Bioinformatics, 32(3), 2016, 417–423
doi: 10.1093/bioinformatics/btv594
Advance Access Publication Date: 17 October 2015
Original Paper

Systems Biology

Scalable clustering algorithms for continuous environmental flow cytometry

Jeremy Hyrkas[1,*], Sophie Clayton[2], Francois Ribalet[2], Daniel Halperin[1,3], E. Virginia Armbrust[2,3] and Bill Howe[1,3]

[1]Department of Computer Science and Engineering, [2]School of Oceanography and [3]eScience Institute, University of Washington, Seattle, WA 98195, USA

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on May 19, 2015; revised on September 20, 2015; accepted on October 12, 2015

Abstract

Motivation: Recent technological innovations in flow cytometry now allow oceanographers to collect high-frequency flow cytometry data from particles in aquatic environments on a scale far surpassing conventional flow cytometers. The SeaFlow cytometer continuously profiles microbial phytoplankton populations across thousands of kilometers of the surface ocean. The data streams produced by instruments such as SeaFlow challenge the traditional sample-by-sample approach in cytometric analysis and highlight the need for scalable clustering algorithms to extract population information from these large-scale, high-frequency flow cytometers.

Results: We explore how available algorithms commonly used for medical applications perform at classification of such a large-scale, environmental flow cytometry data. We apply large-scale Gaussian mixture models to massive datasets using Hadoop. This approach outperforms current state-of-the-art cytometry classification algorithms in accuracy and can be coupled with manual or automatic partitioning of data into homogeneous sections for further classification gains. We propose the Gaussian mixture model with partitioning approach for classification of large-scale, high-frequency flow cytometry data.

Availability and Implementation: Source code available for download at https://github.com/jhyrkas/seaflow_cluster, implemented in Java for use with Hadoop.

Contact: hyrkas@cs.washington.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

We present an exploration of scalable machine learning solutions to a problem motivated by an emerging class of high-frequency, continuous-operation flow cytometers (Dubelaar *et al.*, 1999; Olson and Sosik, 2007; Swalwell *et al.*, 2011). Existing analysis tools assume individual samples are processed independently; these tools assume a small number of discrete, independent samples and are therefore not appropriate for newer devices. We therefore adapt and extend existing methods to propose the first scalable, accurate classification solution for large-scale continuous environmental flow cytometry. While this method has its roots in biological oceanography, we show that the solutions identified for analyzing these datasets are applicable in other domains as well.

1.1 Flow cytometry

Flow cytometry advances individual cells through a thin capillary and then measures each cell's optical properties using laser light. The scattering patterns of the laser can be used to infer cell size, and the wavelengths of the fluorescence signal are used to infer the concentration of various pigments. Flow cytometry can therefore discriminate between cells and detritus or suspended sediments, and between photosynthetic and non-photosynthetic organisms.

Researchers in the environmental sciences use flow cytometry to analyze the microbial populations in water or soil samples, or to measure pollutants in the air (Tarnok *et al.*, 2001). Marine microbial ecologists are in particular interested in the dynamics of phytoplankton, which produce about half of the oxygen on earth and are foundational to the oceanic food web (Field *et al.*, 1998). With conventional flow cytometry, discrete samples are collected for cytometric analysis; e.g. to classify particles by species of phytoplankton. The most common classification method is *manual gating*, where the physical boundaries for clusters of cells are manually identified. While this process incorporates expert knowledge, it can lead to somewhat subjective cluster boundaries (Aghaeepour *et al.*, 2011). Furthermore, as cytometry samples became more numerous, manual classification by direct inspection of scatter plots became infeasible and has been replaced by automated learning algorithms (Aghaeepour *et al.*, 2013).

1.2 The SeaFlow project

The SeaFlow instrument is a state-of-the-art environmental flow cytometer, designed to be deployed on oceanographic research vessels and operated continuously over several weeks (Swalwell *et al.*, 2011). Different populations of phytoplankton associated with different environments are commonly observed during a research cruise as the vessel moves through different water masses. SeaFlow continuously samples surface seawater, generating a time series of cytometry samples (one every 3 min) containing measurements of the optical properties of small phytoplankton cells (less than 10 microns in diameter). After several weeks, there are thousands of 3-min samples to be classified from highly variable environmental conditions, representing hundreds of gigabytes of data (we refer to a dataset containing these samples as a *cruise-scale* cytometry dataset).

