# Feature Selection in High-Dimensional Space with Applications to Gene Expression Data

Nishan Pantha*, Muthukumaran Ramasubramanian*, Iksha Gurung*,
Manil Maskey†, Lauren M. Sanders†, James Casaletto†, Sylvain V. Costes†
*The University of Alabama in Huntsville
†National Aeronautics and Space Administration

***Abstract*—Recent years have seen rapid growth in high-dimensional datasets. Most existing machine learning (ML) algorithms fail in high-dimensional settings where many features could be redundant. A critical process of feature selection is thus applied in such a setting that helps in identifying the most relevant features while removing redundant ones. With the increase in high dimensionality, one is also faced with problems of efficiency and interpretation in performing such selection methods. Therefore, this paper proposes a "novel" feature selection framework that uses an ensemble of interpretable ML algorithms to perform feature selection and the ranking of final features. Finally, this framework is applied to a gene expression dataset obtained through collaboration with the National Aeronautics and Space Administration (NASA)'s Biological and Physical Sciences (BPS) team and helps identify important and relevant genes contributing to specific target attributes through classification tasks.**

## I. Introduction

In recent years, there has been a tremendous increase in data in various domains such as IoT, genomics and bioinformatics, NLP, etc, which have led to the advent of high-dimensional datasets. This growth in dimensionality poses significant challenges to traditional machine learning and predictive modeling techniques, which often suffer from the *curse of dimensionality* [1]. This affects not only the predictive power of such models but also the space and time complexity of algorithms to build such models. To address this issue, feature selection has emerged as a crucial step in preprocessing such high-dimensional data.

Feature selection is an important preprocessing step in machine learning to build predictive models. It is essential to identify and select a subset of relevant features from a larger set of features that are used to build predictive models. The presence of irrelevant features and the absence of relevant ones can significantly hamper the performance of such models. It alleviates the curse of dimensionality, improves model interpretability, reduces overfitting, and decreases computational costs. For decades, the field of feature selection has flourished, driven by the increasing popularity of various machine learning algorithms. These algorithms encompass tree-based models [2] such as decision trees and random forests [3] as well as gradient-boosted methods [4] like Extreme Gradient Boosting (XGBoost) [5]. Furthermore, transformation-based techniques such as Principal Component Analysis (PCA) [6] and SVD [7] has also played a significant role in this domain.

Feature selection methods are pivotal in machine learning for improving model performance and interpretability, and a multitude of such methods have been developed, each falling into one of several broad categories. Filter methods, for instance, assess the relevance of each feature independently using statistical measures like Pearson's Correlation, ANOVA [8], and LDA [9], and are prized for their simplicity and universality across different machine learning algorithms. On the other hand, wrapper methods take a more computational approach, utilizing search algorithms to determine the optimal subset of features by evaluating numerous combinations based on model performance, techniques such as Recursive Feature Elimination [10] and Forward Feature Selection [11] are notable examples. Embedded methods like LASSO [12] and Ridge regression [13] integrate feature selection as part of the model training process, thereby combining the advantages of both filter and wrapper methods. Lastly, dimensionality reduction techniques such as PCA [6] and SVD [7] transform the feature space into a new set of variables that retain the most significant data attributes, although this can make the interpretation of these features more challenging. Each of these methods plays a crucial role in tackling the challenges posed by high-dimensional datasets in predictive modeling.

We present a machine learning framework that integrates tree-based and gradient-boosted methods for feature selection and ranking in high-dimensional datasets, capitalizing on the interpretability of tree-based approaches like Random Forest [3] and the sequential refinement of models such as XGBoost. This framework, which blends wrapper and embedded method advantages, conducts a thorough parameter search, model training across various data splits, and feature aggregation to establish final rankings—essentially framing training as a supervised learning challenge aimed at multi-class classification. Tested on high-dimensional liver RNA sequencing data in collaboration with NASA's Bio-Physical Science team, our framework effectively identifies feature subsets significantly associated with target classification attributes, with detailed insights and results expounded in subsequent chapters.

## II. Related Works

This section reviews research on high-dimensional data feature selection, focusing on universal techniques and those specific to genomics and transcriptomics.

### A. General Feature Selection Survey

We discuss feature selection in high-dimensional data, highlighting filter, wrapper, and embedded methods. Mutual information-based filter methods, like those reviewed by Vergara and Estévez [14], are significant. Yu and Liu [15] enhance this with their Correlation-based Feature Selection (CFS). Regularization techniques (Lasso, Ridge, Elastic Net), covered by Haghighi *et al.* [16], address multicollinearity in predictive models. Fan and Li [17] provide an overview of these methods, emphasizing the need for feature validation.

Ensemble-based feature selection methods, as outlined by Seijo-Pardo *et al.* [18] and Ali *et al.* [19], offer advantages in machine learning models. They propose homogeneous and heterogeneous strategies for feature ranking, which show improved classification accuracy and predictive performance.

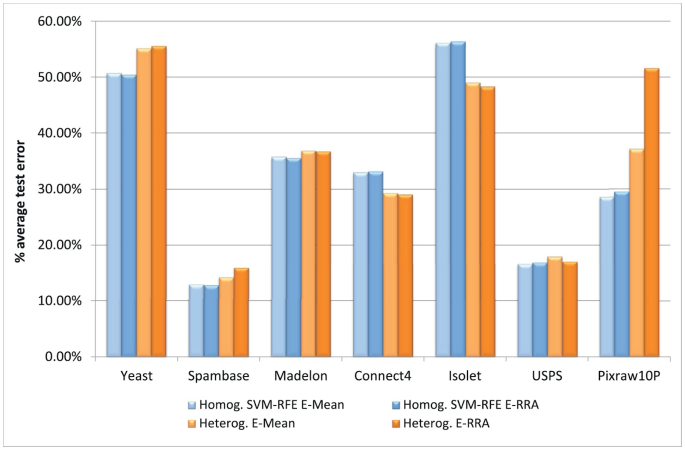



Fig. 1: Comparison between the homogenous and heterogeneous ensembles in terms of average test error by Seijo-Pardo, *et al.* [18].

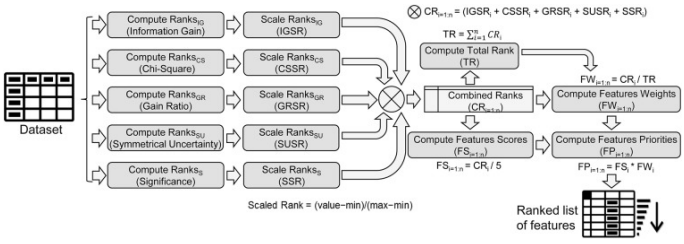



Fig. 2: UFS Algorithm proposed by Ali, *et al.* [19].

