Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

cyanoFilter: An R Package for Identifying Synechococcus Population Obtained via Flow Cytometry

Oluwafemi D. Olusoji
Universiteit Hasselt

Frederik De Laender
Universite De Namur

Jurg Spaak
Universite De Namur

Thomas Neyens
Universiteit Hasselt

Marc Aerts
Universiteit Hasselt

Abstract

Flow cytometry is a technique for identifying cell populations and it offers an energy efficient alternative to microscopy. Synechococcus is a prokaryotic cyanobacterium found in marine, coasts and fresh water habitats. Cyanobacteria are bacteria phylum believed to contribute more than 50% of atmospheric oxygen via photosynthesis. While there are existing studies showing the applications of flow cytometry to identify synechococcus populations present in a water sample, there has been no holistic approach to identifying different cell populations present in a sample. More often than none, scientist and ecologist resolve to relying on experts knowledge and opinion to identify cell population of interest, a process which is highly subjective and makes reproducibility difficult.

This tutorial demonstrates a set of reproducible steps to follow to filter out synechococcus population from data obtained via flow cytometry using the cyanoFilter package in R. The steps also involve data quality checks since these experiments are typically performed at different dilution levels. Furthermore, cyanoFilter can return full expression matrices with indicators for each identified cell type. The steps are demonstrated using resulting data from an experiment involving synechoccus bacteria under some environmental stressor.

cyanoFilter provides a reproducible approach to identifying and extracting synechococcus population as well as other flow cytometry outcomes (debris, doublets and margin events). It also opens the door to further research where full effect of environmental stressors on synechococcus can be studied.

Keywords: flow cytometry, synechococcus, ecology,reproducible research, R.

1. Introduction

Flow cytometry and its outcomes Knowing fully well that outcomes from a flow cytometry experiment can be; debris (dead particles), doublets (cells with disproportionate width-height relationship), margin events (cells too large to measure) and singlets (good cells), these set of steps are designed to identify each possible outcome and return the desired cell population. It is largely applied in biomedical and medical sciences but there are recent applications of the technique in Ecology.

Synechococcus cyano bacteria and its ecological importance They bacteria are also one of the known oldest life forms known to obtain their energy via photosynthesis.

The central aim of this tutorial is to introduce readers to the software package **cyanoFilter**. The software package aims at identifying populations of two *synechococcus* type cyanobacteria obtained via flow cytometry expriments. It also provides ecologist working on these *synechococcus* type cyanobacteria with a toolset that not only helps them identify them but also helps them with indicators for other potential outcomes, which paves way for joint analysis of these populations. The package also reveals that reproducibility is quite possible in experiments whose outcome is largely driven by experts knowledge.

This tutorial is structured as follows; a review of existing methods for identifying cell populations in flow cytomtery experiments alongside the choice of method employed in **cyanoFilter** is presented in section 2. Section 3 introduces the reader to some essential and crucial properties of the synechococcus population to be identified. The next section presents the **cyanoFilter** framework, its functions alongside their purpose. Section 5 presents the *Costa* experiment, whose results will be used to demonstrate the functionalities of **cyanoFilter**, the last section gives illustrative examples on use of the package.

2. Exisitng methods for identifying cell populations in flow cytometry

Manual identification of cell population by experts via a two dimensional dotplot or one dimensional histogram is the most common method of identifying flow cytometry outcomes. This process is largely subjective and makes reproducibility difficult(O'Neill, Aghaeepour, Špidlen, and Brinkman (2013)). Hence, there has been the need to develop software tools to identify cell populations present in flow cytometry experiments. As of present, there are a number of tools available, we review some of them in the paragraph below.

Available tools for identifying flow cytometry outcomes are based on different approaches; Aghaeepour, Nikolic, Hoos, and Brinkman (2011) developed the **flowMeans** package based on *k-means* and a change-point algorithm to identfy populations present in a flow cytometry data. The package also merges local cluster using the Mahalanobis distance metric. The package can identify concave cell populations, however it is limited when it comes to cell populations with irregular shapes (Ge and Sealfon (2012)). Ge and Sealfon (2012) developed the **flowPeaks** package which is based on unsupervised *k-means* and *finite mixture modelling* to identify cell populations in flow cytometry data.

, however these methods focus more on experiments involving human cells.

