Scalable FRaC Variants: Anomaly Detection for Precision Medicine

Cyrus Cousins
Department of Computer Science
Brown University
Email: cyrus_cousins@brown.edu

Christopher M. Pietras
Department of Computer Science
Tufts University
Email: christopher.pietras@tufts.edu

Donna K. Slonim
Department of Computer Science
Tufts University
Email: slonim@cs.tufts.edu

***Abstract*—The FRaC anomaly detection algorithm has been previously used to identify anomalous mRNA expression patterns, and has served as the core of an approach that characterizes individual anomalies by identifying dysregulated molecular functions. However, FRaC operates by training supervised models for each feature in a data set. Thus, scaling to substantially larger data sets, such as those reflecting common sequence variants, would require prohibitive amounts of computation time and memory. Additionally, although FRaC is designed to be relatively robust to irrelevant variables, it is not perfectly so; due to the low sample sizes and large number of variables in molecular data sets, substantially increasing the number of features beyond those in gene expression data sets raises the possibility of overwhelming the signal with noise. In this paper, we examine the scalability of FRaC variants using different feature reduction methods. We demonstrate that it is possible to preserve the anomaly detection accuracy of the original FRaC algorithm while requiring considerably fewer computational resources, allowing these methods to scale to handle other types of genomic data.**

I. Introduction

Recent rapid technological advances have increased the prospects for precision medicine, allowing us to tailor medical care to individual patients or patient groups based on underlying molecular patterns. This problem is well-suited to the machine learning framework of anomaly detection (1), in which predictive models are trained on a population of either all normal or mostly normal samples, and new samples are then individually compared to this population to identify abnormalities or outliers. Anomaly detection has great potential for precision medicine applications. It can be used to detect and explain rare medical abnormalities, such as obscure genetic diseases, or to characterize specific instances of molecularly heterogeneous disorders (2), for which assembling a homogeneous data set may be challenging.

However, finding anomaly detection methods that handle tasks of the size of most genomic classification problems is not trivial. Among other concerns, the chosen methods need to be relatively robust to irrelevant variables, given that the majority of features in most genomic data sets are likely to be irrelevant to the chosen phenotype. The *interpretability* of anomaly detection algorithms is also important. It is not enough to determine that a sample is anomalous; we also want to derive a molecular characterization of that specific anomaly to yield insight into the nature of an individual patient's condition.

In previous work, we developed Feature Regression and Classification (FRaC), a robust feature prediction approach for the anomaly detection problem (3), and we showed that it is more robust to irrelevant variables (4) than top competing methods such as local outlier factor (5) or one-class support vector machines (6). We then used FRaC as a component of CSAX, a method for identifying and interpreting anomalies in individual gene expression samples (7). We applied this approach to a collection of 28 public gene expression data sets, which we refer to here as the "CSAX compendium." These data sets generally suffer from two characteristics that negatively impact their amenability to learning: high dimensionality and low sample sizes. For example, while the classical machine learning data sets in the UCI repository (8) typically have hundreds to hundreds of thousands of samples yet fewer than 1,000 features, the data sets in the CSAX compendium typically have at most a few hundred samples and from a few thousand to over 50,000 features.

Most anomaly detection methods struggle in such cases. Theoretically, all of these problems could be intractable: if an anomaly is only marked by abnormal expression of a single gene, no computational method could ever distinguish that signal from noise. Fortunately, most phenotypes of interest involve large numbers of related genes. In our previous work, we demonstrated that anomaly detection difficulty is to a large degree an inherent characteristic of the data set, reflecting the number of and relationships between relevant features, regardless of the computational method used. We also demonstrated that FRaC and CSAX perform well on many data sets in the CSAX compendium, and that they are on average more effective than prior methods, which appear to be more susceptible to the effects of irrelevant variables.

Although FRaC and CSAX work relatively well, they are computationally slow, because a FRaC predictive model is constructed for each of the features as a function of all others, and CSAX includes bootstrapping over multiple FRaC runs. Parallelization can help, but the overall commitment in CPU time and memory usage is still substantial. While the problem is tractable for data sets the size of those in the CSAX compendium, scaling to much larger problems will require better methods.

With the advent of high-throughput genotyping and the decreased cost of sequencing, however, we have reason to try to solve this problem. It has been postulated that a similar anomaly detection approach using genotype data, in which one tries to find relationships between multiple common sequence variants that distinguish individual patients from the healthy normal population, might be a valuable approach to understanding the heterogeneity of complex diseases. Such an approach has potential not only to detect novel genotypic abnormalities as a diagnostic tool, but also to identify the underlying molecular causes of disease susceptibility.

But genotyping data sets measuring common single nucleotide polymorphisms (SNPs) are different from the real-valued expression data sets. Each variable is typically a ternary categorical variable (a site is either heterozygous, or homozygous for either the major or minor allele). Genotyping arrays commonly include half a million or more features, and high throughput sequencing now allows genotyping essentially all common variants. (Note that rare variants are less useful in an anomaly detection context, because a rare variant, even an irrelevant one, will always appear to be anomalous.)

In this paper, we focus on scalability for both gene expression data sets from the CSAX compendium and on two public SNP data sets. Specifically, we address the issues of computation time and sample complexity.

In FRaC, a model is trained for each feature, using every other feature as input. Previous work has seen most success using support vector machines (SVMs) as the underlying model, likely because they are efficiently trainable and through regularization can mitigate the impact of overfitting. However, with the small sample sizes in many biomedical experiments, the time cost of training hundreds of thousands of SVMs is substantial, and the memory requirement is also prohibitive. Furthermore, even with regularization, it is far too easy for even a linear SVM to overfit on these data sets.

Specific to discrete data is the issue of representation. Many modeling techniques, such as SVMs, assume continuous data, and exactly how to represent discrete data in order to use the modeling techniques is a matter of some debate. We avoid this issue by modeling discrete features using decision trees. We further convert the ternary SNP features to binary vectors as described below.

