QUAliFiER: An automated pipeline for quality assessment of gated flow cytometry data

Greg Finak[1] , Wenxin Jiang[1], Adam Asare[2] and Raphael Gottardo*[1]

[1]Fred Hutchinson Cancer Research Center, 1100 Fairview Avenue North, Seattle, WA 98109, USA
[2]Data Center and Data Analysis, the Immune Tolerance Network ,3 Bethesda Metro Center,Bethesda, MD 20814,USA

Email: Greg Finak - gfinak@fhcrc.org; Wenxin Jiang - wjiang2@fhcrc.org; Adam Asare - aasare@immunetolerance.org; Raphael Gottardo*- rgottard@fhcrc.org;

*Corresponding author

Abstract

Quality assessment (QA) is an important aspect of any high-throughput Flow cytometry (FCM) data analysis pipeline since the structure of flow cytometry data is relatively complex. Technical sources of error can manifest themselves in many different ways. Instrument errors, problems with antibody staining, erroneous gating, and other technical errors can appear as biases in the extracted cell subpopulation statistics or fluorescence intensities which are not necessarily obvious from a high–level view of the data. A systematic approach of quality control of flow cytometry data is necessary to effectively identify sources of technical error. Unfortunately, this is particularly onerous for large flow cytometry data sets consisting of thousands of FCS files. Although the BioConductor flowQ package performs quality control checks on ungated FCS files, it is also necessary to to identify outlier samples by monitoring the consistency of underlying statistical properties of different gated cell populations. We have developed two new packages, **flowWorkspace** and **flowQA**, for quality assessment of gated FCM data. flowWorkspace imports preprocessed and gated data from flowJo into the R environment, making the manually gated data accessible to BioConductor's computational flow tools. The flowQA package takes advantage of the availability of these manual gates to perform an extensive series of statistical quality assessment checks on the different gated cell sub–populations, and using the structure of the data or study to monitor the consistency of these statistics across and within staining panels, individuals, tubes, and channels. flowQA implements Interactive visualzation methods to allow investigators to examine the QA results across these different views of the data.

Background

Flow cytometry (FCM) is a high-throughput technology that offers rapid quantification of a set of physical and chemical characteristics for a large number of cells in a sample. The technology is widely used in health research and treatment, including for monitoring of infection, diagnosis of cancers like lymphoma and leukaemia, and auto–immune diseases [1–9]. It is also used to cross-matching organs for transplantation and in research involving stem cells, vaccine development, apoptosis, phagocytosis, and a wide range of cellular properties including phenotype, cytokine expression, and cell-cycle status [10–15]. Importantly, clinical trials investigating these areas often use flow cytometry to monitor the an individual's immune system, or the progression of the disease over time, generating very large amounts of data in the process.

Instrument errors, problems with antibody staining, erroneous gating, and other technical errors can appear as biases in the extracted cell subpopulation statistics or fluorescence intensities. Such errors are not obvious to detect from a cursory examination of the data. Careful and systematic examination of gated populations is often necessary to identify problematic samples, followed by further analysis to identify the underlying cause of the problem. There is currently a paucity of tools to help investigators effectively perform quality assessment on such large and complex flow cyotmetry data sets [16, 17].

BioConductor provides a suite of open-source tools and software infrastructure to analyze FCM and other high–throughput data [17,18]. The core of this tool set includes **flowCore**, **flowViz**,**flowQ**, and **flowStats**, which together provide functionality for basic data manipulation, visualization, some basic quality control, and automated gating [17, 19, 20].

The flowQ package provides some automated quality control procedures for FCM data using several approach to detect disturbances in the flow cells and unusual patterns in the acquisition of fluorescence and light scattering measurements over time [21]. However, the package is restricted to global measures of quality. It can only handle undated data, cannot leverage the structure of complex data sets to monitor the stability of quality measures through a study, such as the stability of common fluorescence markers across panels in longitudinal studies, or assess the statistical properties of gated cell populations .

The flowFlowJo package provides limited support for importing manually gated flowJo data into R for older versions of the software, but does not support workspace files generated by flowJo for Mac or newer versions of flowJo (¿ ver. 7) [22].