Each phytoplankton cell observed is defined by its forward light scattering collected in orthogonal and perpendicular polarization states (a proxy for cell size and cell calcification state, respectively) and two different wavebands of fluorescence: one associated with chlorophyll *a* pigment (centered on 690 nm for red fluorescence) and one associated with phycoerythrin pigment (centered on 570 nm for orange fluorescence). The same phytoplankton species may exhibit different optical properties as environmental conditions change over time and space. For instance, cell size increases during daylight as cells fix carbon by photosynthesis but decreases when cells undergo cell division; changes in nutrient availability can also influence cell size (Sosik *et al.*, 2010). As a result, the location and shape of clusters of cells in this four-dimensional space will vary as samples are collected at different times and locations.

1.3 Algorithmic challenges of automated analysis

Most samples from conventional flow cytometers are treated as independent datasets to be processed individually. However, SeaFlow samples are collected by the same instrument with the same configuration and show snapshots of populations across a continuous environment. The 3-min mark used to produce samples is somewhat arbitrary, and population change may occur at a much slower pace. As a result, SeaFlow samples are more natural to aggregate than, e.g. cytometry data collected in the same environment but in separate experiments (possibly with different cytometers or under different experimental conditions). We expect that aggregating and classifying large segments of the dataset, as opposed to individual samples, will improve a classifier's ability to estimate population densities and boundaries. Specifically, sparse samples can be more accurately classified using global knowledge, and particles that appear as outliers in a sample may ultimately be identified as a rare population when viewed in context. However, most state-of-the-art cytometry classification algorithms operate in main memory (Kvistborg *et al.*, 2015) and have not been shown to tolerate this kind of global analysis, especially not at the scale of SeaFlow.

Recently, Finak *et al.* (2014) explored single-node parallel and out-of-core methods for analyzing cytometry data, including classification. However, at the scales suggested by continuous flow cytometers, distributed computing platforms that have been shown to scale to hundreds or thousands of computers such as Hadoop (Shvachko *et al.*, 2010) and Spark (Zaharia *et al.*, 2010) become important to study.

In this article, we explore methods for classifying SeaFlow samples individually and propose new methods for classifying all samples as one dataset and for choosing contiguous samples that represent distinct population signatures to classify together. The contributions of this article are as follows:

- We evaluate existing algorithms on three large SeaFlow datasets, ignoring continuity and processing each sample independently to establish a baseline.
- To incorporate continuity, we implement a parallel Gaussian mixture models (GMMs) algorithm in Hadoop, a widely-used, open source distributed system that implements a MapReduce-style computational model (Dean and Ghemawat, 2008). We apply this method to full SeaFlow datasets consisting of millions of particles and compare classification quality against sample-based methods, finding that classification performance is improved and is typically comparable to human judgment.
- We show that automatically partitioning data into large but relatively homogeneous segments and classifying each segment independently can improve classification in scenarios where the underlying population distribution has changed. We find that in the absence of expert knowledge to perform this partitioning, change-point detection can provide reasonable partitions.

2 Limitations of conventional methods

Existing automated algorithms for classifying cytometry data have not been shown to scale to the size of a full SeaFlow dataset, which consists of samples taken every 3 min (roughly, one sample per kilometer traveled). It is possible to classify each sample (which we will sometimes refer to as a *3-min window*) as separate cytometry measurements. We expect that the performance of this approach will be worse than training the model over the entire dataset and using all available information. Furthermore, we expect nearby samples to look very similar, since they are drawn from nearly the same underlying environment. Classifying samples independently does not allow for sharing information between classifiers, another lost opportunity to improve classification quality. To test this hypothesis, we examined conventional sample-based methods and applied them to SeaFlow data, classifying each 3-min window separately. We use these methods as baselines to compare against our proposed methods that can handle the full volume and complexity of the SeaFlow data.

2.1 Methods

GMMs have previously been used as the basis for cytometry classification methods (Finak *et al.*, 2009; Lo *et al.*, 2008), notably by the method flowClust (Lo *et al.*, 2009). But in oceanography, GMMs have only previously been used to analyze relatively small datasets

(Demers *et al.*, 1992) consisting of a few dozen cytometry samples taken from a single site in the ocean. To our knowledge, this method has never been applied to thousands of continuous flow cytometry samples collected across varying aquatic environments such as in the SeaFlow datasets.

We applied GMM on a sample-by-sample basis as a baseline. Applying GMM to cytometry data requires setting a parameter k, the number of clusters to find, for each sample. In general, this is a difficult problem and is a major drawback of GMM.

GMM provides a probability for each classification label, which can be used to measure uncertainty: if every label has a low probability for a given particle, the model is uncertain about how to classify it. In practice, we find that particle labels with relatively low probability tend to be associated with the same particles where our labels differ from the human labels. Visually, these particles tend to appear in boundary regions between obvious clusters (Fig. 1). Anecdotally, we confirmed that experts find these particles ambiguous to classify and that they are less concerned with classification accuracy in these regions. Therefore, GMM provides a natural probabilistic method for ignoring ambiguous particles and improving measure performance.

