

cyanoFilter: An R package to identify phytoplankton populations from flow cytometry data using cell pigmentation and granularity

Oluwafemi D. Olusoji^{a,b,*}, Jurg W. Spaak^b, Mark Holmes^b, Thomas Neyens^a, Marc Aerts^a, Frederik De Laender^b

^a Center for Statistics, Data Science Institute, Hasselt University, Hasselt, 3500, Belgium

^b Research Unit in Environmental and Evolutionary Biology (URBE), Institute of Life-Earth-Environment (ILEE), Namur Institute for Complex Systems (NAXYS), Université de Namur, Namur, 5020, Belgium

ARTICLE INFO

Dataset link: <https://datadryad.org/stash/share/Urm9nRyJMBvHt8tbg4nWTZHY5bzDSLZ2Hftc5DmlSjw>

Keywords:

Flow cytometry
Phytoplankton
Gating
Software
Ecology

ABSTRACT

Flow cytometry is often employed in ecology to measure traits and population size of bacteria and phytoplankton. This technique allows measuring millions of particles in a relatively small amount of time. However, distinguishing between different populations is not a straightforward task. Gating is a process in the identification of particles measured in flow cytometry. Gates can either be created manually using known characteristics of these particles, or by using automated clustering techniques. Available automated techniques implemented in statistical packages for flow cytometry are primarily developed for medicinal applications, while only two exist for phytoplankton. cyanoFilter is an R package built to identify phytoplankton populations from flow cytometry data. The package also integrates gating functions from two other automated algorithms. It also provides a gating accuracy test function that can be used to determine the accuracy of a desired gating function if monoculture flowcytometry data is available. The central algorithm in the package exploits observed pigmentation and granularity of phytoplankton cells. We demonstrate how its performance depends on strain similarity, using a model system of six cyanobacteria strains. Using the same system, we compare the performance of the central gating function in the package to similar functions in other packages.

1. Introduction

Phytoplankton are primary producers, responsible for about 50% of global primary production. Studies seeking a better understanding of the ecology of phytoplankton have relied on flow cytometry (FCM) to measure phytoplankton abundance and traits (Pomati et al., 2013; Stomp et al., 2007; Fontana et al., 2018). Although there are earlier applications of FCM in microbial ecology (Trask et al., 1982; Berglund and Eversman, 1988), the possibility of obtaining individual-level data through FCM to enhance the understanding of community assembly and biodiversity is increasingly exploited (Fontana et al., 2014; Pereira et al., 2019; Dunker et al., 2018).

FCM is a technique which involves the suspension of cells or particles within a fluid stream which is made to pass through one or more laser beams (O'Neill et al., 2013). A crucial step in FCM application is to separate signal from noise, e.g. phytoplankton cells from other particles. Gating, a process of using fluorescence intensity to sequentially separate histograms or scatter plots obtained from plotting FCM data into regions, is the classical approach to meet this challenge. Measured particles are classified based on their properties,

most notably scatter and fluorescence signals. This process can be done manually, termed manual gating, or with automated algorithms, termed automated gating (O'Neill et al., 2013).

Gating is typically done manually with software such as *FCS Express* (Software, 2021), *FloMax* (Sysmex, 2021), *flowJo* (Becton and Company, 2021), *Incyte* (Luminex Corporation, 2021), or using model-based tools such as *flowClust* (Lo et al., 2009), *flowMeans* (Aghaeepour et al., 2010) and *flowEMMi* (Ludwig et al., 2019) or machine learning (Fontana et al., 2018) or tree based algorithm (Commenges et al., 2018) or deep learning combined with image-based cytometry (Dunker et al., 2018). While manual gating is often not fully reproducible, model-based tools, machine and deep learning often require tuning of global parameters that are not related to the biological properties of the measured cells, and cannot readily integrate experts' knowledge (Malek et al., 2015). Moreover, most of these packages were developed and optimized for medical FCM data. Currently, there are two known software packages that offer automated gating for phytoplankton; *flowPhyto* (Ribalet et al., 2011) and *flowEMMi* (Ludwig et al., 2019). *flowPhyto*, which uses *flowClust* (Lo et al., 2009) for gating

* Corresponding author at: Center for Statistics, Data Science Institute, Hasselt University, Hasselt, 3500, Belgium.
E-mail address: oluwafemi.olusoji@uhasselt.be (O.D. Olusoji).

phytoplankton communities contained in geo-referenced seawater, is primarily built for seaFlow (Swalwell et al., 2009), a flow cytometer that is different from traditional flow cytometers. *flowEmmi* uses the Expectation–Maximization (EM) algorithm on two channels to gate microbial communities contained in flow cytometry data (Ludwig et al., 2019). The package requires that manual pre-processing be carried out to remove debris prior to applying the gating functions provided by *flowEMMi*. Moreover, the authors of the package assume that the samples are treated with a fluorescent dye to separate the cells from noise and debris. This assumption does not hold for phytoplankton since these cells are photosynthetic, and they absorb and reflect light naturally (Kirk, 1994; Stomp et al., 2004).

