Bioinformatics Research Project Log

Chandrashekar CR
Supervisor: Dr. Eran Elhaik
Lund University

April 10, 2025

Contents

March 31, 2025

Tasks for the Day

- Understood the mGPS algorithm from the R code and implemented the preprocessing steps.
- Set up the working environment and installed all required libraries.
- Began tracking the project.

Notes and Observations

- Utilizing the `ai_env` environment from previous projects and accessing resources on the bioinformatics server.
- Initial challenges in understanding the R code are anticipated to decrease with further engagement.

April 1, 2025

Tasks for the Day

- Understood and implemented Recursive Feature Elimination followed by the XGBoost machine learning algorithm with the correct hierarchical steps.
- Initialized Git for version control and pushed code to GitHub.
- Gained a deeper understanding of cross-validation principles.

Notes and Observations

- Achieved a general understanding of the workflow and added comments to the MetaSUB preprocessing script in R.
- Acquired knowledge regarding the importance of cross-validation, although implementation is pending.

April 2, 2025

Tasks for the Day

- Acquired information from Vignesh regarding access to the LUNARC server.
- Determined that reimplementing the exact XGBoost model is unnecessary; the focus is on understanding the input data preprocessing.

Notes and Observations

- Git repository initialized for the project, tracking all files except data and research papers.
- The `metasub_global_git.csv` file contains the Geographically Informative Taxa (GITs) required for the XGBoost model.
- The primary objective is to comprehend the preprocessing of input data for XGBoost.
- Key questions identified: What is the shape of the input data? What are the prediction targets?

April 3, 2025

Tasks for the Day

- Implemented basic neural network architectures in PyTorch. Hyperparameter tuning indicates that 200 GITs are sufficient for accurate predictions, despite the dataset containing n data points.
- Integrated Ray parallel processing to optimize hyperparameters, aiming to reduce the estimated 4-5 hour search time.
- Preprocessed data into numerical format by converting categorical variables (continents and cities) using one-hot encoding.
- Deferred the implementation of stratified K-fold cross-validation for later.

Notes and Observations

- Successfully logged into the LUNARC server's login node, but GPU access and utilization for neural network training require further investigation.
- Authentication and login to LUNARC are complete; however, assistance is needed to understand:
  - The location of allocated storage.
  - The process of submitting jobs using SBATCH.
  - The fundamentals of working on High-Performance Computing (HPC).
- Following Recursive Feature Elimination (RFE), the final dataset has a shape of 4070×204, with 200 features and 4 target variables.
- Each data point comprises 200 features (GITs) representing the relative sequence abundance (RSA) of microorganisms. The 4 target variables are continent, city, latitude, and longitude.
- The dataset includes samples from 40 unique cities across 7 continents.
- Categorical variables (continent and city) were encoded using `sklearn`'s `LabelEncoder`, while latitude and longitude were standardized using `StandardScaler`.
- Initial consideration of stratified cross-validation was temporarily replaced with `train_test_split` for initial model development. Stratified cross-validation will be revisited for enhanced model performance.

Neural Network Architecture

- The initial model is a simple feedforward neural network, inspired by the hierarchical structure of the previous XGBoost study.
- The first version includes an input layer (200 nodes), two hidden layers (400 nodes each), a smaller hidden layer (2 nodes), and an output layer (7 nodes for the 7 continents).
- The plan is to initially train this model to predict the continent. Subsequently, the predicted continent probabilities will be concatenated with the original 200 features as input for a second neural network (similar architecture) to predict the city.
- The optimal handling of latitude and longitude values within the neural network remains an open question.
- Long training times due to CPU-based computation on the bioinformatics server are a significant challenge. GPU access on the LUNARC cluster is required.

April 4, 2025

Tasks for the Day

- Finalized the presentation for the weekly lab meeting.
- Focused on obtaining GPU access and understanding HPC architecture.
- Initiated preprocessing for marine and soil datasets, aiming for modular script design applicable to all datasets.

Notes and Observations

Explored various neural network implementations and gained initial understanding of HPC principles.

April 7, 2025

Tasks for the Day

- Developed a functional neural network to predict all target variables (continent, city, latitude, and longitude).
- Completed preprocessing steps for marine and soil datasets.
- Began modularizing code for efficient execution on the HPC.