3. Crucial Synechococcus Properties

4. Software

5. The Costa Experiment

6. Illustrations

7. Discussions

8. Models and software

The basic Poisson regression model for count data is a special case of the GLM framework McCullagh and Nelder (1989). It describes the dependence of a count response variable y_i ($i = 1, \ldots, n$) by assuming a Poisson distribution $y_i \sim \text{Pois}(\mu_i)$. The dependence of the conditional mean $\mathsf{E}[y_i \,|\, x_i] = \mu_i$ on the regressors x_i is then specified via a log link and a linear predictor

$$\log(\mu_i) \quad = \quad x_i^\top \beta, \tag{1}$$

where the regression coefficients β are estimated by maximum likelihood (ML) using the iterative weighted least squares (IWLS) algorithm.

Note that around the `{equation}` above there should be no spaces (avoided in the LaTeX code by `%` lines) so that "normal" spacing is used and not a new paragraph started.

R provides a very flexible implementation of the general GLM framework in the function `glm()` (Chambers and Hastie 1992) in the **stats** package. Its most important arguments are

```
glm(formula, data, subset, na.action, weights, offset,
  family = gaussian, start = NULL, control = glm.control(...),
  model = TRUE, y = TRUE, x = FALSE, ...)
```

where `formula` plus `data` is the now standard way of specifying regression relationships in R/S introduced in Chambers and Hastie (1992). The remaining arguments in the first line (`subset`, `na.action`, `weights`, and `offset`) are also standard for setting up formula-based regression models in R/S. The arguments in the second line control aspects specific to GLMs while the arguments in the last line specify which components are returned in the fitted model object (of class '`glm`' which inherits from '`lm`'). For further arguments to `glm()` (including alternative specifications of starting values) see `?glm`. For estimating a Poisson model `family = poisson` has to be specified.

As the synopsis above is a code listing that is not meant to be executed, one can use either the dedicated `{Code}` environment or a simple `{verbatim}` environment for this. Again, spaces before and after should be avoided.

Type	Distribution	Method	Description
GLM	Poisson	ML	Poisson regression: classical GLM, estimated by maximum likelihood (ML)
		Quasi	"Quasi-Poisson regression": same mean function, estimated by quasi-ML (QML) or equivalently generalized estimating equations (GEE), inference adjustment via estimated dispersion parameter
		Adjusted	"Adjusted Poisson regression": same mean function, estimated by QML/GEE, inference adjustment via sandwich covariances
	NB	ML	NB regression: extended GLM, estimated by ML including additional shape parameter
Zero-augmented	Poisson	ML	Zero-inflated Poisson (ZIP), hurdle Poisson
	NB	ML	Zero-inflated NB (ZINB), hurdle NB

Table 1: Overview of various count regression models. The table is usually placed at the top of the page (`[t!]`), centered (`centering`), has a caption below the table, column headers and captions are in sentence style, and if possible vertical lines should be avoided.

Finally, there might be a reference to a `{table}` such as Table 1. Usually, these are placed at the top of the page (`[t!]`), centered (`\centering`), with a caption below the table, column headers and captions in sentence style, and if possible avoiding vertical lines.

9. Illustrations

For a simple illustration of basic Poisson and NB count regression the `quine` data from the **MASS** package is used. This provides the number of `Days` that children were absent from school in Australia in a particular year, along with several covariates that can be employed as regressors. The data can be loaded by

```
R> data("quine", package = "MASS")
```

and a basic frequency distribution of the response variable is displayed in Figure 1.

For code input and output, the style files provide dedicated environments. Either the "agnostic" `{CodeInput}` and `{CodeOutput}` can be used or, equivalently, the environments `{Sinput}` and `{Soutput}` as produced by `Sweave()` or **knitr** when using the `render_sweave()` hook. Please make sure that all code is properly spaced, e.g., using `y = a + b * x` and *not* `y=a+b*x`. Moreover, code input should use "the usual" command prompt in the respective software system. For R code, the prompt `"R> "` should be used with `"+ "` as the continuation prompt. Generally, comments within the code chunks should be avoided – and made in the regular LaTeX text instead. Finally, empty lines before and after code input/output should be avoided (see above).

As a first model for the `quine` data, we fit the basic Poisson regression model. (Note that JSS prefers when the second line of code is indented by two spaces.)