We therefore introduce a new R package *cyanoFilter* for semi-automated gating of phytoplankton FCM data. The central gating function in the package, *phytoFilter*, uses pigment and granularity information to identify phytoplankton populations contained in a sample. Aside from identifying phytoplankton populations, *cyanoFilter* can also assist in the identification of previously unknown fluorescence and scatter channels useful for differentiating phytoplankton cells. It also imports gating functions from two other R packages (*flowClust* and *cytometree*). The package also contains functions for pre-processing steps that might be needed before the gating. We demonstrate the performance of the package when compared with two other packages; *flowClust* and *cytometree*, using six strains of the cyanobacteria genus *Synechococcus*, which represents a common component of phytoplankton communities worldwide (Petr et al., 2004), and is one of the oldest life forms known to perform photosynthesis (Kirk, 1994). In addition, the strains from this genus harbor a rich diversity of pigmentation phenotypes (Stomp et al., 2007, 2004; Haverkamp et al., 2008), allowing us to manipulate pigmentation similarity and evaluate the resulting performance of our package. Doing so is important, as distinguishing populations becomes increasingly difficult as pigmentation becomes more similar among populations.

2. Materials and methods

The purpose of *cyanoFilter* is to provide a user-friendly toolbox that uses pigmentation and/or cell granularity, reflecting the structure of internal organelles, to identify phytoplankton populations contained in a sample.

We categorize each particle measured by FCM into one of four outcomes: debris, margin events, multiplets and singlets. Debris are non-living particles and therefore often not of interest. Margin events are particles with widths outside the range that is measurable by the flow cytometer. As the dimensions of these particles are not known, these are also typically excluded from analyses. Multiplets are two or more cells clustered together, which result in disproportionate readings of the height–width relationship. If the cells being measured are not of doublet morphology (Meeks and Elhai, 2002), then doublets or multiplets can be avoided during the acquisition phase of FCM. Finally, singlets are those particles that are often the focus of interest and these are further categorized into different populations. The process of properly identifying these outcomes requires the knowledge of both the flow cytometer light scatter channels and properties of the measured cells.

The *cyanoFilter* framework consists of a pipeline of analyses that start from pre-processing the raw FCM output from the flow cytometer to the last stage of producing flowframes (R object name for FCM data) with labels for each population of phytoplankton identified in the FCM data (Fig. 1).

The central algorithm in *cyanoFilter* identifies peaks along all the supplied channels. It does so by fitting one dimensional kernel densities to each channel. The procedure defines specific channel values that separate the identified peaks. Also, the procedure disregards peaks with a height below a user-defined cut-off. More details about this approach to gating can be found in Malek et al. (2015).

The entire gating algorithm is carried out in the following steps;

1. Search for peaks along each light channel, measuring cell pigmentation and granularity.
2. Per channel, identify the separation point between the peaks observed in step 1.
3. Per channel, divide particles into clusters based on the separation point identified in 2 and label each cluster.
4. Consider each channel as a factor, and all cluster labels as factor levels. Generate all possible factorial combinations.
5. Count the number of particles in each of these combinations.
6. Rank clusters in descending order according to proportions of particles they represent, and retain clusters with decreasing rank until a user-defined minimum proportion is achieved. We set this to a default of 80% which implies that clusters containing 80% of the measured particles are returned.

Steps four and five allow grouping cells with similar scattered light profiles across the channels considered. For example, cells with low scattered light across all pigment channels will be clustered together, thereby ending up in the same cluster. This algorithm makes up most of the last three steps in the analysis pipeline described in Fig. 1.

2.1. Implementation

cyanoFilter is written entirely in R (R Core Team, 2021), and offers functions to process standard flow cytometer data files, often referred to as FCS files, but they are called *flowFrames* within R. *flowFrames* contain matrices of channels containing data about light scatter measurement from each cell. Rows in the matrices represent particles while columns are detection channels. *cyanoFilter* also contains functions to visualize cluster phytoplankton populations contained in a FCS file 1. All clustering functions have summary and plot methods that can be accessed with the usual `plot()` and `summary()` functions in R.

