# Finding Single and Multi-Gene Expression Patterns for Psoriasis Using Sub-Pattern Frequency Pruning

Kenneth Smith
*Department of Computer Science*
*University of Missouri - St. Louis*
St. Louis, MO, USA
kenneth.smith@mail.umsl.edu

Jamie Lea
*Department of Computer Science*
*University of Missouri - St. Louis*
St. Louis, MO, USA
jmltf7@umsystem.edu

Sharlee Climer
*Department of Computer Science*
*University of Missouri - St. Louis*
St. Louis, MO, USA
climer@umsl.edu

***Abstract*—Biomarker identification, such as gene expression, is used in several areas of medical research, including aiding in disease prediction and treatment. However, most gene expression analysis focuses on differently expressed genes, ignoring patterns in which the co-expression of non-differently expressed genes are associated with disease risk. In this manuscript, we make three contributions. First, we present an alternative definition for differential expression which captures associations that are missed using mean- or median-based methods, such as fold change. Second, we introduce an algorithm for identifying *all* patterns of analytes associated with a given phenotype within a given threshold of optimal by extensively pruning the solution space. Third, our demonstration on psoriasis gene expression data yields 6320 highly significant gene expression patterns associated with this common disease that are comprised of 2334 unique genes worthy of further exploration. Interestingly, these genes include 1021 genes that are not differentially expressed when examined in isolation. Our approach is computationally efficient and our open-source software is freely available. This method holds potential for biomarker discovery for diverse phenotypes and is also applicable for identifying patterns hidden within non-biological real-valued data sets.**

***Index Terms*—biomarkers, co-expression analysis, gene expression, psoriasis**

## I. Introduction

Biological markers, or biomarkers, measure interactions between a biological system and a possible hazard. They are useful for understanding the cause of, predicting the risk of developing, and monitoring the outcome of treatment associated with, a disease [1]. Various types of biomarkers are studied including genetic variants [2]–[5], protein and gene expression levels [6]–[11], comorbidities, and treatment status (e.g. diabetes, treatment for hypertension, blood pressure variability) [12]. Based on these types of biomarkers, risk factors for Alzheimer Disease (AD) [11], cardiovascular disease [2], [3], [12], psoriasis [6]–[10], type 2 diabetes [2], inflammatory bowel disease [2], and several types of cancer [2], [5] have been identified. Importantly, biomarkers enable precision medicine as they hold the potential to decipher heterogeneity by sifting out subgroups of individuals with common etiologies.

This work was supported in part by National Institute on Aging (NIA) grants 1RF1AG053303-01 and 3RF1AG053303-01S2.

When analyzing continuous valued biomarkers, fold change (FC) measurements are commonly used to identify deferentially expressed genes (DEG) [6], [8]–[10]. However, these studies ignore combinations of genes whose co-expression, as a whole, is associated with disease risk, even if the individual constituents are not [16]. Recently, Le et al. [7] compared several machine learning techniques for psoriasis classification using gene expression data. A Random Forest was used to select the features used in the classification models. This approach identified the genes *FABP5*, *TGM1*,and *BCAR3* as effective in psoriasis classification.

In this paper we present a new algorithm for identifying DEG and combinations of biomarkers associated with disease risk. Large patterns of analytes are identified by pruning away sections of the solution space based on sub-pattern frequency. We apply this method to a psoriasis gene expression data set and identify 6320 risk patterns. These patterns are composed of 1313 DEG and 1021 non-DEG. Finally, we compare our results with the DEG found by the data set authors.

Our paper is organized into the following sections. We begin with an overview of the theory behind our pruning method in Section II. This is followed by a description of our methods in Section III, including data preprocessing, pattern identification, and testing pattern strength with logistic regression. Next, in Section IV, we discuss our results and compare with a previous analysis of the data set. We conclude the paper with a brief discussion.

## II. Theory

This algorithm searches for differentially expressed analytes, along with combinations of analytes, that are observed between two groups, such as diseased cases and normal controls, and referred to as, $G_1$ and $G_2$. Continuous expression values are discretized into high, neutral, and low categories, called analyte states. Details are provided in Section III. A specific combination of analyte states is called a pattern. The association strength between each pattern and the disease is measured using the objective function in (1): the difference between the frequency of the pattern in $G_1$ members and the frequency of the pattern in $G_2$ members. If risk patterns are desired, $G_1$ is populated with cases and $G_2$ is populated with controls. These groups are reversed for protective patterns.

The objective value is optimized for each pattern size (PS), where PS is equal to the cardinality of the analytes in the pattern. In (1), a sample, $samp$, is set to 1 if it contains all of the analyte states in the pattern and 0 otherwise, and $n_1, n_2$ are the number of individuals in $G_1$ and $G_2$, respectively. This objective functions is denoted as $Z$ in the following algorithms.

$$\begin{aligned} Z &= freq_1 - freq_2 \\ &= \frac{1}{n_1} \sum_{j \epsilon G_1} samp_j - \frac{1}{n_2} \sum_{k \epsilon G_2} samp_k \end{aligned} \tag{1}$$

To find the optimal pattern, one could iterate through all possible analyte state combinations for a given PS. This is usually feasible for small PS, but becomes intractable for larger patterns as the run time is $O(m^{PS})$, were $m$ is the total number of analyte states. By using the $G_1$ frequency ($freq_1$) from all PS=2 patterns, along with the following two Lemmas, large portions of the solution space can be pruned away, leaving a tractable number of combinations to iterate through.

