Users' Guide to Ferdinand

Version 3.0

© 2013, Manic Software
Software written by J.G. Lawrence, Department of Biological Sciences, University of Pittsburgh
Available by anonymous FTP from cobamide2.bio.pitt.edu

Minimum Software Requirements

The software was written for 32-bit Windows operating systems, including Windows XP and Windows Vista; it also works on 64-bit Windows machines such as Windows 7 and Windows 8. It also works under Windows emulation in MacOS. Cytometry data files are large; the more memory that is available, the better. Several gigabytes of free memory are recommended. Faster computers also work better.

Users' license

The Ferdinand software package that performs analyses of data produced from flow cytometers. Users are granted a license to use Ferdinand for private, educational or commercial purposes. Users acknowledge that the authors of Ferdinand are not responsible for the results of any use or misuse of Ferdinand. Manic Software retains title and ownership of Ferdinand.

Distribution of Ferdinand

Users are permitted to distribute Ferdinand without modification of its program code or this users' guide.

Tables of Contents

Tables of Contents i
Acknowledgements v
Conventions used in this guide vi
Overview 1
- What is Ferdinand? 1
- What is threshold scoring? 1
- What is Z-Scoring? 1
- How do I get started? 1
  - A Quick Tutorial 1
- Getting Started : A Ferdinand Rapid Reference 2
  - Making a Base Protocol 2
  - Making a Secondary Protocol 2
  - Making a Tertiary Protocol 3
  - Classifying your data 4
  - Analyzing other data files 4
- A Primer on the Basics of Flow Cytometry 5
  - What are events? 5
  - What optical properties are commonly used? 5
  - How are fluorescence properties different from optical properties? 5
  - How are these data stored by Flow Cytometers? 5
  - What are Gates? 6
  - What are Channels? 6
  - What is Spillover? 6
  - What is compensation? 7

Program Interface 8
- Main Menu 8
- Keyboard Shortcuts 8
- Primary Program Panes 8
- Status Bar 9
- Export Options Buttons 9
- Preferences Form 10
  - Preferences | Data Handling 10
  - Preferences | Data Import 11
  - Preferences | File Handling 12
- File Viewer 13
  - File Viewer Menu 13
  - File Viewer Toolbar 14
  - File Viewer Tab Bar 14
  - File Viewer Status Bar 14

Ferdinand Protocols 15
- What are Protocols? 15

Managing Protocols 15
The Protocol | Channel Tab 16
Two roles for Channel Data 16
Managing Channels 16
A Wealth of Channel Data 17
The Protocol | Channel | Channel Data Tab 18
Channel Data | Description box 18
Channel Display Box 18
Channel Data | FCS Import Box 19
Channel Data | Threshold Scoring Box 19
Channel Data | Z-Scoring Box 19
Channel Data | Gaussian Curves for Z-Scoring Box 20
The Protocol | Channel | Compensation Pane 21
Compensation of Raw Data Box 21
Compensation of Baseline Corrected Data Box 21
Creating Spillover Filters 21
Resetting Compensations 21
The Protocol | Channel | Reassign Channel Data Pane 22
Assigning New Values 22
Reassigning Gaussian Data 22
The Protocol | Channel | Channel Notes Pane 22
The Protocol | Global Gates Tab 23
Managing Gates 23
Gate | Description Box 23
Gate | Display Box 24
Gate | FCS Import Box 24
Gate | Threshold Gating Box 24
Gate| Dynamic Gating Box 25
The Protocol | Genotypes Tab 26
Managing Genotypes 26
Managing Peaks 26
Setting the Likelihood threshold 27
Genotypes for singly-labeled cells 27
Genotype Notes 28
The Protocol | Matrix Tab 28
The Protocol | Protocol Notes Tab 28
The Protocol | Spillover Filters Tab 29
Creating Spillover Filters 29
Managing Spillover filters 29
Managing FCS Files 30
Reading a single FCS file : The FCS File Viewer | Load Tab 30
Reading Multiple FCS files 31
Saving Ferdinand files 31
The FCS Menu 31
Reading Series of FCS files via the Ferdinand Queue 32

