Final Report: Pre-Training and Fine-Tuning the Hi-C Foundation Model for the *Arabidopsis thaliana* Species

Summary

This project was designed to adapt the Hi-C Foundation Model, originally optimized for human and animal genomic data, to work effectively with plant species, specifically *Arabidopsis thaliana* (Thale Cress). Despite resource constraints that caused significant modifications to the original experimental design, initial training and validation results from pre-training suggests that domain-specific pre-training on plant Hi-C data shows promise for improving model performance. While the final implementation did not achieve successful resolution enhancement due to computational limitations, the training and validation metrics indicated clear learning progression during the very limited pre-training phase.

Introduction

The Problem

The Hi-C Foundation Model has shown remarkable generalizability across 316 species, and shown amazing results for humans and mice. While it excels at tasks related to humans and well-studied animal species, it falls behind when it is applied to plant genomes. Optimizing the Hi-C Foundation Model for plants would provide key details to the chromatin structure of multiple agricultural plants. *Arabidopsis thaliana* was used as it is consider the “mouse” of plants and has a high amount of Hi-C data available.

Methods

Initial Experimental Design

The original experimental plan was designed to comprehensively evaluate and improve the Hi-C Foundation Model's performance on plant data:

1. Baseline Establishment
    a. Utilize the current Hi-C Foundation Model via Google Colab to generate baseline resolution enhancement results for *Arabidopsis thaliana* Hi-C data
    b. Implementation through the official notebook: https://colab.research.google.com/github/Noble-Lab/HiCFoundation/blob/main/HiCFoundation.ipynb
2. Pre-training Phase
    a. Dataset: 4 Hi-C datasets for training, 2 for validation
    b. Training Parameters:
        i. 100 epochs total
        ii. 50 warmup epochs
        iii. Original command with the parameters for pre-training

```
python3 pretrain.py --batch_size 1 --accum_iter 4 \
  --epochs 100 --warmup_epochs 50 --pin_mem \
  --mask_ratio 0.75 --sparsity_ratio 0.05 \
  --blr 1.5e-4 --min_lr 1e-7 --weight_decay 0.05 \
  --model "vit_large_patch16" --loss_alpha 1 --seed 888 \
  --data_path " input-dirs/pre-train-dirs/” --train_config "train.txt" \
  --valid_config "val.txt" --output "hicfoundation_finetune" \
  --tensorboard 1 --world_size 1 --dist_url "tcp://localhost:10001" --rank 0 \
  --input_row_size 448 --input_col_size 448 --patch_size 16 \
  --print_freq 1 --save_freq 1
```

3. Fine-tuning Phase
    a. Task: Resolution enhancement specific to Arabidopsis genomic features
    b. Dataset: 4 Hi-C datasets for training, 2 for validation
    c. Training Parameters:
        - 100 epochs total
        - 50 warmup epochs
        - Original command with the parameters for fine-tuning

```
python3 finetune.py --batch_size 1 --accum_iter 4 \
  --epochs 100 --warmup_epochs 50 --pin_mem \
  --blr 1e-3 --min_lr 1e-7 --weight_decay 0.05 \
  --layer_decay 0.75 --model vit_large_patch16 \
  --pretrain hicfoundation_finetunef/model/model_best.pth.tar \
  --finetune 1 --seed 888 \
  --loss_type 1 --data_path "example/finetune_resolution” \
  --train_config "train_resolution.txt" \
  --valid_config "val_resolution.txt" \
  --output "hicfoundation_finetune" --tensorboard 1 \
  --world_size 1 --dist_url "tcp://localhost:10001" --rank 0 \
  --input_row_size 224 --input_col_size 224 --patch_size 16 \
  --print_freq 1 --save_freq 1
```

4. Evaluation
    a. Compare pre-trained/fine-tuned model output against baseline using the same test dataset
    b. Metrics: Visual inspection of contact matrices

Modified Experimental Design

Due to severe computational and storage constraints on the available Hamachi server, the method was significantly revised:

1. Resource Limitations

a. Insufficient storage space for multiple large Hi-C datasets
b. Limited GPU memory and computational time
c. Restricted access to high-performance computing resources

