the variable stem loop shown in the black box (right), and were visualized by forna server.

Conclusion

- In this work, we present RfamGen, which is a deep-learning method that designs RNA family sequences in a data-efficient manner.
- RfamGen achieves the first integration of CM architecture into deep generative models, and further provides semantically meaningful representations of sequences.
- Extensive experiments demonstrate its great generation capability and semantically rich latent space, thus building a powerful and general platform for RNA engineering with enormous application potential in biotechnology and medicine.

Thank You for listening!

Fanmeng Wang

2025-6-13