**Lemma 1.** *The frequency of a pattern is bounded above by the frequency of any subset of the pattern.*

*Proof.* Let $P$ be a pattern. Let $P_1$ and $P_2$ partition $P$. Let $S_1$ be the set of individuals possessing the analyte states in $P_1$, and $S_2$ be the set of individuals possessing the analyte states in $P_2$. Since the set of individuals possessing the pattern, $S_1 \cap S_2$, is a subset of $S_1$, $|S_1 \cap S_2| \leq |S_1|$. Thus the number of individuals possessing the pattern cannot be greater than those possessing any subset of the pattern. □

**Lemma 2.** *The objective value is bounded above by* $freq_1$.

*Proof.* The subtrahend of (1) is always non-negative, so it follows that the objective function value cannot exceed the value of the addend. □

Lemma 1 establishes the $freq_1$ of any pattern is bounded above by the $freq_1$ of any sub-pattern. Lemma 2 illustrates the objective value of a pattern, $Z$, is bounded above by the $freq_1$ of the pattern. Taken together, these lemmas show that the objective value of a pattern is bounded above by the $freq_1$ of any sub-pattern. Thus, if any PS=2 pattern has a $freq_1$ below some threshold, all patterns which contain the PS=2 pattern as a sub-pattern must have an objective value below the same threshold. When solving $PS \geq 4$ problems, our algorithm operates in two stages: an approximation stage and an optimal solution stage. The approximation stage provides the threshold used to prune the solution space in the optimal stage.

## III. Method

This section begins with a description of our data pre-processing steps. Next, we discuss how the algorithm finds optimal patterns for PS=1,2,3. The approximation and optimal solution algorithms for PS≥4 are then detailed. Finally, an overview of how the patterns are used to train linear models of compound biomarkers is explored.

### A. Data Preparation

Data pre-processing consists of three steps: 1. Limiting missing data, 2. Discretization, 3. Dataset splitting. First, samples and analytes are iteratively removed until each analyte, and each sample, have a maximum of 10% missing data. Next, the data are discretized into high, neutral, and low categories by per-analyte quantile thresholding. Typically the first and third tertiles are used for the low and high categories respectively. Finally the data set is split into a discovery and a test set, with 60-70% of samples in the discovery set.

### B. Find Optimal Solution via Iteration

The discretized discovery data is transformed into a binary matrix, $\mathbf{E}$. Each value in the discovery data set is encoded by two rows in $\mathbf{E}$, as shown in Table I. Missing data are treated as both high and low for discovery, but are excluded from the validation process.

TABLE I
ANALYTE ENCODING

| Binned Analyte Level | High | Normal | Low | Missing |
|---|---|---|---|---|
| High Expression | 1 | 0 | 0 | 1 |
| Low Expression | 0 | 0 | 1 | 1 |

In addition to finding the optimal solution for each PS, a collection of near optimal solutions, called a solution pool is maintained. The PS=1 solution pool contains all patterns with an objective value $\geq 0.5$. For PS=2 and 3, the solution pools are controlled by the parameter $beta$. When PS$\geq 4$, the solution pools are controlled by the objective values found in the approximation stages. Optimal solutions, along with the solution pools, are determined by iterating through all possible combinations of analyte states for PS=1,2,3. Additionally, the $freq_1$ frequency for all PS=2 patterns is saved for use in later phases.

### C. Find Approximate Solutions

For all PS≥4, Algorithm 1 is used to find a pool of $\beta$ approximate solutions. These approximate solutions are used to determine the lower bound on $Z$ in the optimal solution phase. In Algorithm 1, $\gamma$ is the largest pattern size to be tested, $\beta$ is the maximum number of solutions to keep for each PS, $prevPool$ is initialized to the PS=3 solution pool, $\mathbf{E}$ is the binary matrix calculated above, and $\mathbf{F}$ is an $m \times m$ symmetric matrix containing the $freq_1$ pair-wise frequencies for all PS=2 patterns, where each row and each column represent an analyte state. The $gt\_thresh$ function is described in Algorithm 2. This function returns True if and only if the new pairwise frequencies in the concatenated pattern are all above the current threshold. Algorithm 1 produces a pool of $\beta$ high-quality approximate solutions. Note that this algorithm

is not optimal as it misses patterns that do not contain a sub-pattern that appeared in the PS-1 solution pool. The purpose of the algorithm is to provide thresholds used for pruning by the optimal solver for each PS, as described next.

**Algorithm 1** Find Approximate Solution Pool

**Require:** $\gamma, \beta, prevPool, \mathbf{E}, \mathbf{F}$
**for** PS=4,5,...$\gamma$ **do**
  $curPool \leftarrow \varnothing$
  $thresh \leftarrow 0$
  **for all** $pat$ in $prevPool$ **do**
    **for all** $state$ in $\mathbf{E}$ **do**
      **if** $state$ not in $pat$ **then**
        **if** gt_thresh($\mathbf{F}, pat, state, thresh$) **then**
          $newPat =$ concat($pat, state$)
          $freq_1 =$ calcFreq($\mathbf{E}, newPat, G_1$)
          **if** $freq_1 > thresh$ **then**
            $freq_2$ = calcFreq($\mathbf{E}, newPat, G_2$)
            $Z = freq_1 - freq_2$
            **if** $Z > thresh$ **then**
              $curPool$.updatePool($newPat, Z$)
              $thresh = curPool.worst\_Z$
            **end if**
          **end if**
        **end if**
      **end if**
    **end for**
  **end for**
  $thresh \leftarrow$ Worst objective value in $curPool > 0$
  **print** $thresh$
  $prevPool \leftarrow curPool$
**end for**

