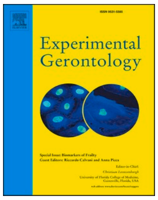
Experimental Gerontology 173 (2023) 112107

Contents lists available at ScienceDirect

Experimental Gerontology

journal homepage: www.elsevier.com/locate/expgero



# A meta-analysis of RNA-Seq studies to identify novel genes that regulate aging

Mohamad D. Bairakdar[a], Ambuj Tewari[a,b], Matthias C. Truttmann[c,d,*]

[a] Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA
[b] Department of Statistics, University of Michigan, Ann Arbor, MI 48109, USA
[c] Department of Molecular & Integrative Physiology, University of Michigan, Ann Arbor, MI, 48109, USA
[d] Geriatrics Center, University of Michigan, Ann Arbor, MI 48109, USA

ARTICLE INFO

Section Editor: Sean Curran

*Keywords:*
Aging
RNAseq
*C. elegans*
Longevity
Reproducibility
Machine learning

ABSTRACT

Aging is a ubiquitous biological process that limits the maximal lifespan of most organisms. Significant efforts by many groups have identified mechanisms that, when triggered by natural or artificial stimuli, are sufficient to either enhance or decrease maximal lifespan. Previous aging studies using the nematode *Caenorhabditis elegans* (*C. elegans*) generated a wealth of publicly available transcriptomics datasets linking changes in gene expression to lifespan regulation. However, a comprehensive comparison of these datasets across studies in the context of aging biology is missing. Here, we carry out a systematic meta-analysis of over 1200 bulk RNA sequencing (RNASeq) samples obtained from 74 peer-reviewed publications on aging-related transcriptomic changes in *C. elegans*. Using both differential expression analyses and machine learning approaches, we mine the pooled data for novel pro-longevity genes. We find that both approaches identify known and propose novel pro-longevity genes. Further, we find that inter-lab experimental variance complicates the application of machine learning algorithms, a limitation that was not solved using bulk RNA-Seq batch correction and normalization techniques. Taken as a whole, our results indicate that machine learning approaches may hold promise for the identification of genes that regulate aging but will require more sophisticated batch correction strategies or standardized input data to reliably identify novel pro-longevity genes.

## 1. Introduction

Lifespan regulation is a critical biological process that has garnered significant attention in recent years.

The application of computational models, in particular machine learning algorithms, to the analysis of large-scale datasets, has facilitated the identification of novel genetic factors regulating lifespan and aging-associated processes (Liu et al., 2020; Palmer et al., 2021; Shokhirev and Johnson, 2022).

Previous studies using machine learning to examine aging genes have shown promise, yet present several challenges (Fabris et al., 2017). Most studies used publicly-available gene ontology (GO) information, protein-protein interaction (PPI) data, gene expression data, or combinations of these as features to characterize aging genes before inputting them into machine learning models (Fabris and Freitas, 2016; Fabris et al., 2019; Fang et al., 2013; Freitas et al., 2011; Huang et al., 2012; Jiang and Ching, 2011; Kerepesi et al., 2018; Wan and Freitas, 2013). Similarly, some studies included RNA interference (RNAi) phenotypes and gene conservation scores as features for the generation of predictive models (Li et al., 2010).

Commonly-used machine learning models included Bayesian classifiers, k-NN, Logistic Regression (LR), Decision Tree, Support Vector Machine (SVM), Random Forest (RF), hierarchy-aware classifiers, Extreme Gradient Boosting (XGBoost or XGB), and Deep Neural Networks (DNN). Other studies resorted to more traditional differential expression workflows to conduct similar analyses (De Magalhães et al., 2009; Fernandes et al., 2016).

Most of these approaches resulted in classifiers that could associate well-known aging genes with aging. Feature extraction on these models showed that they are capable of learning facts about aging genes. However, the potential to propose novel pro-longevity genes was generally limited. One of the reasons for this limitation is the reliance on GO terms and PPI features inputs, which introduce bias into the model, potentially masking novel regulators of aging (Fabris et al., 2019; Wan

[*] Corresponding author at: Department of Molecular & Integrative Physiology, University of Michigan, Ann Arbor, MI 48109, USA.
*E-mail address:* mtruttma@med.umich.edu (M.C. Truttmann).

https://doi.org/10.1016/j.exger.2023.112107
Received 10 November 2022; Received in revised form 17 January 2023; Accepted 23 January 2023
Available online 1 February 2023

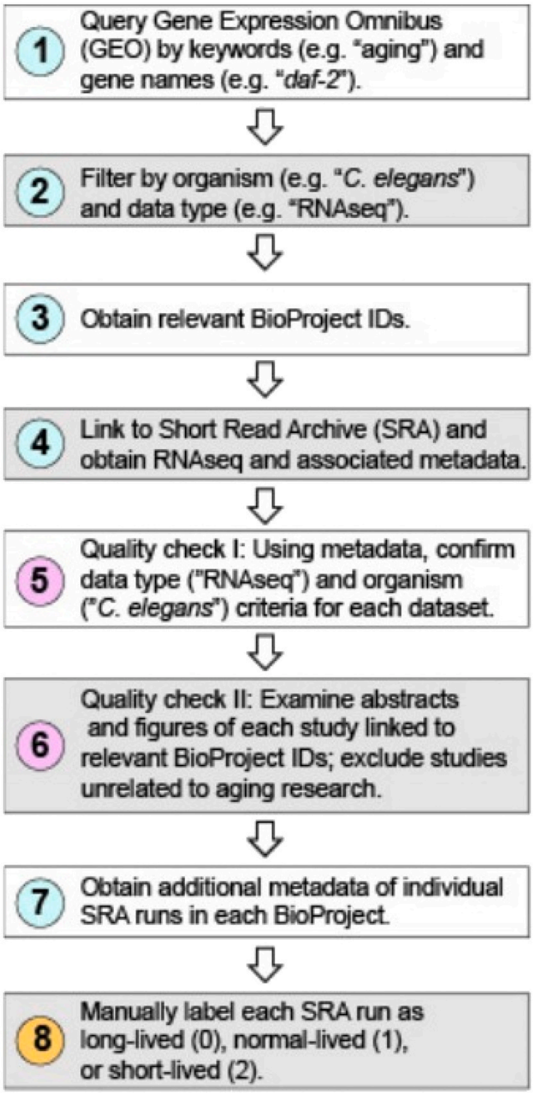

1
Query Gene Expression Omnibus (GEO) by keywords (e.g. "aging") and gene names (e.g. "daf-2").
2
Filter by organism (e.g. "C. elegans") and data type (e.g. "RNAseq").
3
Obtain relevant BioProject IDs.
4
Link to Short Read Archive (SRA) and obtain RNAseq and associated metadata.
5
Quality check I: Using metadata, confirm data type ("RNAseq") and organism ("C. elegans") criteria for each dataset.
6
Quality check II: Examine abstracts and figures of each study linked to relevant BioProject IDs; exclude studies unrelated to aging research.
7
Obtain additional metadata of individual SRA runs in each BioProject.
8
Manually label each SRA run as long-lived (0), normal-lived (1), or short-lived (2).

**Fig. 1.** Data collection and dataset curation process.

et al., 2015). This is because resources such as the GO and PPI databases are based on published findings regarding gene function. Thus, features input from such resources are necessarily limited to the extent to which new knowledge can be generated. Moreover, current knowledge, and hence GO and PPI features, are incomplete, which may limit the predictive power of models relying on these inputs.

When a more unbiased feature such as gene expression data was used as input in previous studies, the training examples were individual genes, rather than individual samples. In other words, if we assume that gene expression data is organized as a matrix in which rows represent genes and columns represent samples from separate individuals or experiments, then the input to the model is a row from this matrix instead of a column. A drawback of this approach, which we refer to as the "per-gene basis" approach to gene expression, is that information about interactions between genes and gene modules cannot be extracted from a model built under this premise. Given these previous limitations, the goal of our study is to establish a model to predict aging-associated genes and gene clusters that adheres as much as possible to the following criteria:

- **Criterion 1**: Using unbiased input features
- **Criterion 2**: Using per-sample rather than per-gene data as input

Unbiased input features in this context refers to the use of gene expression data as opposed to other types of features, for example, gene ontology terms.

We chose to conduct our analysis using *C. elegans* transcriptomics data. Over the past two decades, a wealth of longevity-focused, high-quality *C. elegans* studies identified key regulators of aging and lifespan (Kaeberlein et al., 2006; Leiser et al., 2015a; Uno and Nishida, 2016), perhaps most notably the *daf-2*-dependent insulin signaling pathway (Kenyon et al., 1993; Lin et al., 2001). Importantly, many of these studies included a global transcriptome assessment by RNA sequencing (RNA-Seq), the results of which were often deposited in the National Center for Biotechnology Information (NCBI) Short Read Archive (SRA) database. Here, we take advantage of these publicly available datasets and take an inter-lab, meta-analytical approach to derive novel insights about the genetic basis of longevity and aging.

We test and compare the performance of four different modeling approaches, two that are based on machine learning, and two that are based on differential expression analysis. Each approach is consistent with the aforementioned criteria to a different extent (see Methods). Our machine learning models, which were trained to classify samples based on their longevity class (long- vs. short- vs. normal-lived) using their complete transcriptomics data, achieved a cross validation accuracy of up to 65 %. Finally, we report that despite extensive attempts for batch correction and data normalization, lab-specific experimental bias (noise) remains a key problem that prevents effective inter-lab analyses. Our work provides a glimpse at the promise of machine learning and big data analyses in aging research while highlighting the need for more standardized sample collection and processing to increase inter-lab comparability of RNA-seq datasets.

## 2. Methods

### 2.1. RNA-Seq data collection

The schematic in Fig. 1 describes our data collection pipeline. To obtain RNA-Seq studies published up until August 2020 focused on aging and longevity in *C. elegans*, we queried the NCBI databases in three ways using NCBI's e-utilities on the command line:

1. We queried the Gene Expression Omnibus (GEO) database for any study mentioning one of the following terms: "lifespan", "aging", "life span", "longevity", "senescence", "aging", "longlived", "short-lived", "long-lived", "short-lived", "stress".
2. We queried the GEO database for any RNA-Seq study whose summary mentions a list of aging associated genes that are listed in the GenAge database, Build 20.
3. We queried the Sequence Read Archive (SRA) database by BioProject ID to obtain metadata for additional relevant studies that we are aware of (PRJNA261420).