2.1.1. Installation

The *cyanoFilter* R package is available on bioconductor (Huber et al., 2015a) and its latest development version on GitHub at <https://github.com/fomotis/cyanoFilter>. Note that the package requires at least R version 4.1.0, the current release of R, for it to be successfully installed remotely. Also, we provide a source file for it to be installed locally. More details on this in the appendix C1. The package and its bioconductor dependencies; Biobase (Huber et al., 2015b), flowCore (Ellis et al., 2019), and flowDensity (Malek et al., 2015) can be installed and loaded using the following commands:

```
install.packages("BiocManager")
library(BiocManager)
install(c("flowCore", "flowDensity", "Biobase",
"cyanoFilter"))

## installing locally
install.packages(path_to_cyanoFilter_source, repos
= NULL,
type="source")

## installation from github (requires R 4.1.0)
install.packages("remotes")
remotes::install_github("https://github.com/
fomotis/cyanoFilter")
library(cyanoFilter)
```

2.2. Example workflow

To illustrate the use of the package, we applied FCM to six strains of *Synechococcus* cyanobacteria (WH8101, RS9906, BL10, A15-46, 10CR4-3, ROS8604), obtained from the Roscoff Culture Collection (RCC, roscoff-culture-collection.org) and grown in 12H/12H day/night cycles at 20 °C. The strains were chosen to represent a variety of pigmentation types (Six et al., 2007). Pigment classification is based on

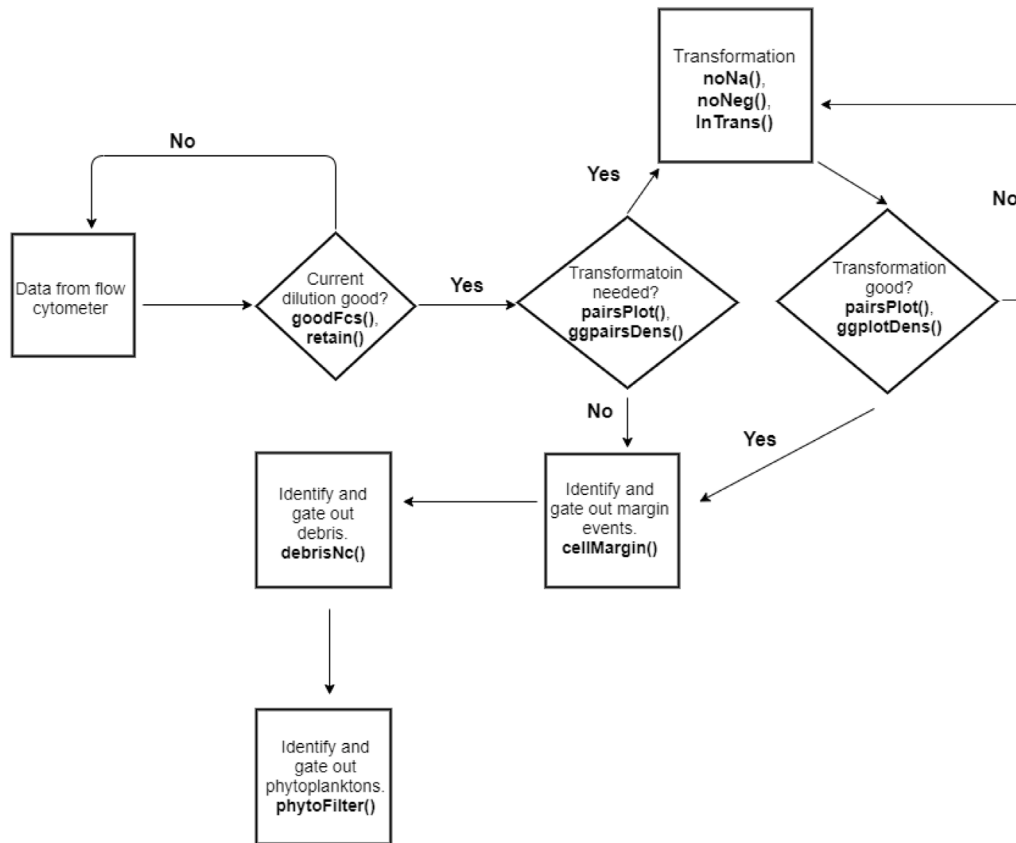


Fig. 1. The cyanoFilter framework. Squares represent data sources or actions to be taken. Rhombuses represent questions posed at certain points in the process. The functions within the package that help perform the required tasks at each level are in boldface.

Table 1
Main functions in *cyanoFilter*.

Function	Description
<code>accTest()</code>	returns accuracy of a gating function based on results from gating supplied monoculture FCM measurements
<code>cellMargin()</code>	removes margin events from supplied flowFrame file based on Side Scatter Width (SSC.W) by default
<code>complexityGate()</code>	calls the gating algorithm on the cell granularity channels
<code>clusterExtract()</code>	extracts specific cluster of interest based on supplied label
<code>debrisNc()</code>	filters out debris based on a channel measuring the presence of chlorophyll <i>a</i>
<code>fullFlowframe()</code>	method for extracting the exact flowFrames supplied for clustering
<code>gateFunc()</code>	general gating function that can execute <code>cytometree()</code> , <code>flowClust()</code> or <code>phytoFilter()</code> depending on users choice.
<code>goodfcs()</code>	identifies FCM data measured at the appropriate dilution using supplied desired minimum and maximum <i>cell/μl</i> . Useful for dilution series measurements.
<code>ggpairsDens()</code>	scatter plot matrix of measured cells along the supplied channels
<code>ggplotDens()</code>	scatter plot of measured cells along two channels
<code>pigmentGate()</code>	calls the gating algorithm on the pigment channels
<code>phytoFilter()</code>	performs the gating by calling <code>pigmentGate()</code> and <code>complexityGate()</code> functions.
<code>reducedFlowframe()</code>	method for extracting the flowframes containing clusters only
<code>summaries()</code>	method to provide basic statistical summaries of <i>cyanoFilter</i> objects

