学习设计 RNA 结构

温度

摘要

近年来，RNA 分子设计在医学、合成生物学、生物技术和生物信息学领域引起了广泛关注，因为许多功能性 RNA 分子被证明参与了转录、表观遗传学和翻译的调控过程。由于 RNA 的功能取决于它的结构特性，RNA 设计问题就是找到一个满足给定结构约束的 RNA 序列。在这里，我们提出了一个新的算法 RNA 设计问题，称为 LEARNA。LEARNA 使用深度强化学习训练策略网络，在给定特定目标结构的情况下，按顺序设计整个 RNA 序列。通过在 20 个 CPU 核上对 65000 个不同的 RNA 设计任务进行一小时的元学习，我们的扩展 Meta-LEARNA 构建了一个 RNA 设计策略，可以开箱即用地应用于解决新的 RNA 设计任务。在方法上，我们相信这是第一次，我们联合优化了政策网络、训练过程的超参数和决策过程的制定的丰富的体系结构空间。对两个广泛使用的 RNA 设计基准以及我们介绍的第三个基准的综合实证结果表明，我们的方法在前者上实现了新的最先进性能，同时在达到之前的最先进性能方面也快了几个数量级。在消融研究中，我们分析了我们的方法不同组成部分的重要性。

关键词：RNA 设计；深度强化学习

1 引言

RNA 是生物细胞中携带信息的主要生物聚合物之一。最近的研究揭示了功能性非蛋白质编码 RNA(ncRNA) 在调控过程和转录控制中的关键作用，这也与某些疾病有关，如帕金森病和阿尔茨海默病。功能性 ncrna 参与表观遗传标记的调制、信使 RNA(mRNA) 稳定性的改变、mRNA 翻译、选择性剪接、信号转导和大分子复合物的支架[1]。因此，ncRNA 分子的工程越来越重要，应用范围从生物技术和医学到合成生物学。事实上，已经报道了在体外和体内成功创建功能性 RNA 序列的尝试。在 RNA 最基本的结构形式中，它是由腺嘌呤 (a)、鸟嘌呤 (G)、胞嘧啶 (C) 和尿嘧啶 (U) 四种核苷酸组成的序列。这种核苷酸序列被称为 RNA 序列，或一级结构。虽然 RNA 序列是蓝图，但 RNA 分子的功能结构是由折叠决定的，将 RNA 序列翻译成其三维三级结构。层序的内在热力学性质决定了由此产生的褶皱。两个对应核苷酸之间形成的氢键是热力学模型的驱动力之一，对三级结构影响很大。包含这些氢键的结构通常被称为 RNA 的二级结构。许多 RNA 三级结构设计算法直接作用于 RNA 二级结构。因此，推动 RNA 工程的发展需要快速准确的 RNA 二级结构设计算法。

2 相关工作

2.1 架构和超参数搜索

此前已有关于联合架构搜索和超参数优化的研究。在这里，本文调整了这种方法，以用于深度强化学习和更丰富的架构空间[2]。尽管强化学习已用于执行架构搜索以及联合架构和超参数搜索，但据本文作者所知，本文是反向的第一个应用：强化学习的架构搜索。

2.2 RNA 设计

大多数针对 RNA 设计问题的算法要么是局部算法，要么是全局算法。局部方法通常对单个序列进行操作，并试图在损失函数的指导下，通过一次改变少量核苷酸来寻找解决方案。

2.3 利用人类解决方案进行 RNA 设计

最近，RNA 设计的另一个不太一般的方向是将人类知识的先验强加给智能体。在这种方法中，大量的模型被训练为人工设计 RNA 设计问题的人类解决方案。此外，为了细化候选解决方案，使用了应用人类策略的自适应步行过程，结合了指导智能体行为的深层域知识。在 Eterna100 基准上报告了所有集合模型的总结果，该基准仅由人工设计的 RNA 设计问题组成，本文在此也报告了这些问题。尽管该方法在这一基准测试中显示出良好的结果，但人类的解决方案和策略不适用于我们从天然 RNA 结构得出的进一步基准测试，并且由于计算成本，本文无法将这项工作纳入比较之中。

