Name: Abhinit Sundar
Instructor(s): Dr. Islam and Professor Koutis
TA: Mr. Omkar Pradip Naik
Research Topic: Graphormer Model

Link to GitHub:

https://github.com/asundar0128/GraphormerModelModification

Note: the GitHub contains both the video/demo of report and the code – there is no separate link to the video – video is DeepLearningModel.mp4 and code is DeepLearningProjectCode.ipynb

Introduction

Antisense oligonucleotides, or ASO, rely heavily on gene sequencing and efficacy scores based on structural and sequence-based descriptors. ASOs are used extensively to treat diseases like cancer, genetic disorders, and viral infections – with short, single-stranded DNA or RNA molecules selectively binding against complementary sequence in target mRNA and modulating gene expression with RNase-H-mediated cleavage, steric blocking of translation, and splicing. Since experimental validation of ASO sequences is very time-consuming and expensive, deep learning techniques help improve ASO discovery by predicting efficacy directly from molecular representations. Existing approaches are too sequence-based, have poor feature engineering with handcrafted descriptors, and often rely on classical ML models, thereby having poor generalization across gene targets, lack of ability to model complex and non-linear relationships between the bases and the functions, and lack of proper support for graph-based models. Graphormer is a transformer-based graph neural network introduced with Microsoft Research and is a very powerful model for large molecular and protein graphs. Graphormer is adapted to predict ASO efficacy with the PFRED dataset to extend its functionality across single task classification, binary multitask classification, and regression. Detailed evaluation metrics like F1 Score, AUROC, confusion matrix, RMSE, and R squared scores are utilized for the classification and regression applications with Graphormer model. Graphormer has limitations currently with UNREACHABLE_NODE_DISTANCE having a maximum number of nodes that the knowledge graph structure can handle before exploding computationally. It also has not fully expanded on one task classification, binary multi-task classification, and regression since the current Graphormer libraries and functions do not have such a provision.

Model Architecture

Graphormer introduces innovative graph representation methods for learning with encoded graph structures. Traditional Laplacian eigenvectors and node degree features are not a dependency for these graph-encoded based architecture. Centrality encoding, spatial encoding, and transformer encoder are all vital to the Graphormer model with distance and connectivity captured across nodes, attention bias incorporating edge and node proximity, and a fully connected attention mechanism tailored for graphs with more than 100 nodes. Prior Graphormer architecture models could not handle a graph structure with nodes greater than

100, which this implementation attempts to address.

Each DNA sequence is transformed into a graph structure, with nodes, node features, and edges. The nodes represent the nucleotides, node features have one-hot encoding of base identity, position index, chemical modifications, and edges allow phosphodiester backbone corrections, optional pairings or secondary structure with RNA folding tools, and custom edge types for modifications like methylation and phosphorothioates.

The Graphormer encoder has multiple transformer layers adapted for graph data and differs from the standard attention-based transformers in several ways: the node embedding layer has a linear projection of initial node features into hidden dimension, spatial encoding uses the precomputed shortest path distances to attention scores with bias, centrality encoding takes the important values of each node to its attention key/query embeddings, multi-head attention uses global attention across nodes, driven by biases, feedforward networks use a position-wise linear layer with activation and dropout, residual connections with shortcut connections after attention and FFN layers, and LayerNorm applied after residuals for training stability.

In essence, the complex dependencies are captured between distant graph nodes. Spatial encoding is implemented with a matrix of relative distances between node pairs and injected into the attention score calculation as an additive bias. Bucket distances help generalize across graph sizes. Centrality encoding measures importance for each node and are embedded into the model to obtain attention projections, focusing more crucially on structurally involved sequences.

This structure, in essence, replaces the fixed positional encodings with NLP transformers and facilitate generalization more holistically.

Attention computation between biases are calculated as follows:

Attentioni,j=d(Qi+ci)·(Kj+cj)T+bi,j – where Q, K are Query and Key vectors, c is a centrality encoding, b is a spatial bias with shortest-path distances, and d is a dimension of the attention space. Only biologically useful bias values will drive the attention mechanism with this specific approach.

A multitask head enables classification and regression at the same time with a multitask wrapper consisting of the following: binary sigmoid classification heads for each task and a linear regression head for efficacy prediction. Multitask output heads allow a shared encoder, binary classification head, and regression head with the graph processed through transformer layers once, fully connected layer followed by a sigmoid activation, a probability score outputted for binary classification, and a linear layer outputting a continuous value between 0 and 1 for regression head trained with Mean Squared Error loss against true efficacy value.

The formula for a weighted sum of classification and regression losses is as follows: L=λcls·BCE(y^cls,ycls)+λreg·MSE(y^reg,yreg).

Regularization and optimization are influenced with dropout and layer normalization to apply both attention weights and fully connected layers and stabilize training across stacked layers. Weight decay prevents overfitting on small datasets like PFRED and a learning rate scheduler allow warm-up steps followed by a cosine decay.