We then filtered these results to meet both organism (*C. elegans*) and method (RNA-Seq) criteria for inclusion. The resulting studies were then manually reviewed to ensure topic relevance to *C. elegans* aging research, which required that discussed mutations or conditions were shown to either enhance or diminish lifespan. This process yielded a total of 74 peer-reviewed studies, which were then further refined using linked SRA metadata to exclude any individual runs in each study that failed to match both organism (*C. elegans*) and method (RNA-Seq) criteria. A full list of studies included in the meta-analysis can be found in Supplementary File S1. From here, each RNA-Seq sample was manually labeled as being either "short-lived" (2), "normal-lived" (1), or "long-lived" (0) based on interpretation of the results associated with each RNA-seq dataset in the original publication. Briefly, data from untreated N2 (wild-type) worms were labeled as "normal" (1), data of short-lived mutants or lifespan-reducing treatments were labeled as "short-lived" (2), and data of long-lived mutants or lifespan-extending conditions were labeled as “long-lived” (0). Studies for which we were unable to assign aging labels with high confidence were excluded from analysis (see Supplementary File S1).

Next, we extracted the age of the *C. elegans* cohorts used for RNA-seq analysis. Some samples were labeled as “young adult,” which we considered to be day 1 adults (labeled to have age “1”). Samples that were labeled as “pre-fertile young adult” were considered to be 0.5 days old. L4, L3, L2, L1, and embryos were labeled as 0, −0.5, −1, −2, and −2.5 days respectively. If we were unable to retrieve the age of worm cohorts used in a particular sample from the information deposited in the SRA, we analyzed the corresponding study to obtain this information. If we still were not able to obtain the information, we considered the sample to be 1 day old, the mode age in the dataset.

### 2.2. RNA-Seq data mapping

RNA-Seq reads obtained from SRA were mapped using the command line tool Kallisto, version 0.46.1 (Bray et al., 2016), to the *C. elegans* reference transcriptome obtained from Ensembl Release 100. We then collapsed transcript level mappings to gene level mappings using the Tximport R package (Soneson et al., 2015). This process yielded raw read counts for 22,113 genes.

### 2.3. Pre-processing gene expression counts

To normalize the data from multiple studies, we tested multiple normalization and batch correction techniques, including trimmed mean of M values (TMM), gene length corrected trimmed mean of M values (GeTMM), Combat-Seq, and Sparse Autoencoder for Clustering, Imputing, and Embedding (SAUCIE).

TMM was first used in the R package edgeR, and was developed to address the shortcomings of naive methods like TPM and RPKM that do not take into account the fact that some genes may be highly expressed (Robinson and Oshlack, 2010). Since there is only a limited amount of space on a flow cell in an RNA-seq experiment, the presence of these highly expressed genes increases the chance of failing to obtain representative gene expression of a given sample. TMM thus excludes these genes, then uses a weighted average of the remaining genes to compute a normalization factor. TMM is usually used when comparing the same gene across samples, but cannot be used to compare across genes within the same sample because it does not correct for gene length. We thus use TMM when normalizing data that is input into traditional differential expression analysis workflows.

GeTMM is an RNA-Seq normalization technique developed by (Smid et al., 2018) that extends the TMM method to account for gene length, which is important to take into account in machine learning per-sample approaches because the model is expected to learn relationships between genes within a single genome.

Combat-Seq is an RNA-seq extension of the original Combat method – a batch correction method for microarray experiments based on empirical Bayes techniques (Zhang et al., 2020). Unlike other batch correction methods, which assume the data follows a normal distribution, Combat-Seq models the data using negative binomial regression. The assumption that other methods make tends to be inappropriate because RNA-seq data is count-based, and is typically over-dispersed. Thus, CombatSeq is superior in this regard. Second, Combat-Seq retains the count nature of the data. This makes it possible to use its output as input to normalization techniques like GeTMM or TMM.

SAUCIE is an alternative batch correction method, originally designed for single cell RNA-seq (scRNA-seq) data (Amodio et al., 2017). In contrast to Combat-Seq, it relies on deep learning for batch correcting scRNA-seq data. SAUCIE inputs gene expression profiles of individual cells into an autoencoder, which is a deep learning technique that compresses the input into a lower dimensional representation, and reconstructs the original high dimensional data from the lower dimensional representation. Many uses have been described for auto-encoders, and in their work, Amodio et al. developed it for batch correction, clustering, data imputation, and dimensionality reduction. SAUCIE performs batch correction by selecting one of the batches at random to be considered as a reference batch. Then, it corrects other batches by minimizing the Maximum Mean Discrepancy (MMD) between the reference batch and the other batch being considered. MMD penalizes differences in `the distribution of the low dimensional representations of the reference batch and the current batch. Thus, the current batch's distribution is shifted in the low dimensional space, so that once the sample is reconstructed into its original higher dimension – the gene expression space – it is batch corrected in this space. Although SAUCIE was designed for scRNA-seq, we hypothesized that it is also applicable to bulk RNA-seq, where instead of having multiple expression profiles per patient per batch (corresponding to multiple cells for that patient), we have just a single expression profile per patient per batch.

Gene expression counts were pre-preprocessed in five distinct ways before being input into machine learning algorithms:

1. GeTMM normalization followed by $\log_2$ transformation.
2. Combat-Seq experiment correction followed by GeTMM normalization and $\log_2$ transformation. The study of origin for each sample is considered as the batch variable.
3. Combat-Seq correction for age followed by GeTMM normalization and $\log_2$ transformation. The age of the worm cohorts used for sample preparation was considered as the batch variable.
4. Combat-Seq experiment correction followed by Combat-Seq correction for age, GeTMM normalization and $\log_2$ transformation.
5. GeTMM normalization followed by $\log_2$ transformation and per-experiment SAUCIE batch correction for each experiment.

As a data quality check, sequencing coverage and depth were calculated for each sample in our dataset. Dataset coverage was calculated by dividing the total number of bases per sample by the number of bases constituting the *C. elegans* reference genome (100,291,840 bp). Sequencing depth was calculated by dividing the total number of bases per sample by the average read size per sample (Hillier et al., 2005).

### 2.4. Model development

#### 2.4.1. Per sample, no gene co-expression inductive bias approach

We evaluated the following machine learning models: support vector machine (SVM) with linear kernel, linear regression (LR), randon forest (RF), extreme gradient boosting (XGB), and a Multilayer Perceptron (MLP), which is a standard type of neural network. We used the Python package Scikit-learn (Pedregosa et al., 2018) for the first three models, the Python package XGBoost for XGB (Chen and Guestrin, 2016), and built an MLP using Pytorch (Paszke et al., 2017). Each of these models encodes a different set of inductive biases, which are a set of assumptions that the model makes in learning the function that maps inputs to outputs. However, none of them incorporates prior knowledge about gene interactions. This is the key feature that distinguishes the approach described in this sub-section from the one described in the following subsection.

Supplementary Table S1 defines the hyperparameters we tested. For the MLP, the order in which the parameters are listed corresponds to the order in which they were evaluated. We trained the MLP for 100 epochs using the Adam optimizer (Kingma and Ba, 2017). The batch size was set to the total number of training instances. In the hidden layers, we used BatchNorm to stabilize training and the ReLU activation function.

A vector comprised of the age and the gene expression profile for each sample was used as input. To train and evaluate our models, we used a 90/10 % train-test split. On the training set, we used 10-fold cross validation to train and fine-tune our models. We report performance on the validation set – comprised of 10 % of the training data at every iteration of cross validation – and on the test set, with accuracy as our evaluation metric. Accuracy is a suitable metric to use because our

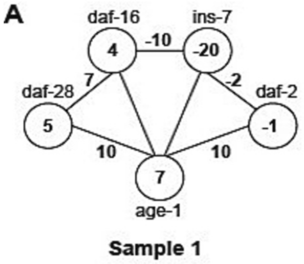


**Fig. 2.** (A) and (B): Examples of how two different samples are represented as graphs based on their gene expression values. Notice how the underlying graph structure is unchanged between both samples illustrated. The edges and their values represent StringDB gene co-expression weights between genes (nodes), which are identical between samples (A) and (B). In contrast, relative gene expression values, shown as numbers inside of nodes, are sample-specific.

dataset is quite balanced: 44.9 % of samples are labeled normal-lived, 35.9 % are labeled long-lived, and 19.2 % are labeled short-lived. For the MLP, we considered the best model to be the one that achieves the best cross validation accuracy across folds for a specific epoch.

Two different methods were used to split the data into training and validation sets: in the first, the data were split randomly, such that our models encountered samples in the training set from the same study as samples in the validation set (referred to as the "Mix split" setting). In the second approach, the data was split such that the model only observed samples from a given set of studies in the training set ("No mix split" setting). Performing well in this latter split setting would indicate that models are able to generalize beyond the specific studies on which they were trained. In both approaches, training and validation sets were stratified by longevity class, such that both sets contained the same relative proportions of long-lived, normal-lived, and short-lived samples as in the original dataset. For the test set, we used samples from studies not observed in the train-validation set. We also trained and evaluated models on five different subsets of samples or genes: i) adult worms only (778 samples), ii) larval (L4) animals and younger (449 samples), iii) all samples, but only considering aging-associated genes as defined by GenAge (1065 genes), iv) all samples, but only considering genes annotated with GO terms related to any of the following: ["aging", "longevity", "lifespan", "senescence", "stress", "aging", "cell death", "age"] (1347 genes), and v) all samples, but only considering genes with counts per million (CPM) of at least 1 in 236 or more samples (15,860 genes). We chose 236 because it is the number of samples in the condition with the smallest number of samples (the short-lived samples).

We note that the approach described in this section meets Criterion 1 and 2 above (see Introduction).

The MLP and XGBoost models were trained using a 4GB NVIDIA GeForce RTX 2060 GPU, while the SVM, RF, and LR models were trained using a CPU.

### 2.4.2. *Per-sample, with gene co-expression inductive bias approach*

To build a model injected with prior knowledge about gene interactions, we used a graph neural network (GNN) for graph classification. This approach involves generating node feature vectors (or node "embeddings") for all nodes in a graph, and then aggregating them (e.g averaging the embeddings) to obtain a graph-level embedding. This procedure is done for each graph in a dataset, and the model is trained to use these graph embeddings to classify each graph.

Given that vectors of gene expression profiles have no inherent graphical structure, we used a previously described methodology (Bertin et al., 2020; Chereda et al., 2020; Dutil et al., 2018; Ramirez et al., 2020) to impose a graph structure on the gene expression data based on gene co-expression networks. Each RNA-seq sample was represented by its own graph, where each node corresponds to one gene whose (1-dimensional) feature is represented by its expression value. Graph edges were obtained from prior modeling of gene interactions retrieved from StringDB version 11.5 (Szklarczyk et al., 2018). Thus, all RNA-seq samples were represented by the same underlying graph structure (Fig. 2). Co-expression weights between genes in StringDB were only considered for those edges for which both genes in the StringDB graph are present in the set of genes that we obtained after collapsing transcript level mappings to gene level mappings (approx. 5,000,000 edges). We did not exclude genes, depicted as nodes, that do not have any neighbors, nor did we exclude nodes that are not present in the largest connected component of the graph.

