

1 **Online supplement:**

2
3 Title: Machine learning driven identification of the gene-expression signature associated
4 with a persistent multiple organ dysfunction trajectory in critical illness.

5
6 Authors: Mihir R. Atreya^{1,2*}, Shayantan Banerjee^{3*}, Andrew J. Lautz^{1,2}, Matthew N.
7 Alder^{1,2}, Brian M. Varisco^{1,2}, Hector R. Wong^{1,2}, Jennifer A. Muszynski^{4,5}, Mark W.
8 Hall^{4,5}, L. Nelson Sanchez-Pinto^{6,7}, and Rishikesan Kamaleswaran^{8,9} for the Genomics
9 of Pediatric Septic Shock Investigators.

10
11 Author affiliations:

- 12 1. Division of Critical Care Medicine, Cincinnati Children's Hospital Medical Center
13 and Cincinnati Children's Research Foundation, Cincinnati, 45229, OH, USA.
- 14 2. Department of Pediatrics, University of Cincinnati College of Medicine,
15 Cincinnati, OH 45267, USA.
- 16 3. Department of Biotechnology, Bhupat and Jyoti Mehta School of Biosciences,
17 Indian Institute of Technology Madras, Chennai, 600 036, India.
- 18 4. Division of Critical Care Medicine, Nationwide Children's Hospital, Columbus,
19 43205, OH, USA.
- 20 5. Department of Pediatrics, Ohio State University, Columbus, 43205, OH, USA.
- 21 6. Department of Pediatrics, Northwestern University Feinberg School of Medicine,
22 Chicago, 60611, IL, USA.
- 23 7. Department of Health and Biomedical Informatics, Northwestern University
24 Feinberg School of Medicine, Chicago, 60611, IL, USA.
- 25 8. Department of Biomedical Informatics, Emory University School of Medicine,
26 Atlanta, 30322, GA, United States.
- 27 9. Department of Biomedical Engineering, Georgia Institute of Technology, Atlanta,
28 30322, GA, United States.

29
30 Corresponding author:

31 Mihir R Atreya, MD, MPH
32 Cincinnati Children's Hospital Medical Center
33 Division of Critical Care Medicine, MLC2005
34 3333 Burnet Avenue
35 Cincinnati, OH, 45229, USA
36 Tel: 513-636-1697
37 Email: Mihir.Atreya@cchmc.org

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

Table of contents:

1. Supplemental Methods.....3
2. Supplementary Tables.....8
3. Supplementary Figure Legend.....15
4. Supplementary Figures..... 16
5. References.....18

Supplementary Methods:

1. Pre-processing of gene expression data in training dataset (GSE66099).

Batch correction: Our study considered the year of measurement of the gene expression data as the batch variable. Ideally, batch corrections are possible only if the variables are not highly correlated with the outcome (MODS in our dataset).

As shown in **supplementary Table 2**, a tight correlation between the batch variable (year) and the outcome of interest is absent. Within each batch, we had measurements from multiple different groups. So, we proceeded with the batch effect removal process.

The 'sva' package in R was used to identify batch effects in our data. Although we had prior information regarding the batch variable (the year of measurement), we wanted to check if SVA could find new covariates explaining the variation in our data. The 'sv' component returned by the sva function contained the two new covariates or the potential batch effects. To check if the new surrogate variables (or SVs) are associated with the observed batch variable, a linear model is fit using the lm() in R. From **supplementary Table 3**, we can observe that the second estimated surrogate variable has a significant correlation with the batch variable. In this case, the coefficient tells us that by changing the batch variable, the value of the SV changes by 8.03, and this result is significant ($P=9e-05$). This shows that the estimated SV is associated with the batch.

2. Derivation of stable features.

The workflow adopted for our machine learning analysis is shown in **Figure 1** (main text). The entire process can be subdivided into three parts. Here, we discuss each part in detail:

A typical machine learning workflow involves dividing the available data into three groups: Train, Validation, and Test. The sample of the data used to fit the model is referred to as the training set. The validation set is used to tune the model's hyperparameters and to derive the best model configuration. The test set provides an unbiased evaluation of the best model derived from the training and the validation set.

PART A: Stratified Cross-Validation

Whenever we are provided with a limited data sample, we train and evaluate our models using Cross-Validation approach. K-fold cross-validation requires a single parameter k, which refers to the number of groups the given data sample is split into. In our case, we chose $k=5$.