An additional issue with the FRaC algorithm is that while some patterns in data may be obvious, others may be subtle. For instance, it may be that gene A is promoted by gene B and less strongly by gene C. It may be that the action of C is masked by that of B, so for instance a decision tree may fail to identify this relationship. As a result, if this relationship is violated in abnormal specimens, the breakdown may go undetected. FRaC, running on a data set with so many irrelevant variables, may miss the impact of the weaker predictor entirely.

In this paper, we discuss techniques that filter and project a data set in various ways to produce simpler learning problems that are more efficiently computable and less susceptible to overfitting. We use randomization to solve reduced problems, which partially addresses the issue of subtle patterns being masked by stronger ones.

A. Background

1) The FRaC Algorithm: FRaC works by computing an anomaly criterion known as *normalized surprisal* (NS). The NS score is an information-theoretic measure of the amount of information carried by by each feature of a data point, conditioned on the other features.

The FRaC algorithm is defined to work on data that is real, categorical, or mixed. For a data point x of f features, the normalized surprisal $NS(\vec{x}) \doteq$

$$\sum_{i=1}^{f}\sum_{j=1}^{p}\begin{cases}\vec{x}_i \text{ defined} & : \quad \text{-}\log\left(\mathbb{P}(\vec{x}_i|p_{ij}(\vec{x}_{1,\dots,i-1,i+1,\dots,f}))\right) - H(f_i)\\ \text{otherwise} & : \quad 0\end{cases}$$

Here $\mathbb{P}(\vec{x}_i|p_{ij}(\vec{x}_{1,\dots,i-1,i+1,\dots,f}))$ is the probability of having true feature value $\vec{x}_i$ given the prediction produced by predictor p_{ij} given $\vec{x}_1,\dots,\vec{x}_{i-1},\vec{x}_{i+1},\vec{x}_f$, and $H(f_i)$ is the *entropy* of feature i (as calculated using the training set).

Predictors can be any supervised learning algorithm, and probabilities are estimated with *error models*, which in the discrete case are confusion matrices, and in the continuous case are density function estimators for the probability density function given by $\vec{x}_i - p_{ij}(\vec{x}_1,\dots,\vec{x}_{i-1},\vec{x}_{i+1},\vec{x}_f)$.

In order to train error models, k-fold cross validation is used, and predictions on the holdout fold, paired with the true value, are used to construct error models. Then, the entire data set is used to train predictors.

In this paper, all continuous features are learned with linear support vector machines. This choice was made because the SVM is a regularized model, and the linear SVM has a particular constrained hypothesis class. As such, although this model is only able to learn linear functions, it is not highly susceptible to overfitting. This is extremely important in learning high dimensional data with small sample sizes. Error models simply fit a Gaussian to the error distribution, as again there is insufficient data to accurately learn a more detailed model. Finally, categorical features are learned using decision trees, and their error models are simply confusion matrices.

For the data sets we consider here, where anomaly detection problems are built from standard classification problems, we in fact do know the right answers. We can therefore train our models on a training set consisting solely of control samples, and compute surprisal scores on samples in a test set consisting of both control and anomalous samples. We can then evaluate the performance of anomaly detection methods by computing the AUC (area under the Receiver Operating Curve (9)), by ranking the anomaly scores of anomalous and control samples in the test set, as in the FRaC and CSAX papers.

Because Normalized Surprisal is a giant sum, FRaC is highly parallelizable. On the other hand, due to cross-validation, each component of the sum requires the training of multiple models, so computing the entire sum is computationally intensive. In this paper, we focus primarily on techniques that modify exactly what is computed in ways that don't negatively impact the accuracy of the technique, while improving time and memory efficiency.

2) The Johnson-Lindenstrauss Lemma: The Johnson-Lindenstrauss (JL) transform is a technique by which a point set can be projected into a low-dimensional space while preserving key properties of the original space (10). Using the ϵ-δ formulation of the transformation, described below, the JL transform is *independent* of the training set, and thus doesn't risk preferentially destroying the very signal FRaC detects, as might a data-dependent transform such as PCA.

In (11), the authors give an algorithm for a storage- and computation-efficient dimensionality-reducing Johnson-Lindenstrauss transformation. In (12), the authors show that, like distances, dot products are approximately preserved by Johnson Lindenstrauss transformations.

The Johnson-Lindenstrauss lemma proves that these distance guarantees hold. Specifically, it states that, given n points, there exists a projection into k-dimensional space such that the square Euclidean distance between *any* two points is perturbed by a factor no less that $1-\epsilon$ and no greater than $1+\epsilon$, so long as the following holds:

$$k \geq \frac{4\ln(n)}{\epsilon^2/2 - \epsilon^3/3}$$

This formulation is actually much stronger than we need. We really don't need *every* pair of points to have their distances preserved: it suffices to have *most* distances preserved. But given this strong distance preservation property, it is reasonable to assume that if learning is possible in the unprojected space, it will be almost as effective in the projected space.

The distributional form of the lemma gives the guarantee that the distance between any two points is similarly constrained with probability $1-\delta$, so long as the following holds:

$$k \geq \frac{\ln\left(\frac{2}{\delta}\right)}{\epsilon^2/2 - \epsilon^3/3}$$

The distribution from which the JL transform is drawn may then be a $k \times d$ matrix where all entries are Gaussian distributed or Uniform$(-1, 1)$ distributed.

We will therefore investigate whether using the JL transform to reduce the dimensionality of the anomaly detection problem allows us to preserve the accuracy of an anomaly detection algorithm while reducing the computational resources needed.

Note that, perhaps counterintuitively, neither formulation of the JL lemma depends on the input dimension. Consider that you can rotate n points in *any* dimensional space into $n-1$ dimensional space. With this in mind, it should not be surprising that the JL lemma is dependent on the original number of points rather than the original dimension. The probabilistic version of the formula doesn't even depend on n, because it is just a statement about the *fraction* of point pairs that have their distance preserved.