Notes and Observations

- Developed a working script for MetaSUB data processing, with pending error handling and file format validation.
- Created a working script to extract relevant features using Recursive Feature Elimination with a Random Forest base model.
- Initiated work on the HPC, understanding basic operations and starting to modularize scripts for GPU compute nodes.

April 8, 2025

Notes and Observations

- Started building multiple neural network models.
- Began learning batch scripting using SLURM.

April 10, 2025

Notes and Observations

- Exploring alternative scaling methods for latitude and longitude, including conversion to radians and trigonometric transformation into a two-dimensional space.

April 11, 2025

Tasks for the day

- These are some of my thoughts. The currrent approach is always making a new neural network model from scratch for each dataset. A final model should be made that utilizes all the iterations done and must work for all type of datasets regardless of the layers of predictions.
- For example, the MetaSUB dataset contains informaiton on the continent, city, latitude and longitude. Whereas the marine dataset contain information only the sea, latitude and longitude. There should be a way that can handle these cases instead of definfing a newtwork from scratch.
- Finish the logic for the latitude and longitude neural network model.
- Compare the three models with metasub dataset on accuracy, confusion matrix, plot on world map.

April 14, 2025

Notes and Observations

- **Combined Models Generally Outperform Separate Models:** Models 2 (nn_model_combined.py) and 4 (nn_combined_model_lat_long.py), which employ combined or hierarchical architectures, tend to show better performance, particularly on the latitude and longitude/XYZ prediction tasks, compared to Model 1 (nn_model.py) and Model 3 (nn_model_lat_long.py) which use separate networks.
- **XYZ Coordinate Transformation Seems Beneficial:** Models 3 and 4, which incorporate the transformation of latitude and longitude into XYZ coordinates, demonstrate significantly lower Mean Absolute Errors for these predictions compared to Model 1, which predicts latitude and longitude directly.
- **Trade-offs Between Classification and Regression:** There isn't one single model that dominates across all tasks. Some models show higher accuracy for continent and city prediction, while others excel in latitude and longitude prediction.
- **Overfitting:** In several instances, the training accuracy is notably higher than the test accuracy, suggesting some degree of overfitting across the models. This is a common challenge in machine learning and something to consider for future improvements.

Model-by-Model Comparison

- **Model 1: nn_model.py (Right Top Corner)**
  - **Continent Prediction:** Strong test accuracy (89.56%) with balanced precision, recall, and F1-score.
  - **Cities Prediction:** Lower test accuracy (78.75%) compared to continent prediction.
  - **Latitude and Longitude Prediction:** High MAEs (0.3964 and 0.3445). Maps show significant spread between predictions and true locations.
  - **Architecture:** Uses separate networks for each task, which may limit learning of shared representations.
- **Model 2: nn_model_combined.py (Left Bottom Corner)**
  - **Continent Prediction:** Slightly lower test accuracy (88.33%) than Model 1.
  - **Cities Prediction:** Improved test accuracy (81.82%) with higher precision and F1-score.
  - **Latitude and Longitude Prediction:** Lowest MAEs (0.2230 and 0.1876). Very accurate predictions, as confirmed by maps.

  - **Architecture:** Hierarchical structure facilitates learning between tasks.

- **Model 3: nn_model_lat_long.py (Right Bottom Corner)**
  - **Continent Prediction:** High test accuracy (89.31%).
  - **Cities Prediction:** Lowest test accuracy (75.92%) among all models.
  - **Latitude and Longitude Prediction:** Very high MAEs (8.5982 and 21.2971). Severe scatter on maps.
  - **Architecture:** Uses XYZ internally but has poor coordinate prediction, possibly due to flawed transformations or output layers.
- **Model 4: nn_combined_model_lat_long.py (Left Top Corner)**
  - **Continent Prediction:** Slightly lower test accuracy (87.35%).
  - **Cities Prediction:** Highest test accuracy (82.92%) among all models.
  - **Latitude and Longitude Prediction:** Low MAEs (4.5070 and 14.4377), but not as low as Model 2.
  - **Architecture:** Combines hierarchical structure with XYZ usage, effective for cities and coordinates.

Comparative Analysis and Best Model

- **Best for Coordinates:** Model 2 clearly outperforms others in latitude and longitude prediction.
- **Best for Cities:** Model 4 achieves the highest accuracy.
- **Continent Accuracy:** All models perform similarly well.
- **Overall Best Model:** Model 2 (nn_model_combined.py) is the most well-rounded, offering competitive classification accuracy and the best coordinate prediction performance.

Opportunities for Improvement

- **Reduce Overfitting:** Apply techniques like dropout, L1/L2 regularization, or early stopping.
- **Fix Model 3:** Investigate XYZ conversion and back-transformation.
- **Refine Hierarchical Designs:** Experiment with alternative fusion strategies or attention mechanisms.
- **Ensemble Strategies:** Combining predictions may improve performance across tasks.

April 16, 2025

Tasks for the Day

- Analyzed preliminary results from the first set of neural network models.
- Began implementing data transformation for coordinate prediction.
- Started developing the combined neural network architecture.

Notes and Observations

- Initial separate neural networks show promising results for continent classification (82% accuracy) but struggle with coordinate prediction (median error ¿ 5000 km).
- Direct prediction of latitude and longitude values is problematic due to the discontinuity at the -180°/+180° longitude boundary. This leads to excessive errors for samples near these boundaries.
- Research into coordinate prediction techniques suggests transforming latitude and longitude into 3D Cartesian coordinates (x, y, z) for better model training.
- Initial implementation of the Cartesian coordinate transformation shows improved gradient flow during training.
- Combined neural network design initiated with a hierarchical architecture to better capture the relationships between continent, city, and coordinate prediction tasks.

April 22, 2025

Tasks for the Day

- Completed implementation of the combined neural network model.
- Set up proper evaluation metrics for all prediction tasks.
- Researched potential advanced architectures for further improvements.

Notes and Observations

- Combined neural network demonstrates significantly improved coordinate prediction compared to separate networks. Median error reduced from ¿5000 km to around 1600 km.
- Multi-task learning appears to create beneficial inductive bias, helping the model learn shared representations that benefit all prediction tasks.
- Evaluation metrics established: accuracy and F1-scores for classification tasks (continent and city), geodesic error (Haversine distance) for coordinate prediction.
- Error analysis reveals that coordinate prediction accuracy is strongly influenced by the correctness of continent and city classification. When both are correct, the median error drops dramatically.
- Research into literature suggests that gradient boosting frameworks with neural networks as weak learners (e.g., GrowNet) could further improve performance, especially for hierarchical prediction tasks.

April 29, 2025

Tasks for the Day

- Started implementing the GrowNet model architecture.
- Conducted comprehensive error analysis on the neural network models.
- Developed visualization pipeline for geographic predictions.

Notes and Observations

- Initial GrowNet implementation completed but encountering stability issues during training. Gradient scaling and learning rate adjustments needed.
- Error analysis confirms strong correlation between classification accuracy and coordinate prediction precision. When continent prediction is incorrect, coordinate error is generally very large.
- Visualizations using GeoPandas show clear patterns of prediction accuracy across different geographic regions. European and North American cities show consistently lower error compared to other regions.
- Task weights in multi-task learning significantly affect performance. Higher weights for continent and city classification (relative to coordinate regression) lead to better overall results.
- All model files are stored at `/home/chandru/binp37/scripts/nn_models/`.

May 5, 2025

Tasks for the Day

- Stabilized the GrowNet implementation with proper gradient handling.
- Conducted preliminary experiments with ensemble approaches.
- Explored alternative coordinate representation methods.

Notes and Observations

- GrowNet stability issues resolved by implementing proper gradient scaling and regularization. The model now shows promising performance, with 86.4% continent accuracy and 75.1% city accuracy.
- Initial ensemble experiments combining XGBoost, neural networks, and GrowNet show significant potential. The combined predictions outperform any single model.
- Alternative coordinate representation methods explored, including spherical coordinates and geohash encoding, but the Cartesian (x, y, z) transformation remains most effective for neural network training.
- Started implementing a more structured ensemble framework that can leverage multiple model types while respecting the hierarchical nature of the prediction tasks.
- Updated GrowNet implementation is at `/home/chandru/binp37/scripts/grownet/grownet_classification.py`.

May 15, 2025

Tasks for the Day

- Implemented and evaluated various ensemble strategies.
- Optimized the coordinate prediction component of GrowNet.
- Conducted comprehensive comparison between all model approaches.

Notes and Observations

- Ensemble strategies evaluated: weighted voting, stacking with XGBoost meta-learner, and feature-weighted linear stacking. The stacking approach with XGBoost meta-learner shows best performance for classification tasks.
- For coordinate regression, simple averaging of predictions degrades performance compared to selecting the single best-performing model (TabPFN).
- Comprehensive comparison reveals clear hierarchy of performance: Ensemble ¿ GrowNet ¿ Combined NN ¿ Separate NN. This pattern is consistent across all prediction tasks and evaluation metrics.
- Ensemble implementation demonstrates impressive results: 91.2% continent accuracy, 89.7% city accuracy, and median coordinate error of 28.4 km.
- Started planning for a more sophisticated ensemble architecture that can adapt to different dataset sizes and feature distributions.
- Core ensemble implementation files at `/home/chandru/binp37/scripts/ensemble/main.py`.

May 23, 2025

Tasks for the Day

- Further refined the ensemble model with threshold-based model selection.
- Implemented proper error propagation analysis framework.
- Started preparing figures and analysis for the final report.

Notes and Observations

- Threshold-based model selection significantly improves ensemble performance. Only models exceeding 93% accuracy on continent classification and 91% on city classification are included in the ensemble.
- Error propagation analysis framework completed. Now able to quantify how errors at each level of the hierarchy affect final coordinate predictions.
- Initial figures for the report created, including world maps showing the distribution of true vs. predicted coordinates and error distribution by continent/city.
- TabPFN model showing exceptional performance for coordinate regression, but encountering memory limitations when scaling beyond 100 features. Need to explore feature selection strategies specifically for this model.
- Updated ensemble implementation with threshold-based selection at `/home/chandru/binp37/scripts/ensemble/main`

June 1, 2025

Tasks for the Day

- Explored hierarchical GrowNet implementations.
- Conducted ablation studies on the ensemble model.
- Started comparison with previous mGPS implementation.

Notes and Observations

- Hierarchical GrowNet implementation completed, showing improved performance over the standard version. The hierarchical approach better captures the dependencies between prediction levels.
- Ablation studies reveal that removing any single model type from the ensemble reduces performance, confirming that each model contributes unique predictive information.
- Initial comparison with mGPS shows promising results. Our ensemble approach achieves comparable city-level accuracy (93% vs. 92%) but significantly lower median coordinate error (18.5 km vs. 137 km).
- Started implementing a more comprehensive benchmark framework to ensure fair comparison between our approach and mGPS.
- Hierarchical GrowNet implementation at `/home/chandru/binp37/scripts/grownet/hierarchical_grownet.py`.

June 4, 2025

Performing recursive feature selection on the tax_metasub_data.csv we get 300 important features.

June 8, 2025

Tasks for the Day

- Refined feature selection approach to optimize for each model type.
- Completed in-radius accuracy analysis for all models.
- Started writing methods section for the manuscript.

Notes and Observations

- Model-specific feature selection implemented. Tree-based models (XGBoost, LightGBM, CatBoost) use all 300 RFE-selected features, while TabPFN uses only the top 95 features due to memory constraints.
- In-radius accuracy analysis shows that the ensemble model places 68.6% of predictions within 50 km of the true location, and 86.6% within 1,000 km. This significantly outperforms all other approaches.
- Methods section drafting started, focusing on the hierarchical ensemble architecture and error propagation analysis.
- Feature selection scripts and analysis at `/home/chandru/binp37/scripts/feature_engineering/rfe_feature_select`

June 15, 2025

Tasks for the Day

- Implemented coordinate transformation utilities for consistent evaluation.
- Continued benchmark comparison with mGPS.
- Started investigating biological significance of important features.

Notes and Observations

- Coordinate transformation utilities implemented for consistent conversion between latitude/longitude and Cartesian coordinates across all models.
- Benchmark comparison with mGPS now complete. Our ensemble achieves a tenfold reduction in median coordinate error (13.72 km vs. 137 km) while maintaining slightly better city-level accuracy (93% vs. 92%).
- Initial investigation of biological significance reveals that many of the most important features correspond to environmental microbes with known geographic distribution patterns.
- Surprisingly, several human-associated microbes also show strong geographic stratification, suggesting potential interactions between human populations and urban environmental microbiomes.
- Coordinate utilities implementation at `/home/chandru/binp37/scripts/ensemble/coordinate_utils.py`.

June 25, 2025

Tasks for the Day

- Created comprehensive confusion matrices for continent and city classification.
- Implemented weighted multi-task learning in all neural network models.
- Started drafting the results section of the manuscript.

Notes and Observations

- Confusion matrices reveal specific patterns of misclassification. European cities are sometimes confused with North American cities, likely due to similar urban environments and human population genetics.
- Weighted multi-task learning significantly improves neural network performance. By assigning higher weights to continent and city classification (compared to coordinate regression), the models achieve better overall results.
- Results section drafting in progress, with focus on the comparative performance of different modeling approaches and the benefits of the ensemble strategy.
- All confusion matrices and performance visualizations are stored in `/home/chandru/binp37/report/figures/`.

July 2, 2025

Tasks for the Day

- Finalized the ensemble architecture with adaptive model selection.
- Completed error group analysis for all models.
- Started work on the discussion section of the manuscript.

Notes and Observations

- Ensemble architecture finalized with adaptive model selection based on dataset characteristics. The system now automatically selects the appropriate models based on dataset size, feature distribution, and computational constraints.
- Error group analysis complete, confirming the strong relationship between classification accuracy and coordinate prediction precision. When both continent and city are correctly classified (90.3% of cases for the ensemble model), the median error drops to just 12.3 km.
- Discussion section drafting focuses on the implications of our findings for microbial biogeography, the advantages of ensemble learning for geographic prediction, and potential applications in forensics and biosurveillance.
- Final ensemble architecture implementation at `/home/chandru/binp37/scripts/ensemble/main.py`.

July 10, 2025

Tasks for the Day

- Created all figures for the manuscript.
- Implemented final performance comparison between all models.
- Drafted abstract and introduction sections.

Notes and Observations

- All manuscript figures completed, including world maps showing prediction accuracy, error distribution by continent and city, and classification performance heatmaps.
- Final performance comparison confirms the superiority of the ensemble approach across all metrics. The ensemble achieves 95.0% continent accuracy, 93.0% city accuracy, and a median coordinate error of 13.72 km.
- Abstract and introduction drafts completed, focusing on the novelty of our hierarchical ensemble approach and its significant improvement over previous methods.
- All final figures are stored in `/home/chandru/binp37/report/figures/` and referenced in the manuscript.

July 19, 2025

I am just keep a track of the things that I have done so far. This part consists of the scripts that I have written and the models that I have created.

/home/chandru/binp37/scripts/ensemble/

This is the root directory for the ensemble modeling scripts.

- `main.py`: This is the main entry point for the ensemble pipeline. It imports all the different models from the subdirectories, handles data loading and preprocessing (specifically for a hierarchical prediction task), and likely orchestrates the training and evaluation of the ensemble.

/home/chandru/binp37/scripts/ensemble/catboost_ensemble/

This folder contains scripts related to the CatBoost model.

- `catboost_classification.py`: Implements a `CatBoostTuner` class for training and tuning a CatBoost classifier. It uses Optuna for hyperparameter optimization and includes methods for training, evaluation, and running the full pipeline.

/home/chandru/binp37/scripts/ensemble/ft_transformer/

This folder is for the FT-Transformer model, a state-of-the-art architecture for tabular data.

- `ft_transformer_classification.py`: Implements a classifier using the FT-Transformer model. It includes `FTClassifier` for training/evaluation and `FTTransformerTuner` for hyperparameter tuning with Optuna.

/home/chandru/binp37/scripts/ensemble/grownet/

This folder contains scripts for GrowNet, a gradient boosting framework using neural networks.

- `grownet_classification.py`: Implements a `GrowNetClassifier` for classification tasks. It uses a series of small MLPs trained sequentially. It also includes a `GrowNetTuner` for hyperparameter optimization.
- `grownet_regressor.py`: Implements a `GrowNetRegressor` for regression tasks, following the same boosting principle as the classifier. It also has a corresponding `GrowNetTuner`.

/home/chandru/binp37/scripts/ensemble/lightgbm_ensemble_model/

This folder is for the LightGBM model.