The general procedure to derive the cross-validation results is as follows:

- Randomly shuffle the data
- Split the dataset into 5 equal-sized subsets
- For each subset,
- Consider one subset as a hold-out or a test set
- Take the remaining four subsets as a single training set
- Derive the best set of hyperparameters and train the model using the training set and test it on the test set.

- Calculate the evaluation metrics such as Sensitivity, Specificity, MCC, and AUC.
- The final results are the average classification metrics calculated across all five folds.

Stratified k-fold cross-validation: The derivation set (GSE66099) used to identify the set of candidate biomarkers had 46 patients with persistent MODS and 155 patients with resolving or no MODS labels. Due to the skewed class distribution, we used Stratified k-fold cross-validation instead of normal k-fold. The class distribution of the dataset was preserved in each of the train-test splits.

PART B:

Dimensionality reduction.

Scaling: We scaled our features (genes) using a popular normalization technique called Min-Max scaling. All the feature values were shifted and scaled so that ended up in the 0-1 range. Below is the formula for Min-Max scaling:

$$X = (X - X_{\min}) / (X_{\max} - X_{\min})$$

X_{\max} and X_{\min} are the maximum and minimum values of a given feature respectively.

Feature Selection (Stage I) Gene expression data is usually highly redundant and highly dimensional (containing measurements from thousands of genes). Thus, dimensionality reduction is necessary to distinguish noise from the true signal.

We used 3 feature selection techniques on our scaled derivation dataset, in addition to conventional differential expression of gene (DEG) analyses.

A. LASSO: The Least Absolute Shrinkage and Selection Operator is a powerful method that performs both regularization and feature selection simultaneously¹.

A linear regression model can be expressed as follows:

$Y_i = \beta_0 + x_{i1}\beta_1 + \dots + x_{ik}\beta_k + \epsilon_i$, where $i=1\dots n$ where Y_i is the response variable, the parameters $\beta_0, \beta_1, \dots, \beta_k$ are the regression coefficients, and we have k number of explanatory variables. The random error or ϵ_i is assumed to have 0 mean and constant variance. Assuming n samples in total, the vector notation used to represent the above formula is: $Y = X\beta + \epsilon$, where Y is the $(n \text{ by } 1)$ response vector, X is the $(n \text{ by } k)$ design matrix representing the k features, β is the $(k \text{ by } 1)$ coefficient vector and ϵ is the $(n \text{ by } 1)$ error vector.

The main goal of linear regression is to fit a straight line to several points, minimizing the squared residuals. LASSO minimizes the sum of squared residuals while placing an upper bound on the model parameters' absolute sum.

Using the formulation used by Buhlmann and van de Geer [3], we get

$\text{minimize} (\sum_{i=1}^n ||Y - X\beta||^2) / n$ subject to $\sum_{j=1}^k ||\beta_j|| \leq t$

where t is the upper bound for the sum of the coefficients. This is equivalent to solving

$(\beta)^\lambda = \text{argmi} (\sum_{i=1}^n ||Y - X\beta||^2) / n + \lambda \sum_{j=1}^k ||\beta_j||$ where $\sum_{i=1}^n ||Y - X\beta||^2 = \sum_{i=1}^n (Y_i - (X\beta)_i)^2$, $\sum_{j=1}^k ||\beta_j|| = \sum_{j=1}^k |\beta_j|$

1 and λ is the shrinkage parameter that controls the amount of penalty that must be
2 applied to the β 's. When we solve this optimization problem, some of the coefficients
3 are shrunk to zero and as a result, the features corresponding to those coefficients are
4 excluded from the model. This makes LASSO a powerful feature selection technique.

5
6 We implemented the LassoCV function from the linear_model module for feature
7 selection purposes. This function uses cross-validation to choose the best model, and
8 we used the default 5-fold cross-validation splitting strategy.

9
10 **2. MRMR:** Minimum Redundancy Maximum Relevance is a feature selection algorithm
11 to find a small subset of features by considering the correlations between the features
12 and their importance.² If two highly correlated features are also highly relevant, then
13 adding both of them would increase the model complexity. So for a set of S features,
14 the relevance between them is defined as, and the redundancy is denoted by $R =$
15 $1/(|S| \binom{|S|}{2}) \sum_{(x_i, x_j \in S)} I(x_i, x_j)$, where I is the mutual
16 information operator. The mRMR score for the set S is given by (D-R). The goal is to
17 find the subset of features S with the maximum (D-R). We used the Python wrapper
18 named "pymrmr" that was published with the original paper, and selected the top 10
19 important features using this method.

