关于基于图注意力自编码器的转录调控关系预测模型 GraphTGI 的复现

易佳乐

2023.12.10

摘要

本文复现的论文提出了一种名为 GraphTGI 的转录调控关系预测模型，该模型基于图注意力自编码器。与传统方法通过生物实验获取转录调控关系或者基于机器学习与深度学习技术预测的方法相比，GraphTGI 考虑到转录调控网络的拓扑结构中蕴含丰富而有用的信息，从而有效提高了预测性能。进行了对比测试，评估了不同图神经网络作为编码器的性能，并进行了相关案例研究分析。实验证明了模型在准确性、有效性和可靠性方面的优越性。

关键词：基因调控网络；转录调控；图神经网络；

1 引言

基因是生物体内携带遗传信息的基本物质单位。基因通过转录生成 RNA 或蛋白质，决定了生物个体的性状表型和生理健康情况。转录调控是调节基因表达的细胞发育和稳态的重要过程。基因转录表达时受到其他因子调节的过程称为转录调控过程，其中的关键因子为转录因子。转录因子 TF 作为转录调控的基本要素，通过与 DNA 结合域的启动子或增强子结合，上调或下调靶基因的转录表达。众多基因间的相互调控关系形成了一个复杂的基因转录调控网络，研究转录调控网络有助于我们更深刻地理解生命的机制，同时也能为我们在研究复杂疾病的靶向药物等问题上提供有力的帮助。

除了通过 EMSA(电泳迁移迁移测定) [1]、ChIPseq(染色质免疫沉淀测序) [2] 和 DAP-seq(DNA 亲和纯化测序) [3]。等生物学技术获取转录调控关系外，也出现了一批基于机器学习与深度学习技术来预测转录调控关系的方法。然而，这些方法并未考虑到转录调控网络的拓扑结构之中也蕴藏了丰富有用的信息，与完整的调节网络相比，其规模仍然是冰山一角。越来越多的证据表明，生物体的表型缺陷和复杂疾病往往与异常的调控网络有关，检测 TF 与靶基因之间的关系已迅速成为生物信息学和生物医学科学的重要研究课题。

本文复现的论文 [4] 提出了一种高效的基于图神经网络的模型，用于学习转录调控网络的拓扑结构信息，进而预测转录调控关系。该工作首先构造了一个转录调控网络图数据，并为图上的基因节点赋予化学特征和序列特征。接着，GraphTGI 从该图数据中学习其拓扑结构的模式信息和基因本身的属性信息，生成节点的嵌入表达，并利用这些嵌入表达对潜在的转录调控关系进行预测。该模型还使用了自注意力机制来更好地区分不同调控关系对同一个基因节点的影响。

2 相关工作

2.1 基于生物实验技术的转录调控关系

目前存在三种常见的生物实验技术，用于获取基因间的转录调控关系，分别是电泳迁移率变动分析技术（Electrophoretic Mobility Shift Assay，EMSA），染色质免疫沉淀测序技术（Chromatin Immunoprecipitation Sequencing，ChIP-seq）和 DNA 亲和纯化测序技术（DNA Affinity Purification Sequencing，DAP-seq）。这三种方法均通过检测蛋白质与 DNA 片段之间的相互作用，即转录因子是否与靶基因结合，来深入研究它们之间的调控关系。

- 电泳迁移率变动分析技术（EMSA）：EMSA 是一种经典的实验方法，通过观察蛋白质与 DNA 结合后的电泳迁移率变动，来确定蛋白质与 DNA 结合的特异性。该技术可定量测量转录因子与 DNA 结合的强度和亲和力。

- 染色质免疫沉淀测序技术（ChIP-seq）：ChIP-seq 结合了染色质免疫沉淀和高通量测序技术，能够全面揭示转录因子与染色质之间的相互作用。通过使用特定抗体沉淀转录因子结合的染色质片段，随后的测序分析可以鉴定这些区域，进而识别调控的基因。

- DNA 亲和纯化测序技术（DAP-seq）：DAP-seq 利用 DNA 亲和纯化技术结合高通量测序，直接测定转录因子与基因组中的 DNA 结合。这种方法避免了使用抗体，使其适用于大规模的高通量实验，并提供了高分辨率的转录因子结合位点信息。

这三种技术在研究转录调控关系中各有优势，选择合适的方法取决于研究的具体需求和条件。EMSA 适用于定量分析蛋白质与 DNA 的结合亲和力，ChIP-seq 则适用于全基因组水平的调控位点鉴定，而 DAP-seq 则在不使用抗体的情况下实现高通量的转录因子结合位点测定。使用生物实验技术可以较可靠地获取转录因子与基因间的调控关系，是构建转录调控网络地面真值（ground truth）的传统方法，但存在实验时间久，实验成本高的问题，难以进行大规模检测。

2.2 基于机器学习方法的转录调控关系

基于机器学习 [5] 的转录调控关系研究借助计算模型和算法，致力于通过分析基因表达和转录因子结合等数据，揭示基因与转录因子之间的复杂关系。首先，收集并预处理生物实验数据，如 ChIP-seq 和基因表达数据。然后，通过特征提取，提取有关基因表达水平、转录因子结合位置和强度等方面的特征信息。接下来，选择合适的机器学习算法，如支持向量机、决策树或神经网络，建立预测模型。模型训练阶段利用训练集调整参数，以更好地拟合数据。通过测试集对模型进行评估，考察其准确度、精确度等性能指标。最后，解释模型结果，理解关键特征的影响，并通过可视化方法呈现研究成果。该方法不仅有助于理解基因调控网络的复杂性，还为预测新的转录调控关系提供了计算工具。通过不断优化和改进模型，能够更准确地预测基因与转录因子之间的相互作用，为生物学研究提供有力支持。

基于传统机器学习的转录调控关系预测算法为构建基因调控网络提出了许多有建设性的见解。但这些方法一般采用人工设计的函数来对数据进行分类或者回归拟合，无法很好地模拟复杂过程。

2.3 基于深度学习方法的转录调控关系

基于深度学习 [6] 的转录调控关系预测算法是利用深度神经网络模型对基因表达和转录因子结合等大规模生物数据进行学习和建模，以揭示基因与转录因子之间的复杂关联。通过多层次的神经网络结构，该方法能够自动地提取抽象层次的特征，更全面地捕捉基因调控网络的复杂性。这些深度学习模型可以逐渐优化权重和参数，以适应不同的转录调控模式，从而提高预测性能。这种方法在处理高维度、非线性和大规模生物数据方面展现了优越性，为准确预测基因与转录因子之间的关系提供了有力工具，有望推动生物医学研究的深度理解。