**Algorithm 2** gt_thresh

**Require:** $\mathbf{F}, pat, state, thresh$
**for all** $analyte\_state$ in $pat$ **do**
  $i \leftarrow analyte\_state$
  $j \leftarrow state$
  **if** $\mathbf{F}_{ij} <= thresh$ **then**
    **return** False
  **end if**
**end for**
**return** True

### *D. Find Optimal Solutions for Larger Patterns*

For each PS, the $thresh$ value found in Algorithm 1 and the matrix $\mathbf{F}$ is used to create an $m \times m$ symmetric matrix, $\mathbf{B}$, where $m$ is the number of analyte states. Matrix $\mathbf{B}$ is initialized according to Algorithm 3. $thresh$ is a lower bound on the objective value of any pattern that is to be added to the pool. The pair-wise $freq_1$ frequencies, calculated above, provide an upper bound on the $freq_1$ frequency of any larger pattern containing the pair. By combining the two Lemmas, along with the lower and upper bounds, any patterns with pair-wise frequencies below the threshold can be removed.

**Algorithm 3** Initialize Matrix $\mathbf{B}$

**Require:** $\mathbf{F}$, $thresh$
$\mathbf{B} \leftarrow \mathbf{0}_{n \times n}$
**for** $i$ in $\{1, 2, \ldots, m\}$ **do**
  **for** $j$ in $\{i+1, i+2, \ldots, m\}$ **do**
    **if** $\mathbf{F}_{ij} > threshold$ **then**
      $\mathbf{B}_{ij} = 1$
      $\mathbf{B}_{ji} = 1$
    **end if**
  **end for**
**end for**
**return** $B$

Next, the sum of each column in $\mathbf{B}$ is calculated. This sum indicates the maximum number of analytes that could be in a pattern with the column index, and result in a pattern with an objective value above $thresh$. The column indices, along with the corresponding column sums, $col\_count$, are put into a priority queue, which keeps the pair sorted in increasing order, based on column sum.

The optimal pattern for each PS$\geq$4, along with the solution pools are determined using Algorithms 4 and 5. Each analyte state is evaluated in increasing order, based on the $col\_count$. In this section we will refer to the analyte state corresponding to $col\_count$ as the candidate state. If $col\_count < PS - 1$, then no pattern of size $PS$, which includes the candidate state, can have an objective value above $thresh$. Since $col\_count < PS - 1$, we know that for any pattern of size $PS$, which includes the candidate state, there exists at least one pair-wise $freq_1 \leq thresh$. From Lemma 1, the $freq_1$ of any such pattern is also $\leq thresh$. Then from Lemma 2, the objective value of any such pattern is $\leq thresh$. Therefore, the candidate state can be removed from the solution space without loss of optimality.

On the other hand, if $col\_count \geq PS-1$, then it is possible that some pattern of size $PS$ which includes the candidate state, has an objective value $> thresh$. In this case, the possible patterns are evaluated. However, only patterns in which all elements have a pair-wise $freq_1$ values with the candidate state $> thresh$ need to be considered. Again, if a pattern contained an analyte state with a pair-wise $freq_1 \leq thresh$, Lemma 1 and 2 allow us to conclude that the objective value of the pattern is $< thresh$. The constituent elements of the patterns to enumerate are collected using Algorithm 6. Once all of these patterns have been evaluated, the candidate state is removed from the solution space, since we have considered all patterns of size $PS$ that include the candidate state. After a candidate state is removed, the $col\_count$ for each remaining analyte state is recalculated.

In order to facilitate the parallelization of the algorithm, a main/worker architecture is used. The main process determines the candidate state and the analyte pool, and then sends the problem to a worker process. The worker process enumer-

ates through the possible patterns and returns all solutions $> thresh$. While evaluating the patterns in Algorithm 5, the solution space is further pruned by considering the $freq_1$ values of PS=3 patterns. Consider evaluating the analyte pool {1,2,3,4,5,6,7} for PS=5 and candidate state = 0. Table II lists the possible patterns generated from this pool, sorted by the first three analyte states. Since the order of the analyte states in the pattern does not matter, we only consider the patterns with the analyte states sorted in increasing order. If $freq_1$ of pattern {1,2,3} is $< thresh$, then all 6 PS=5 patterns in Table II that include {1,2,3} will have an objective value below $thresh$.

TABLE II
EXAMPLE PATTERN LIST

| Candidate State | Analytes From Pool | |
|---|---|---|
| 0 | 123 | 45 |
| 0 | 123 | 46 |
| 0 | 123 | 47 |
| 0 | 123 | 56 |
| 0 | 123 | 57 |
| 0 | 123 | 67 |
| 0 | 124 | 56 |
| ⋮ | ⋮ | ⋮ |
| 0 | 245 | 67 |
| 0 | 345 | 67 |

**Algorithm 4** Find Optimal Solution Pool - Main

**Require:** $queue, PS, \mathbf{B}, \mathbf{E}, thresh$
**while** queue not empty **do**
  $(j,\ col\_count) \leftarrow queue$.pop()
  **if** $col\_count \geq PS - 1$ **then**
    $pool \leftarrow getPool(B, j)$
    sendProblemToSlave($j, pool, thresh, \mathbf{E}, PS$)
  **end if**
  $B_{*j} = \overrightarrow{0}$
  $B_{j*} = \overrightarrow{0}$
  Recalculate column counts. Update queue
**end while**

### E. Pattern Evaluation

The quality of univariate biomarkers can be evaluated using Area Under the Receiver Operator Characteristic Curve (AUROC). To evaluate the quality of an analyte pattern we measure the AUROC of a logistic regression model fitted on the real valued discovery data corresponding to the discrete pattern. While more powerful model classes exist which may be more performant as classifiers, we wish to evaluate the patterns themselves as biomarkers, hence a simpler statistical model is better. We also evaluate the patterns using an exact test on the discretized data. In both cases, samples are removed if they have any missing data in the pattern.

