A Graph-Based Method for Detecting Rare Events: Identifying Pathologic Cells

Enikö Székely ▪ *New York University*

Arnaud Sallaberry ▪ *Université Paul Valéry Montpellier 3*

Faraz Zaidi ▪ *University of Lausanne and Karachi Institute of Economics and Technology*

Pascal Poncelet ▪ *Université Montpellier 2*

With recent technological developments, the acquisition and storage of large datasets from various domains are now common. However, outliers in these datasets can arise due to changes in system behavior, instrument or human error, intentional fraudulent behavior, or natural deviations in population resulting from epidemics and virus infections. The detection of these outliers now has many applications in data cleansing, stopping fraudulent intentions, controlling disease outbreaks, detecting infected individuals, and so on.

A recent outlier detection research area concerns the identification of *rare events*. Members of this group of outliers are similar to each other, but they deviate from the dataset's general behavior, thus arousing suspicions that they were generated by a different mechanism. Furthermore, these groups are usually small compared with clusters or groupings within the entire dataset, which is why they're classified as outliers. Detection of such groups is more challenging when the aim is to detect only the groups of outliers that are similar to each other but markedly different from the entire population. Examples of such rare events include identification of students in a class that excel academically or a group of spammers or autobots that increase the popularity of an individual or an event in an online social network. (See the "Defining an Outlier and a Group of Outliers" sidebar for more information on this topic.)

One critical application of detecting rare events arises in biomedical science and clinical research, specifically, the problem of extracting rare events from flow cytometry standard (FCS) data (see the "Flow Cytometry" sidebar). Such files consist of multiparametric descriptions of thousands to millions of individual and rare cells—called *biological markers of interest*—are used to monitor vascular diseases, oncology, and infectious diseases. The detection of rare events with a high recall—that is, with no false negatives—is critical in this domain because the cost of missing pathologic cells is significantly higher than the cost of misclassifying a healthy group of cells.

Although the problem of identifying single outliers has been extensively studied in the literature, little effort is devoted to detecting small groups of outliers. A novel method to solve this challenging problem lies at the frontiers of outlier detection and clustering of similar groups.

In this article, we address this challenging problem of detecting rare events in FCS files. Our proposed approach is based on initial candidate selection using k-nearest neighbors (kNN), filtering irrelevant candidates, applying a metric for detecting densely connected data items, and rendering a representation for

Defining an Outlier and a Group of Outliers

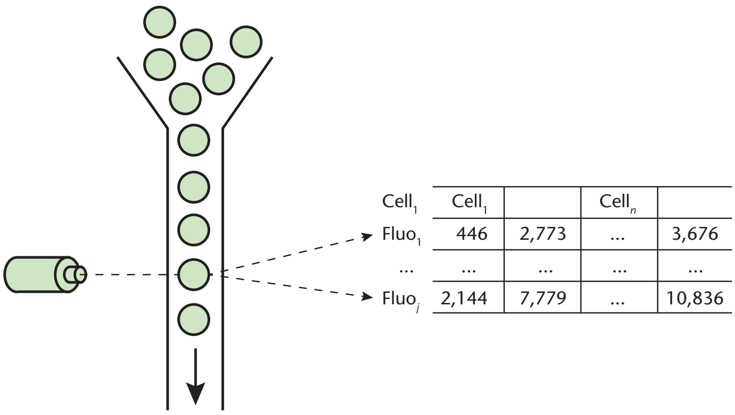
According to Douglas Hawkins, "An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism."[1] Vic Barnett and Toby Lewis define an outlier as "An observation or subset of observations which appear to be inconsistent with the remainder of that dataset."[2]

Other terminologies commonly used for outliers are novelty detection, anomaly detection, one-class classification, noise detection, deviation detection, and exception mining.[3] Outlier detection has been found to be useful in numerous applications, such as intrusion detection in computer networks, fraudulent usage of credit cards, topic detection in news documents and webpages, discovery of temporal changes in evolving online social networks, and identification of inconsistent digital records.

Analogous to an outlier, a group of outliers can be defined as a subpopulation of individuals with general behavior that is similar to each other but that differs from the entire population. The cardinality of this set of rare events is usually small compared with the general grouping of the dataset, which classifies them as outliers. We use the term *rare events* in the main article to refer to this group of outliers, but synonyms such as a cluster of outliers,[4] clustered anomaly,[5] anomaly collection,[6] and microclusters[7] are also used in the literature.

References

1. D.M. Hawkins, *Identification of Outliers*, vol. 11, Chapman and Hall London, 1980.
2. V. Barnett and T. Lewis, *Outliers in Statistical Data*, 2nd ed., Wiley, 1984.
3. V. Hodge and J. Austin, "A Survey of Outlier Detection Methodologies," *Artificial Intelligence Rev.*, vol. 22, no. 2, 2004, pp. 85–126.
4. D.M. Rocke and D.L. Woodruff, "Identification of Outliers in Multivariate Data," *J. Am. Statistical Assoc.*, vol. 91, no. 435, 1996, pp. 1047–1061.
5. F.T. Liu, K.M. Ting, and Z.-H. Zhou, "On Detecting Clustered Anomalies Using Sciforest," *Machine Learning and Knowledge Discovery in Databases*, Springer, 2010, pp. 274–290.
6. H. Dai et al., "Mining Coherent Anomaly Collections on Web Data," *Proc. Conf. Information and Knowledge Management* (CIKM), 2012, pp. 1557–1561.
7. D.-H. Bae et al., "Outlier Detection Using Centrality and Center-Proximity," *Proc. Conf. Information and Knowledge Management* (CIKM), 2012, pp. 2251–2254.



