

flowClean

Christopher Fletez-Brant, Pratip Chattopadhyay

Modified: April 1, 2014. Compiled: November 13, 2025

Introduction

This package contains the `flowCore` method for performing quality control on flow cytometry datasets. This method is described in [1].

```
> library(flowClean)
> library(flowViz)
> library(grid)
> library(gridExtra)
```

Data

Example data is a real FCS file in which we intentionally perturbed the fluorescent intensity (FI) of a subset of cells along the V705 channel ('<V705-A>').

```
> data(synPerturbed)
> synPerturbed
```

flowFrame object '9301d9e4-a160-477f-a5fb-ee7d785d5655'
with 76466 cells and 17 observables:

	name	desc	range	minRange	maxRange
\$P1	FSC-A	NA	262144	0.0000	262144
\$P2	FSC-H	NA	262144	0.0000	262144
\$P3	SSC-A	NA	261589	0.0000	261589
\$P4	Time	NA	2048	0.0000	2048
\$P5	<B515-A>	CD27 FITC	260954	-26.8846	260954
...
\$P13	<V450-A>	CD127 BV421	260964	-35.9838	260964
\$P14	<V545-A>	Aq Blu	260949	-22.2072	260949
\$P15	<V585-A>	CD8 QD585	261965	-111.0000	261965
\$P16	<R660-A>	CD45RA APC	261023	-96.5092	261023
\$P17	<V605-A>	CD4 BV605	261131	-111.0000	261131

212 keywords are stored in the 'description' slot

Quality Control

The full details are available in [1]. The motivating idea for this methodology is that populations in a flow experiment should be collected nearly uniformly with respect to time of collection. The primary actor in flowClean is the `clean`, which tests for deviations from uniformity of collection. Specifically, the collection time is discretized into l periods, each of which can be considered a N -part composition

$$D_{j=1..l} = [P_1, P_2, \dots, P_N]$$

with each P_i the frequency of a population defined as +/- with respect to some threshold; the default is the median FI of a flow parameter. By default $l = 100$.

Each D_j then undergoes the centered log ratio (CLR) transformation [2]:

$$CLR(D_j) = \left[\ln \frac{P_1}{g(D_j)}; \dots; \ln \frac{P_N}{g(D_j)} \right]$$

where

$$g(D_j) = \sqrt[N]{P_1 P_2 \dots P_N}$$

To avoid $-\text{Inf}$ values, substitution of zeroes is performed using the 'modified Aitchison' of [3].

The L_p norm of the subset $CLR(D_j) > 0$, denoted $L_p = \|CLR(D_j)\|^+$, where $p = |CLR(D_j) > 0|$, is then calculated for each D_j and changepoint analysis is performed on the set of all $\|CLR(D_j)\|^+$.

If there are no changes then the FCS is assumed to contain no errors. Otherwise, the means of the periods are compared relative to the mean of the longest period between changepoints and thresholded according to some k , which empirically works well with $k = 1.3$.

Actually calling `clean` requires only specifying a `flowFrame`, which markers are to be analyzed (generally without the 'scatter' parameters), the name to be given to the output (directory structure can be included) and the file extension:

```
> synPerturbed.c <- clean(synPerturbed, vectMarkers=c(5:16),
+                          filePrefixWithDir="sample_out", ext="fcs", diagnostic=TRUE)
```

```
[1] "flowClean has identified problems in synPerturbed.FCS with 24, 25, 26, 27, 28,
```

```
> synPerturbed.c
```

```
flowFrame object '9301d9e4-a160-477f-a5fb-ee7d785d5655'
with 76466 cells and 18 observables:
```

	name	desc	range	minRange	maxRange
\$P1	FSC-A	NA	262144	0.0000	262144
\$P2	FSC-H	NA	262144	0.0000	262144
\$P3	SSC-A	NA	261589	0.0000	261589
\$P4	Time	NA	2048	0.0000	2048
\$P5	<B515-A>	CD27 FITC	260954	-26.8846	260954

```

...           ...           ...           ...           ...
$P14 <V545-A>      Aq Blu      260949    -22.2072      260949
$P15 <V585-A>     CD8 QD585     261965   -111.0000      261965
$P16 <R660-A>    CD45RA APC     261023    -96.5092      261023
$P17 <V605-A>     CD4 BV605     261131   -111.0000      261131
18   GoodVsBad    GoodVsBad     262144     0.0000      262143
222 keywords are stored in the 'description' slot