2. Revised Pre-training
    a. Dataset: Reduced to 2 Hi-C datasets for training, 1 for validation
    b. Training Parameters:
        - 1 epoch only
        - 1 warmup epoch
        - New command with the parameters for pre-training

```
python3 pretrain.py --batch_size 1 --accum_iter 4 \
  --epochs 1 --warmup_epochs 1 --pin_mem \
  --mask_ratio 0.75 --sparsity_ratio 0.05 \
  --blr 1.5e-4 --min_lr 1e-7 --weight_decay 0.05 \
  --model "vit_large_patch16" --loss_alpha 1 --seed 888 \
  --data_path "input-dirs/pre-train-dirs/" --train_config "train.txt" \
  --valid_config "val.txt" --output "hicfoundation_finetune" \
  --tensorboard 1 --world_size 1 --dist_url "tcp://localhost:10001" --rank 0 \
  --input_row_size 448 --input_col_size 448 --patch_size 16 \
  --print_freq 1 --save_freq 1
```

3. Revised Fine-tuning
    a. Dataset: 1 Hi-C dataset used for both training and validation (due to storage constraints)
    b. Training Parameters:
        - 1 epoch only
        - 0 warmup epochs
        - New command with the parameters for fine-tuning

```
python3 finetune.py --batch_size 1 --accum_iter 4 \
  --epochs 1 --warmup_epochs 0 --pin_mem \
  --blr 1e-3 --min_lr 1e-7 --weight_decay 0.05 \
  --layer_decay 0.75 --model vit_large_patch16 \
  --pretrain hicfoundation_pretrain/model/model_best.pth.tar \
  --finetune 1 --seed 888 \
  --loss_type 1 --data_path “ft-inputs" \
  --train_config “train_config.txt" \
  --valid_config “val_config.txt" \
  --output "hicfoundation_finetune" --tensorboard 1 \
  --world_size 1 --dist_url "tcp://localhost:10001" --rank 0 \
  --input_row_size 448 --input_col_size 448 --patch_size 16 \
  --print_freq 1 --save_freq 1
```

