User Guide for *Flow*
(Ontology, statistics and visualization for flow cytometry)

Flow is an open source software tool for flow cytometry analysis that integrates ontology, statistics and visualization for the classification and comparison of multi-color flow cytometry data sets. Together, these provide powerful new tools for the experimentalist or clinician to analyze flow cytometry data in a different way; one we hope will complement and perhaps one day supersede the traditional strategy of sequential gating.

One of the main problems with flow cytometry today is the lack of standardization, both in the analysis process as well as in reporting the outcome of such analysis. As is well known, there is an element of subjectivity in defining gates, and it may be difficult to reproduce the multi-stage process used to define a particular cell type, especially if a novel marker panel is used. Similarly, different laboratories may disagree on the name for a particular cell phenotype, particularly if the phenotype represents one of many different possible activation states or differentiation end points. One possible solution is to use a well-defined common vocabulary that describes the objects and relations of interest formally, so as to minimize the potential for confusion. Such a common vocabulary is known as an ontology, and we believe that *Flow* is unique among flow cytometry packages in integrating both a process ontology from the Ontology for Biomedical Investigations (OBI) to describe the analysis process, as well as a structural ontology from the Cell Ontology of the Open Biomedical Ontologies (OBO) group to standardize both the process and outcome of flow cytometry analysis. In addition to providing a common peer-reviewed vocabulary, the use of ontologies significantly expands the scope for automated processing and machine learning in flow cytometry.

All flow cytometry packages provide some ability to do statistics, but this is typically limited to elementary summary measures – total counts, means, medians and variances. We are interested in exploring the possibility of automatic or user-guided clustering and classification using kernel density estimation and Bayesian mixture models. The idea is that if a cell subset forms a distinct visual subgroup that can be demarcated visually with sequential gating, it should also be possible to extract the subset with the appropriate statistical model. If so, this will remove much of the tedium and subjectivity inherent in gating on multi-color flow cytometry data. To facilitate statistical classification, *Flow* also provides a variety of filters, transformations and projections to slice, dice and otherwise manipulate the data. Of course, a statistical classification is not of much use without mapping back to the biology, and this is achieved through the integration with the Cell Ontology outlined above.

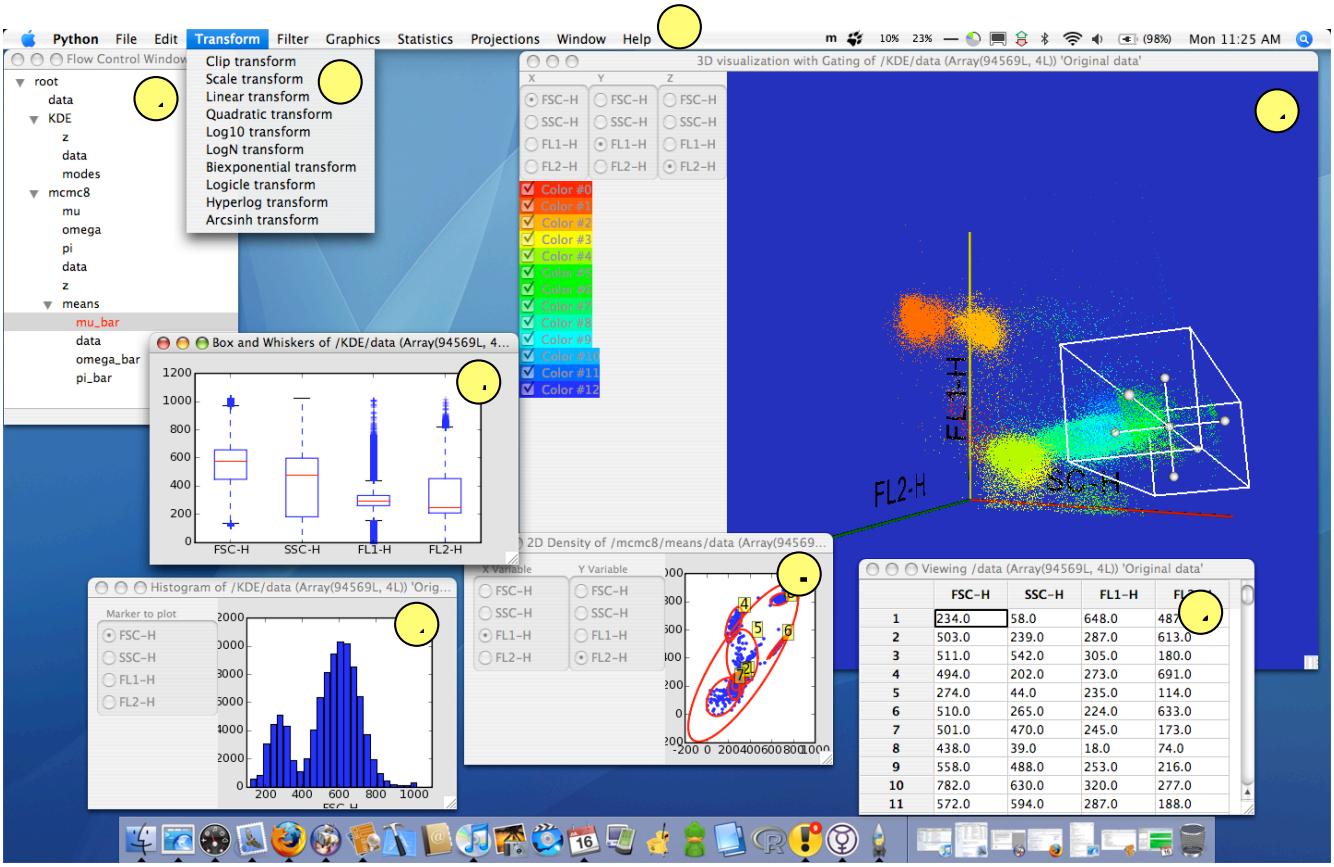
The analysis of multi-color flow cytometry is largely a process of trying to find structure in a high-dimensional space, and visualization is an invaluable aid to such exploratory data analysis. In addition to the standard histogram and dot plot views of flow data, *Flow* also provides graphics to visualize data in three dimensions, along with the ability to rotate, pan and zoom into or out of the 3D view. As a bridge to experimentalist coming from traditional flow cytometry software applications, we have also included the ability to draw and manipulate gates in both two and three dimensions. In the near future, we plan to include graphics techniques for visualizing data in more than three dimensions simultaneously, using parallel coordinate plots, biplots and interactive tours.

Finally, *Flow* was designed from the bottom up to have a plug-in architecture so that new functionality could easily be added or old functionality upgraded. If you are comfortable with programming in

Python, C, C++, Fortran or R, please see the Plugin Developer's Guide to see how you can easily teach *Flow* to do new tricks.

Installation

Please see instructions at http://galen.dulci.duhs.duke.edu/flow/wiki/Software.



Illustrated Map of Flow

Figure 1: Screenshot of *Flow* application.

Some of the major components of *Flow* are shown in Figure 1, and will be briefly described here.

1. Menu bar
   This is the main menu bar. On OS X, the menus available will change depending on which window is currently active. For example, if a 3D graphics window is currently selected, the menu bar will have options to export graphics and for gating.

2. Control window
   The tree widget in this window shows the data sets opened, as well as the history of data manipulation, and is fully described in the next section. This is the control center for *Flow*.

3. Table showing values of selected leaf in the Control window. This is available for all leaves by right clicking and selecting the View option. The selected leaf is the one highlighted and colored red.

4. Transform menu
   This shows the transforms that are currently available. New transforms can easily be added as plugins. Other menus for data manipulation include Filter, Projections and Statistics.

5. Box and whiskers plot
   A traditional box and whiskers plot, showing the spread of values for all the markers.
6. Histogram
   A histogram showing the distribution of the FSC channel. The radio box on the left switches the channel plotted.