the phycobiliprotein forms: type 1 strains have only phycocyanin (PC); type 2 strains have PC and phycoerythrin I (PEI); and type 3 have PC, PEI and phycoerythrin II (PEII). Two strains of each pigment type were used: WH8101 and RS9906 are type 1, BL_10 and A15-46 are type 2, and 10CR4-3 and ROS8604 are type 3 (see appendix B). Each strain was characterized in monoculture using FCM, measuring the presence of chlorophyll *a* (488 nm excitation, 695 nm fluorescence), phycoerythrin

(488 nm excitation, 583 nm fluorescence), and phycocyanin (642 nm excitation, 664 nm fluorescence), as well as cell size (forward scatter) and cell granularity (side scatter). All strains were cultured under light intensity 20 $\mu\text{E ms}^{-2}\text{s}^{-1}$ using PCR S-11 Red Sea Medium (Rippka et al., 2000).

We employed a Guavava easycyte 3L flowcytometer (<https://www.luminexcorp.com/guava-easycyte-flow-cytometers/#overview>) to

measure these cyanobacteria samples. It uses two lasers, one red (642 nm) and one blue (488 nm). We measured 5000 objects per sample, with fluorescence detection 642 nm excitation, 664 nm emission.

2.3. Gating accuracy

To evaluate the performance of *cyanoFilter*, we compare how it correctly identifies six strains of cyanobacteria to the three other automated gating packages; *cytometree*, *flowClust* and *flowMeans*. We choose these three packages because they allow for gating with multiple channels and do not involve the tuning of global parameters. We employed the following scheme. First, we gate each of the monoculture *flowFrames* using the four softwares to label every particle to the correct strain (i.e. population). Second, we formulate all possible unique pairs of the cyanobacteria strains by joining the *flowFrames* of each pair together. We rerun each function on the joined *flowFrames*. Optimal performance of *cytometree()*, *flowClust()*, *flowMeans()* and *phytoFilter()* would lead to correct grouping of the two populations based on the joined *flowFrames*. More specifically, if these packages group two cells as similar in the monoculture, they should also identify those cells as similar when analyzing the joined monoculture *flowframes*, and vice versa. We tested this by randomly sampling pairs of particles and examining if their categorization as belonging to ‘similar’ or ‘different’ groups was the same when based on the joined *flowFrames* as in the monoculture *flowFrames*. If this was true, then we recorded this sample as a success, and a failure otherwise. We repeated this random sampling, and subsequent assessment, ten thousand times, enabling us to compute the accuracy of these gating functions. We define accuracy as the number of successes per ten thousand. To the best of our knowledge, such comparison has not been carried out in phytoplankton flow cytometry literature. The closest to this was carried out by Ludwig et al. (2019), who compared the number of clusters detected across different software when these softwares were applied to gate mock microbial communities.

3. Results

3.1. Metadata pre-processing

The first step in the analysis pipeline (Fig. 1) is the identification of the correct dilution level. For flow cytometers without the ability to perform sample dilution before acquisition, samples are measured using dilution series and each series is measured using the flow cytometer. Thus, every sample has measurements at different dilution levels. It is important to identify which of these measurements is appropriate. To do this, one determines the cell concentration that allows the flow cytometer to properly characterize each cell it measures. Manufacturers of flow cytometers provide the range of values for cell concentrations that is considered acceptable. This range can be supplied to *goodfcs()* in *cyanoFilter* (section C1, appendix) to check if the sample dilution yielded an acceptable concentration in cells/ μl .

3.2. Transformation and visualization

After validating that the concentration of the samples is within the specified range, we proceed to visualize the measured light scatter channels to examine the need for transformations using the *ggpairsDens()* in *cyanoFilter* (fig D1, appendix). The choice of a ‘good transformation’ or the optimal transformation for a FCM data can be made via visual inspection or the use of the *flowTrans* R package (Finak et al., 2021). The package comes bundled with a function for natural logarithm transformation, but others such as *arcsinh*, linear and double exponential transformations are made available in the *flowCore* R package. Here, the transformation does improve detection and visualization of populations in the dataset.