```

The result is an FCS file identical to the input file with a new parameter, 'GoodVsBad', in which 'Good' cells all are given $FI < 10000$ and 'Bad' cells are given $FI \geq 10000$, which allows for easy programmatic gating out of 'Bad' cells from multiple FCS files. This parameter can also be used in plots as any other flow parameter as well.

```

> lgcl <- estimateLogicle(synPerturbed.c, unname(parameters(synPerturbed.c)$name[5]))
> synPerturbed.cl <- transform(synPerturbed.c, lgcl)
> p1 <- xyplot(`<V705-A>` ~ `Time`, data=synPerturbed.cl,
+             abs=TRUE, smooth=FALSE, alpha=0.5, xlim=c(0, 100))
> p2 <- xyplot(`GoodVsBad` ~ `Time`, data=synPerturbed.cl,
+             abs=TRUE, smooth=FALSE, alpha=0.5, xlim=c(0, 100), ylim=c(0, 20000))
> rg <- rectangleGate(filterId="gvb", list("GoodVsBad"=c(0, 9999)))
> idx <- filter(synPerturbed.cl, rg)
> synPerturbed.clean <- Subset(synPerturbed.cl, idx)
> p3 <- xyplot(`<V705-A>` ~ `Time`, data=synPerturbed.clean,
+             abs=TRUE, smooth=FALSE, alpha=0.5, xlim=c(0, 100))
> grid.arrange(p1, p2, p3, ncol=3)

```

SessionInfo

- R Under development (unstable) (2025-11-04 r88984), aarch64-apple-darwin20
- Locale: C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
- Time zone: America/New_York
- TZcode source: internal
- Running under: macOS Ventura 13.7.8
- Matrix products: default
- BLAS:

/System/Library/Frameworks/Accelerate.framework/Versions/A/Frameworks/v

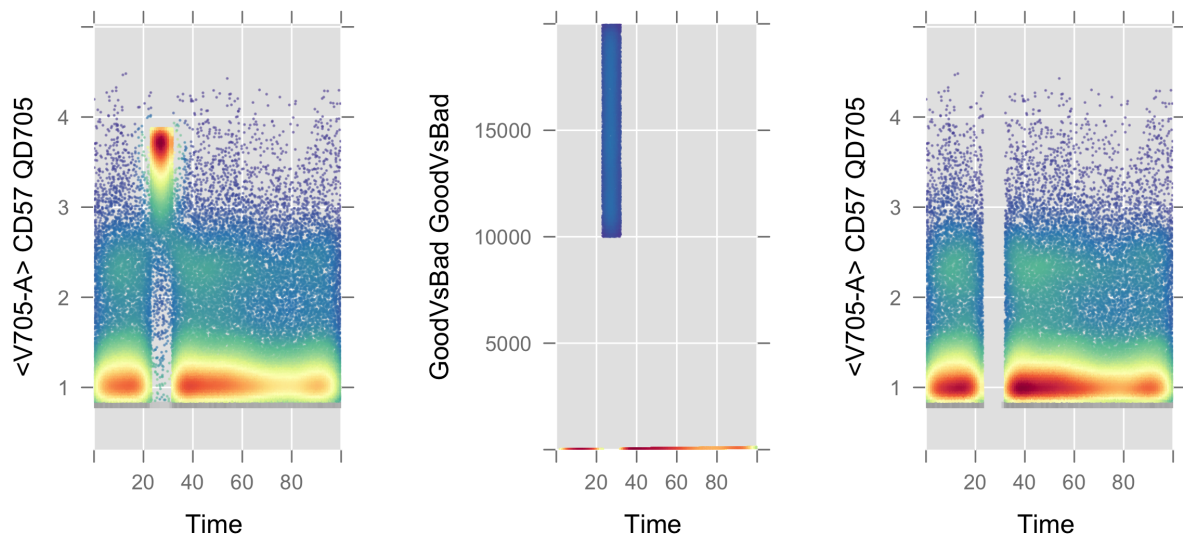


Figure 1: Left) FCS before flowClean. Center) New 'GoodVsBad' parameter. Right) FCS after flowClean and filtering.

- **LAPACK:**

```
/Library/Frameworks/R.framework/Versions/4.6-arm64/Resources/lib/libRlapack.dylib; LAPACK version 3.12.1
```

- Base packages: base, datasets, grDevices, graphics, grid, methods, stats, utils
- Other packages: flowClean 1.49.0, flowCore 2.23.0, flowViz 1.75.0, gridExtra 2.3, lattice 0.22-7
- Loaded via a namespace (and not attached): Biobase 2.71.0, BiocGenerics 0.57.0, IDPmisc 1.1.21, KernSmooth 2.23-26, MASS 7.3-65, RColorBrewer 1.1-3, RProtoBufLib 2.23.0, Rcpp 1.1.0, S4Vectors 0.49.0, bit 4.6.0, changepoint 2.3, cli 3.6.5, compiler 4.6.0, cytolib 2.23.0, deldir 2.0-4, generics 0.1.4, glue 1.8.0, gtable 0.3.6, hexbin 1.28.5, interp 1.1-6, jpeg 0.1-11, latticeExtra 0.6-31, lifecycle 1.0.4, matrixStats 1.5.0, png 0.1-8, rlang 1.1.6, sfsmisc 1.1-22, stats4 4.6.0, tools 4.6.0, zoo 1.8-14

References

- [1] Fletez-Brant C, Spidlen J, Brinkman R, Roederer M, Chattopadhyay P. Quality Control of flow cytometry data through compositional data analysis. In preparation.
- [2] Aitchison J. A concise guide to compositional data analysis. Compositional Data Analysis Workshop; Girona, Italy.

- [3] Fry J, Fry T, McLaren K. Compositional data analysis and zeros in micro data. CoPS/IMPACT Working Paper Number G-120.