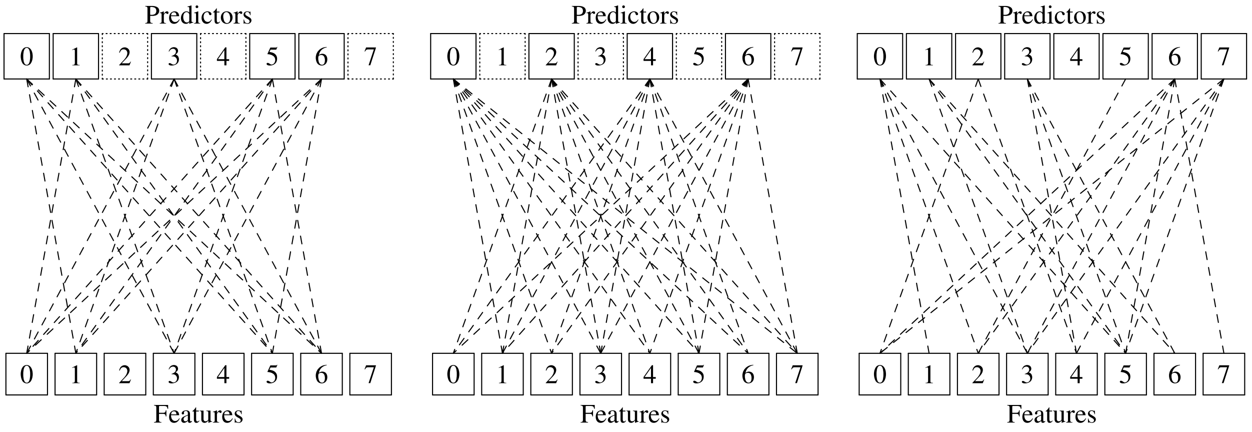
II. Scalable FRaC Variants

Here we describe several techniques, including one using the above lemma, for reducing the magnitude of the FRaC learning problem. A graphical overview of several filtering techniques is given in Figure 1.

A. Filtering

Filtering techniques can be divided into *full filtering* and *partial filtering*. The former is a computationally efficient but heavily lossy technique; the latter, a slower but less lossy technique. Filter techniques identify some property of each feature, rank the features by this property, and remove some features from consideration. In full filtering at a percentage p, p of these features are kept while the rest are removed entirely, and the technique simply runs ordinary FRaC on the remaining features. In partial filtering, these $1-p$ features aren't removed, but we do not construct predictive models for the filtered features. However, these features are used to construct models to predict non-filtered features.

In this manuscript, we evaluate simple random filtering, in which we remove features from the data set at random. For expression data sets, we know that there is likely to be enough correlation between expression of



Filtering: features $\{2, 4, 7\}$. Partial Filtering: features $\{1, 3, 5, 7\}$. Diverse FRaC, $p = \frac{1}{3}$.

Fig. 1. Graphical depiction of FRaC variants. In this example with eight features, predictors for a subset of the features are trained on the whole feature space except for the target feature (partial filtering), or on the chosen subset of the feature space except for the target feature (full filtering). Diverse FRaC chooses both target features and the training features randomly. Lines in the figure indicate that a feature is considered by a predictor.

different genes that random filtering is likely to be effective. This is not as obvious for the genotype data, but these data sets may contain enough signal redundancy to allow random models to preserve anomaly detection accuracy. We considered random partial filtering as well, but we do not present results based on this approach here because our initial experiments found this approach to be inferior to full filtering.

There are many possible ways, other than random selection, to choose features for filtering. Entropy filtering, where one ranks features by information content and keeps only the highest entropy features, is one such method. For nominal features with values $v_1, \ldots, v_k$ we estimate the likelihood of each v_i, $pr(v_i)$, from its frequency f_i in the training set, and define entropy as $\sum_{i=1}^{k} -pr(v_k)\log(pr(v_k))$. For continuous features distributed with density $f(x)$, the *differential entropy* is defined as $-\int_{-\infty}^{\infty} f(x)\log(f(x))$ dx. We estimate this value by fitting a Gaussian kernel density estimator (13) to the feature values over the training set, and computing the differential entropy of $\hat{f}(x)$. Removing low entropy features may be useful because these features are less interesting, particularly in the discrete case, where the features may be highly predictable but have low surprisal if they are not relevant to the anomaly of interest. Such features mostly contribute noise to the FRaC algorithm, so removing them may improve performance.

B. Diverse FRaC

Similar to the filtering techniques, the Diverse FRaC technique is intended to simplify each learning problem by learning each feature on a subset of the remaining features. Specifically, for some probability p, at feature i, each feature $j \neq i$ is selected with probability p. Then, a predictor for i is trained using only the features that were selected.

This approach may combine some of the potential strengths of partial filtering and full filtering, as Figure 1 illustrates. Furthermore, in addition to addressing the computational issue with respect to memory and time costs, this technique also addresses the sample complexity issue, as learning in the reduced spaces is less prone to overfitting. Notably, it also allows subtle patterns to be detected over stronger, particularly when features necessary to learn stronger patterns are absent. This addresses the issue cited in the introduction where a pattern is not learned due to the presence of a stronger patter. To further detect these sorts of patterns, we can train multiple predictors for each feature, each utilizing a different feature subset, though this increases the computational cost of the technique.

C. Ensembles

As mentioned earlier, the normalized surprisal score for a sample is simply the summation of the surprisal scores for all its terms. This makes implementing ensembles of FRaCs very easy - one simply sums all the normalized surprisal scores over all the members of the ensemble. If multiple members of the ensemble have a score for one feature, one can simply combine them by taking the median score for that feature. Ensembles of multiple random full filtered or diverse FRaC models can greatly increase the stability of results.

Feature Schema:	$\mathbb{R}$	$\mathbb{R}$	$\mathbb{R}$	$\mathbb{R}$	{0,1,2}	{0,1,2,3}
Data:	3.4	0	-2	0.6	1	2
1-Hot Transform:	—	—	—	—	$\langle 0,1,0\rangle$	$\langle 0,0,1,0\rangle$

Vector Concatenation: $\langle 3.4, 0, -2, 0.6, 0, 1, 0, 0, 0, 1, 0\rangle$
JL-transform: Apply 11×4 random linear transform.
Possible Result: $\langle 2.5, -3, 1.05, -2.73\rangle$

Fig. 2. Illustration of the 1-hot transform, vector concatenation, and JL transformation preprocessing steps.