### B. Gene Expression Feature Selection Survey

Gene expression datasets require specialized feature selection approaches due to their high dimensionality and complex gene interactions. Saeys *et al.* [20] discuss methods specific to bioinformatics, emphasizing the need for incorporating biological knowledge.

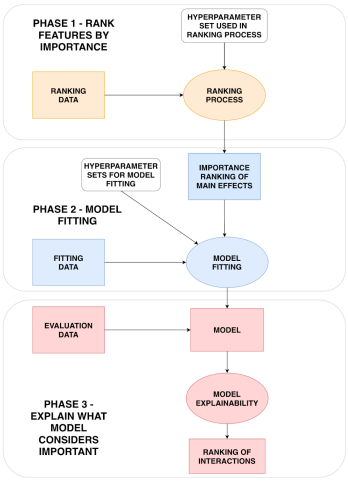



Fig. 3: Stages of ensemble method for feature selection and ranking proposed by Johnsen *et al.* [21].

Addressing challenges like multicollinearity in gene expression data, ensemble methods like those proposed by Johnsen *et al.* [21] using XGBoost with SHAP values, provide insights into gene interactions. Chen *et al.* [22] demonstrate the effectiveness of ANN methods in uncovering deeper gene interactions. Guo *et al.* [23] introduce DeepMetabolism, a deep learning system for phenotype prediction from transcriptomics data, notable for its accuracy and speed.

Our method applies an ensemble-based approach to gene expression datasets, specifically using a ranking framework validated on a dataset from National Aeronautics and Space Administration (NASA)'s Biological and Physical Sciences (BPS) team as explained in the next section.

## III. Dataset

The dataset for our experiments, obtained in collaboration with NASA's BPS division, consists of a nucleic acid sequencing (Ribonucleic acid (RNA)-sequencing) dataset from liver samples of space-flown mice. Despite its small sample size ($n = 112$), it contains a large number of genes, which makes it challenging to identify genes linked to specific traits such as gender, strain, and age. The proposed feature ranking framework is applied to this dataset for identifying important genes. This dataset is part of a larger initiative in space biology—a field focusing on the impacts of spaceflight on living systems, including studying health risks like cancer and immune system issues due to altered gravity and increased radiation.

Space biology experiments [24] often utilize model species and simulators to understand space conditions, with initiatives like NASA GeneLab and the Ames Life Sciences Data Archive

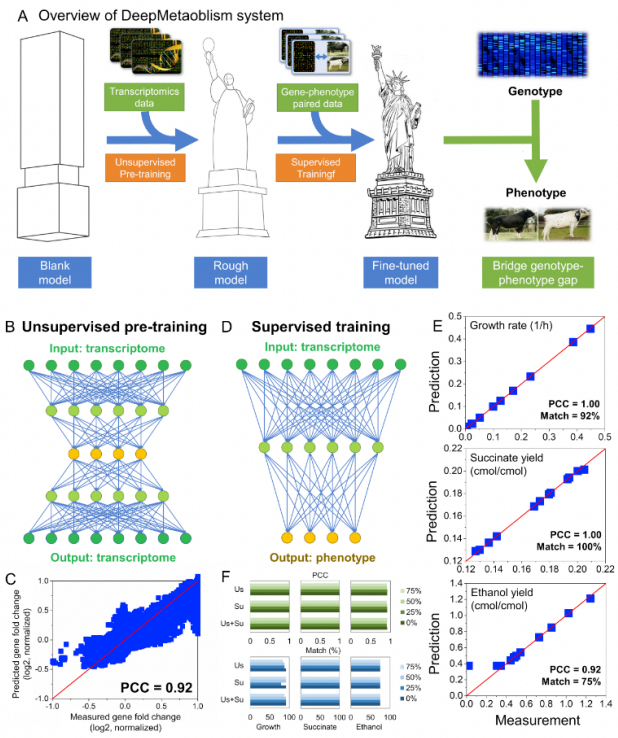



Fig. 4: DeepMetabolism proposed in Guo *et al.* [23].

(ALSDA) supporting open-source and FAIR data sharing from these investigations. These efforts prioritize machine learning readiness of datasets, despite challenges like small sample sizes and heterogeneous data. The goal is to develop algorithms suited for small-scale computing or out-of-distribution environments and transfer learning across different organisms and environments. The RNA-sequencing dataset is a key contribution from GeneLab and ALSDA, aiming to standardize benchmarking methods for identifying relevant gene features.

### A. RNA Sequencing Dataset

Microarrays and RNA-seq technology measure the expression level of thousands of genes from a biological sample. Microarrays measure specific gene activity by matching them with known gene sequences, while RNA sequencing reads all the gene messages in a sample, providing a more detailed view of what's happening in the cells. NASA Genelab has generated transcriptomic datasets using next-gen high-throughput sequencing technology. However, variations in the samples like age, strain, sex, and library preparation could potentially complicate the downstream analysis, including determining the differences in gene expressions. The metadata for this dataset is shown in the table I.

The target metadata represents the target class (Y) in our downstream methods which would be fundamental in solving the supervised classification problem while performing the feature ranking. Some of the initial preparation methods for

TABLE I: Target metadata description for gene expression dataset by BPS

| ***Target Metadata*** | ***Possible Values*** |
|---|---|
| Age at launch (in weeks) | 9, 10, 12, 16, 32 |
| Animal Return | LAR, ISST<br>(LAR: live animal return, ISST: ISS terminal) |
| Condition duration (in days) | 13, 21, 29, 33, 37, 42, 54 |
| Gender | male, female |
| Library Preparation | Ribo-depleted, polyA |
| Dissection Method | Carcass, immediate |
| Strain | C57BL6T, C57BL6J, BALBcT |

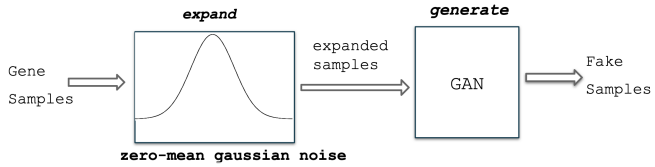



Fig. 5: Data synthesis pipeline for gene expression data.

this dataset can be grouped into the following sections based on the sources in the GeneLab:

- OSD-47 [25]: Study on microgravity's effect on muscle loss, involving transcriptomic liver data from mice (Rodent Research (RR)-1).
- OSD-48 [26]: Evaluation of Rodent Research Project's capabilities on the International Space Station (ISS) with liver samples sequenced from mice (RR-1).
- OSD-137 [27]: Investigation of a treatment for muscle and bone mass loss in space (RR-3 mission), using BALB/c strain mice.
- OSD-168 [28]: Study using External RNA Control Consortium (ERCC)-developed RNA transcripts to address RNA sequencing data variability in liver samples (RR and RR3).
- OSD-173 [29]: RNA-Seq analysis of liver samples from mice flown on STS-135, with ERCC control spike-ins.
- OSD-242 [30]: RR-9 mission focused on the molecular basis of visual impairment and joint tissue degradation in spaceflight, using male mice.
- OSD-245 [31]: RR-6 study on a novel drug's efficacy in mitigating muscle atrophy in mice during spaceflight.