The objective function $Z$ is $freq_1 - freq_2 > 0$, which is a statistic on a $2 \times 2$ contingency table allowing it's significance to be calculated with an exact test. Since the number of individuals in each group is known in advance,

**Algorithm 5** Find Optimal Solution Pool - Worker

**Require:** $j, pool, thresh, \mathbf{E}, PS$
$trip\_pool \leftarrow$ Get All PS=3 patterns from pool
**for all** $trip\_pats$ in $trip\_pool$ **do**
  $freq_1 =$ calcFreq($\mathbf{E}, trip\_pat, G_1$)
  **if** $freq_1 > thresh$ **then**
    $pat\_pool \leftarrow$ Get all pattern of size PS with $trip\_pat$
    **for all** $pat$ in $pat\_pool$ **do**
      $newPat \leftarrow [j, pat]$
      $freq_1 =$ calcFreq($\mathbf{E}, newPat, G_1$)
      **if** $freq_1 > thresh$ **then**
        $freq_2$ = calcFreq($\mathbf{E}, newPat, G_2$)
        $Z = freq_1 - freq_2$
        **if** $Z > thresh$ **then**
          **print** $pat$
        **end if**
      **end if**
    **end for**
  **end if**
**end for**

**Algorithm 6** Get Pool

**Require:** $B$, $j$
$pool = \{\}$
**for** $i$ in $\{1, 2, \ldots, m\}$ **do**
  **if** $B_{ij} == 1$ **then**
    Add $i$ to $pool$
  **end if**
**end for**
**return** $pool$

the distribution of tables is bionmial and a singly-conditioned test is correct; we use Boschloo's Exact test. As the group with larger proportion is chosen before the experiment, a one-sided $p$-value is appropriate. While the method treats missing data as allowed, we can create a stricter test by excluding samples with missing data for the pattern. This can change the objective value somewhat; in the below discussion and in all plots $Z*$ refers to the objective value calculated without missing data. All patterns had significant $Z*$ values after correction using the Holm-Bonferroni method.

RNAseq data typically has widely varying ranges for each analyte, and so often needs to be rescaled for a regression solver to converge. As our method valuably captures distribution extremes, outliers are never removed and so we rescale by subtracting the median and dividing by the inter-quartile range. Patterns with excessive missing data in the discovery set as a whole or in either group are removed. Since it would be nonsensical to evaluate models with insufficient data, we also remove patterns with excessive missing data on the entire test set.

Model training and validation happens in several steps: first a model is tested on the training set, and is kept if the AUROC meets a minimum threshold and if it is significant

at level $\alpha_D$. We also calculate an expected $p$-value for the validation AUROC, with $\alpha_{\mathbb{E}}$, by using the per-pattern test missing data frequency to proportionally reduce group sample size when calculating the U-statistic of the discovery AUROC. Significant models are then evaluated using $k$-fold cross-validation on the discovery set. For each of the $k$ ROC curves, the $y$-values are interpolated onto a common set of $x$-values to calculate a point-wise average ROC curve. The model is kept if the cross-validation AUROC meets a minimum threshold and is significant at $\alpha_{CV}$. Models that pass this step are then subjected to $P_n$ label permutation trials [17] with significance level $\alpha_P$. The last step is to determine a threshold for model evaluation. First we calculate $J^*$ which is the threshold closest to 0.5 that produces a Youden J-statistic in a small neighborhood of $\max\{J\}$. The threshold, either 0.5 or $J^*$, which produces the maximum sensitivity or specificity is used. A model is kept if either the specificity or sensitivity meet a minimum value, $\tau$ when evaluated at the selected threshold.

These criteria provide strict quality control over model performance, significance, and estimated out-of-sample performance. The final step is to validate the models on the hold out test set. The data is rescaled using the same parameters calculated on the discovery set, the AUROC significance determined, and Holm-Bonferroni correction applied at $\alpha_V$. Models with excessive group-wise missing data in the test set, or a low AUROC, are eliminated regardless of whether they are significant after correction.

## IV. Application of the Method to Psoriasis Transcriptomics Data

To show the efficacy of the method we applied it to psoriasis RNA-Seq data from skin punch biopsies and utilize extremely strict significance levels and other thresholds. GSE54456 was downloaded from the NCBI Gene Expression Omnibus (https://www.ncbi.nlm.nih.gov/geo/) and pre-processed as described above. This dataset was chosen as the variation between normal and lesional skin is stark and the authors of this dataset found a very large number of DEG. We can expect strong correlation structure in the analyte data which provides a good benchmark for our method's ability to find differentially expressed patterns. After pre-processing, 151 samples and 16963 analytes remained. The pattern quality criteria are listed in Table V.

The method was applied for PS=1, 2, ..., 9 on a Linux Mint System with 32 GB of RAM and i5-4690K CPU @ 3.50GHz processor, utilizing 3 CPUs. Run times for the approximation and optimal phases are shown in Table III. In the approximation phase, patterns were kept with $\beta = 1000$. The thresholds for each PS were set to the lowest non-zero objective value found during the approximation step. For PS=1, all patterns with an objective value greater than 0.5 were added to the solution pool. A total of 7156 patterns involving 2742 analytes were identified. The solution pool sizes are listed in Table IV. After application of the discovery quality criteria 6931 models involving 2688 analytes were produced.