4. Evaluation

a. Evaluation remains the same

Results and Discussion

Results

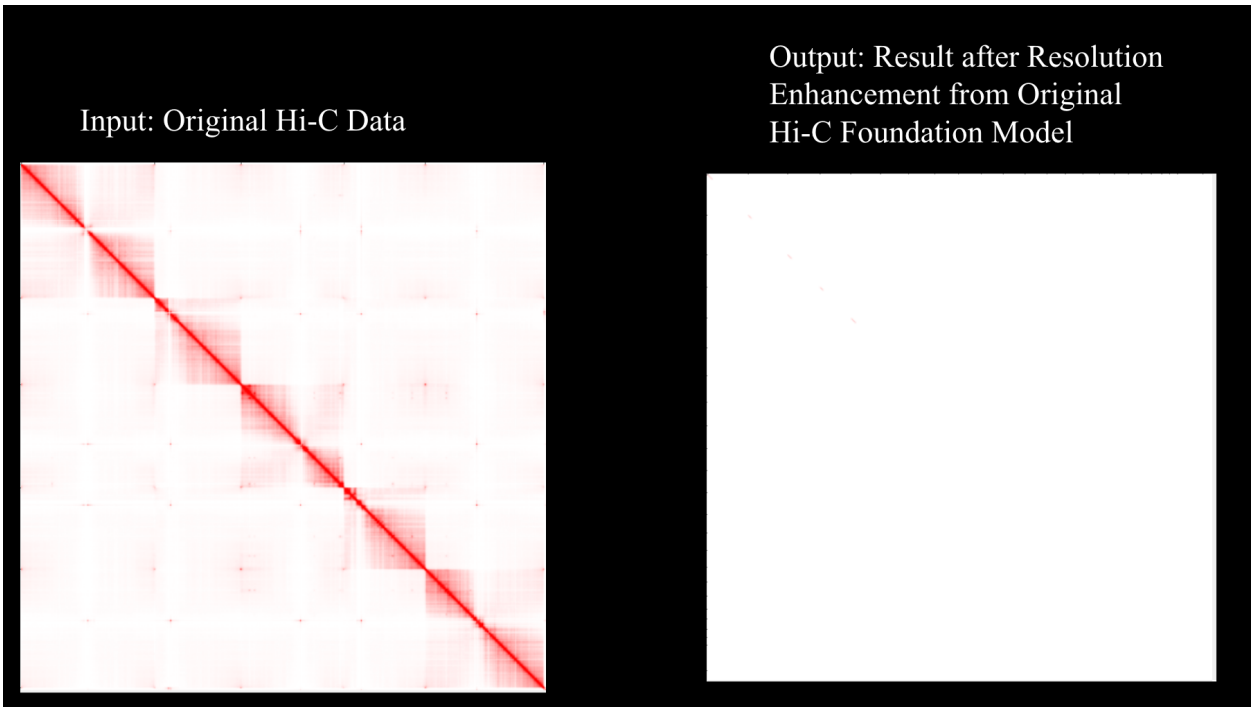

Image 1

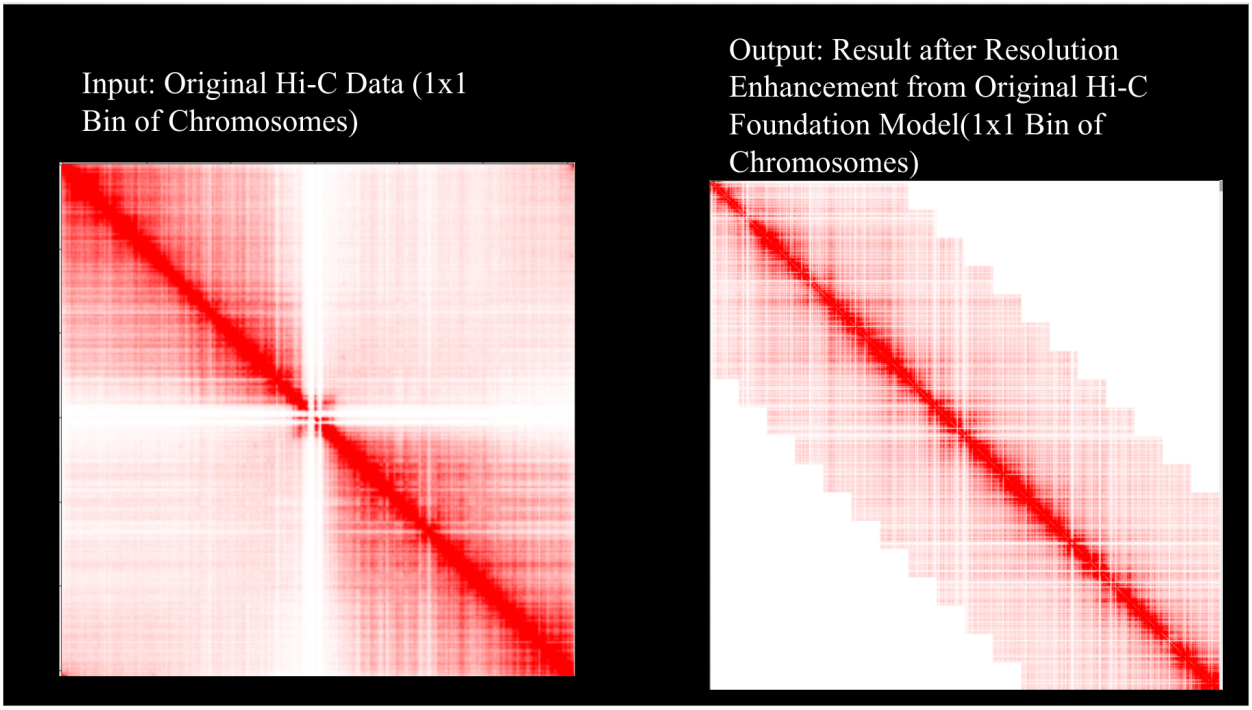

Image 2

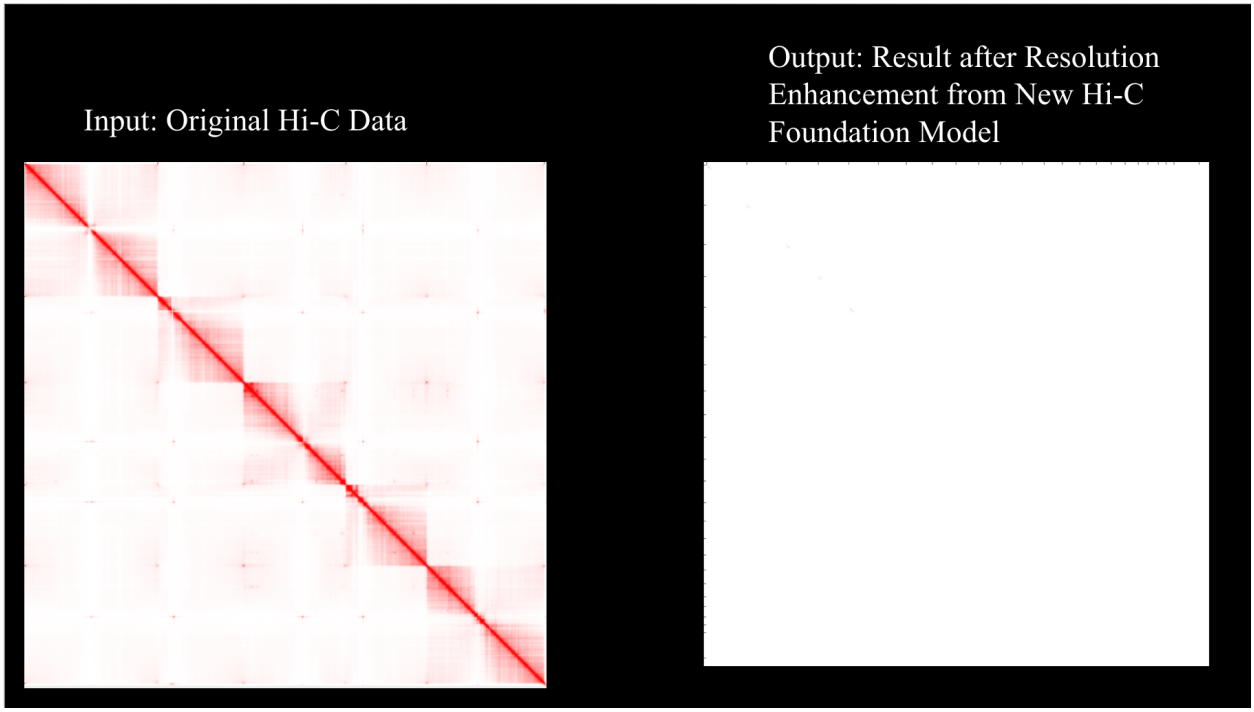

Image 3

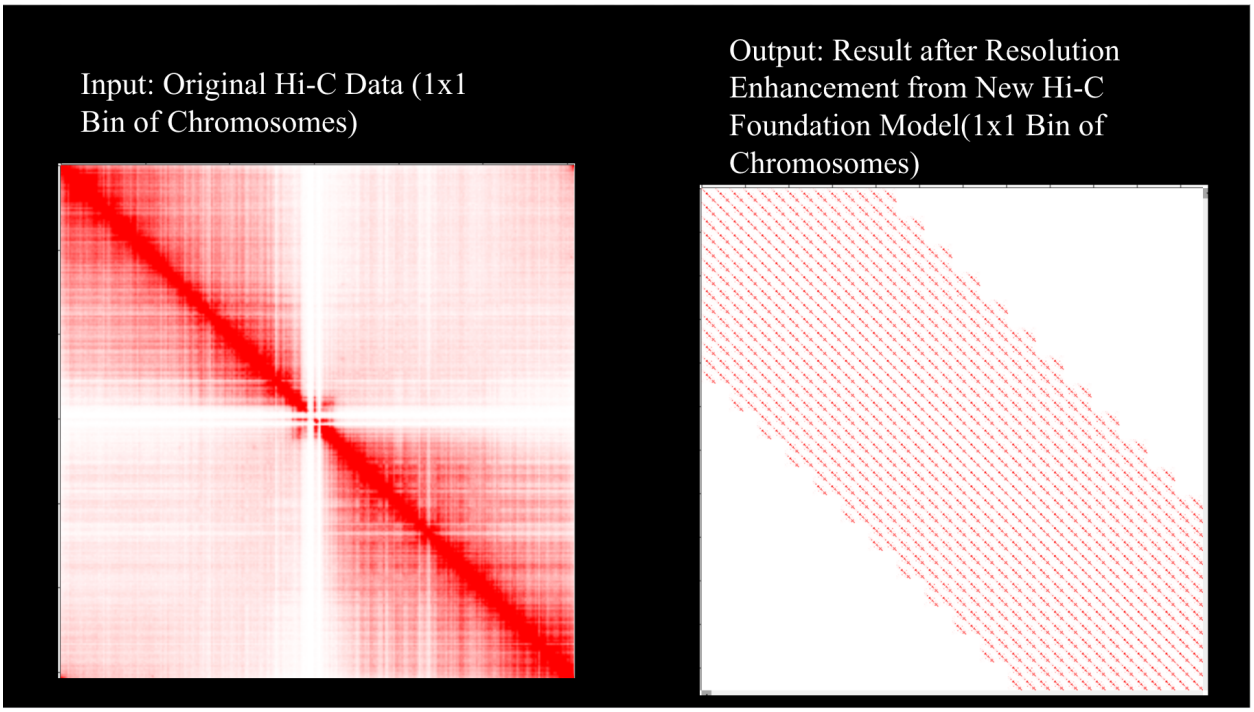

Image 4

Interpretation of Results

The finetuned model did not successfully enhance the baseline Hi-C file as we seen in the comparison between Image 2 (the original model) and Image 4 (the new model). This is most likely due to insufficient computational resources rather than fundamental limitations of the approach. The original Hi-C Foundation Model was trained on a lot more data and had vastly more computational resources compared to what this project could afford. Due to this lack of resources, this project had access to vastly inferior computational resources, making meaningful model adaptation impossible.