### B. Synthetic RNA Transcriptomic Data

The total number of samples in the dataset was low along with very high dimensionality consisting of thousands of gene expression features. This makes it almost impossible to work on downstream tasks for computing feature selection, feature ranking, and finding gene-target relationships. As a result, synthetic data generation is necessary to upscale the sample size. That is to say, synthetic data is generated that closely matches the original sample distribution while gaining a reasonable amount of samples. The synthesis along with preprocessing happens in the stages shown in 5.

*1) Sample Amplification and Expansion:* This is the first stage where zero-mean Gaussian noise is added to the original sample to artificially expand the gene expression sample size. This is done in a way the metadata target such as gender, age, etc. remains preserved during the amplification. For instance, to double the size of a dataset with $n = 112$, one can perform the procedure once per sample, generating 112 new samples for a combined data set size of 224. Different noises should be sampled for each gene. In the case of a sample's expression vector containing 20,000 genes, this would involve sampling 20,000 times from a 0-mean Gaussian distribution. These amplified samples would later be used for training Generative Adversarial Networks (GANs) [32] to further generate new fake samples that closely match the original data distribution.

Additionally, a balanced version of the dataset is also obtained by balancing the class distribution between male and female for gender during the zero-mean Gaussian expansion process. We refer to this dataset as **balanced_expanded**. Similarly, we refer to other non-balanced datasets prefixed with "**unbalanced_**".

*2) Dimensionality Reduction:* The original data samples have a very high number of gene expressions (input feature sets), including genes that may not be statistically significant. Hence, dimensionality reduction has to be performed to reduce the input feature space. To reduce dimensionality in the dataset, genes with near-zero expressions or similar expressions across all samples are removed. This is achieved by using parameterized thresholds to evaluate the percentage of zero-expression samples, the sum of expression across samples, the difference between maximum and minimum expression values, and the variance of expression across samples. Genes that do not meet these criteria are removed from the dataset. The final result is a dataset with 25000 gene expression input features with phenotype metadata targets such as gender, age, libPrep, condition, etc.

*3) Generative Adversarial Networks:* Once expanded samples are generated with a decent number of dimensions – 25000 input gene features after the reduction – GANs [32] are trained to further generate new fake samples that closely match the original data distribution. GANs consists of two competing neural networks, a generator, and a discriminator. The generator creates fake data while the discriminator evaluates their authenticity, and through their comparison, the generator learns to produce increasingly realistic data, ultimately improving the overall performance of the model. Since, GANs are known to require rigorous sample size, the sample expansion/amplification is a necessary prerequisite to having a robust generative network. The objective loss during training the generator uses a correlation-based measure that optimizes the correlation between the real (expanded) samples and the GANs-generated fake samples.

### C. Exploratory Data Analysis

The final dataset can be primarily broken down into:

- ***expanded***: Gaussian-noise added, amplified samples
- ***fake***: GAN-generated
- ***balanced***: Gender-balanced dataset where class distribution for *male*, *female* is balanced

*1) Data Distribution:* Each sample consists of 25,000 gene expression features per sample used for conditional feature ranking. This balanced dataset is used for ranking genes for gender. All other ranking is done using the unbalanced dataset.

*2) Gene Expressions Clustering:* It is observed that gene expressions are highly correlated with each other and thus exhibit multicollinearity. This causes a major issue with any machine learning algorithm that uses such features because permuting one gene feature will have very little effect on the algorithm's performance. In such a situation, the algorithm can get the same information from any of the correlated genes. One way to handle the multicollinear gene features is to perform agglomerative hierarchical clustering [33] [34] based on correlation distance. The resulting hierarchical relationships are represented by dendrograms. Cutting these dendrograms at specific threshold results in the formation of distinct clusters where each cluster consists of many correlated gene features. To mitigate the multicollinearity nature of the dataset, we select a single representative gene feature from each cluster, which can then be used for downstream tasks.

For the clustering process, we use the ***expanded*** dataset as well as the GAN-generated ***fake*** dataset to analyze the clusters at different parameters. We primarily use absolute Spearman and Pearson correlation coefficients to compute the distance metric. We also analyze the effect of linkage types "average" and "ward" [35] [36] [34] in the formation of clusters (see figures 6 and 7).

Pearson's Correlation coefficient is defined as:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y},$$

where

- $cov(X,Y)$ is the covariance between X and Y
- $\sigma_X$ is the standard deviation of $X$ and
- $\sigma_Y$ is the standard deviation of $Y$.

Spearman's rank correlation coefficient is calculated as a Pearson correlation coefficient applied to the rank variables of the input samples:

$$\gamma_s = \rho_{R(X),R(Y)} = \frac{cov(R(X),R(Y)}{\sigma_(R(X))\sigma_(R(Y))},$$

where

- $\rho$ denotes the Pearson correlation coefficient and
- $R(X)$ and $R(Y)$ denotes the rank variables of $X$ and $Y$.

Finally, we use silhouette score [37] [38] to compute the goodness of cluster fit. The silhouette score is a widely used metric for evaluating the quality of clustering algorithms. It measures how similar an object is to its cluster (cohesion) compared to other clusters (separation). The silhouette score

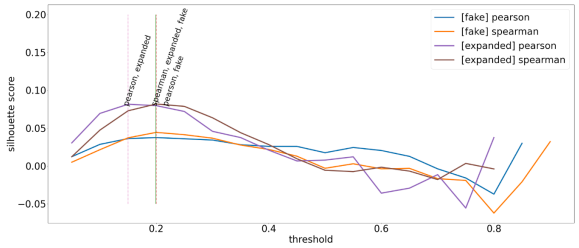



Fig. 6: Cutoff Threshold vs Silhouette Score for correlation-based clustering using average linkage.