Inspecting FCS Files 33
The FCS File Viewer | Header Tab 33
Creating Protocols From FCS Files 34
Selecting Gates 34
Selecting Channels 34
Deselecting Gates and Channels 35
Generating the protocol 35
The FCS File Viewer | Data Tab 35
Managing Ferdinand Files 36
What are Ferdinand Files? 36
Saving data in Ferdinand files 36
Loading Ferdinand files 36
Appending Ferdinand files 36
Peeking into Ferdinand files 37
The File Menu 37
Processing FCS & Ferdinand Files 38
The FCS File Viewer | Processed Tab 38
The FCS File Viewer | Distributions Tab 39
Histogram Display Selector 39
Display Subset Buttons 39
Gaussian Curve fits 40
Deriving a Compensation Matrix 41
Creating Seed Gaussian Distributions 41
The FCS File Viewer | Custom Detector Tab 42
Selecting Primary and Filtering Channels 42
Setting seed mean and variance 42
Setting clipping range 43
Fitting a curve 43
Curve fitting algorithm 43
Using Gaussian Distributions 43
Exploring ranges of clipping values 44
The FCS File Viewer | Variability Tab 45
Validating data across multiple files 45
Removing problematic data 45
The FCS File Viewer | Graphs Tab 46
Classifying Events 47
The FCS File Viewer | Classification Tab 47
The FCS File Viewer | Classification | Summary Pane 47
Exporting the data analysis 48
Three sets of data analyses 48
The FCS File Viewer | Classification | Class Explorer Pane 48
The FCS File Viewer | Classification | Class Histograms Pane 49
The FCS File Viewer | Classification | Automate Summary Pane 50
Performing an automated survey or scoring thresholds 50
The Anatomy of Ferdinand's Classification Output 51

The File Header 51
Simple Threshold Scoring : The Channel Summary 51
Simple Threshold Scoring : Class Analysis 51
Relaxed Z-Scoring : Class Analysis 52
Relaxed Z-Scoring : Proportion Analysis as a function of the Fluorophore 53
Relaxed Z-Scoring : Proportion Analysis as a function of the File 54
Rigorous Z-Scoring 54
Channel Means Report 55
Fitness Calculations 56
Selecting Ferdinand Files 56
Troubleshooting 57
I read an FCS file and I don’t see how to analyze the data. 57
I receive an error reading an FCS file 57
I have a Protocol loaded, but I don’t see any Channels on the Distributions Tab. 57
The distributions of compensated data in the FCS File Viewer | Distribution Tab show many peaks. 57
The Gaussian curves in the FCS File Viewer | Distribution Tab don’t line up with my data. 57
I see distributions when viewing all events, but I don’t see any for Threshold or GlobalZ sets. 57

Acknowledgements

We are indebted to numerous scientists who assisted in trouble-shooting Ferdinand, especially in its early stages. Kristen Butela made enormous contributions to shaping the facilities that must be contained within Ferdinand, and spent countless hours using early versions of the program in her work, thus proving valuable feedback on the visual interface and controls necessary to analyze data effectively. Aletheia Atzinger provided substantial suggestions to improving the data analysis capabilities.

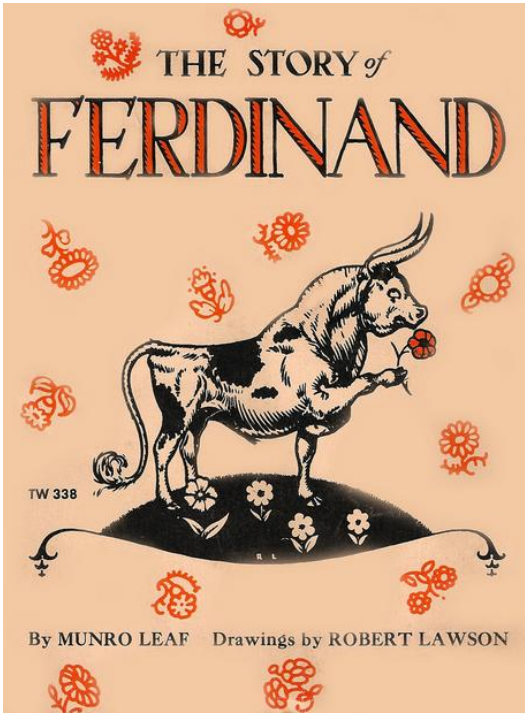
Ferdinand was written in CodeGear/Borland Delphi 2007. All code was written by Jeffrey Lawrence.

The intricacies of the FCS 3.0 file format were gleaned from two useful web sites:

http://murphylab.web.cmu.edu/FCSAPI/FCS3.html

http://www.isac-net.org/index.php?option=com_content&task=view&id=101&Itemid=150

The image of Ferdinand was drawn by the great illustrator Robert Lawson for the eponymous book written by Munro Leaf, published in 1936.