We have developed two new tools to address these issues: **QUAliFiER** (QUality Assessent for Flow ExpeRiments) and **flowWorkspace**. **flowWorkspace** makes manually gated data accessible to the computational flow community. It imports preprocessing, manual gating, and FCS files from an analyses described in flowJo workspaces, and reproduces them using the BioConductor flow tool set. The tool supports workspace files generated by flowJo for MAC up to version 9, making it complementary to the existing flowFlowJo package.

In view of the aforementioned issues, we have developed the **flowQA** package. **flowQA** uses **flowWorkspace** to import the gating template defined in flowJo by users and calculates the statistics from each gated cell population. Outliers are then detected based on the user-defined criteria.

Implementation

The package

flowQA package conducts the quality assessment using both the gated and ungated FCM data and produce visualizations for the further investigation of the samples flagged as outliers due to both biological and non-biological reasons. Its is written in R and released through Bioconductor [18], along with those **R** packages mentioned in the Background section.

The package adopts a formal object-oriented programming discipline, making use of the S4 system [23] to define classes and methods. The class,`qaTask`, is a general container that allows users to define and store all the essential information related to a particular QA task. The function,`getQAStats`, extracts and saves the statistics of each cell population defined by the gates. The core method, `qaCheck` ,does the actual QA for a particular QA task using user-specified outlier detection functions. To visualize the QA results, the method,`plot`, produces dot plots,scatterplots and density plots. Finally,the function,`qa.report`, creates the HTML report with interactive svg plots for all QA tasks.

Results and Discussion

0.1 Importing the QA gating template

`flowWorkspace` package is used to import the gating template from flowJo into R. This gating template includes the sequential gates that identify certain cell sub-poplulations that are of interest for QA purpose. In particular,`openWorkspace` method load the XML gating template.Then `parseWorkspace` parses and gating tree and calculates the statiscics for each gated population defined by the gates. The gated data as well as population statiscis are stored in an object of `GatingSet` class.

```
ws<-openWorkspace("~/QA_MFI_RBC_bounary_eventsV3.xml")

G<-parseWorkspace(ws)
```

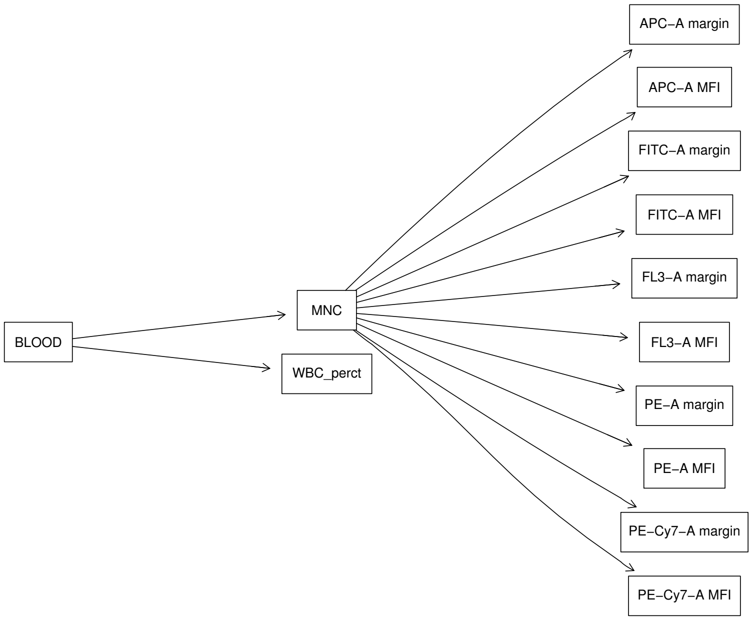

Figure 1: gating template

0.2 Extracting the statistics

The second step is to use function texttttgetQAStats to extract statistics from the gating hierarchies of `GatingSet` and re-organize them in a database that is more suitable for query and grouping. Further quality assessment will be performed based on this database with different filters and at different conditioning levels.

```
getQAStats(db)
```

0.3 Defining qaTasks

One important step before the actual outlier detection is to tell the algorithm: 1.the cell population of interest; 2.type of statistics to use; 3.the sample group within which the statistics is monitored.

We use `qaTask` class as a general container that allow users to define different QA tasks. The class stores the cell population information in `pop` slot and uses `formula` object as a compact and generic symbolic form to define the statistical property and sample groups that are involved. It is generally of the form $y \sim x|g1 * g2 * ...$, y is the statistical property to be checked, it is one of the four types:

"MFI": Median Fluorescence Intensity of the cell population specified by `qaTask`,

"percent": the percentage of the cell population specified by `qaTask` in the parent population,

"count": the number of events of the cell population specified by `qaTask`,

”spike”: the variance of intensity over time of each channel ,which indicating the stability of the fluorescence intensity.

x specifies the variable plotted on x-axis (such as date) in `plot` method.

g1,g2,.... are the conditioning variables, which divide the data into subgroups. The outlier detection is conducted whitin each individual group. They may be omitted,which indicates that the outliers detection is peformed in the entire sample set.

We provide a convenient function `makeQaTask` to read these task definition from a spreadsheet and construct multiple `qaTask` objects at once. Each entry in the spreadsheet corresponds to one QA task and contains these essential information in different columns. Users can also create the individual `qaTask` directly by using `new` method.

```
qaTask.list<-makeQaTask(db,checkListFile)
```

0.4 Quality assessment and visualiztion

`qaCheck` and `plot` perform the actual quality assessment and visualization based on the definitions stored in `qaTask` object. For example, RBC Lysis efficiency is measured by the percentage of the white blood cell(WBC) population and its `qaTask` is defined as:

```
>qaTask.list[["RBCLysis"]]

qaTask: RBCLysis

Level : Tube

Description : Sufficient RBC lysis

Plot type:  xyplot

Gated node:  WBC_perct

Default formula :percent ~ RecdDt | Tube
```

According to the formula *percent*$\sim$*RecordDate*$|$*Tube*, the statistical property *percent* will be used for the cell population *WBC* and the data will be grouped by *Tube*(or staining panel). The *percent* will be plotted against *RecordDate* in a dotplot.Here is how the QA is conducted:

```
qaCheck(qaTask.list[["RBCLysis"]],outlierfunc=outlier.cutoff,lBound=0.8)
```

`qaCheck` method reads the statistics of interest from the database and detect the outliers within each sample group. The `outlierfunc` can be any outlier function as long as takes a numeric vector as input and returns a logical vector as the output. Here the function *outlier.cutoff* provided by the package is used. *lBound* is the threshold equivalent to $\leq$ (*uBound* for $\geq$). By default `plot` method plots all the data specifiled by `qaTask`. A filter can be passed through argument *subset* to select a small subset to visualize.

```
plot(qaTask.list[["RBCLysis"]],subset="Tube=='CD8/CD25/CD4/CD3/CD62L'")
```

