Title

Uncovering Effective Explanations for Interactive Genomic Data Analysis

Journal

Patterns, 1(6)

ISSN

2666-3899

Authors

Huang, Silu
Blatti, Charles
Sinha, Saurabh
et al.

Publication Date

2020-09-01

DOI

10.1016/j.patter.2020.100093

Peer reviewed

Article

Patterns

Uncovering Effective Explanations for Interactive Genomic Data Analysis

Highlights

- Finding feature pairs that separate object classes in genomic datasets is important
- Our interactive GENVISAGE tool rapidly identifies and visualizes these feature pairs
- Several optimizations make GENVISAGE up to 400× faster than baseline approaches
- GENVISAGE finds supported gene pairs that discriminate between drug treatments

Authors

Silu Huang, Charles Blatti, Saurabh Sinha, Aditya Parameswaran

Correspondence

adityagp@berkeley.edu

In Brief

Identifying features that most strongly separate samples from two biological classes is fundamental in the analysis of genomic datasets. This task is typically addressed by finding (1) single features using univariate statistical methods or (2) multi-feature combinations from time-intensive machine learning. Here we present GENVISAGE, a tool that enables researchers to interactively identify visually interpretable and significant feature pairs that separate the classes. With this highly optimized tool, researchers can instantaneously generate and explore hypotheses on very massive genomic datasets.

Huang et al., 2020, Patterns *1*, 100093
September 11, 2020
https://doi.org/10.1016/j.patter.2020.100093

CellPress

Patterns

CellPress
OPEN ACCESS

Article

Uncovering Effective Explanations for Interactive Genomic Data Analysis

Silu Huang,[1,4] Charles Blatti,[2,4] Saurabh Sinha,[1,2] and Aditya Parameswaran[1,3,5,*]
[1]Department of Computer Science, University of Illinois Urbana-Champaign, Urbana, IL 61801, USA
[2]Institute of Genomic Biology, University of Illinois Urbana-Champaign, Urbana, IL 61801, USA
[3]School of Information and Department of Electrical Engineering and Computer Sciences, University of California Berkeley, Berkeley, CA 94704, USA
[4]These authors contributed equally
[5]Lead Contact
*Correspondence: adityagp@berkeley.edu
https://doi.org/10.1016/j.patter.2020.100093

THE BIGGER PICTURE A fundamental task in the analysis of genomics datasets is identifying features that can explain the difference between two groups of biological samples. As studies and data repositories that enable simultaneous analysis of thousands of samples become widespread, it is imperative that feature identification tools return interpretable and significant results rapidly, allowing researchers to interactively generate and explore hypotheses on these massive datasets. Our tool, GENVISAGE, is built around a framework that identifies pairs of features that strongly separate samples of different classes. An extensive suite of optimization techniques enables us to extract literature-supported feature pairs with accompanying interpretable visualizations from exceptionally large genomic datasets in real time. The GENVISAGE optimizations and webserver instance provide a blueprint for future online tools providing interactive feature exploration in massive datasets from genomics and other domains.

Proof-of-Concept: Data science output has been formulated, implemented, and tested for one domain/problem

SUMMARY

Better tools are needed to enable researchers to quickly identify and explore effective and interpretable feature-based explanations for discriminating multi-class genomic datasets, e.g., healthy versus diseased samples. We develop an interactive exploration tool, GENVISAGE, which rapidly discovers the most discriminative feature pairs that separate two classes of genomic objects and then displays the corresponding visualizations. Since quickly finding top feature pairs is computationally challenging, especially for large numbers of objects and features, we propose a suite of optimizations to make GENVISAGE responsive at scale and demonstrate that our optimizations lead to a 400× speedup over competitive baselines for multiple biological datasets. We apply our rapid and interpretable tool to identify literature-supported pairs of genes whose transcriptomic responses significantly discriminate several chemotherapy drug treatments. With its generalizable optimizations and framework, GENVISAGE opens up real-time feature-based explanation generation to data from massive sequencing efforts, as well as many other scientific domains.

INTRODUCTION

A common approach to discovery in biology is to construct experiments or analyses that directly contrast two specific classes of biological objects. Examples of this approach include examining patient samples contrasting tumor versus normal tissue,[1] studying the differences in molecular effects of two competing drug treatments,[2] or characterizing differentially expressed genes versus genes with unaltered gene expression in a carefully designed experiment.[3] To understand the mechanisms that determine these object classes, researchers often employ statistical and machine-learning tools to identify a manageable subset

of features, e.g., genes, that accentuate, discriminate, or help explain the differences between classes, i.e., separate the two classes. We refer to this problem as the separability problem, and many important scientific applications can be abstracted as this problem.

Tools have been developed in several different biological settings[4–9] for the separability problem, by focusing on discovering pairs of features that taken together strongly discriminate the classes. Feature pair methods can provide a better characterization of what distinguishes two object classes by offering insights into the interplay between important features that would not be found using single-feature statistical tests[10] or univariate classifiers.[11] Specifically, predictors built with gene feature pairs are more robust to normalization and can achieve better model performance than predictors using single genes as features.[5,6] On the other hand, methods focused on feature pairs offer the advantage of providing more interpretable or explainable results over more complicated machine-learning approaches that return a complex combination of several features to discriminate the classes, such as multivariate regression with LASSO regularization[12] or pattern mining from random forest models.[13] Some existing papers[7,14–16] employ these more complex machine-learning approaches to heuristically return more interpretable feature pairs. However, these heuristic methods do not fully explore the search space nor do they offer a guarantee on the quality of the returned feature pairs.

The major downside with current methods that address the separability problem with either feature pairs or more complex machine-learning models is that they do not scale to the growing size of genomic datasets. As is often the case with genomics, the biological objects being analyzed (e.g., tissue samples or drug experiments) are frequently represented by high-dimensional numeric feature vectors (e.g., transcript abundance measurements). Additionally, with the rise of low-cost sequencing, the possible number of biological objects in a dataset is also increasing and likely to grow in orders of magnitude over the next decade.[17] Applying these standard methods to datasets with tens of thousands of objects and features results in massive running times that preclude interactive exploration of the data. For example, exhaustively searching for the optimal feature pairs from the full space of possibilities in a typical genomic analysis resulted in running times over an hour on a 200-node compute cluster in Watkinson et al.[9]

One reason that more complex machine-learning and feature-pair-based methods do not scale well with the number of features and objects is the selection of the metric for scoring separability. In Watkinson et al.,[9] a metric called synergy is proposed for evaluating the utility of feature pairs, aiming to capture both linear and non-linear aspects of the separability of the two class, leading to the aforementioned long running times. Consequently, the intrinsic complexity of these metrics makes them difficult to benefit from optimization techniques. Metrics based only on quantifying linear separability, on the other hand, may return a more limited subset of interesting features, but they also may be more intuitive for users to understand and simultaneously enable more performance optimizations and speedups. The linear separability metric has been used in previous studies to identify pairs of genes with expression differences between two cancer types[18] or pairs of motifs that discriminate between different types of genomic sequences.[19] However, the linear separability metric defined in the former study[18] is either zero or one, while in our tool we employ a soft metric, ranging from zero to one, to better measure the linear separability; the latter[19] is not focused on the efficiency optimizations for large-scale scenarios.

Motivated by these observations, we present GENVISAGE, an interactive data-exploration tool designed to provide effective explanations to address the separability problem and scale to the size of large genomic analysis datasets. With GENVISAGE, we not only achieve high separability with our carefully formulated objective but also enable explanations regarding separation via intuitive visualizations, and, at the same time, we can handle large datasets efficiently—the best of all three worlds. Specifically, to enable this scalability, GENVISAGE focuses on returning the top-ranking feature pairs that discriminate the objects of separate classes, rather than returning larger subsets of features using more complexity and longer times to train machine-learning approaches. GENVISAGE is also based around a linear separability metric that provides an intuitive interpretation for feature pairs while enabling and simplifying the design of several important performance optimizations. These optimizations include: (1) elimination of repeated computation for different features pairs; (2) pruning poor ranking pairs during early execution; (3) sampling with a quality guarantee to further reduce running time; and (4) cleverly traversing the search space of feature pairs for improved efficiency. To the best of our knowledge, this type of interactive data-exploration tool is relatively underexplored compared with other areas of visualization in biology. Specifically, there is work on biological network visualization[20,21] and biological time series visualization,[22] but not a lot of work on visualizing experimental data. Some related work[23,24] performs dimensionality reduction for single-cell transcriptomics and visualizes the global structure of the data in two dimensions, whereas a tool like GENVISAGE extracts the most relevant pairs of features that explain the separation of two object sets and then displays the related data in an interpretable visualization.

