BIOINFORMATICS

Vol. 28 ISMB 2012, pages i106–i114
doi:10.1093/bioinformatics/bts232

Ranking of multidimensional drug profiling data by fractional-adjusted bi-partitional scores

Dorit S. Hochbaum[1,†] Chun-Nan Hsu[2,3,†] and Yan T. Yang[1,*]

[1]Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720, [2]Information Sciences Institute, University of Southern California, Marina del Ray, CA 90292 and [3]Institute of Information Science, Academia Sinica, Taipei 115, Taiwan

ABSTRACT

Motivation: The recent development of high-throughput drug profiling (high content screening or HCS) provides a large amount of quantitative multidimensional data. Despite its potentials, it poses several challenges for academia and industry analysts alike. This is especially true for ranking the effectiveness of several drugs from many thousands of images directly. This paper introduces, for the first time, a new framework for automatically ordering the performance of drugs, called fractional adjusted bi-partitional score (FABS). This general strategy takes advantage of graph-based formulations and solutions and avoids many shortfalls of traditionally used methods in practice. We experimented with FABS framework by implementing it with a specific algorithm, a variant of normalized cut—normalized cut prime (FABS-NC′), producing a ranking of drugs. This algorithm is known to run in polynomial time and therefore can scale well in high-throughput applications.

Results: We compare the performance of FABS-NC′ to other methods that could be used for drugs ranking. We devise two variants of the FABS algorithm: FABS-SVM that utilizes support vector machine (SVM) as black box, and FABS-Spectral that utilizes the eigenvector technique (spectral) as black box. We compare the performance of FABS-NC′ also to three other methods that have been previously considered: center ranking (Center), PCA ranking (PCA), and graph transition energy method (GTEM). The conclusion is encouraging: FABS-NC′ consistently outperforms all these five alternatives. FABS-SVM has the second best performance among these six methods, but is far behind FABS-NC′: In some cases FABS-NC′ produces over half correctly predicted ranking experiment trials than FABS-SVM.

Availability: The system and data for the evaluation reported here will be made available upon request to the authors after this manuscript is accepted for publication.

Contact: yxy128@berkeley.edu

1 INTRODUCTION

Automated microscopy is increasingly used in drug discovery, especially predicting the toxicity of new drugs (Perlman and Altschuler, 2004). The so-called high-content screening (HCS) has greatly enhanced investigators' capability of discerning the response of cells treated by various drugs (Conrad and Gerlich, 2010; Denner *et al.*, 2008; Feng *et al.*, 2009; Lang *et al.*, 2006; Mitchison, 2005a; Nichols, 2007; Taylor and Hsaskins, 2007). HCS accomplishes this by analyzing phenotypic features of the cells from tens of thousands cell images produced by HCS. In addition, the decreasing cost of such a method means a wide-spread application (Lin *et al.*, 2010). HCS employs cell imaging assays, tagged with fluorescent dyes—each field of cells contains these tags for its different macromolecules. Automated microscopy is performed to produce a large amount of visual information.

There are three steps during this process (Mitchison, 2005a; Yarrow *et al.*, 2003): fluorescence-tagging, automated microscopy and identification and measurement of target phenotypic feature(s) for further analysis. The analysis step usually poses the most challenge. To extract meaning out of a gigantic image database, traditional tools usually need to be tailored to specific known phenotype's features, instead of unknown yet more informative differences. For example, it has been reported that applying an analysis method that only distinguishes phenotypic changes in cellular level misses on the detecting meaningful morphological modification on subcellular structures (Taguchi *et al.*, 2007; Zhou and Wong, 2006).

In high-throughput drug screening assays, typically a quantity, such as normalized intensity of a reporter fluorescent protein (Morelock *et al.*, 2005), is assumed to be measurable. Differences between samples of two distinct cell populations (such as treated versus untreated) are estimated and tested for significance. Methods using statistics like Z'-factor (Zhang *et al.*, 1999) to evaluate reliability of the measurements have been developed. Comparison of the difference is usually done by performing a multivariate F-test to test whether two populations are distributed differently. But F-test may introduce high errors when the distributions are not normal, which is expected to be the case in many types of cell responses. Moreover, in image-based assays, the use of a measurable quantity is no longer applicable when this quantity is not straightforward to obtain directly and the measurement itself can never be perfect. For example, to measure the composition of morphological subtypes of mitochondria requires pattern recognition algorithms to accurately detect and quantify target events (Peng *et al.*, 2011). Though many advanced algorithms have been developed for years, these pattern recognition algorithms usually require non-trivial tuning and optimization for each study because they may generalize poorly, sometimes not even generalize within a well, due to noise and systematic bias introduced during the sample preparation and imaging process steps, inducing additional overhead when attempts are made to scale up the assay to high throughput.

Another challenge is when a multiplex approach is required, where multiple independent quantities are measured for each single cell. In these cases, response of each single cell will be a

*To whom correspondence should be addressed.
†The author wish it to be known that, in their opinion, the first two authors should be regarded as the joint first Authors.

multi-dimensional vector. How to measure difference between these vectors become an issue because simple Euclidean distance in the multi-dimensional space may not serve the need. One solution is to come up with an appropriate 'metric' to convert multi-dimensional vectors into a scalar that reflects the difference. There is, however, no generally applicable solution about how to come up with this metric. Usually, one or more dimensions in the vector come from an imperfectly measured quantity, such as one that requires advanced pattern recognition in order to automatically extract, as discussed in the previous paragraph. Another issue is that our observation is the result of sampling, which inevitably introduces sampling errors and is further complicated by possible heterogeneous responses by cells (Altschuler and Wu, 2010).

The focus of this research is to address the issues mentioned above for the application of HCS in drug ranking. Drug ranking refers to the ordering of a group of different drugs according to their effectiveness by certain criteria. One of the most used criteria is the relative toxicity among drugs (Paull *et al.*, 1992). Ideally, this provides the important scale to assess relative merit of each candidate drug. However, each cell responds to a certain drug differently, thus making the outcome of any ranking highly dependent on sampling and noise. A conspicuous example is the fragmentation of cells or organelles: the intact and the completely fragmented states are easy to recognize while the degree of partial fragmentation is difficult to gage, thus often involving human experts and time-consuming manual processes. This is infeasible for high-throughput screening such as HCS (Lin *et al.*, 2010; Peng *et al.*, 2011).

Our objective is to develop an efficient and accurate ranking measure (metric learning) that can be used to order candidate drugs according to their effectiveness. To this end, we developed a framework called *Fractional Adjusted Bi-partitional Score* (FABS). This general strategy, introduced here for the first time, takes advantages of graph-based formulations and solutions and avoids many shortfalls of traditionally used methods in practice. We use such a scheme because graph-based construction works well in several areas of data mining (Washio and Motoda, 2003), machine learning (Jordan, 1996) and image processing (Hochbaum, 2001), whereas a recent publication (Lin *et al.*, 2010) also confirms its usefulness in the HCS context.