A recent literature review (Aghaeepour *et al.*, 2013) conducted as part of the FlowCAP project examined 77 algorithms for automated clustering of cytometry data and identified a number of automated methods that perform reasonably well on various datasets from the medical field when compared with manual gating. The study is not directly relevant to our goals, however: The FlowCAP project considers only individual, small-scale samples from the medical domain as opposed to large-scale, continuously observed environmental data. However, the study offers a useful point of reference for comparison to prior work. We applied GMM to the FlowCAP datasets to ensure that the performance was similar to other FlowCAP methods involving mixture models. None of the mixture model-based algorithm is competitive with state-of-the-art classification methods. We found the performance of pure GMM to be generally better than the algorithms based around mixture models, with the exception of flowClust. We present the details of these experiments in the Supplementary Materials.

flowMeans (Aghaeepour *et al.*, 2011) and flowPeaks (Ge and Sealfon, 2012) were among the top-performing algorithms in the FlowCAP project. These algorithms are modeled on ideas from flowClust (Lo *et al.*, 2009). Both methods are implemented as R packages, and both extend the k-means algorithm (Arthur and Vassilvitskii, 2007) by choosing the k parameter, so that the number of actual clusters is overestimated and then merging clusters until convergence (which differs between algorithms).

We explored the performance of GMM, flowMeans and flowPeaks on SeaFlow data, applying each method sample-by-sample. We present the results and analysis in Section 4.

3 Cruise-scale clustering

To contrast with baseline sample-based methods, we explored methods for classifying SeaFlow data using different schemes for binning windows, from small sections of contiguous windows (a few hundred samples) to an entire cruise (thousands of samples).

To study this approach on an unrelated dataset, we trained a GMM over all samples from FlowCAP datasets simultaneously. Even though the samples are drawn from independent populations, we do observe an increase in classification accuracy when using all data at once in four of the five datasets. As expected, the classification performance is still below state-of-the-art for these datasets, since GMM is a very broad algorithm unlike the specialized methods evaluated in the FlowCAP study. However, aggregating the samples into one dataset does push GMM closer to state-of-the-art. Because of the increase in performance, we conjecture that our approach of classifying samples as one data will generalize to other domains, even those where samples are drawn independently. However, we leave a more thorough evaluation of this claim for future work. Details of this experiment are provided in the Supplementary Material.

3.1 Scaling up GMMs

GMM optimized using *expectation-maximization* can be parallelized via MapReduce (Dean and Ghemawat, 2008).

The algorithm begins with the choice of k, the number of Gaussians and initial Gaussian parameters. Each iteration re-estimates Gaussian parameters based on the input data. The algorithm