Cell_1	Cell_1		Cell_n	
Fluo_1	446	2,773	...	3,676
...	...	...	...	...
Fluo_j	2,144	7,779	...	10,836

Figure 1. Principle of a flow cytometer. Cells are suspended in a stream and passed by a laser beam. The light hitting the cells is re-emitted according to the cell characteristics. Values (fluorescent signals from individual cells) are stored in corresponding fields for further analysis.

interactive visual detection of the cluster of interest. Results demonstrate the accuracy and effectiveness of our proposed algorithm when compared with available ground truth. We conducted experiments to detect pathologic cells, and we intend to extend this study to other types of datasets as part of future work. (See the "Outlier Detection Approaches" sidebar for a review of earlier work.)

Flow Cytometry

Flow cytometry is a laser-based, biophysical technology used in cell counting, sorting, biomarker detection, and protein engineering. This technology allows the measurement of blood cell characteristics at very high rates (up to thousands of cells per second). Figure 1 illustrates the flow cytometry process.

Each cell passes through one or more light beams that measure the fluorescent signals from individual cells—these have different possible responses depending on the fluorophores added to the blood sample. This fluorescence process generates considerable information about cells and allows their separation (an antibody is linked to a fluorescent dye and bound to a protein that's discriminative between cells). Finally, fluorescence levels in response to cell markers are stored in flow cytometry standard (FCS) data files.[1]

Current flow cytometers can count up to tens of millions of cells in the normal cell populations found in any healthy patient, such as lymphocytes or monocytes. In patients presenting with a blood pathology, the blood samples also contain microclusters of cells with abnormal signatures—that is, abnormal combinations of cell marker fluorescence levels.

The operator usually performs a visual detection by sequentially inspecting two-dimensional spaces, which are combinations of two markers (see Figure 2). This approach leads to high intervariability (17 to 44 percent)[2] among research laboratories about what defines an abnormal cell population; it's also sensitive to complex multivariate relationships.

Flow cytometers of the current generation have a capacity for analyzing more than 10^5 cells per second. They measure the characteristics of single cells determined by visible and fluorescent light emissions from the markers on the cells. These labeled cells pass a laser that emits light at a specific wavelength according to the specific markers attached to the cell fluoresce. For each cell, a fluorescence intensity value is collected and stored in FCS data files[3] that consist of multiparametric descriptions of thousands to millions of individual cells. Analyzing and sorting subpopulations widely representing immune cells (CD4 + T lymphocytes, CD8 +, B, or NK) is a common practice.[4]

Recent biomedical science and clinical research has addressed the problem of extracting rare events from these data files[3] with 1×10^2 to 1×10^3 cells per milliliter (ml) of blood cells for 20 ml of blood. Rare events in these cells often occur at a very low frequency, with researchers citing this number as between 0.1 to 0.00001 percent of the total population.[5] Recent advances in flow cytometry have enabled it to emerge as an important tool in the systems biological approach to theoretical and clinical research.[3]

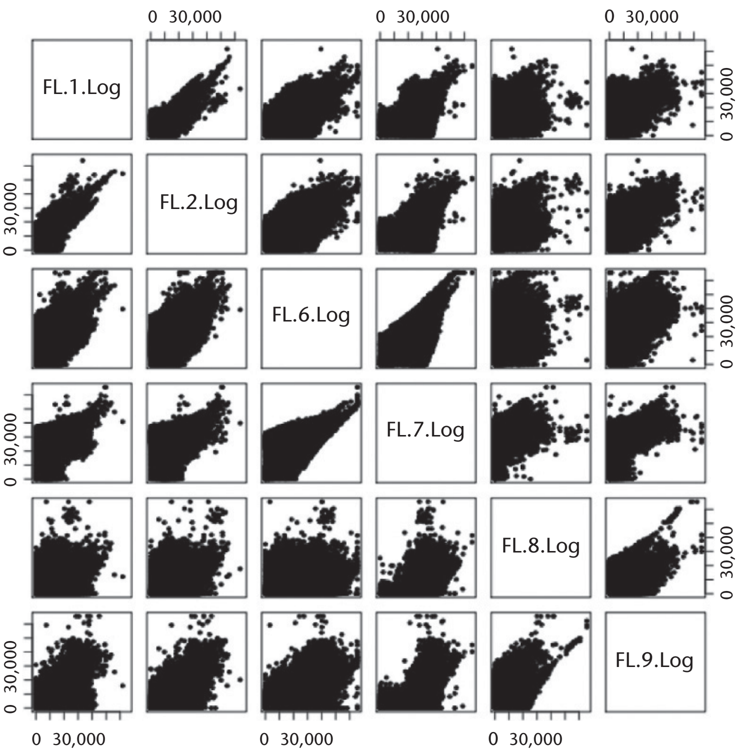


Figure 2. Original data events on a flow cytometry blood sample. The standard approach of visually inspecting combinations of markers high intervariability among research laboratories.