In order to apply our FABS to the images, we use a feature extraction tool first presented in (Peng *et al.*, 2011). This tool takes cell images and output several vectors that represents important geometric and other features of the target images—these vectors are then used as inputs for getting FABS.

One feature of FABS is that it has, as part of the input and as training data, extreme cases labeled as positive and negative controls, which in our case are the intact and the completely fragmented states mentioned previously. The algorithm does not involve any training from in-between cases, which are hard to come by. This completely sidesteps the common problem of a laborious and time-consuming annotation step, performed by experts to assess the relative merit of drugs for a small sample of images used as a training group. Furthermore, our measure takes the advantage of high-volume nature of the dataset, using all available images for computation of FABS for each drug. This reduces the effect of noise and sampling bias. This framework can potentially be used for any task that requires to quantify subtle and implicit differences between populations of high-dimensional feature vectors. By formulating the problem as a biparition problem as in FABS, there is no need to solve an image-based drug ranking problem as a regression problem. Our preliminary formal analysis of FABS shows that the expected error and variance of the estimated scores by FABS will be within a manageable range given the classification error by the bipartition.

To empirically evaluate our framework, we use a model of (NC′) and the respective algorithm recently introduced by Hochbaum (2010c). That algorithm runs efficiently and is furthermore combinatorial. This latter feature differentiates it from ref. (Lin *et al.*, 2010) in which a spectral techniques is used to achieve a bipartitioning. Combinatorial solutions are superior than spectral ones in several regards such as being more efficient and accurate (Hochbaum, 2010b,c), as shown in our experimental results.

2 METHODOLOGY

This section presents a general framework for quantifying the difference in morphological composition between populations of cells. The proposed framework utilizes a procedure named FABS-$\mathcal{A}$, where $\mathcal{A}$ stands for a bipartition algorithm and FABS stands for (*FABS*). We show that using certain graph, theoretical formulations for the bipartition algorithm avoids many shortfalls of the methods used in practice. Its importance lies in teasing apart cell groups based on morphological composition and in detecting whether or not such differences exist.

As previously mentioned, we use a feature extraction tool, capable of processing cell images with different dimensionalities (from static 2D to animated 3D with multiple channels) to generate high-dimensional (in our experiments, 134 D) output vectors, called *feature vectors*. Each feature vector, corresponds to an image of a single cell and contains measurements for the image characteristics, such as the intensity of the image, the shape of a particular object in the image, etc. Each group of cell images (and their corresponding feature vectors) can be associated with a certain population (e.g. populations representing cells to whom a certain drug has been applied).

The method proposed in this section, FABS-$\mathcal{A}$, is capable of receiving—as input—the feature vectors from cells representing different populations and detecting and quantifying the differences between these populations. For example, given the features extracted from the mitochondrial images of two populations of cells, one derived from diseased tissues and the other from healthy tissues, FABS-$\mathcal{A}$ will tell us to what extent the fragmentation levels of their mitochondria are different and estimate the significance of the difference.

We then perform FABS-$\mathcal{A}$ on the processed feature vectors. The input to FABS-$\mathcal{A}$ is the processed feature vectors by principal component analysis (PCA) to reduce dimensionalities of the original data , each of which belongs to a certain population set, namely, P_i, and training data. The training data consists of feature vectors belonging to two populations on the opposite ends of the spectrum, R_1 and R_2. These two population sets represent positive and negative controls, which in this experiment are the completely fragmented and the completely intact mitochondria cell populations.

Computation of FABS-$\mathcal{A}$, the details of which will be discussed shortly, consists of three steps: The first step is to construct a graph from the input data. The second step is to apply a blackbox algorithm ($\mathcal{A}$) to find a *bipartition* on the resulting graph. The third step is to recover a scalar score for each population, based on the fraction of the cases that fall in the side of the partition boundary (cut) that

contains positive controls. The blackbox can be any appropriate bipartitioning algorithm available. The algorithm we propose to use for the blackbox solves the *normalized cut prime* (NC′) problem (Hochbaum, 2010b). We refer to this algorithm as NC′. We shall see in the 'Results' section that this bipartitioning algorithm, in the context of FABS-$\mathcal{A}$ (FABS-NC′), outperforms Support Vector Machine (SVM) algorithm (FABS-SVM). This overall framework provides a flexible general strategy for quantifying the differences among population groups.

The major advantages of FABS-$\mathcal{A}$ include:

(1) it is capable of efficiently processing the high-dimensional input data acquired from the images using feature extraction tool from Peng *et al.* (2011);

(2) the generated output is one-dimensional, in that a single scalar score is generated for each population of multidimensional vectors. As such, the difference between the scores can be used to quantify population differences in an unambiguous way;

(3) the calculation of the output scores is done in a way that reduces the effects of outliers in distinguishing cell populations;

(4) unlike many statistical tests, it does not assume any underlying distribution for the populations;

(5) the labeled training data set required is minimal and easily obtainable, requiring minimum intervention from the experts; and

(6) it scales well in high-throughput applications.

In what follows we describe the three steps of FABS-$\mathcal{A}$ in more details.

2.1 The FABS-$\mathcal{A}$ Algorithm

Step 1: graph construction

As mentioned previously, the input to FABS-$\mathcal{A}$ consists of n (pre-processed) feature vectors, V=$\{v_1,...,v_n\}$, each associated with an HCS image, obtained after feature extraction and PCA pre-processing. This input includes k population sets, $\{P_1,...,P_k\}$. Each population set in this case represents a set of feature vectors corresponding to cells treated with a certain drug. Each feature vector v_i belongs to one of the population sets, indicating in this case what drug has been applied to the particular cell the vector is representing. The input also contains two training sets $\{R_1,R_2\}$, representing the extreme cases such as the completely fragmented and the completely intact mitochondria cell populations. In the graph construction step of FABS-$\mathcal{A}$, an undirected graph $G=(V,E,\mathbf{l},\mathbf{w})$ is created, where each node $v_i \in V$ corresponds to a feature vector. The set of all possible pairs correspond to the set of edges of the graph $E=V\times V$ that form a complete graph. Each feature vector v_i is labeled with l_{v_i}, which is the index of the population set it belongs to. The labeling function, l_{v_i}, assigns a mapping from each feature vector, v_i, to its corresponding population set, determining which population it belongs to. A weight function $w\colon V\times V \rightarrow \Re^+$ associates with each pair of nodes $\{i,j\}$ (an edge) its encoding connection strength, or the similarity strength between the two nodes. For each edge $[i,j]$, the weight w_{ij} and the distance between the two points v_i and v_j have the relationship: one goes up as the other goes down (or vice versa)—this also means that w_{ij} and the similarity between v_i and v_j both go up or down together. Several distance measures can be used for this purpose, among them, Euclidean, city block and Minkowski distances. Notice that constructing these similarity measures makes the dimensionality of the vectors irrelevant to our algorithm.