3 本文方法

3.1 本文方法概述

本文描述了基于强化学习的 RNA 设计问题的新的生成方法。本文首先将 RNA 设计建模为一个决策过程，然后提出了几种策略来产生学习端到端设计 RNA 的智能体。本论文针对 RNA 设计问题提出了深度强化学习算法 LEARNA，以端到端的方式依次构建候选解。通过在大量生物序列上预训练、局部改进步骤训练 agent、扩展架构和超参数优化。使用了 ViennaRNA 提供的 Zuker 算法。强化学习库 tensorforce，TensorFlow 框架。所有计算在 CPU 上完成。

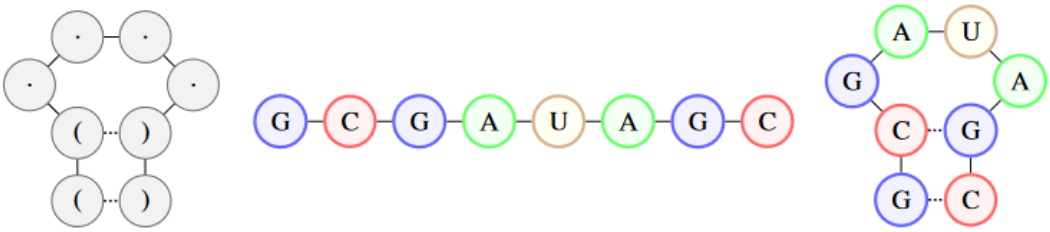

图 1: 方法示意图

3.2 LEARNA

LEARNA 使用深度强化学习来学习策略网络 π θ 的参数 θ。我们的策略网络由输入状态的嵌入层和深度神经网络组成；该神经网络可以调试卷积层、循环层和全连接层的数量，随机初始化参数 θ。我们使用了策略梯度法 PPO 更新参数。

3.3 Meta - LEARNA

Meta - LEARNA 使用元学习方法[3]，将训练集 Ω train 中与目标结构相关的 RNA 设计问题视为任务，并学习在它们之间传递知识。每个目标结构 ω i□Ω train 定义了不同的决策过程 D ω i；使用异步并行 PPO 更新，我们在所有这些上训练单个策略网络。一旦训练完成，参数 θ 固定，通过从学习到的生成式模型中采样，π θ 可以应用于决策过程 D ω。

3.4 Meta-LEARNA-Adapt

Meta-LEARNA-Adapt 结合了前两种策略：首先，我们通过在 Ω train 上运行 Meta - LEARNA 来获得参数 θ 的初始化。然后，当应用于决策过程 D ω 时，使用 LEARNA 策略进一步调整参数 θ。

3.5 决策过程建模

本文提出将给定目标结构 ω 的 RNA 设计问题建模为无折扣决策过程 $\mathcal{D}_\omega := (\mathcal{S}, \mathcal{A}, \mathcal{R}_\omega, \mathcal{P}_\omega)$；其组成为 (状态空间 $\mathcal{S}$, 动作空间 $\mathcal{A}$, 奖励函数 $\mathcal{R}_\omega$ 和转移函数 $\mathcal{P}_\omega$)。RNA 设计问题定义为一个折叠算法，记为 $\mathcal{F}(\cdot)$；用 Ω 表示点括号编码的 RNA 二级结构的集合。

3.6 奖励函数定义

在最终时间步 $\mathcal{T}$ 时刻，智能体将核苷酸分配给候选解 ϕ 的所有位点，环境产生非零报酬 $\mathcal{R}_\omega^T(\phi)$。该奖励基于折叠后的候选解 $\mathcal{F}(\phi)$ 与目标结构 ω 之间的汉明距离 $\mathrm{d_H}(\mathcal{F}(\phi), \omega)$。我们将这个距离对序列长度 $|\omega|$ 进行归一化，得到损失函数 $L_\omega(\mathcal{F}(\phi)) := \mathrm{d_H}(\mathcal{F}(\phi), \omega)/|\omega|$。本文设定奖励函数为：

$$\mathcal{R}_\omega^T(\phi) := (1 - L_\omega(\mathcal{F}(\phi)))^\alpha$$

4 复现细节

4.1 与已有开源代码对比

复现论文引用代码：https://github.com/automl/learna

关键算法部分伪代码：

Procedure 1 Local improvement step(LIS) using Hamming distance $\mathrm{d_H}(\cdot,\cdot)$ and folding algorithm $\mathcal{F}(\cdot)$.