Methods for analyzing FCS consist of grouping individual cell data records into discrete populations based on similarities in light scattering and fluorescence.[6] This is usually done by sequential manual partitioning (or *gating*)—plotting different combinations of descriptors two at a time in a 2D scatter plot and then selecting subgroups of cells using gates. Cells within the gates are selected for further analysis and plotted in another 2D scatter plot with a different axis. However, the main problem with this approach is that it's tedious, it can miss subgroups of rare cells,[7] and there are difficulties in effectively analyzing high-dimensional data.[4]

Proposed Method

Our proposed method takes tabular data as input, with each row corresponding to a data item (a blood cell) along with its numeric attributes and the cluster of interest, which is a set of rare events, as shown in Figure 3.

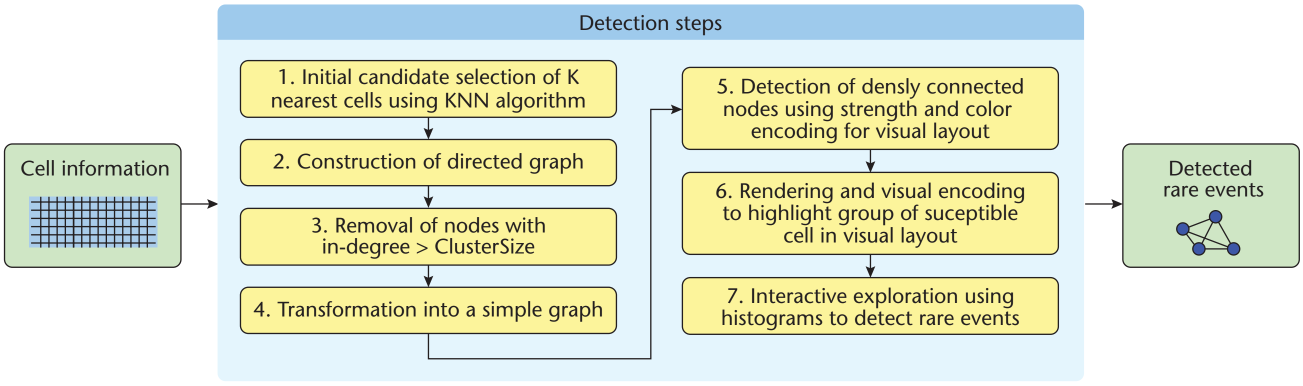

Detection steps
Cell information
1. Initial candidate selection of K nearest cells using KNN algorithm
2. Construction of directed graph
3. Removal of nodes with in-degree > ClusterSize
4. Transformation into a simple graph
5. Detection of densly connected nodes using strength and color encoding for visual layout
6. Rendering and visual encoding to highlight group of suceptible cell in visual layout
7. Interactive exploration using histograms to detect rare events
Detected rare events

Figure 3. Proposed method. Taking tabular data as input, we use these seven steps to detect rare events.

Outlier Detection Approaches

There are different approaches to detecting outliers in the literature.[1] For example, we can determine outliers without any prior knowledge, an approach that processes data and identifies the most distant points as outliers. If the data distribution is known, data points that don't follow the known distribution can be classified as outliers. Alternatively, there can be multiple predetermined classes of normal data, and deviation from these classes reveals outliers. Another perspective considers parametric and nonparametric methods: statistical methods often take some parameters as input, whereas methods based on distance and density don't require any input parameters.[1]

A completely different technique for detecting groups of outliers is based on clustering algorithms, where the idea is to identify clusters that are smaller in size and classify them as outliers.[2] Clustering algorithms such as K-means are ideally suited to finding convex clusters, but these algorithms are also highly sensitive to input parameters and can result in misclassification of clusters and outliers.[3] Usually, clustering algorithms attempt to balance clusters of varying sizes. For example, spectral clustering algorithms have gained in popularity because if their low time complexity and scalability, but they use RatioCut and Ncut to create balanced clusters,[4] thus making them impractical for detecting rare events. Some clustering algorithms allow the generation of different size clusters,[5] but a priori knowledge about cluster size is required to detect rare events, which is difficult in most domains. Furthermore, different clustering algorithms can result in different clusters for the same dataset. With all of these described inconsistencies in clustering algorithms, it's hard to rely on them to detect true positives, especially in critical applications.

Cluster-based approaches often use densities and distances to identify outliers. For instance, DBSCAN,[6] the most common density-based clustering algorithm, uses the notion of density reachability to allow the detection of clusters of arbitrary sizes and shapes, but it can't handle clusters of different densities. Another approach is based on the notion of isolation.[7] Such methods take advantage of the fact that anomalies rarely occur in datasets; based on training using subsamplings and evaluation stages, they instead discover rare events by building forests of binary trees. These methods are effective in revealing global rare events, but they perform suboptimally when rare events are close to the entire population's general behavior. In LOCI,[8] the detection of outlying clusters depends on the choice of nearest-neighbor minimum number of points (MinPts) that define the local neighborhood. Actually, the detection of very small clusters requires a MinPts large enough to contain all points in a cluster—that is, larger than the cluster's size. LOCI thus proposes a multigranularity deviation factor (MDEF) and identifies outliers as points with a neighborhood size that is significantly different than their neighbors' neighborhood size. It then relies on an appropriate choice of neighborhood size and requires the neighborhood's maximum radius as an input parameter.