Step 2: bipartitioning the graph using NC′

We first introduce some notations; given a graph $G=(V,E)$, a bipartition of the graph, or a cut, is defined as $(S,\overline{S})=\{[i,j]|i\in S,j\in \overline{S}\}$, where $\overline{S}=V\setminus S$. The *capacity of a cut* $(S,\overline{S})$, is defined as:

$$C(S,\overline{S})=\sum_{i\in S,j\in\overline{S},[i,j]\in E} w_{ij}.$$

More generally, for any pair of sets $A,B\subseteq V$, the capacity of the cut is denoted by $C(A,B)=\sum_{i\in A,j\in B} w_{ij}$. Similarly, the *capacity of a set*, $D\subset V$, is denoted by $C(D)=C(D,D)=\sum_{i,j\in D,[i,j]\in E} w_{ij}$.

As previously mentioned, in the second step of FABS-$\mathcal{A}$, we use a blackbox algorithm to find a bipartition on the graph. A bipartition algorithm aims at finding the cut that separates the graph into S and $\overline{S}$, according to some underlying objectives. There are many different objectives that can be selected. For instance, the bipartition algorithm for the well-known minimum cut problem is defined with the goal of separating the graph into S and $\overline{S}$ such that $C(S,\overline{S})$ is the minimum among all possible non-empty subsets S and $\overline{S}$. Since the goal is to obtain a bipartition for the FABS-$\mathcal{A}$ calculation process, any bipartition algorithm can be used as a blackbox. However, an extra requirement has to be imposed (either by the internal working of the algorithm or by an external constraint) listed as follows.

REQUIREMENT 1. All positive controls $\mathtt{R}_1$ must be in S (or $\overline{S}$) and all negative controls $\mathtt{R}_2$ must be in $\overline{S}$ (or S).

For a particular blackbox implementation of FABS-$\mathcal{A}$ in Step 2 of Algorithm 1, we choose the previously mentioned bipartitioning algorithm, called NC′, and adjust it to guarantee that the constraint listed in Requirement 1 is satisfied. The resulting FABS-NC′ is semi-supervised in nature and incorporates all information of the corresponding graph. The NC′ problem is defined as finding $\min_{S\subset V} C(S,\overline{S})/C(S,S)$ on a given graph. This objective combines the goal of minimizing the similarity between the two parts of the bipartition, the quantity $C(S,\overline{S})$, with the goal of maximizing the similarity between the elements of S. For a graph $G=(V,E)$, we denote $\mathrm{NC}'(G)=\min_{S\subset V} C(S,\overline{S})/C(S,S)$. An efficient algorithm for this problem was given in (Hochbaum, 2010a,b,c).

The polynomial time algorithm described in (Hochbaum, 2010b) for NC′ was based on showing that solving NC′ is equivalent to solving a certain *parametric s,t-cut* problem. In an s,t-cut problem a node of a graph s is required to be on one side of the bipartition, whereas the node t is required to be on the opposite side.

In the adaptation of the parametric s,t-cut algorithm for the FABS-$\mathcal{A}$ framework, the positive and negative control data are used as *seed nodes* that are forced to join s and t in the graph. This is achieved through setting the nodes in R_1 to be 'infinitely similar' to the source node s, and the nodes of R_2 to be 'infinitely similar' to the sink node t. In terms of the graph that means that we add edges of infinite weight between the source node s and all nodes in R_1, and edges of infinite weight between the nodes of R_2 and t.

Since NC′ can be solved in the running time of a minimum s,t-cut problem (Hochbaum, 2010b), our FABS-NC′ implementation is efficient, solving in polynomial time. We later compare the performance of FABS-NC′, with FABS-SVM, where the bipartitioning algorithm used is SVM, whose objective is to find a high-dimensional hyperplane that is as wide as possible to separate data of different labels (Cristianini and Shawe-Taylor, 2000).

Step 3: computing FABS scores

After a bipartition algorithm has been applied on G, all feature vectors in the graph are partitioned into S and $\overline{S}$. In the third step of FABS-$\mathcal{A}$, a scalar score, $\texttt{FABS}_{P_i}$, is calculated for each population set P_i. $\texttt{FABS}_{P_i}$ is the fraction of the number of feature vectors in P_i that fall in the set S, to the total number of feature vectors in P_i. Formally,

$$\texttt{FABS}_{P_i} = \frac{|S \cap P_i|}{|P_i|}.$$

This is shown pictorially in Figure 1. The FABS scores of the populations are then used to rank them: the higher the FABS score the closer is the population to R_1. The FABS scores are therefore ordered so that $\texttt{FABS}_{P_{\pi(1)}} \geq \texttt{FABS}_{P_{\pi(2)}} \geq \ldots \geq \texttt{FABS}_{P_{\pi(k)}}$, where $(\pi(1), \ldots \pi(k))$ is a permutation of $(1, 2, \ldots, k)$. The ranking of the populations is then given by $(\pi(1), \ldots \pi(k))$.

The entire procedure is summarized in Algorithm 1.

Algorithm 1 FABS-$\mathcal{A}$

Inputs: The feature vectors $\{v_1, ..., v_n\}$ extracted from images (possibly after PCA pre-processing), and their corresponding population sets $\{P_1, ..., P_k\}$; The training data (or extreme sets) $\{R_1, R_2\}$

Step 1: Construct $G = \{V, E, \mathbf{l}, \mathbf{w}\}$, a complete graph from feature vectors;

Step 2: Use a bipartitioning algorithm $\mathcal{A}$ to find a bipartition $(S, \overline{S})$ on G such $R_1 \subseteq S$ and $R_2 \subseteq \overline{S}$;

Step 3: $\forall P_i$, calculate $\texttt{FABS}_{P_i} = \frac{|S \cap P_i|}{|P_i|}$

Step 4: The FABS scores are ordered so that $\texttt{FABS}_{P_{\pi(1)}} \geq \texttt{FABS}_{P_{\pi(2)}} \geq \ldots \geq \texttt{FABS}_{P_{\pi(k)}}$, where $(\pi(1), \ldots \pi(k))$ is a permutation of $(1, 2, \ldots, k)$. The ranking of the populations is then given by $(\pi(1), \ldots \pi(k))$;

Output: An ordered array of population sets based on their FABS score, $\{R_1, P_{\pi_1}, ..., P_{\pi_k}, R_2\}$.

2.2 Significance test

One can further use the FABS scores to test statistical significance of the difference between the effects of two drugs. The idea is to apply bootstrapping to obtain FABS scores from a large number of resampling trials and then perform hypothesis test on the difference of the distributions of FABS. Algorithm 2 gives the test procedure, which takes resulting FABS from repeated experiment and calculate P-values from a t-test for each drug. The obtained P-value is then transformed into a log score $-\log p$.

To see if t-test is appropriate, there are several important assumptions to check. First, the sets of FABS of two drugs must each be normally distributed. We plotted a histogram of FABS scores obtained by our FABS-SVM implementation and observed

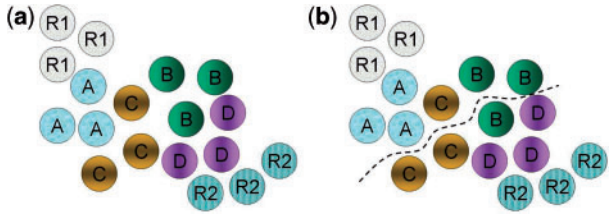

Fig. 1. (**a**) The input with the feature vectors of images associated with positive and negative controls R_1 and R_2 and four different drugs `drug A`, `drug B`, `drug C` and `drug D`; (**b**) The bipartition boundary after the cut is found: if R_2 contains negative controls, such as the completely fragmented state of mitochondria for toxicity criterion, while R_1 contains positive controls, representing cells in a desired normal healthy state with mitochondria rescued from the completely fragmented, then $\texttt{FABS}_{\texttt{drug A}} = 1$, $\texttt{FABS}_{\texttt{drug B}} = 2/3$, $\texttt{FABS}_{\texttt{drug C}} = 1/3$, and $\texttt{FABS}_{\texttt{drug D}} = 0$. Our ranking of the drugs will be: `drug A >> drug B >> drug C >> drug D`, where $x >> y$ indicates that x is more effective than y

that the distributions for each drug in our test data are roughly bell shaped. In addition, for Z-IETD and Z-LEHD, the P-values obtained through Jarque-Bera test (Jarque and Bera, 1987) are 0.5 and 0.0718, respectively, indicating approximate normality for both. Another assumption is that variance for each group must be equal. Though this is usually not the case in drug profiling applications, t-test is robust against unequal variances if the sample sizes are approximately equal for each group, which can be enforced in drug profiling applications. Other assumptions, such as that sample means and sample variances must be statistically independent, can be compensated when the sample is moderately large or larger, which is always the case for HCS. Consequently, the t-test is appropriate for our purposes. When the number of population is high, we can apply Bonferroni correction to avoid errors due to multiple comparisons.

Algorithm 2 Significance test

Step 1: Collect FABS from all subsampling trials for each drug, i.e. randomly sample certain percentages of controls and drugs with replacement from the original database repeatedly and calculate FABS score per drug each time;

Step 2: Perform t-test on FABS obtained with any two different drugs. T-test of `drug A` and `drug B` returns a P-value, $p_{(\texttt{drug A},\texttt{drug B})}$;

Step 3: Return $-\log p_{(\texttt{drug A},\texttt{drug B})}$

2.3 Data preparation

We use a subset of a large image database of Chinese Hamster Ovary cells published in Peng *et al.* (2011). The cells are divided into four groups according to the drug treatments they have received—control, squamocin, squamocin and z-IETD (shortened as z-IETD), and squamocin and z-LEHD (shortened as z-LEHD). Squamocin is known to induce mitochondrial fragmentation and cell apoptosis (i.e. programmed cell death). z-IETD and z-LEHD are inhibitors of caspases that play important roles in mitochondrial fragmentation. The goal of the study was to investigate whether

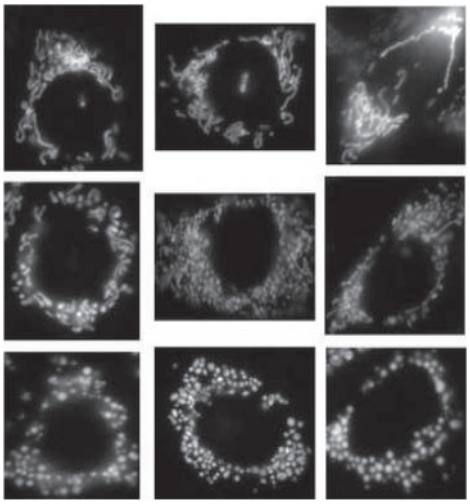

Fig. 2. Example cell images show different fragmentation stages of mitochondria, tagged with a fluorescent dye. Images at the bottom row are cells with the completely fragmented mitochondria, at the top row are those without fragmentation, those in the middle are partially fragmented. From (Lin *et al.*, 2010)

z-IETD and z-LEHD can recover mitochondria from squamosin-induced fragmentation. Figure 2 shows some example cell images of mitochondria at different fragmentation stages. Intact mitochondria usually appear like threads, as shown in the images at the top row, whereas fragmented mitochondria appear like small globules as shown at the bottom row. Even though the totally intact and totally fragmented mitochondria (extreme set cases) can be easily distinguished by visual inspection, it is very hard (if not impossible) to look at a set of mitochondria images that are neither totally intact nor totally fragmented (e.g. a set of mitochondria images representing a population set of say cells treated by a certain drug) and distinguish between these different population sets and determine which extreme sets they are closest to and how they compare against each other (in terms of level of fragmentation). Another challenge is to automate this process. The automation process is critical, because the biological data sets available are very large and screening them manually could be a very time-consuming and laborious task.

The challenge is to quantify and rank partial fragmentation as shown in the middle row. (Peng *et al.*, 2011) concluded that z-LEHD was more effective than z-IETD in rescuing mitochondria from squamocin-induced fragmentation. This conclusion was used as the ground truth to assess the prediction accuracy of different methods later and images treated by squamocin and control were used as extreme cases.

Our database contains 257 images of cells treated with squamocin, 239 with z-IETD, 262 with z-LEHD and 238 control. We applied a feature extraction method to extract 135 features from each cell image to form the feature vector to represent each cell. This feature extraction method is the same as the one that was used to extract *strong detectors* from cell images to determine protein subcellular localization as described by Lin *et al.* (2007). Strong detectors include general purpose features derived from image transformations, such as Haralick texture features and geometric features of the objects extracted from the input image. These features have been shown to be useful in problems like recognizing fluorescent patterns of subcelluar organelles in protein subcellular localization (Huang and Murphy, 2004).