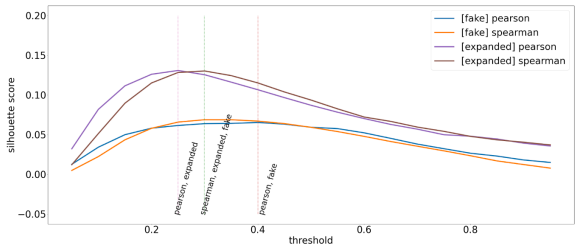



Fig. 7: Cutoff Threshold vs Silhouette Score using ward linkage.

ranges from $-1$ to 1, where a higher value indicates better clustering. The score is calculated for each sample in the dataset and then averaged to obtain a single value. Mathematically, the silhouette score $(s)$ for a $sample\ i$ is defined as:

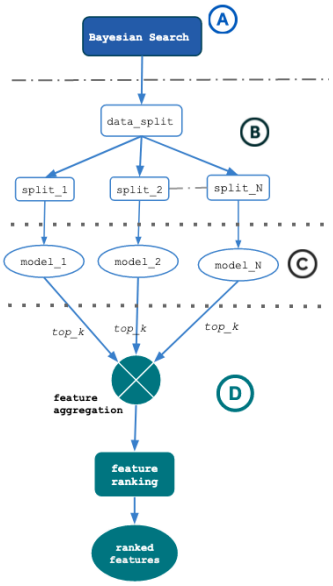
$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))},$$

where $a(i)$ represents the average distance between sample $i$ and all other samples within the same cluster, indicating cohesion. $b(i)$ represents the smallest average distance between sample $i$ and all samples in any other cluster (excluding the one to which $i$ belongs), reflecting the separation. The silhouette score for the entire dataset is then the mean of the silhouette scores for all samples. A score close to 1 suggests good clustering with well-separated clusters, whereas a score close to $-1$ indicates poor clustering with overlapping clusters.

We observe that the best clusters are formed at earlier thresholds in the range of 0.2 to 0.3. Using Spearman, we get optimal thresholds at 0.2 and 0.3 using average and ward linkages respectively. This suggests that the gene feature space has a lot of redundancy, and thus we can apply our ranking methodologies to gain insights into relevant gene features.

## IV. Method and Implementation

We develop an ensemble-based feature selection and ranking framework that works on high-dimensional datasets such as gene expression data used for the experiment, but the framework is generic enough to be applied to any broader




**A)** Perform initial XGBoost parameter search (Bayesian) to find the best parameters for classifying a given target class. These parameters are used in phase **C**).
**B)** Create N different variations of train/test split. Also, shuffle the input features column order for each split.
**C)** Train N independent XGBoost models using the best parameters found in **A)** for classifying the given class.
**D)** i)

1) Extract top-k ($k = 500$ in all experiments) features from each XGBoost model.
2) Remove all features with zero scores.
3) Normalize feature scores in the [0, 1] range.
4) Apply the ranking algorithms (intersection-based or MRR-based) to compute the final top important features.
5) Finally, get the ranked features sorted by their scores.

Fig. 8: Proposed ensemble-based feature ranking framework.

domain. The primary objective is to identify the most pertinent features that correlate to a specified target characteristic (y-value) present in the dataset (in the case of the provided gene dataset: *gender*, *strain*, etc.). To achieve this, we employ a combination of tree-based machine learning algorithms, specifically, Extreme Gradient Boosting (XGBoost). The issue is framed as a supervised multi-class classification task. Various independent XGBoost models are trained on distinct train-test data splits for the target class. Additionally, we randomly shuffle the input column order for each of these models (governed by the $shuffle\_columns$ parameter in the pipeline). This approach compels individual models to learn the same data in a uniquely distinct manner based on input column order. Lastly, we aggregate the **top-k** significant features from all these models to rank the features of interest.

### A. eXTreme Gradient Boosting (XGBoost)

Gradient boosting is a supervised machine learning technique that creates a predictive model in the form of an ensemble of weak models (called weak learners) [4] [5], typically decision trees, to create a strong learner that can make accurate predictions. Weak learners are models that perform slightly better than random predictions The idea is to iteratively/sequentially train new models to correct the errors made by the existing/previous ensemble learners. The trees are constructed greedily, focusing on the features that provide the best split at each step.

### B. Bayesian Search

Bayesian search is a probabilistic approach to search and optimization problems, based on the principles of Bayesian inference. This method combines prior information with new

evidence to update the probabilities of potential solutions. Bayesian search can be used in various applications, such as parameter estimation, model selection, and optimization, providing a flexible and robust framework for handling uncertainty and complex problem spaces.

The Bayesian inference approach uses Baye's theorem which relates the conditional probabilities of two events, say A and B as:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}.$$

The Bayesian parameter search is the first phase (phase A) in our proposed framework that is performed before training the ensemble of the XGBoost model in phase B. We search for the best parameters for the model trained to classify the given target class in the multi-class classification setting. We tune the following different parameters for methodology:

- subsample ratio of columns when constructing each tree
- learning rate and step size shrinkage used in the update to prevent overfitting
- maximum depth of a tree, the higher the value more complex the model
- L1 and L2 regularization parameters
- minimum sum of instance weight needed in a child
- parameter to control the balance of positive and negative weights, useful for unbalanced classes

### C. Ensemble Model Training

After determining optimal parameters, we train several XGBoost models, each on a different train/test split to bolster noise resistance, resulting in NN models with individual train and test scores. Scores are utilized to discard underperforming models for specific classification tasks. To enhance generalization, models are trained with shuffled input feature orders—mitigating sensitivity to feature sequence in tree-based models, a phenomenon noted in multicollinearity scenarios [39]—governed by the $shuffle\_columns$ parameter.

### D. Feature Aggregation

Post ensemble training, we consolidate and rank the $top - k$ features from each XGBoost model to form the final feature list. XGBoost offers feature importance scores using methods like frequency, gain, or SHAP values, each indicating a feature's contribution to model predictions. Our approach predominantly employs *Gain* for determining feature importance, specifically focusing on the $top\ 500$ features. Since models are independently trained, their feature scores differ in range. To rectify this, we normalize the $top - k$ features using min-max normalization to a $[0, 1]$ scale, eliminating features with zero scores. This results in $N$ normalized feature subsets that are subsequently ranked in the final stage.

### E. Feature Ranking

Once we compute the feature subset from each model from the ensemble+aggregation phase, we apply two primary ranking methods to generate the final list of features.

*1) Intersection-based Ranking:* In this method, we select all those features that are common across all the N feature subsets (each subset is contributed by an individual model from ensemble training). The final score for an individual feature is the average of all the scores observed in each subset.

*2) Mean Reciprocal Ranking:* Here, we compute the Mean Reciprocal Rank (MRR) of each feature. MRR is generally used in information retrieval particularly used for ranking search results. It is calculated as the average of the reciprocal ranks of the first relevant item across all the queries in the IR system. The reciprocal rank for a query is defined as the inverse of the rank of the first relevant item retrieved by the system. We apply this idea to the ranking step where we treat each feature subset analogous to a document for which a search is to be made. We treat each feature (overall unified feature) as a query and compute the mean reciprocal rank for each of them.