尽管基于深度学习方法的转录调控关系预测算法在许多方面表现出色，但也存在一些挑战和缺点。首先，深度学习模型通常需要大量的标记数据进行训练，而在生物学领域获取高质量标记数据可能较为困难和昂贵。其次，模型的复杂性可能导致过度拟合，特别是在数据集较小或噪音较多的情况下，可能降低其泛化能力。此外，深度学习模型的黑盒性质使得解释模型的预测结果和理解模型的内在机制变得更为复杂。另外，算法的计算需求较高，需要强大的计算资源和显著的时间成本。

3 本文方法

3.1 本文方法概述

现有的预测转录因子 TF -靶标相互作用的计算方法大多是预测结合位点，而不是直接预测相互作用。为此，考虑到转录因子 TF 与靶基因之间未观察到的相互作用可以通过学习已知的相互作用模式来预测，建立了一个新的空间图卷积模型来计算构建图中每个 TF 目标对交互的可能性。GraphTGI 模型考虑到该可能性，提出的潜在 TF 与靶基因相互作用的图自编码器模型 GraphTGI。GraphTGI 不仅学习所有基因节点的信息，同时还学习基因转录调控网络的全局拓扑信息，更好地模拟了转录调控网络的调控过程。此外，模型借助自注意力机制，为每一对基因间的关系边赋予注意力权重，节点可以根据该注意力权重来判断接收到信息的重要程度。

GraphTGI 模型由多个基于图注意力的编码器层和一个双线性解码器组成。首先从 TRRUST 数据库中构造了一个转录调控网络图数据，并为图上的基因节点赋予化学特征和序列特征。接着，采用基于图注意力的编码器，从该图数据中学习其拓扑结构的模式信息和基因本身的属性信息，生成节点的嵌入表达，并利用这些嵌入表达采用双线性解码器重构图。使用交叉熵函数端到端训练整个模型，对潜在的转录调控关系进行预测。

3.2 数据集

高通量生物实验技术的发展使我们积累了许多先验的转录调控知识，这使得构建转录调控网络的地面真值成为了可能，这种网络的地面真值可以用来评估和改进基于图卷积网络的转录调控关系预测模型，帮助我们更好地理解转录调控网络的结构和功能。

3.2.1 转录因子节点和靶基因节点组成的异质图

使用 TRRUST（Transcriptional Regulatory Relationships Unraveled by Sentence-based Text mining，https://www.grnpedia.org/trrust/）[7] 数据库来构建模型训练所需的图数

据。TRRUST 是一个人工管理的记录了人类转录调控网络数据的数据库，提供了人类基因调控网络中最全面的转录因子-靶基因相互作用信息。在数学上，这些交互数据自然地形成一个图，节点和链接分别表示基因和基因间的互相作用关系，其中基因节点包含了靶基因和转录因子所对应的源基因。TRRUST 还提供了每对转录因子-靶基因间的调控模式信息。

3.2.2 化学物-基因相互作用关系

环境化学物质可能会对转录因子与靶基因之间的相互作用机制产生影响，从而对人类健康产生影响，所以化学物-基因相互作用关系能为预测潜在转录调控关系提供有价值的信息。比较毒理学基因组学数据库（Comparative Toxicogenomics Database, CTD, http://ctdbase.org/）[8] 旨在集成毒理学、基因组学和疾病相关信息，以便深入研究化学物质与基因之间的相互作用和其在疾病发展中的作用。CTD 聚焦于整合多个资源的数据，包括文献注释、基因-化学物质相互作用、基因-疾病关联等，为研究人员提供了一个综合的平台，用于探索化学物质对基因和疾病的潜在影响。通过其用户友好的界面和多层次的数据查询功能，CTD 数据库为毒理学研究和药物开发领域提供了有价值的资源。因此，从 CTD 中收集了化学物-基因相互作用，但 GraphTGI 并不直接使用这些关系构建图数据，而是采用计算相似度的方法将其转化为转录调控网络中基因节点的特征。

3.2.3 转录因子对应的基因和靶基因的 DNA 序列

国家生物技术信息中心 (National Center for Biotechnology Information，NCBI，https://www.ncbi.nlm.nih.gov/) 是我国生物信息学领域的重要机构，致力于推动生物技术和生物信息学的发展与应用。作为国家科技部直属单位，NBIC 在生物信息资源整合、数据管理、生物信息技术研发等方面发挥着关键作用。该中心提供包括基因组学、蛋白质组学、生物信息学工具等在内的多方面服务，并积极支持科研、产业界和政府决策的需求，为推动我国生命科学研究和创新提供了重要的技术和信息支持。从 NCBI 下载转录因子对应的基因和靶基因的 DNA 序列，利用 DNA 序列来进一步全面描述每个基因节点的特征。

3.3 GraphTGI 模型框架

论文中将转录因子-靶基因调控关系的预测任务定义为一个链接预测任务，以确定一对转录因子和靶基因间是否存在相互作用。链接预测任务将在一个由转录因子节点和靶基因节点构成的图上执行，分别使用 $V_T = \{V_{T1}, \ldots, V_{Tm}\}$ 和 $V_G = \{V_{G1}, \ldots, V_{Gn}\}$ 表示转录因子节点集合和靶基因节点集合，模型的目标是学习一个映射函数 $\Phi(V_T, V_G) \to [0, 1]$，从每一对节点对之间得到一个概率分数。

论文提出了一个名为 GraphTGI 的模型，其基本假设是通过学习已知转录关系的数据模式，可以预测转录因子和其靶基因之间潜在的相互作用。在转录调控网络上，具有相似序列特征和化学特征的转录因子更可能和类似的靶基因产生调控关系。图 1描述了如何通过 GraphTGI 模型获取指定转录因子的潜在靶基因，GraphTGI 模型使用了自编码器框架，共包含了三个部分：（1）序列相似度特征和化学性质相似度特征计算模块，从 TRRUST 数据库、CTD 数据库和 NCBI 数据库中构建转录调控网络并为节点添加化学特性/序列相似度特征；（2）基于自注意力机制图卷积网络的编码器，将节点的原始特征映射到一个共享的嵌入

空间，用于学习图数据中的拓扑信息和节点特征信息并生成节点的嵌入表达；（3）用于从节点嵌入表达中重建转录调控网络的双线性解码器。同时，模型使用交叉熵损失函数计算损失并实现端到端的训练。