3 RESULTS

3.1 Formal Analysis of FABS-$\mathcal{A}$

Here, we formally define the drug ranking problem and report a bias–variance analysis of FABS-$\mathcal{A}$ as a solution to this problem. The drug ranking problem can be considered as a regression problem, where given a multi-dimensional observation $v_i = X \in \Re^d$, we assume that a quantity $Y \in [-1, +1]$ is associated with X as our target metric of X. A solution of this regression problem is to learn a regression model from examples that compute Y given X. With the metric quantity Y, given two treatments a and b with population distributions $\mathcal{P}_a$ and $\mathcal{P}_b$, respectively, if

$$\mathbb{E}_{\mathcal{P}_a}(Y|X) - \mathbb{E}_{\mathcal{P}_b}(Y|X) \geqslant 0, \tag{1}$$

then treatment a will be considered to be more effective than treatment b, assuming that $Y = +1$ is the desired phenotypic outcome.

However, it is usually infeasible to manually assign score Y for a sufficient number of training examples consistently. Instead, FABS-$\mathcal{A}$ simplifies the problem as a bipartition problem. In our bipartition scheme, our model will assign $Y_c = 1$ to a given X if $Y \geqslant 0$ and $Y_c = -1$ otherwise, and then use empirical population mean as the estimated population mean of Y. In a drug screening application, this quantity will be used to rank the effectiveness of a treatment.

More formally,

$$Y_c = Y + \text{compl}(Y)$$

where

$$\text{compl}(Y) = \begin{cases} 1 - Y & \text{if } Y \geqslant 0 \\ -1 - Y & \text{if } Y < 0 \end{cases}$$

Instead of directly comparing the expectation of Y, FABS-$\mathcal{A}$ compares the expectation of Y_c to determine which treatment is more effective.

$$\mathbb{E}_{\mathcal{P}_a}(Y_c|X) - \mathbb{E}_{\mathcal{P}_b}(Y_c|X) \geqslant 0, \tag{2}$$

Like Y, Y_c is unknown and must be estimated with a model learned from data. Let $\widehat{Y}_c$ be the estimation of Y_c. Then

$$\widehat{Y}_c = \begin{cases} Y + \text{compl}(Y) & \text{if correctly classified} \\ Y + 1 & \text{if incorrect and } Y < 0 \\ Y - 1 & \text{if incorrect and } Y \geqslant 0 \end{cases}$$

An analysis of bias and variance of the bipartition scheme is as follows. The absolute error made by bipartition instead of regression is

$$|Y - Y_c| = |\Delta \widehat{Y}_c| = \begin{cases} |\text{compl}(Y)| = 1 - |Y| & \text{if correct} \\ 1 + |Y| & \text{otherwise} \end{cases}$$

Let ε be the classification error rate of the bipartition model.

$$\begin{aligned} \mathbb{E}(|\Delta \widehat{Y}_c|) &= (1-\varepsilon)(1 - \mathbb{E}(|Y|) + \varepsilon(1 + \mathbb{E}(|Y|)) \\ &= 1 + (2\varepsilon - 1)\mathbb{E}(|Y|) \leqslant 1 \left(\begin{smallmatrix} \text{when } \varepsilon = 0.5 \\ 1 + (1-1)|Y| = 1 \end{smallmatrix} \right) \end{aligned}$$

The expectation of the absolute error is bounded below one when we use a weak classifier for the bipartition that simply guesses a label randomly.

The variance of the absolute error is

$$Var(|\Delta\widehat{Y}_c|)=\mathbb{E}(|\Delta\widehat{Y}_c|^2)-(\mathbb{E}(|\Delta\widehat{Y}_c|))^2$$
$$=4\mathbb{E}(|Y|)^2\varepsilon(1-\varepsilon),$$

which turns out to be the variance of Bernoulli trial scaled with the square of the expected scale of Y. Again, this is bounded by 1 when $\varepsilon=0.5$ and $\mathbb{E}(|Y|)=1$.

Next, we consider the expectation of $\widehat{Y}_c$, which is interesting because we can infer the expected difference between regression (equation 1) and bipartition (equation 2).

$$\Delta\widehat{Y}_c=Y-\widehat{Y}_c=\begin{cases}1-Y & \text{if } Y\geqslant 0 \text{ and correctly classified}\\ -1-Y & \text{if } Y<0 \text{ and correctly classified}\\ -1-Y & \text{if } Y\geqslant 0 \text{ and incorrect}\\ 1-Y & \text{if } Y<0 \text{ and incorrect}\end{cases}$$

Let $P_+=\Pr(Y\geqslant 0|X)$, the probability that $Y\geqslant 0$ and $\overline{Y}=\mathbb{E}(Y|X)$. We have

$$\mathbb{E}(\Delta\widehat{Y}_c)=(1-\varepsilon)P_+(1-\overline{Y})+(1-\varepsilon)(1-P_+)(-1-\overline{Y})+$$
$$\varepsilon P_+(-1-\overline{Y})+\varepsilon(1-P_+)(1-\overline{Y})$$
$$=(2-4\varepsilon)P_+-1+2\varepsilon-\overline{Y}.$$

The result above implies that when we have a weak classifier $\varepsilon\to 0.5$, $\mathbb{E}(\Delta\widehat{Y}_c)\to-\overline{Y}$ and $\mathbb{E}(\widehat{Y}_c)=\overline{Y}+\mathbb{E}(\Delta\widehat{Y}_c)=0$. That is, regardless of the population, random guessing will not give any distinction between any populations and provide no discerning power. In contrast, when we have a perfect classifier with $\varepsilon\to 0$, $\mathbb{E}(\widehat{Y}_c)\to 2P_+-1$, which is to scale the true probability of $Y\geqslant 0$ for the population to $[-1,1]$, perfectly matching our desire. Consequently, given an accurate bipartition algorithm, FABS-$\mathcal{A}$ can reasonably approximate effectiveness of drugs without exact scores the effectiveness.

3.2 Performance of ranking

We compared the performance of FABS-NC$'$ with four other baselines that has been used in HCS—center ranking PCA ranking and graph transition energy method (GTEM) (Lin *et al.*, 2010). Center ranking first finds the center, which can be the mean, the median or any other measure of the center, of all feature vectors associated with a particular drug or an extreme case, then calculate the distance, such as Euclidean distance, between all pairs of centers. The ranking of the drugs are performed by ordering the drugs according to the centers of the closest to the farthest from the center of the desired extreme case (such as the completely fragmented state for toxicity criterion). PCA ranking is similar to center ranking, except it first projects the feature vectors onto the first few important principal components, then performs center ranking. GTEM (Lin *et al.*, 2010) is also a graph-based approach. GTEM defines graph transition energy as the distance metric and utilizes a spectral graph theoretic regularization to transform the feature space so that extreme cases will be separated widely before ranks populations of cells under different treatments.