Conventions used in this guide

In hopes to increase the clarity of the text – which can get confusing due to the large amount of jargon and specialized terminology - we have followed the following formatting conventions in the preparation of this Users' Guide:

Sections are delineated by three levels of headings

Large Centered Headers Indicate Primary sections

Medium Heading are major topics within Primary sections

Small headings are subsections of each topic

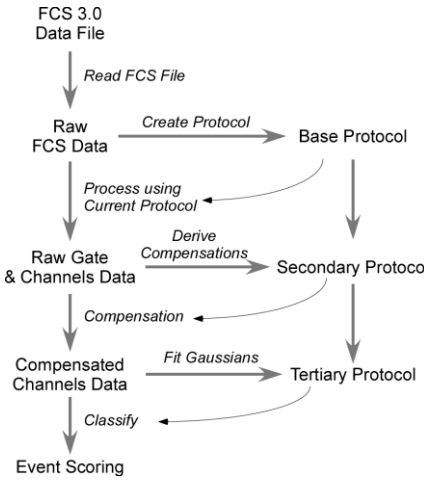
Text is formatted to increase clarity:

- Text is presented in an 11 point Calibri font.
- `Computer code and external data are presented in the Courier font.`
- References to sections of this guide appear in purple typeface.
- References to elements of the Ferdinand visual interface, such as Buttons, Tabs or Check Boxes, appear in blue typeface.
- References to important data structures, such as Protocols, Gates, Channels or Genotypes appear in red typeface.
- Filenames, such as *.fcs, appear as underline type.
- Fields within data structure, such as **ShortName**, appear as **boldface** type.

Overview

What is Ferdinand?

Ferdinand performs a number of analytical tasks when analyzing data collected by Flow Cytometers, machines which measure the optical and fluorescence properties of particles as they pass through lasers. These machines save data in FCS files, which must be read by users. In brief, an external FCS data file can be read by Ferdinand, which then uses information stored in a local Protocol to interpret these data, identify fluorescent Channels, compensate raw signal, cull events that appear to be noise by virtue of their failure to satisfy optical criteria imposed by Gates, fit Gaussian curves to the distribution of signal strength among the remaining fluorescent events, and classify those events via conventional threshold scoring or via Z-scoring.



Analyzing an FCS file *de novo*.

What is threshold scoring?

Threshold scoring uses the compensated signals on fluorescent Channels; if the signal is strong enough, that event is assumed to carry the fluorescent protein that Channel was designed to detect. This can lead to misclassification.

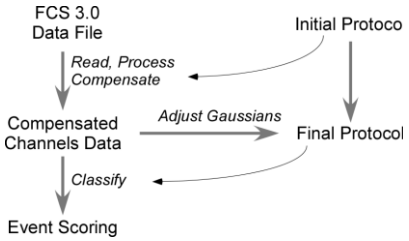
What is Z-Scoring?

Z-Scoring is an analytical method of classifying events collected by flow cytometers; it relies on the Gaussian distribution of fluorescent signal strength to separate signal from cellular events from signal arising from non-cellular events or noise. A full description of Z-scoring may be found here:

> Lawrence, J.G., K. Butela and A. Atzinger. 2013. A Likelihood approach to classifying fluorescent events using Multicolor Flow Cytometry. *Journal of Microbiological Methods. In Review*.

How do I get started?

When run for the first time, users must use Ferdinand to create an initial Protocol. The figure above shows the steps required to complete Z-scoring on an FCS data file lacking any previous Protocol. Here, data used from one or more FCS files is used to create a Protocol that can then be used to process files completely. Detailed instructions are provided below. One this has been accomplished, and the detailed protocol has been saved, this Protocol may be used to process additional FCS data files, streamlining this process. Indeed, as shown to the left, further data files can be classified automatically using a sufficiently robust protocol.



Analyzing an FCS file using a preexisting Ferdinand Protocol.

Ferdinand reading flow cytometric data that conforms to FCS 3.0 standards, reading data saved from UNIX or Windows operating systems using wither 16-bit or 32-bit data structures, saving with or without log-transformation. This does not preclude the possibility that users may find files that Ferdinand does not read correctly.

A Quick Tutorial

For a quick tutorial, see the Getting Started : A Ferdinand Rapid Reference section; this will point you to the things you need to learn first.

Getting Started : A Ferdinand Rapid Reference

Making a Base Protocol

Effective use of Ferdinand for classifying fluorescent events requires a protocol; one can be made from an FCS file as described in the Creating Protocols pages under then Inspecting FCS Files section.