TABLE III
Wall Clock Run Time in Seconds for Approximate and Optimal Phases for Each Pattern Size

| **PS** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| **Approx (s)** | N/A | N/A | N/A | 1 | 1 | 1 | 1 | 1 | 1 |
| **Optimal (s)** | 0 | 246 | 7 | 15 | 14 | 17 | 434 | 3342 | 16755 |

TABLE IV
Summary of Solution Pool Results

| **PS** | Patterns | Analytes | $Z$ | | $Z^*$ | |
|---|---|---|---|---|---|---|
| | | | Min | Max | Min | Max |
| 1 | 2606 | 2606 | 0.5 | 0.734 | 0.44 | 0.66 |
| 2 | 1000 | 592 | 0.5 | 0.594 | 0.348 | 0.562 |
| 3 | 1000 | 340 | 0.469 | 0.531 | 0.4 | 0.531 |
| 4 | 330 | 114 | 0.484 | 0.5 | 0.4 | 0.5 |
| 5 | 758 | 170 | 0.469 | 0.5 | 0.382 | 0.5 |
| 6 | 198 | 65 | 0.469 | 0.5 | 0.469 | 0.5 |
| 7 | 1065 | 150 | 0.453 | 0.5 | 0.453 | 0.5 |
| 8 | 181 | 87 | 0.453 | 0.5 | 0.453 | 0.5 |
| 9 | 18 | 34 | 0.453 | 0.484 | 0.453 | 0.484 |
| Total | 7156 | 2742 | | | | |

### *Validated Results*

Models were then evaluated on a hold-out test set for unbiased performance evaluation. Multiple testing correction was applied separately on each pattern size. Even given the extremely stringent criteria, 6320 highly performant and significant models involving 2334 analytes were found and validated (Table VII). Our method was also able to capture *FABP*, *TGM1*, *BCAR3* in PS1 and PS2, which makes the results comparable to that of the random forest method in [7]. These results show the power of the method to identify biomarker patterns.

## V. Discussion

### *Involvement of DEG*

We then examined the overlap between our patterns and the DEG found by the dataset authors. The dataset authors found 3505 DEG after analyzing 21,099 genes. Our selected set of 16963 analytes included 2844 of those DEG, 1498 of which were found by the method and 1270 of which produced validated models. This indicates that our method can identify

TABLE V
Pattern Quality Criteria

| **Parameter** | $\alpha_D$ | $\alpha_{\mathbf{E}}$ | $k$ | $\alpha_{CV}$ | $P_n$ | $\alpha_P$ | $\alpha_V$ | $\tau$ |
|---|---|---|---|---|---|---|---|---|
| **Value** | 5e-6 | 5e-3 | 3 | 5e-3 | 100 | 0.05 | 5e-6 | 0.7 |

TABLE VI
Number of models passing the Discovery criteria

| **PS** | PS1 | PS2 | PS3 | PS4 | PS5 | PS6 | PS7 | PS8 | PS9 |
|---|---|---|---|---|---|---|---|---|---|
| Models | 2555 | 838 | 992 | 327 | 757 | 198 | 1065 | 181 | 18 |
| Analytes | 2555 | 515 | 337 | 109 | 165 | 65 | 150 | 87 | 34 |

TABLE VII
SUMMARY OF VALIDATION AUC

| PS | Models | Analytes | AUC Range | | Adjusted $p$-value | |
|---|---|---|---|---|---|---|
| | | | min | max | min | max |
| 1 | 2096 | 2096 | 0.926 | 1.000 | 2.83E-12 | 4.92E-06 |
| 2 | 766 | 493 | 0.916 | 1.000 | 9.27E-13 | 4.88E-06 |
| 3 | 917 | 330 | 0.917 | 1.000 | 1.10E-12 | 4.95E-06 |
| 4 | 327 | 109 | 0.907 | 1.000 | 3.62E-13 | 4.65E-06 |
| 5 | 755 | 165 | 0.917 | 1.000 | 8.37E-13 | 3.78E-06 |
| 6 | 198 | 65 | 0.943 | 1.000 | 2.19E-13 | 4.66E-08 |
| 7 | 1063 | 150 | 0.919 | 1.000 | 1.18E-12 | 4.56E-06 |
| 8 | 180 | 87 | 0.921 | 1.000 | 2.00E-13 | 6.62E-07 |
| 9 | 18 | 34 | 0.892 | 1.000 | 1.99E-14 | 1.17E-06 |

TABLE VIII
SUMMARY OF DEG AND NON-DEG IN VALIDATED PATTERNS

| PS | DEG | non-DEG | w/o DEG | w/o non-DEG |
|---|---|---|---|---|
| 1 | 60.59% | 39.41% | 39.41% | 60.59% |
| 2 | 61.62% | 74.41% | 38.38% | 25.59% |
| 3 | 60.31% | 78.63% | 39.69% | 21.37% |
| 4 | 71.25% | 51.38% | 28.75% | 48.62% |
| 5 | 55.36% | 79.21% | 44.64% | 20.79% |
| 6 | 69.70% | 55.56% | 30.30% | 44.44% |
| 7 | 49.58% | 96.05% | 50.42% | 3.95% |
| 8 | 53.89% | 95.00% | 46.11% | 5.00% |
| 9 | 66.67% | 94.44% | 33.33% | 5.56% |

DEG consistent with existing literature. Although not all DEG were captured, this was due to the $Z$ cutoff of 0.5, which is somewhat arbitrary as all but three were significant with Boschloo's test.