**Algorithm 1** Mean Reciprocal Ranking [MRR]

**function** COMPUTE_RANKS(feature)
  $rank\_list \leftarrow []$
  **for** $iteration = 1, 2, \ldots, N$ **do**
    $rank \leftarrow 0$
    **if** feature exists in $current\_feature\_list$ **then**
      $rank \leftarrow index + 1$
    **else if** feature does not exist in $current\_feature\_list$ **then**
      $rank \leftarrow 0$
    **end if**
    $rank\_list.append(rank)$
  **end for**
  **return** $rank\_list$
**end function**

$mrr\_map \leftarrow$ empty_dict ({})
**for** $feature_i \in \{feature_1, feature_2, \ldots, feature_N\}$ **do**
  $ranks_i \leftarrow$ COMPUTE_RANKS($feature_i$)
  $mrr \leftarrow \frac{1}{\text{len}(ranks_i)} \sum \frac{1}{r_{ij}}$ ▷ $r_{ij}$ is the jth rank of ith feature
  $mrr\_map[feature_i] \leftarrow mrr$
**end for**
Sort $mrr\_map$ in descending order based on rank cut-off value

## V. Experiments and Results

The ranking framework is applied to the gene expression dataset described in previous sections.

For the experiment, the ranking is applied for the target attributes of *Gender*, *Strain*, *Condition*, and *Age*. An ensemble of 100 XGBoost models is used for all the experiments (controlled by $n\_runs$ parameter to the pipeline). Some models are filtered out based on the downstream model selection using train and test score comparisons that are computed using the train and test data sets. 90% of the dataset is used for training and the remaining 10% as the test set for each individual model. (It is also to be noted that each model sees a slightly different train-test view of the dataset and order of input gene features.)

We use 3 gene feature sets for comparative analysis:

1) intersection-based genes (***features_intersection***)

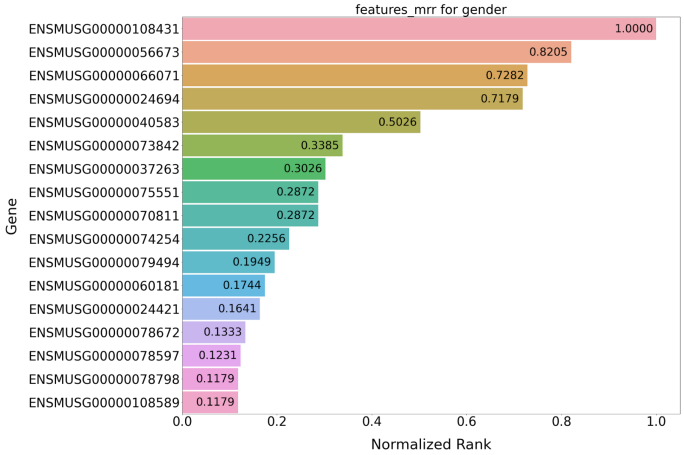



Fig. 9: MRR-aggregated *Gender* genes from an ensemble of 100 XGBoost models.

2) MRR-based (***features_mrr***)
3) full 25k gene features (***features_25k***)

To assess the efficacy of our top-ranked feature sets, we compare the classification performance of models using these sets against a model employing all the features. We train three distinct models, each with a different feature set: *features_mrr*, *features_intersection*, and *features_25k*. These models are trained on the fake dataset and evaluated on a separate, expanded dataset. We find that models using the highest-ranked features perform comparably to those with the original 25k feature set. Additionally, for our final analysis, we exclude genes that rank lower in the top-k ($k = 500$), applying a threshold that varies based on the type of attributes being examined.

### A. Gender

We use the *balanced* gender dataset for ranking genes attributing to **Gender**. The train/test scores from the models are used to filter the models during the final ranking. For *Gender*, no models are removed even after applying the filter of $trainscore = 1.0$ and $testscore = 1.0$. This gives us all 100 models for the final ranking.

We observe that all the 3 feature sets – *features_mrr*, *features_intersection* and *features_25k* – demonstrate equivalent predictive performance for gender classification. This also suggests that there could be a relatively smaller number of *Gender* genes that impact the model's outcome.

### B. Strain

We use the unbalanced dataset for ranking genes attributing to *Strain*. While we utilize the full 100 models for intersection-based ranking, for MRR-based ranking we take those models whose evaluation performance scores on the train and test sets are 1.0 respectively. This gives 76 out of 100 models used for computing the *features_mrr* feature subset. The MRR-based final ranked list has a cutoff threshold of 0.1.

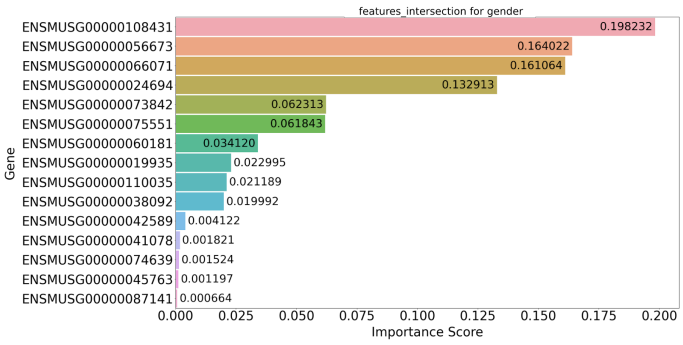



Fig. 10: *Gender* genes that are common across the ensemble of 100 XGBoost models.

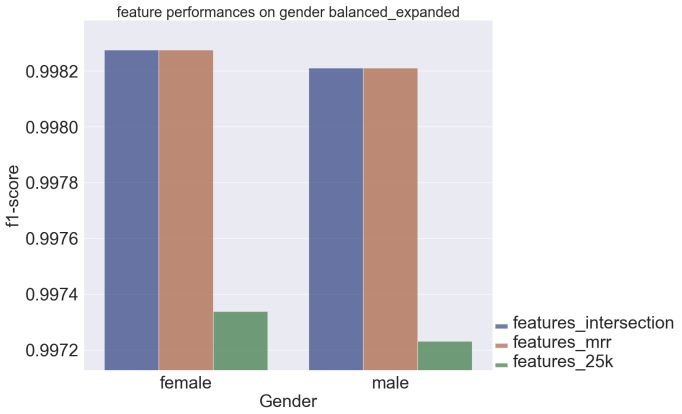



Fig. 11: Performance of a new XGBoost model per *Gender* class using the ranked genes w.r.t the original 25k genes.