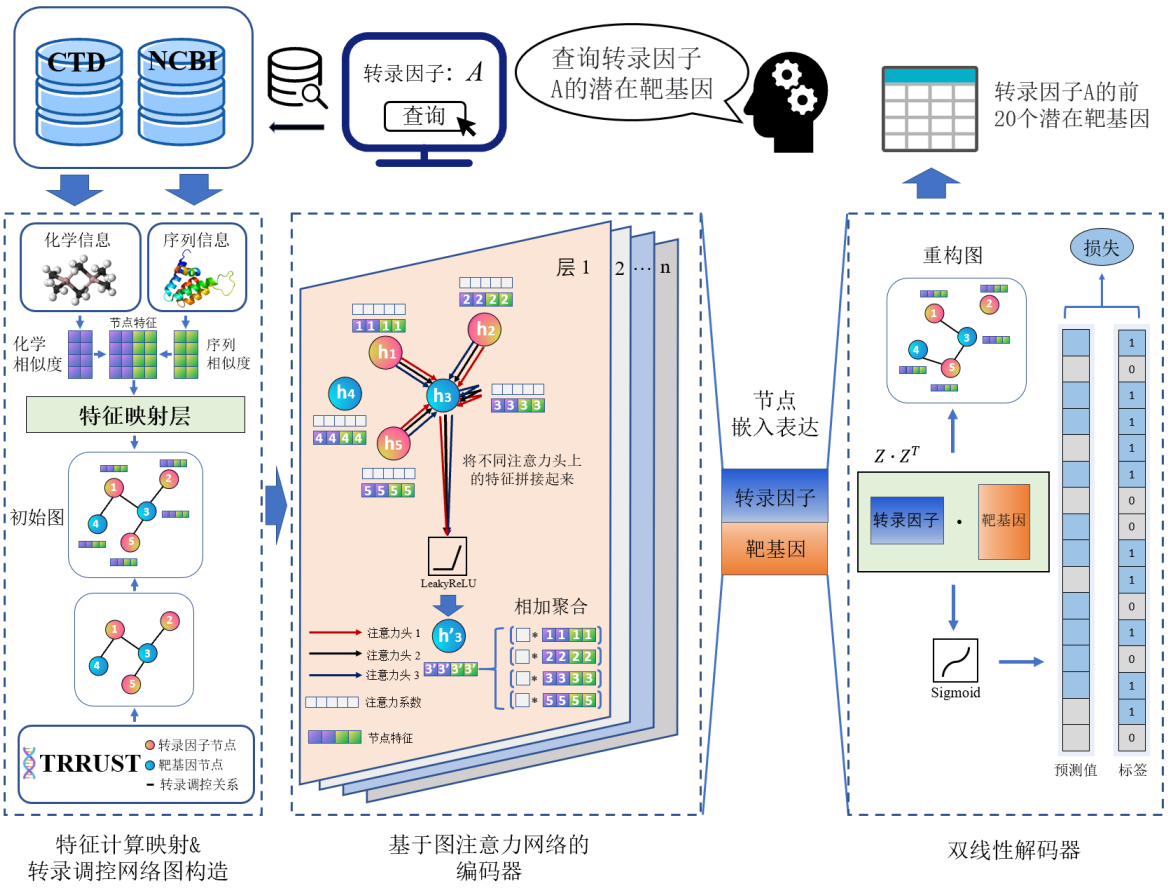

图 1. 用于预测转录因子和靶基因之间相互作用的 GraphTGI 模型示意图

3.4 节点特征构建

对于每个基因节点，使用 $C_i = C_1, \ldots, C_{Nc}$ 来表示其与所有化学物的关联情况，N_c 代表化学物的数量，$C_j = 1$ 表示第 i 个基因和第 j 个化学物之间存在关联，$C_j = 0$ 则表示没有关联。计算了所有基因化学关联向量间的余弦相似度，并存储在化学相似度矩阵 $CSM \in R^{N \times N}$ 中，其中 N 表示节点的数量，$CSM_{i,j}$ 表示第 i 个节点和第 j 个节点间的化学相似度。具体计算公式如下：

$$CSM_{i,j} = \frac{C_i \cdot C_j}{\|C_i\| \times \|C_j\|}$$

针对基因的 DNA 序列信息，使用全局对齐算法计算其两两之间的序列相似性。具体来说，对序列信息 S_a 和 S_b，如果 $S_a(i) = S_b(j)$，则认为两个序列在这个位置成功匹配，反之则是不匹配，并且在不匹配时在 $S_b(j)$ 处添加一个间隔。全局对齐算法通过计算匹配和间隔的次数按不同权重进行计分，计算公式为：

$$score(S_a, S_b) = \frac{(n_{\text{match}} \times S_{\text{match}}) - (n_{\text{gap}} \times S_{\text{penalty}})}{len(S_a)}$$

其中 n_{match} 和 n_{gap} 分别是序列中元素匹配和出现间隔的次数，S_{match} 和 S_{penalty} 分别是匹配时的得分和出现间隔时的惩罚分数，$len(S_a)$ 表示 S_a 的长度。通过计算得到序列相似度矩阵 $CSM \in R^{N\times N}$，$CSM_{i,j} = score(S_i, S_j)$ 表示第 i 个节点和第 j 个节点之间的序列相似度。

3.5 基于自注意力图卷积网络的编码器

自注意力图卷积网络（Self-Attention Graph Convolutional Networks）的编码器采用了自注意力机制，旨在处理图形结构数据。通过自注意力，模型可以动态地分配权重给不同节点的邻居，从而更好地捕捉节点之间的复杂关系。这种编码器能够在图数据中学习到具有变化权重的节点表示，使其适用于各种图形结构任务，如图分类和节点分类。

GraphTGI 模型假设相互作用的节点更有可能具有相关的特征，因此模型旨在学习节点特征之间的相关性。为了使模型能够充分挖掘出转录因子-靶基因相互作用间丰富的信息并能推广应用于完全未知的图上，GraphTGI 采用了图注意力编码器，基于直接邻居节点信息生成每个节点的嵌入表达。

输入 GraphTGI 模型的图数据是一个包含了两类节点的异构图。为了增强初始特征的表达能力，模型采用全连接层将两类节点的特征映射到同一维度的高维向量空间中。使用 $u = \vec{u_1}, \ldots, \vec{u_m}$ 来表示转录因子节点的初始特征向量，其中 $\vec{u_i} \in R^c$，代表了第 i 个转录因子节点的初始特征；使用 $v = \vec{v_1}, \ldots, \vec{v_n}$ 来表示靶基因节点的初始特征向量，其中 $\vec{v_j} \in R^d$，代表了第 j 个靶基因节点的初始特征。本工作按如下公式将它们分别映射到 e 维向量空间中：

$$\dot{u} = (W_{TF} \cdot u) + b_1 \tag{1}$$

$$\dot{v} = (W_{tg} \cdot v) + b_2 \tag{2}$$

其中 $W_{TF} \in R^{e\times c}$ 和 $W_{tg} \in R^{e\times d}$ 分别为作用于转录因子节点特征和靶基因节点特征的转换矩阵。