We applied GENVISAGE to two large genomic datasets with tens of thousands of objects and high-dimensional feature vectors where it is computationally expensive to score the separability for all possible feature pairs. In one, called LINCS, we find pairs of genes whose expression discriminates between perturbagen experiments involving different drug treatments, and in the other, called MSIGDB, we find pairs of annotations (such as pathway membership) that separate differentially expressed cancer genes from other genes. With the carefully designed separability metric of GENVISAGE and its suite of sophisticated optimizations that accelerates evaluation, we are able to accurately return the highest-ranking separating feature pairs for both datasets within 2 min on a single machine. This reflects a 180× and 400× speedup over a competitive baseline for the MSIGDB and LINCS datasets, respectively. We also show that the feature pairs identified by GENVISAGE often more significantly discriminate between the object classes than the corresponding best-ranking individual features, even after accounting for the larger search space. Finally, we performed

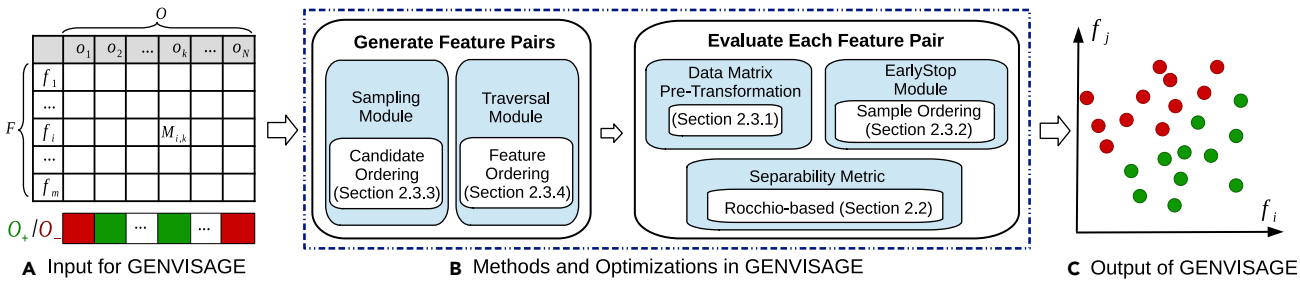

Figure 1. GENVISAGE Workflow

Given (left) a feature-object matrix and green positive and red negative class labels on the objects, GENVISAGE (center) evaluates all pairs of features using several optimizations to identify (right) the top feature pair and its corresponding visualization that best separates the object classes.

an in-depth analysis of nine distinct drug treatments in the LINCS dataset and found 1,070 feature (gene) pairs that had significant separability scores. These gene pairs were enriched in literature support for known relationships between the genes and the drug, as well as known interactions between the genes themselves.

To summarize, GENVISAGE offers researchers the ability to gain additional insights into their object classes beyond singular features without the prolonged duration needed to train a complex machine-learning model. By implementing optimizations that take advantage of a linear separability metric, GENVISAGE enables researchers to quickly explore their data, identify the strongest, most compelling features, and from simple visualizations form hypotheses about the interplay between features and with the object classes. The performance of our tool also allows researchers to investigate multiple definitions of the object classes and investigate alternative hypotheses interactively on the fly, as well as build a feature set to pass on to more in-depth, longer-running machine-learning-based analysis.

METHODS

We begin by formally defining the separability problem, introducing our separability metric, and finally detailing optimizations that enable the rapid identification of the best separating feature pairs.

Problem Definition

Let $\mathcal{M}$ be a feature-object matrix of size $m \times N$, where each row is a feature and each column is an object as shown in Figure 1. One example feature-object matrix is one where each object corresponds to a tissue sample from a cancer patient and each feature corresponds to a gene, where the $(i,j)^{\text{th}}$ entry represents the expression level of the i^{th} gene in the j^{th} tissue sample. We denote the m features as $\mathcal{F} = \{f_1, f_2, \cdots, f_m\}$ and N objects as $\mathcal{O} = \{o_1, o_2, \cdots, o_N\}$. Each entry $\mathcal{M}_{i,j}$ in $\mathcal{M}$ corresponds to the value of feature f_i for object o_j as illustrated in Figure 1.

We are also given two non-overlapping sets of objects, one with a positive label, $\mathcal{O}_+$ and the other with a negative label, $\mathcal{O}_-$. In our example, tumor samples, $\mathcal{O}_+$, may be assigned the positive label, and the healthy tissue samples, $\mathcal{O}_-$, the negative label. The number of labeled objects, n, is equal to $|\widehat{\mathcal{O}}|$ where $\widehat{\mathcal{O}} = \mathcal{O}_+ \cup \mathcal{O}_-$. Also, let l_k be the label of object $o_k \in \widehat{\mathcal{O}}$, i.e., $l_k = 1$ if o_k is positive and $l_k = -1$ if o_k is negative.

GENVISAGE aims to find feature pairs that best separate the objects in $\mathcal{O}_+$ from those in $\mathcal{O}_-$ using only those features, and then output a visualization that demonstrates the separability. (We will define the metric for separability subsequently.) A feature pair that leads to a good "visual" separation between the positive and the negative sets may be able to explain or characterize their differences via an interesting, non-trivial relationship among the features. The overall workflow is depicted in Figure 1. We now formally define the separability problem.

Problem 1 (Separability)

Given a feature-object matrix $\mathcal{M}$ and two labeled object sets $(\mathcal{O}_+, \mathcal{O}_-)$, identify the top-k feature pairs (f_i, f_j) that separate $\mathcal{O}_+$ from $\mathcal{O}_-$ based on a given separability metric.

We will describe our separability metric in the next section, followed by a discussion of optimization techniques. The notation used in the description of the method is summarized in Table S3.

Separability Metric

Given a feature pair (f_i, f_j) as axes, we can visualize the object sets $\mathcal{O}_+$ and $\mathcal{O}_-$ in a two-dimensional (2D) space, where each object corresponds to a point with x value and y value as the object's value on feature f_i and f_j, respectively. A desirable (i.e., both interesting and interpretable) visualization would be one in which the objects are linearly separated, defined as follows. Two sets of objects, i.e., $\mathcal{O}_+$ and $\mathcal{O}_-$, are said to be linearly separable[25] if there exists at least one straight line such that $\mathcal{O}_+$ and $\mathcal{O}_-$ are on opposite side of it. We focus on metrics that capture this linear separation, since it corresponds to an intuitive 2D visualization. Given a feature pair (f_i, f_j) and a line ℓ, we can predict the label of an object o_k, denoted as $\eta_{ij}^{\ell,k}$, using Equation 1, where w_0, w_i and w_j are coefficients of ℓ and $w_j > 0$:

$$\text{Predicted Label}: \quad \eta_{ij}^{\ell,k} = sign(w_i \cdot \mathcal{M}_{i,k} + w_j \cdot \mathcal{M}_{j,k} + w_0). \qquad \text{(Equation 1)}$$

If o_k lies above the line ℓ, i.e., o_k has higher value on the y axis than the point on line ℓ with the same value on the x axis as o_k, then $\eta_{ij}^{\ell,k} = 1$; otherwise, $\eta_{ij}^{\ell,k} = -1$. Let $\theta_{ij}^{\ell,k}$ be the indicator variable denoting whether the sign of the predicted label matches the real label l_k: if $\eta_{ij}^{\ell,k} \cdot l_k = 1$, then $\theta_{ij}^{\ell,k} = 1$; otherwise, $\theta_{ij}^{\ell,k} = 0$.

GENVISAGE's separability metric captures how well the objects in the feature pair's 2D visualization can be linearly separated, formally defined next. Given a feature pair (f_i, f_j) and a line ℓ, the separability score of the line (denoted θ_{ij}^{ℓ}) is defined as the sum of the indicators ($\theta_{ij}^{\ell,k}$) for all objects: $\theta_{ij}^{\ell} = \sum_k \theta_{ij}^{\ell,k}$. Figure 2A shows separability scores θ_{ij}^{ℓ} for different separating lines. For example, the separating line with $\theta_{ij}^{\ell} = 12$ correctly separates six green points and six red points. The final separability score for a feature pair (f_i, f_j) (denoted as θ_{ij}) is defined as the best separability score θ_{ij}^{ℓ} among all possible lines ℓ. Accordingly, we define the overall separability error of the feature pair as $err_{ij} = n - \theta_{ij}$.

Brute-Force Calculation of θ_{ij}

As suggested in Figure 2A, the simplest way to calculate θ_{ij} is to first enumerate all possible separating lines ℓ and calculate θ_{ij}^{ℓ} for each of them. We can easily trim down the search space to $O(n^2)$ lines by linking the points corresponding to every two objects in the 2D plane. This is because the results of all other possible lines can be covered by these $O(n^2)$ lines.[26] Nevertheless, it is still very time consuming to consider $O(n^2)$ lines for each feature pair (f_i, f_j).

Rocchio-Based Measure

We can speed up the process by selecting a single representative line L providing us with an estimate of the true separability score θ_{ij}. To achieve a

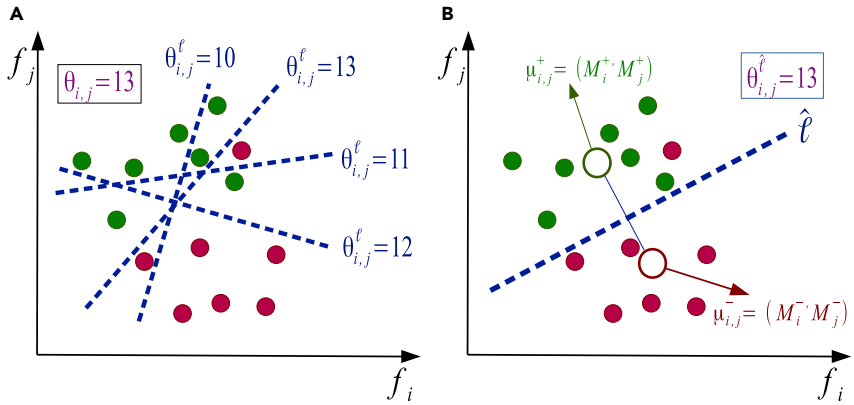

Figure 2. Calculating Separability Score θ_{ij}

The scored separating line can be defined using (A) brute force (few sample lines are shown) or (B) the representative line from a Rocchio-based measure based on the object class centroids (white circles).

fast and reliable estimate, we select our representative line based on Rocchio's algorithm.[27] Let us denote the centroids of positive objects $\mathcal{O}_+$ and negative objects $\mathcal{O}_-$ for a given (f_i, f_j) as $\mu^+_{ij} = (\mathcal{M}^+_i, \mathcal{M}^+_j)$ and $\mu^-_{ij} = (\mathcal{M}^-_i, \mathcal{M}^-_j)$, respectively, where $\mathcal{M}^+_i$ and $\mathcal{M}^+_j$ are the values of the centroids of the positive objects on feature f_i and f_j, and $\mathcal{M}^-_i$ and $\mathcal{M}^-_j$ are the values of the centroids of the negative objects on feature f_i and f_j. The perpendicular bisector of the line joining the two centroids is selected as the representative separating line L (see Figure 2B), with its coefficients corresponding to Equation 1 defined as $w_i = \mathcal{M}^+_i - \mathcal{M}^-_i$, $w_j = \mathcal{M}^+_j - \mathcal{M}^-_j$, and $w_0 = -\left(\frac{(\mathcal{M}^+_i)^2-(\mathcal{M}^-_i)^2}{2} + \frac{(\mathcal{M}^+_j)^2-(\mathcal{M}^-_j)^2}{2}\right)$.