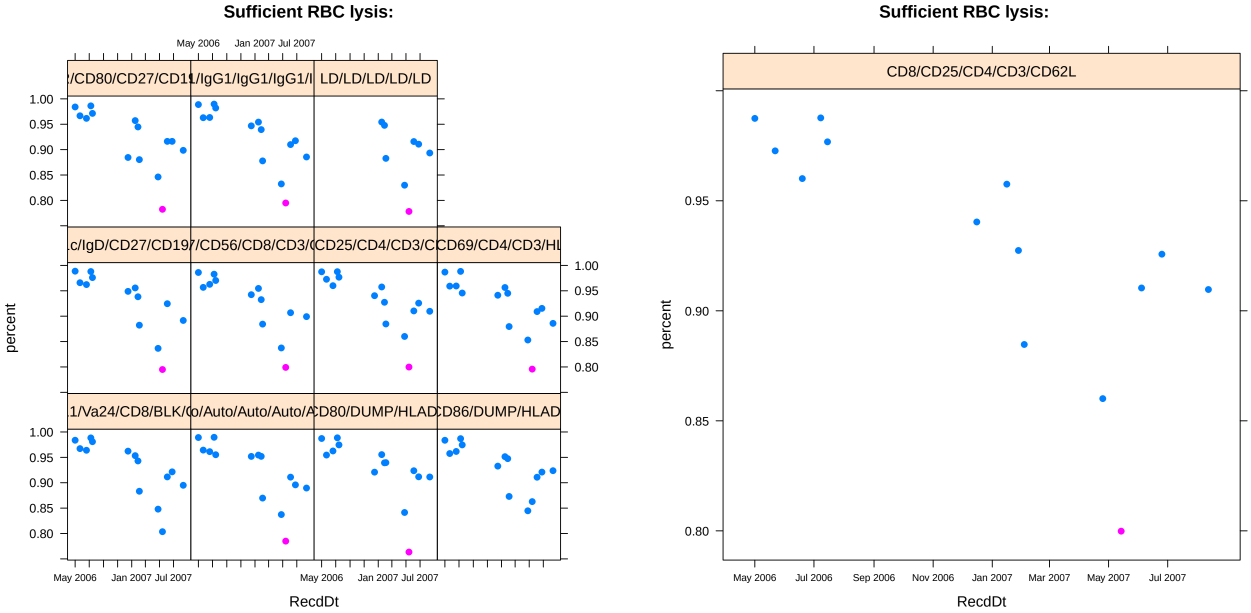

Figure 2: QA result of RBC lysis efficiency

x term in the formula is normally used for plotting. When *plotType* of the `qaTask` is defined as "bwplot" (boxplot),x is also considered as a conditioning variable that divides the data into subgroups within which the `outlierfunc` is applied. For example:

```
>qaTask.list[["MNC"]]

qaTask: MNC

Level : Assay

Description : Consistency of Lymphocyte/MNC Gate

Plot type:  bwplot

Gated node:  MNC

Default formula :percent ~ coresampleid
```

This qaTask detects the significant variance of MNC cell populations among aliquots that have the same *coresampleid*. Plot type of this object tells the method to group data by "coresampleid".

```
qaCheck(qaTask.list[["MNC"]],outlierfunc=qoutlier,alpha=1.5)
```

Interquartile Range based outlier detection function is used here.The boxplot is plotted by:

```
plot(qaTask.list[["MNC"]])
```

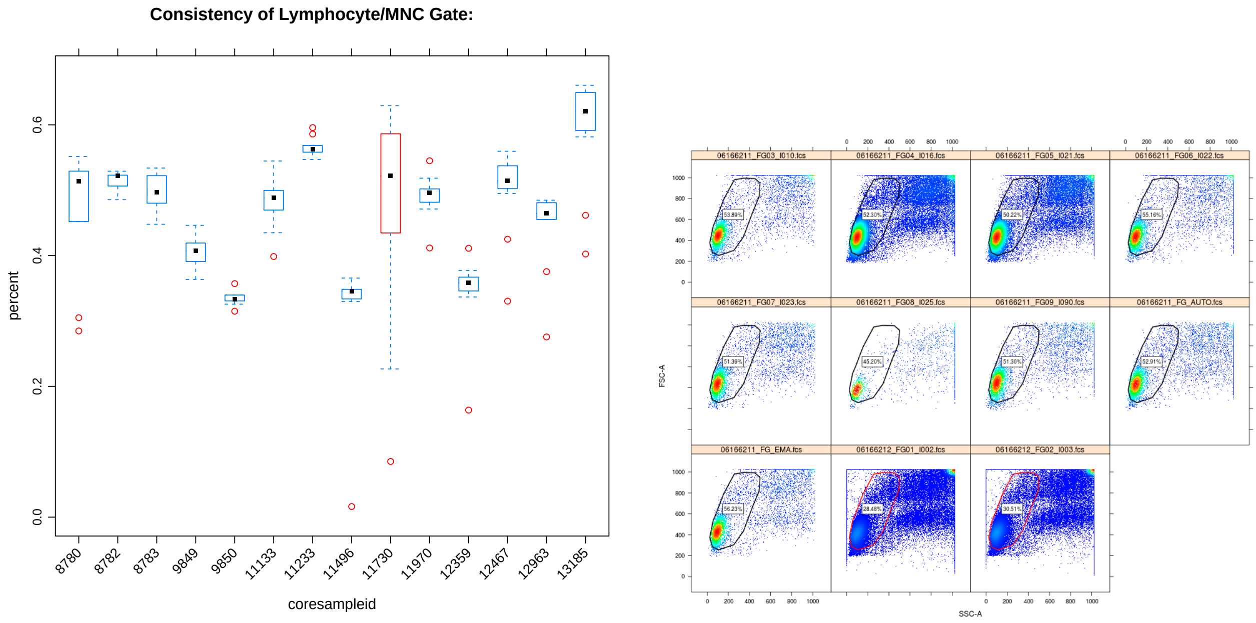

Figure 3: scatterplots of MNC gate

The red circles in the boxplot indicate the possible outlier samples and the box flagged with red color indicates the entire sample groups have significant variances and require the further investigation.Optionally,`plot` method provides the plots in svg format displaying the details of individual sample as tooltips and containing hyperlinks to the 2D scatterplots for the individual FCS files.

