RESEARCH

A theorem proving based approach for automatically synthesizing visualizations of flow cytometry data

Sunny Raj[1*], Faraz Hussain[2], Zubir Husein[1], Neslisah Torosdagli[1], Damla Turgut[1], Sumanta Pattanaik[1] and Sumit Kumar Jha[1]

Abstract

Background: Polychromatic flow cytometry is a widely used technique for gathering and analyzing cellular data. The data generated is high-dimensional, and therefore notoriously difficult to visualize by a human expert. The traditional method of plotting every pair of observables of the original high-dimensional data set leads to a combinatorial explosion in the number of visualizations. A natural solution is to *project the data into a lower-dimensional space while (approximately) preserving key properties and relationships among data points with minimal distortion.* The expert can then easily visualize this low-dimensional embedding of the original dataset.

Results: This paper describes our new approach for visualizing high-dimensional flow cytometry datasets that uses a decision procedure to *automatically synthesize two-dimensional and three-dimensional projections of the (original) high-dimensional data.* We compare our visualization approach to the popular multi-dimensional scaling (MDS) algorithm on a representative set of benchmarks, and shows using a set of randomly selected data points, that our technique produces distortions that are 1.44 to 4.15 times smaller than those of the MDS algorithm.

Conclusions: We describe a new algorithmic technique that uses a symbolic decision procedure to automatically synthesize low-dimensional projections of high-dimensional flow cytometry data. Our algorithm is the first application of decision procedures for automatically generating highly-accurate (low-dimensional) visualizations of large, high-dimensional data sets.

Keywords: symbolic decision procedures; high-fidelity visualization; biomedical informatics; high-dimensional big data; flow cytometry; automated synthesis

*Correspondence: sraj@cs.ucf.edu

[1]Department of Computer Science, University of Central Florida, Orlando, Florida, United States of America

Full list of author information is available at the end of the article

1 Introduction

Polychromatic flow cytometry provides a revolutionary tool for analyzing biological samples by identifying

multiple phenotypic properties of individual cells, including DNA content, RNA, intracellular phosphoproteins, cytokines, and cell-surface proteins [1]. Besides its applications in translational research, flow cytometry is routinely used by researchers to gain a deeper understanding into the fundamental biology of cellular processes [2]. Unlike traditional techniques that can only compute statistical measurements of large populations of cells such as the average concentration of a protein in a cell sample, flow cytometry is capable of measuring a number of phenotypic properties of *each cell* in a sample [3].

The data generated from such detailed measurements creates new opportunities for the experimental scientist. The expert can now identify even small groups of cells that are different from others (representing an experimental success or a clinical abnormality) whose presence could not have otherwise been detected by simply observing the average phenotypic properties of cells. The scientist can also finely characterize temporal changes in the multi-dimensional distribution of multiple phenotypes of cell populations in response to experimental evolutionary pressure or a treatment regimen.

There remain two long-standing barriers to universal adoption of flow cytometry for disease diagnoses:

- Cognitive processing studies have shown that the data analysis capacity of human beings is limited, on average, to about four dimensions that can be processed in parallel [4, 5]. Therefore, polychromatic flow cytometry datasets that often produce data in 10 or more dimensions cannot be easily visualized. We note that generating a series of two-dimensional projections of high-dimensional biomedical data is usually unhelpful because it leads to loss of information about the multi-dimensional relationships among data points – hence, defeating the very purpose of collecting multi-parameter cellular data.
- Flow cytometry produces large datasets, typically with millions of data points per sample, which is well beyond our cognitive memory limits [6]. Hence, statistical summarization of this data that causes the loss of small – but biologically significant – details has been considered a necessary evil. Not surprisingly, this often leads to an inability to detect rare events and can potentially cause significant harm to the subject.

We have addressed these problems by designing a new automated technique for synthesizing low-dimensional visualizations of flow cytometry data. This paper makes the following main contributions:

- We describe SANJAY, a new algorithmic approach for automatically synthesizing 2D and 3D visualizations of high-dimensional flow cytometry data. Using our earlier work [7], SANJAY can describe the dataset using a complex network [8], and uses community detection algorithms to identify subpopulations of similar cells in the network [9]. SANJAY's main contribution is to employ automated algorithmic synthesis techniques [10, 11] and symbolic decision procedures [12] to create low-dimensional projections of high-dimensional big data that can be easily visualized.
- This algorithmic approach avoids statistical summarization and stochastic search and hence provides an *algorithmic method for complete and accurate visualization* of massive flow cytometry datasets in two and three dimensions with *minimal loss of information.*
- We compare our SANJAY to the popular multi-dimensional scaling (MDS) algorithm and show that our projections produce distortions that are on average 2.56 times smaller than those produced by MDS (see Table 1 on page 3).

Table 1: Distortions produced by the MDS approach and our method (SANJAY) when 10 randomly chosen high-dimensional data points from 30 flow cytometry datasets were projected onto two dimensions. The maximum distortion produced by SANJAY was, on average, 2.56 times less than that produced by MDS.