Brute Force versus Rocchio Based

Compared with the brute-force calculation, the Rocchio-based measure is much more lightweight, but at the cost of accuracy in calculating θ_{ij}. Intuitively, the representative line is a reasonable proxy to the best separating line since the Rocchio-based measure assigns each object to its nearest centroid. We further empirically demonstrate that θ^L_{ij} is a good proxy for θ_{ij} in the section Comparison of Different Algorithms. Thus, we will focus on the Rocchio-based measure subsequently, removing L (or ℓ) from the superscripts where it appears, and using θ_{ij} and θ^L_{ij} interchangeably.

Proposed Suite of Optimizations

In this section, we first analyze the time complexity of identifying the top-k feature pairs using the Rocchio-based measure and then propose several optimization techniques to reduce the complexity.

Time Complexity Analysis

For a given feature pair (f_i, f_j), if we have already calculated the class centroids for each feature, the separating line L can be calculated in $O(1)$. We can then calculate the number of correctly separated objects θ_{ij} via $O(n)$ evaluations. Since there are $O(m^2)$ feature pair candidates, the total time complexity is $O(m^2 n)$, which can be very large, since m and n are typically large.

Optimizations: Overview

To reduce the time complexity, we introduce two categories of optimizations: those that reduce the amount of time for fully evaluating a given feature pair (see Pre-transformation for Faster Feature Pair Evaluation and Early Termination) and those that reduce the number of feature pairs that require full evaluation (see Sampling-Based Estimation and Search Space Traversal). In the following, we refer to these optimizations as modules to indicate that they can be used in any combination—however, in reality, careful engineering is necessary to "stitch" these modules together to multiply the effects of the optimizations.

The TRANSFORMATION module (see Pre-transformation for Faster Feature Pair Evaluation) reduces redundant calculations across feature pairs by mapping the feature-object matrix $\mathcal{M}$ into a new space that enables faster evaluation of object labeling. The EARLYSTOP module (see Early Termination) takes advantage of the fact that evaluation of a poorly separating feature pair can be terminated early without having to evaluate the separability of all n objects.

The SAMPLING module (see Sampling-Based Estimation) first identifies likely top-k feature pair candidates by evaluating their separability on a sampled subset of all objects and then conducts full evaluations only on these feature pair candidates. Finally, the TRAVERSAL module (see Search Space Traversal) reduces the number of feature pairs checked by greedily choosing feature pairs based on the separability of the corresponding single features. These optimization modules can be used on their own or combined with each other. In Results, we will show how these optimization modules greatly reduce the running time of finding the top-k separating feature pairs without significantly affecting the accuracy.

Pre-transformation for Faster Feature Pair Evaluation

We observe that there is massive redundancy across θ_{ij}'s computation of different feature pairs. Motivated by this, we propose the TRANSFORMATION module, which will pre-calculate some common computational components once across different features and reuse these components in evaluating the separability for each different feature pair. This TRANSFORMATION module transforms the original $\mathcal{M}_{i,k}$ matrix into another space $\widehat{\mathcal{M}}_{i,k}$ using the identified common feature pair components and updates the separability score equation accordingly. Specifically, with this transformation of the feature-object matrix $\widehat{\mathcal{M}}_{i,k}$, evaluating whether an object was correctly separated is simplified as: if $sign(\widehat{\mathcal{M}}_{i,k} + \widehat{\mathcal{M}}_{j,k}) = 1$, then $\theta^k_{ij} = 1$; otherwise, $\theta^k_{ij} = 0$. Details and an example can be found in Supplemental Experimental Procedures and Figure S1.

Early Termination

Given a feature pair (f_i, f_j), we need to scan all the objects to compute the separability score θ_{ij}. However, since we only need to identify feature pairs in the top-k, we can stop for each feature pair as soon as we can make that determination without scanning all objects; we call this the EARLYSTOP module.

High-Level Idea. We maintain an upper bound τ for the separability error err_{ij} of the top-k feature pairs. Then, the lower bound of the separability score can be denoted as $(n - \tau)$. Given a feature pair (f_i, f_j), we start to scan the object list until the number of incorrectly classified objects exceeds τ. If so, we can terminate early and prune this feature pair since it cannot be among the top-k. Otherwise, (f_i, f_j) is added to the top-k feature pair set and we update τ accordingly.

Enhancement by Object Ordering. Although EARLYSTOP has the potential to always reduce the running time, its benefits are sensitive to the ordering of the objects for evaluation. Since we terminate as soon as we find τ incorrectly classified objects, we can improve our running time if we examine "problematic" objects that are unlikely to be correctly classified relatively early. For this, we order the objects in descending order of the number of single features f_i that incorrectly classify the object o_k, i.e., $\widehat{\mathcal{M}}_{i,k} \leq 0$. Thus, the first object evaluated is the one that is incorrectly classified by the most single features. The benefit of this strategy is illustrated with an example in Figure 3A.

Sampling-Based Estimation

One downside of the EARLYSTOP module is that the improvement in the running time is highly data dependent. Here, we propose a stochastic method, called SAMPLING, which reduces the number of examined objects. Instead of calculating θ_{ij} over the whole object set $\widehat{\mathcal{O}}$, SAMPLING works on a sample set drawn from $\widehat{\mathcal{O}}$.

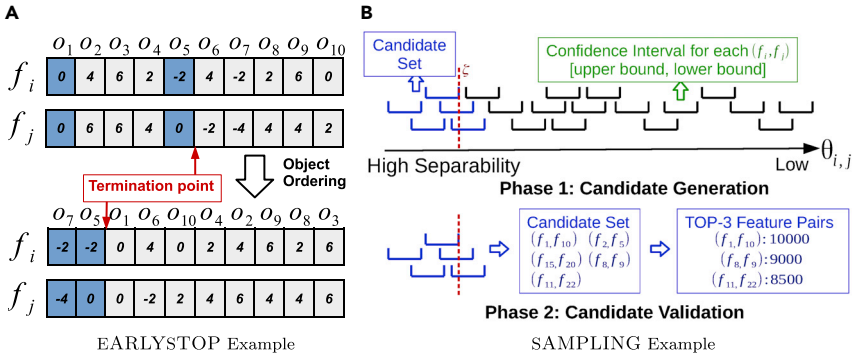

Figure 3. Optimization Module Examples

(A) When evaluating a feature pair with EARLYSTOP module, the transformed $\widehat{\mathcal{M}}$ scores are scanned left to right and each incorrectly classified object is marked in blue. Without object ordering (above), evaluation terminates after five checked objects. When objects are reordered by the most "problematic" (below), the feature pair is rejected after checking only the first two objects.

(B) To calculate the top three feature pairs with SAMPLING, the confidence interval of θ_{ij} is calculated for every feature pair evaluated on the sample set $\mathcal{S}$ (above). The third interval lower bound ζ is obtained (red dotted line), and all feature pairs with a larger upper bound are designated as candidates for validation (blue intervals). The selected candidates (center box) are evaluated on the whole object set $\widehat{\mathcal{O}}$ to compute the exact θ_{ij} and pick the top three (right box).

High-Level Idea. SAMPLING primarily consists of two phases: candidate generation and validation (Figure 3B). In phase 1, we estimate the confidence interval of θ_{ij} for each feature pair using a sampled set of objects and generate the candidate feature pairs for full evaluation based on where their confidence intervals lie. If the confidence interval overlaps with the score range of the current top-k, then it is selected for evaluation. In phase 2 (lower half of Figure 3B), we evaluate only the feature pairs in the candidate set, calculating θ_{ij} over the whole object set, $\widehat{\mathcal{O}}$, to obtain the final top-k feature pairs. Unlike our previous optimizations, SAMPLING returns an approximation of the top-k ranking feature pairs.