We hypothesized that using a graph structure based on prior knowledge about gene interactions confers two potential benefits. First, using prior knowledge can assist the model in learning the true underlying function between gene expression and longevity by constricting the space of functions that can be learned. This is especially important in the case of gene expression data, which is high dimensional, while the number of samples to learn from is comparatively low. Second, if the graphbased model achieves acceptable performance, it is then possible to employ graph feature extraction techniques such as GNNExplainer (Ying et al., 2019) to uncover what subgraphs – gene transcriptional modules – were most helpful for the prediction task.

We used a GNN architecture that combines the SortPooling layer introduced in the work of M.

Zhang et al. (2018) with the Graph convolutional network (GCN) layer, a graph "feature propagation and aggregation" or "feature extraction" layer introduced in Kipf and Welling (2017). The SortPooling layer from M. Zhang et al. (2018) is illustrated in Fig. 3. In the "Backbone feature extraction" step in Fig. 3, we used the feature extraction layer in Kipf and Welling (2017). Using this layer, the

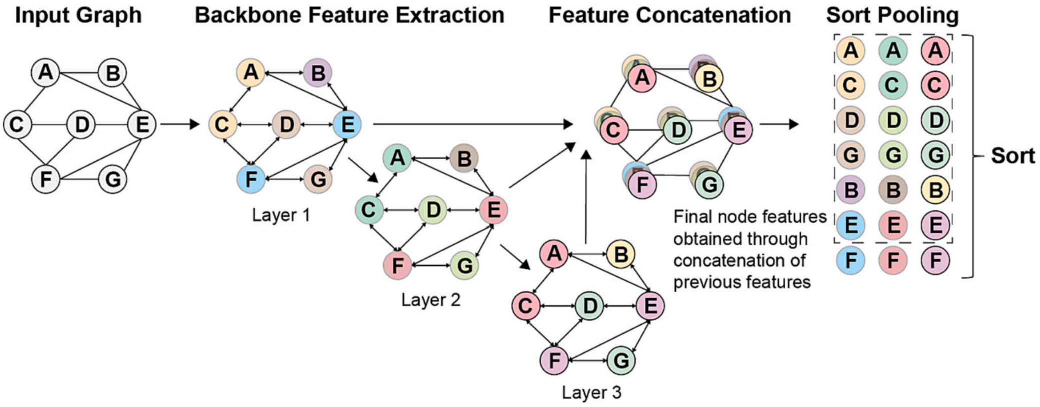

Input Graph
Backbone Feature Extraction
Feature Concatenation
Sort Pooling
A
B
C
D
E
F
G
Layer 1
Layer 2
Layer 3
Final node features obtained through concatenation of previous features
Sort

**Fig. 3.** GNN feature extraction and SortPooling layer are illustrated. In this figure, initial node features are node degrees. Nodes of the same color at iteration *k* have identical k-hop neighborhoods. Colors indicate different feature vectors, which here are 1-D, but can in practice be of any dimension.

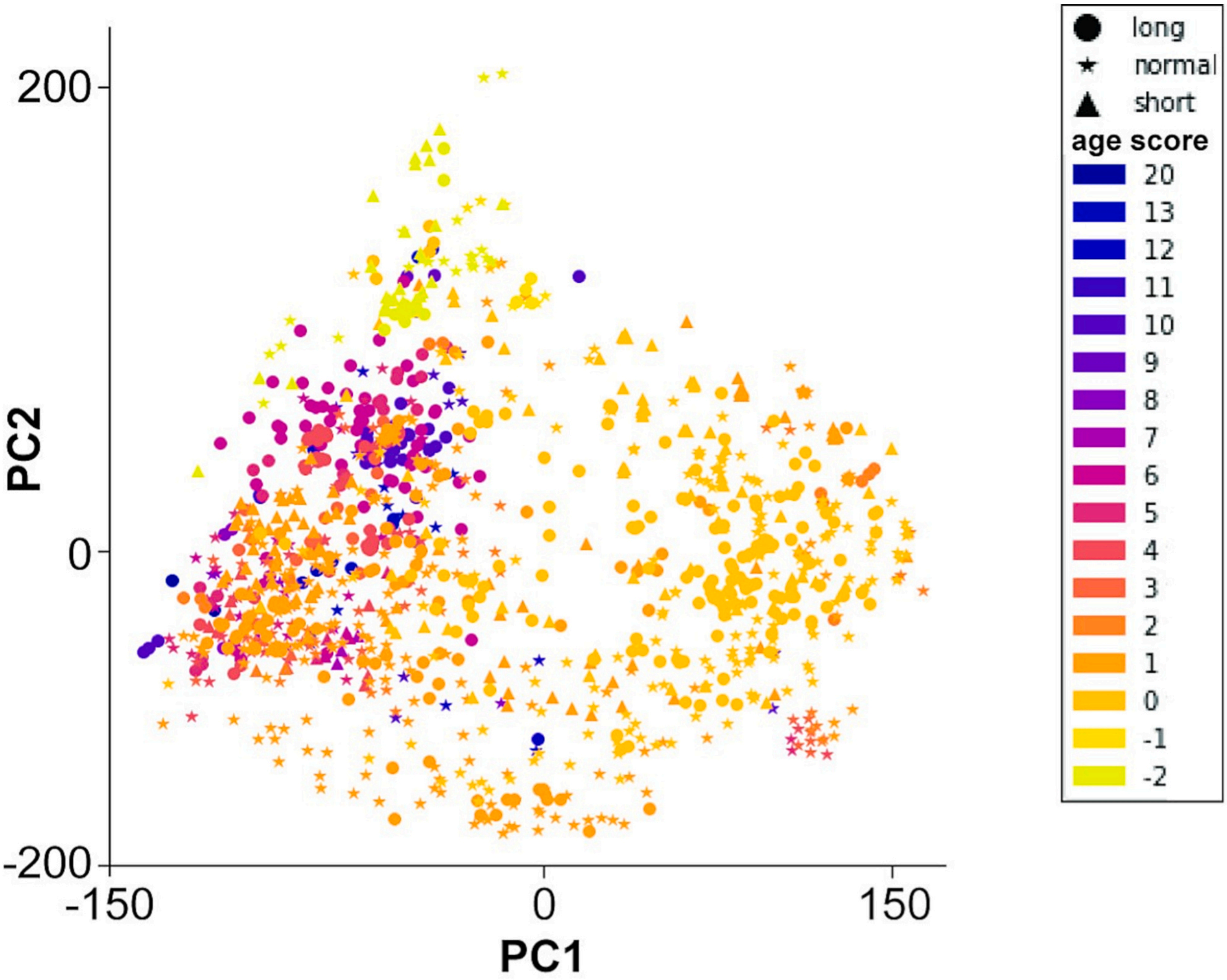


**Fig. 4.** PCA plot of Combat-Seq experiment corrected, GeTMM normalized gene expression data, where samples are color and shape coded based on their age and longevity respectively.

**Table 1**
Per sample, no gene coexpression bias, GeTMM normalized, Mix split accuracy results. ML models achieve high accuracy when the training and validation sets contain samples from the same study.

| Model | Optimal hyperparameters | Cross validation accuracy (%) |
|---|---|---|
| XGB | Max depth = 6 | 87.9 |
| RF | Max depth = 9 | 81.1 |
| SVM | C = [0.1, 1.0, 10, 1000] | 82.9 |
| LR | C = 0.1 | 79.1 |

**Table 2**
Per sample, no gene coexpression bias, GeTMM normalized, No mix split accuracy results. ML models achieve significantly lower accuracy when the studies observed in the training set are different from those observed in the validation set.

| Model | Optimal hyperparameters | Cross validation accuracy (%) |
|---|---|---|
| XGB | Max depth = 9 | 55.2 |
| RF | Max depth = 6 | 55.0 |
| SVM | C = [0.1, 1, 10, 1000] | 55.8 |
| LR | C = 1.0 | 55.7 |

equation that describes the node embedding of node $i$ at layer $k$ of the GNN is:

$$x_i^k = \sigma\left(\sum_{j\in N(i)\cup\{i\}} \frac{e_{j,i}}{\sqrt{\widehat{d_j}\,\widehat{d_j}}} x_j^{k-1}\right) \tag{1}$$

where $\hat{d_i} = 1 + \sum_{j\in \mathrm{N}(i)} e_{j,i}$, with the addition of “1” to avoid a division by 0 if a node has no neighbors, and $\sigma$ is a non-linear function such as ReLU or sigmoid. This indicates that a node's new features will be influenced by the features of the nodes of its neighbors in a way that is proportional to the strength of the connection between them, i.e. the edge weight $e_{j,i}$. In our case, this is the co-expression weight q — from StringDB. We note that since a co-expression graph is undirected, $e_{j,i} = e_{i,j}$. Dividing by $\hat{d_j}\hat{d_j}$ ensures that if node A and node B are neighbors and also have high degrees, then the influence of A on B or vice versa should be less than if they both had low degrees, as we would expect intuitively. At the first layer of the GNN, when $k = 1$, $x_i^{k-1} = x_i^0$, which is the gene expression measurement of gene $i$, a 1-D feature vector. Once node embeddings are obtained at the final layer of the GNN (layer 3 in Fig. 3), node embeddings from all layers are concatenated to form each node's final embedding that is used subsequently as input to SortPooling, as illustrated in Fig. 3, “Feature concatenation”. Before we explain how SortPooling works, we first motivate heuristically why we chose this architecture instead of performing a simple averaging of all node embeddings and using that as the final graph embedding. In our