Input: designed solution ϕ, target structure ω, initial Hamming distance δ
Output: locally improved distance
$\Delta \leftarrow \emptyset$
nucleotide_combinations $\leftarrow \{A, G, U, C\}^\delta$
candidate_solutions $\leftarrow$ replaceMismatchedSites(ϕ, ω, nucleotide_combinations)
for ψ ***in*** *candidate_solutions* **do**
  $\delta \leftarrow \mathrm{d_H}(\mathcal{F}(\psi), \omega)$
  if $\delta = 0$ **then**
    return δ
  end
  $\Delta \leftarrow \Delta \cup \{\delta\}$
end
return minΔ

Procedure 2 Computing reward $\mathcal{R}_\omega^T(\phi)$ using LIS (Algorithm 1), Hamming distance $\mathrm{d_H}(\cdot,\cdot)$ and folding algorithm $\mathcal{F}(\cdot)$.

Input: designed solution ϕ, target structure ω, LIS cut-off parameter ξ
Output: reward $\mathcal{R}_\omega^T(\phi)$
$\delta \leftarrow \mathrm{d_H}(\mathcal{F}(\phi), \omega)$
if $\delta = 0$ **then**
  return δ
else if $\delta < \xi$ **then**
$\delta \leftarrow \mathrm{LIS}(\phi, \omega, \delta)$
end
$L_\omega \leftarrow \delta/|\omega|$
return $(1 - L_\omega)^\alpha$

4.2 实验环境搭建

本文使用了 ViennaRNA version 2.4.8 (MCTS - RNA、RL - LS and LEARNA)，2.1.9 (antaRNA) 和 2.4.9 (RNAInverse) 三个版本所提供的 Zuker 算法的实现。本文实验的实现使用了强化学习库 tensorforce，version 0.3.3 和 TensorFlow version 1.4.0。所有计算均在布罗德韦尔 E5-2630v4 2.2 GHz

CPU 上完成，每 10 个核限制 5 GByte RAM。对于 Meta - LEARNA 的训练阶段，本文使用了其中的两个 CPU，但在评估时，所有方法都只允许使用单个核 (使用核心绑定)。

4.3 软件环境安装

操作系统：Linux Ubuntu 20.02 环境管理器：Miniconda 依赖：Python version 3.6 or higher ViennaRNA (recommended version: 2.4.8) numpy pandas version 0.22 requests pytest (for running tests only) tensorflow version 1.4.0 pynisher hpbandster tqdm Distance tensorforce version 0.3.3 dataclasses

4.4 数据集

为了保证足够大和有用的数据集，我下载了 Rfam 数据库版本 13.0[4]的所有家族，并使用 ViennaRNA 软件包对其进行折叠。论文删除了所有具有多个已知解的二级结构，只保留长度在 50 450 之间的结构以匹配现有数据集。为了聚焦更难的序列，论文只保留了一次 MCTS - RNA 无法在 30 秒内解出的序列。我们选择 MCTS - RNA 进行过滤，因为它是文献中最快的算法。将剩余二级结构拆分为训练集 65000 个，验证集 100 个，测试集 100 个二级结构。

数据集为了在不发生过拟合的情况下合理优化所列出的设计选择，我们需要一个指定的训练和验证数据集。然而，以前的 RNA 设计文献中使用的基准测试没有提供训练/验证/测试分割。因此，我们基于 Rfam 数据库版本 13.0[4]，采用附录 B 中描述的协议创建了基准数据集 Rfam - Learn。本文使用的所有数据集都在附录 D 中详细列出，但我们注意到，所有方法都是使用论文新引入的训练集和验证集 (Rfam - Learn - Train 和 Rfam - Learn - Validation) 进行优化的。

5 实验结果分析

此表为以下方法 MCTS - RNA、antaRNA、RL - LS、RNAInverse、LEARNA、Meta - LEARNA 和 Meta - LEARNA - Adapt 在文献 (Eterna100 和 Rfam - Taneda) 的两个基准以及论文新引入的基准 (Rfam -学习-测验) 上求解的靶标结构的比例。如果在任何一个评估运行中找到了解决方案，则目标结构被视为已解决。