A new approach, called RARE,[9] proposes a two-step process to extract rare cells. First, it prunes the search space by removing obvious clusters that don't contain rare events. Second, it carefully grows these clusters, preserving their consistency. Although this approach has proved efficient for extracting rare events, the major drawback is its dependency on the required input parameters, which are hard to predetermine.

Another approach to detecting outliers is the use of summary statistics and visual representations. Boxplots, along with its variations,[10] have commonly been used to compare univariate distributions as well as to detect outliers. However, these visual representations are hard to read and not scalable with multivariate data.

Recent advances in visual analytics and visual data mining have also introduced approaches to detecting outliers through visual representation and interactive exploration.[11] Visualization techniques exploit the human pattern recognition capacity to detect anomalies and are developed by building user interfaces and interactions to deal with the graphical representation of data.[12] The problem with these approaches is their limited application to large datasets; it becomes hard to interactively explore and find rare events in thousands and millions of data items.

Extensive literature on the outlier detection issue is available in the form of surveys and books,[2,13–16] but these works don't approach the problem of detecting rare events. In the main text, we introduce a novel method to address this issue from an interactive visualization standpoint and demonstrate that proper visual encoding is a powerful technique for exploring very large datasets.

References

1. N. Suri, M. Murty, and G. Athithan, "Data Mining Techniques for Outlier Detection," *Visual Analytics and Interactive Technologies: Data, Text, and Web Mining Applications*, IGI-Global, 2011, p. 19.
2. V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys*, vol. 41, no. 3, 2009, article no. 15.

For each pair of data items, we calculate the Euclidean distance as the similarity metric among data items. The next step is to construct a graph based on this similarity among data items. For each node, we find the nearest neighbors in terms of distances using the kNN algorithm,[8] where K is a priori known. For every node, directed edges are introduced, with the target nodes being a node's K

3. S.R. Gaddam, V.V. Phoha, and K.S. Balagani, "K-Means+ id3: A Novel Method for Supervised Anomaly Detection by Cascading K-Means Clustering and id3 Decision Tree Learning Methods," *IEEE Trans. Knowledge and Data Eng.*, vol. 19, no. 3, 2007, pp. 345–354.
4. U. Luxburg, "A Tutorial on Spectral Clustering," *Statistics and Computing*, vol. 17, no. 4, 2007, pp. 395–416.
5. S. Zhu, D. Wang, and T. Li, "Data Clustering with Size Constraints," *Knowledge Based Systems*, Elsevier, 2010, pp. 883–889.
6. M. Ester et al., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *Proc. 2nd Int'l Conf. Knowledge Discovery and Data Mining* (SIGKDD), 1996, pp. 226–231.
7. F. Tony Liu, K.M. Ting, and Z.-H. Zhou, "On Detecting Clustered Anomalies Using Sciforest," *Machine Learning and Knowledge Discovery in Databases*, Springer, 2010, pp. 274–290.
8. S. Papadimitriou et al., "LOCI: Fast Outlier Detection Using the Local Correlation Integral," *Proc. 19th Int'l Conf. Data Engineering* (ICDE), 2003, pp. 315–326.
9. E. Székely et al., "A Density-Based Backward Approach to Isolate Rare Events in Large-Scale Applications," *Proc. 16th Int'l Conf. Discovery Science* (DS 2013), 2013, pp. 249–264.
10. R. McGill, J.W. Tukey, and W.A. Larsen, "Variations of Box Plots," *Am. Statistician*, vol. 32, no. 1, 1978, pp. 12–16.
11. M.C. Hao et al., "Business Process Impact Visualization and Anomaly Detection," *Information Visualization*, vol. 5, no. 1, 2006, pp. 15–27.
12. Y. Kou et al., "Survey of Fraud Detection Techniques," *Proc. IEEE Int'l Conf. Networking, Sensing and Control*, 2004, pp. 749–754.
13. V. Barnett and T. Lewis, *Outliers in Statistical Data*, Wiley, 1984.
14. V. Hodge and J. Austin, "A Survey of Outlier Detection Methodologies," *Artificial Intelligence Rev.*, vol. 22, no. 2, 2004, pp. 85–126.
15. P.J. Rousseeuw and A.M. Leroy, *Robust Regression and Outlier Detection*, vol. 589, Wiley, 2005.
16. M. Agyemang, K. Barker, and R. Alhajj, "A Comprehensive Survey of Numeric and Symbolic Outlier Mining Techniques," *Intelligent Data Analysis*, vol. 10, no. 6, 2006, pp. 521–538.