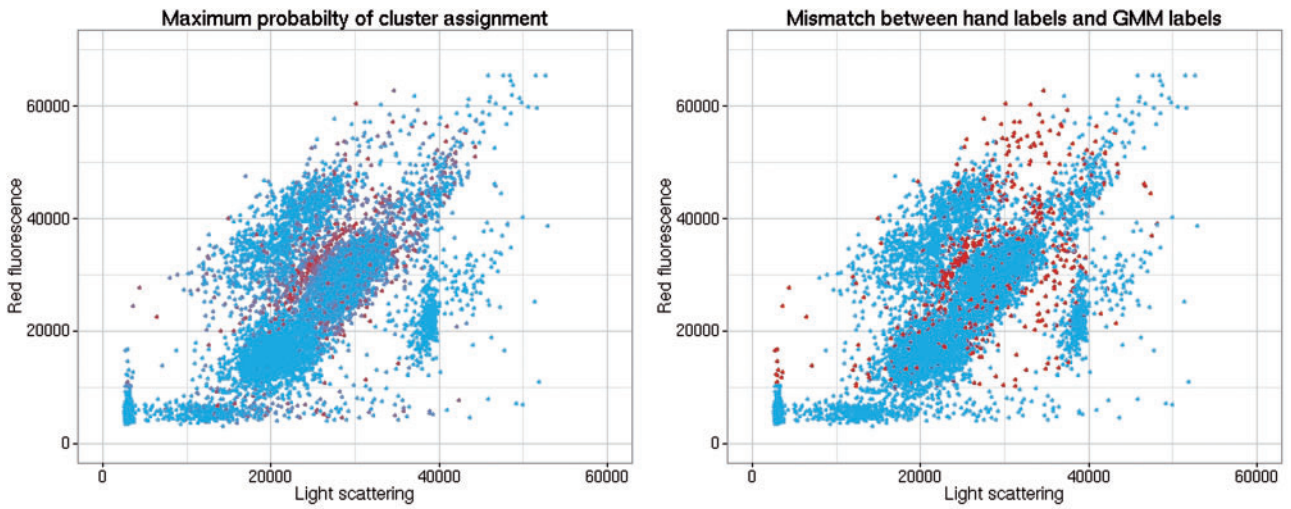

Fig. 1. A two-dimensional view of a sample of a SeaFlow dataset collected over 3 mins. GMM has been applied to find cluster labels. On the left, points are colored by the probability of their most likely label, with darker denoting a lower probability (i.e. higher uncertainty). On the right, darker points indicate points where the label from GMM did not match the manual labels. We observe a strong correlation between mismatched labels and labels that GMM deems uncertain, offering a mechanism to ignore ambiguous particles and improve the model

is repeated for several iterations (20 in our experiments) to fit the Gaussian parameters, and then each particle is assigned a label based on the Gaussian with the highest probability of producing the particle. We initialize the Gaussian means similarly to the k-means++ algorithm (Arthur and Vassilvitskii, 2007), choose the initial variances to be identity matrices and the initial Gaussian likelihoods to be $\frac{1}{k}$.

To optimize the GMM objective function, we implement expectation-maximization on Hadoop following Chu *et al.* (2007) but extended with a combiner to compute partial aggregates. For each observation o and each Gaussian k, the map phase computes the probability that k produced o, partially aggregates these results in the combiner phase, then computes the new mean and variance for each Gaussian in the reduce phase. A complete description of the algorithm implementation and the workflow of our analysis is provided in the Supplementary Material, and all code is available online (https://github.com/jhyrkas/seaflow_cluster).

3.2 Change-point detection

Each SeaFlow cruise covers a large spatial scale, spanning different environments that support different phytoplankton populations. If we analyze each window independently, we adapt to this variation but risk overfitting and overlooking global patterns. In contrast, training a single model over the entire dataset ignores the high-frequency biological variation in the ocean.

To balance the trade-off of sample-at-a-time versus all samples at once on the Seaflow dataset, we used change-point detection (James and Matteson, 2013) to identify statistically similar regimes in the data before training the GMM. Change-point detection algorithms are used for discovering points in time series data when the underlying probability distribution of the data changes. In this example, a change-point might correspond to a different population distribution. An alternative to change-point detection is to use domain expertise and human labeling of different water masses with known boundaries, but our goal was to produce a fully automated solution.

Change-point detection algorithms assume the data consists entirely of observations from a single distribution, but cytometry data contain many microbial populations sampled simultaneously and no way to know from which population a single observation is drawn. Our initial application of change-point detection tried to find change points from the measurements of individual particles instead of change points in the overall distribution of populations, but the natural variability in the data led to many spurious change points that did not reflect the physical environment. To overcome this problem, we computed the mean of all variables in each 3-min window, reducing each window to a single mean observation. While this aggregation obscures information about each sample, we found that using sample means revealed two change-points in one dataset (Cruise 2, described in Section 4.1) that correspond with well-known changes in populations calculated from the ground truth labels (Fig. 2).

Guided by this success, we run change-point detection on the sample means across the entire cruise to separate the cruise into meaningful segments, then independently cluster these segments using our MapReduce GMM method. Figure 3 illustrates the overall workflow. After the observations are ingested in CSV format, we compute the sample means for each 3-min window and these sample means are used to detect change points. The change points are then used to divide the CSV data into independent segments. Finally, a separate GMM is trained on each segment.

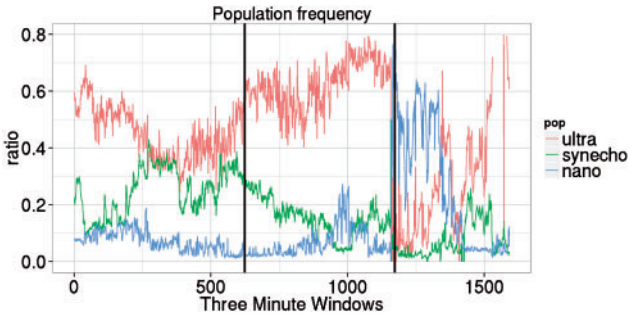

Fig. 2. The colored lines show the relative abundance of three populations for Cruise 2 going forward in time. These labels were determined by manual gating and were not available to the change-point algorithm. The vertical bars show change-points chosen by the algorithm, which seem to correspond to real biological change, although there are noticeable shifts (e.g. around the 300th and 1400th sample) that are not detected

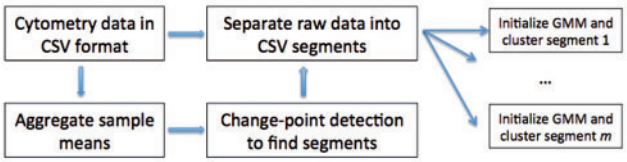

Fig. 3. General workflow for processing cytometry data after applying change-point detection. Samples means are collected and the data is segmented based on the discovered change-points. Each segment is initialized, loaded into Hadoop for clustering

4 Evaluation

We compare the baseline sample-by-sample methods against our large-scale GMM method and evaluate classification accuracy. We find that classifying all data collectively using GMM often outperforms the sample-based methods and that adding change-point detection to segment the datasets into homogeneous regions (corresponding to distinct physical regimes in ocean dynamics) improves results further.

Segments found using change-point detection may still be arbitrarily large and cannot, therefore, be classified using existing libraries such as flowMeans and flowPeaks that rely on loading all data into main memory. As an alternative to change-point detection, we evaluated the performance of flowMeans on fixed 10% subsets of one cruise at a time but found that the predictive performance was worse than using the original 3-min samples. This result illustrates that the performance improvement associated with change-point detection is not simply a result of considering larger subsets of data at a time. The details of these experiments are included in the Supplementary Materials.

4.1 Datasets

We examine three SeaFlow datasets:

- Cruise 1: This dataset consists of 867 samples (~23.6 million particles) and was continuously collected over approximately 2 days from a single location off the Washington coast.
- Cruise 2: This dataset consists of 1599 samples (~12.6 million particles) and was collected while a vessel traveled between coastal and oceanic waters in the Gulf of Alaska for approximately 3.5 days.
- Cruise 3: This dataset consists of 2802 samples (~22.6 million particles) and was collected while a vessel traveled along the Washington coast for approximately 6 days.

These datasets are made up of measurements of phytoplankton particles that SeaFlow detects in the water. The challenge is to correctly label these particles to identify populations present in the dataset. The training labels in these datasets were assigned using manual gating. For evaluation purposes, we assume these labels represent ground truth, but anecdotally we have found that our model can at times outperform the human. More information on these datasets, including information for accessing the data, can be found in the Supplementary Materials.

For each sample, we cluster particles on their optical measurements and compare the cluster labels from each algorithm with the labels provided by manual gating. We use the *F-measure* for cluster evaluation, a popular metric of cluster quality that was previously used, e.g. as the main evaluation criterion in FlowCAP (Aghaeepour *et al.*, 2013).

We evaluate classifier quality by computing the *F*-measure over every sample and averaging.

4.2 Choosing GMM parameters

When running GMM on each 3-min window (our GMM baseline), we hold a fixed k value for the entire cruise. A fixed k is not justified physically since different samples may contain different populations drawn from different environments. However, attempts to use typical methods to choose k automatically (e.g. finding k that maximizes accuracy while using the Akaike or Bayesian information criterion to penalize model complexity) (Posada and Buckley, 2004) resulted in drastic overfitting and poor performance. Instead, we choose a domain-informed estimate of k and reuse the value for the entire cruise.

Since there are potentially millions of data points in a SeaFlow dataset, broad patterns are more important than highly accurate classification of each individual particle. Given that data points in GMM with low maximum probability tend to correspond to points of low confidence in manual gating (Fig. 1), we consider removing these points to measure the effect on performance. We set a threshold of 0.7 as the minimum responsibility for a particle: If a particle has a maximum responsibility less than 0.7, it is labeled as noise and not included in our *F*-measure calculations. In our experiments, using this confidence threshold resulted in a slight boost in the average *F*-measure (on the order of 3%).

4.3 GMM over whole cruise

First, we explore clustering whole cruises using GMM, and how this compares against other automated baselines that only examine one sample at a time. Figures 4 and 5 show the cumulative density function of *F*-measures across all samples of Cruises 1 and 3, while Table 1 lists the average *F*-measures.

For these cruises, clustering all data simultaneously using GMM outperformed both the sample-by-sample method and the segmented method. Both of these cruises operated within a relatively homogenous water mass, so this result is not surprising.

flowMeans is the best baseline method, followed by flowPeaks. The ability to cluster all data in the GMM scalable algorithm provides more predictive power than either flowMeans or flowPeaks, both of which are memory constrained. GMM applied sample-by-sample is the worst-performing baseline, suggesting that that the accuracy of the Hadoop implementation is attributable to the ability to train the model using all available data rather as opposed to a fundamental improvement of the GMM method itself.