D. Preprojection

In this technique, we take a data set that may include categorical variables and convert it to an entirely real data set. This is accomplished by converting categorical k-ary features to 1-hot[1] vectors, and concatenating all of these vectors with a vector representing any real features. We then apply the JL transform to the entire data set, reducing it to a low-dimensional space, and then perform ordinary FRaC in the projected space. The process is illustrated in Figure 2.

This projection addresses all three issues discussed in the introduction: clearly computation time will be substantially improved, as we train fewer models in simpler spaces. The approach also aids in sample complexity, as low-dimensional models are less prone to overfitting. Lastly, by performing this projection, we end up posing a large number of similar regression problems, some of which may be dominated by strong relationships and others of which rely on weaker relationships.

Let us first discuss the interpretation of this technique over real data sets with linear regressors. After performing the projection, each feature is a linear combination of other features. The task is thus to learn a *sum* of functions, given a feature space such that each feature is a also a linear combination of features from the original space. Of course, the very features we are trying to learn may also be components of the features from which we are learning, but so long as no linear combination of features exactly matches the linear combination of features we are trying to learn, we still must learn something for each combination feature.

The same interpretation applies to nonlinear regression techniques, except the linear combinations interpretation is somewhat less relevant.

The interpretation for discrete data sets is a bit more subtle. Each feature in the input space corresponds to one class of one discrete feature, and features in the projected space now represents sums of these input features.

TABLE I
Number of features, normal samples, and anomaly samples for each data set.

data set	features	normal	anomaly
`breast.basal`	3167	56	19
`biomarkers`	19739	74	53
`ethnic`	19739	95	96
`bild`	20607	48	7
`smokers2`	19739	40	39
`hematopoiesis`	13322	97	91
`autism`	7267	317	228
`schizophrenia`	171763	280	54

One attractive property of the preprojection technique is that it is very likely that there is *something* to learn in each of the projected features. With the original FRaC algorithm, it is easy for some features to be unlearnable from the remaining features. In that case, unless the target features are uniformly distributed in the training data[2], each such feature adds only noise to the NS score. When a substantial number of these features exist, as seems likely in many biological data sets, this noise may degrade the performance of FRaC. However, because projected FRaC features are linear combinations of the original features, it is unlikely that any projected feature is unlearnable, so this issue may be mitigated.

One issue with JL preprocessing is that it becomes more difficult to identify relevant features and feature relationships. We can however examine the output in aggregate, and it may be possible to identify input features that are present in many of the highly predictive projected features.

III. Experiments

A. Data Sets

We present results from six expression data sets and two SNP datsets. The expression data sets are taken from the CSAX compendium (7), and were selected for being relatively predictable (FRaC AUCs above approximately 0.6) and for having a range of feature and sample sizes. One important conclusion from the original FRaC and CSAX papers was that the difficulty of an anomaly detection task is an inherent property of the data set; performance of all anomaly detection methods was highly correlated across different data sets, which can be either ”easy” (no matter how you look at them, the anomalous samples stand out), or ”hard” (there is little or no signal in the data with which to predict which samples are anomalous). We therefore selected data sets for this study that appeared to range from “feasible” to

[1] A 1-hot vector for a categorical feature over k categories is a k dimensional vector, where category j maps to a vector $\vec{v}$ such that $\vec{v}_j = 1$ and $\vec{v}_l = 0$ for any $l \neq j$. For example, categories 1 and 3 out of $\{1, 2, 3\}$ would map to $\langle 1, 0, 0\rangle$ and $\langle 0, 0, 1\rangle$, respectively.

[2] Note that features that are uniformly distributed over the training set and are predicted uniformly given any true label make no contribution to NS, as the surprisal values are always equal to the entropy value, thus subtractively cancelling.

"easy," and augmented them with two SNP data sets. We chose publicly available anonymized SNP data to avoid human subjects concerns.

Three of the expression data sets - biomarkers (14), ethnic (15; 16; 17; 18; 19; 20; 21), and breast.basal (22) - were used for initial methods development, while the other three - bild (23), smokers2 (24), and hematopoiesis (25) - were brought in later.

The first SNP data set is a subset of an autism data set from GEO, GSE6754 (26). This is a small SNP data set, with only 7267 features, so it was possible to run FRaC on the entire data set. However, likely because of the complexity of the molecular causes of autism, FRaC has no predictive power on even the full data set (mean AUC of 0.50), so insights gained here are mainly to do with how much time improvement we can expect without making things worse.

The second SNP data set is a schizophrenia data set compiled from several different sources. The training set consists of 270 normal HapMap (27) samples from GSE5173 (28), while the test set consists of 10 normal samples from GSE21597 (29) and 54 schizophrenic samples from GSE12714 (30). This data set has 171763 features and is far too large to run in a reasonable amount of time. Entropy filtering on this data set produced almost perfect results, with an AUC around 1.0. However, given the difficulty of even diagnosing schizophrenia with such accuracy, we don't believe that the high AUC represents true relationships between variants linked to schizophrenia. Rather, we suspect that FRaC is instead detecting differences in ancestry that are confounded with disease status in this data set. Results from this data set, therefore, though not necessarily biologically informative, are useful for determining how to run such algorithms on larger collections of genotyping data. They also demonstrate that it is possible to identify signals related to variation in SNP data sets by anomaly detection methods.

For each data set except schizophrenia, we construct five replicates. Each replicate consists of a training set containing a randomly selected two-thirds of the normal samples. The test set consists of the remaining normal samples as well as all non-normal samples. The schizophrenia data set consists of only a single replicate, constructed as noted above.

B. Settings

In these experiments, we use Support Vector Machines with a linear kernel for all six expression data sets, exactly as in the original FRaC paper, and implemented with libSVM (31). In initial experiments, SVMs did not appear to work well on the discrete SNP data, taking more time and space to compute while producing less accurate anomaly scores compared to decision tree models. Thus, for the SNP data sets, we train decision trees as the predictive models, implemented with the Waffles library (32).

1) Filtering and Filtering Ensembles: Initial experiments with filtering revealed that partial filtering was consistently worse than full filtering in time, space, and AUC preservation across all data sets, so partial filtering results are not presented here. Random selection of filtered features proved to be the most effective method, though entropy-based filtering methods proved effective on some data sets. However, random filtering at small values, though fast, is not particularly stable, and results could vary wildly depending on exactly which features were kept. On some data sets, AUCs fell within an absolute range of up to .2, even within the same replicate. To remove this source of variability, we moved to ensembles of full filtered FRaC, in which FRaC was run 10 times at a filtering value of .05 (5% of features kept). Each sample's final NS score is simply the median of all NS scores generated by members of the ensemble. Entropy filtering results for $p = .05$ are also presented.

2) Diverse: Diverse FRaC was run with $p = \frac{1}{2}$. This value was selected as it halves the size of each learning problem, resulting in substantially less memory and computation time use.

We also ran experiments with ensembles of diverse FRaC. In these experiments, we ran 10 instances of diverse FRaC with $p = \frac{1}{20}$ in order to fairly compare with the full filtering ensembles.

3) JL: The JL experiments were run with a projected dimension of 1024. This number was selected as it is a round number, results in a computationally efficient data set reduction, and gives the probabilistic JL guarantee with $\delta = 0.05$ and $\epsilon = 0.057$ (in other words, 19 of every 20 pairs of points have their square distance distorted by a factor in $[0.943, 1.057]$).

For the schizophrenia data set, we ran additional JL experiments with projected dimensions of 2048 and 4096.

IV. Results and Discussion

In Table II, we report the results of running the original FRaC algorithm on each of these data sets. The table shows the mean and standard deviation of the AUCs across five replicates of each method. We note that the last row, in italics, represents only an estimate for the full Schizophrenia data set. Based on the time required for the small autism SNP data set, we extrapolated to determine the expected running time and memory usage.

Tables III and IV show the results for the random filtering ensembles, JL transform, entropy filtering, di-

TABLE II
Full run results, showing mean AUC, AUC standard deviation, CPU time, and memory usage across all replicated for each data set. We were not able to run full FRaC on the schizophrenia data set. Therefore, time and memory performance for this data set were estimated by extrapolation from the performance on the autism data.

data set	AUC	Time (h)	Mem (GB)
breast.basal	0.73 (0.06)	1.02	4.59
biomarkers	0.88 (0.05)	58.21	152.54
ethnic	0.71 (0.03)	96.67	195.11
bild	0.84 (0.08)	36.51	106.59
smokers2	0.66 (0.04)	29.23	82.57
hematopoiesis	0.88 (0.02)	56.56	90.69
autism	0.50 (0.03)	188.40	3.39
schizophrenia	N/A	*44,000*	*148*

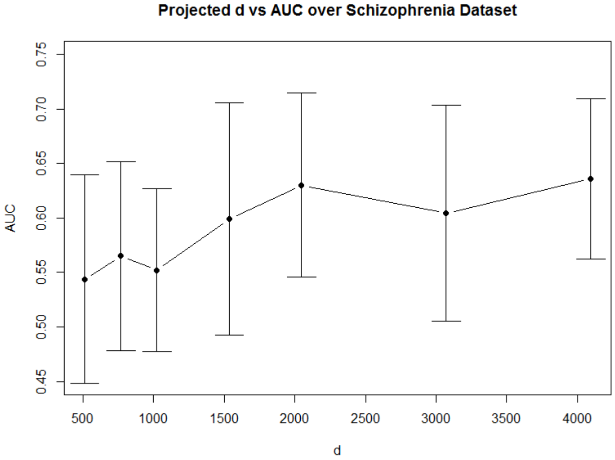


Fig. 3. JL transform AUC performance on the schizophrenia data set with various numbers of projected dimensions (d). Each data point is the average AUC of ten different projections at a given (d), with error bars representing standard deviation.

verse, and diverse ensemble FRaC algorithms on the six expression data sets and the autism data set, presented *as a fraction of the full results* from Table II. In other words, we are no longer concerned with how well we can do at anomaly detection, but how well we can do (or how quickly we can do it) compared to running the full method. Table V contains results for some methods on the schizophrenia data set. Because we did not run the original FRaC algorithm on the full data set, this table contains raw AUCs, but fractional run time and memory usages based on the estimates in Table II.

Surprisingly, we find that it is possible to perform anomaly detection with essentially the same accuracy as in the full runs, but much more quickly. We see in the tables that on average, all four methods other than entropy filtering have nearly identical performance to running FRaC on the full data sets, but significantly reduced time and memory usage. Entropy filtering is less consistent; it works extremely well on some data sets but quite poorly on others.

Across the six expression data sets, all four of the non-entropy methods exhibit comparable levels of AUC preservation. JL pre-projection seems to perform best in both run time and memory usage. Its prediction accuracy is also very good. However, the complex projected models make it more difficult to tell which of the original features are contributing to anomaly scores. This might be acceptable for the simple anomaly detection task, but in most biological applications, the goal is not only to identify anomalous samples, but to identify the molecular reasons that they are being considered anomalous. Therefore, for the best interpretability, one should use the random filter ensembles method, which still does a good job of preserving accuracy while being nearly as fast as the JL transform. Diverse filtering and diverse ensembles, we found, also learn well, but are typically too slow and memory intensive to use effectively on larger data sets.

For the SNP data sets, our conclusions are a little less clear. Random filter ensembles appear to perform as well as full FRaC on the autism data set. On the other hand, this is a famously genetically heterogeneous disorder, so it is perhaps not surprising that the anomaly detection task is essentially impossible (the full FRaC AUC hovers around 0.50) to begin with. It seems unlikely that random subsets could perform substantially *worse* than what is already effectively random guessing. We therefore used this data set primarily to estimate efficiency for the larger SNP data set.

Our overall goal is to come up with recommendations for scaling to discrete data sets. On the large schizophrenia data set, the true AUC, time, and memory usage are unknown. Given the relatively high memory usage for the diverse ensemble runs, we did not run these on the schizophrenia data. However, we did try random ensembles, entropy filtering, and the JL transform on the schizophrenia data.