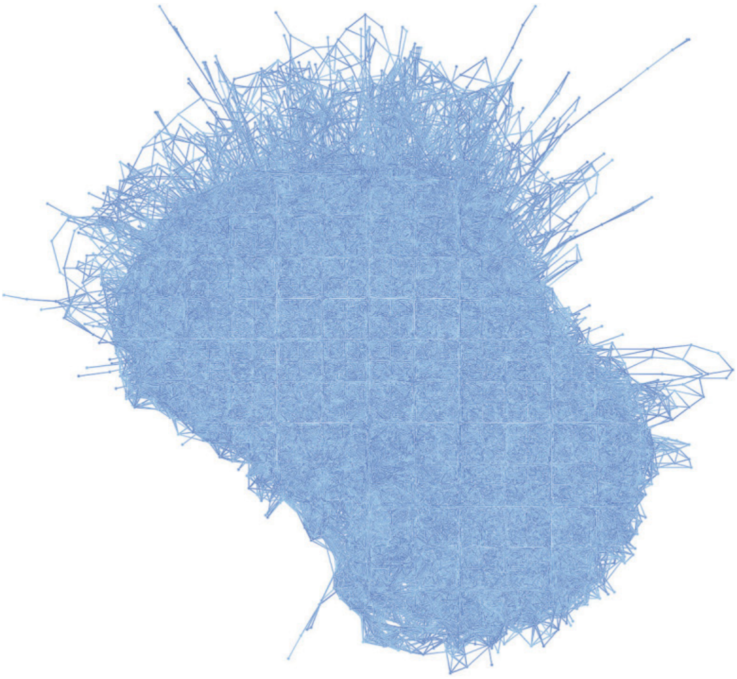
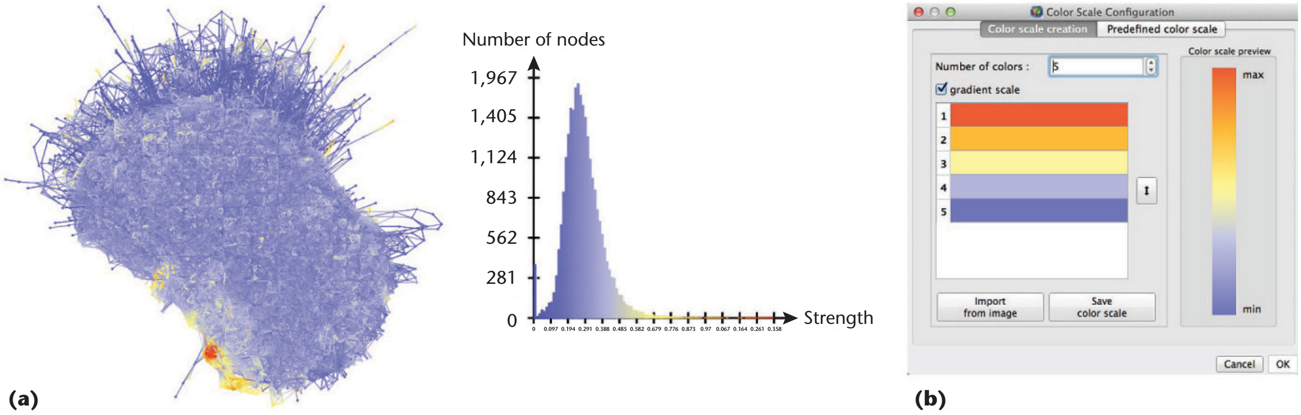


Figure 4. Visual representation of the simple graph generated in step 4 of our proposed method. A force directed algorithm FM3 is used to render the graph.[9,10]

nearest neighbors in terms of the Euclidean distance calculated earlier. The choice of K value depends on the estimated size of the rare events that we're trying to detect. For problems where this size can't be estimated, we interactively test different values to find an appropriate threshold. For pathologic cells, because we know that the usual cluster size is 50, we consider $K = 100$ so as to ensure that we don't miss any true positives.

Hence, we obtain a directed graph where each node is connected to its K most similar data items. Because the maximum number of data items in a cluster is a priori known, we apply a filter to remove all nodes with an in-degree greater than the known cluster size. This is because all these nodes, which are similar to many other nodes, represent regular data items that can be found readily in the graph and thus can't belong to the group of rare events. After filtering nodes with an in-degree higher than the cluster size, we ignore the orientation of edges to obtain a simple graph; the edge direction isn't used in further processing. Figure 4 shows the graph we obtain as a result.[9,10]

The next step aims to find clusters of nodes based on their structural similarity. We use the strength metric to detect densely connected groups of nodes within the graph. David Auber and his colleagues introduced this metric to quantify the neighborhood cohesion of a given edge and thus determine if it's an intra- or an inter-community edge within a network.[11] The metric assigns nodes and edges a high value if they're connected densely to each other (just like a clustering coefficient), but it considers cycles of size 4 as well.



Number of colors : 5

☑ gradient scale

1
2
3
4
5

Figure 5. Computing the strength metric in step 6 and visually encoding it on nodes/edges, from blue (low values) to red (high values).[9] (a) The histogram shows the frequency distribution of different strength values; rare events appear in red at the bottom left of the graph. (b) In the color-mapping plugin Tulip,[10] the user defines a list of colors—the first one is given to nodes having the highest values and the last to nodes having the lowest. A linear interpolation between consecutive colors of the list is used for the other values.

The strength of an edge e given by $w_s(e)$ is defined as follows:

$$w_s(e) = \frac{\gamma_{3,4}(e)}{\gamma_{\max}(e)},$$

where $\gamma_{3,4}(e)$ is the number of cycles of sizes 3 or 4 that the edge e belongs to and $\gamma_{\max}(e)$ is the maximum possible number of such cycles. Based on this definition, we define the strength of a vertex as follows:

$$w_s(u) = \frac{\sum_{e \in \mathrm{adj}(u)} w_s(e)}{\deg(u)},$$

where adj(u) is the set of edges adjacent to u, and deg(u) is the degree of vertex u. The time complexity to calculate the strength metric over all vertices (V) and edges (E) is $O(|E| \bullet (\deg_{\max})^2)$, where $\deg_{\max}$ is the maximum degree of the graph. In our case, because the maximum degree is bounded by a constant factor, the calculation remains constant in linear time in terms of number of edges in the graph.

Figure 5 shows the result of calculating the strength metric and applying color encoding on the graph nodes and vertices. The scale depends on the node value interval; we used the Tulip plugin for color mapping.[12] The user defines a list of colors, with the first one given to nodes having the highest values and the last one to nodes having the lowest one. A linear interpolation between consecutive colors of the list is used for the other values. In our example, nodes and edges in red highlight the potential rare events in the figure (Figure 5a shows the result, and Figure 5b shows the color scale).

We prefer the strength metric to the clustering coefficient because triads are more frequently present in these datasets as compared to cliques of size 4. If we use a clustering coefficient as a metric to detect densely connected groups of nodes, a large number of true negatives will be detected. This will ultimately slow down the interactive detection process because domain experts would require further manual verification. We didn't use metrics to calculate cliques of size 5 and above because such a calculation would become too slow for large datasets, as discussed in detail elsewhere.[13]

The final step is to visually detect the presence of rare events that form a cluster in the graph. We plot the histogram of strength of nodes along with their frequencies (as shown in Figure 5a), which immediately reveals that many nodes have very low strength value. Domain experts interactively remove those nodes from the graph by using histograms of frequency distribution, eventually leaving only a few nodes with high strength value, as shown in Figure 6a.

Domain experts then manually remove true negatives and obtain the required rare events, which is a set of densely connected nodes that have a high similarity to each other, as shown in Figure 6b. High similarity of nodes is depicted with color encoding: the red color indicates high similarity, and a gradual degradation to blue indicates low similarity among pairs of nodes.

Case Studies and Prototype

A comparative study of different graph-drawing software packages clearly shows that Tulip scales well to rendering graphs and networks with hundreds of thousands of nodes and edges,[14] thus making it the ideal platform to implement our proposed method.

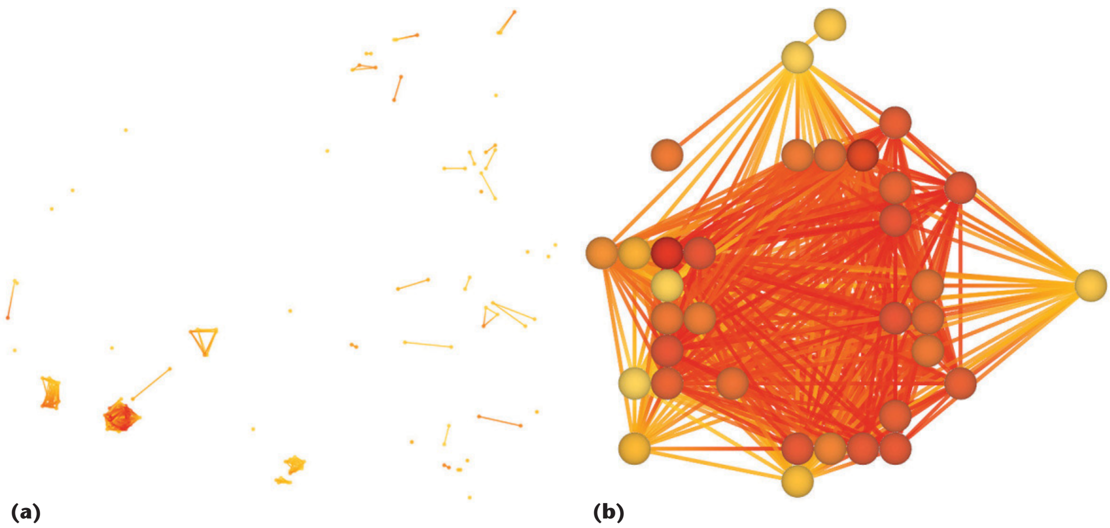


Figure 6. Detecting the presence of rare events. (a) Interactive and manual removal of nodes in step 7 involves selecting nodes with low strength values in the histogram. Nodes with high strength couldn't be removed via histograms. (b) After performing step 7 of our proposed method, a set of rare events appears.

The computation time of the strength metric takes only a few seconds (even on the larger datasets described hereafter), and manual removal is fast enough to preserve the tool's interactivity.

Real-World Datasets

A domain expert performed experiments on real-world datasets with our method as we explained to him how to interact with the tool (see Table 1). He validated the results obtained using classical methods.

Figure 7 shows the results obtained for the three datasets presented in Table 1. When compared with the available ground truth, these results demonstrate that our proposed method successfully found the pathologic cells within the doctor-provided dataset. In all three cases, the domain expert and the ground truth also identified three to seven false positives, which is an acceptable result because none of the true positives were missed by the proposed method.

Table 2 shows the number of nodes filtered at different stages of the process. The number of false positives detected is negligible compared with the size of the dataset provided.

Benchmark

We also ran experiments on synthetic datasets that we constructed by injecting grown pathological blood cells into a cell sample from a healthy patient. The size of the rare population injected was 50 in a dataset containing 700,000 cells.