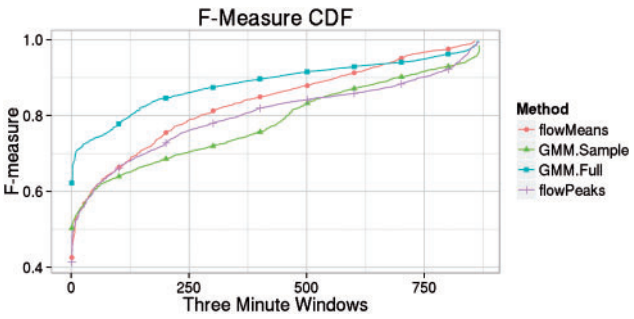

Fig. 4. Cumulative density function of *F*-measures across all samples of Cruise 1. GMM.Sample is the sample-by-sample GMM clustering previously explored. GMM.Full represents all samples classified by the Hadoop-GMM implementation. The latter implementation outperforms all baselines

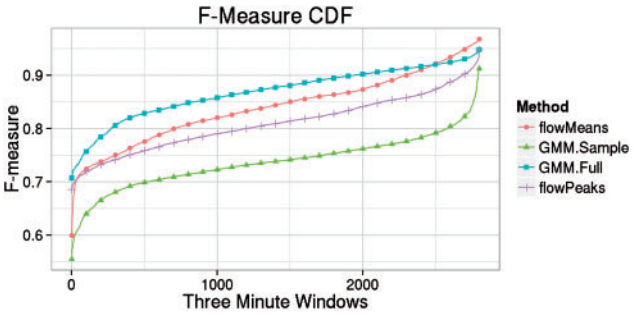

Fig. 5. Cumulative density function of *F*-measures across all samples for cruise 3. The Hadoop-GMM implementation over all samples outperforms the best baseline

Table 1. Average *F*-measures for each method for each cruise

Algorithm	Cruise 1	Cruise 2	Cruise 3
GMM full	**0.884**	0.769	**0.868**
GMM sample	0.782	**0.869**	0.737
flowMeans	0.833	0.797	0.839
flowPeaks	0.79	0.764	0.804
GMM change-point	—	**0.864**	—

Results in bold represent the best results on a given dataset. More statistics are provided in the Supplementary Materials.

4.4 Clustering subsets of a cruise

For Cruise 2, clustering the whole cruise using GMM performs worse than the sample-based baselines. This dataset was collected as a vessel moved across highly variable environmental conditions. Using change-point detection to identify homogeneous segments, performance improves to be comparable to the best baseline (GMM sample-by-sample, in this case). Figure 6 shows the cumulative density plot for this analysis, and Table 1 lists the average *F*-measures.

Figure 7 breaks down the performance of the change-point method. The two change-points found in Cruise 2 seem to correspond to actual shifts in population ratios as determined by the ground truth labels. This correspondence suggests that we see a drop in performance when clustering all samples simultaneously because the trained model ignores local variability in the environments from which the data were drawn.

We note both methods perform poorly at the end of this cruise. The difficulty comes from a short section of the third segment of the cruise, where the population density shifted rapidly due to a passage into a different oceanic environment (Palevsky *et al.*, 2013), a shift

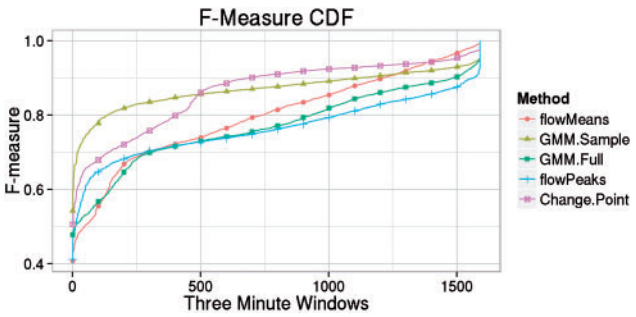

Fig. 6. Cumulative density function of *F*-measures across all samples of Cruise 2. The Hadoop-GMM implementation, augmented with change-point detection to partition the dataset, performs comparably to the best baseline