- `lightgbm_classification.py`: Implements a `LightGBMTuner` class for training and tuning a LightGBM classifier. It uses Optuna for hyperparameter search and provides functions to train and evaluate the model.

/home/chandru/binp37/scripts/ensemble/random_forest/

This folder is for the Random Forest model.

- `randomforest_classification.py`: Contains a `RandomForestTuner` class that defines a complete pipeline (`run_complete_pipeline`) for tuning, training, and evaluating a Random Forest classifier.

/home/chandru/binp37/scripts/ensemble/simple_nn/

This folder contains scripts for simple, fully-connected neural networks.

- `nn_classification.py`: Implements an `NNTuner` class to find the best hyperparameters for a neural network classifier using Optuna.
- `nn_regression.py`: Contains an `NNTuner` class for tuning a neural network regressor, optimizing for a regression metric.

/home/chandru/binp37/scripts/ensemble/xgboost_ensemble/

This folder is dedicated to XGBoost models.

- `xgboost_classification.py`: Implements an `XGBoostTuner` class for hyperparameter tuning and training of an XGBoost classifier using Optuna.
- `xgboost_regression.py`: Defines a `XGBoostRegressorTuner` class, which currently specifies default parameters for an XGBoost regressor.

/home/chandru/binp37/scripts/grownet/

These scripts implement different versions of the GrowNet model, a gradient boosting framework that uses neural networks as weak learners.

`hirarchical_grownet.py`

This script implements a specialized, hierarchical version of the GrowNet model tailored for a multi-task prediction problem. Its primary goal is to simultaneously predict a hierarchy of geographical labels: continent, city, and geographical coordinates (x, y, z).

- **Hierarchical Structure:** The model architecture is explicitly hierarchical. The prediction for continents is used as an input feature for predicting cities, and both continent and city predictions are used to predict the final coordinates.
- **Multi-Task Learning:** It uses a combined loss function to train all three tasks concurrently.
- **Learnable Uncertainty:** The script employs learnable uncertainty weights (`log_sigma`) to automatically balance the contribution of the loss from each task (continent classification, city classification, and coordinate regression). This helps prevent one task from dominating the training process.
- **Custom Boosting:** It uses a custom gradient boosting approach where each new weak learner is trained to correct the residuals of the combined ensemble.

`grownet_classification.py`

This script provides a more standard implementation of the GrowNet model for multi-class classification tasks. It follows the core principles of stage-wise training of weak learners.

- **Stage-wise Training:** The model is built by sequentially adding weak learners (`WeakLearner`). The first learner is trained on the original features, while subsequent learners are trained on the original features concatenated with the cumulative predictions from the previous stages.
- **Corrective Step:** After all weak learners are trained individually, a global "corrective step" is performed to fine-tune all model parameters together, allowing the weak learners to adjust to each other.
- **Simpler Boosting:** This implementation's boosting mechanism is based on passing the cumulative output to the next learner, rather than explicitly calculating and fitting on gradients and Hessians.
- **Trainer Class:** It includes a `GrowNetTrainer` class that encapsulates the stage-wise training and corrective step logic.

`grownet_classification_revised.py`

This script is a revised, more robust, and theoretically grounded implementation of GrowNet for classification. It more closely follows the principles of traditional gradient boosting algorithms like XGBoost, but with neural networks.

- **Gradient-based Boosting:** Unlike the previous script, this version explicitly computes first and second-order gradients (gradients and Hessians) of the loss function. Each new weak learner is trained to fit these gradients, which is a more direct application of the gradient boosting framework.
- **Global Corrective Network:** It introduces a separate, global corrective network that is trained at the end to fine-tune the entire ensemble's predictions, offering a final refinement step.
- **Wrapper Class:** It features a high-level `MicrobiomeGrowNetClassifier` wrapper class that simplifies the entire workflow, including data splitting, training, evaluation, and plotting results.
- **XGBoost Comparison:** The script includes a function to directly train and compare the performance of the GrowNet model against a standard XGBoost classifier on the same dataset, providing a useful performance benchmark.

July 20, 2025

Tasks for the Day

- Further analyzed the error distribution in coordinate prediction by model type.
- Started compiling performance comparison tables across all models for the final report.
- Conducted initial feature importance analysis to understand which microbial taxa contribute most to geographic predictions.