3.3. Gating margin events

The next step in the pipeline is to remove margin events. Margin events are normally stacked up close to maximum or minimum measurable width of the flow cytometer. This is visible when one plots the channel measuring cell width. To identify and remove them in the *flowFrames*, we supplied the channel (column) measuring cell width, *NIR.R.W* in case of our flow cytometer, to the *cellMargin()* function. This function uses the first three steps in the *cyanoFilter* gating algorithm listed above to separate margin events from the properly measured cells. The resulting object of class *MarginEvents* has a plot and a summary method which produce a two-dimensional plot of the measured particles along the supplied channel, basic summary statistics, the estimated cut point and the number of margin and nonmargin events.(section C4, D2, appendix).

3.4. Gating debris

To identify and remove debris, we used the channel (column) which measures the presence of chlorophyll *a*, in the *debrisNc()* function. Cells with lower levels of chlorophyll *a* are either dead cells or non-photosynthetic particles, and should cluster around or close to 0. Therefore, we expected two peaks along the chlorophyll *a* channel. However, this might not be evident if the number of debris particles is very small (fig D3, appendix). In these cases, we calibrated the function to use two standard deviations away from the center of the peak detected as a cut-point. Cells with chlorophyll *a* levels below this cut-point are categorized as debris. Users have the option to change this if required. The *debrisNc()* function also comes with a *plot()* and *summary()* method (section C5, appendix).

3.5. Gating cyanobacteria strains

To identify the cyanobacteria strains using the *phytoFilter()* function, we supplied channels (columns) measuring the presence of chlorophyll *a*, phycoerythrin and phycocyanin as pigment channels. We consider cell size as an aspect of cell granularity, thus we also supplied the channels measuring cell size and granularity (section C6, appendix). The resulting object also has a *plot()* (Fig. 2 a) and *summary()* method that produces matrix scatter plots and summary information about the gating.

3.6. Performance

The performance of *cyanoFilter*, *cytometree* and *flowMeans* increases with increasing distance between the multivariate depths of the strains, as well as increasing distance between the RGB color vectors (see Figs. 2b and 2c). The accuracy of the fore-mentioned software also decreases below 60% when these strains are close in terms of color and depth. *flowClust* has an accuracy above 70% when the strains are close in terms of the two characteristics (color and multivariate depths), and its performance is quite stable across the range of distance observed. The performance of *flowClust* also increases with increasing distance between the depth and RGB color vectors of these strains. While *cyanoFilter*, *cytometree* and *flowMeans* outperform *flowClust* at certain distances between the depths of these strains, increasing distance does not always lead to increasing accuracy (see figure D4 in section D of the appendix). Moreover, *flowClust* is also performing quite well in these scenarios (see Figs. 2b and 2c). *flowMeans* only produced results for some strain combinations. The package fails for about 40% of the strain combinations we considered.

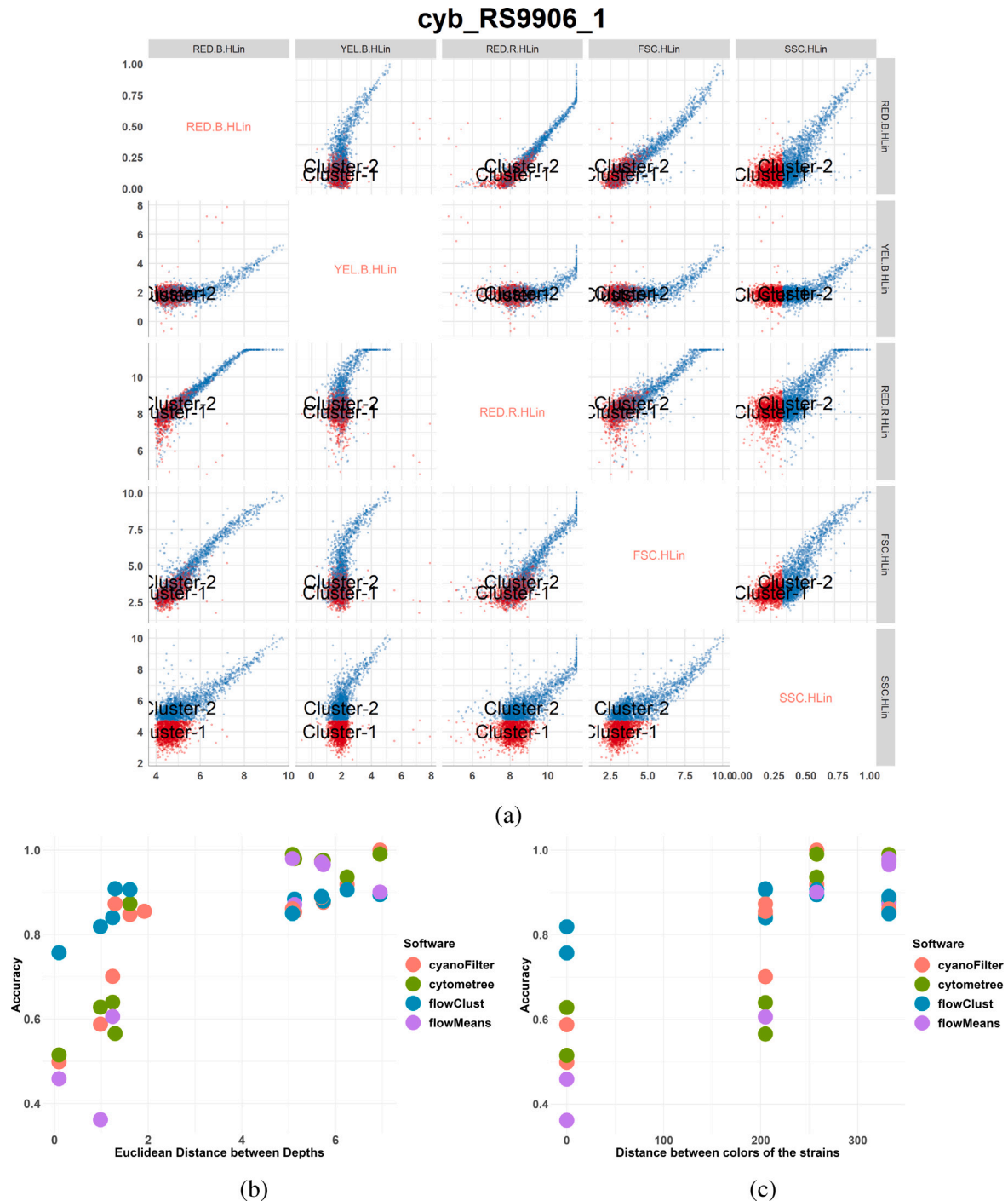


Fig. 2. (a) Matrix scatter plot showing the result from gating one of the FCM files. The algorithm detected two clusters (cluster-1 and cluster-2). Every point in this scatter plot is a cell. The figure shows that only one of the channels, SSC.HLin (cell granularity), had multiple peaks. (b) Accuracy against the Euclidean distance between the multivariate depth (Struyf and Rousseeuw, 2000) of the pigment and granularity channels. (c) Accuracy against the Euclidean distance between the RGB color code of the strains used in the FCM experiment.

4. Discussion & summary

From the accuracy test for the particular experiment we performed, we observed that *cyanoFilter* performs moderately when the strains have similar color and multivariate depths. Indeed, we expected the accuracy of the algorithm to increase with increasing distance between the multivariate depths since it relies on the minimum intersection between peaks to be able to divide cells into groups, and the multivariate depths themselves are a measure of location in a multidimensional space. Thus, closer peaks would imply closer depths, and these are

harder to distinguish. Therefore, we expect that gating strains that absorb light around similar peaks will be challenging for *cyanoFilter*. Our accuracy test also shows that this is the case for *cytometree* and *flowMeans*.

It is important to carefully consider the possible effects of chromatic adaptation, a process where levels of biliproteins responsible for absorbing specific light color are controlled by the spectral quality of the light the phytoplankton is exposed to Stomp et al. (2004), Kehoe (2010), since *cyanoFilter* uses pigments and cell granularity to gate phytoplankton species. If chromatic adaptation leads to a change in the

pigmentation of phytoplankton species such that it absorbs light along similar spectra as another species, then there is a chance of reduced accuracy for *cyanoFilter* if there are no other light scatter channels that can differentiate these species. The drop in accuracy when the strains share similar color and close multivariate depth along light scatter channels supports this. However, this problem is not unique to *cyanoFilter*. Other packages will suffer a similar reduction in accuracy if chromatic adaptation occurs such that no channel offers additional information that can help differentiate the cells.

cyanoFilter offers a suite of functions that helps with pre-processing of FCM files and identification of phytoplankton populations therein. It can cluster phytoplankton in FCM experiments with the advantage of easy integration of knowledge about the cells via the light scatter channels measured. For example, we used information about chlorophyll *a* to differentiate debris from phytoplankton cells. Users can either base this identification on user-defined channels, or let the algorithm identify which channels provide the best clustering of populations in a sample. While the example uses 6 cyanobacteria strains, the package is applicable to FCM measurements of phytoplankton cells that scatter specific wavelengths of light, since such light can be used to estimate their size and their photosynthetic pigments. Thus, channels measuring such light scatter can be used to identify and distinguish these phytoplankton cells (Ribalet et al., 2011). The package also incorporates gating functions from *cytometree* and *flowClust*. Furthermore, users can use the accuracy test function provided within this package to determine which gating function is best suited for their experiment. This function requires the availability of monoculture FCM experiments. With this package, we hope to contribute to making analyses of phytoplankton FCM data easier and more reproducible.