In addition to use NC$'$ [solved with Hochbaum's PseudoFlow algorithm, HPF, the implementation of which is obtained from (Chandran and Hochbaum, 2009; Hochbaum, 2008, 2010a)] as our bipartition algorithm in the FABS framework, we also tested other bipartition procedures. One classical technique is the SVM (Burges, 1998; Cristianini and Shawe-Taylor, 2000). When using SVM for FABS, we satisfy Requirement 1 by setting training data as the positive and negative controls: all R_1 points are in S and all R_2 points are in $\overline{S}$. To see the performance of this particular implementation of (FABS-SVM), the kernel used is radial basis function and the parameters are the following: C value is 10^4 and the kernel parameter is 1. The implementation package used is LIBSVM (Chang and Lin, 2011).

Another approach, often used in image segmentation is based on finding the Fielder eigenvector of the graph (referred to as the *spectral technique*) as a heuristic solution for the normalized cut problem (Shi and Malik, 2000). The spectral technique however is unsupervised, and thus does not satisfy Requirement 1. To resolve this issue, we modified the weights of the graph to ensure that Requirement 1 is satisfied. The implementation package used is Normalized Cuts Segmentation Code, Timothee Cour (2004). However, its performance was much worse than all other methods and was removed from the results.

The comparative study that we performed used the median for all center measures and Euclidean distance for all distance measures. The edge weights between two feature vectors v_i and v_j increase or decrease in the opposite direction with respect to the distance between them and is quantified by $w_{ij}=e^{-||v_i-v_j||_2+\epsilon}$, for $0<\epsilon\ll 1$.

Prior to feeding the input feature vectors extracted from the images into FABS-$\mathcal{A}$, we first pre-process these vectors to transform them from a high-dimension space to a space of fewer dimensions. In this process, the data are reduced to fewer dimensions, and we only preserve the dimensions that are of the most significance to our experiment. The dimension reduction is performed by using PCA and the number of principal components used is determined by adding the largest number of most significant principal components that explain up to 80% of the total variation in the dataset considered. We also tested whether applying GTEMs feature transformation step as a preprocessing step before applying FABS-NC$'$ may improve the performance.

To guarantee statistical validity of our comparison, we subsampled the available cell images from the entire database, i.e. we drew samples with replacement for certain percentage from the database to test methods. The subsampling percentages 30, 60, 70 and 80% tried for drug images (501 images). For each fixed drug percentage, we changed percentages of labeled controls by increasing from 10% to 100% to see the effects of the number of labeled controls on the final prediction accuracy of the ranking (495 images in total). The subsampling trials are performed 1000 times for each combination. The *prediction accuracy* of any ranking method is the fraction of correctly ranked trials—this can be determined, since we have the ground truth—out of the grand total of 1000 trials.

Figures 3 and 4 graphically summarize the results in the experiment. Each graph shown is for 30, 60, 70 or 80% fixed drug percentage (testing percentage). The x-axis is the percentage of labeled controls used, whereas y-axis displays the average prediction accuracy over 1000 trials described in Section 3.2.

Each curve in the graphs indicates a particular ranking method—they include FABS-NC$'$, FABS-SVM, Center ranking, PCA ranking, GTEM. The results of FABS-Spectral is poor with our particular implementation and from the figures. The vertical lines in Figures 3

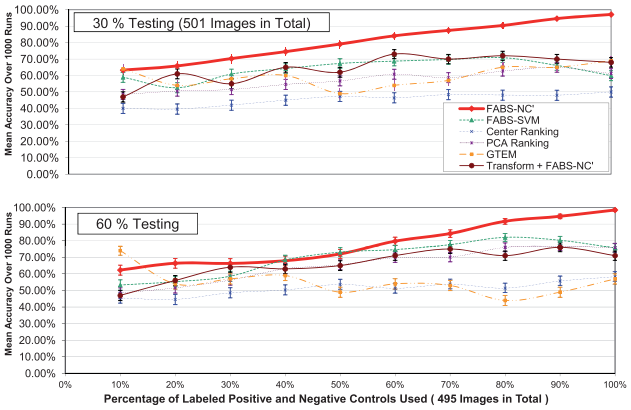

Fig. 3. The accuracy comparison among different ranking methods. The vertical bars in the graph are 95% confidence intervals. The testing percentages used are: 30 and 60%

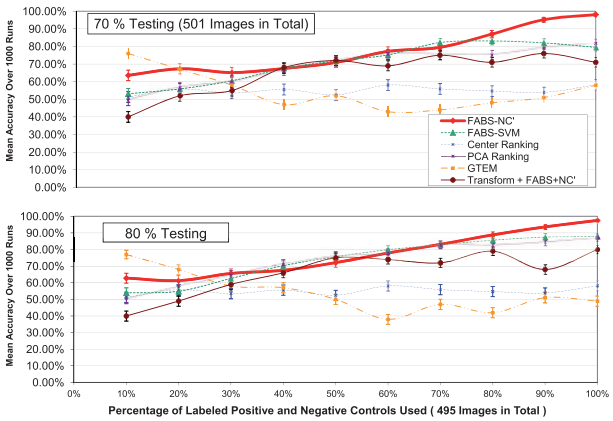

Fig. 4. (Continued) the testing percentages used are: 70 and 80%

and 4 are 95% confidence intervals for the accuracy of each ranking method.

For all testing percentages, the prediction accuracy of FABS-NC′ steadily increases as more labeled controls become available, especially when more images are tested (70 and 80%)—the slope increases then levels off from the left to the right. The overall accuracy is nearly 98% for all graphs at the end of the x-axis, indicating that the method is highly accurate with as little as 500 labeled controls. It is also robust considering that the trend of prediction curve remains the same for different testing percentages.

Moreover, we can see that FABS-NC′ has an advantage over other ranking methods for this particular mitochondria dataset. Its curve is often above all other methods, except for 10% labeled controls; testing percentage 70%: 70% labeled controls; and testing percentage 80%: 10% and interval 40–50%. Notice that for the low number of testing (30%), FABS-NC′ outperforms all other methods—when using all labeled controls for ranking, it is over half more accurate than the next best algorithm.

Overall, FABS-SVM also performs well, although sometimes trailing behind FABS-NC′ by a large margin. PCA ranking performs poorly when testing images are few (30%). Center ranking is generally of low quality, giving small accuracy for

Table 1. Matrices of GDM between different pairs of drugs for different implementations of FABS-SVM and FABS-NC′

FABS-SVM	squamocin	Z-IETD	Z-LEHD
squamocin	0	∞	∞
Z-IETD	∞	0	3.43
Z-LEHD	∞	3.43	0
FABS-NC′			
squamocin	0	∞	∞
Z-IETD	∞	0	4.36
Z-LEHD	∞	4.36	0

all testing percentages. Notice, however, GTEM gives the best results when the number of labeled controls is very low (10%), indicating its usefulness when training data are few—neverthless, its advantage dimishes as more labeled training cases becomes available, producing inaccurate rankings comparing to FABS. The results show that applying the feature transformation step of GTEM as a pre-processing step of FABS-NC′ performs better than GTEM but not as well and as stable as FABS-NC′.