20
21 **3. Random Forests based variable importance technique:** Random forests comprise
22 several decision trees trained on a random subset of observations using a random
23 subset of features.³ No single tree sees all the features or all the samples at once, and
24 this makes it less prone to overfitting. Each tree, in turn, is a series of yes/no questions
25 based on a combination of features. At each question (or node), the tree divides into
26 two branches containing samples that are more similar to one another and different
27 from the ones in the other branch. Thus, the importance of each feature is based on
28 how "pure" (containing samples belonging to a single class) each of the branches is.

29
30 We used the RandomForestClassifier function from the ensemble module and a
31 collection of 100 estimators to derive the feature importance. We finally selected the top
32 10 ranked features for our feature pool.

33
34 We added the DEGs identified from our analysis in the training dataset to the genes
35 chosen by each of the above three feature selection strategies. This formed our pooled
36 list of features, which was then passed onto the next feature selection stage.

37
38 **Feature Selection Stage II (Recursive Feature Elimination):** Our main goal was to
39 identify a small subset of features to remove redundancy and avoid overfitting. This final
40 feature selection approach tries to remove redundant features from the pooled feature
41 set by recursively removing them and building a model on those that remain. This
42 process is also known as Recursive Feature Elimination. This ensures that our final set
43 of features obtained after this stage contributes most to the output.

44
45 The REFCV function from the feature selection module was used to implement this final
46 feature selection strategy. For each classifier implemented in our study, the RFECV

1 function was called with a 3-fold cross-validation splitting strategy, and a “roc_auc”
2 method of scoring was used as the function parameters.

3 4 **Part C: Model fitting**

5 After finding the optimal set of features from the high-dimensional gene expression
6 data, the next step was to use these features to train our model. Hyperparameter tuning
7 is a very crucial step in finding the best set of parameters for a given classifier. Grid
8 search uses an exhaustive search and evaluation strategy for a given classifier to
9 achieve this objective. It checks for every combination of hyperparameters in the grid,
10 evaluates the model based on predefined metrics, and outputs the combination that
11 gives the best results. It is a bit computationally expensive, especially if one uses a
12 cross-validated grid search technique to search for the optimal parameters in a
13 parameter grid.

14
15 The GridSearchCV function from the model_selection module with the default 3-fold
16 cross-validation strategy and a “roc_auc” scoring metric was used to search for the best
17 set of hyperparameters.

18
19 **Derivation of the final set of stable features (genes) from the cross-validation**
20 **experiments:** All the steps of the machine learning workflow discussed up to this point
21 are based on a single run of the 5-fold cross-validation experiment. We repeated this
22 process seven times, choosing a different 5-fold split every time. Hence, we had 35
23 highly relevant features that were predictive of a MODS outcome.

24
25 We used the RepeatedStratifiedKFold function from the model_selection module in
26 scikit-learn to perform our cross-validation experiments.

27
28 The fraction of times a particular feature was chosen out of the 35 runs was used to
29 rank the genes from strongest association to weakest. Genes associated with outcome
30 of interest in $\geq 80\%$ of repeated cross-fold validation experiments were chosen for
31 downstream analyses and optimization.

32
33 **Determining an optimal set of parameters using the validation dataset (E-MTAB-**
34 **10938).** Our overall goal was to derive a single classifier and test its generalizability
35 using independent test cohorts. We tuned the parameters for our classifier using an
36 independent pediatric dataset (E-MTAB-10938).

37
38 Following is the list of parameters used to define that classifier:

39
40 **The scaling technique:** Different scaling techniques were implemented to transform
41 the dataset used to validate and test the machine learning models. Three scaling
42 techniques were experimented with: Standard Scaler, Minmax Scaler, and Robust
43 Scaler.