By applying RARE on this dataset, we detected 31 rare cells. We also conducted experiments with LOCI, which is considered to be one of the most efficient approaches for detecting rare events. To evaluate the best parameter setting, we chose various values of the maximum radius in LOCI {3,000, 4,000, 5,000, 6,000}. Each time, we obtained a score of 1 for points in the rare event, indicating inliers and rare events that couldn't be detected. For values of a radius larger than 6,000, we ran into memory problems. We performed additional experiments with DBSCAN, which usually reports high recall (generally, 100 percent), but for most parameter values, the rare events were left unclustered and belonginh to the subset classified as noise.

Table 1. Datasets used for experimentation with available ground truth.

Patient	Disease	Total nodes	Pathologic cells
P1	Intracranial aneurysm	1,895,261	25
P2	Intracranial aneurysm	2,524,916	15
Pc	Cancer	2,470,042	7

Table 2. Number of nodes after different filtration steps in the detection process.

Patient	Total nodes	Nodes after Step 3	Nodes after Step 6	After removal of disconnected nodes	Pathologic cells	False positives
P2	1,895,261	126,049	33	31	25	6
P2	2,524,916	138,647	80	22	15	7
P3	2,470,042	182,626	20	10	7	3

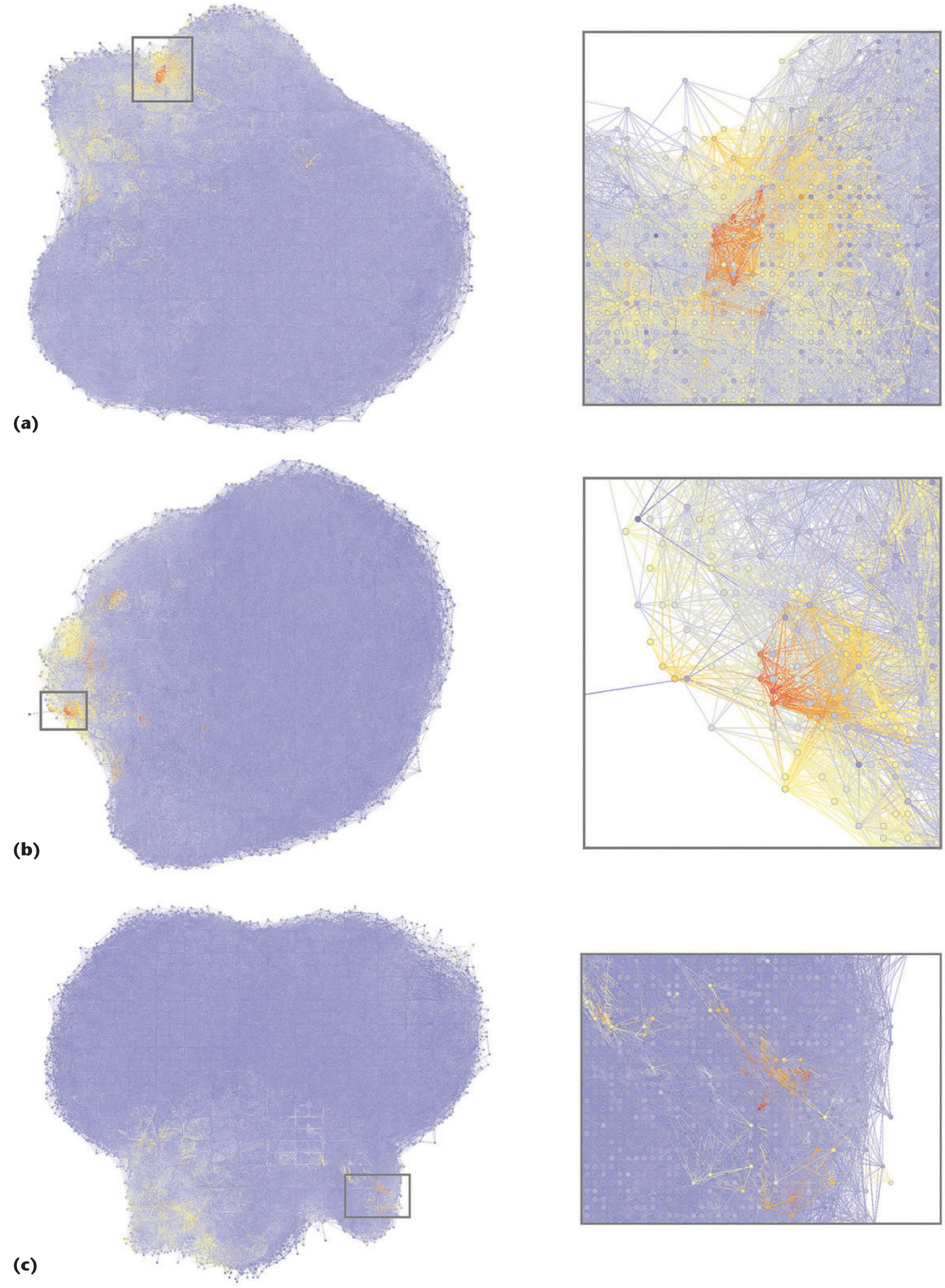


Figure 7. Results for the three datasets in Table 1. For (a) patient 1, (b) patient 2, and (c) patient 3, the highlighted region shows the pathologic cells found using our proposed method.