- Open Ferdinand; by default, it should show the FCS File Viewer | Load Tab.
- Navigate to the directory with the FCS files of interest
- Open an FCS file; to make the process most efficient, choose a file representing a single fluorophore control
- Navigate to the FCS File Viewer | Header Tab
- Click the Show Protocol Creator Button
- Indicate Gates and Channels by double-clicking the Creator Grid
- Save the protocol and process the FCS data automatically by selecting the appropriate Check Boxes
- Click the Create Protocol Button
- Navigate to the Protocol Tab. Edit the fields under the Channels Pane and the Global Gates Pane so that **Names** and **Channel Tags** are sensical. Specify useful colors for graphic representation. Save this Protocol.

You now have a base-line Protocol. There are two things missing from it that are important to begin classifying fluorescent events. First, the Gates need to be configured to eliminate noise events. Second, the Compensation Matrix needs to be created to correct for spillover between Channels (see the A Primer on the Basics of Flow Cytometry pages in the Overview Section).

Making a Secondary Protocol

The Gates may be configured using the file already open.

- Navigate to the FCS File Viewer | Distributions Tab.
- Scroll through the distributions of values for each Gate (on the top of the list on the right-hand side).
- Note the values that seem atypical; assistance from someone experienced with cytometric data can be helpful.
- Note the ranges that appear to represent cellular events
- Modify the **Minimum Value** and **Maximum Value** for all Gates to exclude atypical values; this is done in the Protocol | Global Gates Tab.
- Save this Protocol.

The Compensation Matrix is made from Spillover Coefficients calculated from the data. This is done using a series of FCS files, each with cells expressing single fluorophores, so that spillover from the Channel of interest and secondary Channels can be determined accurately.

- Navigate to the FCS File Viewer | Distributions Tab.
- Navigate to distributions of values for the Channel where the histogram displays *raw* values for the Channel designed to detect the fluorophore expressed in this file.
- Press the Derive… Button
- Specify the range of raw values that exclude obvious noise events (usually those with very low signal strength)
- Press the Raw Compensations Button
- Navigate to the Protocol | Channels | Compensation Tab
- Verify that spillover coefficients now appear in the Compensations of Raw Data Group Box corresponding to this fluorophore

- Save this Protocol.
- Importantly, repeat these steps after loading successive FCS files, each with cells expressing a single fluorophore. This will completely populate the matrix of Spillover Coefficients.
- Save this Protocol.

Making a Tertiary Protocol

The final step is to fit initial Gaussian curves to the data. These means and variances of these curves will be used as seed values to fit Gaussian curves to FCS files with experimental data; they will work very well if the average signal strengths for fluorophores in those files are even remotely similar to the values to be saved in this Protocol.

- Open an FCS file that expressed all of the fluorophores you are using in your experiments. Alternatively, you may use the series of single-color controls that were used above to create the Compensation Matrix. If so, you'll need to re-open those when it comes time to fit curves to the compensated data.
- After the files are opened, the data should be compensated correctly using your Secondary Protocol.
- Navigate to the FCS File Viewer | Distributions Tab.
- Navigate to distributions of values for the Channel where the histogram displays *compensated* values for the Channel designed to detect a fluorophore expressed in this file (there may be more than one).
- You should see a relatively clean Gaussian distribution of your data. If not, you may have excessive noise events that may be removed by use of the Gates, or you may not have constructed an appropriate Compensation Matrix.
- The curve fit to these data may or may not be reasonable.

To fit curves to relatively clean data

- Note the values near the maximum of each Gaussian distribution.
- Navigate to the Protocol | Channels | Channel Data Tab.
- Navigate in turn through each of the Channels. For each:
    - Edit the values for the mean in the Calculate Mean Edit Box.
    - Press the Calculator Button; this saves this value in the **Mean** Field of the Gaussian Curve.
    - Save the Channel's data by pressing the Update Channel Button or navigating to the next Channel.
- Save this Protocol.
- Navigate to the FCS File Viewer | Distributions Tab. Press the Set Detectors Button at the top of the pane, making sure that the Fit Curves Check Box has been selected.
- Navigate to the FCS File Viewer | Distributions Tab.
- Gaussian curves should now be fit well to the peaks in your distributions of compensated data for each Channel.

To fit curves to relatively noisy data, or to refine curve fits

- Navigate to the FCS File Viewer | Custom Detector Tab.
- Follow the instructions in the FCS File Viewer | Custom Detector pages under the Processing FCS & Ferdinand Files section, saving the data for each fluorophore.
- Save this Protocol.
- Navigate to the FCS File Viewer | Distributions Tab. Press the Set Detectors Button at the top of the pane; the Fit Curves Check Box need not be selected as the mean and variance values have been fit precisely.
- Navigate to the FCS File Viewer | Distributions Tab.
- Gaussian curves should now be fit well to the peaks in your distributions of compensated data for each Channel.

Classifying your data

The final step in data analysis is to classify your data; this uses the Protocol you have created above to classify fluorescent events in any FCS files using the same fluorophores. If you are using doubly-labeled cells, see the Protocol | Genotypes Tab section for details on specifying Genotypes that use signal on multiple Channels.

- Make sure the Tertiary Protocol you created above is loaded; this is the Initial Protocol used for analyzing your experimental data.
- Open one or more FCS files with experimental data; multiple files should represent replicates of the same experiment. It should be processed, and Gaussian curves fit to your data, automatically
- Navigate to the FCS File Viewer | Distributions Tab.
    - Inspect the distributions of data for each Gate; compare the distributions for *All Data* to those for *Filtered Data* to ensure that appropriate classes of noise events have been excluded. If not, adjust the Protocol accordingly.
    - Inspect the distributions of raw data for each Channel to ensure that raw signal was detected.
    - Inspect the distributions of compensated data for each Channel to ensure that raw signal was detected.
- Navigate to the FCS File Viewer | Classification Tab
- Well, there you go.
- You may save your work in a *.Ferdinand file by pressing the File | Save As… Menu Item.

Analyzing other data files

After the first one, analyzing other files is straightforward. Simply follow the steps in the Classifying your data section above.

A Primer on the Basics of Flow Cytometry

It is much easier to understand Ferdinand if you are familiar with how flow cytometry works. Below, we provide some guidance and terminology used in Ferdinand and in this Users' Guide. Excellent review papers have been published on the topic that we encourage readers to consult.

A Flow Cytometers directs a stream of fluid through a series of one or lasers. This fluid should be carrying particles of interest with fluorescent properties that can be discriminated by the cytometer. Optical detectors measure light deflected by these particles; the light results in a voltage change in the detector which is recorded by the cytometer. Data for each event is recorded by the operating software. The properties of each event are stored in a data file; Ferdinand reads files stored in industry-standard FCS v3.0 files.

What are events?

Events are generated when anything passes through the laser. We'd like to think that every event corresponds to a cell or other useful particle in the fluid flow. This is not true; many events correspond to dust, cellular debris, cell aggregates or simply junk (*e.g.,* bits of agar from solid media used to propagate cells). These particles may or may not have fluorescence. Part of the job Ferdinand has is sorting out meaningful events (cellular events) from non-meaningful events resulting from anything that is not a cell.

What optical properties are commonly used?

Optical properties are terribly useful is discriminating between cellular and noncellular events. Commonly used properties include the following:

Forward Scatter. This measures the deflection of light in the same direction as the laser. The magnitude of Forward Scatter is often used as a metric of the size of the particle.

Side Scatter. This measures the deflection of light orthogonal to the direction of the laser. The magnitude of Side Scatter is often used as a metric of the complexity of the particle; *e.g.*, the membrane content of classes of eukaryotic cells gives them different Side Scatter properties from each other and from bacterial cells.

Pulse Width. This measures the duration of the event, or how long the lasers were deflected. This is also taken as a measure of the size of the particle; cell aggregate will have longer pulse widths than single cells.

How are fluorescence properties different from optical properties?

Fluorescence is measured by detectors that are orthogonal to the laser. But, unlike the measurement of optical properties, the cytometer is measuring specific wavelengths of light that are different from the wavelength of the laser. Moreover, optical filters are employed so that only particular wavelengths are measured. Different detectors – that is, detectors with different sets of filters – are configured to detect different wavelengths of light, each optimized to the emissions spectrum of a different fluorophore. For example, different detectors may be configured to detect the GFP *vs*. DsRed fluorescent proteins.

How are these data stored by Flow Cytometers?

Ferdinand reads files from flow cytometers that store their data in FCS 3.0 format. The first part comprises a file header; this contains information about the cytometer itself, the format of the file, the details of the run (date, time, operator and such) and, most critically, the parameter list. The parameters describe the nature of the bits of information that are stored for each event. Parameters correspond to features like side scatter, pulse width, or the signal strength on each of the detectors. There can be dozens of parameters for each event. Ferdinand uses this parameter list to construct its Gates and Channels within its Protocols; that is, that value in each Gate or Channel corresponds to the value in one of

the parameters. Lastly, there is a large data block in FCS files, where the values for each parameter for each event is stored.

What are Gates?