Method	Solved Sequences [%]		
	Eterna100	Rfam-Taneda	Rfam-Learn-Test
MCTS-RNA	57	79	97
antaRNA	58	66	**100**
RL-LS	59	62	62
RNAInverse	60	59	95
LEARNA	67	79	97
Meta-LEARNA	**68**	**83**	**100**
Meta-LEARNA-Adapt	**68**	**83**	99

图 2: 实验结果图表

5.1 主要基准分析

5.1.1 Eterna100

Eterna100 针对高达 68% 的目标结构，论文的所有方法在 Eterna100 基准上取得了明确的最新结果。此外，Meta - LEARNA 仅需约 25 秒即可达到任何其他方法 (≈1765 × 更快) 的最终性能，并在不到 30 秒的时间内获得最新的结果。这一表现在所进行的 5 次评估中都是稳定的。值得注意的是，论文的方法的所有版本在每个单独的评估运行中已经取得了新的最先进的性能。

5.1.2 Rfam - Taneda

对于 Rfam - Taneda 基准，经过 110 秒 (≈2 × 更快) 后，LEARNA 与 MCTS - RNA 的最新结果相当。Meta - LEARNA 和 Meta - LEARNAAdapt 在不到 5 秒的 (≈63 × 快) 内实现了这一先前的最先进的性能，在 400 秒和 90 秒后分别实现了新的最先进的结果 (见附录 J)，解决了 83% 的目标结构。

5.1.3 Rfam −Learn - Test

Meta - LEARNA 和 antaRNA 均能解出所有的靶结构 (分别在 29 分钟和 20 分钟)。除 RL - LS 外，所有算法均能求解至少 95% 的目标结构。

5.2 结果分析总结

综上所述，论文提出的新的深度强化学习算法在三个基准测试集上都取得了最好的性能，同时在两个基准测试集上比所有其他算法都快得多。论文的元学习方法 Meta - LEARNA 学习了 RNA 设计底层动力学的表示，并且能够将这些知识迁移到新的 RNA 设计任务中。正如论文在附录 H 中的额外分析所显示的，它也比现有的方法具有更好的序列长度。

6 总结与展望

人工智能与 RNA 设计相结合是一个新兴的研究领域，它利用人工智能技术来预测、设计和优化 RNA 的三维结构和功能。这种方法可以为生物医学、药物开发和基因编辑等领域提供新的工具和机遇。本文首先将 RNA 设计建模为一个决策过程，然后提出了几种策略来产生学习端到端设计 RNA 的智能体。本论文针对 RNA 设计问题提出了深度强化学习算法 LEARNA，以端到端的方式依次构建候选解。在实现过程中的尚有几点不足的地方，比如在训练可视化方面做的工作还是不够充分，没有很直观方便的图形界面来展示模型是如何被训练的；还有在实现过程中，由于自身实力方面的限制，对论文发现的创新点不够等等。将人工智能与 RNA 设计相结合具有极其重要的意义，例如，将来可以利用人工智能技术来发现和开发新的 RNA 药物，特别是针对抗生素耐药性等难治性疾病的 RNA 药物，也可以利用人工智能技术来提高基因编辑的效率和精度，特别是针对复杂的多基因疾病的基因编辑，还可以利用人工智能技术来探索和理解 RNA 的多样性和复杂性，特别是非编码 RNA 的功能和调控机制等等。

参考文献

[1] VANDIVIER L E, ANDERSON S J, FOLEY S W, et al. The conservation and function of RNA secondary structure in plants[J]. Annual review of plant biology, 2016, 67: 463-488.

[2] RUNGE F, STOLL D, FALKNER S, et al. Learning to design RNA[J]. arXiv preprint arXiv:1812.11951, 2018.

[3] LEMKE C, BUDKA M, GABRYS B. Metalearning: a survey of trends and technologies[J]. Artificial intelligence review, 2015, 44: 117-130.

[4] KALVARI I, ARGASINSKA J, QUINONES-OLVERA N, et al. Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families[J]. Nucleic acids research, 2018, 46(D1): D335-D342.