Notes and Observations

- Error distribution analysis reveals that errors are not uniformly distributed across geographic regions. European and North American cities show significantly lower median errors across all models, likely due to better representation in the training dataset.
- Performance comparison tables show that the ensemble approach outperforms all single models across nearly all metrics, especially in coordinate prediction precision.
- Feature importance analysis using permutation importance reveals that approximately 30 taxa account for over 80% of the predictive power, indicating a potential for model simplification.
- Scripts and results for error distribution analysis are stored in `/home/chandru/binp37/scripts/ensemble/analysis/e`

July 22, 2025

Notes and Observations

- Encountered issues with the TabPFN model when trying to scale it to the full feature set. The model appears to have memory limitations when handling more than 100 features.
- Attempted to resolve this by implementing feature selection specifically for TabPFN, but this resulted in degraded performance.
- Current workaround: Using the top 95 features (based on feature importance) specifically for TabPFN while maintaining the full feature set for other models in the ensemble.
- The modified TabPFN implementation is at `/home/chandru/binp37/scripts/ensemble/tab_pfn/tab_pfn_reduced_fe`

July 25, 2025

Tasks for the Day

- Completed comprehensive visualization pipeline for geographical prediction results.
- Finalized all figures for the manuscript, including error distribution maps and classification performance heatmaps.
- Conducted hyperparameter sensitivity analysis for the GrowNet model to understand its robustness.

Notes and Observations

- The visualization pipeline effectively demonstrates the superior performance of the ensemble model, particularly in showing the dramatic reduction in coordinate error compared to previous approaches.
- Visualization scripts are stored in `/home/chandru/binp37/scripts/visualization/`.
- GrowNet hyperparameter sensitivity analysis shows that the model is particularly sensitive to the number of weak learners and the learning rate. Performance plateaus after 25-30 weak learners, suggesting an optimal configuration.
- The sensitivity analysis results are stored in `/home/chandru/binp37/results/grownet_sensitivity_analysis/`.

July 29, 2025

Tasks for the Day

- Attempted to implement cross-dataset validation using the marine microbiome dataset.
- Started optimizing the ensemble model for improved inference speed.
- Prepared detailed model architecture diagrams for the manuscript.

Notes and Observations

- Cross-dataset validation is proving challenging due to significant differences in taxonomic profiling between the MetaSUB and marine datasets. The marine dataset uses a different sequencing depth and taxonomic classification tool.
- Current plan is to reprocess both datasets using a consistent pipeline before attempting cross-dataset validation again.
- Inference optimization shows promising results: reduced ensemble model inference time by 37% by implementing parallel prediction for base models and optimizing the meta-model structure.
- Architecture diagrams have been finalized and are stored in `/home/chandru/binp37/report/workflows/`.

August 2, 2025

Notes and Observations

- Conducted error analysis to understand the specific cases where the ensemble model fails to make accurate predictions.
- The analysis reveals that most significant errors occur in cities that are geographically distant from other sampling locations in the same continent (e.g., isolated cities).
- Another common error pattern emerges in cities with significant seasonal variations, suggesting that temporal factors may influence microbial signatures.
- Error analysis scripts and results are stored in `/home/chandru/binp37/scripts/analysis/error_analysis/`.
- Future improvement: consider incorporating temporal information as an additional feature to improve prediction accuracy.

August 5, 2025

Tasks for the Day

- Investigated the memory leak issue in the GrowNet implementation that occurs during large batch training.
- Conducted ablation studies to understand the contribution of each model in the ensemble.
- Began implementing a lightweight version of the ensemble model for potential deployment.

Notes and Observations

- Memory leak issue was traced to improper handling of computational graphs in PyTorch when accumulating gradients across multiple weak learners. Fixed by explicitly detaching intermediate tensors and implementing more aggressive garbage collection.
- The fix is implemented in `/home/chandru/binp37/scripts/grownet/grownet_classification_fixed.py`.
- Ablation studies reveal that removing any single model type from the ensemble results in performance degradation, but the most critical components are XGBoost and TabPFN. Removing these decreases continent accuracy by 3.2% and 2.7%, respectively.
- Lightweight implementation focuses on using only XGBoost, LightGBM, and TabPFN with reduced hyperparameter complexity, achieving 92% of the full ensemble's performance with 35% of the computational cost.
- Lightweight implementation is at `/home/chandru/binp37/scripts/ensemble/lightweight_ensemble/main.py`.