We also validated 826 PS1 models that were not considered DEG by the dataset authors. The key difference between our method and FC is that our method identifies analytes for which the extreme values have a significant group association. FC identifies analytes that have a significant difference in group medians, which may not capture differences in the extremes of the analyte distribution. Fig. 1 and Fig. 2 compare the PS1 $Z^*$ and validation AUROC scores with $log_2FC$, respectively. The *DDB1* gene illustrates the difference between the two methods well as the low values were dominated by cases ($Z^*_D = 0.609$), produced a strong model ($\text{AUROC}_V = 0.96$) even though the medians were very close ($FC = -0.318$). Fig. 5 shows all validated PS1 with FC$<0.5$ and, as can be seen, there are clear but subtle group differences in expression state.

One goal of co-expression research is specifically to identify patterns that are not reliant on DEG. Not all co-expression patterns are combinations of DEG, the involvement of other analytes that are not DE is extremely important [18]. Our method is able to find larger co-expression patterns that involve, or are comprised entirely of, genes that are not differentially expressed. As shown in Tables VIII, as size increases patterns become less reliant on DEG and more reliant on non-DEG. Of the 2334 genes involved in the patterns found, 1021 are not DEG. For PS$>$2 40% of validated models have no DEG, 80%, involve at least one non-DEG, and only 20% do not involve non-DEG, suggesting our method finds valid higher-ordered patterns missed by methods that rely solely on DEG. Fig. 3 shows a PS5 pattern with strong differential expression characteristics: controls have lower and more consistent levels with strong inter-analyte correlation, while cases have higher levels over a broader range, and with weaker inter-analyte correlation. Fig. 4 has a complex correlation structure with group differences in expression state, correlation, and variance. In conclusion, our method is efficient, produces strong differential expression patterns, and finds patterns unavailable to FC based methods.

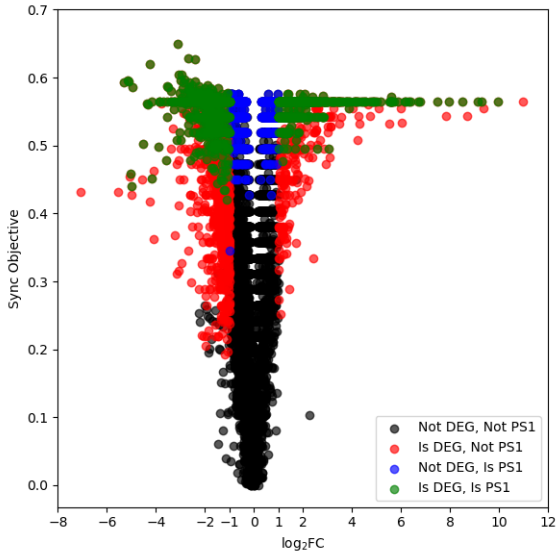


Fig. 1. Comparison of PS1 $Z^*$ and FC.

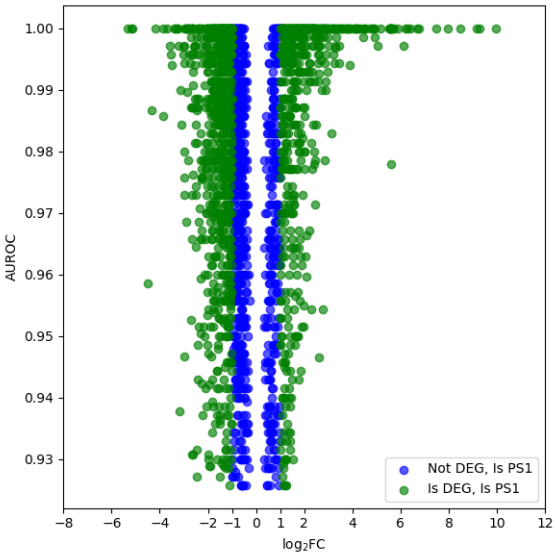


Fig. 2. Comparison of PS1 $Z^*$ and FC.