In essence, these improve generalization, reduce overfitting, and support better convergence on biological data.

Technical Innovations

Technical innovations that Graphormer offer compared to traditional GCNs (Graph Convolutional Networks) and GAT (Graph Attention Network) include a position-independent encoding that avoids over-smoothing, a transformer-style global attention over nodes, enabling long-range dependency modeling, and a tunable UNREACHABLE_NODE_DISTANCE for handling large, sparse graphs. Position-independent structural encoding has several limitations with traditional Graph Neural Networks like GCNs and GATs, relying heavily on node degrees, adjacency matrices, and Laplacian eigenvectors for encoding structure. Over-smoothing with node embeddings converging to similar values after multiple layers and limited receptive field for GNNs to struggle with long-range dependencies are major limitations with position-independent structural encoding. Graphormer replaces positional encodings with centrality encoding and spatial bias matrices to guide attention mechanisms, allowing every node to attend every other node and eliminating the need for iterative message parsing. Long-range interactions for modeling are crucial with biological sequences and centrality and spatial bias mechanisms allow graph centrality to be embedded into the attention mechanism, helping identify structurally important motifs in a DNA sequence. A matrix of shortest path distances between node pairs is precomputed and added to the self-attention score, allowing the model to learn relative positioning chemically. Global graph comprehension is enabled while maintaining the need for local structure, which governs the essence of ASO-RNA binding.

UNREACHABLE_NODE_DISTANCE and Distance Bucketing helps configure tokens for undefined and infinite distances, used in the spatial bias matrix, avoiding misleading connections and preserving sparse graph integrity. Distance bucketing use node-to-node spatial distances bucketed into intervals instead of raw distances to generalize more efficiently across similar patterns.

Unlike most GNNs that utilize fully connected transformer attention on graphs, Graphormer uses multi-head self-attention between all nodes in the graph, without considering adjacency – global context allows every node to influence each other and enable modeling of distal interactions, multi-head attention allows different heads to learn distinct patterns like base pairing, structural motifs, and motif repetition. Aggregation bottlenecks allow transformer attention to not rely on fixed aggregation functions and helps with high-dimensional node repesentations.

With HuggingFace and Incredible PyTorch sources, key innovating attributes are support for SMILES and large biological graphs, scalability to large sequences and large node-count graphs needed for ASO modeling, and superior performance on benchmarks like Tox21 and PCQM4Mv2. HuggingFace integrated Tokenizer APIs for graph inputs with SMILES and biomolecules, pretrained Graphormer checkpoints for chemistry, biology/drug discovery, and exportable transformers.Trainer for reproducibility and fine-tuning. Microsoft's PCQM4Mv2 and Tox21 pretraining demonstrate strong zero-shot and fine-tuning performance on downstream molecular biology tasks. Domain specific enhancements for ASOs include encoding phosphorothioate linkages and 2’-O modifications as typed edges, using base pairing potential from RNA folding tools in spatial bias, and embedding binding site accessibility into centrality scores.

Collectively, all these pipelines and framework integrations allow Graphormer to model both structure and function of ASO sequences, which is very biologically driven.

Demo – End to End Pipeline

The input allows each sample from PFRED.csv with a DNA sequence, efficacy label, and associated chemical or structural metadata. The key columns for the PFRED.csv are DNA Sequence, Efficacy, Accession ID, and Gene Description, with each row in the dataset representing one ASO-target pair. Every sequence must then be subsequently converted to a graph structure with nodes, node features, and edges. Each nucleotide – adenine, thymine, cytosine, and guanine – becomes a node with one-hot encoding, positional indexing, and chemical modification indicators like methylation correlating to the node features. Edges are connected to adjacent nucleotides with optional edges for base-pair interactions and optional weighted edges for thermodynamic data. The resulting preprocessed sequences are converted into graph representations with domain-specific encodings and a model inference as follows:

Model inputs are then passed to a graph with Graphormer tokenization/encoder that applies centrality encoding to each node, computes spatial encoding biases between node pairs, and node features embedded and passed through Graphormer transformer encoder layers. Attention biases help with customized attention masks to allow spatial information and unreachable node penalties to affect token relevance.

Multitask models have one shared encoder, two separate heads – one for regression (efficacy prediction) and one for classification (high/low efficacy). Forward propagation is performed to generate regression output with an efficacy score between 0 and 1 and a binary classification output with thresholds to determine ASOs likelihood for effectiveness. Precise numerical predictions and categorical decision-making are amplified with postprocessing and visualization carried out to create Metrics like RMSE and R^2 for regression and F1 Score, AUROC, and confusion matrix for binary performance. Scatter plots of predicted vs. true efficacy, ROC and PR curves, and heatmaps for attention heads all can potentially be saved to CSV files, plotted with matplotlib, and visualized with interactive tools like TensorBoard and Streamlit.