August 8, 2025

Notes and Observations

- Successfully integrated the hierarchical GrowNet model with the broader ensemble framework, allowing for cross-model voting at each prediction level.
- The integration improved city-level prediction accuracy by 1.2% but did not significantly affect continent or coordinate predictions.
- Observed that model combination strategies differ in effectiveness depending on the prediction task:
  - For classification (continent and city): weighted voting based on cross-validation performance works best.
  - For regression (coordinates): selecting the single best model (TabPFN) outperforms averaging or meta-model approaches.
- Updated ensemble implementation is at `/home/chandru/binp37/scripts/ensemble/integrated_ensemble/main.py`.

August 10, 2025

Tasks for the Day

- Conducted benchmark tests comparing our ensemble model with state-of-the-art methods in geospatial prediction.
- Investigated the biological significance of the most important features identified by our models.
- Started preparing a comprehensive documentation of the codebase for potential open-source release.

Notes and Observations

- Benchmark tests confirm that our ensemble approach outperforms the previous state-of-the-art mGPS model, achieving a tenfold reduction in median coordinate error (from 137 km to 13.72 km).
- Biological analysis reveals that many of the top predictive taxa are environmental microbes with known geographic distribution patterns, particularly those adapted to specific climate conditions.
- Several unexpected taxa also emerged as important predictors, including certain human-associated microbes that show strong geographic stratification. This suggests potential interactions between human populations and urban environmental microbiomes.
- Documentation is being created using Sphinx and will be hosted on a GitHub repository once finalized.
- Benchmark results and biological significance analysis are stored in `/home/chandru/binp37/results/benchmarks/` and `/home/chandru/binp37/results/biological_analysis/`, respectively.

August 13, 2025

Tasks for the Day

- Performed extensive runtime and memory usage profiling across all models.
- Optimized the coordinate prediction component of the ensemble model for better accuracy in edge cases.
- Implemented automated data quality checks for future datasets.

Notes and Observations

- Performance profiling reveals that the FT-Transformer component is the most resource-intensive, accounting for 43% of the total training time despite contributing only modestly to the final ensemble performance.
- Considering removing FT-Transformer from the lightweight ensemble implementation to further improve efficiency.
- Coordinate prediction optimization focused on handling edge cases near the poles and the international date line. Previously, these regions showed abnormally high error rates due to the discontinuity in longitude values.
- The improved coordinate handling is implemented in `/home/chandru/binp37/scripts/ensemble/coordinate_utils.p`
- Data quality check pipeline is stored in `/home/chandru/binp37/scripts/data_preprocess/quality_check.py`.

August 15, 2025

Notes and Observations

- Discovered that certain rare microbial signatures can lead to extreme outlier predictions in the coordinate regression component of the ensemble.
- This occurs primarily when a sample contains unique combinations of taxa that were poorly represented in the training set.
- Implemented an outlier detection and correction mechanism based on the uncertainty estimates from the base models. If uncertainty is above a threshold, the model defaults to city-centroid coordinates instead of attempting precise prediction.

- This approach reduced the maximum error from over 15,000 km to under 6,000 km while maintaining the same median error.
- The outlier handling code is implemented in `/home/chandru/binp37/scripts/ensemble/outlier_detection.py`.

August 18, 2025

Tasks for the Day

- Finalized the manuscript draft for internal review, integrating all results and visualizations.
- Completed the open-source package structure for eventual release of the model implementation.
- Performed final validation tests on holdout data that was not used during model development.

Notes and Observations

- The manuscript draft is complete and stored at `/home/chandru/binp37/report/final_draft/manuscript.tex`.
- Open-source package includes comprehensive documentation, example notebooks, and containerized deployment options. The package is stored at `/home/chandru/binp37/package/`.
- Final validation tests on holdout data confirm the robustness of our approach, with performance metrics consistent with those observed during development.
- Interestingly, the holdout test revealed slightly better performance on samples from Asia and South America compared to our development test set, suggesting that the model generalizes well across diverse geographic regions.
- Next steps will focus on preparing for journal submission and addressing any feedback from internal review.