1 **Number of top stable features:** A list of 111 stable genes was identified through
2 repeated cross-validation experiments. The tunable parameter was combination of the
3 top n genes (n=5,10,15...111).
4

5 **The sampling technique-classifier combination:** Owing to data imbalance, we
6 experimented with different undersampling (random undersampling, Repeated Edited
7 Nearest Neighbours, Cluster Centroids, Instance Hardness Threshold, NearMiss,
8 Edited Nearest Neighbours, Tomek Links, All KNN, Condensed Nearest Neighbour,
9 One-Sided Selection), and oversampling (SMOTE, Random OverSampler, ADASYN,
10 KMeans SMOTE, Borderline SMOTE, SVM SMOTE) techniques to balance the class
11 distribution better so that standard machine learning techniques can be implemented
12 directly.
13

14 Binary classifiers such as Naive Bayes, Linear Discriminant Analysis, Support Vector
15 machines, K-nearest neighbors, Decision trees, Random forests, ExtraTrees, and
16 AdaBoost were implemented to differentiate between persistent and resolving MODS.
17

18 **Classification thresholds:** Many machine learning classifiers can generate a
19 classification probability before it gets mapped to a class label. Using the default
20 threshold of 0.5 for imbalanced classification problems may lead to misleading results.
21 A simple approach is tuning the threshold to map probabilities to class labels. We
22 employ a grid search technique for the best threshold between 0 and 1 with step size
23 0.001.
24

25 The classifier built using the best parameters from 1-4 above was implemented on the
26 two independent datasets (GSE144406 and E-MTAB-5882).
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

1 **Supplementary Tables:**
 2
 3

4 **Table 1.** Characteristics of gene-expression datasets including training, validation, and
 5 test sets used in the study.
 6
 7

Type	Database name (Dataset ID)	Platform	Collection timepoint and follow-ups (if any)	Mean Age (in yrs.)	Total sample size included in analyses.
Training	GEO (GSE66099)	Affymetrix Human Genome U133 Plus 2.0 Array	Day 1 of meeting pediatric septic shock criteria.	3.6 ± 3.1	201
Validation	ArrayExpress (E-MTAB-10938)	Illumina HiSeq 4000	Within 48 hours of meeting pediatric septic shock criteria.	0.8 ± 0.5	32
Test	GEO (GSE144406)	Illumina NextSeq 500	At diagnosis of MODS, 72 hours after, and eight days later	6.8 ± 6.3	61
Test	Array Express (E-MTAB-5882)	Illumina Human HT-12 v4 Expression Beadchip	Hyperacute period within two h, 24h and 72 h of injury.	37.9 ±15.4	84

8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21

All datasets used biospecimens isolated from peripheral whole blood collected in RNA stabilization tubes.

1 **Table 2:** Number of gene expression measurements made by year for the training
2 dataset (GSE66099) by group of interest.

3

Outcome	2004	2005	2006	2007	2008	2010
Persistent MODS	2	7	5	8	5	17
Resolving or No MODS	7	40	14	23	23	50

4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41

1 **Table 3:** Results of regressing the surrogate variables returned by the sva() and the
 2 actual batch effects.
 3

Formula	Coefficients			
	Components	Estimate	Std. Error	Significance level
Surrogate Variable 1 ~ Batch variable	Intercept	2007.45	0.146	<2e-16
	Batch	2.6029	2.08	0.213
Surrogate Variable 2 ~ Batch variable	Intercept	2007.45	0.14	<2e-16
	Batch	8.03	2.01	9e-05

4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31

1 **Table 4.** Organ dysfunctions by MODS trajectory on day 1, 3, and 7 of septic shock
 2 diagnosis in the training dataset (GSE66099).
 3
 4

	Persistent MODS	Resolving MODS	P value
Day 1 MODS	N=44	N=63	0.74
Cardiovascular	43	57	0.57
Respiratory	44	45	0.67
Renal	36	11	<0.01
Hepatic	22	5	<0.01
Hematologic	33	13	<0.01
Neurologic	9	0	<0.01
Day 3 MODS	N=44	N=26	<0.01
Cardiovascular	38	32	<0.01
Respiratory	43	29	<0.01
Renal	36	8	<0.01
Hepatic	23	4	<0.01
Hematologic	28	12	<0.01
Neurologic	14	0	<0.01
Day 7 MODS	N=46	N=0	<0.01
Cardiovascular	32	3	<0.01
Respiratory	42	7	<0.01
Renal	34	4	<0.01
Hepatic	23	1	<0.01
Hematologic	24	1	<0.01
Neurologic	14	0	<0.01

5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23

1 **Table 5.** Organ support on day 1, 3, and 7 of septic shock diagnosis in the training
 2 dataset (GSE66099).