With classification, efficacy is predicted to evaluate if it is above a meaningful threshold like 0.5. With respect to regression, normalized inhibition efficacy is directly predicted for regression.

Graph construction is done with a molecular toolkit.

Name: Abhinit Sundar
Instructor(s): Dr. Islam and Professor Koutis
TA: Mr. Omkar Pradip Naik
Research Topic: Graphormer Model

Application to the PFRED Dataset

The Graphormer multitask model for the PFRED dataset with 522 entries includes DNA sequences with antisense sequence in 5’-3’ orientation and a normalized gene score for efficacy with accession and gene descriptions as a possible feature expansion. Visualizations include the following: a predicted vs true efficacy plot, confusion matrix for binary classification, and precision-recall/ROC curves for multitask heads.

Future Directions

Graphormer can be extended to support multi-modal ASO with HELM strings, RNA secondary structures, and 3D folding graphs, active learning with training to minimize uncertainty and create more accurate predictions, integrated thermodynamics with Gibbs Free Energy, MIRANDA binding scores, and mismatched penalties into graph edge weights. These directions have the potential to enhance capabilities, generalization, and applicability in real-world biopharmaceutical pipelines.

Multimodal ASO modeling helps move beyond just the DNA sequence, influencing functional efficacy with chemical modifications, RNA secondary structures, and protein interaction motifs. HELM integration helps with chemical modifications like 2’-O-methyl and phosphorothioates, both represented with additional node attributes and typed edges in molecular graphs. RNA secondary structures can be examined with tools like ViennaRNA and RNA structure, with these defining binding accessibility and modeling base-pair interaction edges. 3D Folding Graphs follow experimental or predicted 3D structures using AlphaFold, for instance, to embed spatial node positions, allowing Graphormer to learn from the true molecular context.

To implement these specifications accurately, the node/edge features must be extended to include chemical modification IDs, base-pair bonding types and strengths, and 3D Euclidean distance matrices. Multi-modal inputs with cross and hierarchical attention modules can be integrated on top of the existing Graphormer encoder.

Active learning frameworks attempt to improve the existing large-scale ASO screening, which is expensive and labor-intensive. Active learning thus allows an optimized exploration space for both sequence and chemical engineering with a limited budget. Proposed enhancements include uncertainty estimation with Monte Carlo Dropout and Deep Ensembles along with acquisition functions that handle uncertainty sampling, expected model change, and diversity sampling. The sequences with highest variance in predictions and the samples with the most impact model weights are chosen to ensure sequence diversity with Tanimoto or scaffold distance. An iterative process with Graphormer trained on a small labeled seed set, ranked unlabeled candidates with uncertainty/information gain, top-k candidates selected for labeling and updating training set, and retraining the model can collectively reduce the number of wetlab experiments while not sacrificing model performance. Thermodynamic integration depends on the ASO's ability to effectively bind to its RNA target with thermodynamic descriptors like Gibbs Free Energy, melting temperature, and binding site accessibility. Gibbs Free Energy will thus be calculated using nearest-neighbor models, indicating duplex stability with negative change in Gibbs Free Energy indicating strong binding, and encoding Gibbs Free Energy as a global node feature for lowering attention bias in the model.

MIRANDA binding scores compute ASO-RNA pairing scores with base complementarity, gap penalties, and binding energy. These scores can subsequently be converted into soft edge weights or penalty-based edge attributes. Target site accessibility can use RNAplfold to compute local folding probabilities and added as node features to represent likelihood of target exposure. Mismatched penalties introduce penalty scores as additional edge types or attention masks with an augmented graph input to include Gibbs Free Energy as scalar node attributes, local accessibility as node dropout probabilities, and energy-based edge reweighting for self-attention mechanism. More improved translations will occur as a result with biological intuition embedded into the system compared to just predictive modeling.

Transfer learning and few-shot finetuning are possible with pretrained Graphormer on large public datasets or synthetic ASO datasets with labels. Few-shot finetuning uses MAML or adapter layers to quickly adapt to new gene families with less than 10 samples. Rapid deployment for rare disease targets where experimental data is less is a plausible use case of pretrained Graphormer and few-shot finetuning.

Attention visualization, gradient based attribution, and counterfactuals allow graph nodes to be traced, sequence positions and modifications to contribute to efficacy, and generating minimal edits to ASOs to change efficacy predictions for causality. More optimized hypothesis generation is used as a result.

Modern drug discovery pipelines use automated platforms for ASO sequence generation, filtering, and optimization, with a possibility to wrap Graphormer inference as an API to score ASO candidates in real-time, combine beam search or reinforcement learning to suggest top-k ASOs per target gene, and integrate this novel framework with LLMs to evaluate off-target effects.