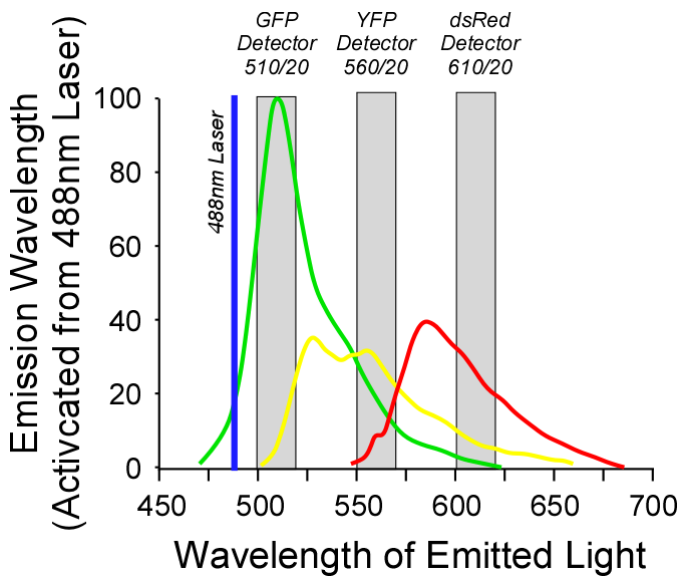
Ferdinand uses Gates to filter non-cellular events from likely cellular events. By definition, a Gate is a property that usefully applies to every event, and can be used to separate events of interest from those that are not of interest. Often, optical properties are used as Gates: forward scatter, side scatter and pulse width. Ferdinand can also use fluorescence properties as a Gate, *e.g.,* fluorescence of the nucleic acid stain Syto-62. If the values of an event on a Gate fall outside a defined acceptable range, those events are excluded from analysis. It is the job of the Ferdinand Protocol to store these ranges of acceptable ranges.

What are Channels?

Channels always correspond to parameters which report fluorescence values; that is, they always correspond to parameters that measure voltage from detectors designed for detecting emitted light from fluorescent molecules. What makes Channels different from Gates is that every useful event may not have a meaningful value on every Channel. For example, consider a cytometer with two fluorescence detectors, one optimized for GFP and the other for DsRed. Consider a sample bearing a mixture of GFP-expressing and DsRed-expressing cells. Each event would have meaningful values for all Gates: side scatter, pulse width or Syto-62 staining. However, only some of the event would have strong values on the GFP detector, while others would have strong values on the DsRed detector. These parameters would be assigned to GFP and DsRed Channels in Ferdinand. **Differential signal on Ferdinand's Channels are how different classes of events are identified**.

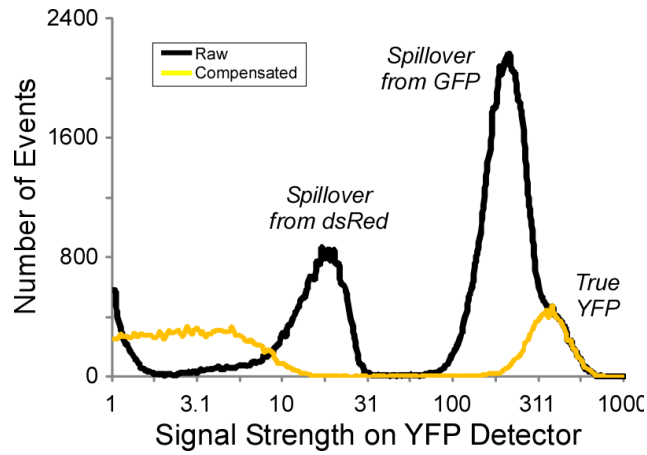
What is Spillover?

While different detectors may be optimized for measuring wavelengths of light emitted by different fluorophores, they are not perfect. Light from one fluorophore may "spill over" and register on the detector optimized for a different fluorophore. In Ferdinand, that means a single event will show non-zero raw signal on more than one Channel.



The figure to the left shows spectral overlap between GFP, YFP and DsRed. All three proteins are being excited from a 488 nm laser and the strength of the emitted light is plotted against its wavelength. GFP has maximal emissions near 510 nm; YFP has maximal emission between 530 and 560 nm, and DsRed has maximal emission near 560 nm. It should be clear, however, that the emissions spectra of the three fluorophores overlap significantly. The grey boxes show a set of filters used to discriminate between the three. The GFP detector allows light between 510 and 520 nm to pass, whereas the YFP filter allows 500-570 nm light to pass and the DsRed detector allows 600-620 nm light to pass. Each of these filters is optimized for a different fluorophore: that is, signal strength is maximal for its cognate fluorophore. However, there is obviously spillover. The GFP filter set will allow light emitted by YFP to pass through and activate the GFP detector. The YFP detector is sensitive to light from GFP and DsRed as well as YFP. The data from each of these detectors is saved as a parameter in the FCS file, which correspond to a Channel in Ferdinand.