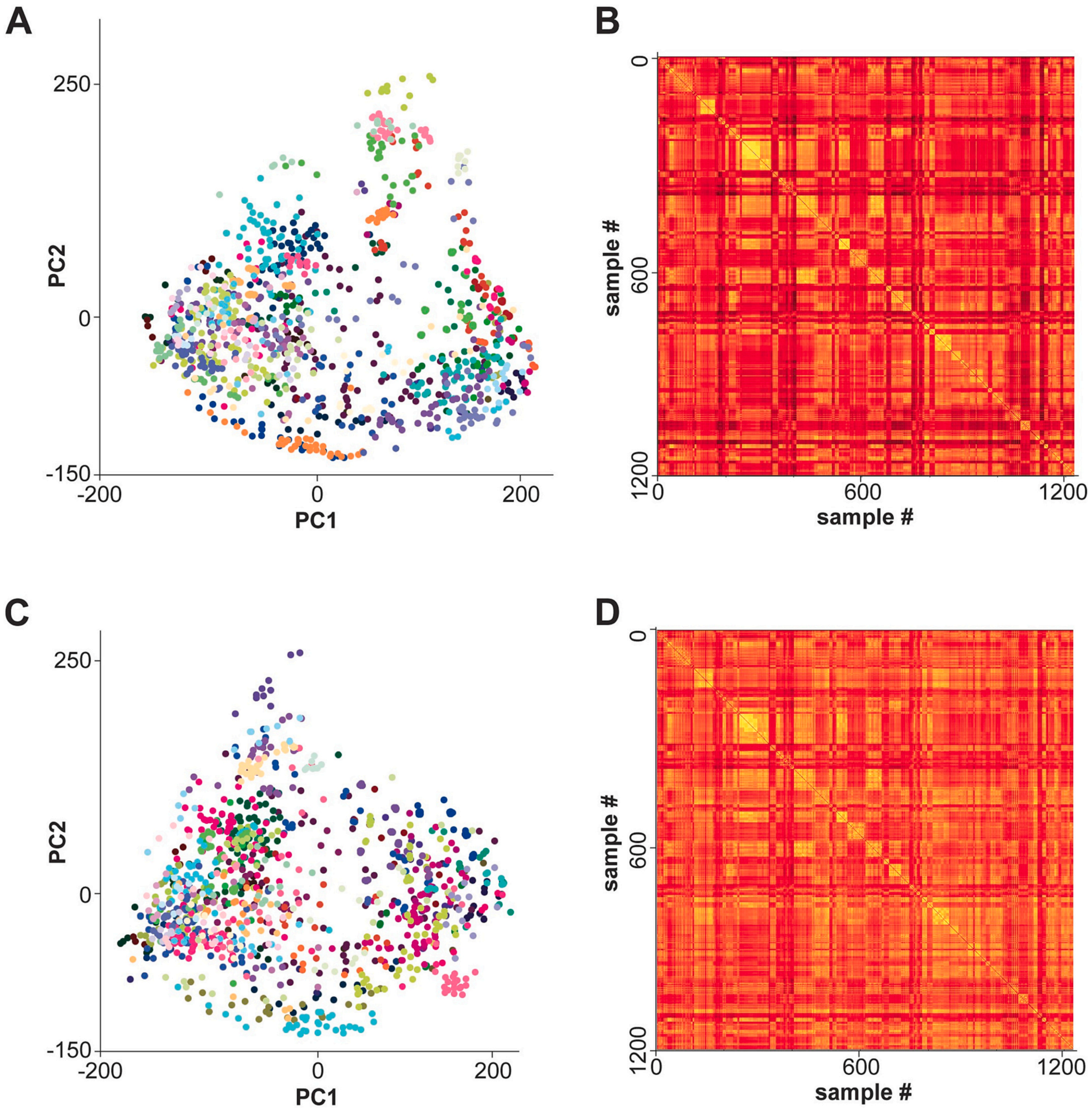


**Fig. 5.** (A) PCA plot of GeTMM normalized gene expression data, where each color corresponds to a different study. (B) Similarity matrix of GeTMM normalized gene expression data, with samples from the same study aligned. Similarity is measured as -(Euclidean distance) between samples. The brighter the color, the more similar the samples. We set the similarity of a sample with itself to be the largest distance between two samples, hence the thin dark diagonal line. (C) Same as (A), but Combat-Seq normalized before GeTMM normalized. (D) Same as (B), but Combat-Seq normalized before GeTMM. Panels C and D, especially the latter, show that batch correction using Combat-Seq is effective in reducing batch effects.

application, there are about 20,000 nodes in the input graph, the genes in the *C. elegans* genome. Thus, if, at the final layer of the GNN, we average all the resulting node embeddings, we end up with a single feature vector that must attempt to compress or summarize information about this large graph. This could lead to some loss of information. To alleviate this issue, we use SortPooling, which works as follows. Suppose the GNN has $h$ backbone feature extraction layers. At each layer, denote by $f_t$ the dimension of node embeddings at layer $t$. Then, since we concatenate node features at the "Feature concatenation" step in Fig. 3, the input to the SortPooling layer is a $n \times \sum_{t=1}^{h} f_t$ matrix, where each row represents a node, and each column is a feature dimension. The output of SortPooling is a $k \times \sum_{t=1}^{h} f_t$ matrix, where $k$ is a hyperparameter. In Fig. 3, $k$ is set to 5, the number of rows in the dashed rectangle. As the name of the layer implies, the nodes that we choose to keep are obtained by sorting. Sorting is done by considering the last feature dimension of nodes, in descending order. Thus, if node A's last feature is 5, and node B's is 10, then node B takes precedence over node A. In case of ties, the second to last feature dimension is used, and so on until the first feature

**Table 3**
Per sample, no gene coexpression bias, Combat-Seq experiment corrected, GeTMM normalized, No mix split accuracy results. A neural-based model achieves the highest cross validation accuracy.

| Model | Optimal hyperparameters | Cross validation accuracy (%) |
|---|---|---|
| XGB | Max depth = 9 | 57.4 |
| RF | Max depth = 9 | 55.9 |
| SVM | C = [0.1, 1.0, 10, 1000] | 59.4 |
| LR | C = 1000 | 60.4 |
| MLP | Learning rate = 0.0001<br>number of MLP layers = 3 dropout = 0.8<br>number of hidden dimensions = 1024 weight decay = 0.001 | 64.9 |

**Table 4**
Per sample, no gene coexpression bias, Combat-Seq experiment corrected, GeTMM normalized, No mix split accuracy results. A non-neural based model achieves the highest test set accuracy, around 10 % better than a naive model that predicts the majority class only.

| Model | Test set accuracy (%) |
|---|---|
| LR | 55.4 |
| MLP | 50.0 |

dimension. In this sense, if we set $k$ to 1000, and the graph has 20,000 nodes, the resulting 1000 genes are interpreted as the 1000 most important genes for the prediction. In this regard, the GNN in this way can learn to prioritize certain genes over others based on how informative they are.

After processing each sample through the SortPooling layer, we concatenated its age with the resulting graph-level embedding obtained after SortPooling. We then passed the result to an MLP for classification. We used BatchNorm and the ReLU activation function for all hidden layers. We trained the GNN using the Adam optimizer. Supplementary Table S2 lists the hyperparameters we used in the order listed. In the case of k = 22,113, which corresponds to keeping all genes, we did not use the SortPooling layer. For training and evaluation, we used the same protocol as described in Section 2.4.1. As in the case of the MLP, we considered the best GNN model to be the one that achieves the best cross validation accuracy across folds for a specific epoch. We used a batch size equal to the dataset size whenever possible, and used the largest possible batch size that is a power of 2 otherwise. We batch corrected the input data by experiment using Combat-Seq, and GeTMM then $\log_2$ transformed it. We implemented our model using the Pytorch Geometric library (Fey and Lenssen, 2019), a graph machine learning Python library.

The approach described here meets Criterion 2, but not Criterion 1, given the introduction of prior knowledge.

The GNN model is trained using a 4GB NVIDIA GeForce RTX 2060 GPU.

Refer to text for details on how the architecture is structured. Adapted from M. Zhang et al. (2018)

### 2.4.3. Differential expression analysis approach

For differential expression (DE) analysis, we used a workflow implemented in the R package Limma/Voom. This approach is based on statistical tests, obtained by fitting a linear model to the data and an empirical Bayes method to compute t-statistics to determine if a gene is significantly differentially expressed between conditions (Law et al., 2014; Phipson et al., 2016). Before passing our raw count data into the Limma/Voom DE pipeline, we batch corrected it using Combat-Seq, and computed associated TMM normalization factors which are passed in with the count data. To control the false positive rate, we employed the Benjamini-Hochberg method (Benjamini and Hochberg, 1995). We choose an adjusted p-value of 0.05 for our significance level, and a $\log_2$ fold change of 1.5 to determine differentially expressed genes. We note that the approach described here does not meet Criterion 2, since the analysis is done on a per-gene basis, but meets Criterion 1.

### 2.4.4. Differential expression meta-analysis approach

We obtained a list of differentially expressed genes on a per study basis, then looked for genes that were flagged as differentially expressed across multiple studies. The approach described here meets Criterion 1, but not Criterion 2.

### 2.5. Code and data availability

All code used to download, map, and analyze more than a thousand SRA gene expression samples on a single local machine is disclosed on Github. Raw- and expert-labeled datasets of *C. elegans* samples labeled as “long-lived”, “normal”, or “short-lived” are made publicly available in a Docker image which can be found on DockerHub.

## 3. Results

After data curation and pre-processing (see Fig. 1 and Methods section), we used data from 74 independent studies, totaling 1241 gene expression profiles, in our analysis. We excluded one study (see Supplementary Fig. S1), resulting in 1229 total gene expression profiles from 73 studies. Our final dataset consisted of 44.9 % normal-lived, 35.9 % long-lived, and 19.2 % short-lived samples. As an additional quality check, we assessed project-specific metadata, including the used sequencing equipment and strategy used (see Supplementary Fig. S2A and B) as well as the per sample coverage and sequencing depth (number of reads per sample) (see Supplementary Fig. S2C and D). We found that most samples were sequenced using Illumina HighSeq or Illumina NextSeq equipment (see Supplementary Fig. S2A) with slightly more studies using single-end (56 %) versus paired-end (44 %) sequencing strategies (see Supplementary Fig. S2B). Across samples, 84.7 % showed an at least 10fold coverage (Median: 25.9) (see Supplementary Fig. S2C) and 86.9 % had a sample depth of at least 10 million reads (Median: 27.2 million) (see Supplementary Fig. S2D), suggesting that the dataset consisted predominantly of high-quality samples (Liu et al., 2013). Next, we performed 2D (Fig. 4) and 3D (see Supplementary Fig. S3D) Principal Component Analyses (PCA). We found that, following Combat-Seq correction and GeTMM normalization, samples clustered by age.

We also observed the same clustering comparing long-lived only, normal-lived only and short-lived only datasets (see Supplementary Fig. S3A–C).

### 3.1. Inter-lab variation is a major impediment to the application of ML models

To test different modeling approaches, we first evaluated a per-sample, non-gene interaction biased approach using input not subjected to batch Combat-Seq normalization, in the “Mix” and “No mix” split settings. For this, we trained SVM with linear kernel, LR, RF, and XGB models. As shown in Tables 1 and 2, we observed a significant difference in cross validation accuracy between models trained in “Mix” vs. “No mix” split settings.