Dataset ID	Maximum distortion for MDS	Maximum distortion for SANJAY	Ratio of distortions (MDS/SANJAY)	Dataset ID	Maximum distortion for MDS	Maximum distortion for SANJAY	Ratio of distortions (MDS/SANJAY)
1	3197.845	1000	3.197	16	3150.466	1200	2.625
2	2711.12	1200	2.259	17	2497.225	1100	2.270
3	1953.082	1000	1.953	18	2925.544	1400	2.089
4	2917.223	1200	2.431	19	3813.344	1300	2.933
5	3483.532	1400	2.488	20	3700.842	1300	2.846
6	2925.941	1100	2.659	21	3011.87	1200	2.509
7	4233.021	1800	2.351	22	3252.494	1000	3.252
8	2898.038	1300	2.229	23	3381.443	1200	2.817
9	1876.719	1300	1.443	24	2963.938	1100	2.694
10	4314.192	1500	2.876	25	3428.368	1600	2.142
11	3543.691	1400	2.531	26	2712.258	1200	2.260
12	2449.823	1300	1.884	27	3679.701	1500	2.453
13	3835.263	1500	2.556	28	3286.024	1200	2.738
14	4153.369	1000	4.153	29	2449.747	1000	2.449
15	2858.641	1000	2.858	30	4160.04	1400	2.971

2 Background

2.1 Automated Gating of Flow Cytometry Data

Machine learning methods have been deployed for automatically labeling subpopulations of cells in flow cytometry data sets – a process popularly referred to as gating. In particular, supervised and semi-supervised machine learning algorithms [13, 14] have been extensively investigated for automatically identifying related cells.

Sequential gating [15] enables two-dimensional visualization of any two colors or dimensions of data from a polychromatic flow cytometer. The human expert then attempts to manually identify subsets of cells that correspond to the same subpopulation. While the process is computationally simple, the result is highly subjective and depends on the intuition of the oncologist. Further, an n-dimensional flow cytometry data has $n \times (n-1)/2$ possible two-dimensional visualizations. Thus, a 20-color polychromatic flow cytometer will produce 190 different 2-dimensional visualizations and it is a cognitive challenge for a human expert to verify clinical or experimental conjectures against all 190 visualizations obtained from a biological sample.

Probability binning [16] is an unsupervised quantitative methodology for analyzing polychromatic flow cytometry data that identifies the difference between the distribution of cells in a given sample and a standard control sample. Frequency difference gating [17] extends this approach by enabling multidimensional gating of the bins identified by the probability-binning algorithm that contain the largest differences between the given and the control sample.

Cluster analysis methods [18, 19] employ varying levels of expression of antigens to construct subsets of cells that share the same combination of fluorochromes markers. While the technique is unsupervised, the result is only a semi-quantitative two-dimensional visual description (such as a heat map) of the data set and still needs to be interpreted subjectively by an expert for biological correctness. Standard machine learning

algorithms such as k-means [20] and expectation maximization [21] have been applied to perform cluster analyses of polychromatic flow cytometry data.

The most popular clustering algorithm that operates by building and refining partitions is the k-means algorithm [22, 23]. The popular k-means algorithms have also been applied to flow cytometry data [21]. The k-means algorithm requires three inputs from the user: the number of clusters, an initial cluster assignment, and a metric to measure distance between data points. As the k-means algorithms converge only to one of the local minima, different initializations of the k-means algorithm can lead to different final clustering of the data. Such sensitivity to initial conditions is undesirable for an objective flow cytometry data exploration framework.

Principal Component Analysis is a particularly popular approach for generating two-dimensional visualizations of flow cytometry data [19]. However, low-dimensional visualizations lose a lot of information because of the low correlation between different fluorochromes, and such plots mostly serve as an exploratory tool in the hands of well-trained experts.

In our recent work [7], we have proposed the use of complex network models and their topological properties for discriminating between cancer and normal patients. In our approach, each node in the complex network corresponds to the measurements obtained from a single cell and an edge between two nodes exists if the Euclidean distance between them is smaller than a threshold. The evolution of the network through time can be derived by studying periodically acquired patient samples. By constructing such complex network models for multiple normal patients, we propose to develop a stochastic generative model that describes the flow cytometry data for normal patients. In particular, topological properties such as number of connected components, edge density, number of clusters, etc. are studied. The goal of our stochastic generative modeling is to capture the natural diversity that occurs in the normal patient population (age, race, gender, BMI), and thereby compute the probability that a given flow cytometry sample does not arise from this stochastic generative model. Rare behavior identification algorithms, including our own work [24], can then be employed to compute the probability that a given flow cytometry sample indicates the presence of a physiological anomaly in a patient.

2.2 Decision Procedures

This paper is the first effort that we know of towards the application of symbolic decision procedures for the algorithmic synthesis of projections from high-dimensional data to low-dimensional visualizations. We show that our SANJAY approach based on bit-vector decision procedures outperforms classical multi-dimensional scaling approach by consistently creating projections with at least 80% less distortion.

More recently, a number of decision procedures for verifying various decidable fragments of logic involving arithmetic and function symbols have been proposed and implemented using the popular SMTLIB standard [25]. In particular, a number of decision procedures for bit-vectors involving arithmetic and logical operations have been successfully implemented [26, 27]. Many of these approaches build upon the foundation work of Martin Davis, Hilary Putnam, George Logemann and Donald W. Loveland who introduced the DPLL algorithm for checking the satisfiability of propositional logic formulas in 1962 [28].

2.3 Some notations and definitions

We now describe some basic ideas relevant to our use of decision procedures for the automated synthesis of visualizations.

Definition 1 (Basic bitvector operations) A bitvector is a vector of Boolean values of a given length.

Given two bit-vectors, their bitwise logical operations are performed by applying the logical operation to the corresponding bits of the bit-vectors.

$$\neg x = \forall i \in \{0, 1, \ldots, l-1\} \neg x_i$$
$$x \vee y = \forall i \in \{0, 1, \ldots, l-1\} (x_i \vee y_i)$$
$$x \wedge y = \forall i \in \{0, 1, \ldots, l-1\} (x_i \wedge y_i)$$

The above equations define the formal semantics of bit-vector NOT, OR, and AND operations. Similarly, arithmetic operations such as addition and subtraction can be defined on bit-vectors by extending the standard definition of these operations from the decimal to the binary representation.