The JL experiments proved to be more variable than expected on the discrete data, which might reflect the use of decision trees rather than SVMs for the discrete data. Initial results with 1024 dimensions suggested that there was some signal (an AUC > 0.5) on this data set, but also suggested that perhaps a larger number of dimensions might be required to capture relevant patterns from among so many features. As we increased the dimensionality, we did indeed see better performance (Figure 3). Further experimentation will be required to determine how to find the best tradeoff between AUC performance and time/space requirements for future large SNP data sets.

The poor performance of the JL preprocessing technique on the SNP data set, and to a lesser extent on the autism data, may also indicate that this transform is less effective for discrete data. The prediction task in

TABLE III
Results for the Random Filter Ensemble, JL Projection, and Entropy Filtering techniques. AUC, Computation time, and memory usage shown as fractions of the full run utilization. Each quantity is reported as an average across five replicates, and the average across all data sets is reported as well. Standard deviation for the AUC fraction is reported in parentheses.

	Ensemble of Random Filtering			JL			Entropy Filtering		
data set	AUC %	Time %	Mem %	AUC %	Time %	Mem %	AUC %	Time %	Mem %
breast.basal	1.01 (0.03)	0.278	0.005	0.98 (0.02)	0.258	0.078	0.97 (0.06)	0.028	0.004
biomarkers	1.09 (0.05)	0.046	0.003	1.08 (0.02)	0.003	0.003	1.01 (0.06)	0.004	0.003
ethnic	0.90 (0.03)	0.057	0.003	0.87 (0.04)	0.003	0.003	0.79 (0.06)	0.004	0.003
bild	0.97 (0.05)	0.029	0.003	0.98 (0.05)	0.003	0.003	0.93 (0.09)	0.002	0.003
smokers2	1.11 (0.07)	0.058	0.003	1.10 (0.09)	0.002	0.003	0.95 (0.06)	0.003	0.003
hematopoiesis	1.02 (0.02)	0.050	0.003	1.05 (0.02)	0.006	0.007	1.07 (0.02)	0.005	0.003
autism	1.02 (0.06)	0.030	0.028	0.94 (0.05)	0.008	0.548	0.90 (0.06)	0.005	0.043
Avg	1.02	0.078	0.007	1.00	0.040	0.092	0.95	0.007	0.009

TABLE IV
Results for the Diverse and Diverse Ensemble methods. Information is presented as in Table III.

	Diverse			Diverse Ensemble		
data set	AUC %	Time %	Mem %	AUC %	Time %	Mem %
breast.basal	0.99 (0.01)	0.455	1.123	0.99 (0.01)	0.597	0.395
biomarkers	1.09 (0.05)	0.355	0.521	1.09 (0.04)	0.402	0.510
ethnic	0.91 (0.04)	0.401	0.518	0.92 (0.04)	0.372	0.518
bild	0.96 (0.05)	0.547	0.531	0.94 (0.06)	0.533	0.534
smokers2	1.12 (0.09)	0.270	0.536	1.12 (0.08)	0.290	0.540
hematopoiesis	1.02 (0.02)	0.226	0.514	1.03 (0.01)	0.259	0.517
autism	0.97 (0.06)	0.166	0.744	1.02 (0.01)	0.099	0.786
Avg	1.01	0.346	0.641	1.02	0.365	0.543

TABLE V
Results for JL Pre-projection, random filtering ensembles, and entropy filtering on the schizophrenia data set. Information is presented as in Table III.

method	AUC	Time %	Mem %
Entropy Filtering	1.00 (N/A)	0.004	0.017
Ensemble of Random Filtering	0.86 (0.01)	0.040	0.017
JL, 1024 comps	0.55 (0.08)	0.000	0.015
JL, 2048 comps	0.63 (0.09)	0.000	0.032
JL, 4096 comps	0.64 (0.08)	0.001	0.075

the transformed, discrete case is somewhat unusual, as it is akin to predicting a weighted sum of feature values. We may find success in future work by applying preprocessing techniques tailored to preserve the structure of discrete data. Additionally, using entropy-minimizing decision trees in the transformed space may also have negatively impacted performance, as this model is *not* invariant under linear transformation. This suggests that it is important to select a preprocessing technique that is compatible with the learning models employed.

On the schizophrenia data set, entropy filtering at 5% of features had an AUC of 1.0, while keeping all the performance benefits we see of random filtering. Given the complexity of the schizophrenia diagnosis, and the hybrid nature of the data set, we suspect that the method is very accurately learning to distinguish markers of ancestry or ethnicity in the control set from ancestry in the affected patients, as they come from different populations. Supporting this hypothesis, several of the key features implicated in the entropy models have allele frequencies that differ substantially across the the HapMap populations. On the other hand, two of the top 20 predictive SNP models for the single random schizophrenia run (with an AUC of 0.86) are SNPs just adjacent to the two genes PLXNA2 and GRIN2B, both of which have been implicated in the disease (33; 34). The hypergeometric probability of finding 2 out of the top 100 known schizophrenia genes (35) by sampling 20 from a pool of 4173 (the number of SNP features in our random models) is 0.011, suggesting that the method does appear to be finding predictive SNPs and SNP interactions on this data set.

For many data sets, random filtering is highly effective. This will likely be true in any case where there is a strong and diffuse signal. So, for example, in data sets measuring gene expression in cancer we would not be surprised to see random filtering performing well. Such data sets include breast.basal, which compares different types of breast cancer, and biomarkers, which compares ER positive and ER negative tumors. Smokers2, which focuses on discriminating mucosal cells from current and never-smokers, is again likely to contain a widespread signal, so that random selection of features can be expected to perform well, as observed.

For the hematopoiesis data set, which aims to identify blood cells of lymphoid origin from among those with myeloid origins, entropy filtering is most effective. Although the 20 most predictive genes in the random filtering runs show borderline enrichment for T-cell related functional processes (in DAVID (36), unadjusted EASE scores below 0.05), there is a stronger single using entropy filtering, where even the more cautious GO enrichment tool (37) highlights leukocyte and immune related processes as significant among the most predictive genes. The most influential predictive gene models for this data set include those for *MAFB*, a transcription factor that plays a well-studied role in hematopoiesis (38), and *CCR6*, a gene that regulates B cell maturation (39).