The experimental results suggest that, overall, FABS with NC′ implementation is the best ranking method among all for this particular mitochondria database. Remarkably, FABS-NC′ generalizes better than any other methods as more training and test examples become available.

3.3 Significance

Table 1 displays the significance score $-\log P$ between different pairs of drugs for FABS-NC′ and FABS-SVM implementations when we sub-sampled 30% of the labeled controls and 30% of drug treatment results. An infinity score (∞) is obtained when P is very close to zero, indicating that the distance between the two corresponding drugs is very large. The results show that FABS-NC′ is more discriminant then FAB-SVM because the significance scores for FABS-NC′ are larger than those for FABS-SVM.

We also performed a Monte Carlo simulation to test whether the observed difference of the FABS-NC′ scores of 30% of Z-IETD and Z-LEHD data using 80% of control data for training is significant against pairs of null data sets sampled from the same drug treatment populations. In 1000 random resamplings, no difference of the scores of the null data set pairs is higher than the observed score, yielding a close to zero P-value.

3.4 Comparison of running time

In this section, we compare the running times of three FABS-$\mathcal{A}$ procedures, where $\mathcal{A}$ here, as mentioned in previous sections, is one of bipartition algorithms including NC′ (Hochbaum, 2010b), SVM (Cristianini and Shawe-Taylor, 2000) and Spectral (Shi and Malik, 2000), among themselves and against PCA ranking, Center ranking and GTEM. The specification of the computer environment for this comparison is a Windows computer with 2.4GHz Intel(R) Core(TM)2 Duo CPU 2.40 GHz and 2 GB memory.

Figures 5 and 6 display running times of various methods, excluding the times for subsampling—which have a median of 0.01 second, maximum of 0.02 second and minimum of 0.006 second—for different testing percentages: x-axis increases with the number

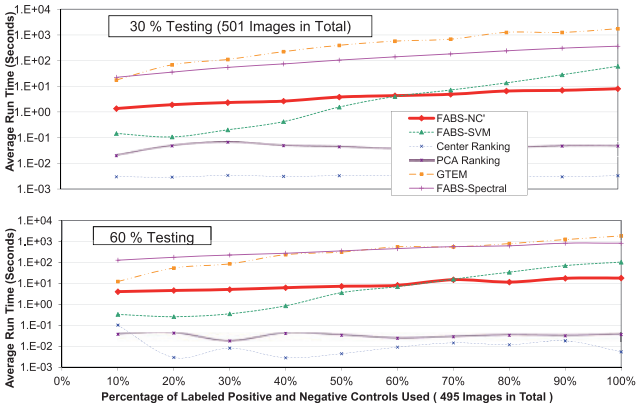

Fig. 5. The running time comparison among different methods. The testing percentages used are: 30 and 60%

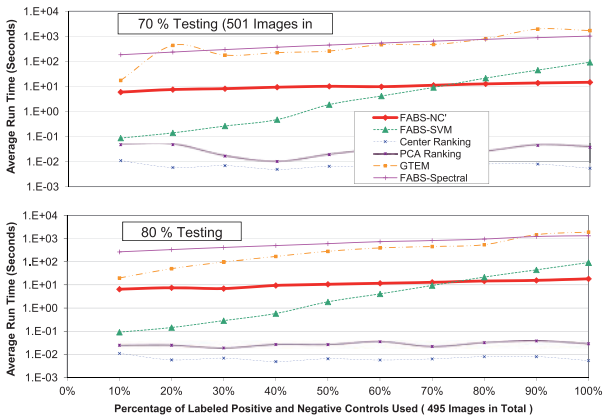

Fig. 6. (Continued) the testing percentages used are: 70 and 80%

of positive controls and negative controls used, representing more and more training data becoming available, while y-axis is the running time. The six curves in the figures are the different methods including various implementations of FABS-$\mathcal{A}$—notice that FABS-NC$'$ is represented by the thickest curve. There are 501 testing data: 265 Z-IETD and 291 Z-LEHD.

From the figures, among FABS-$\mathcal{A}$, we can observe that for all testing percentages considered, FABS-Spectral takes the most running time, lagging behind both FABS-NC$'$ and FABS-SVM by large margins. For FABS-NC$'$, the running time steadily lengthens as the number of positive and negative controls increases, however, not as dramatic as FABS-SVM, whose running time, shorter than these of other procedures initially, grows exponentially—in one case (testing percentage 70%), running 100% of positive and negative controls requires around 1000 times more seconds than running 10% of positive and negative controls. This is to compare with FABS-NC$'$: for the same testing percentage, using all positve and negative controls only requires twice as much running time than that of using only 10%—10% corresponds to around 50 controls in total, a relative small number of images that can be obtained through HCS. This observation, combined with the results from Section 3.2, indicates that even though FABS-SVM has the initial advantage for running time, this is off-set by the initial more accurate results produced by FABS-NC$'$. Moreover, it appears that FABS-NC$'$ scales much better with increasing input data than FABS-SVM. Looking at the other methods besides FABS-$\mathcal{A}$, we can observe that GTEM takes relatively long time on the par with FABS-Spectral—this is in contrast with PCA ranking and center ranking whose running times are the lowest among all methods: this result is expected, since FABS-$\mathcal{A}$ use PCA for pre-processing (i.e. doing PCA is already added as a part of computational costs), therefore FABS-$\mathcal{A}$ can only take longer time than PCA ranking. However, from Section 3.2, it is clear that this extra computational costs bring significant improvements in accuracy, which combined with scalability of FABS-NC$'$, makes FABS-NC$'$, overall, an attractive candidate for ranking this database.

4 DISCUSSION

In this article, we describe a new drug ranking framework called FABS. It is graph based, producing a single scalar score for each drug for ranking. The formulation and solution sidesteps many pitfalls of other traditional methods. The article also reports FABS-NC$'$ semi-supervised implementation and its comparative study. Not only is this implementation better than four other considered methods, it also outperforms FABS-SVM and FABS-spectral implementations on a mitochondria databases. This preliminary result suggests that FABS-NC$'$ is good for ranking toxicity of drugs targeting mitochondria for a specific database.