Candidate Generation. Let $\mathcal{S}$ be a sample set drawn uniformly from $\widehat{\mathcal{O}}$. Given a feature pair (f_i, f_j), let $\theta_{ij}(\mathcal{S})$ be the number of correctly separated objects in $\mathcal{S}$. We can estimate $\tilde{\theta}_{ij}$ from $\theta_{ij}(\mathcal{S})$ using $\tilde{\theta}_{ij} = \frac{\theta_{ij}(\mathcal{S})}{|\mathcal{S}|} \cdot n$ by assuming the ratio of correctly separated samples in $\mathcal{S}$ is the same as that in $\widehat{\mathcal{O}}$. Using Hoeffding's inequality,[28] we can show that by selecting $\Omega\left(\frac{1}{\varepsilon^2}\right)$ samples, θ_{ij} is within the confidence interval $[\tilde{\theta}_{ij} - \varepsilon n, \tilde{\theta}_{ij} + \varepsilon n]$ with high probability (for details see Supplemental Experimental Procedures). Since the sample size $|\mathcal{S}|$ is independent of the number of objects, this module helps GENVISAGE scale to datasets with large n.

Following the top half of Figure 3B, we can first calculate the confidence interval of θ_{ij} for each feature pair (f_i, f_j). Next, we compute the lower bound of θ_{ij} for the top-k feature pairs, denoted as ζ as shown by the red dotted line. Finally, we can prune feature pairs away whose upper bound is smaller than ζ, keeping the candidate set $\mathcal{C}$ of feature pairs depicted by blue intervals. These feature pairs $\mathcal{C}$ will be further validated in phase 2, i.e., candidate validation. Typically, $|\mathcal{C}|$ will be orders of magnitude smaller than m^2, the original search space for all feature pairs.

Candidate Validation. We re-evaluate all of the candidates generated from phase 1 to produce our final feature pair ranking. This evaluation is performed using the whole object set $\widehat{\mathcal{O}}$ and the top-k feature pairs are reported (lower half of Figure 3B).

Enhancement by Candidate Ordering. In Early Termination we proposed an enhancement that allows us to terminate computation early by manipulating the order of the objects; here, we similarly found a way to reduce the running time by changing the order in which feature pair candidates are validated in phase 2. Instead of directly validating each feature pair candidate, we first order the candidates in descending order according to the upper bound of each candidate's confidence interval. We then sequentially calculate the full separability score θ_{ij} for each feature pair and update ζ correspondingly. Recall that ζ is the current estimate of the lower bound of θ_{ij} for the top-k feature pairs. Finally, we terminate our feature pair validation when the next feature pair's upper bound is smaller than the current value of ζ (Figure 4).

Search Space Traversal

The optimizations discussed so far check fewer than n objects for each feature pair and reduce the number of feature pairs for full evaluation. Our TRAVERSAL module aims to reduce the number of feature pairs considered from m^2 to a smaller number. Instead of examining each feature pair, we only examine a limited number of feature pairs, but in an optimized traversal order. The number of examined feature pairs, χ, determines a trade-off between efficiency and accuracy. Fewer feature pairs checked will result in faster running times, though at the cost of accuracy to the top-k. The order of the feature pairs must be determined carefully, and we propose two alternative orderings based on the ranking of single features by their separability scores $\theta_{i,i}$. The first traversal order, called horizontal traversal, prioritizes feature pairs that have at least one high-ranking single feature in the considered feature pair. The second order, called vertical traversal, prioritizes feature pairs where both features have high single-feature score rankings (see Figure S2 for more details and an example).

RESULTS

In this section, we illustrate that GENVISAGE rapidly identifies meaningful, significant, and interesting separating feature pairs in real biological datasets. First, we describe the datasets and the algorithms used in our evaluation. Each algorithm that we evaluate represents a combination of optimization modules for ranking top-k feature pairs using our Rocchio-based measure—we report the running time and accuracy of the algorithms. Second, we compare the top-k feature pairs returned by GENVISAGE with the corresponding top-k single features, and examine their significance and support in existing publications. Lastly, we present some sample visualizations to illustrate the separability of the object classes.

Evaluation Setup

Datasets

We consider datasets from two biological applications (see Table 1): (1) in MSIGDB, we find gene annotations such as pathways and biological processes that separate the differentially expressed genes from the undisturbed genes in specific cancer studies; (2) in LINCS, we find genes whose expression levels can distinguish experiments in which specific drug treatments were administered from others.

In MSIGDB, we are given a feature-object matrix with genes as the objects and gene properties as the features. Rather than being a 0/1 membership indicator matrix, the values of this feature-object matrix indicate the strength of the relationship between the gene and the set of genes that have been annotated with the gene property. Matrix values are calculated using random walks[29] on a heterogeneous network built from prior knowledge found in gene annotation and protein homology databases (see

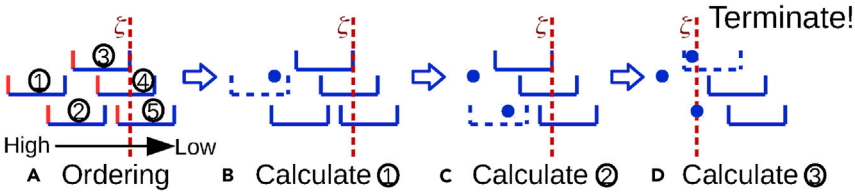

Figure 4. Candidate Ordering Enhancement

(A) Feature pair candidates are sorted by the upper bounds of their confidence intervals (solid red boundary), and the lower bound of the top three feature pairs, i.e., ζ, is set (red dotted line).

(B–D) For each feature pair, we calculate θ_{ij} (filled blue circle) using all objects and update ζ if necessary. Note that ζ is increased in (D) after evaluating the third feature pair and, since ζ is larger than the upper bound of the fourth feature pair, candidate validation can terminate and return the top-ranking pairs.

Supplemental Experimental Procedures for more details). The positive genes for each dataset in MSIGDB are the set of differentially expressed genes (DEGs) in a specific cancer study downloaded from the Molecular Signatures Database (MSIGDB).[30] Each of our tests is an application of GENVISAGE to such a dataset, reporting pairs of properties that separate DEGs of that cancer study (the "positive" set) from all other genes (the "negative" set).

In LINCS, the feature-object matrix contains expression values for different genes (features) across many drug treatment experiments (objects) conducted on the MCF7 cell line by the LINCS L1000 project.[31] The values of the matrix are gene expression values as reported by the "level-4" imputed Z scores measured in the L1000 project. In each dataset, the positive object set includes multiple experiments that used the same drug, at varying dosages and for varying durations. We applied GENVISAGE on each dataset so as to find the top pairs of genes (feature pairs) whose expression values separate the LINCS experiments relating to a single drug from all other LINCS experiments.

Note that the average number of positive objects in any dataset is far fewer than the average number of negative objects. To address this imbalance, we adjust θ_{ij}^{ℓ} to a weighted sum form: $\theta_{ij}^{\ell} = \sum_{o_k \in \mathcal{O}_-} \theta_{ij}^{\ell,k} + \frac{|\mathcal{O}_-|}{|\mathcal{O}_+|} \cdot \sum_{o_k \in \mathcal{O}_+} \theta_{ij}^{\ell,k}$.

Algorithms

We evaluated six combinations of our optimization modules from the section Proposed Suite of Optimizations, listed in Table 2. For our baseline, we use the algorithm with only the matrix pre-transformation optimization module (TRANSFORMATION). The rightmost column of Table 2 shows the varying time complexity of the algorithms. Consider the HORIZSAMPOPT as an example. First, TRANSFORMATION takes $O(mn)$ time. Then, TRAVERSAL requires a sorting over the feature set, taking $O(m\log m)$ time. Finally, with SAMPLING over χ feature pairs, the running time is reduced from $O(m^2 n)$ time to $O(\chi|\mathcal{S}| + |\mathcal{C}|n)$ time, where the first and second terms represent the time for candidate generation and candidate validation, respectively. Note that $|\mathcal{C}|$ is typically orders of magnitude smaller than χ in HORIZSAMPOPT, as discussed in Sampling-Based Estimation. Combinations of modules beyond the six reported were always inferior to one of those shown in the sense that they returned the same top-k feature pairs and had a longer running time. We implemented the algorithms in C++, and conducted the evaluations on a machine with 16 CPUs and 61.9 GB of RAM.

Comparison of Different Algorithms

In this section, we first justify that Rocchio-based measure is a good proxy for the best possible separating score computed by a brute-force method. We then compare the performance of the algorithms in terms of the running time and the separability of their top 1,000 feature pairs.

Accuracy of Rocchio-Based Approximation

As discussed in Separability Metric, when using brute force we need to consider $O(n^2)$ lines in order to find the best separating line $\ell^* \leftarrow \arg_\ell \max\{\theta_{ij}^{\ell}\}$, with a time complexity of $O(n^2 m^2)$ when considering all feature pairs. An alternative is to use Rocchio-based representative separating line L, dramatically reducing $O(n^2)$ lines considered to $O(1)$. To give some concrete numbers, we attempted to run the brute-force approach using only $n = 600$ objects and $m = 150$ features, and it took more than 3 h for the run to complete. Note that in our experiments on real data as shown in Table 1, both m and n are around 20K for each single dataset in MSIGDB (m and n are even larger in LINCS), which when extrapolated suggests it would take around 2×10^7 hours using the brute-force approach for one dataset. Since the brute-force method becomes computationally infeasible for datasets with large n, we compared the Rocchio-based measure with the brute-force-based measure using specially defined small object sets, $\widehat{\mathcal{O}}$, for the ten datasets in MSIGDB. For this comparison, the upregulated genes in each MSIGDB test was defined as the set of positive objects and the downregulated genes as the set of negative objects, resulting in an average number of 295 objects for each comparison. We call the brute-force-based separability score the true separability score, since it examines all possible separating lines. We first find the best feature pair using Rocchio-based measure and the brute-force-based measure separately (potentially different feature pairs) and then calculate the ratio of the true separability scores of the Rocchio versus the brute-force best feature pairs. We observe that the Rocchio-based method picks a best feature pair that has true separability score similar to the best pair picked by brute force, with the ratio of the two scores being better than 0.94 in all ten datasets (Figure S3a). Second, for the best feature pairs identified by Rocchio-based method for the ten datasets, we calculate the ratio of the Rocchio-based separability score and the brute-force-based separability score, and find the difference to be greater than 0.96 on average (Figure S3b).