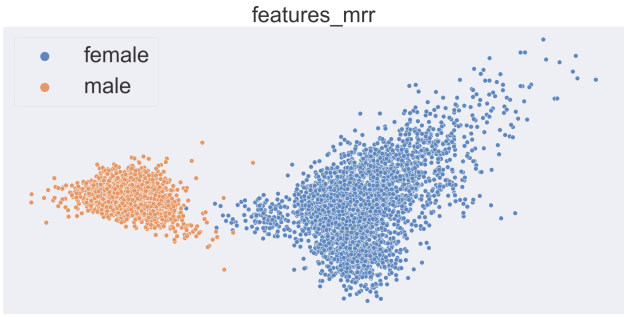
### C. Condition

We use the unbalanced dataset for ranking genes attributing to ***condition***. We use all 100 models for both intersection-based and MRR-based ranking. We observe that only a single gene consistently emerges within the $top - 500$ features across all the 100 models. Also, the MRR-based final ranked list has a cutoff threshold of 0.1. Additionally, we observe that there's only a single gene feature *ENSMUSG00000020766* for an intersection-based feature set.

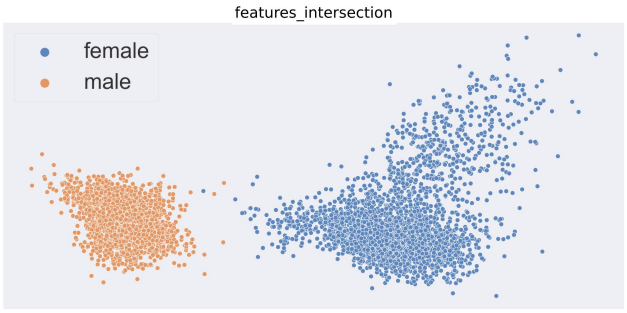
### D. Age

We use the unbalanced dataset for ranking genes attributing to *age*. We use all 100 models for both intersection-based and MRR-based ranking. We also treat this target attribute as a classification problem because the rodents are observed under discrete age (in week) conditions as mentioned in the GeneLab RNA sequencing datasets section.

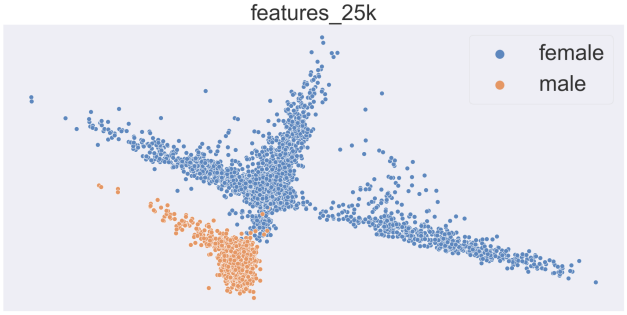
In conclusion, we observe that the ranked gene features have comparable performance in predictive modeling as compared to the original 25000 genes.




(a) `total explained_variance_ratio = 0.972`




(b) `total explained_variance_ratio = 0.945`




(c) `total explained_variance_ratio = 0.446`

Fig. 12: `PCA plots (n_components=2) of` *`balanced_expanded`* `dataset using ranked and and full 25k features for` *`Gender`*`.`

## VI. Conclusion

We propose a novel ranking framework for feature selection and ranking in high-dimensional datasets. It is specifically tailored to use tree-based ensemble models (such as XGBoost). By applying this framework to gene expression data provided by NASA's Bio-Physical Science team, it is shown that the framework is effective in computing relevant features for target metadata features such as *gender*, *condition*, *strain* and *age*. By selecting a subset of ranked features, it is demonstrated that the ranked features can achieve comparable performance in predictive models w.r.t the original full 25000 feature set. This also reduces the computational complexity and enhances the interpretability of the model. This finding highlights the efficiency of this ranking framework in identifying the most important features for prediction tasks while minimizing the risk of overfitting.

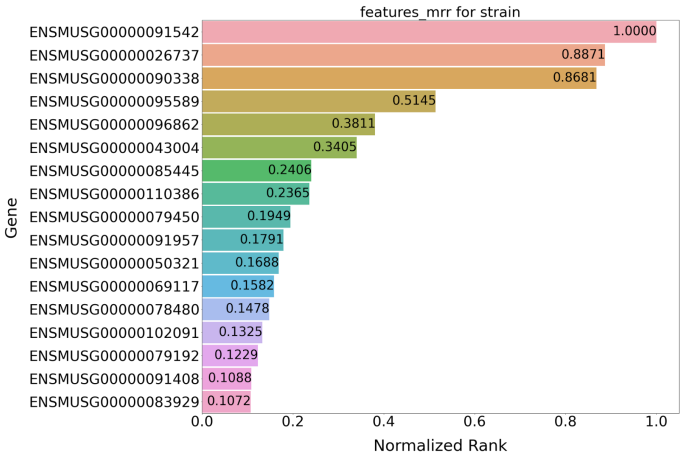



Fig. 13: MRR-aggregated *Strain* genes from the ensemble of 76 XGBoost models.

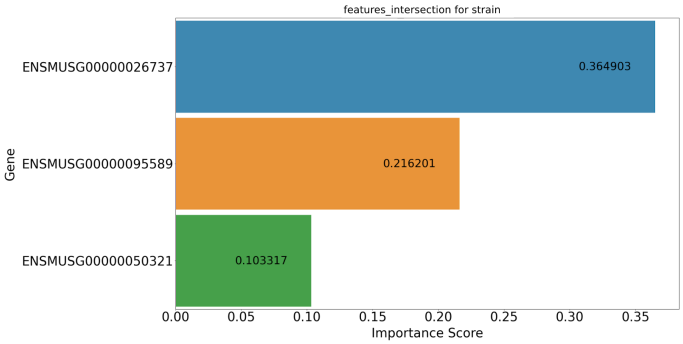



Fig. 14: *Strain* genes that are common across the ensemble of 100 XGBoost models.

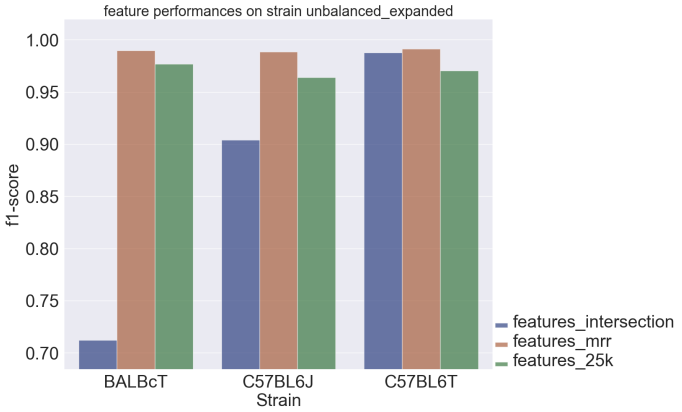



Fig. 15: Performance of a new XGBoost model per *Strain* class using the ranked genes w.r.t the original 25k genes.