3.6 双线性解码器

双线性解码器是一种用于处理多模态学习任务的深度学习结构，通过同时考虑多个输入模态之间的交互关系，有助于提取丰富的跨模态特征表示。该解码器结合了线性层和非线性操作，能够有效捕捉不同输入模态之间的复杂关联，从而提高任务性能，例如图像与文本之间的联合学习或多源信息融合。通过引入双线性项，这种结构能够建模输入模态之间的高阶关系，为多模态数据的表示学习提供了强大的工具。

GraphTGI 模型使用一个双线性解码器，从编码得到的嵌入特征中解码重构出转录调控网络。具体来说，模型使用一个转换矩阵将所有转录因子-靶基因节点对的嵌入特征融合，再引入 sigmoid 函数将融合后的复合特征映射到 [0,1] 之间，一次预测这对转录因子-靶基因节点间是否存在链接。

3.7 评价指标

为了评估模型的预测性能，使用 K 折交叉验证（K=2,5,10）测试了 GraphTGI 模型的性能。所有的样本被随机分配到个大致相等的分组中，K 折交叉验证会依次将每个分组中的数

据作为测试集，其他所有分组中的数据则作为训练集。为了量化 K 折交叉验证的性能，引入了四个常见的评价指标，包括：准确度（accuracy），精密度（precision），召回率（recall），F1 分数（F1-score），它们的计算公式分别如下：

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{F1-Score} = \frac{2 \times P \times R}{P + R} \tag{4}$$

其中 TP 和 TN 分别代表了被正确预测的正样本和负样本的数量，FP 和 FN 分别表示了被错误预测的正样本和负样本的数量，P 表示精密度分数，R 表示召回率分数。这些指标的得分值越大表示模型的性能越好。事实上，由于很难通过生物实验确保一对基因间不存在调控关系，所以并没有数据能够用于描述负样本。为了对模型进行性能评估，本工作采用随机采样的方法，每一次训练都从无标签数据中随机采样出和正样本集同样数量的样本作为负样本集。由于转录调控网络的稀疏性，大多数转录因子和基因间并不存在调控关系，因此可以认为随机采样得到的样本中大多数是真实的负样本，只有少量是定义错误的负样本，错误负样本的数量小到可以忽略不计其对模型预测结果的影响。

4 复现细节

4.1 与已有开源代码对比

本文复现了 GraphTGI 模型，原文使用不同的图神经网络代替原编码器时模型的整体测试性能，这些图神经网络包括：GraphSage，TAG（Topology Adaptive Graph Convolutional），GIN（Graph Isomorphism Network）以及另一种带注意力机制的图卷积网络 AGNN（Attention-based Graph Neural Network）。在该对比实验中，模型统一使用拼接策略构建输入特征，模型输出的节点嵌入表达的维度设为 32。为了保证公平性，避免使用单一随机种子对模型性能的影响，使用不同的随机种子对每一个图神经网络分别做五次测试，并记录其平均值和标准差。本文在此基础上测试了另一种图神经网络 NNConv(Neural Network Convolution)，其他实验设置不变。此外，原文使用了 2、5、10 折交叉验证，本文使用 2、4、6 折交叉验证探究了不同的划分以及在 4 折交叉验证下不同 embedding-size 大小对于 GraphTGI 性能的影响。

4.2 数据集

TRRUST 提供了每对转录因子-靶基因间的调控模式信息，包括激活（33.5%），抑制（20.5%）和未知（46%），此处的未知数据表示基因间存在调控关系，但并不清楚该调控关系属于激活还是抑制。然而，由于 GraphTGI 仅聚焦于预测基因间是否存在转录因子-靶基因关系，而不关心这些关系的类型，所以这些信息暂未使用。GraphTGI 将所有“激活”，“抑制”和“未知”的关系都视为同一种关系，即“存在关系”，具体到模型中体现为使用同一个标签

表示这些数据，而不使用三个标签来表示。值得注意的是，由于转录调控网络的复杂性，在不同的转录调控过程中，同一对基因或许会以不同的转录模式进行交互，因此在 TRRUST 中部分数据既属于激活类型，又属于抑制类型。对于这些具有多个注释类型的转录因子-靶基因对，本文将其定义为冗余数据，并只保留一个相关数据，删除其余的数据。此外，一一对比了 TRRUST 和比较毒理学基因组学数据库（CTD）中的基因，只保留了在两个数据库中成功匹配的部分。最终构建了一个具有 3400 条已经湿实验验证的转录因子-靶基因调控关系的网络，囊括了 657 个的转录因子和 1780 个靶基因。

CTD 数据库记录了 9516 种化学物质和 11125 个基因间的 124344 条基因-化学物关系，平均为每个基因描述了 11 种不同类型的化学分子。GraphTGI 并不直接使用这些关系构建图数据，而是采用计算相似度的方法将其转化为转录调控网络中基因节点的特征。因此，在 CTD 数据库中，只有和 TRRUST 数据库中的基因成功匹配的数据才得以保留。

4.3 实验环境搭建

CPU:12th Gen Intel(R) Core(TM) i5-12500 3.00 GHz 机带 RAM 8.00 GB
选择深度学习框架 mxnet 作为图神经网络 dgl 的后端

环境	版本
python	3.7.0
dgl	0.6.0.post1
mxnet	1.7.0.post2
pytorch	1.13.1
matplotlib	3.3.4
numpy	1.21.6
pandas	1.1.5
scikit-learn	1.0.2
scipy	1.6.2

4.4 复现过程

4.4.1 采样

采样的目的是为了构建图的时候提供每条边的权重信息。具体来说，采样的逻辑是从已知的 TF-TG 关联中随机采样相同数量的未知关联作为负样本，然后将这些采样的正负样本组合成一个数据框。这样，构建图时可以通过这些权重信息来反映 TF 和 TG 之间的相互作用关系的强度。

```
# 读取包含 TF-tg 关联的 CSV 文件
all_associations = pd.read_csv('./data/all_TF_tg_pairs.csv',
                               names=['TF', 'tg', 'label'])

# 获取所有正样本
```

```
known_associations = all_associations.loc[
                                    all_associations['label'] == 1]

# 随机获取相同数量的负样本
unknown_associations = all_associations.loc[
                                    all_associations['label'] == 0]

random_negative = unknown_associations.sample(
n=known_associations.shape[0], random_state=random_seed, axis=0)

# 合并正负样本并重置索引
sample_df = known_associations.append(random_negative)

sample_df.reset_index(drop=True, inplace=True)
```