Running Time

Figure 5 depicts the running times of our different selected algorithms. Each plotted box corresponds to one algorithm, representing the distribution of running times for finding the top-k feature pairs (by Rocchio score) for all datasets.

First, let us compare the median running times among different algorithms. For MSIGDB, the BASELINE takes more than 2 h, EARLYORDERING takes less than 1 h, SAMPONLY and SAMPOPT take around 6 min and 5 min, respectively, while HORIZSAMPOPT and

Table 1. Dataset Statistics

	$\lvert\mathcal{F}\rvert = m$	$\lvert\mathcal{O}\rvert = N$	$\lvert\mathcal{S}\rvert$	χ	# of $\widehat{\mathcal{O}}$	avg($\lvert\mathcal{O}_+\rvert$)	avg($\lvert\mathcal{O}_-\rvert$)
MSIGDB	19,912	22,209	400	10^7	10	295	21,914
LINCS	22,268	98,061	400	10^7	40	165	97,897

For each dataset, the number of features m, objects N, sample size $\lvert\mathcal{S}\rvert$ used by SAMPLING module, feature pairs χ examined by TRAVERSAL module, number of object sets # of $\widehat{\mathcal{O}}$, average positive set size avg($\lvert\mathcal{O}_+\rvert$), and average negative set size avg($\lvert\mathcal{O}_-\rvert$).

VERTSAMPOPT both take only 1 min on average. Overall, the optimizations result in a reduction of the running time by over 180 × . We next examine the effect of different modules on the running time. (1) EARLYSTOP: we observe that the EARLYSTOP module helps achieve a 2× speedup, with the average number of checked objects (genes) reduced from 20K to 5K (Table S1); (2) SAMPLING: the SAMPLING module helps reduce the running time dramatically, with 20× reduction from BASELINE to SAMPOPT, since on average only 2M candidates are generated from all possible 200M feature pairs (Table S1); (3) TRAVERSAL: the modules HORIZSAMPOPT and VERTSAMPOPT achieve an additional 6× speedup compared with SAMPOPT by terminating after only considering $\chi = 10^7$ feature pairs, approximately $\frac{1}{20}$ of all possible feature pairs. This speedup of HORIZSAMPOPT and VERTSAMPOPT is approaching the limit set by the feature ordering overhead (around 6 s) and the time for the TRANSFORMATION module (around 8 s) (Table S1). The improvement over SAMPOPT is not stronger, since the candidate generation phase of SAMPOPT is able to remove a vast amount of the feature pairs from full evaluation that would also be ignored by HORIZSAMPOPT and VERTSAMPOPT (Table S1).

Next, consider the log-scale interquartile range (IQR) of the running times for the different selected algorithms (Figure 5). We observe that EARLYORDERING has the largest interquartile range, indicating that the EARLYSTOP module, which tries to reduce the number of objects evaluated for each feature pair, is very dependent on the object set and feature values. As we discussed in Early Termination, EARLYSTOP has no guarantee of improving the running time. In fact, the algorithm can occasionally be worse than the BASELINE as shown in Figure 5B because EARLYSTOP incurs additional overhead for checking the criteria for pruning and early termination when scanning the object list for each feature pair. Similar results for LINCS are shown in Figure 5B (see Supplemental Experimental Procedures).

Separability Quality

As discussed previously, we found the accuracy of the baseline method that computes the Rocchio-based estimate of top-k features to be high. Here, we study the impact of our modules on accuracy. The EARLYSTOP module is deterministic and produces the same top-k feature pairs as the baseline method only with optimized computation. The SAMPLING module, on the other hand, is stochastic and can only provide an approximation of the top-k feature pair ranking. Finally, the TRAVERSAL module is heuristic and may output top-k feature pairs that are very different from the ranking produced by the BASELINE algorithm, and since BASELINE returns the true Rocchio-based separability score of each feature pair, we measured the quality of each selected algorithm by counting the number of common feature pairs returned in the top 100 between the BASELINE and the given algorithm. Figure 6 shows this separability quality comparison.

Let us first focus on MSIGDB. EARLYORDERING, as expected, has exactly the same separability quality as the BASELINE. We also observe that the SAMPONLY and SAMPOPT rankings are nearly identical to the top 100 feature pairs of the BASELINE, owing to the probabilistic guarantee described in Supplemental Experimental Procedures. The HORIZSAMPOPT and VERTSAMPOPT algorithms output a median of 92 and 48 feature pairs in common with BASELINE, respectively, because of the heuristic TRAVERSAL module. In the MSIGDB results, HORIZSAMPOPT performs much better than VERTSAMPOPT, with the median much higher and the IQR much narrower, as shown in Figure 6A. This suggests, as we hypothesized, that interesting separating feature pairs exist outside of only the combinations of the top single features as in VERTSAMPOPT. We repeated this quality analysis for LINCS and found that the SAMPLING-based algorithms returned identical top-100 feature pairs for all 40 datasets. The quality of the TRAVERSAL-based algorithms was again lower, although the performance separation of the HORIZSAMPOPT and the VERTSAMPOPT algorithms was not as large as for MSIGDB.

Takeaways

If the accuracy is paramount, SAMPOPT is recommended; if the running time is paramount to the user, HORIZSAMPOPT is recommended.

Table 2. Selected Algorithms Using Different Optimization Modules

	EARLYSTOP	SAMPLING	Candidate Ordering	TRAVERSAL	Approximation	Complexity
BASELINE	no	No	no	any	no	$O(m^2n)$
EARLYORDERING	yes	No	no	any	no	$O(m^2n)$
SAMPONLY	no	Yes	no	any	yes (guarantee)	$O(mn + m^2\lvert\mathcal{S}\rvert + \lvert\mathcal{C}\rvert n)$
SAMPOPT	no	Yes	yes	any	yes (guarantee)	$O(mn + m^2\lvert\mathcal{S}\rvert + \lvert\mathcal{C}\rvert n)$
HORIZSAMPOPT	no	Yes	yes	horizontal	yes (heuristic)	$O(mn + \chi\lvert\mathcal{S}\rvert + \lvert\mathcal{C}\rvert n)$
VERTSAMPOPT	no	Yes	yes	vertical	yes (heuristic)	$O(mn + \chi\lvert\mathcal{S}\rvert + \lvert\mathcal{C}\rvert n)$

All algorithms, including the BASELINE, are using TRANSFORMATION. In addition, EARLYSTOP and TRAVERSAL are coupled with object ordering and feature ordering by default, respectively. Each algorithm (row) shows which optimization modules are employed, whether the algorithm is returning the exact answer or an approximation answer, and the running time complexity for that combination. The terms "guarantee" and "heuristic" indicate that the returned answer is with and without stochastic guarantee, respectively. In addition, m and n are the number of features and objects, $\mathcal{S}$ is the sampled set size, χ is the limit on the number of feature pairs considered, and $\mathcal{C}$ is the number of generated feature pair candidates.

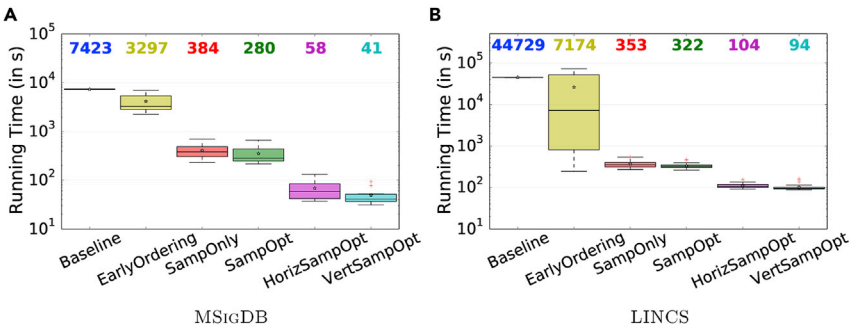

Figure 5. Running Time Comparison

A boxplot for each algorithm (A, MSIGDB; B, LINCS) is shown with the median value appearing in matching color above. For each boxplot, whiskers are set to be 1.5× the interquartile range, the outliers are shows as red dots, and the average is marked by a black star. The number on the top shows the median running time for each algorithm.

Feature Pair versus Single Feature

In this section, we quantify the statistical significance of the top-ranking results of the selected algorithms. We show that we often find separating feature pairs that are more significant than the best single separating feature. To assess the significance of a separating feature or feature pair, we first calculate the p value using the one-sided Fisher's exact test on a 2×2 contingency table. This contingency table is constructed with the rows being the true positive and negative labels, the columns being the predicted positive and negative labels, and the values being the number of objects that belong to each table cell. Using the Fisher's exact test p value, we assert that feature pairs can provide a better separability compared with single features, i.e., (1) feature pairs have stronger p values compared with the corresponding individual features even after appropriate multiple hypothesis correction and (2) there exist high-ranked pairs of features that are poorly ranked on their own as single features.