Definition 2 (Bitvector concatenation) Two bit-vectors of length l and l' can be concatenated into a single bit-vector of length $l + l'$.

$$xy = \forall i \in \{0, 1, \ldots, l + l' - 1\} b_i \text{ where,}$$
$$b_i = \begin{cases} x_i & \text{if } i < l \\ y_{i-l} & \text{otherwise.} \end{cases}$$

Relational operations on bitvector are defined similarly, using both signed and unsigned interpretations [25]. As these formulas naturally arise in software and hardware verification, several solvers for bit-vector decision procedures are widely deployed. The top solvers in the 2015 SMT-COMP competition for bit-vectors include Boolector, CVC4, STP, Yices, Mathsat and Z3. Most of these solvers use a combination of bit-blasting and rewriting to translate the bitvector decision problem into a combination of lemmas that can be discharged using results from number theory and satisfiability solving [29].

Definition 3 (Distortion) Distortion is defined as the change of distance between two points when they are projected from a high dimension space to a lower dimension space. Let the distance between points x and y in the original space be $d(x, y)$. Let the projections of x and y in the lower dimension space be x' and y' respectively. Let $d(x', y')$ be the distance between the projected points. The distortion due to this projection is defined by:

$$distortion(x, y) = |d(x', y') - d(x, y)|$$

3 Graphical Representation of Flow Cytometry Data

There is an inherent complex network structure in polychromatic flow cytometry data arising from the well-governed biological process of cell differentiation. Using our earlier approach [7], we build a complex network representation of the observed flow cytometry data set.

Definition 4 (Flow Cytometry Network) Given N m-dimensional data points representing N cells, each representing m observed properties measured by a polychromatic flow cytometer, the flow cytometry network with threshold T (a $T - FCN$) is a graph $G = (V, E)$ where V is the set of nodes and E is the set of edges, such that:

- a node $v \in V$ denotes the m quantities measured for a single cell, i.e. $v = (v_0, v_1, \ldots, v_{m-1})$, and
- $(v, v') \in E$ if and only if $||(v_0, \ldots, v_{m-1}) - (v'_0, \ldots, v'_{m-1})|| \leq T$.

The second property above specifies that there's an edge between two nodes (i.e. between data points representing a pair of cells), when the Manhattan distance between them is less than threshold T. Note that the Manhattan distance between vectors $v = (v_0, \ldots, v_{m-1})$ and $u = (u_0, \ldots, u_{m-1})$ is defined to be $\Sigma_{i=0}^{m-1} |v_i - u_i|$.

Given flow cytometry data, a T-FCN (flow cytometry network) is determined by the threshold T that is

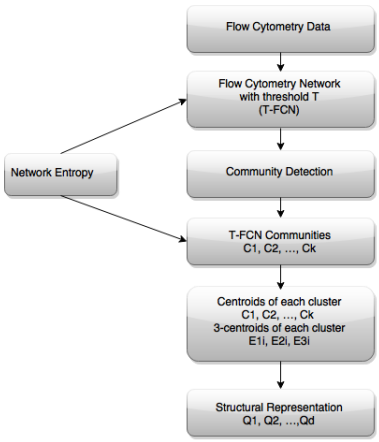

Figure 1: From flow cytometry data to complex networks and its structural representation.

used to decide whether two nodes in the flow cytometry network are connected by an edge in the T-FCN. The threshold T is typically learned from experimental data. As T is varied from ∞ to 0, the T-FCN goes from being a clique of N nodes to being a network with N components – each node being a component by itself. The variation in T causes changes in the distribution of the topological properties.

Using information theoretic arguments [30, 31], we can compute the value of T that maximizes the information content or entropy of the distribution of the topological properties. Thus, the generated T-FCN is the most informative network describing the flow cytometry data set.

3.1 Community Detection in Flow Cytometry Data

Several existing algorithms are capable of identifying communities in large complex networks [32]. Due to the massive size of the network generated by a typical flow cytometry dataset, one can readily rule out the use of matrix and spectral graph theory based methods. Modularity based methods are known to be biased against small communities and are hence not a method of choice for identifying communities in flow cytometry networks, where small communities may represent rare but interesting anomalies [33].

Keeping in mind the our high-assurance requirement for biomedical applications, and the large size of flow cytometry datasets, we suggest the use of a parallel version of the Walktrap algorithm for community detection [7] in our flow cytometry networks [9].

The main idea behind Walktrap approach is based on the intuition that random walks of a graph must be trapped in densely connected communities of the T-FCN that are only sparsely connected to the rest of the network. As several random walks can be instantiated in parallel on multiple processing nodes, the approach is readily deployable on large supercomputing clusters [34].

3.2 Structural Representation of Flow Cytometry Networks

Each flow cytometry data set is represented by a T-FCN that maximizes the information content of the network. A flow cytometry network T-FCN is then decomposed into a number of communities $C_1, \ldots, C_n$, using methods described in Section 3.1 where each C_i is itself a T-FCN. The centroid of a community gives the approximate position of all the points in the community. To preserve the relative position of the communities, we compute the centroids $O_1, \ldots, O_n$ of the communities and seek to approximately preserve the distance between these centroids. In order to preserve the geometry of the individual communities, we also must compute the 3-centroids E_i^1, E_i^2, E_i^3 for each community C_i when projecting into two dimensions (and 4-centroids when projecting into three dimensions). To calculate $3-$centroids of a community C_i we break the community into 3 component communities C_i^1, C_i^2, C_i^3 using k-means clustering algorithm where the input k for the k-means algorithm is equal to 3. We then calculate one centroid for each of the 3 component communities for a total of 3 com-

ponent centroids E_i^1, E_i^2, E_i^3 for each community C_i. For projecting onto two dimensions the set of points $\{O_1, E_1^1, E_1^2, E_1^3, O_2, E_2^1, E_2^2, E_2^3, \ldots, O_n, E_n^1, E_n^2, E_n^3\}$, that we will also denote by $Q_1, \ldots, Q_d$ where $d = 4n$, and n is the number of communities in the T-FCN, serves as a structural representation of the flow cytometry network.

4 Automated synthesis of projections using decision procedures

Given the structure-defining points $\{Q_1, \ldots, Q_d\} = \{O_1, E_1^1, E_1^2, E_1^3, O_2, E_2^1, E_2^2, E_2^3, \ldots, O_n, E_n^1, E_n^2, E_n^3\}$ in m dimensions, SANJAY synthesizes an embedding $\{R_1, \ldots, R_d\}$ of the points in two-dimensional or any other lower dimensional space that approximately preserves the pairwise manhattan distances between these points up to an error of $\epsilon > 0$. The following expression specifies relationship between the original points $Q_1, \ldots, Q_d$ and the synthesized lower-dimensional projection $R_1, \ldots, R_d$ with respect to the distortion ϵ:

$$\exists R_1, R_2 \ldots, R_d, \forall i, j \in \{1, \ldots d\}, \text{where } i \neq j :$$
$$\bigwedge ||R_i - R_j|| \leq ||Q_i - Q_j|| + \epsilon,$$
$$\bigwedge ||R_i - R_j|| \geq ||Q_i - Q_j|| - \epsilon$$

To help in discussing our projection algorithm, we now state, without proof, a lemma that describes the requirement for the location of a point in 2D or 3D space to be fixed.

Lemma 1 (Fixing points in two and three dimensions) For any given point in two-dimensional space, its distance from three unique points uniquely identify its coordinates. Similarly, for any point in three-dimensional space, its distance from four unique points uniquely identify its coordinates [35].

Therefore, the two-dimensional projection of all points in a community C_i can be obtained using the 2D projections of the 3-centroids E_i^1, E_i^2, E_i^3 of that community. Similarly, the three-dimensional projections of the points in a community can be obtained from the projections of the 4-centroids $E_i^1, E_i^2, E_i^3, E_i^4$ of the community.

However, a direct translation of the problem to bit-vector decision procedures involves a tradeoff between computational tractability and the accuracy of the obtained projections. Large values of ϵ lead to decision problems that can be readily solved by decision procedures but correspond to poor projections. Small ϵ values represent high-quality distance-preserving projections but create computationally challenging instances of the decision problem.

The SANJAY algorithm solves the problem by using an *iterative refinement* to derive the points $R_1, R_2, \ldots, R_d$ in the lower-dimensional space from the pairwise distances between the points $Q_1, \ldots, Q_d$ in the higher dimension. The algorithm starts by synthesizing the highest-order bit in the bit-vector representation of these points, and then searches for the other bits.

The SANJAY algorithm is illustrated in Algorithm 1 on page 8. The algorithm accepts the pairwise distances $D_{i,j} (1 \leq i, j, \leq d)$ between every pair of d points as an input. It also accepts two other inputs: the length L_{MAX} of the bit-vector representing the projected points to be synthesized and the number of bits L_{NUM} that should be learned in every iteration of the projection synthesis loop.

In Algorithm 1, a point Q_i is represented by the bit vector representation $({}_{P_{x_i}^{L_{CURR}}}{}_{a^{L_{REM}}}, {}_{P_{y_i}^{L_{CURR}}}{}_{b^{L_{REM}}})$ where ${}_{P_{x_i}^{L_{CURR}}}{}_{a^{L_{REM}}}$ is the x-coordinate and ${}_{P_{y_i}^{L_{CURR}}}{}_{b^{L_{REM}}}$ is the y-coordinate. The $P_{x_i}^{L_{CURR}}$ and $P_{y_i}^{L_{CURR}}$ is the part of the vector that has been calculated by the algorithm, the $a^{L_{REM}}$ and $b^{L_{REM}}$ is the part of the vector

Algorithm 1 The SANJAY algorithm for automated synthesis of two dimensional visualizations for flow cytometry data.

Require:

Pairwise distances $D_{i,j}, 1 \leq i,j \leq d, i \neq j$ between every pair of d points $\{Q_1, \ldots Q_d\}$ to be projected in the higher-dimensional space

Maximum distortion ϵ

The maximum length L_{MAX} of the bitvectors used to store points

The number of bits L_{NUM} to be learned in each iteration of the refinement process

Ensure:

Synthesized points $\{R_1, \ldots, R_d\}$ in the lower dimension

1: $L_{CURR} \leftarrow 0$ {Current no. of bits in synth. points}
2: $L_{REM} \leftarrow L_{MAX}$ {Remaining bits to be synthesized}
3: For all i, $P_{x_i}^0 \leftarrow \phi$
4: For all i, $P_{y_i}^0 \leftarrow \phi$
5: **repeat**
6: For all i, compute $A_{x_i}^{L_{NUM}}$ and $A_{y_i}^{L_{NUM}}$ such that $(1-\epsilon)D_{i,j}^2 \leq \max_{a,b,c,d \in \{0,1\}} ||(P_{x_i}^{L_{CURR}} A_{x_i}^{L_{NUM}} a^{L_{REM}}, P_{y_i}^{L_{CURR}} A_{y_i}^{L_{NUM}} b^{L_{REM}}) - (P_{x_j}^{L_{CURR}} A_{x_j}^{L_{NUM}} c^{L_{REM}}, P_{y_j}^{L_{CURR}} A_{y_j}^{L_{NUM}} d^{L_{REM}})||^2 \leq (1+\epsilon)D_{i,j}^2$
7: For all i, $P_{x_i}^{L_{CURR}+L_{NUM}} \leftarrow P_{x_i}^{L_{CURR}}.A_{x_i}^{L_{NUM}}$
8: For all i, $P_{y_i}^{L_{CURR}+L_{NUM}} \leftarrow P_{y_i}^{L_{CURR}}.A_{y_i}^{L_{NUM}}$
9: $L_{CURR} \leftarrow L_{CURR} + L_{NUM}$
10: $L_{REM} \leftarrow L_{REM} - L_{NUM}$
11: **until** $L_{REM} = 0$
12: For all i, $R_i \leftarrow (P_{x_i}^{L_{MAX}}, P_{y_i}^{L_{MAX}})$
13: **return** $\{R_1, \ldots R_d\}$

that has still not been calculated. When all the bits of any vector $a^{L_{REM}}$ is 1 then we denote it by $1^{L_{REM}}$ similarly when all the bits of the vector is 0 we denote it by $0^{L_{REM}}$. The bit vector $a^{L_{REM}}$ has the property that $0^{L_{REM}} \leq a^{L_{REM}} \leq 1^{L_{REM}}$. So, any point Q_i with representation $(P_{x_i}^{L_{CURR}} a^{L_{REM}}, P_{y_i}^{L_{CURR}} b^{L_{REM}})$ can take all the values within the square with corners $(P_{x_i}^{L_{CURR}} 0^{L_{REM}}, P_{y_i}^{L_{CURR}} 0^{L_{REM}}), (P_{x_i}^{L_{CURR}} 0^{L_{REM}}, P_{y_i}^{L_{CURR}} 1^{L_{REM}}), (P_{x_i}^{L_{CURR}} 1^{L_{REM}}, P_{y_i}^{L_{CURR}} 0^{L_{REM}}), (P_{x_i}^{L_{CURR}} 1^{L_{REM}}, P_{y_i}^{L_{CURR}} 1^{L_{REM}})$.

Algorithm 1 initializes the length L_{CURR} of the projected points to 0. The algorithm also initializes the length L_{REM} of the remaining bit-vectors to be synthesized with the value L_{MAX}. This means that the point P_i can take all the values within the square denoted by the points $(1^{L_{MAX}}, 1^{L_{MAX}}), (1^{L_{MAX}}, 0^{L_{MAX}}), (0^{L_{MAX}}, 1^{L_{MAX}}), (0^{L_{MAX}}, 0^{L_{MAX}})$. This square spans the whole search space, which implies that at the start of the first iteration, the point P_i can be found anywhere in this search space.

A bit-vector decision procedure then searches for a better approximation of the projected point by searching for the next L_{NUM} **higher order bits** $A_1^1, A_2^1, \ldots, A_{L_{NUM}}^1$ in the binary representation of the projection of the points by solving the following decision problem:

$$B_i = ||(P_{x_i}^{L_{CURR}} A_{x_i}^{L_{NUM}} a^{L_{REM}}, P_{y_i}^{L_{CURR}} A_{y_i}^{L_{NUM}} b^{L_{REM}}) - (P_{x_j}^{L_{CURR}} A_{x_j}^{L_{NUM}} c^{L_{REM}}, P_{y_j}^{L_{CURR}} A_{y_j}^{L_{NUM}} d^{L_{REM}})||^2 \quad (1)$$

$$(1-\epsilon)D_{i,j}^2 \leq \max_{a,b,c,d \in \{0,1\}} B_i \leq (1+\epsilon)D_{i,j}^2 \quad (2)$$

Each iteration of the algorithm breaks down the previous square into $2^{2L_{NUM}}$ sub-squares in which the point P_i can be found and Equation 2 using bit vector decision procedure selects the best possible sub-square for the point P_i. At the end of the iteration, each of the points is projected to a sub-square with the diagonal $(P_{x_i}^{L_{CURR}} A_{x_i}^{L_{NUM}} 0^{L_{REM}-L_{NUM}}, P_{y_i}^{L_{CURR}} A_{y_i}^{L_{NUM}} 0^{L_{REM}-L_{NUM}})$ and $(P_{x_i}^{L_{CURR}} A_{x_i}^{L_{NUM}} 1^{L_{REM}-L_{NUM}}, P_{y_i}^{L_{CURR}} A_{y_i}^{L_{NUM}} 1^{L_{REM}-L_{NUM}})$,where $P_{x_i}^{L_{CURR}}$ and $P_{y_i}^{L_{CURR}}$ denote bit vectors of L_{CURR} bits, $A_{x_i}^{L_{NUM}}$ and $A_{y_i}^{L_{NUM}}$ denote bit vectors of L_{NUM} bits, and $0^{L_{REM}-L_{NUM}}$ is a zero bit vector of $L_{REM} - L_{NUM}$ bits.

As the algorithm iterates, it builds finer abstractions of the bit-vector representation of the points being projected. When the algorithm has computed L_{MAX} number of bits in the bit-vector representation of the projected points, it assigns the generated bit-vectors to the output $R_1, \ldots, R_d$.

Table 2: Average distortions produced by the MDS approach and our method (SANJAY) when 10 randomly chosen high-dimensional data points from 30 flow cytometry datasets were projected onto two dimensions.

Dataset ID	Maximum distortion for MDS	Maximum distortion for SANJAY	Average distortion for MDS	Average distortions for SANJAY	Dataset ID	Maximum distortion for MDS	Maximum distortion for SANJAY	Average distortion for MDS	Average distortions for SANJAY
1	3197.845	1000	1042.49	540.81	16	3150.466	1200	1034.49	733.81
2	2711.12	1200	1024.41	653.32	17	2497.225	1100	919.57	623.09
3	1953.082	1000	649.25	537.57	18	2925.544	1400	1056.84	822.47
4	2917.223	1200	897.46	765.38	19	3813.344	1300	1117.45	757.51
5	3483.532	1400	1089.60	806.36	20	3700.842	1300	989.50	773.67
6	2925.941	1100	1069.45	634.07	21	3011.87	1200	1057.58	684.83
7	4233.021	1800	1374.40	1010.73	22	3252.494	1000	1412.67	605.78
8	2898.038	1300	949.88	709.49	23	3381.443	1200	915.02	712.82
9	1876.719	1300	765.93	752.59	24	2963.938	1100	824.36	741.10
10	4314.192	1500	1011.76	892.91	25	3428.368	1600	1178.13	1033.5
11	3543.691	1400	1050.42	882.83	26	2712.258	1200	949.24	713.37
12	2449.823	1300	1050.39	760.07	27	3679.701	1500	1114.25	833.66
13	3835.263	1500	1241.76	849.70	28	3286.024	1200	935.43	611.76
14	4153.369	1000	985.72	613.44	29	2449.747	1000	1004.84	561.34
15	2858.641	1000	1249.67	612.48	30	4160.04	1400	1178.41	874.19

Table 3: Distortions produced by our method (SANJAY) and Random Projections when 10 randomly chosen high-dimensional data points from 30 flow cytometry datasets were projected onto two dimensions.

Dataset ID	Maximum distortion for SANJAY	Maximum distortion for Random Projections	Average Distortion SANJAY	Average Distortion Random Projections	Dataset ID	Maximum distortion for SANJAY	Maximum distortion for Random Projections	Average Distortion SANJAY	Average Distortion Random Projections
1	1000	4069	540.81	1289.22	16	1200	6732	733.81	1791.50
2	1200	4179	653.32	1226.57	17	1100	4298	623.09	1361.34
3	1000	3982	537.57	1095.53	18	1400	4922	822.47	1480.31
4	1200	5289	765.38	1637.11	19	1300	6719	757.51	1912.72
5	1400	5045	806.36	1654.74	20	1300	5583	773.67	1806.02
6	1100	5092	634.07	1555.50	21	1200	5311	684.83	1535.20
7	1800	5364	1010.73	1608.83	22	1000	4447	605.78	1440.14
8	1300	3566	709.49	1111.89	23	1200	4731	712.82	1355.46
9	1300	4357	752.59	1439.54	24	1100	6251	741.10	1944.26
10	1500	4262	892.91	1376.79	25	1600	5919	1033.55	1943.44
11	1400	4945	882.83	1578.53	26	1200	5385	713.37	1762.98
12	1300	4370	760.07	1395.63	27	1500	4886	833.66	1519.03
13	1500	4747	849.70	1363.12	28	1200	5884	611.76	1648.05
14	1000	7029	613.44	2084.72	29	1000	5398	561.34	1513.42
15	1000	6161	612.48	1916.68	30	1400	3900	874.19	1047.50

5 Experimental Results

We performed our experimental evaluation on a 64-core 1.40GHz AMD Opteron(tm) Processor 6376 processor with 64 GB of RAM. We analyzed 30 flow cytometry data sets, each of them having 12 dimensions each. For each data set, we used MDS , random projections [36] and our SANJAY technique to search for two-dimensional projections of 10 randomly selected data points from the original (high-dimensional) data, while attempting to maintain the original inter-point distances. We then computed the maximum and the average error (*distortion*) of the projections produced by all three techniques. The comparison between SANJAY and MDS is presented in Table 1 on page 3 and Table 2 on page 9. The comparison between SANJAY and random projections is presented in Table 3 on page 9.

Our approach performed at least 1.44 times better and sometimes as much as 4.15 times better than MDS in terms of minimizing the maximum distance distortion among all the projected points. The average distortion produced by SANJAY algorithm were as much as 2.33 times better than those produced by MDS algorithm. When compared with random projections our approach performed 7.02 times better at minimizing maximum distortion between the points.

Figure 2 on page 12 shows the results of using SANJAY to project 1000 randomly chosen points from 6 of the 30 flow cytometry datasets discussed above. One can visually verify that in most of the figures there is only one large cluster – ostensibly representing cells exhibiting normal or expected behavior. We can attempt to use such automatically generated visualizations to identify patients whose flow cytometry data indicates a significant number of cells showing abnormal behavior.

6 Conclusion and Future Work

In this paper, we described a new algorithmic technique for automatically generating low dimensional visualizations of high-dimensional flow cytometry data. We used symbolic decision procedures to exhaustively search for low-dimensional projections in a finite, discretized search space where the user is allowed to define the size of search space. Our results show that visualizations synthsized using our technique (SANJAY) were better than those produced by the multi-dimensional scaling (MDS) algorithm and random projections in terms the maximum distortion in the pairwise distances.

The results themselves are not surprising as symbolic decision procedures are often used for solving optimization and search problems. However, their use in generating such high-fidelity visualizations has not been reported before. Building upon our current and earlier work [7], we are developing a web-enabled cyberinfrastructure that allows users to access our decision procedure based flow cytometry data analysis framework.