While we expected a decrease in accuracy in the “No Mix” split setting compared to the “Mix” split setting, the drop was significantly higher than expected. The best performing model in the Mix split setting achieved an accuracy of 87.9 %, compared to an accuracy of 55.8 % in the No mix split setting. This indicates a strong batch effect in the data since the models perform well when trained and validated on samples from the same study, but not nearly as well when the studies observed during training are not observed in the validation set. We then generated a PCA plot in which data points were colored based on experiment (Fig. 5A) demonstrating that samples from the same experiment

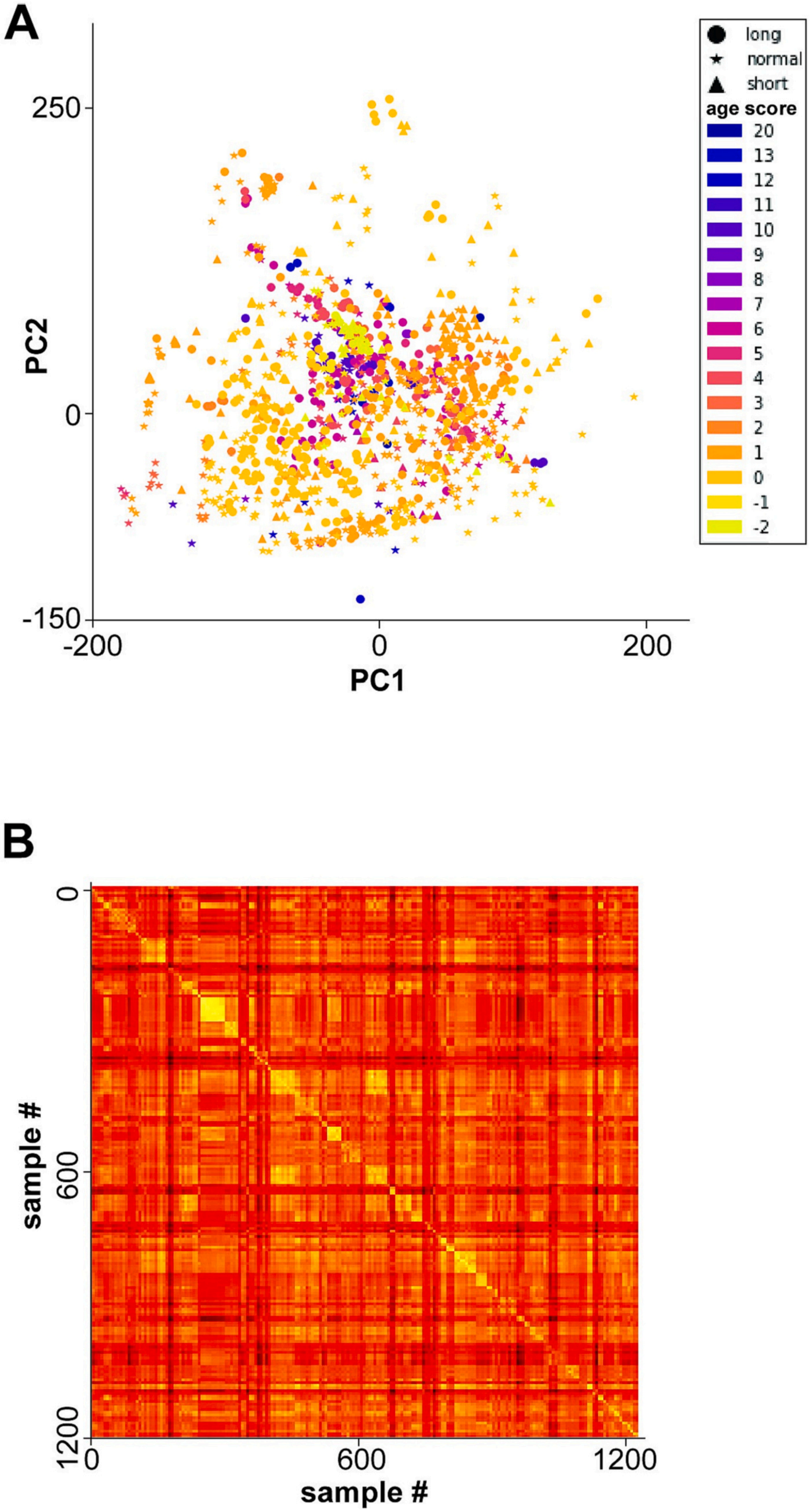


**Fig. 6.** (A) PCA plot of Combat-Seq corrected by age, GeTMM normalized gene expression data, where each color corresponds to a different age. Samples cluster by age despite batch correction, which may indicate that age may act as a proxy for the study a sample originates from (B) Similarity matrix of Combat-Seq age corrected, GeTMM normalized gene expression data with samples from the same study lined up next to each other. Similarity is measured as -(Euclidean distance) between samples. The brighter the color, the more similar the samples. We set the similarity of a sample with itself to be the largest distance between two samples, hence the thin dark diagonal line. Compared to the similarity matrix in Fig. 5D, we observe a slight reduction in similarity when correcting by age.

**Table 5**
Per sample, no gene coexpression bias, GeTMM normalized, Combat-Seq corrected by age, No mix split accuracy results. Batch correction by age using Combat-Seq does not improve performance.

| Model | Optimal hyperparameters | Cross validation accuracy (%) |
|---|---|---|
| RF | Max depth = 9 | 54.4 |
| SVM | C = [0.1, 1.0, 10, 1000] | 55.6 |
| LR | C = 1000 | 54.5 |

**Table 6**
Per sample, no gene coexpression bias, GeTMM normalized, Combat-Seq corrected by experiment and age, No mix split accuracy results. Batch correction by both experiment and age using Combat-Seq does not improve performance.

| Model | Optimal hyperparameters | Cross validation accuracy (%) |
|---|---|---|
| RF | Max depth = 3 | 55.3 |
| SVM | C = [0.1, 1.0, 10, 1000] | 59.5 |
| LR | C = 0.1 | 57.4 |

**Table 7**
Per sample, no gene coexpression bias, GeTMM normalized, SAUCIE corrected by experiment, No mix split accuracy results. Batch correction by experiment using SAUCIE significantly decreases performance.

| Model | Optimal hyperparameters | Cross validation accuracy (%) |
|---|---|---|
| RF | Max depth = 3 | 51.4 |
| SVM | C = 10 | 50.2 |
| LR | C = 1.0 | 54.6 |

**Table 8**
Per sample, gene co-expression graph bias model, No mix split accuracy results. A neural graph-based approach does not improve performance compared to a standard neural approach.

| Model | Optimal hyperparameter | Cross validation accuracy (%) |
|---|---|---|
| GNN | Learning rate = 0.0001<br>number of MLP layers = 3<br>concatenate input graph = True<br>k (number of nodes after pooling) = 22,113<br>number of backbone layers = 2 | 63.4 |

clustered together. To further test this batch effect, we calculated a similarity matrix using reverse scale Euclidean distance (Fig. 5B), which confirmed that samples cluster based on origin lab rather than genotype or lifespan. Together, these results indicate that lab-specific experimental features, such as sample preparation details, may be confounding factors that complicate inter-lab comparability of *C. elegans* bulk RNA-Seq datasets using machine learning. Since sequencing strategies, sample depth, and sample coverage are rather uniform and of high quality in our dataset (see Supplementary Fig. S2A–D) it appears unlikely that these features significantly contribute to the observed clustering. Thus, inherent lab-specific differences in worm culturing methods, environmental factors (temperature, humidity), and strain adaptations are likely key drivers of the observed inter-lab variability in RNA-seq datasets, which is consistent with previous studies investigating inter-lab comparability of wild-type *C. elegans* lifespan data (Urban et al., 2021).

As shown in Fig. 5C and D, batch correction using Combat-Seq reduced sample clustering by experiment but was not sufficient to eliminate this problem. Using age-subsetted data corrected by experiment (see Supplementary Fig. S4A and B), we again observed that samples clustered by experiment, despite the correction. We then ran our algorithms again on the Combat-Seq experiment corrected, GeTMM data in the No mix split setting only.

We observed a noticeable increase in model performance (accuracy) ranging from 0.9 % (RF-based model) to 4.7 % (LR-based model) using Combat-Seq (Table 3). We note that the best performing nonneural based model achieved a 15.5 % increase in accuracy compared to a naive model that only predicts the majority class, (normal-lived), which makes up 44.9 % of the dataset. We obtained a further 4.3 % increase in cross-validation accuracy after training a neural-based model, the MLP. While the MLP was the best performing model when evaluated using cross-validation, the LR model performed significantly better on the test set (Table 4). We thus extracted the top 10 longevity predictors according to our LR model (see Supplementary File S2). Interestingly, three of the top 10 genes (*cyp14A2*, *cyp-14A4*, *cyp—35B1*) encode for cytochrome P450 family proteins, which are iron-binding proteins that catalyze monooxygenase reactions on a wide range of endobiotic and xenobiotic substrates (Larigot et al., 2022).

Next, we revisited the impact of sample age at the time of sequencing as a potential confounding variable. Our dataset contained RNA-Seq data from animals collected during the larval stages up to 20 days of adulthood, an approximate 23 day time spread. We thus hypothesized that a batch-correction by age might result in improved model performance. To test this hypothesis, we batch-corrected samples by age, rather than by experiment, using Combat-Seq. The resulting PCA plot and similarity matrix are shown in Fig. 6A–B, respectively. We observed a decrease in accuracy compared to the experiment-corrected data (Table 5), suggesting that the age at which *C. elegans* samples were processed may not be a major contributor to the observed batch effect. While Combat-Seq was not developed to handle correcting data twice by two batch variables, we attempted to correct by both experiment and age, which did not yield any benefits (Table 6). Using SAUCIE instead of Combat-Seq for batch correction decreased model performance notably (Table 7).

In an attempt to circumvent the observed batch effect problem, we next trained models using preselected, normalized and Combat-Seq-corrected data subsets and tested model performance. First, we used data subsets filtered to only contain information of genes known to be linked to aging in GeneAge (Supplementary Table S3A) or genes annotated with aging-related GO terms (Supplementary Table S3B). Compared to the reference model (Table 2), introducing either of these biases decreased performance across models. Next, we trained models only using data obtained from studies analyzing adult (Supplementary Table S3C) or larval-stage worms (Supplementary Table S3D), which similarly decreased accuracy. Models trained on data from only L4 and younger worms performed equally to, or better than models trained on adult data only. It is plausible that the wider spread of ages in the adult group may contribute to increased noise in this subsetted dataset. Finally, we built a model only including genes for which the CPM was 1 in at least 236 samples (Supplementary Table S3E). While performance improved across models, the observed gains were small, with a maximum performance improvement of 0.6 %. Together, our results show that using a per-sample no gene coexpression inductive bias approach can be used to build classifiers to predict aging-associated genes from RNA-Seq datasets, but more sophisticated batch correction techniques are needed to allow such an approach to reach its full potential.