Single Feature

Finding top-k single features is a special case of finding feature pairs by setting $i = j$. For each single feature obtained, we compute the p value with Fisher's exact test, denoted as *pval*. Next, we define the Bonferroni-corrected p value as $corrected\_pval = pval \times m \times n$, since there are $m \times n$ possible hypotheses, one for each possible single feature and separating line. We say a selected feature is significant if the corrected p value is smaller than the threshold 10^{-5}, i.e., $-\log_{10}(corrected\_pval) \geq 5$. In Figure 7, we plot the distribution of the corrected p value of the top 100 features reported for each dataset in MSIGDB and LINCS. We observe that 10 out of 10 datasets in MSIGDB and 32 out of 40 datasets in LINCS have at least one significant single feature, and will focus on these datasets for further analysis. We observe very small p values, $\leq 10^{-50}$, in the left part of Figures 7A and 7B, indicating that single features are sufficient to separate the object classes for several datasets well.

Feature Pair

We next build the contingency tables and calculate the p value for the top-k feature pairs. To correct for m^2 possible feature pairs and the n^2 possible ways to choose the separating lines for each feature pair, we apply a Bonferroni p value correction to produce the $corrected\_pval = pval \times m^2 \times n^2$. We plot the distribution of the corrected p values for the top-k feature pairs in Figure 7. Once again, the threshold for defining a significant feature pair is set to 10^{-5}. We find that 10 out of 10 datasets in MSIGDB and 27 out of selected 32 datasets in LINCS have at least one significant feature pair by this metric. Visual comparison of the top 100 single features with the top 100 feature pairs (Figure 7) per dataset reveals several datasets where the corrected p values of the feature pairs are more significant than those of the best single features, even after accounting for the larger search space. Admittedly this is not always the case; e.g., for five LINCS datasets no feature pair was found to be significant at $corrected\_pval \leq 10^{-5}$ while at least one single feature did meet this threshold. Overall, this analysis suggests that rapid discovery of top feature pairs may identify more significant patterns in the given dataset than does a traditional single-feature analysis. In the following, we further illustrate that feature pairs can also provide better and newer insights compared with single features.

Improvement from Single Feature to Feature Pair

Having computed the corrected p value for each single feature and feature pair in the top 100 for our datasets, we now examine the improvement of each feature pair from its two corresponding single features in terms of p value. For each feature pair (f_i, f_j), we define the improvement quotient as the ratio between the corrected p value of (f_i, f_j) and the better one of the corrected p value of f_i or f_j, i.e., $improv\_quot = \frac{corrected\_pval(f_i,f_j)}{\min(corrected\_pval(f_i),corrected\_pval(f_j))}$. We examined only the *improv_quot* for the top 20 feature pairs for each of the ten runs in MSIGDB and 32 runs in LINCS. We found that on average across these datasets, 9.3 of the top 20 feature pairs in MSIGDB and 8 of the top 20 feature pairs in LINCS are more significant than their corresponding single features ($-\log_{10}(improv\_quot) > 5$). The distribution of the *improv_quot* is plotted in Figure S4. Overall, these histograms show that there is an improvement from single features to some feature pairs in terms of the separability significance. Next, we will explore the improved feature pairs more carefully, commenting on their redundancy, reliability, and relevance.

New Insights from Feature Pairs

To assess the quality of the top-ranking feature pairs, we focused on the LINCS dataset where the objects are experimental treatments on the MCF7 breast cancer cell line with the same drug and the features are expression values for different genes. For the evaluations above, we used object sets for the 40 drugs with the largest number of LINCS experiments. For the following analysis, we refine our list to those that are common drugs and have at least 60 LINCS experiments on the MCF7 cell line. These nine drugs are vorinostat, trichostatin, estradiol, tamoxifen, doxorubicin, gemcitabine, daunorubicin, idarubicin, and pravastatin. For each chosen drug, we ran the SAMPOPT algorithm of GENVISAGE to rank the top 1,000 feature (gene) pairs for separating the LINCS experiments of the drug from all other MCF7 experiments.

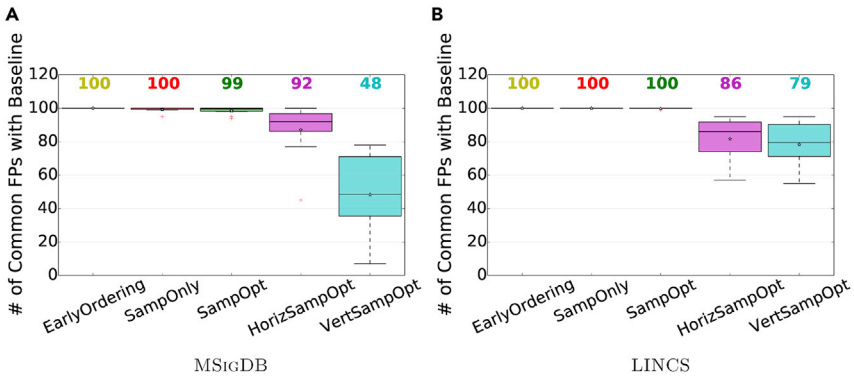

Figure 6. Separability Quality Comparison
Boxplots in the style of Figure 5 comparing the number of feature pairs returned by each method from the 100 best feature pairs of the baseline.

For all drugs except pravastatin, all of the top-1,000 ranked feature pairs were found to be significant, i.e., $-\log_{10}(corrected\_pval)>5$ (see Table 3). As described in Feature Pair versus Single Feature, we are especially interested in feature pairs whose corrected p value is better than the corrected p values of their corresponding single features ($-\log_{10}(improv\_quot)>0$). We found 1,070 "improved" feature pairs with larger separability over their single feature among the top 1,000 of these evaluation drug sets. One drug, trichostatin, had especially strong single features and showed no feature pairs that significantly improved on them. The remaining seven drugs, however, benefited from the feature pair analysis, yielding between 9 (tamoxifen) and 369 (doxorubicin) improved feature pairs (Table 3).

Many of the aforementioned 1,070 significantly improved feature pairs are partially redundant, in the sense that they comprise a common best-ranked single feature (gene). In fact, we found for all drug runs except doxorubicin that at least 20% of the improved feature pairs for that run contained a shared gene (the results are presented in Table S4). An example of this is with the object set for the drug (small molecule) estradiol. We found the gene PRSS23 as the single feature with the highest separability and many feature pairs containing PRSS23 and a second gene as having an improved corrected p value, for example (PRSS23, RAP1GAP), (PRSS23, TSC22D3), and (PRSS23, BAMBI). We looked for evidence of the relationship between the drug estradiol and these feature pair genes in the Comparative Toxicogenomics Database[32] and with our own literature survey. From this search, we found evidence for the pronounced effect of estradiol in increasing expression levels of PRSS23,[33] RAP1GAP,[34] and BAMBI,[35] and decreasing expression of TSC22D3.[36] So although the top single feature (gene PRSS23) reoccurred in multiple top feature pairs, each secondary feature gene was also meaningfully related to the administered drug in this case.

We next examined the 1,070 improved feature pairs, found over the nine LINCS datasets, to determine their consistency with existing biological knowledge bases (see Supplemental Experimental Procedures for details). The interaction networks from these sources covered 23,167 genes and had at least one known interaction between 2.17% of all possible gene pairs. Our 1,070 improved feature pairs were mapped to 996 unique gene pairs in this interaction dataset. The number is reduced because in some cases, (1) the feature identifier could not be mapped to a corresponding gene identifier, (2) two differently named features were mapped to the same gene, or (3) the same feature pair was present in more than one drug experiment. Of the 996 unique feature pairs with significant *improv_quot*, 133 gene pairs (13.4%) were found to have at least one known interaction. This 6-fold enrichment demonstrates that GENVISAGE more often finds pairs of genes that have a known relationship than is expected by chance. One example is (GLRX, NME7) that is especially good for separating vorinostat experiments from all others. Not only are both of these genes known to have increased mRNA expression in response to vorinostat,[37,38] but the two genes are annotated by STRING to both be in database pathways of nucleotide biosynthesis, co-express with each other in other model organisms, and mentioned together often in literature abstracts. In the next section, we will demonstrate that the positive objects and negative objects are visually separated under this feature pair, as shown in Figure 8.

In Table S2, for the LINCS nine drug datasets we examine several of the "improved" feature gene pairs reported by GENVISAGE analysis. Of 39 feature pairs in this table, 12 have three types of accompanying evidence: (1) a literature-based relationship between the drug and the first gene; (2) a literature-based relationship between the drug and the second gene; and (3) an interaction network relationship between the pair of genes. Six have two of the three types of evidence and there are only three with no evidence at all. Particularly interesting are the top improved feature pairs in which neither of the single gene features ranked well alone. An example is the gene pair CDKN1A and CEBPB for separating doxorubicin experiments from others. Either gene feature alone is not within the top 600 genes for separating doxorubicin experiments from others. However, the combination of the pair is significant at a corrected p value of 2×10^{-25} and is the second most improved feature pair for doxorubicin. This feature pair also has all three types of accompanying evidence; doxorubicin is known to increase expression of CDKN1A and CEBPB,[39] and the pair of genes are annotated in STRING to have evidence for co-expression and text-mining relationships. This feature pair can be used to form an interesting hypothesis for further analysis or experiment. The potential for finding more significant and previously unidentified features is why GENVISAGE is designed to recover top-ranking feature pairs instead of just single features.