7 Acknowledgment

We acknowledge support from KnowledgeVis LLC, Leidos Biomedical Research Inc., the National Science Foundation Software and Hardware Foundations Project #1422257, NVIDIA Corporation, the Royal Bank of Canada, the Oak Ridge National Laboratory, and the National Science Foundation Exploiting Parallelism and Scalabilty (XPS) project #1438989.

Author details

[1]Department of Computer Science, University of Central Florida, Orlando, Florida, United States of America. [2]School of Computing, University of Utah, Salt Lake City, Utah, United States of America.

References

1. Janes, M.R., Rommel, C.: Next-generation flow cytometry. Nature biotechnology **29**(7), 602–604 (2011)
2. Shapiro, H.M.: Practical Flow Cytometry. John Wiley & Sons, Hoboken, New Jersey, USA. (2005)
3. Givan, A.L.: Flow Cytometry: First Principles. John Wiley & Sons, New York, USA. (2013)
4. Kyllonen, P.C., Christal, R.E.: Reasoning ability is (little more than) working-memory capacity?! Intelligence **14**(4), 389–433 (1990)

5. Doumas, L.A., Hummel, J.E., Sandhofer, C.M.: A theory of the discovery and predication of relational concepts. Psychological review **115**(1), 1 (2008)
6. Baddeley, A.: Working memory. Science **255**(5044), 556–559 (1992)
7. Petkova, A., Jha, S.K., Deo, N.: Discriminative Stochastic Models for Complex Networks Derived from Flow Cytometry Big Data. In: Forty-Fourth Southeastern International Conference on Combinatorics, Graph Theory, and Computing, Boca Raton (2013)
8. Wang, X.F.: Complex networks: topology, dynamics and synchronization. International Journal of Bifurcation and Chaos **12**(05), 885–916 (2002)
9. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences **105**(4), 1118–1123 (2008)
10. Jha, S., Seshia, S.A.: A theory of formal synthesis via inductive learning. arXiv preprint arXiv:1505.03953 (2015)
11. Jha, S.K.: Towards automated system synthesis using sciduction. PhD thesis, University of California, Berkeley (2011)
12. Jha, S., Limaye, R., Seshia, S.A.: Beaver: Engineering an efficient smt solver for bit-vector arithmetic. In: Computer Aided Verification, pp. 668–674 (2009). Springer
13. Ramanna, S., Jain, L.C., Howlett, R.J.: Emerging Paradigms in Machine Learning. Springer, Germany. (2013)
14. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Germany. (2006)
15. Sutherland, D.R., Anderson, L., Keeney, M., Nayar, R., Chin-Yee, I.: The ishage guidelines for cd34+ cell determination by flow cytometry. Journal of hematotherapy **5**(3), 213–226 (1996)
16. De Rosa, S.C., Brenchley, J.M., Roederer, M.: Beyond six colors: a new era in flow cytometry. Nature medicine **9**(1), 112–117 (2003)
17. Roederer, M., Hardy, R.R.: Frequency difference gating: a multivariate method for identifying subsets that differ between samples. Cytometry **45**(1), 56–64 (2001)
18. Perfetto, S.P., Chattopadhyay, P.K., Roederer, M.: Seventeen-colour flow cytometry: unravelling the immune system. Nature Reviews Immunology **4**(8), 648–655 (2004)
19. Lugli, E., Pinti, M., Nasi, M., Troiano, L., Ferraresi, R., Mussi, C., Salvioli, G., Patsekin, V., Robinson, J.P., Durante, C., *et al.*: Subject classification obtained by cluster analysis and principal component analysis applied to flow cytometric data. Cytometry Part A **71**(5), 334–344 (2007)
20. Zeng, Q.T., Pratt, J.P., Pak, J., Ravnic, D., Huss, H., Mentzer, S.J.: Feature-guided clustering of multi-dimensional flow cytometry datasets. Journal of Biomedical Informatics **40**(3), 325–331 (2007)
21. Lo, K., Brinkman, R.R., Gottardo, R.: Automated gating of flow cytometry data via robust model-based clustering. Cytometry Part A **73**(4), 321–332 (2008)
22. MacQueen, J., *et al.*: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297 (1967). Oakland, CA, USA.
23. Lloyd, S.P.: Least squares quantization in pcm. Information Theory, IEEE Transactions on **28**(2), 129–137 (1982)
24. Ghosh, A.K., Hussain, F., Jha, S.K., Langmead, C.J., Jha, S.: Decision Procedure Based Discovery of Rare Behaviors in Stochastic Differential Equation Models of Biological Systems. In: Proceedings of the 2nd IEEE International Conference on Computational Advances in Bio and Medical Sciences (ICCABS 2012), pp. 1–6. IEEE Computer Society, Las Vegas, NV (2012)
25. Ranise, S., Tinelli, C.: The smt-lib standard: Version 1.2. Technical report, Technical report, Department of Computer Science, The University of Iowa, 2006. Available at www. SMT-LIB. org (2006)
26. De Moura, L., Bjørner, N.: Z3: An efficient smt solver. In: Tools and Algorithms for the Construction and Analysis of Systems, pp. 337–340. Springer, Germany. (2008)
27. Brummayer, R., Biere, A.: Boolector: An efficient smt solver for bit-vectors and arrays. In: Tools and Algorithms for the Construction and Analysis of Systems, pp. 174–177. Springer, Germany. (2009)
28. Davis, M., Logemann, G., Loveland, D.: A machine program for theorem-proving. Communications of the ACM **5**(7), 394–397 (1962)
29. De Moura, L., Bjørner, N.: Satisfiability modulo theories: introduction and applications. Communications of the ACM **54**(9), 69–77 (2011)
30. El Gamal, A., Kim, Y.-H.: Network Information Theory. Cambridge university press, United Kingdom. (2011)
31. Rosvall, M., Bergstrom, C.T.: An information-theoretic framework for resolving community structure in complex networks. Proceedings of the National Academy of Sciences **104**(18), 7327–7331 (2007)
32. Lancichinetti, A., Fortunato, S.: Community detection algorithms: a comparative analysis. Physical review E **80**(5), 056117 (2009)
33. Newman, M.E.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences **103**(23), 8577–8582 (2006)
34. Pons, P., Latapy, M.: Computing communities in large networks using random walks. In: Computer and Information Sciences-ISCIS 2005, pp. 284–293. Springer, Germany. (2005)
35. Blumenthal, L.: Theory and Applications of Distance Geometry. Oxford University Press, United Kingdom. (1953)
36. Achlioptas, D.: Database-friendly random projections: Johnson-lindenstrauss with binary coins. Journal of Computer and System Sciences **66**, 671–687 (2003)