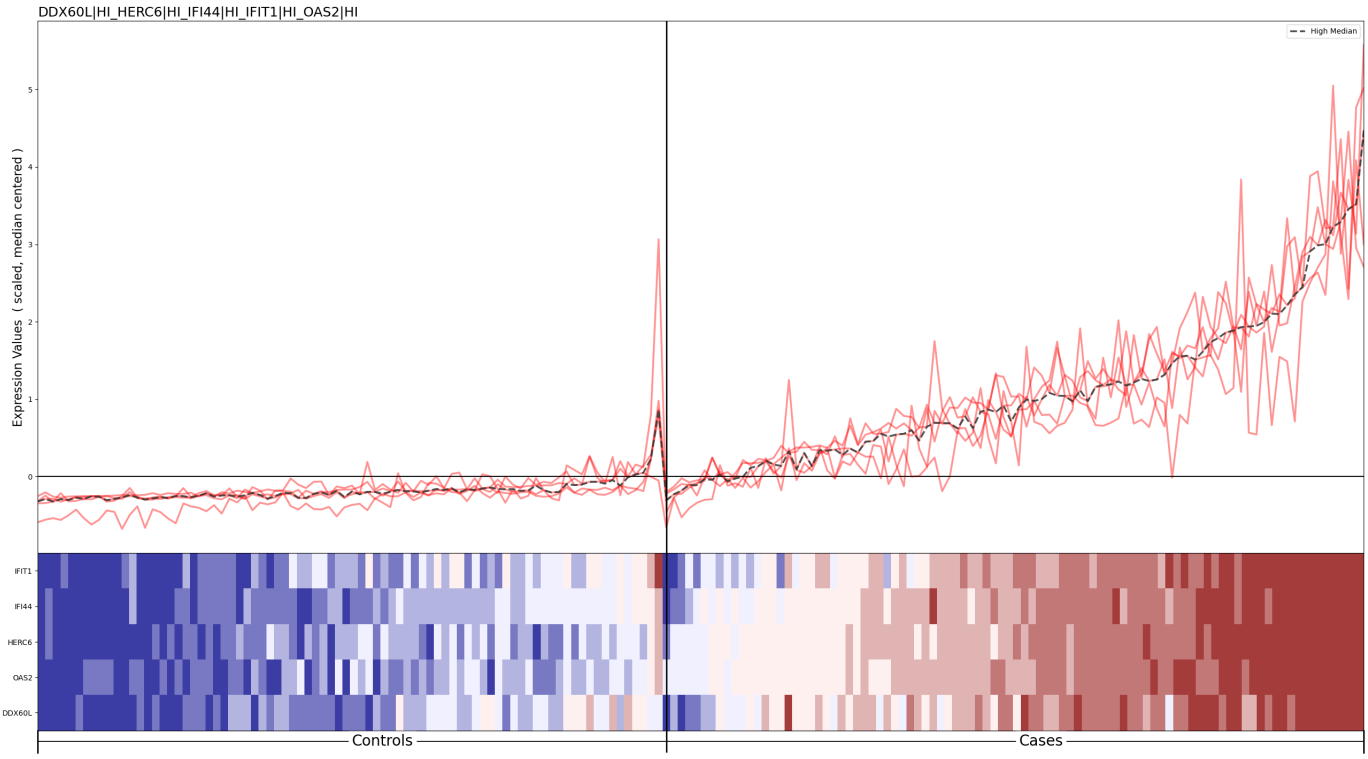


Fig. 3. This analyte pattern has strong differential co-expression, and differential correlation.

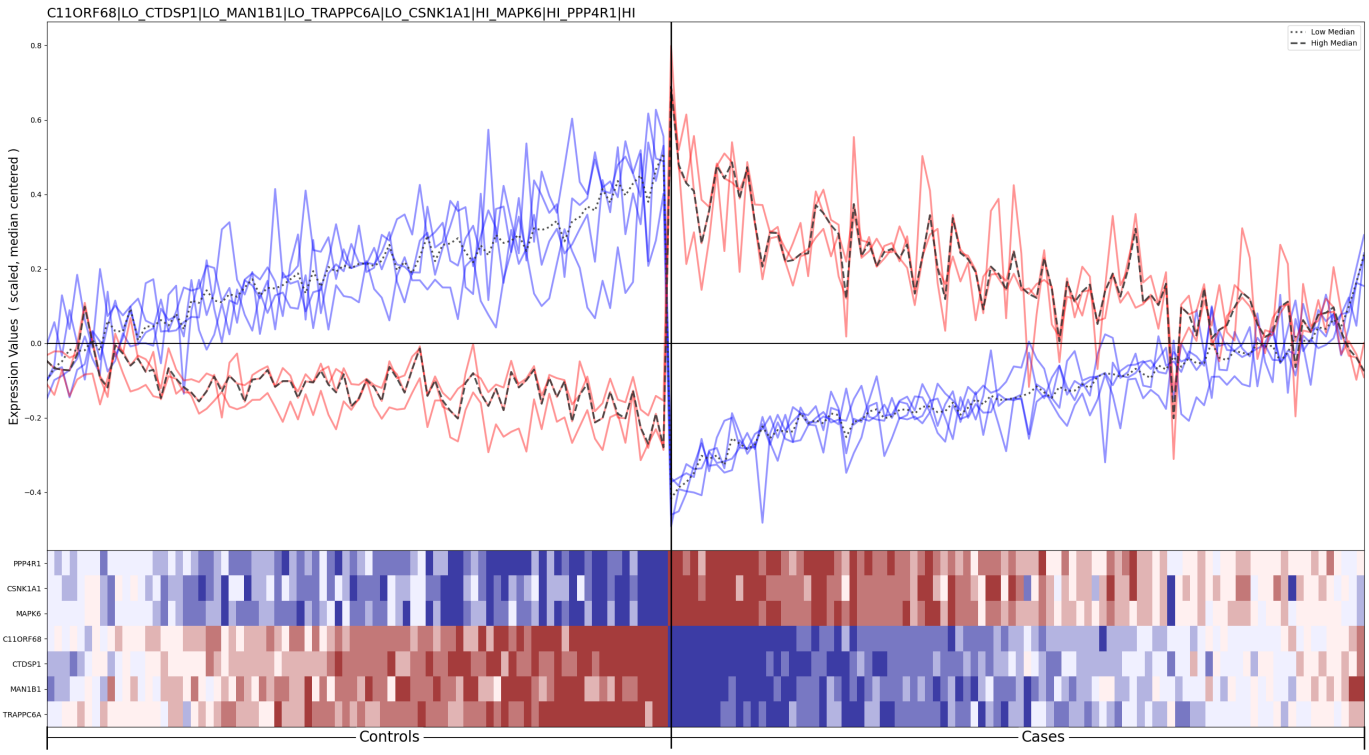


Fig. 4. Our method identifies differential co-expression while also simultaneously identifying expression state, elucidating complex pattern structure.