Output Visualizations

As discussed in the Introduction, the output of GENVISAGE is not simply a ranking of the top feature pairs with their scores but also a visualization that helps users to interpret the separability. In Figure 8, we depict sample output visualizations from

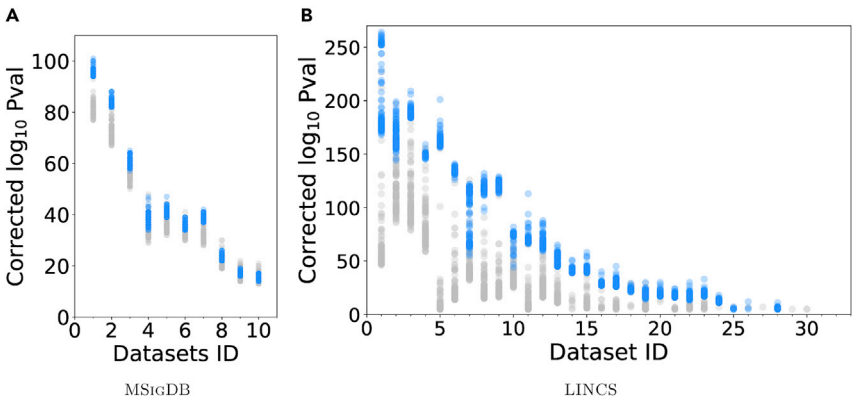

Figure 7. Single-Feature Bonferroni-Corrected p Value Distribution versus Feature Pairs' Corrected p Value Distribution

For each test the x axis shows the significance ($-\log_{10}(corrected\_pval)$) of the top 100 best single features (gray dots) and feature pairs (blue dots) for the (A) MSIGDB and (B) LINCS datasets. We order the datasets by their best corrected single-feature p value and discard the datasets where no single feature has a corrected p value better than 10^{-5}.

the MSIGDB and LINCS runs. For MSIGDB, we select the feature pair with the highest improved p value, i.e., *improv_quot*, using the SAMPOPT algorithm. For our LINCS representative, we visualize the gene feature pair (GLRX, NME7) for the drug vorinostat as described in the previous section. For the MSIGDB example (Figure 8A), we observe that the feature values for negative objects are clustered around zero, while the genes differentially expressed in papillary thyroid carcinomas from this MSIGDB study have larger values overall, indicating stronger connections to the two Gene Ontology terms features, cell adhesion and response to reactive oxygen species. This is consistent with studies that have highlighted the overexpression of important cell adhesion genes in thyroid cancer.[40] For the LINCS example (Figure 8B), positive objects mostly have elevated expression for the two reported genes (GLRX and NME7) compared with the negative objects. The direction of this differential gene expression for both genes is consistent with literature for vorinostat experiments.[37,38] The above two examples illustrate how visualization of significant feature pairs can be a useful way to explain the separability of object sets and understand the data.

Table 3. Feature Pair Statistics by Drug Treatment

Drug	Num Exprs	Avg. Signif.	Top 1,000 Signif.	Top 1,000 Improved
Vorinostat	904	235.5	1,000	287
Trichostatin	689	277.1	1,000	0
Estradiol	325	166.8	1,000	203
Tamoxifen	122	105.8	1,000	9
Doxorubicin	104	28.0	1,000	369
Gemcitabine	97	52.5	1,000	116
Daunorubicin	91	40.9	1,000	28
Idarubicin	78	30.1	1,000	58
Pravastatin	61	−7.5	0	0
Grand total		43.1	9,068	1,070

For each chosen drug from LINCS, the number of experiments in MCF7 cell line that were performed with that drug (NumExprs), and statistics for the top 1,000 feature pairs for that drug including the average $-\log_{10}(corrected\_pval)$ (Avg. Signif.), number of feature pairs with $-\log_{10}(corrected\_pval)>5$ (Top 1,000 Signif.), and number with $-\log_{10}(improv\_quot)>0$ (Top 1,000 Improved).

DISCUSSION

The GENVISAGE algorithm with its optimization modules enables researchers to visualize and explore the interplay between important pairs of genomic features rapidly, rather than relying on slow machine-learning feature extraction methods or only examining the simple list of top single features. The optimization modules led to a two orders of magnitude speedup in the task of returning the top feature pairs for separating the biological classes in our two benchmark datasets, MSIGDB and LINCS. The quality of these top feature pairs was confirmed by their agreement with literature and interaction databases, and the features are easily understood with intuitive heatmap visualizations. GENVISAGE relies on the Rocchio-based separability measure, which well approximates the best possible linear separator quickly and enables optimizations such as TRANSFORMATION that can precompute important quantities from each individual feature.

One potential improvement over the current Rocchio-based measure is to take into account the different variance[41] in the two labeled object sets. In particular, if the positive and negative object sets have very different variance along the direction that connects the two centroids, we should not use the perpendicular bisector as our representative separating line. Formally speaking, let var_{ij}^{+} and var_{ij}^{-} be the variance of the positive object set and the negative object set, respectively, along the direction that connects the two centroids μ_{ij}^{+} and μ_{ij}^{-}. Instead of taking the perpendicular bisector, we can use a perpendicular line that separates the two centroids with a ratio of $\frac{\sqrt{var_{ij}^{+}}}{\sqrt{var_{ij}^{-}}}$. This approach can potentially improve the accuracy of Rocchio-based measure, but as a trade-off it will also incur additional computation cost. Specifically, we need to compute the variance for each feature pair, which in total has a complexity of $O(m^2n)$. Sampling techniques can potentially provide an efficient estimate of the variance and once again provide a speedup for ranking results by this more sophisticated measure.

Additionally, because of the dependency on linearity, feature pairs with distinct object class distributions that form complex, non-convex, non-isotropic patterns are potentially very interesting, but will not be well ranked by GENVISAGE. Finally, in GENVISAGE, the optional SAMPLING module and TRAVERSAL modules make stochastic or greedy decisions in order to estimate the quality of and prune the potential candidate feature pairs for evaluation. While this greatly benefits the amount of time required to find the top-ranking pairs, it has the potential to do so at the cost of ranking accuracy. Overall, we observed

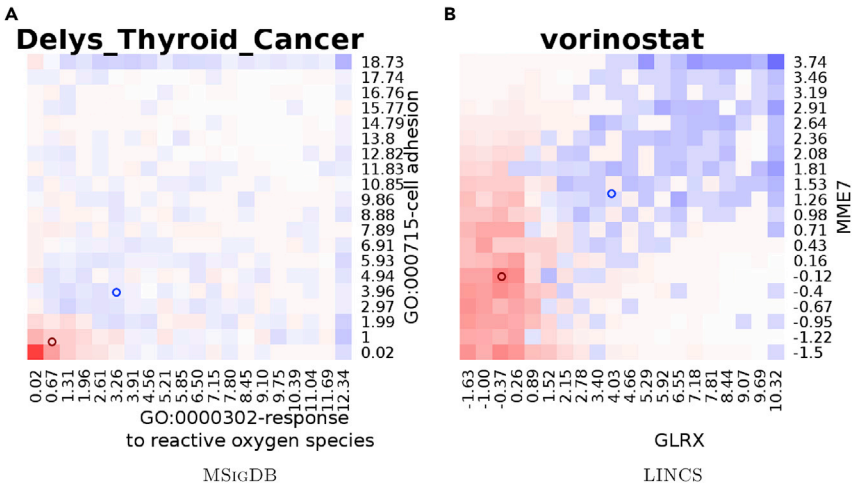

Figure 8. Visualization Output of GENVISAGE

Heatmap visualization with the pair of top features providing the x and y axes and the name of the run providing the plot title. The relative density of objects determines the color of the heatmap cells, with blue indicating a greater proportion of positive objects and red indicating a greater proportion of negative objects. The class centroids are represented by blue circles (positive class) and red circles (negative class). The two examples shown are representatives from (A) MSIGDB and (B) LINCS datasets.

that for our settings, the sacrifice in accuracy was slight for the SAMPOPT feature pair rankings and more substantial when using the HORIZSAMPOPT and VERTSAMPOPT rankings with the greedy candidate traversal. However, users of GENVISAGE are able to optimize the trade-off with performance and accuracy by modifying the sample size, $|\mathcal{S}|$, used by the SAMPLING module or the number of candidate feature pairs examined, χ, by TRAVERSAL module depending on the needs of their research and dataset.

EXPERIMENTAL PROCEDURES

The details of the experimental procedures are enumerated in the Results section and in Supplemental Information.

Resource Availability

GENVISAGE resources are available at a public, free-to-use webserver for discriminating two gene sets: http://genvisage.knoweng.org:443/.

Lead Contact

Aditya Parameswaran, adityagp@berkeley.edu, is the lead contact for this work.

Materials Availability

This work did not generate any non-code materials.

Data and Code Availability

The code associated with this work is available at https://github.com/KnowEnG/Genvisage.

SUPPLEMENTAL INFORMATION

Supplemental Information can be found online at https://doi.org/10.1016/j.patter.2020.100093.

ACKNOWLEDGMENTS

The authors gratefully acknowledge funding support for this work from the National Institutes of Health Big Data to Knowledge (BD2K) initiative [1U54GM114838, 3U54EB020406-02S1], the National Science Foundation [IIS-1733878], 3M, and Microsoft.