4.4.2 构建图

初步构建一个基于 DGL（Deep Graph Library）的图，其中包括转录因子 TF 节点、目标基因节点，以及它们之间的边。每个节点有一些特征，这些特征包括了转录因子和目标基因的信息。图的边表示转录因子和目标基因之间的相互作用关系。步骤如下：

(1) 初始化图

(2) 构建节点特征

(3) 构建边

(4) 将图设置为只读，在构建之后不再进行修改

```
# 初始化图 g1，初始时不包含节点和边
g1 = dgl.DGLGraph(multigraph=True)

# 构建节点特征
TF_data = nd.zeros(shape=(g.number_of_nodes(), TFSM.shape[1] +
                      TFSM_2.shape[1]), dtype='float32', ctx=ctx)

TF_data[:TFSM.shape[0], :TFSM.shape[1]] = nd.from_numpy(TFSM)

TF_data[:TFSM.shape[0], TFSM.shape[1]:TFSM.shape[1] +
                          TFSM_2.shape[1]] = nd.from_numpy(TFSM_2)

TF_data[TFSM.shape[0]: TFSM.shape[0] + tgSM.shape[0],
```

```
                        :tg_TF_SM.shape[1]] = nd.from_numpy(tg_TF_SM)

TF_data[TFSM.shape[0]: TFSM.shape[0] + tgSM.shape[0],
    tg_TF_SM.shape[1]:tg_TF_SM.shape[1] + tg_TF_SM_2.shape[1]]
                                        = nd.from_numpy(tg_TF_SM_2)

g.ndata['TF_features'] = TF_data

#构建边
g.add_edges(sample_TF_vertices, sample_tg_vertices,data={'inv':
    nd.zeros(samples.shape[0], dtype='int32', ctx=ctx),'rating':
    nd.from_numpy(samples[:, 2].astype('float32')).copyto(ctx)})

#将图设置为只读，以避免在训练过程中意外修改图的结构
g.readonly()
```

4.4.3 图编码与解码

图编码就是对 TF 节点和目标基因节点进行特征投影，然后将这个特征投影应用到图 G 上，然后再循环遍历将每一层的计算都应用到图 G 上。GATConv 是利用注意力机制来为每个节点分配不同的权重，这有助于网络更好地理解图中的局部结构，从而更有效地学习图的表示。使用一个双线性解码器，从编码得到的嵌入特征中解码重构出转录调控网络。

```
# 对TF（转录因子）节点的输入特征进行投影
class TFEmbedding(nn.Block):
    def __init__(self, embedding_size, dropout):
        super(TFEmbedding, self).__init__()
        # 创建一个序列模块seq用于添加全连接层和dropout层
        seq = nn.Sequential()
        with seq.name_scope():
            seq.add(nn.Dense(embedding_size, use_bias=True))
            seq.add(nn.Dropout(dropout))
        self.proj_TF = seq

    def forward(self, ndata):
        # 对TF节点的输入特征进行线性映射和dropout处理
        extra_repr = self.proj_TF(ndata['TF_features'])
        return extra_repr

# 进行多层图注意力机制的计算 使用两层GATConv
self.layers.add(
```

```
    GATConv(
        embedding_size, # 每个节点的输入维度（嵌入大小）
        embedding_size, # 每个节点的输出维度（嵌入大小）
        2, # 注意力头的数量
        feat_drop=dropout, # 节点特征的丢弃方法
        attn_drop=0.5, # 注意力权重的丢弃率
        negative_slope=0.5, # LeakyReLU激活函数的负斜率
        residual=True, # 是否使用残差连接
        allow_zero_in_degree=True # 是否允许节点度数为零
    )
)
self.layers.add(GATConv(embedding_size, embedding_size,
    2, feat_drop=dropout, attn_drop=0.5, negative_slope=0.5,
    residual=True, allow_zero_in_degree=True))

# 对 TF 和靶基因节点的嵌入向量进行bilinear 解码操作
class BilinearDecoder(nn.Block):
    def __init__(self, feature_size):
        super(BilinearDecoder, self).__init__()
        # 使用sigmoid激活函数
        self.activation = nn.Activation('sigmoid')
        # 获取输入特征转化为更高级特征后的权值矩阵
        with self.name_scope():
            self.W = self.params.get('dot_weights',
                    shape=(feature_size, feature_size))
```

4.4.4 构建图神经网络模型

在构建图时，要确保理解图的构建逻辑，特别是节点和边的表示，以及节点和边上的特征。在构建图时，确保边的权重反映了 TF 和 TG 之间的关联强度。

```
class GraphTGI(nn.Block):
    def __init__(self, encoder, decoder):
        super(GraphTGI, self).__init__()

        self.encoder = encoder
        self.decoder = decoder

    def forward(self, G, TF, tg):
        # 获取图G的节点特征
        h = self.encoder(G)
```

表 1. 使用不同图卷积层的 GraphTGI 模型性能对比

Model	AUC	Accuracy	Precision	Recall	F1-score
GraphTGI	0.8832	0.7967	0.8008	0.7946	0.7960
GraphSage	0.8642	0.7836	0.7915	0.7728	0.7809
TAG	0.8594	0.7780	0.7849	0.7675	0.7754
GIN	0.8534	0.7739	0.7857	0.7535	0.7690
AGNN	0.8647	0.7860	0.7983	0.7666	0.7813
NNConv	0.8575	0.7728	0.7835	0.7635	0.7743

```
11
12          # 获取TF节点和目标基因节点的特征向量
13          h_TF = h[TF]
14          h_tg = h[tg]
15
16          return self.decoder(h_TF, h_tg), G.ndata['h']
```

4.5 创新点

原文使用不同的图神经网络代替原编码器对 GraphTGI 模型的性能进行对比实验，在此基础上，本文测试了另一种图神经网络 NNConv 对模型性能的影响。此外，通过更改数据集划分以及模型的参数进一步探究了 GraphTGI 模型性的最佳性能。

5 实验结果分析

本文在原作者的基础上测试了另一种图神经网络 NNConv(Neural Network Convolution)，汇总结果如表1所示。和其他图神经网络相比，GraphTGI 模型在 AUC 上取得了至少 1.85% 的提升，在其他指标上也都取得了最佳的性能。