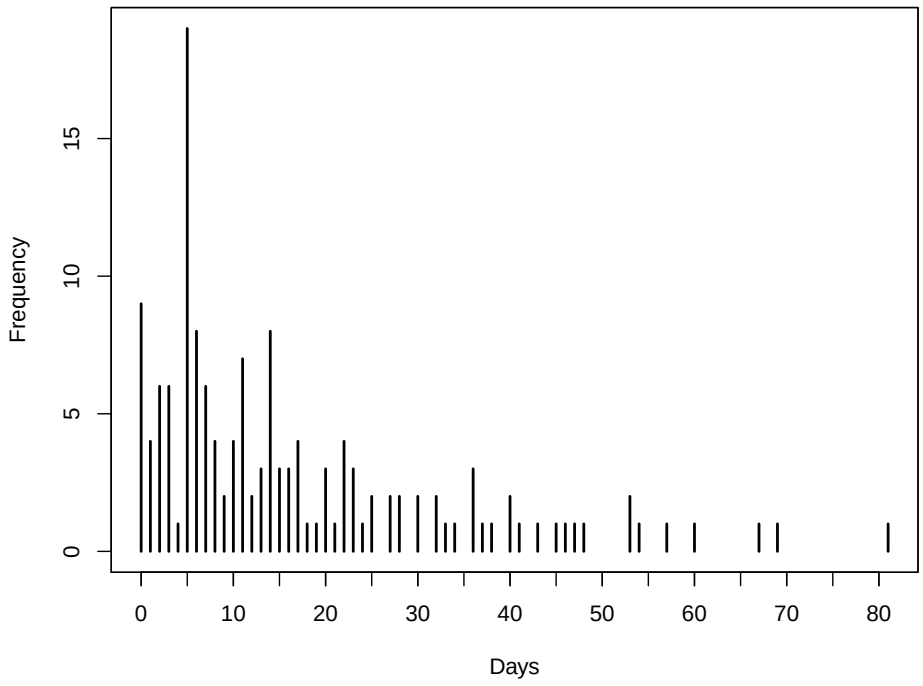

Figure 1: Frequency distribution for number of days absent from school.

```
R> m_pois <- glm(Days ~ (Eth + Sex + Age + Lrn)^2, data = quine,
+    family = poisson)
```

To account for potential overdispersion we also consider a negative binomial GLM.

```
R> library("MASS")
R> m_nbin <- glm.nb(Days ~ (Eth + Sex + Age + Lrn)^2, data = quine)
```

In a comparison with the BIC the latter model is clearly preferred.

```
R> BIC(m_pois, m_nbin)

       df      BIC
m_pois 18 2046.851
m_nbin 19 1157.235
```

Hence, the full summary of that model is shown below.

```
R> summary(m_nbin)

Call:
glm.nb(formula = Days ~ (Eth + Sex + Age + Lrn)^2, data = quine,
    init.theta = 1.60364105, link = log)

Deviance Residuals:
```

```
    Min       1Q   Median       3Q      Max
-3.0857  -0.8306  -0.2620   0.4282   2.0898

Coefficients: (1 not defined because of singularities)
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.00155    0.33709   8.904  < 2e-16 ***
EthN        -0.24591    0.39135  -0.628  0.52977
SexM        -0.77181    0.38021  -2.030  0.04236 *
AgeF1       -0.02546    0.41615  -0.061  0.95121
AgeF2       -0.54884    0.54393  -1.009  0.31296
AgeF3       -0.25735    0.40558  -0.635  0.52574
LrnSL        0.38919    0.48421   0.804  0.42153
EthN:SexM    0.36240    0.29430   1.231  0.21818
EthN:AgeF1  -0.70000    0.43646  -1.604  0.10876
EthN:AgeF2  -1.23283    0.42962  -2.870  0.00411 **
EthN:AgeF3   0.04721    0.44883   0.105  0.91622
EthN:LrnSL   0.06847    0.34040   0.201  0.84059
SexM:AgeF1   0.02257    0.47360   0.048  0.96198
SexM:AgeF2   1.55330    0.51325   3.026  0.00247 **
SexM:AgeF3   1.25227    0.45539   2.750  0.00596 **
SexM:LrnSL   0.07187    0.40805   0.176  0.86019
AgeF1:LrnSL -0.43101    0.47948  -0.899  0.36870
AgeF2:LrnSL  0.52074    0.48567   1.072  0.28363
AgeF3:LrnSL       NA         NA      NA       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(1.6036) family taken to be 1)

    Null deviance: 235.23  on 145  degrees of freedom
Residual deviance: 167.53  on 128  degrees of freedom
AIC: 1100.5

Number of Fisher Scoring iterations: 1


              Theta:  1.604
          Std. Err.:  0.214

 2 x log-likelihood:  -1062.546
```

10. Summary and discussion

- As usual ...