### *3.2. Per sample, with coexpression inductive bias results: a graph-based approach does not yield performance gains*

We next tested models injected with prior knowledge based on gene co-expression graphs. To build a per-sample model that is biased with prior knowledge (i.e bias) regarding gene co-expression, we used a graph neural network (GNN) (Fig. 3). As input to the GNN, we used Combat-Seq experiment corrected, GeTMM normalized data, which we previously found to perform best. The GNN did not increase performance compared to the MLP-based model (Tables 8 and 3), suggesting

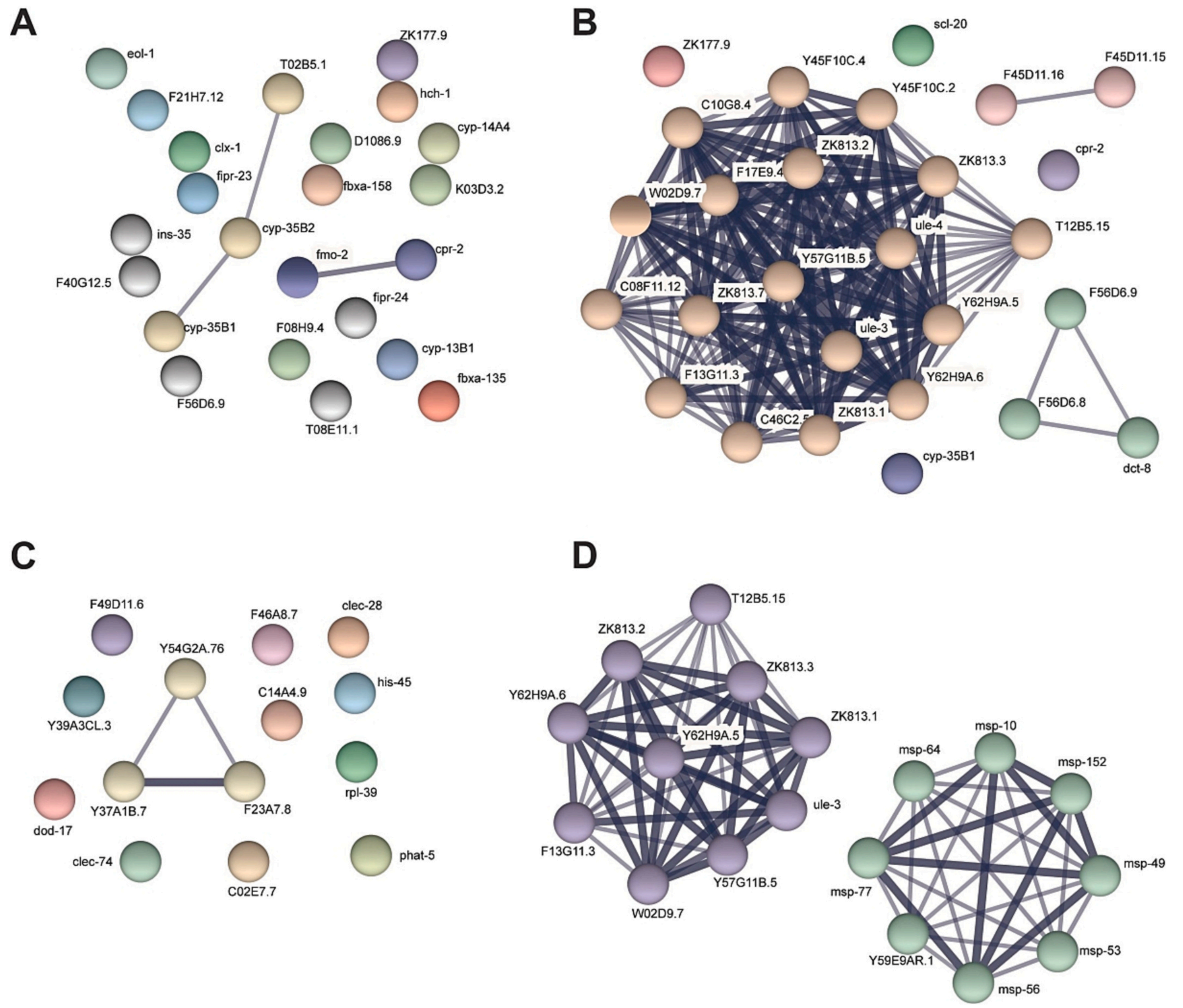


**Fig. 7.** STRING network analysis using results from combined approach as input. Edge thickness represents the strength of data support for an interaction between two connected genes. (A) upregulated genes in long-lived vs normal-lived. (B) up-regulated genes in long-lived vs short-lived. (C) down-regulated genes in long-lived vs short-lived. (D) down-regulated genes in short-lived vs normal-lived.

that either the graph-based bias is not as informative as anticipated, or that further innovations in the GNN architecture may be needed to properly exploit the information in the gene co-expression graph (i.e an architecture that is specifically suited for gene co-expression networks coupled with gene expression information may need to be developed).

### 3.3. Differential expression analysis identifies potential regulators of aging

As an alternative to machine learning approaches, we performed a traditional differential expression analysis in which we pooled all samples from all studies before conducting the analysis. We used the output (See Supplementary File S3) as the basis for a STRING-based network analysis (Fig. 7A–D and Supplementary Fig. S5). The edges were created using the "confidence" setting in StringDB, and the thickness of the edges represents the strength of data support for an interaction between the genes in question. All possible interaction sources that StringDB offers were included; these are "Text mining", "Experiments","Databases","Co-expression", "Neighborhood","Gene Fusion", and "Co-occurrence." Comparing genes upregulated in long-lived versus short-lived worms (Fig. 7B), we identified two interaction hubs. Hub 1 consists of genes *fmo-2* and *cpr-2*. *Fmo-2* is a flavin-containing monooxygenase known to extend lifespan and stress resistance in *C. elegans* when overexpressed (Leiser et al., 2015b). *Cpr-2* is the *C. elegans* ortholog of Cathepsin B, a lysosomal cysteine protease linked to several autoimmune and neurodegenerative diseases in humans (Drobny et al., 2022; Hook et al., 2020).

Whether or not *cpr-2* contributes to *fmo-2*-dependent lifespan extension is unknown. Hub 2 consists of *cyp—35B1*, *cyp-35B2* – two cytochrome P450 proteins – and *T02B5.1*, a predicted membrane-embedded carboxyl esterase. In contrast, only a single gene of unknown function, *F47G4.14*, was down-regulated in long-lived versus normal-lived datasets. To the best of our knowledge, none of the genes in Hub 2 have been studied in the context of lifespan regulation. Interestingly, genes significantly down-regulated in short-lived versus normal-lived datasets (Fig. 7D) clustered into two distinct hubs. Hub 3 (right in Fig. 7D) is comprised of seven major sperm protein (*msp*)-type genes

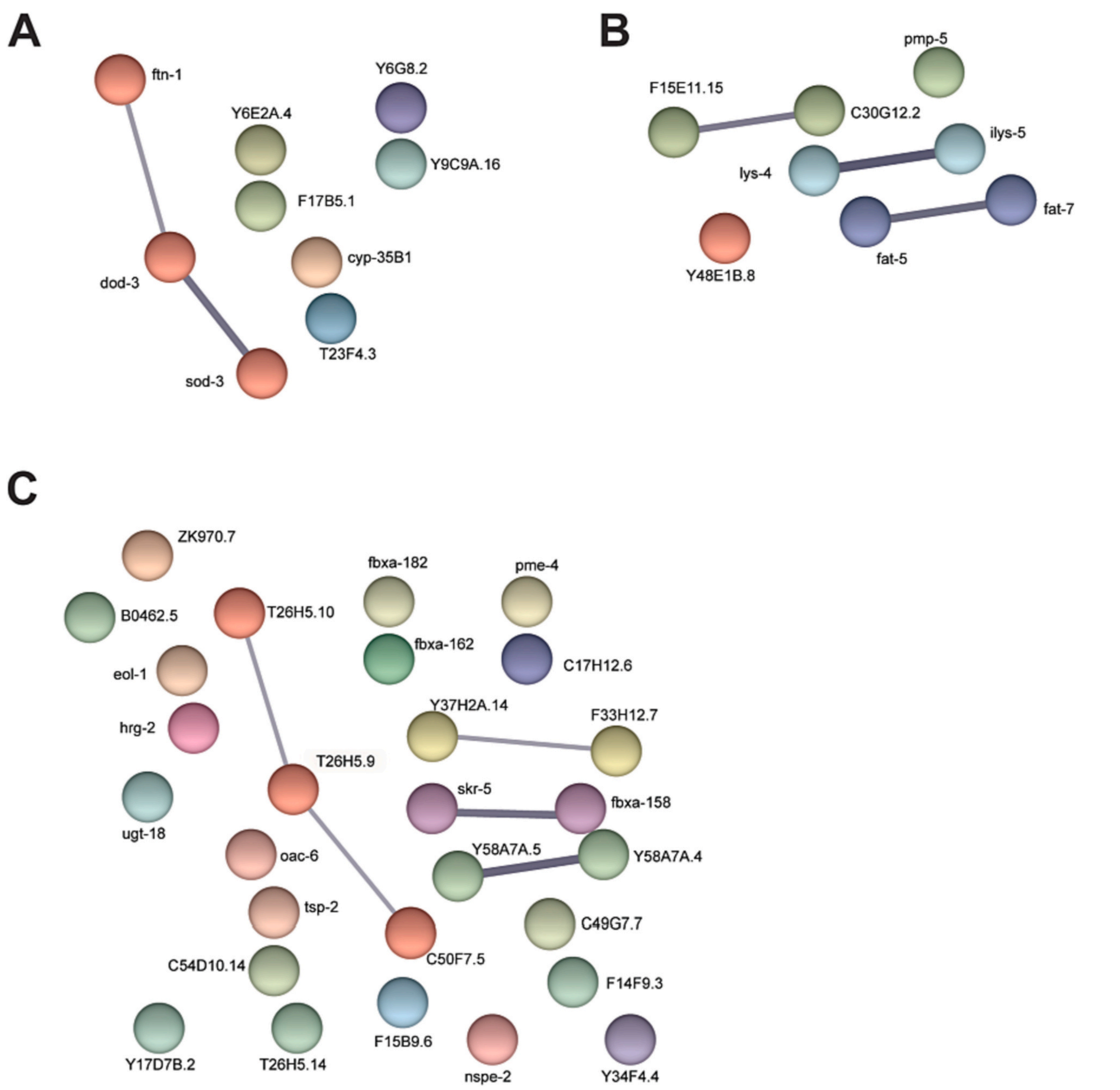


**Fig. 8.** STRING network analysis using results from meta-analysis approach as input where genes are up/down-regulated in at least 25 % of studies that compared the two conditions in question. (A) up-regulated genes in long-lived vs normal-lived (B) down-regulated genes in short-lived vs normallived. (C) up-regulated genes in short-lived vs normal-lived.