此外，原文使用了 2、5、10 折交叉验证，本文使用 2、4、6 折交叉验证探究了不同的划分对于 GraphTGI 性能的影响。2、4、6 折交叉验证下 GraphTGI 模型的 ROC 曲线如图2、图3和图4所示，结果汇总如表2所示。

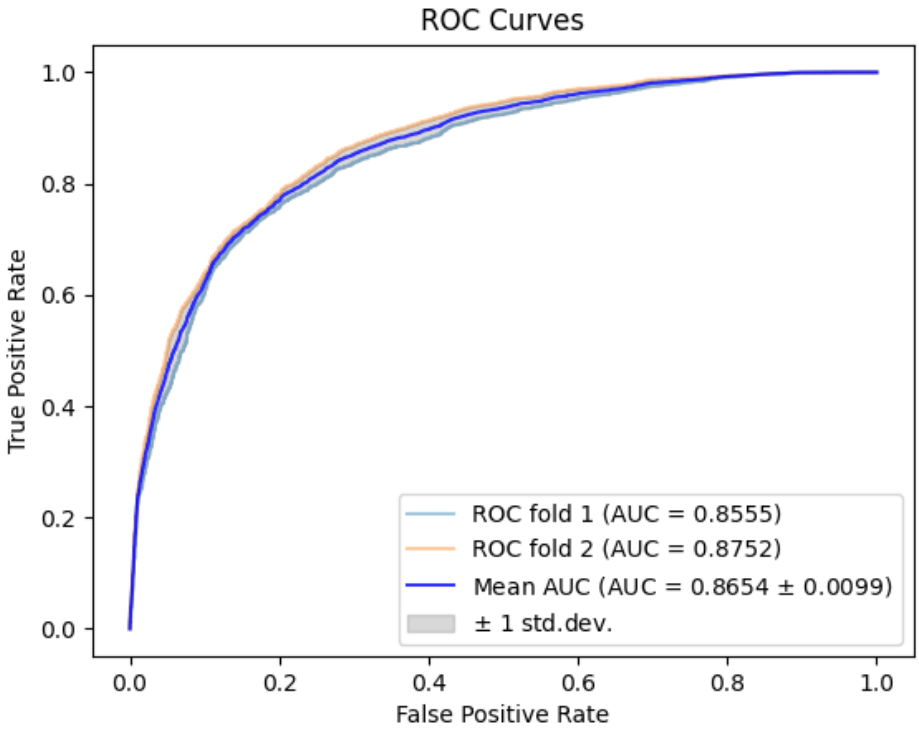

图 2. 二折交叉验证下 GraphTGI 模型的 ROC 曲线

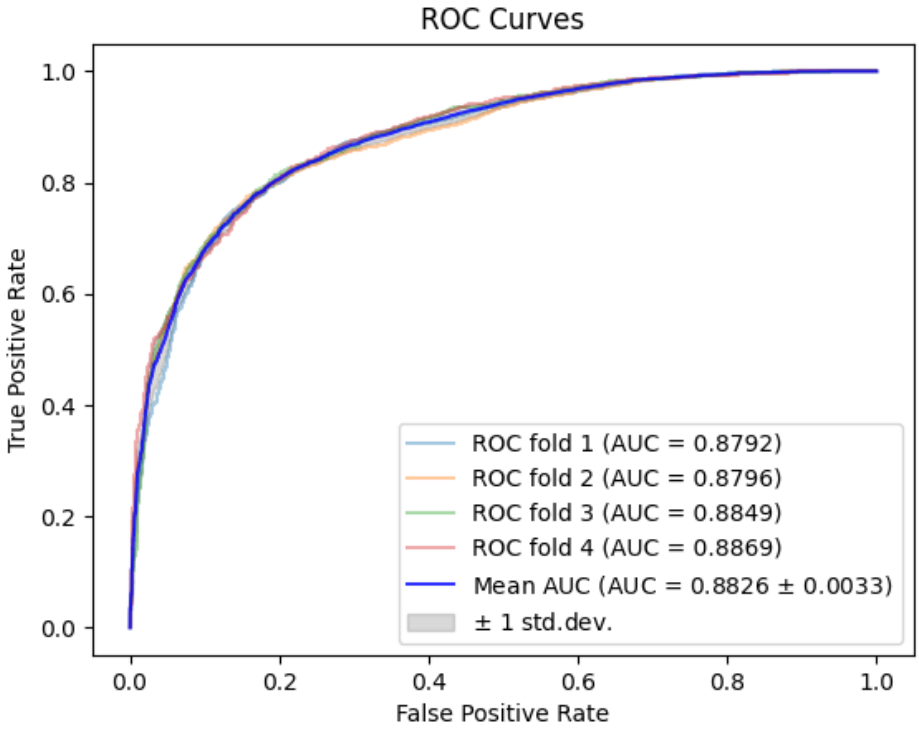

图 3. 四折交叉验证下 GraphTGI 模型的 ROC 曲线

表 2. 不同 K 折交叉下验证中测试 GraphTGI 性能

K	AUC	Accuracy	Precision	Recall	F1-score
2	0.8654	0.7835	0.7975	0.7599	0.7782
4	0.8826	0.7988	0.7935	0.8097	0.8009
6	0.8840	0.8015	0.7879	0.8272	0.8063