7. Dot plot
   A dot plot with the FL-1 channel on the x-axis and the FL-2 channel on the y-axis. Dot plots can be overlaid with contour plots or (as shown) the confidence ellipses of the clusters found using Bayesian mixture models.

8. 3D plot
   A 3D plot of FL-1, FL-2 and FSC, with the axes labeled accordingly. Each point in space represents an event, in this case colored according to the most likely cluster component it belongs to. The radio box on the left allows selection of any 3 arbitrary markers to visualize, and the checklist of colored boxes below can be edited to give meaningful names to each component. The 3D view is interactive and can be rotated, zoomed and panned with the mouse. A 3D box gate in white is also shown, which can also be scaled, translated and rotated with the mouse.

Input and Output

This is done via the File menu.

File

The native file format of *Flow* is the Hierarchical Data Format (HDF5) developed by NCSA. To view an HDF file, select the Open option. To import data in FCS or CSV format, select the appropriate option from the Import sub-menu. One can also Save (in HDF5 format) or Quit from the File menu.

The Control Window

A flow cytometry analysis session can be quite complicated, with the import of multiple FCS files, each of which may need its own filters, transforms, projections and statistical manipulations to make some sense of. The Control window's purpose is to help keep track of such multi-staged analysis of multiple data sets, so that one can easily switch to an arbitrary stage in the analysis of some data set and proceed from there.

The Control window represents the entire flow cytometry analysis session as a tree structure where each node represents a transformation group, with each group typically containing leaves representing a flow data set and meta-information about that data set. By convention, a group that does not have a leaf called 'data' is understood to share the same data as its parent. For example, the Control window in Figure 2 shows two groups labeled **3FITC_4PE_004** and **KDE**. The group **3FITC_4PE_004** just contains **data** with no additional meta-information, while the KDE group is a child of the **3FITC_4PE_004** group and contains meta-information about data (shared with its parent since no 'data' leaf is shown) in the leaves **z** and **modes**.

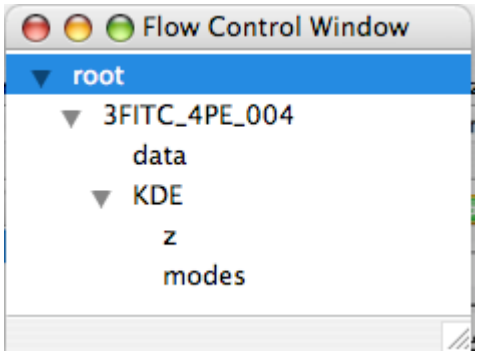


Figure 2: *Flow* control window.

Any node or leaf can be selected by clicking on it, and subsequent operations will be performed on the selected entity. Such operations include filtering, transforming or projecting the data, running a statistical routine such as kernel density estimation or Bayesian mixture modeling or displaying the selection graphically. In addition, it is also possible to operate on the tree itself, either to view the selected entity as a table or to alter the tree organization. The manipulation of the tree structure can be done from the Edit menu or by using the context menu that pops up on right-clicking the mouse within the Control window.

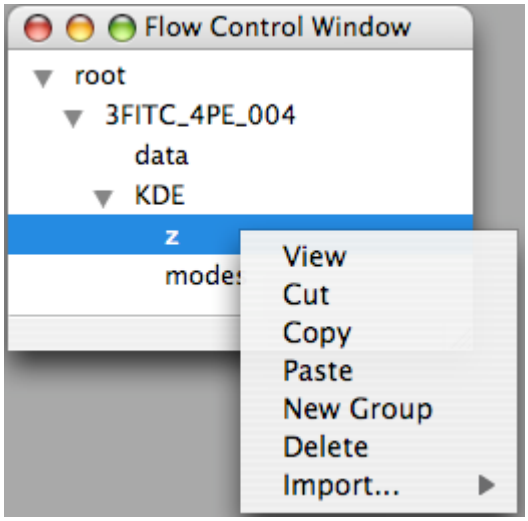


Figure 3: Context menu for flow control window.

View brings up a table showing the values of the selected leaf. Cut, Copy and Paste are useful for moving leaves from one group to another, while Delete removes the selected entity and all its children, if any. Import allows one to insert a new data set from FCS or CSV files at an arbitrary location in the tree. Drag and drop operations for the Control tree elements are planned.

Manipulating Data

There are basically 4 different types of operations on data allowed by *Flow*. A filter extracts a sub-sample of the original data, a transformation modifies the actual data, a projection reduces the dimensionality of the data and a statistical operation creates new meta-information about the data. Each such operation typically creates a new child group with the modified data or newly created meta-information. Since all these are implemented as plugins, we will only describe the core operations that come with a standard installation of the software; optional plugins should be described within the application's Help documentation.

Filters

- Filter by channel

  This brings up a radio box in which to select the desired channels/markers. Non-selected channels/markers will not appear in the resulting child group. The number of events is unchanged, however, the number of channels/markers not selected reduces the dimension.

- Filter events: index

  This allows the user to select which events to retain, by specifying *start*, *stop* and *stride* values For example, if start=10, stop=20 and stride=2, the events retained are those in rows 10, 12, 14, 16 and 18. Note that the retained events *exclude* the stop value, so to get row 20 as well, stop must be given as 21.

- Filter events: random choice without replacement

  This allows the user to randomly select a number *n* of events to retain. These *n* events will be chosen randomly without replacement.

- Filter events: random choice with replacement

  This allows the user to randomly select a number *n* of events to retain. These *n* events will be chosen randomly with replacement. Unlike the entry immediately above, *duplicate* events may be generated.

Transformations

- Clip

  This transformation *clips* the data to lie within specified *minimum* and *maximum* values. If any data value is below the *minimum*, it is set to the *minimum*. Similarly, if any data value is above the *maximum*, it is set to the *maximum*.

- Scale

  $$x' = lower' + (upper - lower')\frac{x - min(x)}{x - max(x)}$$

- Linear

  $$x' = ax + b$$

- Quadratic

  $$x' = ax^2 + bx + c$$

- Log10

  $x \to \log(x)$

- LogN

  $x \to \ln(x)$

- Biexponential

  The biexponential transform is specified by the inverse function

  $f^{-1}(x) = a e^{by} - c e^{-dy} + f$

  where a, b, c, d and f are constants.

  In practice, the roots for a suitable range of x values are found numerically, and cubic spline interpolation used to estimate the remaining values for efficiency.

- Hyperlog

  The hyperlog transform is specified by the inverse function

  $f^{-1}(x) = 10^{xd/r} + b(d/r)x - 1$

  where b, d and r are constants determined by the data.

  In practice, the roots for a suitable range of x values are found numerically, and cubic spline interpolation used to estimate the remaining values for efficiency.

- Logicle

  The logicle transform is specified by the inverse function

  $f^{-1}(x) = Te^{-(m-w)}(e^{x-w} - p^2 e^{-(x-w)/p} + p^2 - 1)$

  where T is the maximal channel number of the data, m is the number of "decades" in natural logs desired, p is a function of w, and w determines the switchover point from linear to log scaling.

  In practice, the roots for a suitable range of x values are found numerically, and cubic spline interpolation used to estimate the remaining values for efficiency.

- Arcsinh

  $x \to \sinh^{-1}(x)$

Projections

- PCA

  Principal components analysis (PCA) is an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.

- ICA

  Independent component analysis (ICA) separates a multivariate signal into additive subcomponents by maximizing the statistical independence of the estimated components. Deviation from a Gaussian distribution is used to measure the statistical independence.

Statistics

- Summary statistics

  Minimum, maximum, mean, median, standard deviation values for the selected data.

- K-means

  The K-means algorithm is one of the simplest algorithms for unsupervised clustering, and a complete description can be found in most pattern recognition or machine learning textbooks. Currently, we simply wrap the K-means algorithm provided by pycluster (or BioPython), using an Euclidean metric, and the only parameter that needs to be given is the estimated number of components.

- KDE
  Kernel density estimation is an extension of the ideas behind the histogram that provides a smoothed estimate of the underlying density given data. In our implementation, we discover the modes (maxima) of the smoothed density by following the trajectory of 'seed' points using the simple and fast *mean shift algorithm*. By looking at which modes the *k* seeds whose initial position is closest to an event end up, we then assign the event to a mode by a majority mode among these *k* seeds. Parameters that need to be entered are
  - the bandwidth of the kernel *h* – this determines the local scale of smoothing. Small values result in a spiky density estimate, large values in a smooth one.
  - the number of seeds *k*
  - the dip test ratio – spurious modes can appear because the local density is flat. The dip test checks that there is a dip between any two modes discovered, and eliminates the lower mode in any pair whose dip does not meet the given ratio as spurious.

- MCMC
  This fits a Bayesian mixture of Gaussian densities to the data, using Markov Chain Monte Carlo (MCMC) methods. The only parameter that is required is the number of mixture components to fit to. The outcome of the fitting will be a new group with entries for the mixture proportions π, the mixture means μ, and the mixture covariance matrices Σ.

Visualization and Gating

There are several varieties of graphics available for plotting from *Flow*. Most of these will be familiar and require little explanation. All the graphics windows have an associated File menu with an Export Graphics option, supporting PNG, EPS and PDF for the 1D and 2D plots, and PNG, JPEG and EPS for the 3D plots. There is no limit to the number of each type of graphics window that can be open at any one time.

- Simple
  The simple graphics plots the value of the column entry against its index value.

- Box and whiskers
  This makes a box and whiskers plot for each column, in which the box extends from the lower to upper quartile values of the data, with a line at the median. The whiskers extend from the box to 1.5*(75%-25%)) of the data range. Flier points are those past the end of the whiskers.

- Parallel coordinates
  The parallel coordinates plot shows a line joining the value of each column for every row. It is best used for a visual presentation of the mean or mode of a dataset or mixture component rather than the events themselves.

- Histogram
  Histograms are constructed with a default of 25 bins, with a radio box for the selection of the marker to display.

- 2D density
  The 2D density or dot plot shows a random sampling of 1000 events, with the x- and y-axes selected by a radio box. Simple gating can be done by drawing a rectangle selection box within the figure, in which a gate is drawn by depressing the left mouse button dragging a mouse. An arbitrary number of such gates can be drawn, and the selected events captured by selecting the Capture Gated Events item in the Gating menu.

  The Visuals menu associated with the 2D density window has the following options, which can be plotted separately or in combination:
  - scatter – plot events as points in 2D
  - contour – show density of events as a series of colored contours
  - confidence ellipse – only applicable for data processed by a Bayesian mixture model. Draws the 95% contours associated with each component. An event within the contour has a 95% probability of being derived from that component.

- 3D density
  The 3D density plot shows all the events in the selected data, with the x-, y- and z-axes selected by a radio box. The plot can be rotated by left click and dragging, zoomed by right click and dragging, and panned by middle click and dragging.

  If each event has a label assigned by some clustering or classification algorithm, the points will be colored according to their label assignments, and a checkbox showing the label-color assignments will be shown below the radio box. Only labels selected in the checkbox will be shown in the display. However, to actually capture only events with a particular label, an extra

step of selecting Gate on All Visible Events from the Gating menu is necessary.

In addition to filtering by color, it is also possible to gate visually by selecting the Add Box item in the Gating menu. This will result in the display of a 3D box, with handles that allow for resizing. The box can be dragged by first selecting the handle in the center of the box and then dragging, and rotated by first clicking on any exposed face of the box and dragging. Selecting the Capture gated events item in the Gating menu will result in a new data set that excludes all events outside the box.

More Information

See the online documentation at http://galen.dulci.duhs.duke.edu/flow/wiki/Documentation.

Bibliography