The result of spillover is that raw signal is detected from events that do not carry the fluorescent protein the detector was designed to detect. Consider the example histogram on the right from a mixture of GFP-, YFP- or DsRed-expressing cells; it classifies events counts by their signal strength on the YFP detector. The black line denotes that histogram for the raw values on YFP Channel in Ferdinand, whereas the yellow line shows the histogram for *compensated* signal. There are large numbers of events which show strong signal on the YFP detector that do not express YFP. This represents spillover of GFP and DsRed signal on the YFP detector. Indeed, the events which represent cells that truly express YFP are buried in the large spillover peak from GFP-expressing cells. However, compensation has removed these confounding events, leaving only the true YFP-expressing cells.



What is compensation?

Compensation is the process by which actual strength of fluorescent signal is calculated from the signal measured by fluorescence detectors. To begin, we measure spillover coefficients, the relative amount a fluorescent signal spills over onto other detectors; an example is shown at the right. For each detector, the strongest signal arises from the protein that detector was designed for; this is normalized to 100. However, most detectors are readily activated by other proteins; the exception is the mKalama detector, which is minimally influenced by other proteins.

Table of spillover coefficients calculated by Ferdinand

	Cytometer Detector				
Protein	GFP	YFP	DsRed	mKalama	CFP
GFP	100.00	41.07	11.54	3.21	2.04
YFP	43.56	100.00	39.12	0.39	0.32
DsRed	0.88	5.65	100.00	0.60	0.50
mKalama	0.25	0.20	0.20	100.00	50.89
CFP	1.65	1.26	1.24	3.56	100.00

$$\underset{\textit{Measured}}{\begin{bmatrix} 546.3 \\ 223.3 \\ 69.6 \\ 20.0 \\ 10.2 \end{bmatrix}} = \underset{\textit{Spillover Matrix}}{\begin{bmatrix} 100.00 & 41.07 & 11.54 & 3.21 & 2.04 \\ 43.56 & 100.00 & 39.12 & 0.39 & 0.32 \\ 0.88 & 5.65 & 100.00 & 0.60 & 0.50 \\ 0.25 & 0.20 & 0.20 & 100.00 & 50.89 \\ 1.65 & 1.26 & 1.24 & 3.56 & 100.00 \end{bmatrix}} \underset{\textit{Actual}}{\begin{bmatrix} ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}}$$

$$\mathbf{C}^{-1} = \begin{bmatrix} 1.219 & -0.539 & 0.0197 & 0.005 & -0.014 \\ -0.504 & 1.245 & -0.066 & 0.002 & -0.007 \\ 0.057 & -0.425 & 1.024 & 0.003 & -0.008 \\ -0.037 & 0.015 & -0.007 & 1.018 & -0.036 \\ -0.005 & 0.002 & -0.002 & -0.518 & 1.019 \end{bmatrix}$$

$$\underset{\textit{Compensation Matrix}}{\begin{bmatrix} 1.219 & -0.539 & 0.0197 & 0.005 & -0.014 \\ -0.504 & 1.245 & -0.066 & 0.002 & -0.007 \\ 0.057 & -0.425 & 1.024 & 0.003 & -0.008 \\ -0.037 & 0.015 & -0.007 & 1.018 & -0.036 \\ -0.005 & 0.002 & -0.002 & -0.518 & 1.019 \end{bmatrix}} \underset{\textit{Measured}}{\begin{bmatrix} 546.3 \\ 223.3 \\ 69.6 \\ 20.0 \\ 10.2 \end{bmatrix}} = \underset{\textit{Actual}}{\begin{bmatrix} 547.0 \\ -1.8 \\ 7.2 \\ 2.5 \\ -2.3 \end{bmatrix}}$$

We begin by recognizing that the set of Measured signals (M) is a product of the Actual signal (A) and the compensation matrix (C), so that:

$$\mathbf{M} = \mathbf{C}*\mathbf{A}$$

We can then solve for the actual signal by matrix inversion, multiplying both sides by the inverse of the compensation matrix ($\mathbf{C}^{-1}$) as:

$$\mathbf{C}^{-1}*\mathrm{M} = \mathbf{C}^{-1}*\mathbf{C}*\mathbf{A} = \mathbf{A}$$

An example is shown to the left, with the vertical rows representing GFP, YFP, DsRed, mKalama and CFP. The measured signal shows strong GFP signal (546.3), but significant spillover onto the YFP and DsRed detectors (223.3 and 69.6). We simply invert the matrix of spillover coefficients, and multiply this inverted matrix against the matrix of raw values on the Channels to derive the actual values. In Ferdinand, we report the inferred "actual" signal as the "compensated" signal. Here, there is still strong GFP signal among the compensated values, but signal on other Channels is now minimal.