Computational details

If necessary or useful, information about certain computational details such as version numbers, operating systems, or compilers could be included in an unnumbered section. Also, auxiliary packages (say, for visualizations, maps, tables, ...) that are not cited in the main text can be credited here.

The results in this paper were obtained using R 3.5.1 with the **MASS** 7.3.50 package. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/.

Acknowledgments

All acknowledgments (note the AE spelling) should be collected in this unnumbered section before the references. It may contain the usual information about funding and feedback from colleagues/reviewers/etc. Furthermore, information such as relative contributions of the authors may be added here (if any).

References

Aghaeepour N, Nikolic R, Hoos HH, Brinkman RR (2011). "Rapid cell population identification in flow cytometry data." *Cytometry Part A*. ISSN 15524922. doi:10.1002/cyto.a.21007. NIHMS150003.

Chambers JM, Hastie TJ (eds.) (1992). *Statistical Models in S*. Chapman & Hall, London.

Ge Y, Sealfon SC (2012). "Flowpeaks: A fast unsupervised clustering for flow cytometry data via K-means and density peak finding." *Bioinformatics*. ISSN 13674803. doi:10.1093/bioinformatics/bts300.

McCullagh P, Nelder JA (1989). *Generalized Linear Models*. 2nd edition. Chapman & Hall, London. doi:10.1007/978-1-4899-3242-6.

O'Neill K, Aghaeepour N, Špidlen J, Brinkman R (2013). "Flow Cytometry Bioinformatics." *PLoS Computational Biology*, **9**(12). ISSN 1553734X. doi:10.1371/journal.pcbi.1003365.

A. More technical details

Appendices can be included after the bibliography (with a page break). Each section within the appendix should have a proper section title (rather than just *Appendix*).

For more technical style details, please check out JSS's style FAQ at https://www.jstatsoft.org/pages/view/style#frequently-asked-questions which includes the following topics:

- Title vs. sentence case.
- Graphics formatting.
- Naming conventions.
- Turning JSS manuscripts into R package vignettes.
- Trouble shooting.
- Many other potentially helpful details...

B. Using BibTeX

References need to be provided in a BibTeX file (`.bib`). All references should be made with `\cite`, `\citet`, `\citep`, `\citealp` etc. (and never hard-coded). This commands yield different formats of author-year citations and allow to include additional details (e.g., pages, chapters, ...) in brackets. In case you are not familiar with these commands see the JSS style FAQ for details.

Cleaning up BibTeX files is a somewhat tedious task – especially when acquiring the entries automatically from mixed online sources. However, it is important that informations are complete and presented in a consistent style to avoid confusions. JSS requires the following format.

- JSS-specific markup (`\proglang`, `\pkg`, `\code`) should be used in the references.
- Titles should be in title case.
- Journal titles should not be abbreviated and in title case.
- DOIs should be included where available.
- Software should be properly cited as well. For R packages `citation("pkgname")` typically provides a good starting point.

Affiliation:

Oluwafemi Olusoji
Center for Statistics
Universiteit Hasselt
3590, Diepenbeek, Belgium.
E-mail: oluwafemi.olusoji@uhasselt.be
and
Evolutionary Biology Unit
Universite De Namur
5020, Namur, Belgium.
Frederik De Laender
Evolutionary Biology Unit
Universite De Namur
5020, Namur, Belgium.
E-mail: oluwafemi.olusoji@unamur.be
Jurg Spaak
Evolutionary Biology Unit
Universite De Namur
5020, Namur, Belgium. E-mail: jurg.spaak@uhasselt.be
Thomas Neyens
Center for Statistics
Universiteit Hasselt
3590, Diepenbeek, Belgium.
E-mail: thomas.neyens@uhasselt.be
Marc Aerts
Center for Statistics
Universiteit Hasselt
3590, Diepenbeek, Belgium.
E-mail: marc.aerts@uhasselt.be

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd