BMC Bioinformatics

RESEARCH

Open Access

Transfer learning models for bacterial strain dissemination biomarkers using weighted non-parallel proximal support vector machines

Ugochukwu O. Ugwu[1,2,3*], Richard A. Slayden[4] and Michael Kirby[1,5]

*Correspondence:
Ugochukwu O. Ugwu
uouresearch@gmail.com
[1]Pattern Analysis Laboratory, Colorado State University, Fort Collins, CO 80523, USA
[2]Tufts Advanced Microscopic Imaging Center, Tufts University, Medford, MA 02155, USA
[3]Research Computing Center, University of Chicago, Chicago, IL 60637, USA
[4]Department of Microbiology, Immunology and Pathology, Colorado State University, Fort Collins, CO 80523, USA
[5]Department of Mathematics, Colorado State University, Fort Collins, CO 80523, USA

Abstract

Background Integrating genomic datasets from homogenous or disparate sources to identify genes that are commonly or uniquely expressed remains a largely underexplored area. Such integrative analysis can reveal biologically relevant genes that are common or exclusive across datasets or within specific conditions or cohorts. Identifying these gene expression profiles and employing them to classify disease status can aid in the development of vaccines, diagnostics and targeted therapeutics with efficacy against difficult-to-treat medically important pathogens and cancer.

Results This work develops new methodologies to integrate transcriptomic patterns from the lungs and spleen tissues infected by Francisella tularensis – Schu4 and Live Vaccine Strain (LVS). Our objective is to (i) identify biologically relevant gene features indicative of respiratory infection, disease severity, and bacterial dissemination to the spleen, and (ii) develop a Weighted ℓ_1-norm Non-Parallel Support Vector Machines (ℓ_1-WNPSVM) that will utilize the selected genes to predict disease status. The ℓ_1-WNPSVM is trained on the lungs data and validated on the spleen data, introducing a form of transfer learning, with uninfected controls and Schu4 or LVS samples as classes. Currently, a direct application of existing NPSVM-type methods to analyze gene expression datasets, where the number of genes significantly exceeds the number of samples, is computationally impractical due to their large memory requirements. This work addresses these challenges and also generalizes to models of similar formulations by incorporating dimensionality reduction and gene selection into the NPSVM-type frameworks. The ℓ_1-WNPSVM method outperforms traditional machine learning techniques such as ANN, XGBoost, AdaBoost, GradBoost, KNN, SVM, Naive Bayes, Random Forest, Logistic Regression, and Decision Tree, achieving a 97% balanced accuracy on imbalanced data.

Conclusions We discovered sets of 235 genes exclusively expressed in the lungs and spleen tissues and utilized them to classify bacterial strains and controls, enabling prediction of disease status. Gene ontology is performed to reveal underlying metabolic pathways. Our analysis shows that signal transduction and disease (cancer) pathways are the most significant pathways activated in the lungs while gene expression (transcription), immune system, and disease (cancer) pathways are activated

in the spleen. Collectively, these pathways indicate a significant host response to infection, including how the bacteria interact with host tissues during dissemination.

Keywords Data Integration, Dimensionality Reduction, Gene Selection, Machine Learning, Optimization, Pathway Analysis, Support Vector Machine

Introduction

This paper develops novel optimization and Machine Learning (ML) algorithms that leverage weighted ℓ_2-norm regularization, dimensionality reduction, gene selection, and transfer learning to address complex biological challenges, enhance model performance, and enable scalability across diverse applications. Our primary objective is to identify host biomarkers of bacterial respiratory infections, elucidate dissemination patterns, and uncover underlying metabolic pathways of the host-pathogen interactions. This development is achieved by analyzing gene expression datasets, from the lungs and spleen tissues of genetically identical mice, infected intranasally, with two bacterial strains, *Francisella tularensis* - Schu4 and Live Vaccine Strain (LVS). We anticipate that our approach will advance biomarker and biological pathway discovery and aid in the development of therapeutics with efficacy against difficult-to-treat medically important pathogens and cancer.

Francisella tularensis is the causative agent of tularemia, also known as *rabbit fever*. According to the U.S. Centers for Disease Control and Prevention (CDC), tularemia is a rural disease that has been reported in all U.S. states except Hawaii.[1] *Francisella tularensis* is highly virulent with bioterrorism potential even at a very low dose. *Francisella tularensis* Schu4 strain is a highly virulent Type A strain that causes severe disseminated disease in humans and animals. *Francisella tularensis* LVS strain is a less virulent attenuated Type B strain that causes a milder form of disease. These strains are routinely used in drug and vaccine research studies because they are genetically stable and have well-defined coding capacity and virulence factors. Importantly, they provide a valuable framework for studying mechanisms underlying mild to severe disease progression and pathogen dissemination.

Our contributions

Schu4 induces a robust immune response due to its high virulence, making it useful for studying host-pathogen interactions and immune evasion mechanisms while LVS elicits a less intense immune response resulting in different levels of virulence. Accordingly, we have chosen to utilize host transcriptional data from infected lungs and spleen to define a set of biomarkers from biologically relevant gene features indicative of respiratory infection, disease severity and bacterial dissemination to the spleen.

Here and throughout this paper, we will use the matrices $\mathcal{G}_1 \in \mathbb{R}^{m\times n}$ and $\mathcal{G}_2 \in \mathbb{R}^{m\times n}$ to represent gene expression datasets from the lungs and spleen tissues, respectively, where each matrix consists of m genes and n samples. The columns of $\mathcal{G}_1$ and $\mathcal{G}_2$ will include controls, Schu4 samples, and LVS samples, in that order. The class matrices, $C_1 \in \mathbb{R}^{n_1\times m}$ and $C_2 \in \mathbb{R}^{n_2\times m}$, in our context, represent all the training data points from $\mathcal{G}_1$ or $\mathcal{G}_2$ that correspond to Class 1 (uninfected controls) and Class -1 (infected

[1] https://emergency.cdc.gov/agent/tularemia/faq.asp

Schu4 or LVS samples), respectively, with $n > n_1 + n_2$. The variables n_1 and n_2 represent the number of samples in each class.

Also, as with most transcriptional data, $\mathcal{G}_1 \in \mathbb{R}^{m\times n}$ and $\mathcal{G}_2 \in \mathbb{R}^{m\times n}$ are high-dimensional and have significantly more genes than samples, i.e., $m \gg n$. Thus, necessitating the need for gene selection, a process that reduces the number of genes by removing redundant and irrelevant genes. Notably, the gene selection methods can be categorized into supervised approach utilizing label information, unsupervised approach involving the use of unlabeled data to identify patterns or structures within the dataset, and semi-supervised or semi-unsupervised approach relying on both labeled and unlabeled data. For more discussions on gene selection techniques, see, e.g., [1] and references therein. We will focus on identifying the most informative gene expression profiles that potentially reflect host immune responses to bacterial dissemination across tissues, then use the selected group of genes to carry out binary classification tasks.

Our approach to gene selection and classification involves modifying the ℓ_1-norm generalized eigenvalue-type *Non-Parallel Proximal Support Vector Machine* [2, ℓ_1-NPSVM] to make it applicable to transcriptional data analysis. Currently, a direct application of existing ℓ_1-NPSVM architecture [3, 4] to gene expression datasets, is computationally impractical due to their large memory requirements. Specifically, the solution methods [3, 4] involve large-scale matrix-matrix computations, with $\tilde{C}_i^T$ and with $\tilde{C}_i$, $i = 1, 2$, at each iteration, thus resulting in matrices of size $(m + 1) \times (m + 1)$. This makes solving (1) infeasible for simultaneous gene selection and classification tasks. Moreover, the computational cost at each iteration is doubled especially since problems (1) are solved concurrently.

The ℓ_1-NPSVM formulation was first proposed in [2], and it is given by the following large-scale minimization problems

$$\min_{\tilde{\mathrm{w}}_i\in\mathbb{R}^{m+1}} \frac{\|\tilde{C}_i\tilde{\mathrm{w}}_i\|_1}{\|\tilde{C}_j\tilde{\mathrm{w}}_i\|_1}, \quad \tilde{C}_1 \in \mathbb{R}^{n_1\times(m+1)}, \quad \tilde{C}_2 \in \mathbb{R}^{n_2\times(m+1)}, \quad i, j \in \{1, 2\}, \quad i \neq j, \tag{1}$$

where $\tilde{C}_1 := [C_1\ \mathrm{e}_1]$, $\tilde{C}_2 := [C_2\ \mathrm{e}_2]$, and $\mathrm{e}_1 \in \mathbb{R}^{n_1}$, $\mathrm{e}_2 \in \mathbb{R}^{n_2}$ are vectors of ones. The quantity $\|\cdot\|_1$ denotes the ℓ_1-norm of a vector y given by

$$\|\mathrm{y}\|_1 = \sum_{i=1}^{s} |y_i|, \quad \mathrm{y} = [y_1, y_2, \ldots, y_s]^T \in \mathbb{R}^s,$$

where $|y|$ denotes the absolute value of $y \in \mathbb{R}$. Additionally, the solutions of (1), $\tilde{\mathrm{w}}_1 := [\mathrm{w}_1^T\ b_1]^T$ and $\tilde{\mathrm{w}}_2 := [\mathrm{w}_2^T\ b_2]^T$, determine two non-parallel separating hyperplanes,

$$\mathrm{x}^T\mathrm{w}_1 + b_1 = 0, \quad \text{and} \quad \mathrm{x}^T\mathrm{w}_2 + b_2 = 0, \quad \mathrm{w}_1, \mathrm{w}_2 \in \mathbb{R}^m, \tag{2}$$

where $b_1, b_2 \in \mathbb{R}$ are the *bias* terms and $\mathrm{x} \in \mathbb{R}^m$ is a new sample point to be classified to either Class 1 (C_1) or Class -1 (C_2) according to (24). The superscript ^{T} denotes transposition.

The contributions of this paper can be summarized in threefold:

1. We propose the use of weighted ℓ_2-norm for the regularization of (1). This leads to the minimization problems

$$\min_{\tilde{\mathrm{w}}_i \in \mathbb{R}^{m+1}} \frac{\|\tilde{C}_i \tilde{\mathrm{w}}_i\|_1 + \delta \|\tilde{\mathrm{w}}_i\|_M^2}{\|\tilde{C}_j \tilde{\mathrm{w}}_i\|_1}, \quad i, j \in \{1, 2\}, \quad i \neq j, \tag{3}$$

where

$$\delta \|\tilde{\mathrm{w}}_i\|_M^2 := \delta \tilde{\mathrm{w}}_i^T M \tilde{\mathrm{w}}_i, \quad i = 1, 2, \tag{4}$$

and $M \in \mathbb{R}^{(m+1)\times(m+1)}$ is a symmetric positive definite (SPD) matrix or a diagonal matrix different from the identity matrix *I*. The quantity $\delta \|\tilde{\mathrm{w}}_i\|_M^2, i = 1, 2$, are referred to as the *regularization terms*, and for a fixed *i* and *j* with $i \neq j, \delta > 0$ is a regularization parameter that balances the influence of $\|\tilde{\mathrm{w}}_i\|_M^2$ on the solutions of (3). A common approach to determine δ is by cross-validation or grid search. Our approach of regularizing (1) is consistent with the Grassmannian regularization which specifies *M* in relation to the *Graph Laplacian Matrix*, see, e.g., [5, 6]. Also, by following [3, 4], we can express (1) as constrained minimization problems

$$\min_{\tilde{\mathrm{w}}_i} \frac{1}{2} \|\tilde{C}_i \tilde{\mathrm{w}}_i\|_1 + \frac{\delta}{2} \|\tilde{\mathrm{w}}_i\|_M^2, \quad \text{s.t.} \quad \|\tilde{C}_j \tilde{\mathrm{w}}_i\|_1 = 1, \quad i, j \in \{1, 2\} \quad i \neq j. \tag{5}$$

We will refer to (5) as the Weighted ℓ_1-NPSVM (ℓ_1-WNPSVM). When $M = I$, the resulting problems, referred to as the ℓ_1-GEPSVM [4], are special cases of the ℓ_q -NPSVM, $q > 0$, which is described in [3]. The ℓ_q-NPSVM regularizes (1) by adding $\delta \|\tilde{\mathrm{w}}_i\|_q^q$ to the numerator. We will show in the *Methods* section that $\|\tilde{\mathrm{w}}_i\|_q^q \approx \|\tilde{\mathrm{w}}_i\|_M^2$ for $q > 0$, where M is a diagonal matrix that changes at each iteration and also depends on q and $\tilde{\mathrm{w}}_i, i = 1, 2$. Other related but different regularization techniques are presented in, e.g., [7–10]. Our approach generalizes the regularization methods described in [3, 4]. This is shown via a proposition in the *Methods* section.

2. We introduce new methodologies that integrate dimensionality reduction and feature selection into the ℓ_1-WNPSVM framework. This is to enable practical application to gene expression datasets, particularly in situations where the number of genes significantly exceeds the number of samples ($m \gg n$). The dimensionality reduction of $\mathcal{G}_1$ and $\mathcal{G}_2$ is accomplished by solving constrained Weighted ℓ_1-norm Generalized Eigenvalue-type Problems (ℓ_1-WGEPs)

$$\min_{\mathrm{z}_i \in \mathbb{R}^n} \frac{1}{2} \|\mathcal{G}_i \mathrm{z}_i\|_1 + \frac{\delta}{2} \|\mathrm{z}_i\|_M^2 \quad \text{s.t.} \quad \|\mathcal{G}_j \mathrm{z}_i\|_1 = 1, \quad i, j \in \{1, 2\}, \quad i \neq j, \tag{6}$$

where $M \in \mathbb{R}^{n\times n}$ is a SPD matrix that may depend on $\mathrm{z}_i, i = 1, 2$. After solving (6), $\mathcal{G}_1$ and $\mathcal{G}_2$ are projected onto one-dimensional subspaces spanned by the solutions, $\mathrm{z}_{1,\delta}$ and $\mathrm{z}_{2,\delta}$, to identify $k \ll m$ most informative genes. This reduces $\mathcal{G}_1$ and $\mathcal{G}_2$ to small size matrices of dimensions $k \times n$, that are suitable for use in (5) for binary classification. The minimization problems (6) can be thought of as *unsupervised multiclass gene selection problems* since each $\mathcal{G}_i, i = 1, 2$, consist of controls, Schu4 samples and LVS samples as classes, and the most informative genes across the classes are selected synchronously.

The projection directions are determined by the solutions of (6). The best direction for each dataset is determined by computing the maximum projection scores of $\mathcal{G}_1$ and $\mathcal{G}_2$ with respect to $\mathrm{z}_{1,\delta}$ and $\mathrm{z}_{2,\delta}$, i.e.,

$$\mathcal{G}_1 : \max_{1\le i\le m}\left\{\frac{|\mathrm{g}_i^T \mathrm{z}_{1,\delta}|}{\|\mathrm{z}_{1,\delta}\|_2}, \frac{|\mathrm{g}_i^T \mathrm{z}_{2,\delta}|}{\|\mathrm{z}_{2,\delta}\|_2}\right\}, \quad \text{and} \quad \mathcal{G}_2 : \max_{1\le i\le m}\left\{\frac{|\hat{\mathrm{g}}_i^T \mathrm{z}_{1,\delta}|}{\|\mathrm{z}_{1,\delta}\|_2}, \frac{|\hat{\mathrm{g}}_i^T \mathrm{z}_{2,\delta}|}{\|\mathrm{z}_{2,\delta}\|_2}\right\}, \tag{7}$$

where $\mathrm{g}_i, \hat{\mathrm{g}}_i \in \mathbb{R}^n$, $i = 1, 2, \ldots, m$, are the gene expression vectors in $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively. Note that a direction that gives the maximum score for each dataset is used for gene selection. We will select $k \ll m$ most significant genes from $\mathcal{G}_1$ and $\mathcal{G}_2$ by computing their projection scores with a chosen projection direction, where the top k projection scores correspond to the top k most informative genes. Details of dimensionality reduction and gene selection are provided by Algorithms 1 and 2 in the *Methods* section. The processes implemented by these algorithms and more, are summarized in Fig. 1 below.

3. We discovered sets of 235 genes that are uniquely expressed in the lungs and spleen tissues. Notably, the gene sets from the lungs and spleen do not overlap. Among these genes, 235 and 234 gene identifiers from the lungs and spleen, respectively, are available for transcriptomic pathway analysis in Reactome. Our functional enrichment analyses identified 65 of the 235 identifiers and 75 of the 234 identifiers that are linked to 503 and 399 pathways, respectively. The top 10 most significant pathways activated in the lungs are associated with essential functions in signal transduction and disease (cancer) pathways, while the leading pathways in the spleen comprise of

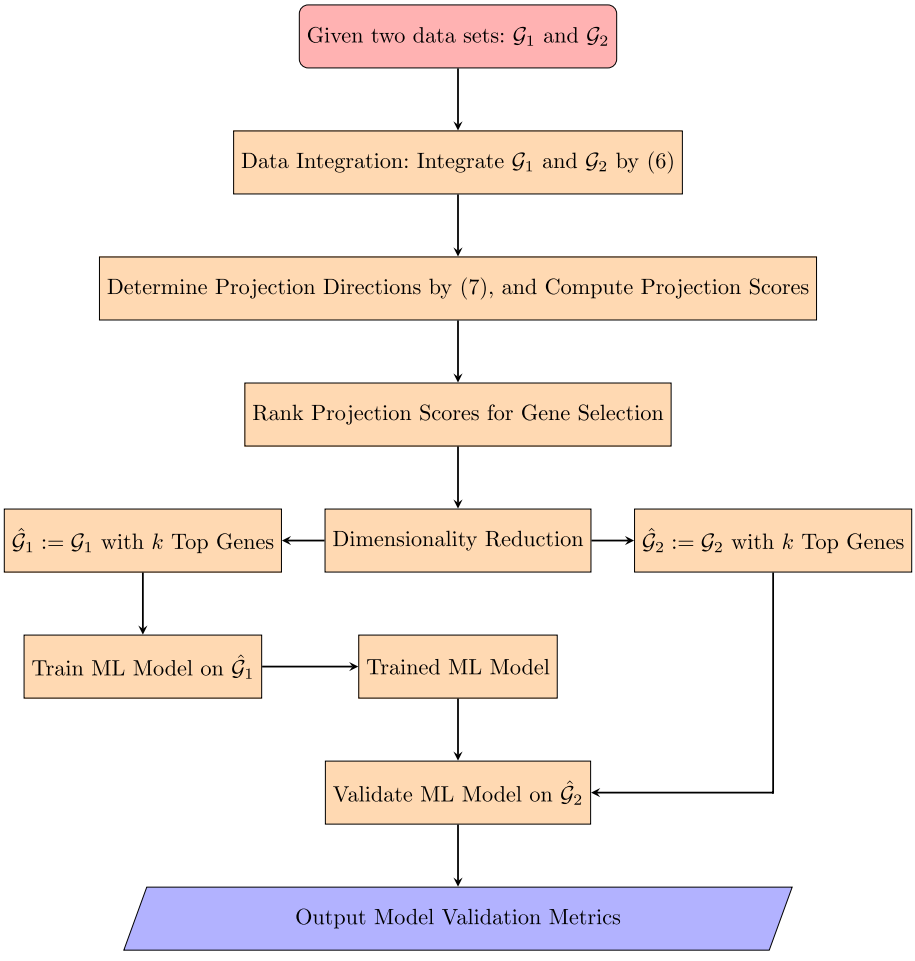

Fig. 1 A pipeline for model implementation

gene expression (transcription), immune system, and disease (cancer) pathways. A more elaborate discussion focusing on pathways discovery is presented in *Discussion* section.

In the remainder of this section, we will provide some details on related methods.

Related methods

The idea to use two non-parallel separating hyperplanes for binary classification tasks was first proposed in a seminal work by Mangasarian and Wild [11]. It requires that each hyperplane be closest to the samples of one class and farthest from the samples of the other class. Therefore, the assignment of x to C_1 or C_2 depends on the hyperplane it is most proximal to.

Several different formulations of (1) have been proposed and applied in the literature, with binary classification applications to functional data [12], crossed data [11], satellite ship image data [13], ellipsoidal uncertainty data [14], gene expression data [15] including genomic and proteomic problems [16], biomedical data [17], lungs cancer cells [18], as well as applications to multiclass [11] and multiview learning [19] problems. It is noteworthy that the (regularized) ℓ_1-NPSVM has been applied to binary classification tasks in [3, 4, 2], however, this paper is the first to apply (1) in the context of microarray or transcriptomic pattern analysis. Our approach differs significantly from the models described in [15], including those presented in [20, 21] that utilize multi-layer deep neural networks for bioinformatics analysis.

The use of ℓ_1-norm in (1) instead of squared ℓ_2-norm is to reduce the sensitivity of the solutions to noise and outliers. The minimization problems (1) with the squared ℓ_2 -norm are discussed in [11]. They are commonly referred to as the *Generalized Eigenvalue Proximal Support Vector Machine* (GEPSVM). The GEPSVM results in the *Rayleigh Quotients,*

$$\min_{\tilde{\mathrm{w}}_i \in \mathbb{R}^{m+1}} \frac{\|\tilde{C}_i \tilde{\mathrm{w}}_i\|_2^2}{\|\tilde{C}_j \tilde{\mathrm{w}}_i\|_2^2}, \quad i, j \in \{1, 2\}, \quad i \neq j, \tag{8}$$

where for any matrix A and a vector x of compatible sizes, $\|A\mathrm{x}\|_2 := \sqrt{\mathrm{x}^T A^T A \mathrm{x}}$. The solutions of (8) are readily determined by solving the generalized eigenvalue problems

$$\tilde{C}_i^T \tilde{C}_i \tilde{\mathrm{w}}_i = \lambda_i \tilde{C}_j^T \tilde{C}_j \tilde{\mathrm{w}}_i, \quad i, j \in \{1, 2\}, \quad i \neq j. \tag{9}$$

The eigenvectors corresponding to the smallest eigenvalues of $(\tilde{C}_i^T \tilde{C}_i, \tilde{C}_j^T \tilde{C}_j)$, for $i, j = 1, 2$, $i \neq j$, are the optimal solutions of (8). Also, the objective functions in (8) share the same eigenvectors but have reciprocal eigenvalues $1/\lambda_i$. Therefore, the eigenvector corresponding to the largest eigenvalue of (9) when $i = 1$ and $j = 2$ is also an eigenvector corresponding to the smallest eigenvalue of (9) when $i = 2$ and $j = 1$.

Additionally, the Rayleigh quotient in (8) is bounded and ranges over an interval determined by their minimum and maximum eigenvalues when the denominator is positive definite [22]. However, the defining matrices in (8) are rank-deficient, as they have a maximum rank of m despite being of order $m + 1$. This makes the minimization problems (8) prone to becoming singular, with $N(C_i) \cap N(C_j) \neq 0$, where $N(A)$ denotes the null space of the matrix A and 0 is the zero vector [7]. Consequently, even when (8) is defined by using positive semi-definite matrices, it often fails to produce meaningful

approximations of the generalized eigenvectors [3]. As a remedy to the difficulty in solving (8), [11] proposed a solution approach that replaces (8) with nearby problems that are more stable to solve. Specifically, they consider regularized minimization problems

$$\min_{\tilde{\mathrm{w}}_i \in \mathbb{R}^{m+1}} \frac{\|\tilde{C}_i \tilde{\mathrm{w}}_i\|_2^2 + \delta \|\tilde{\mathrm{w}}_i\|_2^2}{\|\tilde{C}_j \tilde{\mathrm{w}}_i\|_2^2}, \quad i, j \in \{1, 2\}, \quad i \neq j. \tag{10}$$

The replacement of (8) by (10) is known as Tikhonov regularization [23] in *standard form*. It should be noted that although the squared ℓ_2-norm regularization terms in (10) can improve the stability and classification accuracy of the GEPSVM, it may also amplify the influence of outliers in the samples and cause decreased accuracy. In comparison, the ℓ_1-NPSVM has been found to be more robust to outliers than the GEPSVM, as discussed in [3, 4]. Moreover, the GEPSVM has been demonstrated in [11] to yield better accuracy than the Support Vector Machine [24, SVM] when applied to solve classical 'XOR' and 'Cross Planes' problems.

This paper is organized as follows. The *Methods* section discusses the solution methods for the minimization problems (5) and (6), and describes how (6) can be applied to achieve dimensionality reduction and gene selection. The data integration process involves fusing uninfected (controls), Schu4 samples, and LVS samples from the lungs and spleen tissues. Once a small set of genes that indicate bacterial dissemination of Schu4 and LVS strains has been identified for both tissues, transfer learning is carried out by using (5). Information from the lungs is used to train transfer learning models while the trained models are applied to the spleen data to compute balanced classification accuracy scores. Transcriptomic pattern analysis of a select group of genes, data preprocessing, and model training are discussed in the *Results* section. The *Discussion* section discusses the performance of various ML models, such as Artificial Neural Networks (ANN) [25], Adaptive Boosting (AdaBoost) [26], Gradient Boosting (GradBoost) [27], Extreme Gradient Boosting (XGBoost) [28], K-Nearest Neighbors (KNN) [29], Logistic Regression [30], Random Forest [31], Naive Bayes [32], Support Vector Machine (SVM), and Decision Tree [33]. These models are trained with uninfected controls and Schu4 or LVS samples as classes and are compared with our proposed ℓ_1-WNPSVM model. Also presented in this section is the pathway (gene enrichment) analysis. The *Conclusion* section provides concluding remarks.

Methods

This section describes the solution methods for (5) and (6). Since the problems are analogous, we will instead, without loss of generality, consider the minimization problem

$$\min_{\mathrm{z} \in \mathbb{R}^n} \frac{1}{2}\|A\mathrm{z}\|_1 + \frac{\delta}{2}\|\mathrm{z}\|_M^2 \quad \text{s.t.} \quad \|B\mathrm{z}\|_1 = 1, \quad A \in \mathbb{R}^{m \times n}, \quad B \in \mathbb{R}^{m \times n}, \tag{11}$$

where the vectors $\mathrm{a}_i \in \mathbb{R}^n$, and $\mathrm{b}_i \in \mathbb{R}^n$, $i = 1, 2, \ldots, m$, denote the rows of A and B, respectively. Here and below, we will use boldface letters, e.g., a, to denote vectors and capital letters, e.g., A, to denote matrices.

By following [3, 4], we will approximate $\|\cdot\|_1$ by a weighted ℓ_2-norm, then reformulate (11) into a compact and stable quadratic minimization problem. Hence, for $\epsilon > 0$,

$$\begin{aligned}\|Az\|_1 &= \sum_{i=1}^{m} |\mathrm{a}_i^T \mathrm{z}| = \sum_{i=1}^{m} \frac{|\mathrm{z}^T \mathrm{a}_i|}{|\mathrm{z}^T \mathrm{a}_i|} |\mathrm{a}_i^T \mathrm{z}| = \sum_{i=1}^{m} \frac{\mathrm{z}^T \mathrm{a}_i \mathrm{a}_i^T \mathrm{z}}{|\mathrm{z}^T \mathrm{a}_i|} \\ &\approx \mathrm{z}^T \left(\sum_{i=1}^{m} \frac{\mathrm{a}_i \mathrm{a}_i^T}{((\mathrm{z}^T \mathrm{a}_i)^2 + \epsilon^2)^{\frac{1}{2}}} \right) \mathrm{z} = \mathrm{z}^T \Gamma_\epsilon(\mathrm{z}) \mathrm{z} =: \|\mathrm{z}\|^2_{\Gamma_\epsilon},\end{aligned} \tag{12}$$

where

$$\Gamma_\epsilon := \Gamma_\epsilon(\mathrm{z}) = A^T \mathrm{diag}\left(\psi_\epsilon(\mathrm{z}, \mathrm{a}_1), \psi_\epsilon(\mathrm{z}, \mathrm{a}_2), \ldots, \psi_\epsilon(\mathrm{z}, \mathrm{a}_m)\right) A, \tag{13}$$

and $\psi_\epsilon(\mathrm{z}, \mathrm{a}_i) := ((\mathrm{z}^T \mathrm{a}_i)^2 + \epsilon^2)^{-\frac{1}{2}}$.

The matrix Γ_ϵ is symmetric, and the first term in line 2 of (12) accounts for the possibility of $|\mathrm{z}^T \mathrm{a}_i| \approx 0$, by adding a small parameter ϵ to the denominator. This is different from [3, 4] which assumes that $|\mathrm{z}^T \mathrm{a}_i| \neq 0$ and expresses $\|Az\|_1$ as in line 1 of (12). Also,

$$\|B\mathrm{z}\|_1 = \sum_{j=1}^{m} \mathrm{sign}(\mathrm{b}_j^T \mathrm{z}) \mathrm{b}_j^T \mathrm{z} = \varphi(\mathbf{y})^T \mathrm{z}, \qquad \varphi(\mathbf{z}) := B^T \mathrm{sign}\,(B\mathrm{z}) \in \mathbb{R}^n, \tag{14}$$

where $\mathtt{sign}(\mathrm{x})$ is an element-wise sign of a vector x. Substitution of (12) and (14) into (11) gives

$$\min_{\mathrm{z} \in \mathbb{R}^n} \frac{1}{2} \|\mathrm{z}\|^2_{\Gamma_\epsilon} + \frac{\delta}{2} \|\mathrm{z}\|^2_M \quad \text{s.t.} \;\; \varphi(\mathbf{z})^T \mathrm{z} = 1. \tag{15}$$

Equation (15) leads to the constrained quadratic programming problem

$$\min_{\mathrm{z}} \frac{1}{2} \mathrm{z}^T \mathcal{J}_\epsilon(\mathrm{z}) \mathrm{z} \quad \text{s.t.} \;\; \varphi(\mathbf{z})^T \mathrm{z} = 1, \tag{16}$$

where $\mathcal{J}_\epsilon(\mathrm{z}) = \Gamma_\epsilon(\mathrm{z}) + \delta M$.

We will employ an *iterative optimization strategy* described in [3, 4] to compute the solution of (16). The basic idea is as follows. Let $\mathrm{z}^{(k)}$, $k \geq 1$, be the currently available approximation of the solution

$$\mathrm{z}^* = \arg\min_{\mathrm{z}} \left\{ \frac{1}{2} \mathrm{z}^T \mathcal{J}_\epsilon(\mathrm{z}) \mathrm{z} \quad \text{s.t.} \;\; \varphi(\mathbf{z})^T \mathrm{z} = 1 \right\}. \tag{17}$$

Then

$$\mathcal{J}_\epsilon(\mathrm{z}^{(k)}) = \Gamma_\epsilon(\mathrm{z}^{(k)}) + \delta M \;\; \text{and} \;\; \varphi(\mathrm{z}^{(k)}) = B^T \mathrm{sign}\left((B\mathrm{z}^{(k)})\right).$$

To derive an update for $\mathrm{z}^{(k+1)}$, denote the Langrangian of (16) as

$$\mathcal{L}(\mathrm{z}, \alpha) = \frac{1}{2} \mathrm{z}^T \mathcal{J}_\epsilon(\mathrm{z}^{(k)}) \mathrm{z} - \alpha(\varphi(\mathrm{z}^{(k)})^T \mathrm{z} - 1),$$

where α is the Langrangian multiplier. Then

$$0 = \frac{\partial \mathcal{L}(\mathrm{z}, \alpha)}{\partial \alpha} = \varphi(\mathrm{z}^{(k)})^T \mathrm{z} - 1 \implies \varphi(\mathrm{z}^{(k)})^T \mathrm{z} = 1. \tag{18}$$

$$0 = \frac{\partial \mathcal{L}(\mathrm{z}, \alpha)}{\partial \mathrm{z}} = \mathcal{J}_\epsilon(\mathrm{z}^{(k)}) \mathrm{z} - \alpha \varphi(\mathrm{z}^{(k)}) \implies \mathrm{z} = \alpha \mathcal{J}_\epsilon(\mathrm{z}^{(k)})^{-1} \varphi(\mathrm{z}^{(k)}). \tag{19}$$

Substitute (19) into (18) to determine α, then use the result in (19) to obtain the desired update for $\mathrm{z}^{(k+1)}$ as

$$\mathrm{z}^{(k+1)} := \frac{\mathcal{J}_\epsilon(\mathrm{z}^{(k)})^{-1}\varphi(\mathrm{z}^{(k)})}{\varphi(\mathrm{z}^{(k)})^T\mathcal{J}_\epsilon(\mathrm{z}^{(k)})^{-1}\varphi(\mathrm{z}^{(k)})}. \tag{20}$$

Thus, at each iteration, a new approximation $\mathrm{z}^{(k+1)}$ of z^* is computed by using the current $\mathcal{J}_\epsilon(\mathrm{z}^{(k)})$ and $\varphi(\mathrm{z}^{(k)})$. This process continues repeatedly until certain convergence conditions are met.

The solution method so described is summarized by Algorithm 1. This algorithm solves (11) and the minimization problem

$$\min_{\mathrm{z}\in\mathbb{R}^n} \frac{1}{2}\|B\mathrm{z}\|_1 + \frac{\delta}{2}\|\mathrm{z}\|_M^2 \quad \text{s.t.} \quad \|A\mathrm{z}\|_1 = 1, \quad A \in \mathbb{R}^{m\times n}, \quad B \in \mathbb{R}^{m\times n}, \tag{21}$$

simultaneously, by reversing the roles of A and B in step 14. The influence of the parameters, ϵ and δ, will be examined in the *Results* section. The convergence results for Algorithm 1 are readily shown by following [3, 4]. By following Algorithm 1, the computational complexity per iteration is $O(mn^2 + n^3)$. The total complexity after k iterations is $O(k \cdot (mn^2 + n^3))$. For $m \gg n$, the term $O(k \cdot mn^2)$ dominates, but this may depend on the number of informative features selected, say, $\hat{k} \ll m$.

In the sequel, the following proposition demonstrates that our approach of regularizing (1) with a weighted ℓ_2-norm generalizes the regularization methods described in [3, 4].

Proposition 0.1 Consider the minimization problem [3]

$$\min_{\mathrm{z}\in\mathbb{R}^n} \frac{1}{2}\|A\mathrm{z}\|_1 + \frac{\delta}{2}\|\mathrm{z}\|_q^q \quad \text{s.t.} \quad \|B\mathrm{z}\|_1 = 1, \quad A \in \mathbb{R}^{m\times n}, \quad B \in \mathbb{R}^{m\times n}, \tag{22}$$

analogous to (11) and (21). Then (22) can be expressed in the form (16).

Proof Let $\phi_\epsilon(z_i) := \left(z_i^2 + \epsilon^2\right)^{\frac{1}{2}}$ for $\epsilon > 0$. Then

$$\|\mathrm{z}\|_q^q \approx \sum_{i=1}^n \left(z_i^2 + \epsilon^2\right)^{(q-2)/2} z_i^2 = \mathrm{z}^T\left(\sum_{i=1}^n \phi_\epsilon(z_i)^{(q-2)}\right)\mathrm{z} = \mathrm{z}^T D_{\epsilon,q}(\mathrm{z})\mathrm{z} = \|D_{\epsilon,q}^{\frac{1}{2}}(\mathrm{z})\mathrm{z}\|_2^2, \ 0 < q < 2,$$

where $\mathrm{z} := [z_1, z_2, \ldots, z_n]^T$, and

$$D_{\epsilon,q}(\mathrm{z}) = \mathrm{diag}\left(\phi_\epsilon(z_i)^{(q-2)}, \phi_\epsilon(z_2)^{(q-2)}, \ldots, \phi_\epsilon(z_n)^{(q-2)}\right) \in \mathbb{R}^{n\times n}. \tag{23}$$

Setting $M := D_{\epsilon,q}(\mathrm{z})$ completes the proof. □

In the computed examples of *Results* section, we will compare the performance of ℓ_1 -WGEPs for $M := D_{\epsilon,1}(\mathrm{z})$ and $M := D_{\epsilon,1}(\mathrm{z})^2$ to illustrate that the latter matrix captures more information than the former in terms of the overall projection scores.

Algorithm 1 The Solution Method for the Approximate Solution of (11) and (21)

Input: $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{m\times n}$, where $\mathbf{a}_i \in \mathbb{R}^n$, and $\mathbf{b}_i \in \mathbb{R}^n$, $i = 1, 2, \ldots, m$, denote the rows of A and B, respectively, $M \in \mathbb{R}^{n\times n}$ may or may not depend on $\mathbf{z}_1$, $\mathbf{z}_2$ and q

Output: The solutions, $\mathbf{z}_{1,\delta} \in \mathbb{R}^n$ of (11), and $\mathbf{z}_{2,\delta} \in \mathbb{R}^n$ of (21)

Initialize $\epsilon > 0$, `tol`, $\delta > 0$, $\mathbf{z}_1^{(0)}$ and $\mathbf{z}_2^{(0)}$
while $k \geq 0$ **do**
  $k := k + 1$
  if $\mathcal{M} := M(\mathbf{z}_1)$ **then**
    Compute $M(\mathbf{z}_1^{(k-1)}) := D_{\epsilon,q}\left(\mathbf{z}_1^{(k-1)}\right)$ or $D_{\epsilon,q}\left(\mathbf{z}_1^{(k-1)}\right)^2$ by (23) $O(n)$
    else
      $\mathcal{M} := M$
    end
  end
  Compute $\Gamma_\epsilon\left(\mathbf{z}_1^{(k-1)}\right)$ by (13) $O(mn^2)$
  Compute $\varphi\left(\mathbf{z}_1^{(k-1)}\right)$ by (14) $O(mn)$
  Compute $\mathcal{J}_\epsilon\left(\mathbf{z}_1^{(k-1)}\right) := \Gamma_\epsilon\left(\mathbf{z}_1^{(k-1)}\right) + \delta\mathcal{M}$ $O(mn^2)$
  Compute $\mathbf{z}_1^{(k)}$ by (20) $O(n^3)$
  Repeat steps 4-13 with $\mathbf{z}_2^{(0)}$ to compute $\mathbf{z}_2^{(k)}$ by reversing the roles of A and B
  if $\frac{\left\|\mathbf{z}_1^{(k)}-\mathbf{z}_1^{(k-1)}\right\|_2}{\left\|\mathbf{z}_1^{(k-1)}\right\|_2} \leq$ `tol` and $\frac{\left\|\mathbf{z}_2^{(k)}-\mathbf{z}_2^{(k-1)}\right\|_2}{\left\|\mathbf{z}_2^{(k-1)}\right\|_2} \leq$ `tol` **then**
    break
  end
  $\mathbf{z}_{1,\delta} := \mathbf{z}_1^{(k)}$ and $\mathbf{z}_{2,\delta} := \mathbf{z}_2^{(k)}$
end

Algorithm 2 The ℓ_1-WGEPs for Dimensionality Reduction and Gene Selection

Input: $\mathcal{G}_1 \in \mathbb{R}^{m\times n}$ and $\mathcal{G}_2 \in \mathbb{R}^{m\times n}$, $m \gg n$, m is the number of genes and n is the number of samples

Output: $\tilde{\mathcal{G}}_1 \in \mathbb{R}^{k\times n}$, $\tilde{\mathcal{G}}_2 \in \mathbb{R}^{k\times n}$, $k \ll m$, k is the number of top informative genes to be selected for binary classification

1 Solve the minimization problems (6) for $\mathbf{z}_{1,\delta}$ and $\mathbf{z}_{2,\delta}$ by using Algorithm 1 with $A := \mathcal{G}_1$ and $B := \mathcal{G}_2$

2 Project each row of $\mathcal{G}_1$ and $\mathcal{G}_2$ onto $\mathbf{z}_{j,\delta}$, $j = 1, 2$, to determine the best projection direction for each dataset, by computing the projection scores

$$\mathcal{G}_1 : \max_{1\leq i\leq m}\left\{\frac{|\mathbf{g}_i^T\mathbf{z}_{1,\delta}|}{\|\mathbf{z}_{1,\delta}\|_2}, \frac{|\mathbf{g}_i^T\mathbf{z}_{2,\delta}|}{\|\mathbf{z}_{2,\delta}\|_2}\right\}, \text{ and } \mathcal{G}_2 : \max_{1\leq i\leq m}\left\{\frac{|\hat{\mathbf{g}}_i^T\mathbf{z}_{1,\delta}|}{\|\mathbf{z}_{1,\delta}\|_2}, \frac{|\hat{\mathbf{g}}_i^T\mathbf{z}_{2,\delta}|}{\|\mathbf{z}_{2,\delta}\|_2}\right\}$$

3 Rank the projection scores corresponding to $\mathcal{G}_1$ and $\mathcal{G}_2$ with respect to a chosen best direction, and then use the (ranked) indices to select the top k most informative genes in $\mathcal{G}_1$ and $\mathcal{G}_2$

4 Denote the results of step 3 by $\tilde{\mathcal{G}}_1$ and $\tilde{\mathcal{G}}_2$, respectively

Algorithm 2 describes the ℓ_1-WGEPs for dimensionality reduction and gene selection. By applying Algorithm 2 to $\mathcal{G}_1 \in \mathbb{R}^{m\times n}$ and $\mathcal{G}_2 \in \mathbb{R}^{m\times n}$, the matrices are reduced to small size matrices, $\tilde{\mathcal{G}}_1 \in \mathbb{R}^{k\times n}$, and $\tilde{\mathcal{G}}_2 \in \mathbb{R}^{k\times n}$, respectively. The rows of $\tilde{\mathcal{G}}_1$ and $\tilde{\mathcal{G}}_2$ contain the top $k \ll m$ most informative genes selected for the binary classification of controls and Schu4 or LVS samples. In other words, to solve (5) by Algorithm 3, we use $C_1 \in \mathbb{R}^{n_1\times k} \subset \tilde{\mathcal{G}}_1^T \in \mathbb{R}^{n\times k}$, and $C_2 \in \mathbb{R}^{n_2\times k} \subset \tilde{\mathcal{G}}_2^T \in \mathbb{R}^{n\times k}$, where $n_1 + n_2 < n$. Note that n_1 and n_2 are the number of controls and Schu4 or LVS samples in $\mathcal{G}_1$ or $\mathcal{G}_2$, respectively.

Algorithm 3 describes the solution methods for the proposed ℓ_1-WNPSVM formulations. The outputs of Algorithm 3, $\mathrm{w}_1, \mathrm{w}_2, b_1$, and b_2, determine two non-parallel separating hyperplanes (2). A new sample point x is classified to either Class 1 (C_1) or Class -1 (C_2) by using

$$\text{Class}(\mathrm{x}) := \text{sign}(|\mathrm{x}^T\mathrm{w}_1 + b_1| - |\mathrm{x}^T\mathrm{w}_2 + b_2|), \tag{24}$$

where $\text{sign}(x)$ is the sign function of $x \in \mathbb{R}$. This is equivalent to

$$\text{Class(x)} := \begin{cases} 0 & \text{if } |\text{x}^T \text{w}_1 + b_1| \leq |\text{x}^T \text{w}_2 + b_2| \\ 1 & \text{if otherwise.} \end{cases}$$

Algorithm 3 The ℓ_1-WNPSVM (5) for Binary Classification

Input: $C_1 \in \mathbb{R}^{n_1 \times k}$, $C_2 \in \mathbb{R}^{n_2 \times k}$, n is the total number of samples with $n_1, n_2 < n$, and k is the number of top informative genes to be selected by Algorithm 2

Output: $\mathbf{w}_1, \mathbf{w}_2, b_1,$ and b_2

1 Solve the minimization problems (5) for $\tilde{\mathbf{w}}_{1,\delta}$ and $\tilde{\mathbf{w}}_{2,\delta}$ by using Algorithm 1 with $A := \tilde{C}_1 \in \mathbb{R}^{n_1 \times (k+1)}$ and $B := \tilde{C}_2 \in \mathbb{R}^{n_2 \times (k+1)}$, where $\tilde{C}_1 := [C_1 \ \mathbf{e}_1]$, and $\tilde{C}_2 := [C_2 \ \mathbf{e}_2]$

2 Set $\tilde{\mathbf{w}}_{1,\delta} := [\mathbf{w}_1^T \ b_1]^T$ and $\tilde{\mathbf{w}}_{2,\delta} := [\mathbf{w}_2^T \ b_2]^T$

In the following section, we discuss the binary classification of uninfected controls and Schu4 or LVS samples, and compare ℓ_1-WNPSVM method with several baseline algorithms.

Results

This section discusses the performance of several baseline ML classifiers, such as Random Forest, Logistic Regression, SVM, Decision Tree, Naive Bayes, KNN, AdaBoost, GradBoost, XGBoost, and ANN, on binary classification problems. These methods will be compared with the ℓ_1-WNPSVM in Jupyter Notebook, version 6.3.0. All computations are carried out on a Dell computer running Windows 11 with 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz and 16 GB RAM.

The lungs and spleen datasets

The lungs and spleen datasets utilized in this study are publicly available through the Gene Expression Omnibus (GEO) database using accession number GSE22203 (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE22203). This dataset utilizes F. tul arensis Schu4 and LVS strains to understand how infection with virulent F. tularensis leads to rapid bacterial dissemination and lethal infection. The distinct virulence profiles of both strains provide a model to assess differences in host interaction and response genes. For additional details on data collection and preprocessing, including bacterial cultures, murine infection models, histopathology, RNA sequencing, and microarray analysis; see [34].

The lungs and spleen datasets consist of 6 controls, 24 Schu4 samples, and 24 LVS samples in that order, resulting in $n = 54$ samples. The tissue samples are collected at four time points, 12, 24, 48, and 120 h, with a total of $m = 37,632$ genes measured at each time point under the same experimental conditions.

The lungs and spleen datasets are imbalanced, with only 6 uninfected controls each. Therefore, to preprocess and integrate both datasets as well as address the imbalance, the *Synthetic Minority Oversampling Technique* [35, SMOTE] is applied to the available 6 controls in other to generate additional 18 synthetic controls. The SMOTE algorithm[2] applies linear interpolation to synthesize a new control between two control points. This stage is carried out in MATLAB (R2021a). Genes with standard deviations less than 0.9 are thrown out from the lungs and spleen datasets, leaving a total of 19, 735 genes in each dataset.

[2] \ nique-smote.

Integrating the lungs and spleen datasets for dimensionality reduction and gene selection

The ℓ_1-WGEPs (6) are used to integrate the preprocessed lungs and spleen datasets. The goal of data integration is to identify a small group of informative genes that significantly contribute to the process of bacterial dissemination of Schu4 and LVS across tissues, and more so, dimensionally reduce the sizes of the lungs ($\mathcal{G}_1$) and spleen ($\mathcal{G}_2$) datasets.

The solutions of (6) are determined by Algorithm 1 with $M := D_{\epsilon,1}(\mathrm{z})^2$ and $\mathrm{tol} = 10^{-4}$, while step 2 of Algorithm 2 or (7) determines the best direction to project each dataset for gene selection. The projection directions, $\mathrm{z}_{1,\delta}$ and $\mathrm{z}_{2,\delta}$, are found to be best for the spleen and lungs datasets, respectively.

Figure 2 displays the projection scores corresponding to $\mathcal{G}_1$ and $\mathcal{G}_2$ for different values of the parameters, δ and ϵ. It is evident from Fig. 2 that $\delta = 1000$ and $\epsilon = 10^{-3}$ yield the highest projection scores for both datasets, indicating that the corresponding ℓ_1-WGEPs capture more information than with other δ and ϵ values shown in Fig. 2.

We also compare the amount of information that the ℓ_1-WGEPs capture with $M := D_{\epsilon,1}(\mathrm{z})^2$ to those of $M := D_{\epsilon,1}(\mathrm{z})$ and $M := I$, and found that the latter choices of M yield lower projection scores than the former choice for the same parameter values in Fig. 2. Figures 2, 3 and 4 demonstrate that regularizing (6) with $\frac{\delta}{2}\|\mathrm{z}\|^2_{M:=D_{\epsilon,1}(\mathrm{z})^2}$ captures more informative gene expression profiles across tissues than with the custom regularization terms, $\frac{\delta}{2}\|\mathrm{z}\|_1 \approx \frac{\delta}{2}\|\mathrm{z}\|^2_{M:=D_{\epsilon,1}(\mathrm{z})}$ and $\frac{\delta}{2}\|\mathrm{z}\|_2^2$. In addition, Figs. 3 and 4 show that the projection scores corresponding to different ϵ and δ are relatively the same for $M := D_{\epsilon,1}(\mathrm{z})$ and $M := D_{\epsilon,2}(\mathrm{z})$, respectively.

A total of 235 and 217 genes are selected from the lungs and spleen datasets, respectively, by using the `KneeLocator` operator [36] from the `Kneed` python package. The `KneeLocator` locates the maximum curvature of the black curves in Figs. 2, 3 and 4. As shown in Fig. 2, the maximum curvature of the 'black' curves occur at (235, 10.72) and (217, 8.87) for the lungs and spleen data, respectively, which are greater than those

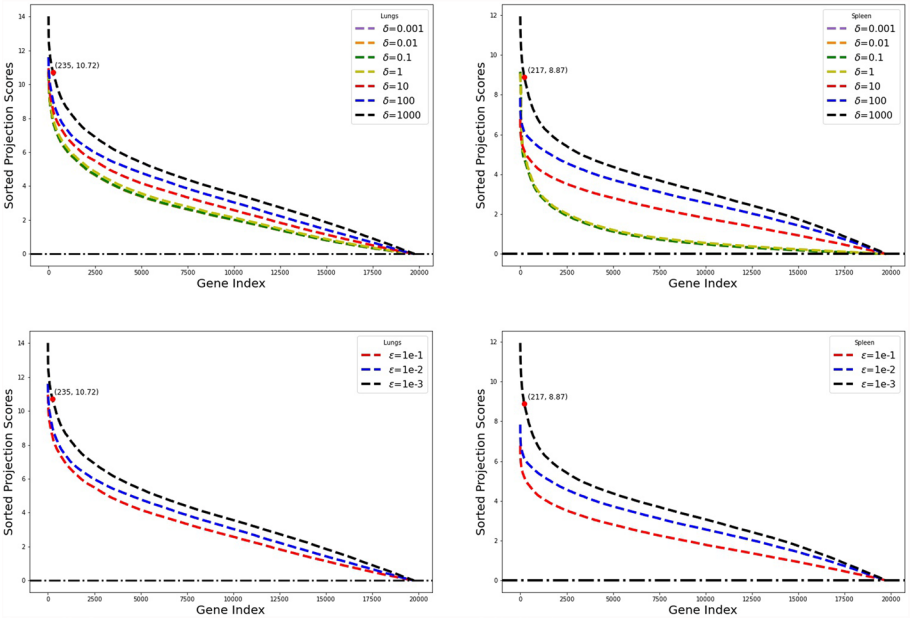

Fig. 2 Projection scores for the lungs (L) and spleen (R) datasets corresponding to different δ and ϵ values with $M := D_{\epsilon,1}(\mathrm{z})^2$

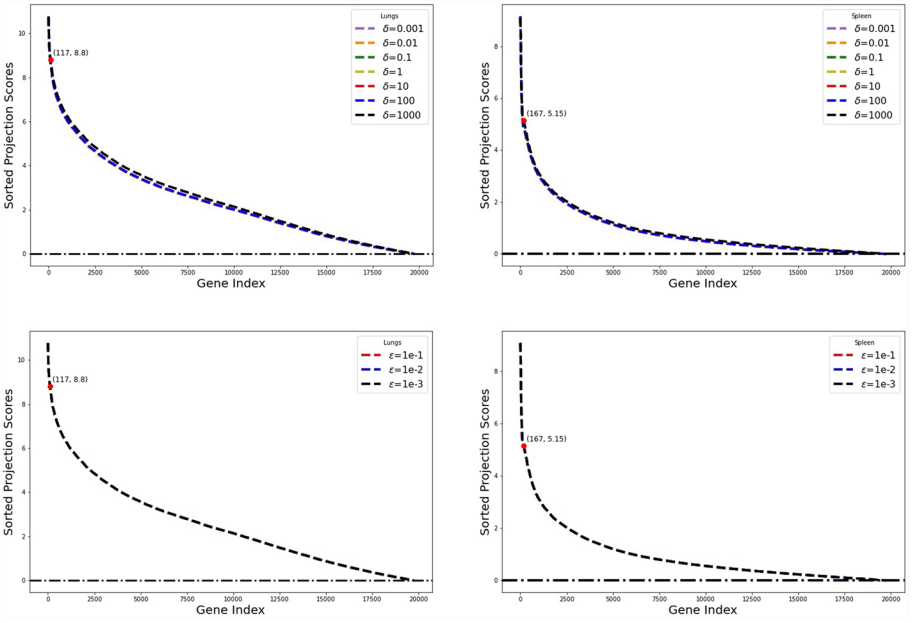

Fig. 3 Projection scores for the lungs (L) and spleen (R) datasets corresponding to different δ and ϵ values with $M := D_{\epsilon,1}(\mathrm{z})$

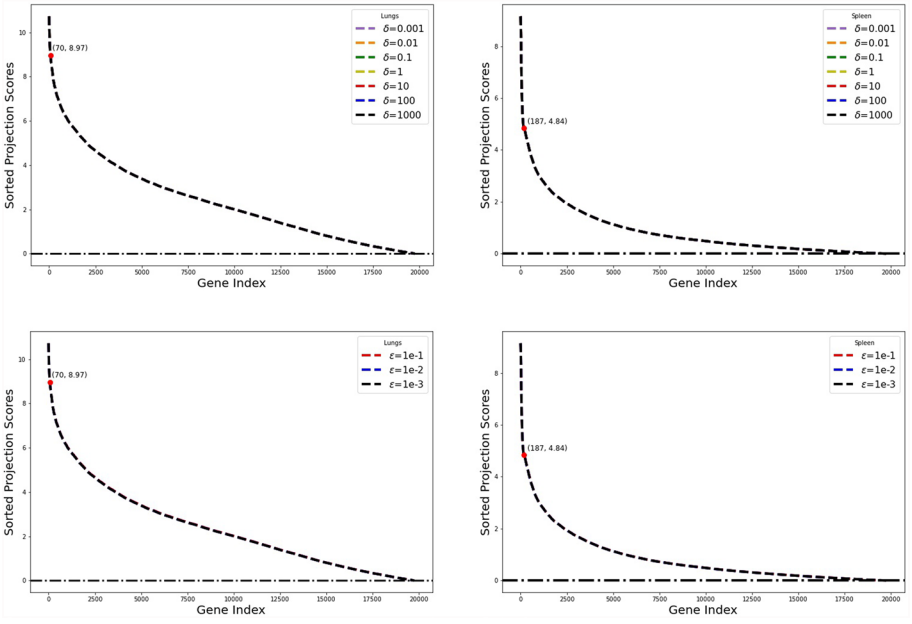

Fig. 4 Projection scores for the lungs (L) and spleen (R) datasets corresponding to different δ and ϵ values with $M := D_{\epsilon,2}(\mathrm{z})$

of Figs. 3 and 4. The projection scores corresponding to the top 235 or 217 most informative genes, in both tissues, are to the left of the maximum curvatures.

The gene expression profiles selected from both tissues are exclusive to the datasets. Supplemental Figure 1 (S1) and Supplemental Figure 1 (S2) show the top 50 most informative gene expression profiles corresponding to the lungs and spleen datasets, respectively. The heatmaps display the expression levels of selected genes across different

experimental conditions, arranged in partitions, with the color bars representing expression levels.

Specifically, the rows of S1 and S2 correspond to the selected genes, while the columns are divided into partitions corresponding to experimental conditions (controls, 12 h, 24 h, 48 h, and 120 h) after infection by Schu4 and LVS. The first partition represents controls, which serve as a baseline for both Schu4 and LVS. Partitions 2 to 5 represent time points (12 h, 24 h, 48 h, and 120 h) after Schu4 infection. The same order is repeated for the LVS strain. Notably, the red signals in the left-most columns (controls) of Figures S1 and S2 highlight genes that are highly expressed in the baseline conditions. These high-expression genes may be involved in essential or constitutive biological processes. Differences in expression intensity across partitions suggest dynamic host responses to infection over time. Some genes display early or sustained upregulation (e.g., red signals at specific time points), while others remain stable or show downregulation. Figure S1 shows the expression levels are predominantly blue, indicating that most genes have lower expression values across postinfection conditions. This figure has patches of red or yellow regions that indicate isolated high-expression genes in specific partitions. Distinct patterns corresponding to the same time points between the two strains (e.g., 12 h for Schu4 vs. LVS) could indicate strain-specific effects on gene expression.

Training and validating transfer learning models with the top 235 most informative genes

In all computed examples presented in this section, we will use information from the lungs to model the host response patterns to bacterial dissemination to the spleen, by training ML models on the lungs host response data, and then validating them on the spleen host response data. Here, gene transcription patterns identified from the lungs are utilized to predict the gene transcription patterns in the spleen. We emphasize that our approach can be employed to overcome the challenges of limited labeled data.

Specifically, we will use the top 235 most informative genes from the lungs to train transfer learning models with uninfected controls and Schu4 or LVS samples as classes, then validate the trained models using the top 235 most significant genes from the spleen. Note that the lung transcription data with SMOTE-generated controls are used in the training process. However, SMOTE-generated controls in the spleen transcription data are not used as part of the test data but only for preprocessing and data integration purposes.

The model parameters/hyperparameters used in the training process are displayed in Table 1. For each method, the best set of parameters is determined by using `GridSearchCV` with $\mathrm{RepeatedStratifiedKFold(n\_splits{=}5, n\_repeats{=}10, random\_state{=}10)}$. The determined parameters are then used to train transfer learning models with the training data randomly shuffled 100 times. For the Random Forest, Logistic Regression, linear SVM, Decision Tree, KNN, AdaBoost, GradBoost, and XGBoost, the best parameters correspond to the default parameters in scikit-learn. The Naive Bayes uses 10^{-2} and 10^{-3} as variance smoothing for Schu4 and LVS datasets, respectively.

For ANN, we use `TensorFlow Keras` to define a two-layer neural network with the best parameters as $\mathrm{batch\_size} = 64$, `epochs` $= 40$, $\mathrm{optimizer} =$ "rmsprop" and $\mathrm{neurons} = 128$. The trained networks consist of two layers of neurons, in addition to the input layer. The hidden layer has 128 neurons while the output layer has one neuron.

Table 1 Parameters used for ML training

Method	Parameters
Random Forest	$\mathrm{max\_features} : [\mathrm{sqrt}, \log 2]$, $\mathrm{n\_estimators} : [100, 200, 50, 300]$ $criterion : [gini, entropy]$
Logistic Regression	$\mathrm{C} : [1, 10, 100, 0.1, 0.01]$, $\mathrm{solver} : [\mathrm{lbfgs}, \mathrm{liblinear}]$
SVM	$\mathrm{C} : [1, 50, 10, 100, 10^{-2}, 10^{-3}, 10^{-1}]$, $\mathrm{kernel} : [\mathrm{linear}]$
Decision Tree	$\mathrm{max\_depth} : [2, 3, 5, 10]$, $\mathrm{min\_samples\_leaf} : [1, 5, 10, 15], criterion : [gini, entropy]$
Naive Bayes	$\mathrm{var\_smooting} : [1, 10, 10^{-2}, 10^{-3}, 10^{-1}]$
KNN	$\mathrm{n\_neighbors} : [5, 3, 7, 10]$
AdaBoost	$\mathrm{n\_estimators} : [50, 100, 200, 300], learning\_rate : [1, 0.1, 0.01, 0.001]$
GradBoost	$\mathrm{n\_estimators} : [100, 200, 300]$, $\mathrm{learning\_rate} : [0.1, 0.01, 0.001, 1]$, $max\_depth : [3, 4, 2, 5]$
XGBoost	$\mathrm{eta} : [0.3, 0.2, 0.1, 0.01, 0.5, 0.001]$, $\mathrm{max\_depth} : [6, 4, 5, 7]$
ANN	$\mathrm{batch\_size} : [64, 32]$, $\mathrm{epochs} : [40, 60]$, $optimizer : [rmsprop, adam]$, $neurons : [128, 32, 64]$
ℓ_1-WNPSVM	$\delta : [10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100, 1000], \epsilon : [10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}]$

The activation functions of the hidden and output layers are "ReLu" and "sigmoid", respectively.

The best parameters for the ℓ_1-WNPSVM are $\delta = 10^{-1}$ and $\epsilon = 10^{-2}$ with $\mathtt{tol} = 10^{-6}$ in Algorithm 1. We will compare the performance of ℓ_1-WNPSVM with the regularization matrices, $D_{\epsilon,1}(\mathrm{z})^2$, $D_{\epsilon,q}(\mathrm{z})$, $q \in \{0.1, 1, 2\}$, and $\mathcal{L}_i$, $i \in \{1, 2\}$, with

$$\mathcal{L}_1 := \widetilde{\mathcal{L}}_1^T \widetilde{\mathcal{L}}_1 + \omega I \quad \text{and} \quad \mathcal{L}_2 := \widetilde{\mathcal{L}}_2^T \widetilde{\mathcal{L}}_2 + \omega I, \tag{25}$$

where $\omega > 0$,

$$\widetilde{\mathcal{L}}_1 = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}, \quad \text{and} \quad \widetilde{\mathcal{L}}_2 = \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \\ & & & 1 \end{bmatrix}.$$

The SPD matrices $\mathcal{L}_1$ and $\mathcal{L}_2$ are tridiagonal matrices that are very popular in image processing applications. We will use $\omega = 3$ in all computed examples. This choice of ω is consistent with the experiments presented in [37]. The datasets are standardized by subtracting the mean and dividing by the standard deviation.

Discussion

Here we discuss the performance of the methods presented in Table 2. Four different examples considered below present the best and worst performing models, based on (Avg.) Bal. Acc. score. The balanced accuracy is defined in terms of the evaluation factors of the confusion matrix, i.e.,

$$\text{Balanced Accuracy} := \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right),$$

where the True Negative (TN) denotes the number of uninfected controls correctly classified, True Positive (TP) denotes the number of Schu4 or LVS samples correctly classified, False positive (FN) denotes the number of incorrectly classified Schu4 or LVS

Table 2 Results showing the performance of the methods with Avg. Bal. Acc. scores

Train (Lungs)	Test (Spleen)	Method	TN	TP	FP	FN	Bal. Acc. %	Avg. Bal. (%) Acc. ± std
Schu4	Schu4	Random Forest	6	15	0	9	81.25	81.67 ± 0.01
		Logistic Regression	6	16	0	8	83.33	83.33 ± 0.00
		SVM	6	18	0	6	87.50	87.50 ± 0.00
		Decision Tree	4	13	2	11	60.42	60.42 ± 0.00
		Naive Bayes	6	11	0	13	72.92	72.92 ± 0.00
		KNN	6	15	0	9	81.25	81.25 ± 0.00
		AdaBoost	6	19	0	5	89.58	89.58 ± 0.00
		GradientBoost	6	17	0	7	85.42	85.42 ± 0.00
		XGBoost	6	12	0	12	75.00	75.00 ± 0.00
		ANN	6	16	0	8	83.33	83.08 ± 0.02
		ℓ_1-WNPSVM ($M := \mathcal{L}_1$)	6	20	0	4	91.67	91.67 ± 0.00
		ℓ_1-WNPSVM ($M := \mathcal{L}_2$)	6	20	0	4	91.67	91.67 ± 0.00
		ℓ_1-WNPSVM ($M := I$)	5	22	1	2	87.50	87.50 ± 0.00
		ℓ_1-WNPSVM ($M := D_{\epsilon,1}(\mathrm{z})^2$)	4	21	2	3	77.08	77.08 ± 0.00
		ℓ_1-WNPSVM ($M := D_{\epsilon,1}(\mathrm{z})$)	5	22	1	2	87.50	87.50 ± 0.00
		ℓ_1-WNPSVM ($M := D_{\epsilon,0.1}(\mathrm{z})$)	4	21	2	3	77.08	77.08 ± 0.00
	LVS	Random Forest	6	17	0	7	85.42	86.00 ± 0.01
		Logistic Regression	6	15	0	9	81.25	81.25 ± 0.00
		SVM	6	17	0	7	85.42	85.42 ± 0.00
		Decision Tree	4	12	2	12	58.33	58.33 ± 0.00
		Naive Bayes	6	14	0	10	79.17	79.17 ± 0.00
		KNN	6	15	0	9	81.25	81.25 ± 0.00
		AdaBoost	6	18	0	6	87.50	87.50 ± 0.00
		GradientBoost	6	16	0	8	83.33	83.33 ± 0.00
		XGBoost	6	12	0	12	75.00	75.00 ± 0.00
		ANN	6	16	0	8	83.33	82.52 ± 0.02
		ℓ_1-WNPSVM ($M := \mathcal{L}_1$)	6	23	0	1	97.92	97.92 ± 0.00
		ℓ_1-WNPSVM ($M := \mathcal{L}_2$)	6	23	0	1	97.92	97.92 ± 0.00
		ℓ_1-WNPSVM ($M := I$)	6	23	0	1	97.92	97.92 ± 0.00
		ℓ_1-WNPSVM ($M := D_{\epsilon,1}(\mathrm{z})^2$)	6	23	0	1	97.92	97.92 ± 0.00
		ℓ_1-WNPSVM ($M := D_{\epsilon,1}(\mathrm{z})$)	6	23	0	1	97.92	97.92 ± 0.00
		ℓ_1-WNPSVM ($M := D_{\epsilon,0.1}(\mathrm{z})$)	6	23	0	1	97.92	97.92 ± 0.00

samples, and False Negative (FP) denotes the number of uninfected controls incorrectly classified. The Bal. Acc. is often considered to be well-suited for datasets with imbalanced classes.

Figure 5 shows the sorted balanced accuracy scores, for each method, over 100 different models validated on the test (spleen) data. The Average Balanced Accuracy (Avg. Bal. Acc.) scores of the methods in Fig. 5 are reported in Table 2. The factors, TP, TN, FN and FP, including Bal. Acc. scores are also reported in Table 2 for transfer learning models trained with no sample shuffles. For reproducibility of the results presented therein, we use $\mathrm{random\_state} = 10$.

Example 0.1 Here, transfer learning models are trained and tested on the lungs and spleen transcriptional datasets, respectively, with uninfected controls and Schu4 samples as classes. Table 2 shows that the Decision Tree is the least-performing model with an (Avg.) Bal. Acc. score of 60.42% while the ℓ_1-WNPSVM with $\mathcal{L}_i$, $i \in \{1, 2\}$, in (25) yield the highest (Avg.) Bal. Acc. score of 91.67%.

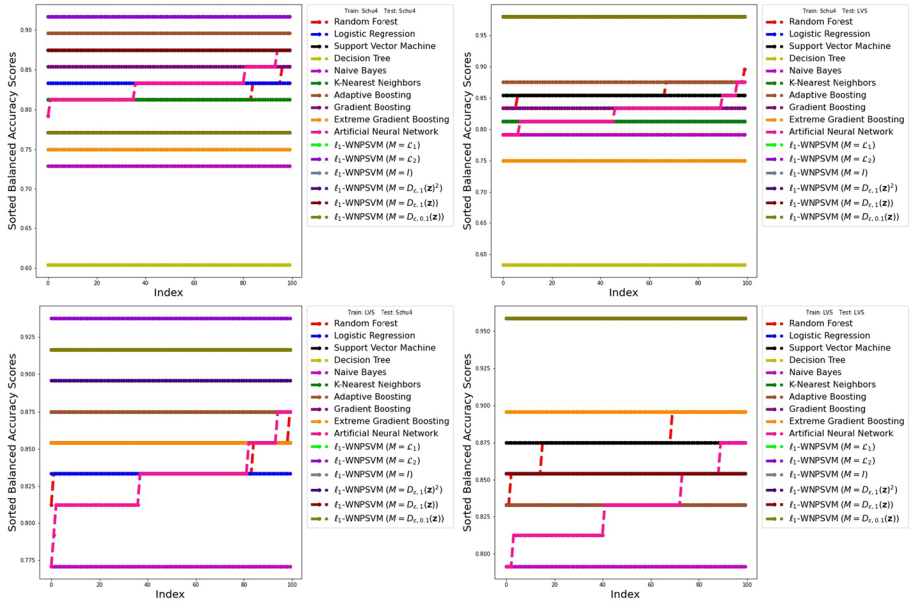

Fig. 5 Sorted balanced accuracy scores of the methods over 100 different models validated on the test data

Example 0.2 This example trains transfer learning models on the lungs transcriptional data with uninfected controls and Schu4 samples as classes. The trained models are tested on the uninfected controls and LVS samples from the spleen. Table 2 shows that the Decision Tree performs the least with an (Avg.) Bal. Acc. score of 58.33%, while the ℓ_1-WNPSVM with the regularization matrices, $D_{\epsilon,1}(\mathrm{z})^2$, $D_{\epsilon,q}(\mathrm{z})$, $q \in \{0.1, 1, 2\}$, and $\mathcal{L}_i$, $i \in \{1, 2\}$, perform the best with an (Avg.) Bal. Acc. score of 97.92%.

Example 0.3 In this example, transfer learning models are built on the lungs transcriptional data with uninfected controls and LVS samples as classes. The trained models are applied to classify uninfected controls and Schu4 samples from the spleen. Table 3 shows that the Naive Bayes performs the worst with the least (Avg.) Bal. Acc. score of 77.08% while the ℓ_1-WNPSVM with the regularization matrix $\mathcal{L}_2$ is the best method with the highest (Avg.) Bal. Acc. score of 93.75%.

Example 0.4 This example trains and tests transfer learning models with uninfected controls and LVS samples as classes. Table 3 shows that the Naive Bayes results in the least performing model with an (Avg.) Bal. Acc. score of 79.17% while the ℓ_1-WNPSVM with the regularization matrices, $D_{\epsilon,1}(\mathrm{z})^2$, $D_{\epsilon,0.1}(\mathrm{z})$, and $\mathcal{L}_i$, $i \in \{1, 2\}$, give the highest (Avg.) Bal. Acc. score of 95.83%.

Overall, the ℓ_1-WNPSVM with $M := \mathcal{L}_2$ is the best method in all computed examples. This illustrates the potential superiority of regularizing (1) with a weighted ℓ_2-norm over other choices described above. We remark that the Python codes for implementing the methods are available on GitHub: https://github.com/uougwu/WNPSVM/blob/main/wnp-svm/demos/WNP_SVM.ipynb

Figure 6 displays the performance of the transfer learning models using the top 50 gene expression profiles shown in Figures S1 ans S2. Notably, the performance of the models is comparable to those trained using the top 235 most relevant gene expression profiles. Consistently, the ℓ_1-WNPSVM methods outperform traditional ML algorithms.

Table 3 Results showing the performance of the methods with Avg. Bal. Acc. scores

Train (Lungs)	Test (Spleen)	Method	TN	TP	FP	FN	Bal. Acc. %	Avg. Bal. (%) Acc. ± std
LVS	Schu4	Random Forest	6	16	0	8	83.33	83.67 ± 0.01
		Logistic Regression	6	16	0	8	83.33	83.33 ± 0.00
		SVM	6	18	0	6	87.50	87.50 ± 0.00
		Decision Tree	6	17	0	7	85.42	85.42 ± 0.00
		Naive Bayes	6	13	0	11	77.08	77.08 ± 0.00
		KNN	6	17	0	7	85.42	85.42 ± 0.00
		AdaBoost	6	18	0	6	87.50	87.50 ± 0.00
		GradientBoost	6	17	0	7	85.42	85.42 ± 0.00
		XGBoost	6	17	0	7	85.42	85.42 ± 0.00
		ANN	6	16	0	8	83.33	83.00 ± 0.02
		ℓ_1-WNPSVM ($M := \mathcal{L}_1$)	6	20	0	4	91.67	91.67 ± 0.00
		ℓ_1-WNPSVM ($M := \mathcal{L}_2$)	6	21	0	3	93.75	93.75 ± 0.00
		ℓ_1-WNPSVM ($M := I$)	5	24	1	0	91.67	91.67 ± 0.00
		ℓ_1-WNPSVM ($M := D_{\epsilon,1}(\mathrm{z})^2$)	5	23	1	1	89.58	89.58 ± 0.00
		ℓ_1-WNPSVM ($M := D_{\epsilon,1}(\mathrm{z})$)	5	24	1	0	91.67	91.67 ± 0.00
		ℓ_1-WNPSVM ($M := D_{\epsilon,0.1}(\mathrm{z})$)	5	24	1	0	91.67	91.67 ± 0.00
	LVS	Random Forest	6	18	0	6	87.50	87.79 ± 0.01
		Logistic Regression	6	16	0	8	83.33	83.33 ± 0.00
		SVM	6	18	0	6	87.50	87.50 ± 0.00
		Decision Tree	6	19	0	5	89.58	89.58 ± 0.00
		Naive Bayes	6	14	0	10	79.17	79.17 ± 0.00
		KNN	6	17	0	7	85.42	85.42 ± 0.00
		AdaBoost	6	16	0	8	83.33	83.33 ± 0.00
		GradientBoost	6	17	0	7	85.42	85.42 ± 0.00
		XGBoost	6	19	0	5	89.58	89.58 ± 0.00
		ANN	6	15	0	9	81.25	83.21 ± 0.02
		ℓ_1-WNPSVM ($M := \mathcal{L}_1$)	6	22	0	2	95.83	95.83 ± 0.00
		ℓ_1-WNPSVM ($M := \mathcal{L}_2$)	6	22	0	2	95.83	95.83 ± 0.00
		ℓ_1-WNPSVM ($M := I$)	5	21	1	3	85.42	85.42 ± 0.00
		ℓ_1-WNPSVM ($M := D_{\epsilon,1}(\mathrm{z})^2$)	6	22	0	2	95.83	95.83 ± 0.00
		ℓ_1-WNPSVM ($M := D_{\epsilon,1}(\mathrm{z})$)	5	21	1	3	85.42	85.42 ± 0.00
		ℓ_1-WNPSVM ($M := D_{\epsilon,0.1}(\mathrm{z})$)	6	22	0	2	95.83	95.83 ± 0.00

It is evident from Tables 2 and 3, as well as Fig. 6, that the ℓ_1-WNPSVM methods (with $M \neq I$) often result in higher balanced accuracy than ℓ_1-WNPSVM ($M = I$) method proposed in [4]. These results underscore the significance of introducing a weighted regularization in the NPSVM framework.

The next section discusses pathway discovery with the selected top informative genes. Of the 235 top genes associated with the lungs and spleen datasets, only 234 gene identifiers from the spleen dataset are available for pathway analysis.

Transcriptomic pathway analysis

This section presents pathway analysis, also known as *functional enrichment analysis*, of available top 235 and 234 most informative gene identifiers in the lungs and spleen, respectively, by using Reactome,[3] a curated database of pathways and reactions in human biology. Our goal is to illustrate that the proposed gene selection approach can

[3] https://reactome.org/

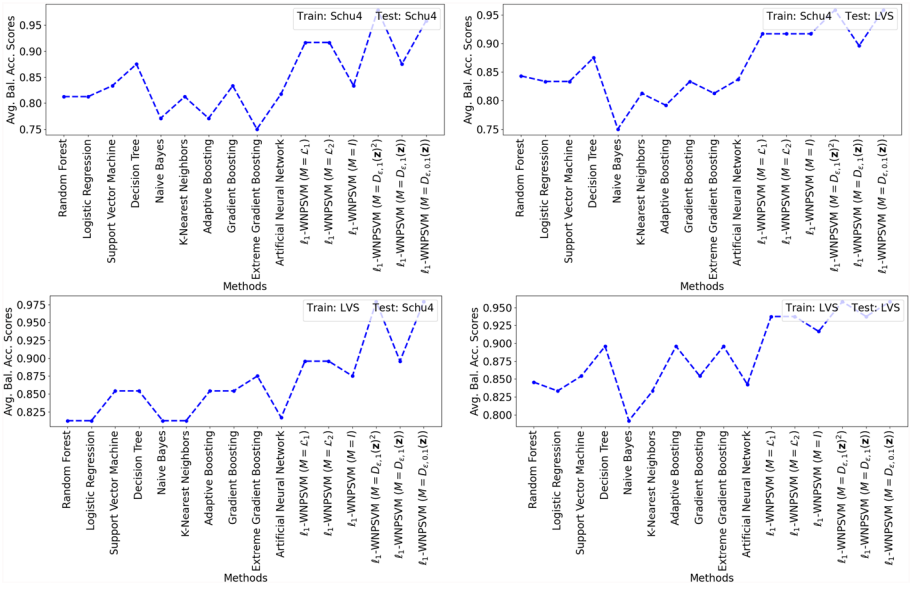

Fig. 6 Illustrating the performance of the methods using the top 50 most informative genes shown in Figures S1 and S2

accurately identify groups of related host genes or features that are involved in relevant biological processes such as immune response.

Reactome uses *hypergeometric distribution* to determine pathways that are overrepresented (enriched) in a submitted list of identifiers also referred to as *gene name list*. This overrepresentation analysis examines if a given feature list contains more genes for pathway X than normal by chance, then assigns a probability score, p-value, that is corrected for False Discovery Rate (FDR) using the Benjamani-Hochberg method.

In our overrepresentation analysis, all mice identifiers were converted to their human equivalent with no molecular interactors included. Reactome reports show that of the 235 most informative identifiers from the lungs data, 55 identifiers were found, and they are mapped to 67 Reactome entities (UniPort Id), with 503 pathways hitting at least one of the identifiers. As for the spleen data, 75 out of 234 identifiers were found and mapped to 85 Reactome entities while 399 pathways were hit by at least one of the 75 identifiers. Moreover, 180 identifiers from the lungs data and 159 identifiers from the spleen data were neither found nor mapped to any entity in Reactome.

Tables 4 and 5 present the top 10 most significant pathways in the lungs and spleen, respectively, including the associated identifiers. The order of significance of the pathways is based on their p-values. Pathways with the smallest p-values, in that order, are considered to represent the most relevant features or genes. In Table 4, Pathways 1, 2, 3, 7, 8, and 10, are categorized as signal transduction pathways while Pathways 4, 5, 6, and 9, are disease (cancer) pathways. Similarly, in Table 5, Pathways 1, 2, 4, 5, and 8, are involved in gene expression (transcription), while Pathway 3 is an immune system pathway. Pathways 6, 7, 9, and 10, are disease (cancer) pathways. Collectively, these pathways indicate a significant host response to infection.

We remark that the top pathways in the spleen are dominated by RUNX2, which is known to regulate the expression of genes implicated in cell migration during normal development and bone metastasis of the breast cancer cells. RUNX2 also mediates the transcription of the MMP13 gene involved in the migration of innate immune system

Table 4 Top 10 pathways associated with the lungs data

Pathway Name	Identifiers
1.Calcitonin-like ligand receptors	Calca
2.EGFR Transactivation by Gastrin	Hras1, Prkca
3.Activated NTRK2 signals through RAS	Hras1, Ntrk2
4.RAS GTPase cycle mutants	Hras1
5.Signaling by RAS GTPase mutants	Hras1
6.Signaling by RAS GAP mutants	Hras1
7.Activated NTRK2 signals through FRS2 and FRS3	Hras1, Ntrk2
8.Gastrin-CREB signalling pathway via PKC and MAPK	Hras1, Prkca
9.RAS signaling downstream of NF1 loss-of-function variants	Hras1
10.SOS-mediated signalling	Hras1

Table 5 Top 10 pathways associated with the spleen data

Pathway Name	Identifiers
1.RUNX2 regulates genes involved in differentiation of myeloid cells	Runx2
2. RUNX2 regulates chondrocyte maturation	Runx2
3. LRR FLII-interacting protein 1 (LRRFIP1) activates type I IFN production	Catnb, Lrrfip1
4. RUNX1 regulates transcription of genes involved in differentiation of myeloid cells	Runx2
5. RUNX2 regulates genes involved in cell migration	Runx2
6. Defective Mismatch Repair Associated With MLH1	Pms1
7. Defective Mismatch Repair Associated With PMS2	Pms1
8. YAP1- and WWTR1 (TAZ)-stimulated gene expression	Runx2
9. TGFBR1 LBD Mutants in Cancer	Tgfbr1
10.TGFBR2 Kinase Domain Mutants in Cancer	Tgfbr1

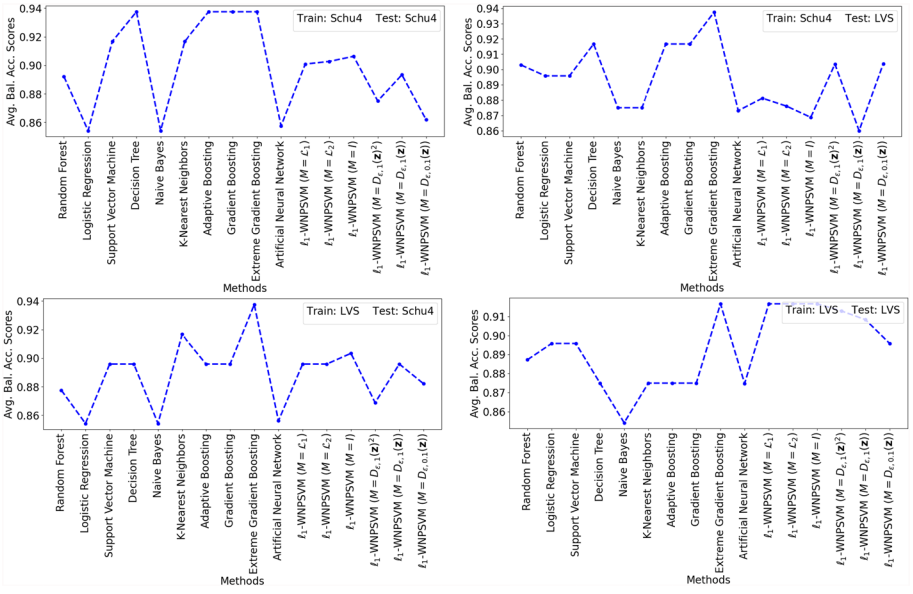

Fig. 7 Illustrating the performance of the methods using the 3 most dominant top gene identifiers for the lungs and spleen datasets in Tables 4 and 5, respectively, with the average balanced accuracy computed over 100 random initializations

cells in response to injury. On the other hand, the top 10 pathways in the lungs are dominated by Hras1, a cell signaling gene.

Figure 7 illustrates the performance of the transfer learning models trained on the 3 most dominant top genes in the lungs associated with signal transduction pathways (cf.

Table 4) and validated on the spleen dataset with immune system response-related genes (cf. Table 5). Specifically, the models are trained on the lungs dataset using the top genes: *Hras1, Prkca, Ntrk2*, from Table 5 along with the model parameters utilized for the models in Tables 2 and 3. The trained models are validated on the spleen dataset using the top 3 genes: *Runx2, Catnb, Lrrfip1*. Figure 7 shows that the performance of most traditional ML methods improved significantly using the 3 most relevant gene identifiers associated with lungs and spleen pathways, achieving a maximum accuracy of 93.75%, while those of the ℓ_1-WNPSVM methods hovers around 91%.

Conclusions

This paper aims to identify a small set of genes or features that are indicative of respiratory infection and dissemination to secondary sites. This work utilized a standardized infection model of disease and two related bacterial strains of Francisella tularensis - Schu4 and Live Vaccine Strain (LVS), with different levels of virulence that affords the opportunity to identify classifiers of infection and dissemination. Further, the use of host transcriptional data from the lungs and spleen tissues of genetically identical mice infected via the respiratory route with Schu4 and LVS allows the utilization of the identified small group of genes to perform binary classification tasks.

To achieve this objective, we propose and apply optimization and Machine Learning (ML) algorithms to analyze gene transcription datasets from the lungs and spleen tissues. We utilize weighted ℓ_1-norm Generalized Eigenvalue-type Problems (ℓ_1-WGEPs) to integrate the two datasets. Then dimensionally reduce their sizes by identifying the top k most informative host gene response profiles in both tissues associated with bacterial infection and dissemination.

The optimal solutions of ℓ_1-WGEPs determine the best low-dimensional subspaces to project the datasets. The projection scores corresponding to a chosen projection direction are ranked and utilized to select the top k most informative genes as biomarkers.

Consequently, the selected biomarkers from the lungs data are used to train non-parallel support vector machines incorporating transfer learning. The training process uses uninfected controls and Schu4 or LVS samples from the lungs as classes, and analogously, validate the models using the spleen data.

The performance of baseline ML algorithms, such as ANN, XGBoost, GradBoost, AdaBoost, KNN, SVM, Naive Bayes, Random Forest, Logistic Regression, and Decision Tree, are compared with our Weighted ℓ_1-norm Non-Parallel Proximal Support Vector Machine (ℓ_1-WNPSVM). Differently from these methods, the ℓ_1-WNPSVM uses two non-parallel separating hyperplanes for binary classification. The average balanced accuracy scores of the methods over 100 folds are reported. The ℓ_1-WNPSVM is the best method with a maximum (average) balanced accuracy score of 97%.

Furthermore, pathways analysis of the most significant genes in lung and spleen tissues are examined for relevant biological pathways that may aid in not only biomarkers of infection, dissemination and disease progression, but also provide information about the host response to infection that can inform the development of host-directed therapeutics. Our analysis can lead to a better understanding of infectious disease progression as well as the development of biomarkers that are able to predict clinical outcomes of disease during treatment.

We remark that while the top informative genes selected in this study are biologically relevant and contribute to highly accurate classification of bacterial strains and controls, their precise roles in drug discovery, target therapeutics and development of diagnostics require further elucidation and validation through rigorous functional studies. Future research will focus on addressing these as well as exploring the adaptability of the proposed methods to other bacterial or viral infection studies.

Abbreviations

LVS	Live Vaccine Strain
WNPSVM	Weighted Non-Parallel Support Vector Machine
WGEP	Weighted Generalized Eigenvalue Problem
GEPSVM	Generalized Eigenvalue Proximal Support Vector Machine
ANN	Artificial Neural Networks
XGBoost	Extreme Gradient Boosting
AdaBoost	Adaptive Boosting
GradBoost	Gradient Boosting
KNN	K-Nearest Neighbors
SVM	Support Vector Machine
ML	Machine Learning
SMOTE	Synthetic Minority Oversampling Technique
TN	True Negative
TP	True Positive
FN	False Negative
FP	False Positive

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-025-06221-1.

Supplementary file 1 (pdf 14139 KB)

Acknowledgements
Not applicable

Author contributions
U.O.U. and M.K. conceived the original idea. U.O.U. developed the model architecture, implemented the code, conducted the experiments, prepared all figures, analyzed the results, and wrote the original manuscript. R.A.S. and M.K. provided the dataset, domain expertise, and funding. U.O.U. R.A.S. and M.K. reviewed and edited the manuscript. All authors discussed the results and contributed to the final manuscript.

Funding
Not applicable

Availability of data and materials
The datasets (lungs and spleen) utilized in this article are publicly available through the Gene Expression Omnibus (GEO) database using accession number GSE22203 (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE22203). This dataset uses F. tularensis Schu4 and LVS strains to understand how infection with virulent F. tularensis leads to rapid bacterial dissemination and lethal infection.

Delarations

Ethics approval and consent to participate
Not applicable

Consent for publication
Not applicable

Competing interests
Not applicable

Received: 13 November 2024 / Accepted: 7 July 2025
Published online: 17 February 2026

References

1. Mahendran N, Vincent P, Srinivasan K, Chang CY. Machine learning based computational gene selection models: a survey, performance evaluation, open issues, and future research directions. Front Genet. 2020;11: 603808.
2. Li CN, Shao YH, Deng NY. Robust ℓ_1-norm non-parallel proximal support vector machine. Optimization. 2016;65:169–83.

3. Sun XQ, Chen YJ, Shao YH, Li CN, Wang CH. Robust nonparallel proximal support vector machine with ℓ_p-norm regularization. IEEE Access. 2018;6:20334–47. https://doi.org/10.1109/access.2018.2822546.
4. Yan H, Ye Q, Zhang T, Yu DJ, Xu Y. ℓ_1-norm GEPSVM classifier based on an effective iterative algorithm for classification. Neural Proc Lett. 2017;1–26.
5. Viola M, Sangiovanni W, Toraldo G, Guarracino MR. Semi-supervised generalized eigenvalues classification. Ann Oper Res. 2019;276:249–66.
6. Belkin M, Niyogi P, Sindhwani V. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J Mach Learn Res. 2006;7:2399–434.
7. Guarracino MR, Cifarelli C, Seref O, Pardalos PM. A classification method based on generalized eigenvalue problems. Optim Methods Softw. 2007;22:73–81.
8. Shao YH, Deng NY, Chen WJ, Wang Z. Improved generalized eigenvalue proximal support vector machine. IEEE Signal Process Lett. 2013;20:213–6. https://doi.org/10.1109/LSP.2012.2216874.
9. Viola M, Sangiovanni M, Toraldo G, Guarracino MR. A generalized eigenvalues classifier with embedded feature selection. Optim Lett. 2017;11:299–311. https://doi.org/10.1007/s11590-015-0955-7.
10. Guarracino MR, Sangiovanni M, Severino G, Toraldo G, Viola M. On the regularization of generalized eigenvalues classifiers. Proc AIP Conf. 2016;1776(1):040005.
11. Mangasarian OL, Wild EW. Multisurface proximal support vector machine classification via generalized eigenvalues. IEEE Trans Pattern Anal Mach Intell. 2006;28:69–74.
12. Chen Y, Yang Z. Generalized eigenvalue proximal support vector machine for functional data classification. Symmetry. 2021;13:833. https://doi.org/10.3390/sym13050833.
13. Guo T. Generalized nonparallel proximal support vector machine with applications on ship detection using satellite images. In: Signal and information processing: networking and computers. Springer; 2021. pp. 216–23.
14. Xanthopoulos P, Guarracino MR, Pardalos PM. Robust generalized eigenvalue classifier with ellipsoidal uncertainty Ann. Oper Res. 2014;216:327–42.
15. Guarracino MR, Cuciniello S, Pardalos PM. Classification and characterization of gene expression data with generalized eigenvalues. J Optim Theory Appl. 2009;141:533–45.
16. Guarracino MR, Cifarelli C, Seref O, Pardalos PM, A parallel classification method for genomic and proteomic problems. In: 20th International conference on advanced information networking and applications, IEEE Computer Society, 1 2006.
17. Guarracino MR, Cuciniello S, Feminiano D, A parallel classification and feature reduction method for biomedical applications, In: PPAM, 2007, pp. 1210–1219.
18. Guarracino MR, Xanthopoulos P, Pyrgiotakis G, Tomaino V, Moudgil BM, Pardalos PM. Classification of cancer cell death with spectral dimensionality reduction and generalized eigenvalues. Artif Intell Med. 2011;53:119–25.
19. Sun S, Xie X, Dong C. Multiview learning with generalized eigenvalue proximal support vector machines. IEEE Trans Cybernet. 2018;49:688–97.
20. Khan S, AlQahtani SA, Noor S, Ahmad N. PSSM-Sumo: deep learning based intelligent model for prediction of sumoylation sites using discriminative features. BMC Bioinf. 2024;25:284.
21. Khan S, Khan M, Iqbal N, Rahman MA, Karim MK. Deep-PiRNA: Bi-layered prediction model for PIWI-interacting RNA using discriminative features. Comput Mater Contin. 2022;72:2243–58.
22. Parlett BN. The symmetric eigenvalue problem. Philadelphia: SIAM; 1998. p. 357.
23. Tikhonov AN, Arsen VY. Solutions of ill-posed problems. New York, Wiley 1977.
24. Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995;20:273–97.
25. Jain AK, Mao J, Mohiuddin KM. Artificial neural networks: a tutorial. Computer. 1996;29:31–44. https://doi.org/10.1109/2.485891.
26. Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci. 1997;55:119–39.
27. Friedman JH. Greedy function approximation: a gradient boosting machine. Ann Stat. 2001;29:1189–232.
28. Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining 2016;785-794.
29. Guo G, Wang H, Bell D, Bi Y, Greer K. KNN model-based approach in classification. In On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, 3–7, Proceedings, Springer Berlin Heidelberg, (2003), pp. 986-996.
30. Peng C, Lee KL, Ingersoll GM. An introduction to logistic regression analysis and reporting. J Educ Res. 2002;96:3–14.
31. Breiman RF. Mach Learn. 2001;45(1):5–32.
32. Zhang H. The optimality of Naive Bayes. FLAIRS Proc; 2004.
33. Breiman L, Friedman JH, Olshen RA, Stone CJ, Classification and regression trees. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software 1984.
34. Kingry LC, Troyer RM, Marlenee NL, Bielefeldt-Ohmann H, Bowen RA, Schenkel AR, Dow SW, Slayden RA. Genetic identification of unique immunological responses in mice infected with virulent and attenuated Francisella tularensis. Microbes Infect. 2011;13:261–75.
35. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority oversampling technique, J Artif Intell Res 2002.
36. Satopaa V, Albrecht J, Irwin D, Raghavan B. Finding a"kneedle"in a haystack: Detecting knee points in system behavior. In: 2011 31st international conference on distributed computing systems workshops, IEEE, pp. 166-171, 2011.
37. Reichel L, Ugwu UO. Weighted tensor Golub-Kahan-Tikhonov-type methods applied to image processing using a t-product. J Comput Appl Math. 2022;415: 114488.

Publisher's Note