and *Y59E9AR.1*, all of which are expressed in the male gonad. Hub 4 (left in Fig. 7D) mainly consists of *ZK813* and uterine lumin expressed (*ule*) family genes, which are associated with the egg chondrion and the spermatica. Functionally, both Hub 3 and 4 are strongly linked to fertility and reproduction. Whether these results represent a causal link between down-regulation of fertility-associated genes and lifespan shortening or merely are the consequence of a decline in reproduction in otherwise challenged animals remains to be tested. A similar caveat must be considered when interpreting the network consisting of genes up-regulated in long-lived versus short-lived datasets (Fig. 7B), which shows partial overlap with Hub 4 and centers around genes critical for reproduction and gonad function.

### 3.4. Differential expression meta-analysis indicates a role for integral membrane proteins in lifespan regulation

We next performed a per-gene meta-analysis, identifying differentially expressed genes on a per-study basis, and then mining these results for genes that were differentially expressed across multiple studies (See Supplementary File S4). A STRING-based network analysis of these hits (Fig. 8A–C, Supplementary Figs. S6 and S7) revealed an interaction hub consisting of *dod-3*, *sod-3*, and *ftn-1* that were significantly upregulated in at least 25 % of our datasets in long-lived vs. normal-lived datasets (Fig. 8A). *Sod-3* is a superoxide dismutase involved in removing superoxide radicals, *dod-3* is a predicted membrane protein, and *ftn-1* is a Ferritin-type protein. All three of these genes are regulated by *daf-16* and have previously been implicated in lifespan extension in long-lived mitochondrial mutants (Senchuk et al., 2018). Comparing long-lived vs short-lived datasets, we identified a gene hub containing *irg-1* and *irg-2* that was significantly down-regulated in long-lived datasets (Supplementary Fig. S6). Both *irg-1* and *irg-2* are involved in innate immune responses, processes known to be linked to lifespan regulation (Jaiswal et al., 2021; Michelucci et al., 2013). Using a more stringent approach (up or down regulation in at least 50 % of relevant datasets), we identified a single gene as significantly upregulated (*dod-3*), and a single gene as significantly downregulated (*H02F09.3*) in the long-lived datasets. Both *dod-3* and *H02F09.3* are predicted integral membrane proteins of unknown function, and their roles in lifespan regulation have yet to be determined.

## 4. Discussion

Recent years have seen seminal advances in multiple areas of predictive bioinformatics, perhaps most notably in the context of protein structure prediction (AlphaFold, for example) (Jumper et al., 2021). In this study, we developed both traditional and machine learning models optimized to predict genes linked to *C. elegans* lifespan regulation. In contrast to most published work, our primary goal was to use unbiased per-sample data vectors as input. Previous meta-analyses by De Magalhães et al. (2009) as well as Palmer et al. (2021) used *C. elegans* gene expression data, but 1) relied on traditional differential expression (per-gene) analysis and 2) did not focus on predicting pro- versus anti-longevity genes, but rather on determining which genes tend to be over or under-expressed with age. Nevertheless, these approaches generally resulted in useful models and yielded interesting results. For instance, Kerepesi et al. (2018) found that their XGBoost model predicts the *sir-2* gene – whose effect on longevity is still debated – to be aging related.

Our approach to classify samples instead of genes turned out to be a challenging task, since typical experiments consist of a limited number of samples (1−100), each quantifying the expression of around 20,000 genes. While this approach suffers from high dimensionality, it has the potential to discover novel gene modules and/or pathways contributing to metazoan aging. The models we developed in this study using unbiased data as input did not perform as well as anticipated. We attribute this primarily to per-lab/per-study variance, as our algorithms performed well when training and validation sets contained samples from the same study, but performed significantly worse when this was not the case. Ultimately, neither normalization nor batch-correction methods were successful in rectifying this decline in performance, suggesting that more advanced batch correction techniques may be needed to disentangle longevity signals in bulk RNA-Seq gene expression data. One approach may be to follow a standardized sample processing routine when preparing samples for RNA-Seq experiments (Urban et al., 2021), but in practice, implementation is often confounded by study, likely due to experimenterspecific handling and culturing methods.

A relatively simple step to improve the value of published RNA-Seq datasets for inter-lab comparisons would be to provide more detailed meta-information for each disclosed RNA-Seq dataset. This could be implemented on multiple levels, including mandatory metadata deposition when uploading sequencing data to public servers or with publications. For example, we found that the age of the worms at collection for RNA-Seq sample prep was not always disclosed. This is particularly problematic for a short-lived organism like *C. elegans* for which we expect a dynamic gene expression landscape over a short period of time. Another inherit limitation of our approach was that samples were classified into three user-defined categories with explicit boundaries (“short-lived”, “normal-lived”, “long-lived”), though a lack of a standardized definition of aged cohorts limits the robustness of machine learning approaches. An alternative approach for future studies may be to model the problem as a regression task, where the prediction is the number of days that the organism remains alive.

Supplementary data to this article can be found online at https://doi.org/10.1016/j.exger.2023.112107.

## CRediT authorship contribution statement

**Mohamad Bairakdar**: Conceptualization, Methodology, Software, Data curation, Visualization, Original draft preparation, Investigation, Editing. **Ambuj Tewari**: Conceptualization, Supervision, Reviewing and Editing. **Matthias Truttmann**: Conceptualization, Data curation, Visualization, Original draft preparation, Investigation, Supervision, Reviewing and Editing.

## Data availability

A link to download the code and data used in this study is provided in the manuscript

## Acknowledgments

We would like to thank Joao Pedro De Magalhães, Johann Gagnon-Bartsch, Matthias Fey, Bennet Fauber, and Gustavo Magdaleno for their helpful discussions and Nicholas Urban, Kate Van Pelt, and William Giblin for proofreading. This project was supported by grant R35 GM142561 to MCT.

## References

Amodio, M., Dijk, D.V., Srinivasan, K., Chen, W.S., Mohsen, H., Moon, K.R., Campbell, A., Zhao, Y., Wang, X., Venkataswamy, M., et al., 2017. Exploring Single-Cell Data With Deep Multitasking Neural Networks. https://doi.org/10.1101/237065.

Benjamini, Y., Hochberg, Y., 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. J. R. Stat. Soc. Ser. B Methodol. 57 (1), 289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x.

Bertin, P., Hashir, M., Weiss, M., Frappier, V., Perkins, T.J., Boucher, G., Cohen, J.P., 2020. Analysis of Gene Interaction Graphs As Prior Knowledge for Machine Learning Models.

Bray, N.L., Pimentel, H., Melsted, P., Pachter, L., 2016. Near-optimal probabilistic rna-seq quantification. Nat. Biotechnol. 34 (5), 525–527. https://doi.org/10.1038/nbt.3519.

Chen, T., Guestrin, C., 2016. XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. https://doi.org/10.1145/2939672.2939785.

Chereda, H., Bleckmann, A., Menck, K., Perera-Bel, J., Stegmaier, P., Auer, F., Kramer, F., Leha, A., Beißbarth, T., 2020. Explaining Decisions of Graph Convolutional Neural Networks: Patient-Specific Molecular Subnetworks Responsible for Metastasis Prediction in Breast Cancer. https://doi.org/10.1101/2020.08.05.238519.

De Magalhães, J.P., Curado, J., Church, G.M., 2009. Meta-analysis of age-related gene expression profiles identifies common signatures of aging. Bioinformatics 25 (7), 875–881. https://doi.org/10.1093/bioinformatics/btp073.

Drobny, A., Prieto Huarcaya, S., Dobert, J., Kluge, A., Bunk, J., Schlothauer, T., Zunke, F., 2022. The role of lysosomal cathepsins in neurodegeneration: mechanistic insights, diagnostic potential and therapeutic approaches. Biochim. Biophys. Acta Mol. Cell Res. 1869 (7), 119243 https://doi.org/10.1016/j.bbamcr.2022.119243.

Dutil, F., Cohen, J.P., Weiss, M., Derevyanko, G., Bengio, Y., 2018. Towards Gene Expression Convolutions Using Gene Interaction Graphs.

Fabris, F., Freitas, A.A., 2016. New kegg pathway-based interpretable features for classifying ageing-related mouse proteins. Bioinformatics 32 (19), 2988–2995. https://doi.org/10.1093/bioinformatics/btw363.

Fabris, F., De Magalhães, J.P., Freitas, A.A., 2017. A review of supervised machine learning applied to ageing research. Biogerontology 18 (2), 171–188. https://doi.org/10.1007/s10522-017-9683-y.

Fabris, F., Palmer, D., Salama, K.M., De Magalhães, J.P., Freitas, A.A., 2019. Using deep learning to associate human genes with age-related diseases. Bioinformatics 36 (7), 2202–2208. https://doi.org/10.1093/bioinformatics/btz887.

Fang, Y., Wang, X., Michaelis, E.K., Fang, J., 2013. Classifying aging genes into dna repair or nondna repair-related categories. In: Intelligent Computing Theories and Technology Lecture Notes in Computer Science, pp. 20–29. https://doi.org/10.1007/978-3-642-39482-9 3.

Fernandes, M., Wan, C., Tacutu, R., Barardo, D., Rajput, A., Wang, J., Thoppil, H., Thornton, D., Yang, C., Freitas, A., et al., 2016. Systematic analysis of the gerontome reveals links between aging and age-related diseases. Hum. Mol. Genet. https://doi.org/10.1093/hmg/ddw307.

Fey, M., Lenssen, J.E., 2019. Fast Graph Representation Learning with PyTorch Geometric (ICLR Workshop on Representation Learning on Graphs and Manifolds).