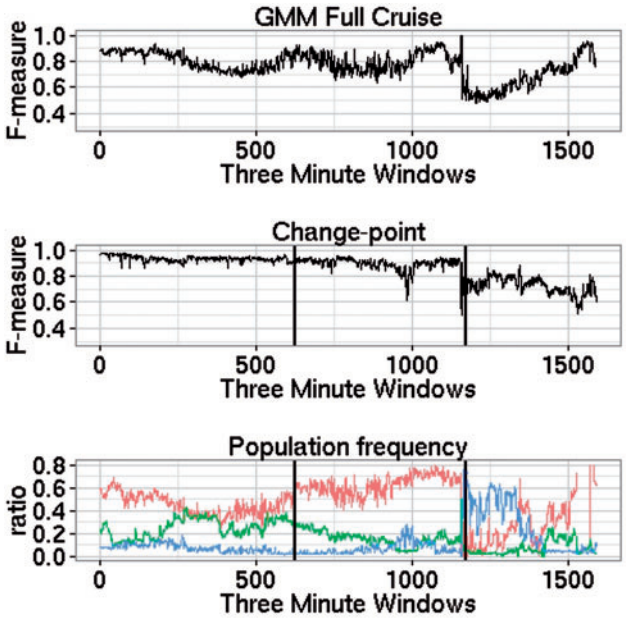

Fig. 7. The top plots show the *F*-measure of each algorithm moving forward in time during Cruise 2, and the bottom plot shows the population diversity of the three most prevalent populations. Vertical lines indicate change-points found using change-point detection. The change-points seem to correspond with changes in population diversity, and clustering these segments separately improves classification in each segment. We speculate that segmenting the end of the cruise further would lead to better classification, as there is an obvious change point near the 1400th window that goes undetected

which is detected by the change-point detection algorithm. Unfortunately, this shift is accompanied by a massive drop in the amount of data collected per 3-min window. When the populations shift again, the amount of data collected sharply increases. The second shift was undetected by our method, and as a result, the model overfits to the end of the segment. If we were able to detect this change-point and further segment the cruise, the classification accuracy in this segment would increase further.

4.5 Runtime evaluation

Our evaluation emphasizes scalability and classification accuracy rather than wall-clock runtime, but in some scenarios, runtime may be an important factor in selecting an algorithm. For the FlowCAP NDD dataset (Aghaeepour *et al.*, 2013) and Cruise 1 from the SeaFlow dataset, we ran flowMeans over each sample serially and used an eight-machine Hadoop environment to classify the full cruise dataset. We ran the experiments on Amazon Web Services using m3.large virtual machine instances. Overall, clustering individual samples using flowMeans was roughly 3–4x faster than using Hadoop to cluster the full concatenated dataset. This result is expected, as the runtime complexity of GMM can increase super-linearly in the size of the input data.

More importantly, there are various well-documented inefficiencies in the Hadoop architecture, such as the need to rescan the input data on every iteration of GMM (Bu *et al.*, 2010) and the need to write replicated data to disk after every step of the algorithm for fault tolerance that contribute to a slower runtime. However, the advantage of using Hadoop over algorithms like flowMeans is the ability to scale to datasets of arbitrary size (very long cruises, multiple cruise campaigns, etc.). Also there have been recent distributed GMM implementations in systems such as Spark that are much faster than the Hadoop implementation (Maas *et al.*, 2015). If runtime is absolutely crucial, we believe these systems could be readily used instead. Our primary contribution is in the design of the distributed approach, which is more general than the current implementation in Hadoop.

5 Discussion and future work

Aghaeepour *et al.* (2013) showed that there are automated methods capable of accurately classifying data collected by conventional flow cytometers. However, the SeaFlow cytometer represents a new class of cytometer, where data are continuously collected over a period of days or weeks as the instrument passes though different environmental conditions. Previous cytometry classification methods are not equipped to scale or variability of these datasets; they can only classify the data in small segments based on available main memory. We have improved classification accuracy by using a scalable algorithm that classifies the entire dataset or segments of the data collected from homogeneous environmental regimes.

We believe scale is a problem that could be overcome by some methods that performed well in the FlowCAP survey. *k-means* is a scalable algorithm that on its own is insufficient for cytometry classification, as populations in cytograms rarely form spherical shapes. However, both flowMeans and flowPeaks extend *k*-means by adding sophisticated initialization and cluster-merging techniques. If these extra steps could be parallelized and scaled to tens or hundreds of gigabytes of data, they might perform well on large-scale cytometry data.

Some recent algorithms, such as Dundar *et al.* (2014), classify cytometry samples individually and then look for more global patterns between samples. It is worth exploring the application on SeaFlow datasets to see how the algorithm performs on applications with thousands of samples.

The variability of environments and the corresponding changes in populations is likely the hardest problem to overcome in the SeaFlow data. Breaking up a difficult dataset into homogeneous regions using change-point detection resulted in a significant improvement in classification in one dataset. We plan to explore this method further to determine how widely it can be applied to cytometry data. Future research directions include exploring variations of change-point detection, alternative algorithms that achieve similar results and iteratively subdividing regions to find groups within groups. Change-point detection may not only provide good partitioning for better classification but also provide insight into when the underlying environments change as data are collected.

Funding

This work was supported, in part, by NSF awards IIS-1247469 (to B.H.) and OCE-1154074, a Gordon and Betty Moore Foundation grant GBMF3776 and a Simons Foundation Investigator award (all to E.V.A.) and the University of Washington eScience Institute.

Conflict of Interest: none declared.

References

Aghaeepour,N. *et al*. (2011) Rapid cell population identification in flow cytometry data. *Cytometry A*, **79**, 6–13.

Aghaeepour,N. *et al*. (2013) Critical assessment of automated flow cytometry data analysis techniques. *Nat. Methods.*, **10**, pp. 228–238.

Arthur,D. and Vassilvitskii,S. (2007) K-means++: the advantages of careful seeding. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, Society for Industrial and Applied Mathematics, pp. 1027–1035.

Bu,Y. *et al*. (2010) Haloop: efficient iterative data processing on large clusters. *Proc. VLDB Endow.*, **3**, 285–296.

Chu,C.-T. *et al*. (2007) Map-reduce for machine learning on multicore. In: *Advances in Neural Information Processing Systems*. Vol. 19, pp. 281–288.

Dean,J. and Ghemawat,S. (2008) Mapreduce: simplified data processing on large clusters. *Commun. ACM*, **51**, 107–113.

Demers,S. *et al*. (1992) Analyzing multivariate flow cytometric data in aquatic sciences. *Cytometry*, **13**, 291–298.

Dubelaar,G.B.J. *et al*. (1999) Design and first results of CytoBuoy: a wireless flow cytometer for in situ analysis of marine and fresh waters. *Cytometry*, **37**, 247–254.

Dundar,M. *et al*. (2014) A non-parametric Bayesian model for joint cell clustering and cluster matching: identification of anomalous sample phenotypes with random effects. *BMC Bioinformatics*, **15**, 314.

Field,C. *et al*. (1998) Primary production of the biosphere: integrating terrestrial and oceanic components. *Science*, **281**, 237–240.

Finak,G. *et al*. (2009) Merging mixture components for cell population identification in flow cytometry. *Adv. Bioinform.*, **2009**, e1003806.

Finak,G. *et al*. (2014) Opencyto: an open source infrastructure for scalable, robust, reproducible, and automated, end-to-end flow cytometry data analysis.

Ge,Y. and Sealfon,S.C. (2012) flowPeaks: a fast unsupervised clustering for flow cytometry data via k-means and density peak finding. *Bioinformatics*, **28**, 2052–2058.

James,N.A. and Matteson,D.S. (2013) ecp: an R package for nonparametric multiple change point analysis of multivariate data. *ArXiv e-prints*.

Kvistborg,P. *et al*. (2015) Thinking outside the gate: single-cell assessments in multiple dimensions. *Immunity*, **42**, 591–592.

Lo,K. *et al*. (2008) Automated gating of flow cytometry data via robust model-based clustering. *Cytometry A*, **73**, 321–332.

Lo,K. *et al*. (2009) flowclust: a bioconductor package for automated gating of flow cytometry data. *BMC Bioinformatics*, **10**, 145.

Maas,R. *et al*. (2015) Gaussian mixture models use-case: in-memory analysis with myria. In: Proceedings of the 3rd VLDB Workshop on In-Memory Data Mangement and Analytics, ACM, New York, NY, USA, pp. 3:1–3:8.

Olson,R. and Sosik,A. (2007) Submersible imaging-in-flow instrument to analyze nano- and microplankton: imaging FlowCytobot. *Limnol Oceanogr Methods*.

Palevsky,H.I. *et al*. (2013) The influence of net community production and phytoplankton community structure on co2 uptake in the gulf of alaska. *Global Biogeochem. Cycles*, **27**, 664–676.

Posada,D. and Buckley,T.R. (2004) Model selection and model averaging in phylogenetics: advantages of akaike information criterion and Bayesian approaches over likelihood ratio tests. *Syst. Biol.*, **53**, 793–808.

Shvachko,K. *et al*. (2010) The hadoop distributed file system. In: *Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, MSST '10, Institute of Electrical and Electronics Engineers, pp. 1–10.

Sosik,H.M. *et al*. (2010) Flow cytometry in phytoplankton research. In: Suggett,D.J. *et al*. (eds), *Chlorophyll A Fluorescence in Aquatic Sciences: Methods and Applications* Springer, Netherlands, pp. 171–185.

Swalwell,J. *et al*. (2011) SeaFlow: a novel underway flow-cytometer for continuous observations of phytoplankton in the ocean. *Limnol. Oceanogr. Methods*, **9**, 466–477.

Tarnok,A. *et al*. (2001) Rapid screening of possible cytotoxic effects of particulate air pollutants by measurement of changes in cytoplasmic free calcium, cytosolic ph, and plasma membrane potential in alveolar macrophages by flow cytometry. *Cytometry*, **43**, 204–210.

Zaharia,M. *et al*. (2010) Spark: cluster computing with working sets. In: *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, USENIX, pp. 15–28.