Promising Indicators from Previous Pre-Training

Despite the overall failure to achieve resolution enhancement, it was also found that there was decreasing training loss during the first limited pre-training. This indicates that this model has the ability to enhance resolution on Hi-C data for Thale Cress given more time and resources. The training and validation data as shown below shows that the model was improving as the training and loss was going down and the SSMI loss remained a high level very consistently.

Table 1: Training Data from the first Pre-Training Run

	Epoch 0	Epoch 1	Epoch 2	Epoch 3
Training loss	~17.84	~10.16	~8.51	~8.11
SSMI Loss	~0.98	~0.98	~0.98	~0.98
Count Loss	~10.19	~2.5	~0.87	~0.47
Contrastive Loss	~6.67	~6.67	~6.67	~6.67

Table 2: Validation Data from the first Pre-Training Run

	Epoch 0	Epoch 1	Epoch 2	Epoch 3
Training loss	~9.08	~8.02	~9.32	~7.97
SSMI Loss	~0.98	~0.98	~0.98	~0.98
Count Loss	~1.43	~0.37	~1.67	~0.32
Contrastive Loss	~6.67	~6.67	~6.67	~6.66

Significant Challenges and Technical Observations

The final model evaluation revealed significant challenges:

1. No Improvement in Resolution: The modified model failed to enhance resolution compared to the baseline as seen between Image 1 (the original model) and 3 (the new model)
2. Potential Degradation: Visual inspection suggested the output might be worse than the original input as seen between Image 2 (the original model) and Image 4 (the new model).

3. Insufficient Training: With only 1 epoch of pre-training and fine-tuning, the model had insufficient opportunity to learn plant-specific features

Key technical findings from the implementation:

- Memory Constraints: Even with reduced batch sizes, GPU memory limitations severely impacted training
- Storage Issues: The inability to maintain separate training and validation datasets likely contributed to overfitting
- Convergence: The model showed no signs of convergence within the limited training time

Reflection / Future Plans

Immediate Improvements

With more computational resources, the following improvements should be implemented:

1. Full-Scale Training: Implement the original training protocol with 100+ epochs
2. Larger Dataset: Utilize all available Arabidopsis Hi-C datasets (10+ experiments)
3. Cross-Species Pre-training: Include Hi-C data from multiple plant species such as rice and corn to improve generalization

Methodological Innovations

Future work could explore:

1. Plant-Specific Architecture: Modify the model architecture to better capture plant chromatin features
2. Transfer Learning: Investigate optimal strategies for transferring knowledge from animal to plant domains
3. Multi-Task Learning: Simultaneously train on multiple plant-relevant tasks beyond resolution enhancement

Broader Applications

Successful adaptation of Hi-C Foundation Model to plants could enable:

- Crop Improvement: Enhanced understanding of 3D genome organization in agricultural species
- Evolutionary Studies: Comparative analysis of chromatin architecture across plant phylogeny
- Gene Regulation: Better prediction of enhancer-promoter interactions in plants

Reflections and Lessons Learned

Technical Insights

This project provided valuable experience in:

- Working with large-scale genomic data
- Adapting state-of-the-art deep learning models to new domains
- Managing computational constraints in bioinformatics research
- Understanding the unique features of plant genome organization

Research Process

Key learnings from the research process:

1. Resource Planning: The importance of accurately estimating computational requirements before project initiation
2. Incremental Validation: Even failed experiments provide valuable insights about model behavior
3. Domain Expertise: The critical need for understanding biological differences between training and target domains

Personal Growth

Working with genomic data proved to be quite fun and opened new research directions. The intersection of machine learning and genomics represents a rapidly evolving field with immense potential for biological discovery which I am excited to see in the future.

Conclusion

While this project did not achieve its primary objective of improving Hi-C Foundation Model performance for plant species due to computational constraints, it established there is feasibility to the approach done in this project and identified key challenges specific to plant genomic data. These early results suggest that with adequate resources, domain-specific pre-training could significantly improve model performance on plant Hi-C data. Future work with appropriate computational resources could build upon this foundation to develop plant-optimized models for 3D genome analysis to greatly improve agricultural knowledge.