The `formula` can be modified through the argument of `qaCheck` and `plot` methods allowing users to interactively explore the data by changing different conditioning levels and applying simple aggregations to the statisics. The formula defined by the QA task below extracts the "percent" independently for each individual channel:

```
>qaTask.list[["BoundaryEvents"]]
qaTask: BoundaryEvents
Level : Channel
Description : Off-scale Boundary Events
Plot type:  xyplot
Gated node:  margin
Default formula :percent ~ RecdDt | channel
```

IF we want to combine boundary events of all channels and check the overall percentage for each fcs file, the aggregation function "sum" can be simply added to the formula as well as the conditioning variable changed to *fcsFile* indicating that data is grouped at FCS file level.

```
qaCheck(qaTask.list[["BoundaryEvents"]]
,sum(percent) ~ RecdDt | fcsFile
,outlierfunc=outlier.cutoff
,uBound=0.0003
)
```

The QA results will still be visualized in chanel-wise pannels by using the original formula:

```
plot(qaTask.list[["BoundaryEvents"]],percent ~ RecdDt | channel)
```

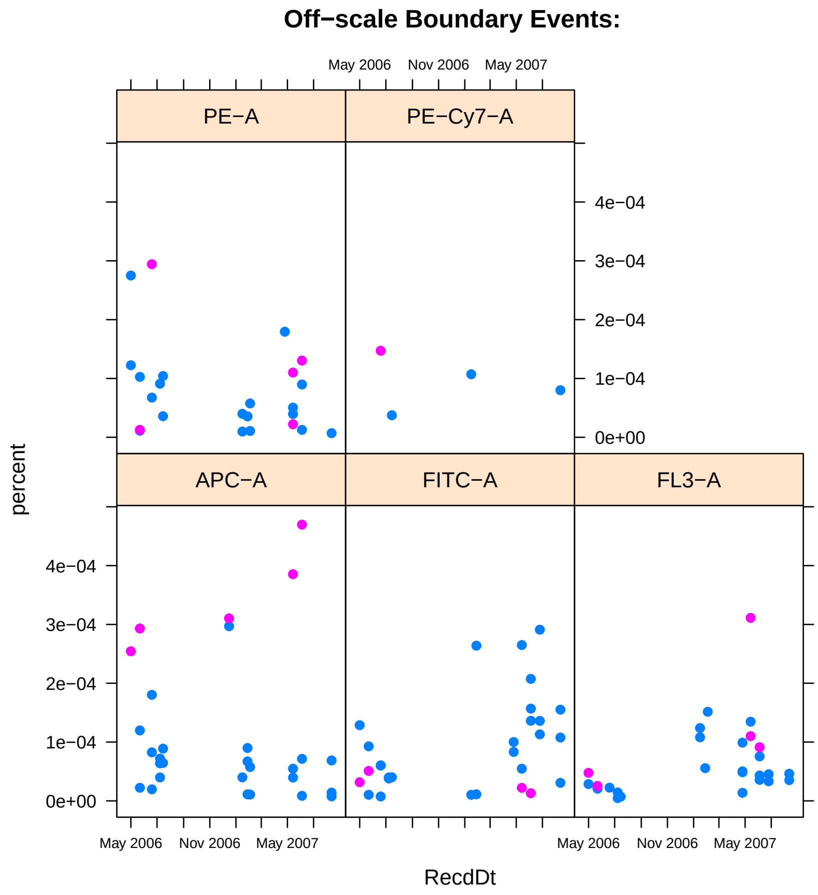

Figure 4: QA result of of boundary events

With all the statistical data accumulated in the QA database overtime, it is easy to perform QA tasks over a long period of the time for longitudinal studies. Below is the QA for monintoring fluorescence stability overtime using t-distribution based outlier detection function.

```
qaCheck(qaTask.list[["MFIOverTime"]]
,outlierfunc=outlier.t
,rFunc=lm
,alpha=0.05
)
plot(qaTask.list[["MFIOverTime"]],y=MFI~RecdDt|stain
,subset="channel%in%c('FITC-A')"
,rFunc=lm
)
```

Note that the linear regression is applied in each group in order to capture the significant MFI change over time. The sample outliers within each group is also detected based on the residue.