Furthermore, we foresee the ranking results from the experiments to create, release and publish a benchmark dataset in collaboration with NASA's BPS team. This framework's ap-

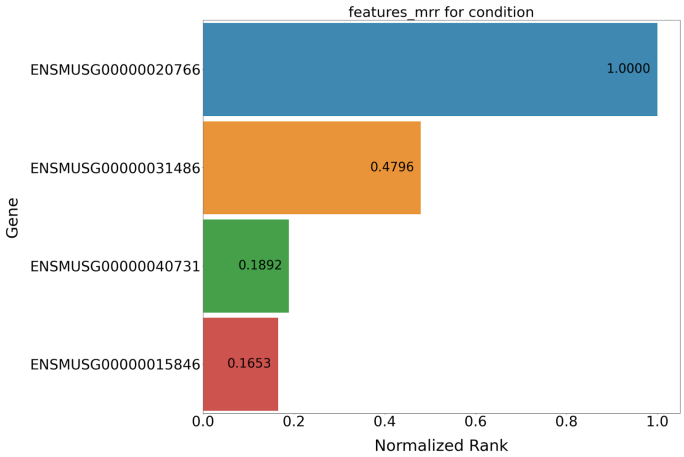



Fig. 16: MRR-aggregated *Condition* genes from the ensemble of XGBoost models.

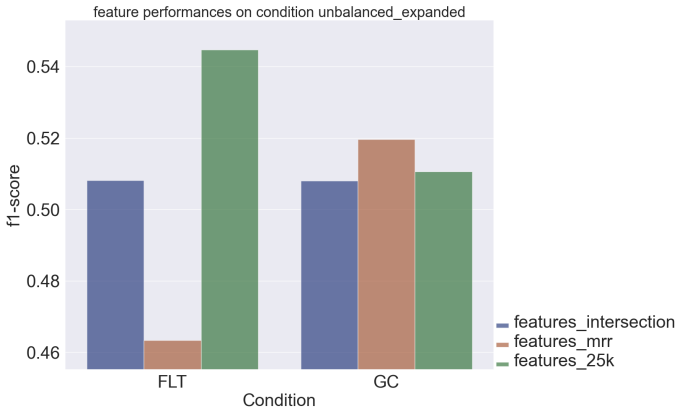



Fig. 17: Performance of a new XGBoost model per *Condition* class using the ranked genes w.r.t the original 25k genes.

plication to gene expression data has the potential to contribute to the advancement of personalized medicine and biomarker discovery, as it allows for a more efficient analysis of complex biological data. By identifying the most relevant features for a specific metadata attribute, researchers and clinicians can gain valuable insights into the underlying molecular mechanisms and develop targeted therapeutic interventions.

## VII. Acknowledgement

We express our sincere appreciation to the team at the NASA-IMPACT, UAH lab for their invaluable support in our work on machine learning and bioinformatics. Specifically, we extend our gratitude to the members of NASA's Biophysical Science (BPS) team, namely Lauren M. Sanders, Sylvain V. Costes, and James Casaletto, for their critical insights on the dataset provided through NASA's GeneLab.

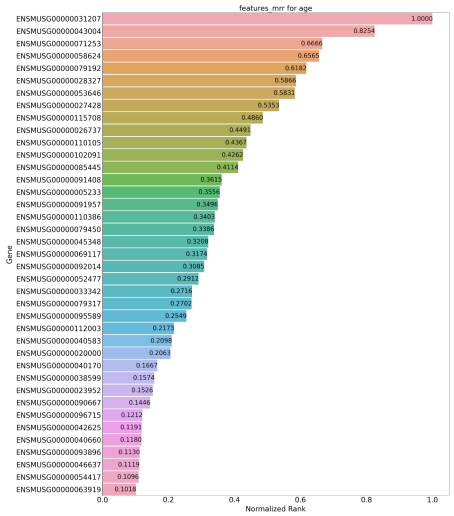



Fig. 18: MRR-aggregated *Age* genes from the ensemble of XGBoost models.

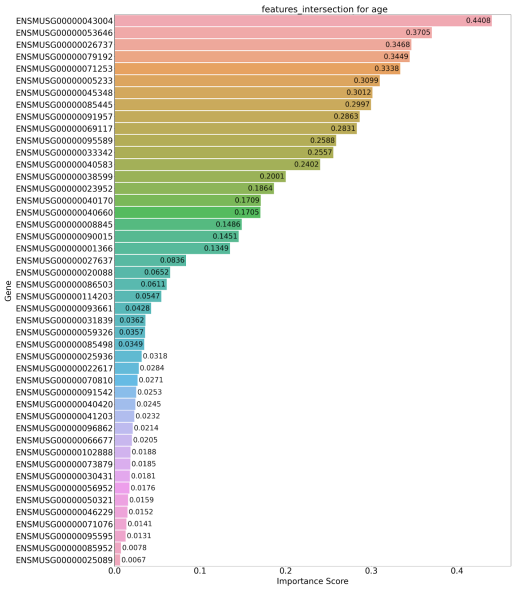



Fig. 19: *Age* genes that are common across the ensemble of XGBoost models.

## References

[1] M. Köppen, "The curse of dimensionality," in *5th online world conference on soft computing in industrial applications (WSC5)*, vol. 1, 2000, pp. 4–8.

[2] L. A. Clark and D. Pregibon, "Tree-based models," 1992.

[3] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[4] J. H. Friedman, "Greedy function approximation: A gradient boosting machine." *Annals of Statistics*, vol. 29, pp. 1189–1232, 2001.

[5] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[6] I. T. Jolliffe, "Principal component analysis," in *International Encyclopedia of Statistical Science*, 2002.

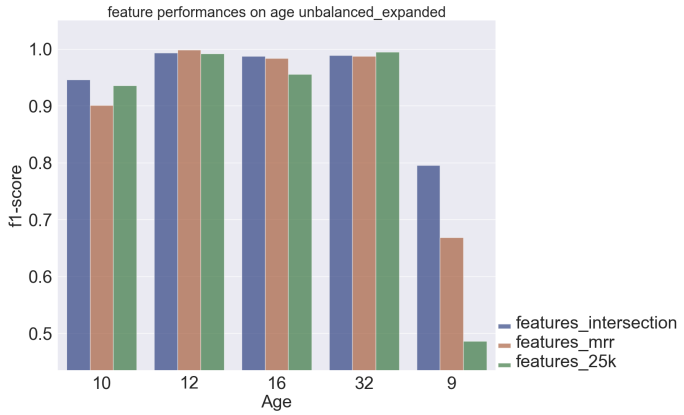



Fig. 20: Performance of a new XGBoost model per *Age* class using the ranked genes w.r.t the original 25k genes.