Figures 5 and 6 show the results obtained by applying our method to the same synthetic dataset, in which we detected 37 cells. Unlike the real-world dataset, some pathologic cells weren't detected, but all detected cells were from the rare injected population. Even though our method doesn't find all the injected cells, it outperformed RARE by finding six more cells.

We've proposed a novel method for detecting a group of rare events in large networks.

The results we obtained on a real-world biological dataset clearly demonstrate the superiority of our proposed method's accuracy. Furthermore, the algorithm is highly efficient in terms of time complexity once we've calculated the K-nearest neighbors. We intend to explore this method with other real-world datasets, most notably, on social networks, where the detection of groups of outliers and rare events has many applications.

Acknowledgments

We thank Renaud Cezar from the Institute for Research in Biotherapy, Montpellier, France. His expertise helped us understand the data and validate the final results. This work was mainly funded by Labex Numev.

References

1. H.M. Shapiro, *Practical Flow Cytometry*, 4th ed., Wiley-Blackwell, 2003.
2. A. Bashashati and R.R. Brinkman, "A Survey of Flow Cytometry Data Analysis Methods," *Advances in Bioinformatics*, 2009, pp. 1–19.
3. E.A. O'Donnell, D.N. Ernst, and R. Hingorani, "Multiparameter Flow Cytometry: Advances in High-Resolution Analysis," *Immune Network*, vol. 13, no. 2, 2013, pp. 43–54.
4. E. Lugli, M. Roederer, and A. Cossarizza, "Data Analysis in Flow Cytometry: The Future Just Started," *Cytometry Part A*, vol. 77, no. 7, 2010, pp. 705–713.
5. A.D. Donnenberg and V.S. Donnenberg, "Rare-Event Analysis in Flow Cytometry," *Clinics in Laboratory Medicine*, vol. 27, no. 3, 2007, pp. 627–652.
6. N. Aghaeepour et al., "Critical Assessment of Automated Flow Cytometry Data Analysis Techniques," *Nature Methods*, vol. 10, 2013, pp. 228–238.
7. A.D. Donnenberg, V.S. Donnenberg, and G. Byrne, "Rapid Data Handling in Flow Cytometric Rare Event Analysis," *Biotech Int'l*, vol. 21, no. 1, 2009, p. 16.
8. E. Fix and J.L. Hodges, "Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties," tech. report 4, US Air Force School of Aviation Medicine, 1951.
9. S. Hachul and M. Jünger, "Drawing Large Graphs with a Potential-Field-Based Multilevel Algorithm," *Proc. 12th Int'l Symp. Graph Drawing*, LNCS 3383, Springer, 2005, pp. 285–295.
10. S. Hachul and M. Jünger, "An Experimental Comparison of Fast Algorithms for Drawing General Large Graphs," *Proc. 13th Int'l Symp. Graph Drawing*, LNCS 3843, Springer, 2006, pp. 235–250.
11. D. Auber et al., "Multiscale Visualization of Small World Networks," *Proc. IEEE Symp. Information Visualization* (InfoVis), 2003, pp. 75–81.
12. D. Auber, "Tulip: A Huge Graph Visualization Framework," *Graph Drawing Software, Mathematics and Visualization Series*, P. Mutzel and M. Jünger, eds., Springer Verlag, 2003, pp. 105–126.
13. G. Melançon and A. Sallaberry, "Edge Metrics for Visual Graph Analytics: A Comparative Study," *Proc. 12th Int'l Conf. Information Visualization* (IV), 2008, pp. 610–615.
14. B. Pinaud and P. Kuntz, "GVSR: An On-line Guide for Choosing a Graph Visualization Software," *Proc. 18th Int'l Symp. Graph Drawing*, LNCS 6502, Springer, 2011, pp. pages 400–401.

Enikö Székely *is a postdoctoral researcher at the Courant Institute of Mathematical Sciences (CIMS) at New York University. Her research interests include data mining, machine learning, and visualization. Székely received a PhD in computer science from the University of Geneva. Contact her at eniko.szekely@nyu.edu.*

Arnaud Sallaberry *is an assistant professor at the University of Montpellier 3, France. His research interests include graph visualization, visual data mining, interactive visualizations, and network analysis. Sallaberry received a PhD in computer science from the University of Bordeaux 1, France. Contact him at arnaud.sallaberry@lirmm.fr.*

Faraz Zaidi *is a postdoctoral researcher at the University of Lausanne, Switzerland, but he also holds a permanent position as an assistant professor at the Karachi Institute of Economics and Technology, Karachi, Pakistan. His research interests include data mining, information visualization, social network analysis, graphs, and algorithms. Zaidi received a PhD in computer science from the University of Bordeaux 1, France. Contact him at faraz@pafkiet.edu.pk.*

Pascal Poncelet *is a full professor at the University of Montpellier 2, France, and head of the data mining research group at the Montpellier Laboratory of Informatics, Robotics and Microelectronics. His research interests include advanced data analysis techniques for emerging applications, data mining techniques, and new algorithms for mining patterns. Poncelet received a PhD in computer science from University of Nice-Sophia Antipolis. Contact him at pascal.poncelet@lirmm.fr.*