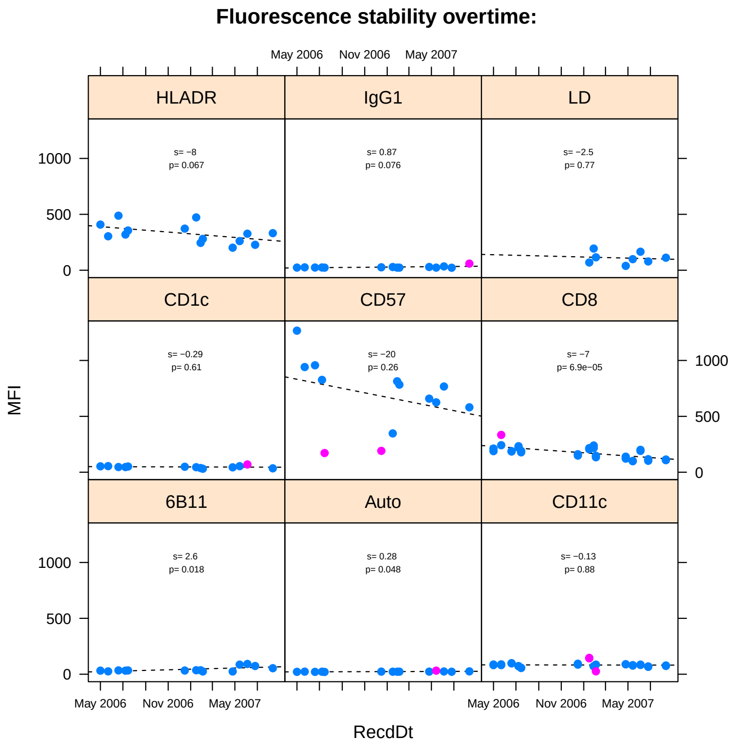

Figure 5: QA result of of MFI stability

0.5 Creating quality assessment report

Besides the visualization of each QA task using the `plot` method,we provide `qa.report` function to create QA report in HTML format for all QA tasks.The report contains the summary tables and SVG plots.

Conclusion

flowQA is a recently developed **R** package dedicated to quality assessment of gated FCM data, addressing the increasing demand for software capable of processing and minitor the voluminous amount of FCM data efficiently via an objective, reproducible and automated means. It calculates and stores the statiscis from the gated FCM data and conduct QA checks and visualization interactively.

Availability and requirements

Project name: flowQA

Project homepage: http://bioconductor.org

Operating systems: Platform independent

Programming language: R

Other requirements: R, Bioconductor

License: Artistic 2.0

Author's contributions

WJ,GF and RG developed the methodology and software, and performed the analyses. AS participated in its design and coordination. WJ and GF draft the manuscript. All authors read and approved the final manuscript.

Acknowledgements

The authors thank xxx for their advice on xxx. This work was supported by the NIH grant xxx from xxx.