Although entropy filtering found a strong signal in schizophrenia and performed better than other methods on the hematopoiesis data set, it was not a top performer on any other data set. We therefore hesitate to recommend it until we can predict on which data sets it will be most helpful.

On some data sets (smokers2 and biomarkers), most scalable FRaC variant techniques produce significantly better AUCs than FRaC on the full data set is able to. On these data sets, it may be the case that the trained FRaC models are overfitting on the full data set. However, we also note that our reported full results here, computed from five replicates of the original FRaC algorithm, are different from those reported in the CSAX paper, with 20 replicates and a larger fraction of the normal samples used in the training set. The smokers and biomarkers data sets have the lowest performance here as compared to the performance reported in the CSAX paper. We therefore suspect that the large improvement over full FRaC in these cases is because our initial estimate of the full performance is actually a bit low.

In conclusion, we have shown that in most cases, running a reduced FRaC variant is as good as, or possibly even better than, running the original FRaC algorithm. As high dimensional data and low sample sizes are common in precision medicine, it may be that this insight could be of great value in other analysis tasks. Furthermore, techniques such as the JL transform, and the idea of examining random subsets of data (as with features in diverse FRaC) are quite broadly applicable.

Acknowledgements

We thank Mark Crovella and Ben Hescott for helpful suggestions regarding the JL transform. We gratefully acknowledge the support of NIH R01HD076140 and R01HD058880.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 15, pp. 1–58, 2009.

[2] P. N. Robinson, "Deep phenotyping for precision medicine," *Hum. Mutat.*, vol. 33, no. 5, pp. 777–780, May 2012.

[3] K. Noto, C. Brodley, and D. Slonim, "Anomaly detection using an ensemble of feature models," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 2010, pp. 953–958.

[4] ——, "FRaC: a feature-modeling approach for semi-supervised and unsupervised anomaly detection," *Data Mining and Knowledge Discovery*, vol. 25, no. 1, pp. 109–133, 2012.

[5] M. Breunig, H. Kriegel, R. Ng, and J. Sander, "Lof: identifying density-based local outliers," *ACM SIGMOD Records*, vol. 29, no. 2, pp. 93–104, 2000.

[6] B. Scholkopf, A. Smola, R. Williamson, and P. Bartlett, "New support vector algorithms," *Neural Computing*, vol. 12, no. 5, pp. 1207–1245, 2000.

[7] K. Noto, S. Majidi, A. G. Edlow, H. C. Wick, D. W. Bianchi, and D. K. Slonim, "Csax: Characterizing systematic anomalies in expression data," *Journal of Computational Biology*, vol. 22, no. 5, pp. 402–13, 2015.

[8] D. Newman and A. Asuncion, "UCI machine learning repository," http://www.ics.uci.edu/mlearn/MLRepository.html, 2007.

[9] K. A. Spackman, "Signal detection theory: Valuable tools for evaluating inductive learning," in *Proceedings of the Sixth International Workshop on Machine Learning*, 1989, pp. 160–163.

[10] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Contemporary mathematics*, vol. 26, no. 189-206, p. 1, 1984.

[11] D. Achlioptas, "Database-friendly random projections: Johnson-lindenstrauss with binary coins," *Journal of computer and System Sciences*, vol. 66, no. 4, pp. 671–687, 2003.

[12] A. Kabán, "Improved bounds on the dot product under random projection and random sign projection," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 487–496.

[13] M. Rosenblatt *et al.*, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832–837, 1956.

[14] X. Lu, X. Lu, Z. C. Wang, J. D. Iglehart, X. Zhang, and A. L. Richardson, "Predicting features of breast cancer with gene expression patterns," *Breast cancer research and treatment*, vol. 108, no. 2, pp. 191–201, 2008.

[15] N. Niu, Y. Qin, B. L. Fridley, J. Hou, K. R. Kalari, M. Zhu, T.-Y. Wu, G. D. Jenkins, A. Batzler, and L. Wang, "Radiation pharmacogenomics: a genome-wide association approach to identify radiation response biomarkers using human lymphoblastoid cell lines," *Genome research*, vol. 20, no. 11, pp. 1482–1492, 2010.

[16] K. R. Kalari, S. J. Hebbring, H. S. Chai, L. Li, J.-P. A. Kocher, L. Wang, and R. M. Weinshilboum, "Copy number variation and cytidine analogue cytotoxicity: a genome-wide association approach," *BMC genomics*, vol. 11, no. 1, p. 1, 2010.

[17] L. Li, B. L. Fridley, K. Kalari, G. Jenkins, A. Batzler, R. M. Weinshilboum, and L. Wang, "Gemcitabine and arabinosylcytosin pharmacogenomics: genome-wide association and drug response biomarkers," *PloS one*, vol. 4, no. 11, p. e7765, 2009.

[18] B. L. Fridley, G. Jenkins, D. J. Schaid, and L. Wang, "A bayesian hierarchical nonlinear model for assessing the association between genetic variation and drug cytotoxicity," *Statistics in medicine*, vol. 28, no. 21, pp. 2709–2722, 2009.

[19] L. Li, B. Fridley, K. Kalari, G. Jenkins, A. Batzler, S. Safgren, M. Hildebrandt, M. Ames, D. Schaid, and L. Wang, "Gemcitabine and cytosine arabinoside cytotoxicity: association with lymphoblastoid cell expression," *Cancer research*, vol. 68, no. 17, pp. 7050–7058, 2008.

[20] J. N. Ingle, D. J. Schaid, P. E. Goss, M. Liu, T. Mushiroda, J.-A. W. Chapman, M. Kubo, G. D. Jenkins, A. Batzler, L. Shepherd *et al.*, "Genome-wide associations and functional genomic studies of musculoskeletal adverse events in women receiving aromatase inhibitors," *Journal of Clinical Oncology*, vol. 28, no. 31, pp. 4674–4682, 2010.

[21] X.-L. Tan, A. M. Moyer, B. L. Fridley, D. J. Schaid, N. Niu, A. J. Batzler, G. D. Jenkins, R. P. Abo, L. Li, J. M. Cunningham *et al.*, "Genetic variation predicting cisplatin cytotoxicity associated with overall survival in lung cancer patients receiving platinum-based chemotherapy," *Clinical cancer research*, vol. 17, no. 17, pp. 5801–5811, 2011.