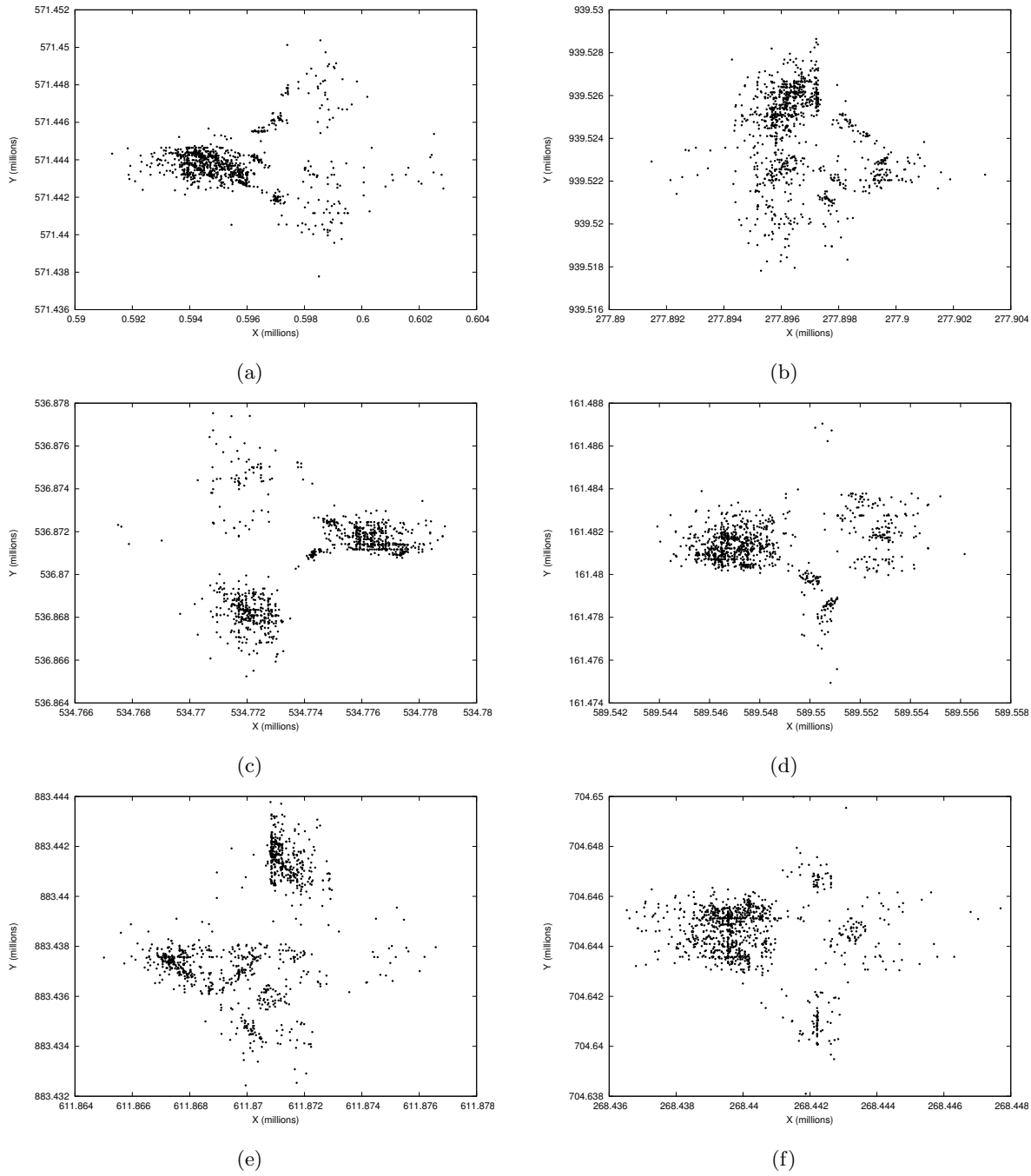

Figure 2: Figures (a), (b), (c), (d), (e), and (f), show plots of the two dimensional projections synthesized by the SANJAY algorithm for 1000 randomly chosen data points from 6 flow cytometry datasets (dataset IDs 9, 24, 11, 14, 17, and 5 respectively in Table 1 on page 3). For these and 24 other flow cytometry datasets, Table 1 lists the maximum distance distortion when 12-dimensional flow cytometry data is projected onto 2 dimensions.