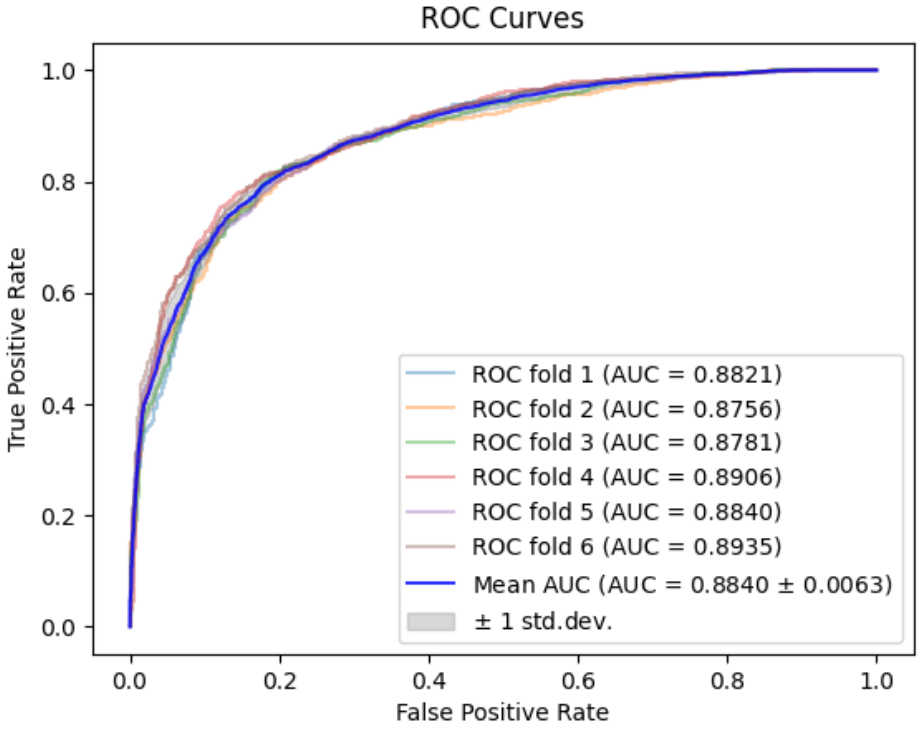

图 4. 六折交叉验证下 GraphTGI 模型的 ROC 曲线

在 4 折交叉验证下, 不同 embedding-size 大小对于 GraphTGI 性能的影响。四折交叉验证下，embedding-size 大小分别为 8、16、32 下，GraphTGI 模型的 ROC 曲线如图5、图6和图7所示，结果汇总如表3所示。

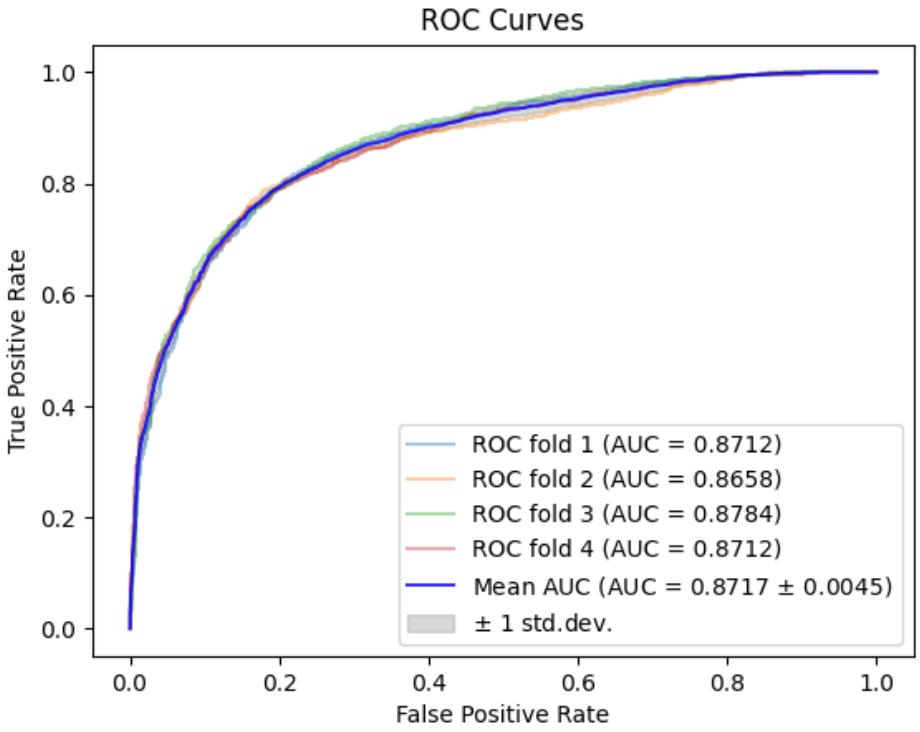

图 5. embedding-size 大小为 8 的 GraphTGI 模型的 ROC 曲线

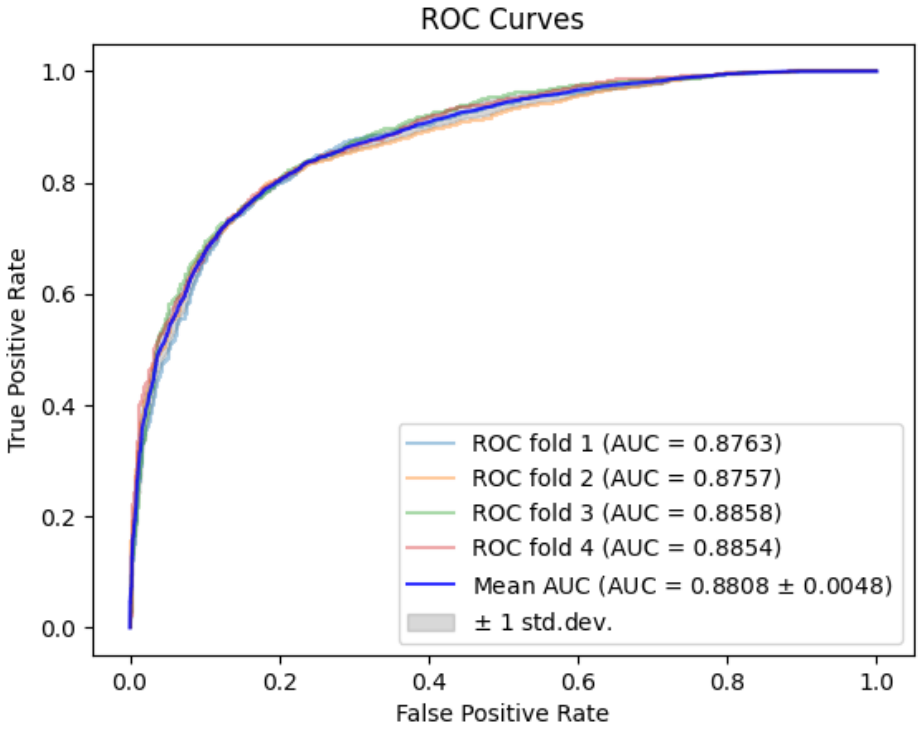

图 6. embedding-size 大小为 16 的 GraphTGI 模型的 ROC 曲线

表 3. 四折交叉验证，不同嵌入大小下的 GraphTGI 性能

size	AUC	Accuracy	Precision	Recall	F1-score
8	0.8717	0.7943	0.8102	0.7716	0.7892
16	0.8808	0.8020	0.7977	0.8097	0.8034
32	0.8826	0.7988	0.7935	0.8097	0.8009