[22] T. Sørlie, R. Tibshirani, J. Parker, T. Hastie, J. S. Marron, A. Nobel, S. Deng, H. Johnsen, R. Pesich, S. Geisler *et al.*, "Repeated observation of breast tumor subtypes in independent gene expression data sets," *Proceedings of the National Academy of Sciences*, vol. 100, no. 14, pp. 8418–8423, 2003.

[23] A. H. Bild, G. Yao, J. T. Chang, Q. Wang, A. Potti, D. Chasse, M.-B. Joshi, D. Harpole, J. M. Lancaster, A. Berchuck *et al.*, "Oncogenic pathway signatures in human cancers as a guide to targeted therapies," *Nature*, vol. 439, no. 7074, pp. 353–357, 2006.

[24] J. O. Boyle, Z. H. Gümüş, A. Kacker, V. L. Choksi, J. M. Bocker, X. K. Zhou, R. K. Yantiss, D. B. Hughes, B. Du, B. L. Judson *et al.*, "Effects of cigarette smoke on the human oral mucosal transcriptome," *Cancer Prevention Research*, vol. 3, no. 3, pp. 266–278, 2010.

[25] N. Novershtern, A. Subramanian, L. N. Lawton, R. H. Mak, W. N. Haining, M. E. McConkey, N. Habib, N. Yosef, C. Y. Chang, T. Shay *et al.*, "Densely interconnected transcriptional circuits control cell states in human hematopoiesis," *Cell*, vol. 144, no. 2, pp. 296–309, 2011.

[26] P. Szatmari, A. D. Paterson, L. Zwaigenbaum, W. Roberts, J. Brian, X.-Q. Liu, J. B. Vincent, J. L. Skaug, A. P. Thompson, L. Senman *et al.*, "Mapping autism risk loci using genetic linkage and chromosomal rearrangements," *Nature genetics*, vol. 39, no. 3, pp. 319–328, 2007.

[27] I. H. . Consortium *et al.*, "Integrating common and rare genetic variation in diverse human populations," *Nature*, vol. 467, no. 7311, pp. 52–58, 2010.

[28] R. Redon, S. Ishikawa, K. R. Fitch, L. Feuk, G. H. Perry, T. D. Andrews, H. Fiegler, M. H. Shapero, A. R. Carson, W. Chen *et al.*, "Global variation in copy number in the human genome," *nature*, vol. 444, no. 7118, pp. 444–454, 2006.

[29] R. Scholtysik, M. Kreuz, W. Klapper, B. Burkhardt, A. C. Feller, M. Hummel, M. Loeffler, M. Rosolowski, C. Schwaenen, R. Spang *et al.*, "Detection of genomic aberrations in molecularly defined burkitts lymphoma by array-based, high resolution, single nucleotide polymorphism analysis," *Haematologica*, vol. 95, no. 12, pp. 2047–2055, 2010.

[30] T. Vrijenhoek, J. E. Buizer-Voskamp, I. van der Stelt, E. Strengman, G. Risk, C. Sabatti, A. G. van Kessel, H. G. Brunner, R. A. Ophoff, J. A. Veltman *et al.*, "Recurrent cnvs disrupt three candidate genes in schizophrenia patients," *The American Journal of Human Genetics*, vol. 83, no. 4, pp. 504–510, 2008.

[31] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[32] M. S. Gashler, "Waffles: A machine learning toolkit," *Journal of Machine Learning Research*, vol. MLOSS 12, pp. 2383–2387, July 2011. [Online]. Available: http://www.jmlr.org/papers/volume12/gashler11a/gashler11a.pdf

[33] S. Mah, M. Nelson, L. Delisi, R. Reneland, N. Markward, M. James, D. Nyholt, N. Hayward, H. Handoko, B. Mowry *et al.*, "Identification of the semaphorin receptor plxna2 as a candidate for susceptibility to schizophrenia," *Molecular psychiatry*, vol. 11, no. 5, pp. 471–478, 2006.

[34] L. Martucci, A. H. Wong, V. De Luca, O. Likhodi, G. W. Wong, N. King, and J. L. Kennedy, "N-methyl-d-aspartate receptor nr2b subunit gene grin2b in schizophrenia and bipolar disorder: Polymorphisms and mrna levels," *Schizophrenia research*, vol. 84, no. 2, pp. 214–221, 2006.

[35] J. Piñero, N. Queralt-Rosinach, À. Bravo, J. Deu-Pons, A. Bauer-Mehren, M. Baron, F. Sanz, and L. I. Furlong, "Disgenet: a discovery platform for the dynamical exploration of human diseases and their genes," *Database*, vol. 2015, p. bav028, 2015.

[36] G. Dennis, B. T. Sherman, D. A. Hosack, J. Yang, W. Gao, H. C. Lane, and R. A. Lempicki, "DAVID: Database for Annotation, Visualization, and Integrated Discovery," *Genome Biol.*, vol. 4, no. 5, p. P3, 2003.

[37] J. A. Blake, K. R. Christie, M. E. Dolan, H. J. Drabkin, D. P. Hill, L. Ni, D. Sitnikov, S. Burgess, T. Buza, C. Gresham *et al.*, "Gene Ontology Consortium: going forward," *Nucleic Acids Res.*, vol. 43, no. Database issue, pp. D1049–1056, Jan 2015.

[38] M. H. Sieweke, H. Tekotte, J. Frampton, and T. Graf, "MafB represses erythroid genes and differentiation through direct interaction with c-Ets-1," *Leukemia*, vol. 11 Suppl 3, pp. 486–488, Apr 1997.

[39] R. Krzysiek, E. A. Lefevre, J. Bernard, A. Foussat, P. Galanaud, F. Louache, and Y. Richard, "Regulation of CCR6 chemokine receptor expression and responsiveness to macrophage inflammatory protein-3alpha/CCL20 in human B cells," *Blood*, vol. 96, no. 7, pp. 2338–2345, Oct 2000.