AUTHOR CONTRIBUTIONS

Methodology, S.H. and A.P.; Software, S.H.; Investigation, S.H. and C.B.; Data Curation, C.B.; Writing – Original Draft, S.H, C.B., S.S., and A.P.; Writing – Review & Editing, S.H., C.B., S.S., and A.P.; Supervision, S.S. and A.P.; Funding Acquisition, S.S. and A.P.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: May 10, 2020
Revised: July 13, 2020
Accepted: August 5, 2020
Published: September 11, 2020

REFERENCES

1. Tang, Z., Li, C., Kang, B., Gao, G., Li, C., and Zhang, Z. (2017). GEPIA: a web server for cancer and normal gene expression profiling and interactive analyses. Nucleic Acids Res. *45*, W98–W102.
2. Lamb, J., Crawford, E.D., Peck, D., Modell, J.W., Blat, I.C., Wrobel, M.J., Lerner, J., Brunet, J.P., Subramanian, A., Ross, K.N., et al. (2006). The Connectivity Map: using gene-expression signatures to connect small molecules, genes, and disease. Science *313*, 1929–1935.
3. Liberzon, A., Birger, C., Thorvaldsdóttir, H., Ghandi, M., Mesirov, J.P., and Tamayo, P. (2015). The Molecular Signatures Database (MSigDB) hallmark gene set collection. Cell Syst. *1*, 417–425.
4. Hanczar, B., Zucker, J.D., Henegar, C., and Saitta, L. (2007). Feature construction from synergic pairs to improve microarray-based classification. Bioinformatics *23*, 2866–2872.
5. Geman, D., d'Avignon, C., Naiman, D.Q., and Winslow, R.L. (2004). Classifying gene expression profiles from pairwise mRNA comparisons. Stat. Appl. Genet. Mol. Biol. *3*. Article19. https://doi.org/10.2202/1544-6115.1071.
6. Shi, P., Ray, S., Zhu, Q., and Kon, M.A. (2011). Top scoring pairs for feature selection in machine learning and applications to cancer outcome prediction. BMC Bioinformatics *12*, 375.
7. Shen, R., Luo, L., and Jiang, H. (2017). Identification of gene pairs through penalized regression subject to constraints. BMC Bioinformatics *18*, 466.
8. Sinha, S., Adler, A.S., Field, Y., Chang, H.Y., and Segal, E. (2008). Systematic functional characterization of cis-regulatory motifs in human core promoters. Genome Res. *18*, 477–488.
9. Watkinson, J., Wang, X., Zheng, T., and Anastassiou, D. (2008). Identification of gene interactions associated with disease from gene expression data using synergy networks. BMC Syst. Biol. *2*, 10.
10. Dudoit, S., Yang, J.Y.H., Callow, M.J., and Speed, T.P. (2002). Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. Stat. Sin. *12*, 111–139.

11. Lai, C., Reinders, M.J., van't Veer, L.J., and Wessels, L.F. (2006). A comparison of univariate and multivariate gene selection techniques for classification of cancer datasets. BMC Bioinformatics *7*, 235.

12. Peng, J., Zhu, J., Bergamaschi, A., Han, W., Noh, D.Y., Pollack, J.R., and Wang, P. (2010). Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer. Ann. Appl. Stat. *4*, 53–77.

13. Breiman, L. (2001). Random forests. Machine Learn. *45*, 5–32.

14. Basu, S., Kumbier, K., Brown, J.B., and Yu, B. (2018). Iterative random forests to discover predictive and stable high-order interactions. Proc. Natl. Acad. Sci. U S A *115*, 1943–1948.

15. Shah, R.D., and Meinshausen, N. (2014). Random intersection trees. J. Machine Learn. Res. *15*, 629–654.

16. Schwarz, D.F., König, I.R., and Ziegler, A. (2010). On safari to Random Jungle: a fast implementation of Random Forests for high-dimensional data. Bioinformatics *26*, 1752–1758.

17. Stephens, Z.D., Lee, S.Y., Faghri, F., Campbell, R.H., Zhai, C., Efron, M.J., Iyer, R., Schatz, M.C., Sinha, S., and Robinson, G.E. (2015). Big data: astronomical or genomical? PLoS Biol. *13*, e1002195.

18. Unger, G., and Chor, B. (2008). Linear separability of gene expression data sets. IEEE/ACM Trans. Comput. Biol. Bioinform. *7*, 375–381.

19. Sinha, S. (2003). Discriminative motifs. J. Comput. Biol. *10*, 599–615.

20. Suderman, M., and Hallett, M. (2007). Tools for visually exploring biological networks. Bioinformatics *23*, 2651–2659.

21. Barsky, A., Munzner, T., Gardy, J., and Kincaid, R. (2008). Cerebral: visualizing multiple experimental conditions on a graph with biological context. IEEE Trans. Vis. Comput. Graph. *14*, 1253–1260.

22. Craig, P., and Kennedy, J. (2003). Coordinated graph and scatter-plot views for the visual exploration of microarray time-series data. In IEEE Symposium on Information Visualization 2003 (IEEE Cat. No. 03TH8714) (IEEE), pp. 173–180.

23. Amir, el-A.D., Davis, K.L., Tadmor, M.D., Simonds, E.F., Levine, J.H., Bendall, S.C., Shenfeld, D.K., Krishnaswamy, S., Nolan, G.P., and Pe'er, D. (2013). viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. Nat. Biotechnol. *31*, 545–552.

24. Kobak, D., and Berens, P. (2019). The art of using t-SNE for single-cell transcriptomics. Nat. Commun. *10*, 5416.

25. Medin, D.L., and Schwanenflugel, P.J. (1981). Linear separability in classification learning. J. Exp. Psychol. Hum. Learn. Mem. *7*, 355–368.

26. Vapnik, V.N. (1998). Statistical Learning Theory (Wiley-Interscience).

27. Rocchio, J. (1971). Relevance feedback in information retrieval. In The SMART Retrieval System—Experiments in Automatic Document Processing, G. Salton, ed. (Prentice-Hall), pp. 313–323.

28. Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. J. Am. Stat. Assoc. *58*, 13–30.

29. Blatti, C., and Sinha, S. (2016). Characterizing gene sets using discriminative random walks with restart on heterogeneous biological networks. Bioinformatics *32*, 2167–2175.

30. Subramanian, A., Tamayo, P., Mootha, V.K., Mukherjee, S., Ebert, B.L., Gillette, M.A., Paulovich, A., Pomeroy, S.L., Golub, T.R., Lander, E.S., and Mesirov, J.P. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc. Natl. Acad. Sci. U S A *102*, 15545–15550.

31. Subramanian, A., Narayan, R., Corsello, S.M., Peck, D.D., Natoli, T.E., Lu, X., Gould, J., Davis, J.F., Tubelli, A.A., Asiedu, J.K., et al. (2017). A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. Cell *171*, 1437–1452.

32. Grondin, C.J., Davis, A.P., Wiegers, T.C., Wiegers, J.A., and Mattingly, C.J. (2018). Accessing an expanded exposure science module at the comparative toxicogenomics database. Environ. Health Perspect. *126*, 014501.

33. Chan, H.-S., Chang, S.J., Wang, T.Y., Ko, H.J., Lin, Y.C., Lin, K.T., Chang, K.M., and Chuang, Y.J. (2012). Serine protease PRSS23 is upregulated by estrogen receptor α and associated with proliferation of breast cancer cells. PLoS One *7*, e30397.

34. Moggs, J.G., Tinwell, H., Spurway, T., Chang, H.S., Pate, I., Lim, F.L., Moore, D.J., Soames, A., Stuckey, R., Currie, R., et al. (2004). Phenotypic anchoring of gene expression changes during estrogen-induced uterine growth. Environ. Health Perspect. *112*, 1589–1606.

35. Spink, B.C., Bennett, J.A., Pentecost, B.T., Lostritto, N., Englert, N.A., Benn, G.K., Goodenough, A.K., Turesky, R.J., and Spink, D.C. (2009). Long-term estrogen exposure promotes carcinogen bioactivation, induces persistent changes in gene expression, and enhances the tumorigenicity of MCF-7 human breast cancer cells. Toxicol. Appl. Pharmacol. *240*, 355–366.

36. Sengupta, S., Obiorah, I., Maximov, P.Y., Curpan, R., and Jordan, V.C. (2013). Molecular mechanism of action of bisphenol and bisphenol A mediated by oestrogen receptor alpha in growth and apoptosis of breast cancer cells. Br. J. Pharmacol. *169*, 167–178.

37. Qi, Y.-f., Huang, Y.X., Dong, Y., Zheng, L.H., Bao, Y.L., Sun, L.G., Wu, Y., Yu, C.L., Jiang, H.Y., and Li, Y.X. (2014). Systematic analysis of time-series gene expression data on tumor cell-selective apoptotic responses to HDAC inhibitors. Comput. Math. Methods Med. *2014*, 867289.

38. Soldi, R., Cohen, A.L., Cheng, L., Sun, Y., Moos, P.J., and Bild, A.H. (2013). A genomic approach to predict synergistic combinations for breast cancer treatment. Pharmacogenomics J. *13*, 94–104.

39. Zhao, W.-J., Wei, S.N., Zeng, X.J., Xia, Y.L., Du, J., and Li, H.H. (2015). Gene expression profiling identifies the novel role of immunoproteasome in doxorubicin-induced cardiotoxicity. Toxicology *333*, 76–88.

40. Gorka, B., Skubis-Zegadło, J., Mikula, M., Bardadin, K., Paliczka, E., and Czarnocka, B. (2007). NrCAM, a neuronal system cell-adhesion molecule, is induced in papillary thyroid carcinomas. Br. J. Cancer *97*, 531–538.

41. Balakrishnama, S., and Ganapathiraju, A. (1998). Linear Discriminant Analysis—A Brief Tutorial (Institute for Signal and Information Processing, Mississippi State).