[7] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, pp. 1253–1278, 2000.

[8] L. St, S. Wold *et al.*, "Analysis of variance (anova)," *Chemometrics and intelligent laboratory systems*, vol. 6, no. 4, pp. 259–272, 1989.

[9] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, "Linear discriminant analysis: A detailed tutorial," *AI communications*, vol. 30, no. 2, pp. 169–190, 2017.

[10] X. Zeng, Y.-W. Chen, and C. Tao, "Feature selection using recursive feature elimination for handwritten digit recognition," in *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2009, pp. 1205–1208.

[11] M. Dash and H. Liu, "Feature selection for classification," *Intelligent data analysis*, vol. 1, no. 1-4, pp. 131–156, 1997.

[12] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[13] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[14] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," *Neural Computing and Applications*, vol. 24, no. 1, p. 175–186, Jan 2014.

[15] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," p. 8.

[16] M. Haghighi, J. C. Caicedo, B. A. Cimini, A. E. Carpenter, and S. Singh, "High-dimensional gene expression and morphology profiles of cells across 28,000 genetic and chemical perturbations," *Nature Methods*, vol. 19, no. 1212, p. 1550–1557, Dec 2022.

[17] J. Fan and R. Li, "Statistical challenges with high dimensionality: Feature selection in knowledge discovery," no. arXiv:math/0602133, Feb 2006, arXiv:math/0602133. [Online]. Available: http://arxiv.org/abs/math/0602133

[18] B. Seijo-Pardo, I. Porto-Díaz, V. Bolón-Canedo, and A. Alonso-Betanzos, "Ensemble feature selection: Homogeneous and heterogeneous approaches," *Knowledge-Based Systems*, vol. 118, p. 124–139, Feb 2017.

[19] M. Ali, S. I. Ali, D. Kim, T. Hur, J. Bang, S. Lee, B. H. Kang, and M. Hussain, "uefs: An efficient and comprehensive ensemble-based feature selection methodology to select informative features," *PLOS ONE*, vol. 13, no. 8, p. e0202705, Aug 2018.

[20] Y. Saeys, I. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, p. 2507–2517, Oct 2007.

[21] P. V. Johnsen, S. Riemer-Sørensen, A. T. DeWan, M. E. Cahill, and M. Langaas, "A new method for exploring gene-gene and gene-environment interactions in gwas with tree ensemble methods and shap values," p. 2020.05.13.20100149, Jun 2020. [Online]. Available: https://www.medrxiv.org/content/10.1101/2020.05.13.20100149v2

[22] Y. Chen, Y. Li, R. Narayan, A. Subramanian, and X. Xie, "Gene expression inference with deep learning," *Bioinformatics*, vol. 32, no. 12, p. 1832–1839, Jun 2016.

[23] W. Guo, Y. E. Xu, and X. Feng, *DeepMetabolism: A Deep Learning System to Predict Phenotype from Genome Sequencing*, May 2017. [Online]. Available: http://biorxiv.org/lookup/doi/10.1101/135574

[24] E. Afshinnekoo, R. T. Scott, M. J. MacKay, E. Pariset, E. Cekanaviciute, R. Barker, S. Gilroy, D. Hassane, S. M. Smith, S. R. Zwart *et al.*, "Fundamental biological features of spaceflight: advancing the field to enable deep-space exploration," *Cell*, vol. 183, no. 5, pp. 1162–1184, 2020.

[25] G. J. G. R. C. S, "Rodent research-1 (rr1) national lab validation flight: Mouse liver transcriptomic, proteomic, epigenomic and histology data," 2015. [Online]. Available: https://osdr.nasa.gov/bio/repo/data/studies/OSD-47

[26] G. R. M. O. G. J. G. S. C. S. C. K. C. R. B. V. L. P. S. P. K. S.-B. A. B. N, "Rodent research-1 (rr1) nasa validation flight: Mouse liver transcriptomic, proteomic, epigenomic and histology data," 2015. [Online]. Available: https://osdr.nasa.gov/bio/repo/data/studies/OSD-48

[27] S. R. C. M. G. R. G. J, "Rodent research-3-casis: Mouse liver transcriptomic, proteomic, epigenomic and histology data," 2017. [Online]. Available: https://osdr.nasa.gov/bio/repo/data/studies/OSD-137

[28] G. J, "Rr-1 and rr-3 mouse liver transcriptomics with and without ercc control rna spike-ins," 2020. [Online]. Available: https://osdr.nasa.gov/bio/repo/data/studies/OSD-168

[29] C. S. C. K. G. S. L. P. S. S.-B. A. F. H. B. V. L. P. S. G. C. U. C. S. J. K. G. JM, "Sts-135: Mouse liver transcriptomics using rna-seq," 2018. [Online]. Available: https://osdr.nasa.gov/bio/repo/data/studies/OSD-173

[30] G. J. L. P. S. S.-B. A. F. H. B. N. C. Y. N. S. D. M. C. S. G. SG, "Effect of spaceflight on liver from mice flown on the iss for 33 days: transcriptional analysis," 2019. [Online]. Available: https://osdr.nasa.gov/bio/repo/data/studies/OSD-242

[31] G. J. L. P. S. S.-B. A. F. H. B. N. B. V. D. M. C. Y. C. S. G. SG, "Transcriptional analysis of liver from mice flown on the rr-6 mission," 2019. [Online]. Available: https://osdr.nasa.gov/bio/repo/data/studies/OSD-245

[32] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[33] S. Patel, S. Sihmar, and A. Jatain, "A study of hierarchical clustering algorithms," in *2015 2nd international conference on computing for sustainable global development (INDIACom)*. IEEE, 2015, pp. 537–541.

[34] F. Murtagh and P. Contreras, "Methods of hierarchical clustering," *arXiv preprint arXiv:1105.0121*, 2011.

[35] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, 1963.

[36] F. Murtagh and P. Legendre, "Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion?" *Journal of classification*, vol. 31, pp. 274–295, 2014.

[37] S. Aranganayagi and K. Thangavel, "Clustering categorical data using silhouette coefficient as a relocating measure," in *International conference on computational intelligence and multimedia applications (ICCIMA 2007)*, vol. 2. IEEE, 2007, pp. 13–17.

[38] Contributors to Wikimedia projects, "Silhouette (clustering) - Wikipedia," Jan. 2023, [Online; accessed 11. May 2023]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Silhouette_(clustering)&oldid=1136412923

[39] A. Liaw, M. Wiener *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.