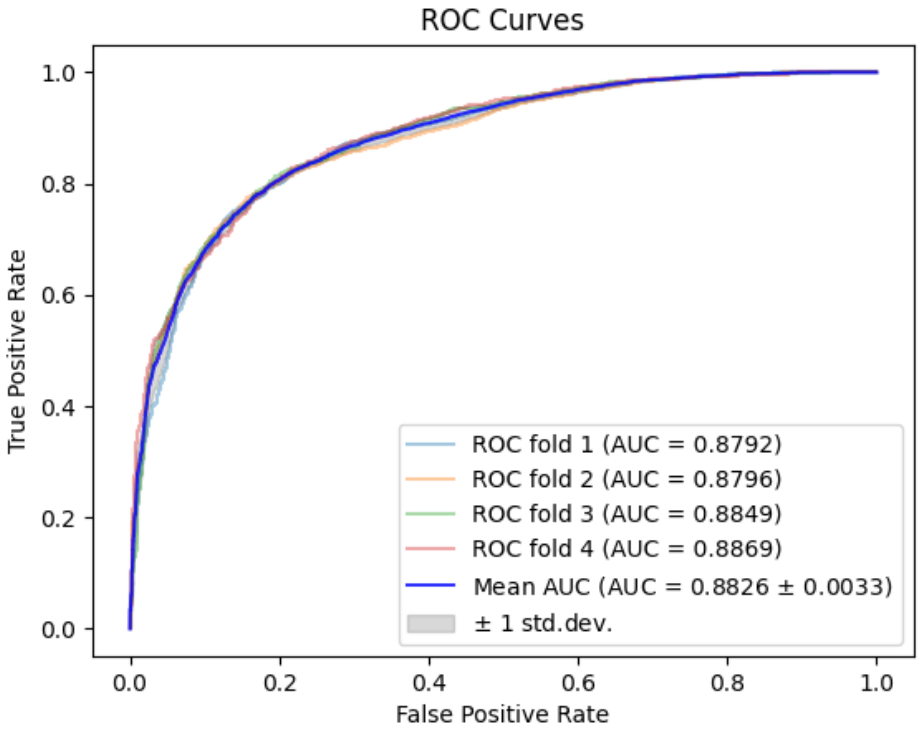

图 7. embedding-size 大小为 32 的 GraphTGI 模型的 ROC 曲线

6 总结与展望

如今，预测转录调控相互作用关系及其模式仍然是一个重要的基础性挑战。转录调控通过调节基因的表达，从而影响生物体的发育、分化和应激反应等生命过程。研究转录调控网络有利于人们更好地理解生命的各项机制，帮助揭示严重复杂疾病产生的原因，并发掘新的治疗靶点等。

首先对转录调控的作用及研究转录调控关系的意义进行阐述，相关的工作可以分为生物实验方法，基于机器学习的预测方法和基于深度学习的预测方法三类。简要探讨了这些方法的优点和不足，进而总结了存在的可以改进的方向，然后对论文提出的基于图注意力自编码器的转录调控预测模型 GraphTGI 的框架进行介绍。因为转录调控网络的拓扑结构具有重要的信息，所以模型使用 TRRUST 数据库构造转录调控图数据，以此引入了转录调控网络的拓扑信息来提高预测性能。考虑到环境化学物对基因的影响作用和基因本身 DNA 序列含有的信息，使用基因化学相似度和基因序列相似度作为节点的特征。此外，模型借助自注意力机制，为每一对基因间的关系边赋予注意力权重，节点可以根据该注意力权重来判断接收到信

息的重要程度。使用一个双线性解码器，从编码得到的嵌入特征中解码重构出转录调控网络。最后使用 K 折交叉验证测试模型的性能并引入了四个常见的评价指标, 包括：准确度，精密度，召回率，F1 分数来量化 K 折交叉验证的性能。

GraphTGI 模型解决了过往转录调控预测任务中存在的部分问题，并取得了不错的效果。然而，仍然存在不足之处需要进一步改进，主要包括以下方面：

- 在预测性能和应用方面进行改进，以构建知识图 (KG) 作为信息挖掘的基础数据。对于构建的 KG 中的每一种类型的子图，分别建立一个单独的图神经网络来学习节点表示，然后将其整合计算概率分数进行预测。
- 尝试使用多子图卷积网络架构，将知识图中的全局信息与各子图上的唯一信息进行提取和融合。通过整合基因的多组学信息，更有效、准确地构建转录调控网络知识图谱，从而更精确地推断转录调控关系。

参考文献

[1] Lance M Hellman and Michael G Fried. Electrophoretic mobility shift assay (emsa) for detecting protein–nucleic acid interactions. *Nature protocols*, 2(8):1849–1861, 2007.

[2] Tarjei S Mikkelsen, Manching Ku, David B Jaffe, Biju Issac, Erez Lieberman, Georgia Giannoukos, Pablo Alvarez, William Brockman, Tae-Kyung Kim, Richard P Koche, et al. Genome-wide maps of chromatin state in pluripotent and lineage-committed cells. *Nature*, 448(7153):553–560, 2007.

[3] Anna Bartlett, Ronan C O'Malley, Shao-shan Carol Huang, Mary Galli, Joseph R Nery, Andrea Gallavotti, and Joseph R Ecker. Mapping genome-wide transcription-factor binding sites using dap-seq. *Nature protocols*, 12(8):1659–1672, 2017.

[4] Zhi-Hua Du, Yang-Han Wu, Yu-An Huang, Jie Chen, Gui-Qing Pan, Lun Hu, Zhu-Hong You, and Jian-Qiang Li. Graphtgi: an attention-based graph embedding model for predicting tf-target gene interactions. *Briefings in Bioinformatics*, 23(3):bbac148, 2022.

[5] Ruiqing Zheng, Min Li, Xiang Chen, Fang-Xiang Wu, Yi Pan, and Jianxin Wang. Bixgboost: a scalable, flexible boosting-based method for reconstructing gene regulatory networks. *Bioinformatics*, 35(11):1893–1900, 2019.

[6] Sirajul Salekin, Jianqiu Michelle Zhang, and Yufei Huang. Base-pair resolution detection of transcription factor binding site by deep deconvolutional network. *Bioinformatics*, 34(20):3446–3453, 2018.

[7] Heonjong Han, Jae-Won Cho, Sangyoung Lee, Ayoung Yun, Hyojin Kim, Dasom Bae, Sunmo Yang, Chan Yeong Kim, Muyoung Lee, Eunbeen Kim, et al. Trrust v2: an expanded reference database of human and mouse transcriptional regulatory interactions. *Nucleic acids research*, 46(D1):D380–D386, 2018.

[8] Allan Peter Davis, Cynthia J Grondin, Robin J Johnson, Daniela Sciaky, Jolene Wiegers, Thomas C Wiegers, and Carolyn J Mattingly. Comparative toxicogenomics database (ctd): update 2021. *Nucleic acids research*, 49(D1):D1138–D1143, 2021.