	Persistent MODS	Resolving MODS	P value
Day 1 MODS	N=44	N=63	0.74
Vasoactive support	39	45	0.26
Ventilatory support	41	35	0.03
Renal replacement	13	1	<0.01
Day 3 MODS	N=44	N=26	<0.01
Vasoactive support	37	32	<0.01
Ventilatory support	43	29	<0.01
Renal replacement	21	1	<0.01
Day 7 MODS	N=46	N=0	<0.01
Vasoactive support	32	4	<0.01
Ventilatory support	42	8	<0.01
Renal replacement	22	1	<0.01

3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33

1 **Table 6.** Genes features (n=111) identified through supervised machine learning
 2 predictive of a persistent MODS trajectory in the training dataset (GSE66099), listed in
 3 decreasing order of strength of association upon repeated cross-validation experiments.
 4

#	Gene	Fraction	#	Gene	Fraction	#	Gene	Fraction
1	<i>RETN</i>	1.000	38	<i>PNPLA6</i>	0.886	75	<i>RASGRP1</i>	0.828
2	<i>ADAMTS3</i>	1.000	39	<i>LTF</i>	0.886	76	<i>PTX3</i>	0.828
3	<i>LDHA</i>	1.000	40	<i>HLA-DPA1</i>	0.886	77	<i>HIPK2</i>	0.828
4	<i>LCN2</i>	1.000	41	<i>MS4A4A</i>	0.886	78	<i>CD86</i>	0.828
5	<i>IL1R2</i>	1.000	42	<i>CENPW</i>	0.886	79	<i>ELANE</i>	0.828
6	<i>DDIT4</i>	0.971	43	<i>FGFBP2</i>	0.886	80	<i>LY9</i>	0.828
7	<i>CEACAM8</i>	0.971	44	<i>CEACAM1</i>	0.886	81	<i>THBS1</i>	0.828
8	<i>MERTK</i>	0.971	45	<i>TAGAP</i>	0.886	82	<i>NR3C2</i>	0.828
9	<i>MPO</i>	0.971	46	<i>PRG2</i>	0.857	83	<i>NARF</i>	0.828
10	<i>ARL4A</i>	0.971	47	<i>DAAM2</i>	0.857	84	<i>HCAR3</i>	0.828
11	<i>CDKN3</i>	0.971	48	<i>ORM1</i>	0.857	85	<i>CFD</i>	0.828
12	<i>PRTN3</i>	0.971	49	<i>IFI44L</i>	0.857	86	<i>CCNE2</i>	0.828
13	<i>MTMR11</i>	0.971	50	<i>SLCO4A1</i>	0.857	87	<i>IFIT5</i>	0.828
14	<i>ANLN</i>	0.971	51	<i>BEX1</i>	0.857	88	<i>CLEC4D</i>	0.828
15	<i>IL1RAP</i>	0.971	52	<i>IFIT1</i>	0.857	89	<i>GADD45A</i>	0.828
16	<i>HLA-DMB</i>	0.971	53	<i>NELL2</i>	0.857	90	<i>ROMO1</i>	0.828
17	<i>ZBTB16</i>	0.971	54	<i>RPS6KA5</i>	0.857	91	<i>PADI4</i>	0.800
18	<i>NUSAP1</i>	0.942	55	<i>COL17A1</i>	0.857	92	<i>NUF2</i>	0.800
19	<i>GGH</i>	0.942	56	<i>PARP8</i>	0.857	93	<i>CEBPE</i>	0.800
20	<i>MMP8</i>	0.942	57	<i>CX3CR1</i>	0.857	94	<i>UPP1</i>	0.800
21	<i>PRC1</i>	0.942	58	<i>TBC1D4</i>	0.857	95	<i>CEACAM21</i>	0.800
22	<i>CD24</i>	0.942	59	<i>TOP2A</i>	0.857	96	<i>TSPAN13</i>	0.800
23	<i>CTSL</i>	0.942	60	<i>HSP90AA1</i>	0.857	97	<i>KLRF1</i>	0.800
24	<i>MAFF</i>	0.942	61	<i>TCEAL9</i>	0.857	98	<i>TSPO</i>	0.800
25	<i>NFE2</i>	0.942	62	<i>ARG1</i>	0.857	99	<i>DDAH2</i>	0.800
26	<i>BLM</i>	0.942	63	<i>SUCNR1</i>	0.857	100	<i>GNA15</i>	0.800
27	<i>OLFM4</i>	0.942	64	<i>KIF14</i>	0.857	101	<i>ASPM</i>	0.800
28	<i>MAP3K7CL</i>	0.942	65	<i>TGFBI</i>	0.857	102	<i>KCNE1</i>	0.800
29	<i>CEACAM6</i>	0.914	66	<i>OLAH</i>	0.857	103	<i>CD3E</i>	0.800
30	<i>FCER1A</i>	0.914	67	<i>CR1L</i>	0.857	104	<i>RTN1</i>	0.800
31	<i>CEP55</i>	0.914	68	<i>ETS2</i>	0.857	105	<i>CTSO</i>	0.800
32	<i>TLR7</i>	0.914	69	<i>TUBG1</i>	0.857	106	<i>CCL5</i>	0.800
33	<i>GPI</i>	0.914	70	<i>UHRF1</i>	0.857	107	<i>CACNA2D3</i>	0.800
34	<i>SLC46A2</i>	0.914	71	<i>CTSG</i>	0.828	108	<i>NR1D2</i>	0.800
35	<i>FCGR2B</i>	0.914	72	<i>HGF</i>	0.828	109	<i>DDX58</i>	0.800
36	<i>SLC51A</i>	0.914	73	<i>NDUFA1</i>	0.828	110	<i>NKG7</i>	0.800
37	<i>H1-2</i>	0.886	74	<i>ZNF600</i>	0.828	111	<i>LRG1</i>	0.800