CRedit authorship contribution statement

Oluwafemi D. Olusoji: Wrote the manuscript. Developed the algorithm and wrote the corresponding R codes. **Jurg W. Spaak:** Read and edited the manuscript, Developed the algorithm and wrote the corresponding R codes. **Mark Holmes:** Performed the experiment generating the data for the example. **Thomas Neyens:** Read and edited the manuscript, Provided the resampling framework for testing the accuracy of the softwares. **Marc Aerts:** Read and edited the manuscript, Provided the resampling framework for testing the accuracy of the softwares. **Frederik De Laender:** Read and edited the manuscript, Provided the resampling framework for testing the accuracy of the softwares.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data made available via Dryad, <https://datadryad.org/stash/share/Urm9nRyJMBvHt8tbg4nWTZHY5bzDSLZ2Hftc5DmlSjw> digital repository.

Acknowledgments

Many thanks to several productive discussions with Jurgen Claesen on several aspects of the package.

Funding

This work was supported by funds from the bilateral co-operation (BOF) between Hasselt University and Université de Namur. F.D.L. was supported by the ARC, Belgium grant DIVERCE, a concerted research action from the special research fund (Convention no 18/23-095), and by the Fund for Scientific Research, FNRS, Belgium (PDR T.0048.16).

Software availability

Source code is freely available on GitHub <https://github.com/fomotis/cyanoFilter>, and is also available on <https://bioconductor.org>.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.ecolmodel.2021.109743>.

References

- Aghaeepour, N., Nikolic, R., Hoos, H.H., Brinkman, R.R., 2010. Rapid cell population identification in flow cytometry data. *Cytom. Part A* 79A (1), 6–13.
- Becton, D., Company, 2021. FlowJo, LLC. <https://www.flowjo.com>. (Accessed: 10/08/2021).
- Berglund, D.L., Eversman, S., 1988. Flow cytometric measurement of pollutant stresses on algal cells. *Cytometry* 9 (2), 150–155.
- Commenges, D., Alkassim, C., Gottardo, R., Hejblum, B., Thiébaud, R., 2018. cytometree: A binary tree algorithm for automatic gating in cytometry analysis. *Cytom. Part A* 93 (11), 1132–1140.
- Dunker, S., Boho, D., Wäldchen, J., Mäder, P., 2018. Combining high-throughput imaging flow cytometry and deep learning for efficient species and life-cycle stage identification of phytoplankton. *BMC Ecol.* 18 (1), 51.
- Ellis, B., Haaland, P., Hahne, F., Le Meur, N., Gopalakrishnan, N., Spidlen, J., Jiang, M., Finak, G., 2019. flowCore: flowCore: Basic structures for flow cytometry data, R package version 1.50.0.
- Finak, G., Manuel-Perez, J., Gottardo, R., 2021. flowTrans: Parameter optimization for flow cytometry data transformation, R package version 1.44.0.
- Fontana, S., Jokela, J., Pomati, F., 2014. Opportunities and challenges in deriving phytoplankton diversity measures from individual trait-based data obtained by scanning flow-cytometry. *Front. Microbiol.* 5 (324).
- Fontana, S., Mridul, K., Moldoveanu, M., Spaak, P., Pomati, F., 2018. Individual-level trait diversity predicts phytoplankton community properties better than species richness or evenness. *ISME J.* (12, 2), 356–366.
- Haverkamp, T.H.A., Acinas, S.G., Doeleman, M.W., Stomp, M., Huisman, J., Stal, L.J., 2008. Diversity and phylogeny of Baltic Sea picocyanobacteria inferred from their *its* and phycobiliprotein operons. *Environ. Microbiol.* 10 (1), 174–188, 2007.
- Huber, W., Carey, V.J., Gentleman, R., Anders, S., Carlson, M., Carvalho, B.S., Bravo, H.C., Davis, S., Gatto, L., Girke, T., Gottardo, R., Hahne, F., Hansen, K.D., Irizarry, R.A., Lawrence, M., Love, M.I., MacDonald, J., Obenchain, V., Ole's, A.K., Pag'es, H., Reyes, A., Shannon, P., Smyth, G.K., Tenenbaum, D., Waldron, L., Morgan, M., 2015a. Orchestrating high-throughput genomic analysis with Bioconductor. *Nature Methods* 12 (2), 115–121, URL <http://www.nature.com/nmeth/journal/v12/n2/full/nmeth.3252.html>.
- Huber, W., Carey, V.J., Gentleman, R., Anders, S., Carlson, M., Carvalho, B.S., Bravo, H.C., Davis, S., Gatto, L., Girke, T., Gottardo, R., Hahne, F., Hansen, K.D., Irizarry, R.A., Lawrence, M., Love, M.I., MacDonald, J., Obenchain, V., Ole's, A.K., Pag'es, H., Reyes, A., Shannon, P., Smyth, G.K., Tenenbaum, D., Waldron, L., Morgan, M., 2015b. Orchestrating high-throughput genomic analysis with Bioconductor. *Nature Methods* 12 (2), 115–121, URL <http://www.nature.com/nmeth/journal/v12/n2/full/nmeth.3252.html>.
- Kehoe, D.M., 2010. Chromatic adaptation and the evolution of light color sensing in cyanobacteria. *Proc. Natl. Acad. Sci.* 107 (20), 9029–9030.
- Kirk, J., 1994. *Light and Photosynthesis in Aquatic Ecosystems*. Cambridge Univ. Press, Cambridge, England.
- Lo, K., Hahne, F., Brinkman, R.R., Gottardo, R., 2009. flowClust: A Bioconductor package for automated gating of flow cytometry data. *BMC Bioinformatics* 10 (145).
- Ludwig, J., zu Siederdisen, C.H., Liu, Z., Stadler, P.F., Müller, S., 2019. flowEMMI: An automated model-based clustering tool for microbial cytometric data. *BMC Bioinformatics* 20 (1), 1–17.
- Luminex Corporation, 2021. Guava® easyCyte™ Flow Cytometer. <https://www.luminexcorp.com/guava-easycyte-flow-cytometers/#software>. (Accessed: 10/08/2021).
- Malek, M., Taghiyar, M.J., Chong, L., Finak, G., Gottardo, R., Brinkman, R.R., 2015. FlowDensity: Reproducing manual gating of flow cytometry data by automated density-based cell population identification. *Bioinformatics* 31 (4), 606–607.
- Meeks, J.C., Elhai, J., 2002. Regulation of cellular differentiation in filamentous cyanobacteria in free-living and plant-associated symbiotic growth states. *Microbiol. Mol. Biol. Rev.* 66 (1), 94–121.
- O'Neill, K., Aghaeepour, N., Špidlen, J., Brinkman, R., 2013. Flow cytometry bioinformatics. *PLoS Comput. Biol.* 9 (12).
- Pereira, G.C., Andrade, L.P., Espíndola, R.P., Ebecken, N.F.F., 2019. Ecological networks simulation by fuzzy ecotoxicological rules. *Ecol. Model.* 409, 108733.
- Petr, D., Dale, A.C., Aloisie, P., Petr, H., Vladan, O., Remo, S., 2004. *Synechococcus*: 3 billion years of global dominance. *Mol. Biol.* (23), 5538–5551.

- Pomati, F., Kraft, N.J., Posch, T., Eugster, B., Jokela, J., Ibelings, B.W., 2013. Individual cell based traits obtained by scanning flow-cytometry show selection by biotic and abiotic environmental factors during a phytoplankton spring bloom. *PLoS ONE* 8 (8).
- R Core Team, 2021. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, <https://www.R-project.org/>.
- Ribalet, F., Schruth, D.M., Armbrust, E.V., 2011. flowPhyto: Enabling automated analysis of microscopic algae from continuous flow cytometric data. *Bioinformatics* 27 (5), 732–733.
- Rippka, R., Coursin, T., Hess, W., Lichtlé, C., Scanlan, D.J., Palinska, K.A., Iteman, I., Partensky, F., Houmard, J., Herdman, M., 2000. *Prochlorococcus marinus* Chisholm et al. 1992 subsp. *pastoris* subsp. nov. strain PCC 9511, the first axenic chlorophyll a2/b2-containing cyanobacterium (Oxyphotobacteria). *Int. J. Syst. Evol. Microbiol.* 50 (5), 1833–1847.
- Six, C., Thomas, J.-C., Garczarek, L., Ostrowski, M., Dufresne, A., Blot, N., Scanlan, D.J., Partensky, F., 2007. Diversity and evolution of phycobilisomes in marine *Synechococcus* spp.: A comparative genomics study. *GenomeBiology.Com* 8 (12), R259.
- Software, D.N., 2021. FCS Express Flow Cytometry Software. <https://denovosoftware.com/>. (Accessed: 10/08/2021).
- Stomp, M., Huisman, J., de Jongh, F., Veraart, A.J., Gerla, D., Rijkeboer, M., Ibelings, B.W., Wollenzien, U.I.A., Stal, L.J., 2004. Adaptive divergence in pigment composition promotes phytoplankton biodiversity. *Nature* 432 (7013), 104–107.
- Stomp, M., Huisman, J., Vörös, L., Pick, F.R., Laamanen, M., Haverkamp, T., Stal, L.J., 2007. Colourful coexistence of red and green picocyanobacteria in lakes and seas. *Ecol. Lett.* 10 (4), 290–298.
- Struyf, A., Rousseeuw, P.J., 2000. High-dimensional computation of the deepest location. *Comput. Statist. Data Anal.* 34 (4), 415–426.
- Swalwell, J.E., Petersen, T.W., van den Engh, G., 2009. Virtual-core flow cytometry. *Cytom. Part A* 75A (11), 960–965.
- Sysmex, 2021. CyFlow Space. <https://www.sysmex-europe.com/n/products/products-detail/cyflow-space.html>. (Accessed: 10/08/2021).
- Trask, B.J., van den Engh, G.J., Elgershuizen, J., 1982. Analysis of phytoplankton by flow cytometry. *Cytometry* 2 (4), 258–264.