There are some advantages of our measure. First, FABS is one-dimensional, that is, a single scalar, giving an unambiguous way to rank drugs. Its computation considers all samples of each drug and uses a fraction as the final score. This diminishes the effect of outliers and noise, because, if the number of images is large for each drug, as in the case of HCS, outliers, which are few in number, can not influence the result—a fraction, in a significant way. This similarly is the reason for noise reduction. More importantly, our measure FABS-NC$'$ is acquired through a combinatorial algorithm, which is efficient. This is essential since the number of cells observed in a HCS is large and the applicability of any metric learning algorithm is greatly reduced if it cannot process them sufficiently fast. The last noteworthy advantage of our framework is that the training data for the semi-supervised formulation are the positive and negative controls, which are easily recognizable and obtained without time-consuming annotation, sidestepping the limitation of training sample size of many metric learning algorithms.

Our future work includes to investigate whether the introduction of node weights, in our construction of the graph in `Step 2` of Algorithm 1 will benefit the prediction results. This is especially relevant because of a recent development for solving generalized version of NC$'$ utilizing node weights (Hochbaum, 2010c). Moreover, we could also expand our FABS application into other criteria and situations for determining the ranking of the drugs and test on more databases to see the effectiveness of our method as they become available.

ACKNOWLEDGMENTS

We wish to thank Professor Chung-Chi Lin and his team at the National Yang-Ming University, Taiwan for providing us the image database used in our experiments.

Funding: National Science Foundation awards (No. DMI-0620677, CMMI-1200592 and CBET-0736232 to D.8.H. partial). The National Heart, Lung, and Blood Institute award (1UH2HL108780-01 to C.N.H. partial)

REFERENCES

Altschuler,S.J. and Wu,L.F. (2010) Cellular heterogeneity: Do differences make a difference? *Cell*, **141**, 559–563.

Burges,C. (1998) A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Disc.*, **2**, 121–167.

Chandran,B. and Hochbaum,D.S. (2009) A computational study of the pseudoflow and push-relabel algorithms for the maximum flow problem. *Operations Res.*, **57**, 358–376.

Chang,C.-C. and Lin,C.-J. (2011) Libsvm: a library for support vector machines. *ACM Trans. Intell. Sys. Technol.*, **2**, 27:1–27:27.

Conrad,C. and Gerlich,D. (2010) Automated microscopy for high-content rnai screening. *J. Cell Biol.*, **188**, 453–461.

Cristianini,N. and Shawe-Taylor,J. (2000) *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Cambridge, United Kingdom.

Denner,P. *et al.* (2008) High-content analysis in preclinical drug discovery. *Comb. Chem. High Throughput Screen*, **11**, 216–230.

Feng,Y. *et al.* (2009) Multi-parameter phenotypic profiling: using cellular effects to characterize small-molecule compounds. *Nat. Rev. Drug Discov.*, **8**, 567–578.

Hochbaum,D.S. (2001) An efficient algorithm for image segmentation, markov random fields and related problems. *J. ACM*, **48**, 686–701.

Hochbaum,D.S. (2008) The pseudoflow algorithm: A new algorithm for the maximum flow problem. *Operations Res.*, **58**, 992–1009.

Hochbaum,D.S. (2010a) *HPF: Hochbaum's Pseudo-Flow Algorithm Implementation: http://riot.ieor.berkeley.edu/riot / Applications / Pseudoflow / maxflow.html*, Last updated on July 26, 2010.

Hochbaum,D.S. (2010b) Polynomial time algorithms for ratio regions and a variant of normalized cut. *IEEE Trans. Pattern Anal. Mach. Intell.*, **32**, 889–898.

Hochbaum,D.S. (2010c) Replacing spectral techniques for expander ratio and normalized cut by combinatorial flow algorithms. arXiv:1010.4535v1 [math.OC], ArXiv e-prints, 2010.

Huang,K. and Murphy,R.F. (2004) Boosting accuracy of automated classification of fluorescence microscope images for location proteomics. *BMC Bioinformatics*, **5**, 78.

Jarque,C.M. and Bera,A.K. (1987) A test for normality of observations and regression residuals. *Int. Stat. Rev.*, **55**, 163–172.

Jordan,M. (ed.) (1996) *Learning in Graphical Models*. North Atlantic Treaty Organization, Scientific Affairs Division.

Lang,P. *et al.* (2006) Cellular imaging in drug discovery. *Nat. Rev. Drug Discov.*, **5**, 343–356.

Lin,C.-C. *et al.* (2007) Boosting multiclass learning with repeating codes and weak detectors for protein subcellular localization. *Bioinformatics*, **23**, 3374–3381.

Lin,Y.-S. *et al.* (2010) A spectral graph theoretic approach to quantification and calibration of collective morphological differences in cell images. *Bioinformatics*, **26**, i29–i37.

Mitchison,T.J. (2005) Small-molecule screening and profiling by using automated microscopy. *Chembiochem*, **6**, 33–39.

Morelock,M.M. *et al.* (2005) Statistics of assay validation in high throughput cell imaging of nuclear factor κb nuclear translocation. *Assay Drug Dev. Technol.*, **3**, 483–499.

Nichols,A. (2007) High content screening as a screening tool in drug discovery. *Methods Mol. Biol.*, **356**, 379–387.

Paull,K.D. *et al.* (1992) Identification of novel antimitotic agents acting at the tubulin level by computer-assisted evaluation of differential cytotoxicity data. *Cancer Res.*, **52**, 3892.

Peng,J.-Y. *et al.* (2011) Automatic morphological subtyping reveals new roles of caspases in mitochondrial dynamics. *PLoS Comput. Biol.*, **7**, e1002212.

Perlman,Z.M. *et al.* (2004) Multidimensional drug profiling by automated microscopy. *Science*, **306**, 1194–1198.

Shi,J. and Malik,J. (2000) Normalized cut and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, **22**, 888–905.

Taguchi,N. *et al.* (2007) Mitotic phosphorylation of dynamin-related gtpase drp1 participates in mitochondrial fission. *Biol. Chem.*, **282**, 11521–11529.

Taylor,D.L. and Hsaskins,J.R. (2007) *High Contnet Screening: A Powerful Approach to Systems Cell Biology and Drug Discovery.* Totowa, NY: Humana.

Timothee,C. *et al.* (2004) *Normalized Cuts Segmentation Code, for MATLAB*, http://www.cis.upenn.edu/∼jshi/software/.

Washio,T. and Motoda,H. (2003) State of the art of graph-based data mining. *ACM SIGKDD Explorations Newsletter*, **5**, 59–68.

Yarrow,J.C. *et al.* (2003) Phenotypic screening of small molecule libraries by high throughput cell imaging. *Comb. Chem. High Throughput Screen*, **6**, 279–286.

Zhang,J. *et al.* (1999) A simple statistical parameter for use in evaluation and validation of high throughput screening assays. *J. Biomol. Screen.*, **4**, 67–73.

Zhou,X. and Wong,S. (2006) Informatics challenges of high-throughput microscopy. *IEEE Signal Proc. Mag.*, **23**, 63–72.