References

1. Braylan RC: **Impact of flow cytometry on the diagnosis and characterization of lymphomas, chronic lymphoproliferative disorders and plasma cell neoplasias**. *Cytometry A* 2004, **58A**:57–61.
2. Hengel RL, Nicholson JK: **An update on the use of flow cytometry in HIV infection and AIDS**. *Clin. Lab. Med.* 2001, **21**(4):841–856.
3. Illoh OC: **Current applications of flow cytometry in the diagnosis of primary immunodeficiency diseases**. *Arch. Pathol. Lab. Med.* 2004, **128**:23–31.
4. Kiechle FL, Holland-Staley CA: **Genomics, transcriptomics, proteomics, and numbers**. *Arch. Pathol. Lab. Med.* 2003, **127**(9):1089–1097.
5. Mandy FF: **Twenty-five years of clinical flow cytometry: AIDS accelerated global instrument distribution**. *Cytometry A* 2004, **58A**:55–56.
6. Orfao A, Ortuno F, de Santiago M, Lopez A, San Miguel J: **Immunophenotyping of acute leukemias and myelodysplastic syndromes**. *Cytometry A* 2004, **58A**:62–71.
7. Bagwell CB: **DNA histogram analysis for node-negative breast cancer**. *Cytometry A* 2004, **58A**:76–78.
8. Keeney M, Gratama JW, Sutherland DR: **Critical role of flow cytometry in evaluating peripheral blood hematopoietic stem cell grafts**. *Cytometry A* 2004, **58A**:72–75.
9. Bashashati A, Brinkman RR: **A survey of flow cytometry data analysis methods**. *Advances in Bioinformatics* 2009, :584603.
10. Krutzik PO, Irish JM, Nolan GP, Perez OD: **Analysis of protein phosphorylation and cellular signaling events by flow cytometry: techniques and clinical applications**. *Clin. Immunol.* 2004, **110**(3):206–221.
11. Maecker H, Maino V: *Flow cytometric analysis of cytokines*, Washington, DC: ASM Press. Manual of Clinical Laboratory Immunology, 6th edition 2002 .
12. Pozarowski P, Darzynkiewicz Z: **Analysis of cell cycle by flow cytometry**. *Methods Mol. Biol.* 2004, **281**:301–312.
13. Pala P, Hussell T, Openshaw PJ: **Flow cytometric measurement of intracellular cytokines**. *J. Immunol. Methods* 2000, **243**(1-2):107–124.
14. Vermes I, Haanen C, Reutelingsperger C: **Flow cytometry of apoptotic cell death**. *J. Immunol. Methods* 2000, **243**(1-2):167–190.
15. Lehmann AK, Sornes S, Halstensen A: **Phagocytosis: measurement by flow cytometry**. *J. Immunol. Methods* 2000, **243**(1-2):229–242.
16. Shulman N, Bellew M, Snelling G, Carter D, Huang Y, Li H, Self SG, McElrath MJ, De Rosa SC: **Development of an automated analysis system for data from flow cytometric intracellular cytokine staining assays from clinical vaccine trials.** *Cytometry Part A : the journal of the International Society for Analytical Cytology* 2008, **73**(9):847–856.
17. Hahne F, Le Meur N, Brinkman R, Ellis B, Haaland P, Sarkar D, Spidlen J, Strain E, Gentleman R: **flowCore: A Bioconductor software package for high throughput flow cytometry data analysis**. *BMC Bioinformatics* 2008.
18. Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, Hornik K, Hothorn T, Huber W, Iacus S, Irizarry R, Leisch F, Li C, Maechler M, Rossini AJ, Sawitzki G, Smith C, Smyth G, Tierney L, Yang JYH, Zhang J: **Bioconductor: Open software development for computational biology and bioinformatics**. *Genome Biol.* 2004, **5**(10):R80.
19. Hahne F, Khodabakhshi A, Bashashati A, Wong CJ, Gascoyne RD, Weng A, Seyfert-Margolis V, Bourcier K, Asare A, Lumley T, Gentleman R, Brinkman R: **Per-channel basis normalization methods for flow cytometry data**. *Cytometry Part A* 2010, **77A**:121–131.
20. Sarkar D, Le Meur N, Gentleman R: **Using flowViz to visualize flow cytometry data**. *Bioinformatics* 2008, **24**(6):878–879.
21. Bashashati A, Brinkman RR: **A Survey of Flow Cytometry Data Analysis Methods**. *Advances in Bioinformatics* 2009, **2009**.
22. Gosink JJ, Means GD, Rees WA, Su C, Rand HA: **Bridging the Divide between Manual Gating and Bioinformatics with the Bioconductor Package flowFlowJo**. *Advances in Bioinformatics* 2009, :809469.
23. Chambers JM: *Programming with Data: A Guide to the S Language*. Springer 2004.

Figures

Figure 1 - Sample figure title

A short description of the figure content should go here.

Figure 2 - Sample figure title

Figure legend text.

Tables

Table 1 - Sample table title

Here is an example of a *small* table in LaTeX using `\tabular{...}`. This is where the description of the table should go.

My Table		
A1	B2	C3
A2	...	..
A3	..	.

Additional Files

Additional file 1 — Sample additional file title

Additional file descriptions text (including details of how to view the file, if it is in a non-standard format or the file extension). This might refer to a multi-page table or a figure.

Additional file 2 — Sample additional file title

Additional file descriptions text.