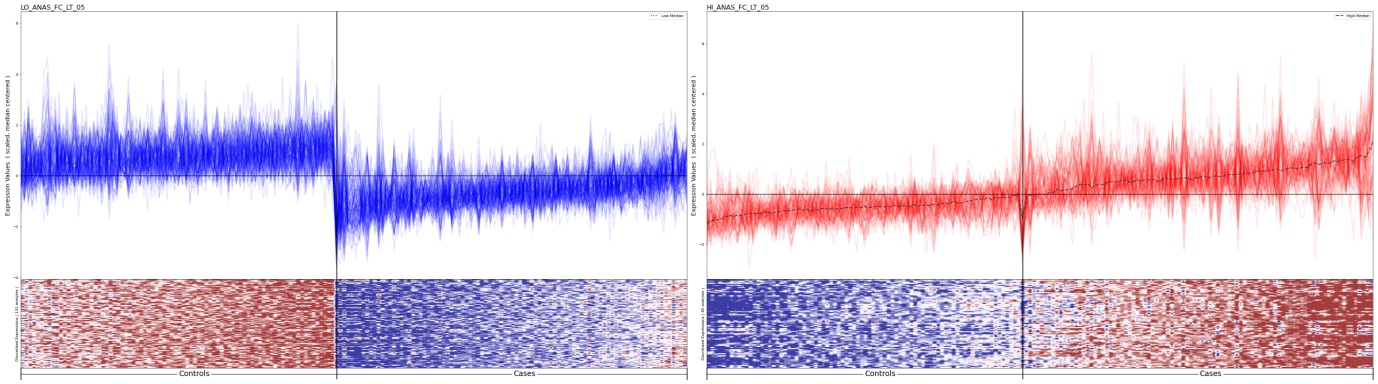


Fig. 5. Validated PS1 analytes with $log_2 FC < 0.5$ have clear group differences.

## REFERENCES


[1] R. Mayeux, "Biomarkers: Potential Uses and Limitations," NeuroRx, vol. 1, no. 2, pp. 182–188, 2004.

[2] A. V. Khera et al., "Genome-wide polygenic scores for common diseases identify individuals with risk equivalent to monogenic mutations," Nat. Genet., vol. 50, no. 9, pp. 1219–1224, 2018.

[3] M. Inouye et al., "Genomic Risk Prediction of Coronary Artery Disease in 480,000 Adults: Implications for Primary Prevention," J. Am. Coll. Cardiol., vol. 72, no. 16, pp. 1883–1893, 2018.

[4] N. R. Wray, K. E. Kemper, B. J. Hayes, M. E. Goddard, and P. M. Visscher, "Complex trait prediction from genome data: Contrasting EBV in livestock to PRS in humans," Genetics, vol. 211, no. 4, pp. 1131–1141, 2019.

[5] C. Long, G. Lv, and X. Fu, "Development of a general logistic model for disease risk prediction using multiple SNPs," FEBS Open Bio, vol. 9, no. 11, pp. 2006–2012, 2019.

[6] W. R. Swindell, A. Johnston, J. J. Voorhees, J. T. Elder, and J. E. Gudjonsson, "Dissecting the psoriasis transcriptome: Inflammatory- and cytokine-driven gene expression in lesions from 163 patients," BMC Genomics, vol. 14, no. 1, 2013.

[7] N. Q. K. Le, D. T. Do, T.-T.-D. Nguyen, N. T. K. Nguyen, T. N. K. Hung, and N. T. T. Trang, "Identification of gene expression signatures for psoriasis classification using machine learning techniques," Med. Omi., vol. 1, no. December 2020, p. 100001, 2020.

[8] M. Suárez-Fariñas, K. Li, J. Fuentes-Duculan, K. Hayden, C. Brodmerkel, and J. G. Krueger, "Expanding the psoriasis disease profile: Interrogation of the skin and serum of patients with moderate-to-severe psoriasis," J. Invest. Dermatol., vol. 132, no. 11, pp. 2552–2564, 2012.

[9] X. Wang, X. Liu, N. Liu, and H. Chen, "Prediction of crucial epigenetically-associated, differentially expressed genes by integrated bioinformatics analysis and the identification of S100A9 as a novel biomarker in psoriasis," Int. J. Mol. Med., vol. 45, no. 1, pp. 93–102, 2020.

[10] B. Li et al., "Transcriptome analysis of psoriasis in a large case-control sample: RNA-seq provides insights into disease mechanisms," J. Invest. Dermatol., vol. 134, no. 7, pp. 1828–1838, 2014.

[11] C. R. Jack and D. M. Holtzman, "Biomarker modeling of alzheimer's disease," Neuron, vol. 80, no. 6, pp. 1347–1358, 2013.

[12] J. Hippisley-Cox, C. Coupland, and P. Brindle, "Development and validation of QRISK3 risk prediction algorithms to estimate future risk of cardiovascular disease: Prospective cohort study," BMJ, vol. 357, no. May, pp. 1–21, 2017.

[13] J. Lea and S. Climer, "A Search and Filter Strategy for Identifying Differentially Co-Expressed Analyte Modules," Proc. - 2020 IEEE Int. Conf. Bioinforma. Biomed. BIBM 2020, pp. 2974–2976, 2020.

[14] N. J. Wald and R. Old, "The illusion of polygenic disease risk prediction," Genet. Med., vol. 21, no. 8, pp. 1705–1707, 2019.

[15] C. Bennette and A. Vickers, "Against quantiles: Categorization of continuous variables in epidemiologic research, and its discontents," BMC Med. Res. Methodol., vol. 12, 2012.

[16] S. Climer, "COVID-19 and the differential dilemma," Patterns, vol. 2, no. 5, May 2021, doi: 10.1016/J.PATTER.2021.100260.

[17] M. Ojala and G. C. Garriga, "Permutation Tests for Studying Classifier Performance," J. Mach. Learn. Res., vol. 11, pp. 1833–1863, Aug. 2010.

[18] S. Climer, "COVID-19 and the differential dilemma," PATTER, vol. 2, no. 5, May 2021, doi: 10.1016/j.patter.2021.100260.