Freitas, A.A., Vasieva, O., De Magalhães, J., 2011. A data mining approach for classifying dna repair genes into ageing-related or non-ageing-related. BMC Genomics 12 (1). https://doi.org/10.1186/1471-2164-12-27.

Hillier, L.W., Coulson, A., Murray, J.I., Bao, Z., Sulston, J.E., Waterston, R.H., 2005. Genomics in c. elegans: so many genes, such a little worm. Genome Res. 15 (12), 1651–1660. https://doi.org/10.1101/gr.3729105.

Hook, V., Yoon, M., Mosier, C., Ito, G., Podvin, S., Head, B.P., Rissman, R., O'Donoghue, A.J., Hook, G., 2020. Cathepsin b in neurodegeneration of alzheimer's disease, traumatic brain injury, and related brain disorders. Biochim. Biophys. Acta Proteins Proteom. 1868 (8), 140428 https://doi.org/10.1016/j.bbapap.2020.140428.

Huang, T., Zhang, J., Xu, Z.-P., Hu, L.-L., Chen, L., Shao, J.-L., Zhang, L., Kong, X.-Y., Cai, Y.-D., Chou, K.-C., et al., 2012. Deciphering the effects of gene deletion on yeast longevity using network and machine learning approaches. Biochimie 94 (4), 1017–1025. https://doi.org/10.1016/j.biochi.2011.12.024.

Jaiswal, A.K., Yadav, J., Makhija, S., Mazumder, S., Mitra, A.K., Suryawanshi, A., Sandey, M., Mishra, A., 2021. Irg1/itaconate metabolic pathway is a crucial determinant of dendritic cells immune-priming function and contributes to resolute allergen-induced airway inflammation. Mucosal Immunol. 15 (2), 301–313. https://doi.org/10.1038/s41385-021-00462-y.

Jiang, H., Ching, W.-K., 2011. Classifying dna repair genes by kernel-based support vector machines. Bioinformation 7 (5), 257–263. https://doi.org/10.6026/97320630007257.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al., 2021. Highly accurate protein structure prediction with alphafold. Nature 596 (7873), 583–589. https://doi.org/10.1038/s41586-021-03819-2.

Kaeberlein, T.L., Smith, E.D., Tsuchiya, M., Welton, K.L., Thomas, J.H., Fields, S., Kennedy, B.K., Kaeberlein, M., 2006. Lifespan extension in *caenorhabditis elegans* by complete removal of food. Aging Cell 5 (6), 487–494. https://doi.org/10.1111/j.1474-9726.2006.00238.x.

Kenyon, C., Chang, J., Gensch, E., Rudner, A., Tabtiang, R., 1993. A c. elegans mutant that lives twice as long as wild type. Nature 366 (6454), 461–464. https://doi.org/10.1038/366461a0.

Kerepesi, C., Daróczy, B., Sturm, A., Vellai, T., Benczúr, A., 2018. Prediction and characterization´ of human ageing-related proteins by using machine learning. Sci. Rep. 8 (1) https://doi.org/10.1038/s41598-018-22240-w.

Kingma, D.P., Ba, J., 2017. Adam: A Method for Stochastic Optimization.

Kipf, T.N., Welling, M., 2017. Semi-Supervised Classification with Graph Convolutional Networks.

Larigot, L., Mansuy, D., Borowski, I., Coumoul, X., Dairou, J., 2022. Cytochromes p450 of caenorhabditis elegans: implication in biological functions and metabolism of xenobiotics. Biomolecules 12 (3), 342. https://doi.org/10.3390/biom12030342.

Law, C.W., Chen, Y., Shi, W., Smyth, G.K., 2014. Voom: precision weights unlock linear model analysis tools for rna-seq read counts. Genome Biol. 15 (2) https://doi.org/10.1186/gb-2014-15-2-r29.

Leiser, S.F., Miller, H., Rossner, R., Fletcher, M., Leonard, A., Primitivo, M., Rintala, N., Ramos, F.J., Miller, D.L., Kaeberlein, M., et al., 2015a. Cell nonautonomous activation of flavincontaining monooxygenase promotes longevity and health span. Science 350 (6266), 1375–1378. https://doi.org/10.1126/science.aac9257.

Leiser, S.F., Miller, H., Rossner, R., Fletcher, M., Leonard, A., Primitivo, M., Rintala, N., Ramos, F.J., Miller, D.L., Kaeberlein, M., et al., 2015b. Cell nonautonomous activation of flavincontaining monooxygenase promotes longevity and health span. Science 350 (6266), 1375–1378. https://doi.org/10.1126/science.aac9257.

Li, Y., Dong, M., Guo, Z., 2010. Systematic analysis and prediction of longevity genes in caenorhabditis elegans. Mech. Ageing Dev. 131 (11−12), 700–709. https://doi.org/10.1016/j.mad.2010.10.001.

Lin, K., Hsin, H., Libina, N., Kenyon, C., 2001. Regulation of the caenorhabditis elegans longevity protein daf-16 by insulin/igf-1 and germline signaling. Nat. Genet. 28 (2), 139–145. https://doi.org/10.1038/88850.

Liu, Y., Zhou, J., White, K.P., 2013. Rna-seq differential expression studies: more sequence or more replication? Bioinformatics 30 (3), 301–304. https://doi.org/10.1093/bioinformatics/btt688.

Liu, G.-H., Bao, Y., Qu, J., Zhang, W., Zhang, T., Kang, W., Yang, F., Ji, Q., Jiang, X., Ma, Y., et al., 2020. Aging atlas: a multi-omics database for aging biology. Nucleic Acids Res. 49 (D1) https://doi.org/10.1093/nar/gkaa894.

Michelucci, A., Cordes, T., Ghelfi, J., Pailot, A., Reiling, N., Goldmann, O., Binz, T., Wegner, A., Tallam, A., Rausell, A., et al., 2013. Immune-responsive gene 1 protein links metabolism to immunity by catalyzing itaconic acid production. Proc. Natl. Acad. Sci. 110 (19), 7820–7825. https://doi.org/10.1073/pnas.1218599110.

Palmer, D., Fabris, F., Doherty, A., Freitas, A.A., de Magalhães, J. P., 2021. Ageing transcriptome meta-analysis reveals similarities and differences between key mammalian tissues. Aging 13 (3), 3313–3341. https://doi.org/10.18632/aging.202648.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic Differentiation in Pytorch.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2018. Scikit-Learn: Machiné Learning in Python.

Phipson, B., Lee, S., Majewski, I.J., Alexander, W.S., Smyth, G.K., 2016. Robust hyperparameter estimation protects against hypervariable genes and improves power to detect differential expression. Ann. Appl. Stat. 10 (2) https://doi.org/10.1214/16-aoas920.

Ramirez, R., Chiu, Y.-C., Hererra, A., Mostavi, M., Ramirez, J., Chen, Y., Huang, Y., Jin, Y.-F., 2020. Classification of cancer types using graph convolutional neural networks. Front. Phys. 8 https://doi.org/10.3389/fphy.2020.00203.

Robinson, M.D., Oshlack, A., 2010. A scaling normalization method for differential expression analysis of rna-seq data. Genome Biol. 11 (3) https://doi.org/10.1186/gb-2010-11-3-r25.

Senchuk, M.M., Dues, D.J., Schaar, C.E., Johnson, B.K., Madaj, Z.B., Bowman, M.J., Winn, M.E., Van Raamsdonk, J.M., 2018. Activation of daf-16/foxo by reactive oxygen species contributes to longevity in long-lived mitochondrial mutants in caenorhabditis elegans. PLoS Genet. 14 (3) https://doi.org/10.1371/journal.pgen.1007268.

Shokhirev, M.N., Johnson, A.A., 2022. An integrative machine-learning meta-analysis of highthroughput omics data identifies age-specific hallmarks of alzheimer's disease. Ageing Res. Rev. 81, 101721 https://doi.org/10.1016/j.arr.2022.101721.

Smid, M., Braak, R.R.J.C.V.D., Werken, H.J.G.V.D., Riet, J.V., Galen, A.V., Weerd, V.D., Vlugt-Daane, M.V.D., Bril, S.I., Lalmahomed, Z.S., Kloosterman, W.P., et al., 2018. Gene length corrected trimmed mean of m-values (getmm) processing of rna-seq data performs similarly in intersample analyses while improving intrasample comparisons. BMC Bioinformatics 19 (1). https://doi.org/10.1186/s12859-018-2246-7.

Soneson, C., Love, M.I., Robinson, M.D., 2015. Differential analyses for rna-seq: transcript-level estimates improve gene-level inferences. F1000Res. 4, 1521. https://doi.org/10.12688/f1000research.7563.1.

Szklarczyk, D., Gable, A.L., Lyon, D., Junge, A., Wyder, S., Huerta-Cepas, J., Simonovic, M., Doncheva, N.T., Morris, J.H., Bork, P., et al., 2018. String v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. Nucleic Acids Res. 47 (D1) https://doi.org/10.1093/nar/gky1131.

Uno, M., Nishida, E., 2016. Lifespan-regulating genes in c. elegans. npj Aging Mech. Dis. 2 (1) https://doi.org/10.1038/npjamd.2016.10.

Urban, N.D., Cavataio, J.P., Berry, Y., Vang, B., Maddali, A., Sukpraphrute, R.J., Schnell, S., Truttmann, M.C., 2021. Explaining inter-lab variance in c. elegans n2 lifespan: making a case for standardized reporting to enhance reproducibility. Exp. Gerontol. 156, 111622 https://doi.org/10.1016/j.exger.2021.111622.

Wan, C., Freitas, A., 2013. Prediction of the pro-longevity or anti-longevity effect of caenorhabditis elegans genes based on bayesian classification methods. In: 2013 IEEE International Conference on Bioinformatics and Biomedicine. https://doi.org/10.1109/bibm.2013.6732521.

Wan, C., Freitas, A.A., De Magalhães, J.P., 2015. Predicting the pro-longevity or anti-longevity effect of model organism genes with new hierarchical feature selection methods. IEEE/ACM Trans. Comp. Biol. Bioinform. 12 (2), 262–275. https://doi.org/10.1109/tcbb.2014.2355218.

Ying, R., Bourgeois, D., You, J., Zitnik, M., Leskovec, J., 2019. Gnnexplainer: Generating Explanations for Graph Neural Networks.

Zhang, M., Cui, Z., Neumann, M., Chen, Y., 2018. An end-to-end deep learning architecture for graph classification. AAAI 4438–4445.

Zhang, Y., Parmigiani, G., Johnson, W.E., 2020. Combat-Seq: Batch Effect Adjustment for Rna-Seq Count Data. https://doi.org/10.1101/2020.01.13.904730.