5
 6 *Fraction: Indicates fraction of repeated cross-validation experiments in which the genes
 7 identified were associated with persistent MODS trajectory.

1 **Table 7.** Gene set predictive of sepsis mortality published by Sweeney et al. listed in
 2 alphabetical order used to predict risk of persistent MODS in the validation and test
 3 datasets.
 4

#	Gene	#	Gene	#	Gene
1.	<i>AIM2</i>	24.	<i>G0S2</i>	47.	<i>RGS1</i>
2.	<i>APH1A</i>	25.	<i>GSTM1</i>	48.	<i>SEPP1</i>
3.	<i>B4GALT4</i>	26.	<i>HIF1A</i>	49.	<i>TGFB1</i>
4.	<i>BPI</i>	27.	<i>HIST1H3H</i>	50.	<i>TRIB1</i>
5.	<i>C11orf74</i>	28.	<i>IFI27</i>	51.	<i>TST</i>
6.	<i>CCR2</i>	29.	<i>IKZF2</i>	52.	<i>VNN3</i>
7.	<i>CD163</i>	30.	<i>IL1R2*</i>	53.	<i>CIT (N/A)</i>
8.	<i>CD24</i>	31.	<i>IL8</i>	54.	<i>PLK1 (N/A)</i>
9.	<i>CD5</i>	32.	<i>KCNJ2</i>	55.	<i>OR52R1(N/A)</i>
10.	<i>CEACAM8*</i>	33.	<i>LY86</i>	56.	<i>NT5E (N/A)</i>
11.	<i>CEP55</i>	34.	<i>MAFF</i>	57.	<i>ABCB4(N/A)</i>
12.	<i>CFD</i>	35.	<i>MKI67</i>	58.	<i>CBFA2T3 (N/A)</i>
13.	<i>CKS2</i>	36.	<i>MPO*</i>		
14.	<i>CLEC10A</i>	37.	<i>MT1G</i>		
15.	<i>CST3</i>	38.	<i>MTMR11*</i>		
16.	<i>CTSG</i>	39.	<i>NDUFV2</i>		
17.	<i>CTSS</i>	40.	<i>OCLN</i>		
18.	<i>CX3CR1</i>	41.	<i>PAM</i>		
19.	<i>DDIT4*</i>	42.	<i>PER1</i>		
20.	<i>DEFA4</i>	43.	<i>POLD3</i>		
21.	<i>DHRS7B</i>	44.	<i>PSMA6</i>		
22.	<i>EIF5A</i>	45.	<i>RAB40B</i>		
23.	<i>EMR3</i>	46.	<i>RCBTB2</i>		

5
 6 *Indicates genes that overlap with those top 20 genes predictive of persistent MODS
 7 trajectory identified in our gene set.

8
 9 N/A -Indicates 6 genes that were not consistently found across the validation and test
 10 sets used in our study to predict risk of persistent MODS.
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22

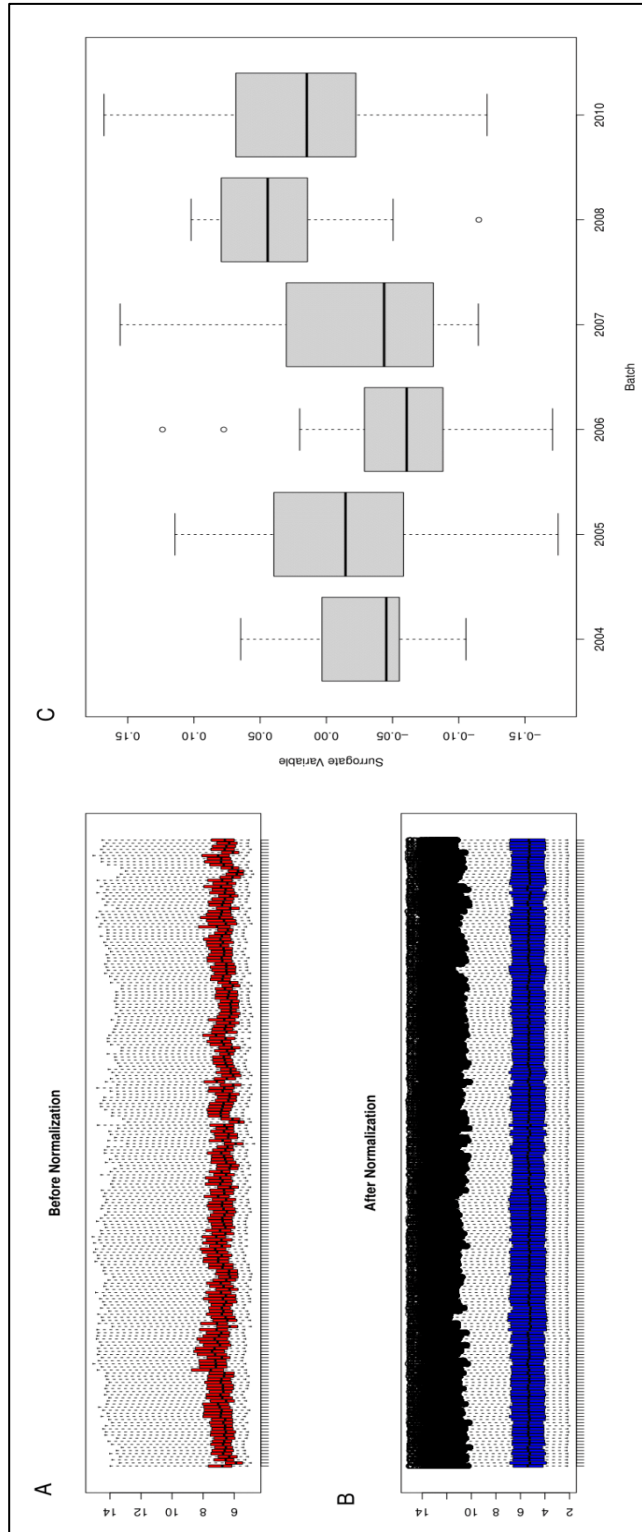
1 **Supplementary Figure Legend:**

2
3 **Figure 1.** Preprocessing of the expression measurements belonging to the derivation
4 dataset. **(A,B):** The effect of normalization on the average gene expression values. The
5 x-axis represents the samples, and the y-axis represents the gene expression values.
6 Based on the figures, the average expression values of the samples were more stable
7 and consistent after normalization and suitable for analysis. **(C)** Association of surrogate
8 variables with the actual batch variable. Since the samples were processed at different
9 time points spread over six years, we had to remove the resulting variation (batch
10 effect) from the data using the `Combat()` in the “sva” package in R. The current figure
11 shows the association between one of the inferred batch effects through SVA and the
12 actual batch variable (year). We passed the full model (without any batch variable) and
13 the batch variable as separate arguments to the `Combat()`. The output consists of a
14 corrected expression set with the batch effects removed completely.

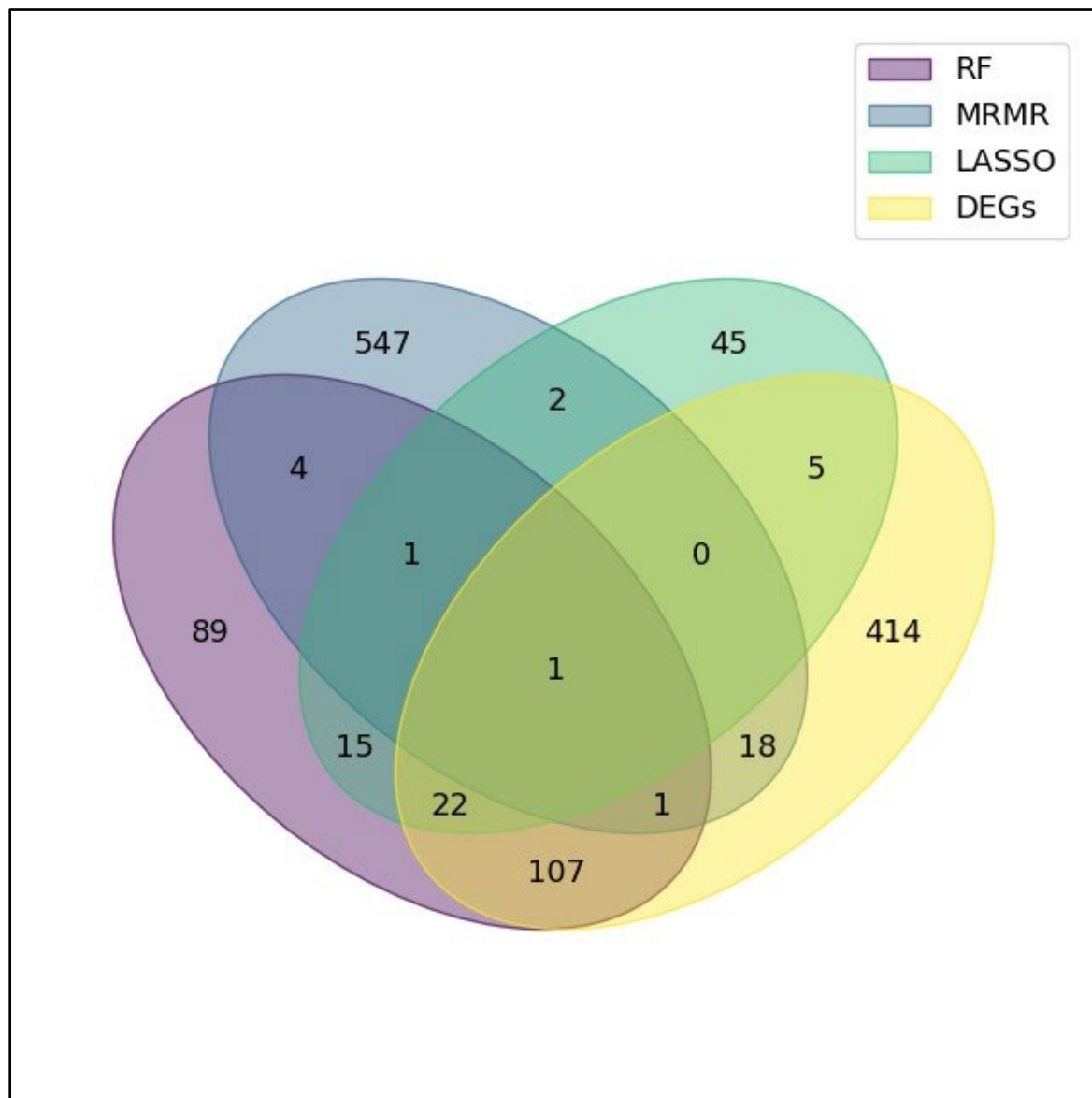
15
16 **Figure 2.** Venn diagram showing number of genes identified between the different
17 feature selection methods deployed least absolute shrinkage and selection operator
18 (LASSO), Minimum Redundancy and Maximum Relevance (MRMR), and Random
19 forests (RF) based variable importance technique AND the list of differentially
20 expressed genes (DEGs) in the training dataset (GSE66099) across repeated cross-fold
21 validation experiments.

1 **Supplementary Figures:**

2
3 **Figure 1.**



1 **Figure 2.**



1 **References:**

2

3 1. Tibshirani, R. Regression Shrinkage and Selection via the Lasso. *Journal of the*

4 *Royal Statistical Society. Series B (Methodological)* **58**, 267–288 (1996).

5 2. Ding, C. & Peng, H. Minimum redundancy feature selection from microarray gene

6 expression data. *J Bioinform Comput Biol* **3**, 185–205 (2005).

7 3. Gregorutti, B., Michel, B. & Saint-Pierre, P. Correlation and variable importance in

8 random forests. *Stat Comput* **27**, 659–678 (2017).