Program Interface

Ferdinand functions from a single main program window.

Main Menu

A standard Menu bar is located at the top. There are five menus:

- File Menu : Commands to load, save and analyze Ferdinand files
- Protocol Menu : Commands to handle Ferdinand Protocols
- FCS Menu : Commands to read FCS files
- Fitness Menu : Commands to compare the proportions of fluorescent classes in multiple Ferdinand files
- Help Menu : Commands to explore the program itself.

The commands within each of the Menus will be outlined in the appropriate sections below. They may be executed by selecting them with the mouse, navigating using the keyboard (keyboard-navigation uses the letters underlined in each menu command to select them), or by using the Keyboard Shortcut .

Keyboard Shortcuts

Many of the most common commands within the Ferdinand Main Menu may be accessed using standard Windows shortcuts. Those available are as follows:

Shortcut	Menu	Menu command
Ctrl-F	File	Load a *.Ferdinand file
Ctrl-P	File	Process the current Ferdinand File
Ctrl-S	File	Save Ferdinand file as…
Ctrl-Alt-P	File	Open the Preferences Form
Ctrl-Q	File	Quit Ferdinand
Ctrl-N	Protocol	Create a new, blank Protocol
Ctrl-O	Protocol	Open an existing *.protocol file
Ctrl-U	Protocol	Update Protocol with on-screen information
Ctrl-R	FCS	Read a single FCS 3.0 data file
Ctrl-M	FCS	Read multiple FCS 3.0 data files
Ctrl-A	FCS	Add the selected FCS files to the analysis queue
Ctrl-D	FCS	Delete the selected analysis queue entry
Ctrl-C	FCS	Clear the analysis queue
Ctrl-E	FCS	Execute the file analyses specified in the queue
F1	Help	Load this help file

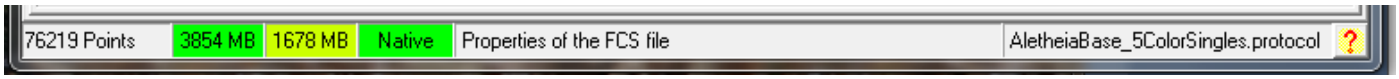
Primary Program Panes

Program control is effected over three primary tabs :

- Protocol : Create and manage Ferdinand Protocols
- FCS File Viewer : Read and analyze cytometric data
- Fitness : Compare proportions of fluorescent classes

Status Bar

The Status Bar appears at the bottom of the Ferdinand program widow. It shows several pieces of critical data.



- Event counts; the file read above has 76,219 events read
- Available physical memory, above 3854 megabytes. This panel is shown in green, so that Ferdinand believes it has plenty of memory. Color codes reflect the amount of memory available; as memory decrease, so will performance. The color table is shown at right.

Amount (Megabytes)	Color
2000 and above	
1600 – 2000	
1300 – 1600	
1000 – 1300	
800 – 1000	
600 – 800	
400 – 600	
200 – 400	
Less than 200	

- Available virtual memory. This is shown in yellow-green above, as less is available.
- The compensation mode, set to *Native*. This cannot be changed outside the developer version of Ferdinand.
- The Hint panel provides clues as to the role of the visual element over which the mouse is hovering. Here is read, "Properties of the FCS file," since the mouse was over the Header Pane of an FCS file. This is useful in clarifying the functions of buttons, check boxes and radio buttons used to control Ferdinand.
- The current Protocol is specified at the right-hand side of the Status Bar. When a native Ferdinand file is read – where a Protocol is stored with the data – this will indicate "FileStream". In the figure above, a rather long-winded Protocol (read from AletheiaBase_5ColorSingles.protocol) is active.
- At the extreme right-hand side, a Help Button appears (the one with the yellow question mark); click this button to show this help file.

Export Options Buttons

There are many opportunities to export various forms of data from Ferdinand. This may be accomplished through a number of Menu Items or, primarily, through on-screen Export Buttons. An example is shown to the right. The Export Button is shown on the left, and the Output Options Selector Buttons are shown in a group to its right. Only one of these buttons may be depressed at any one time; the nature of the output depends on which button has been depressed:

Text Button. In this case, the output is sent to a File Viewer Form. This is Ferdinand's internal text-management facility which allows users to edit, print and save text data. In the example above, the Text Button has been depressed.

Excel Button. In this case, the data will be sent to Microsoft Excel, if that program is installed and available on the computer running Ferdinand. If Ferdinand does not detect Excel on your computer (*e.g.*, you are running Excel on MacOS and running Ferdinand in emulation), this button will be disabled.

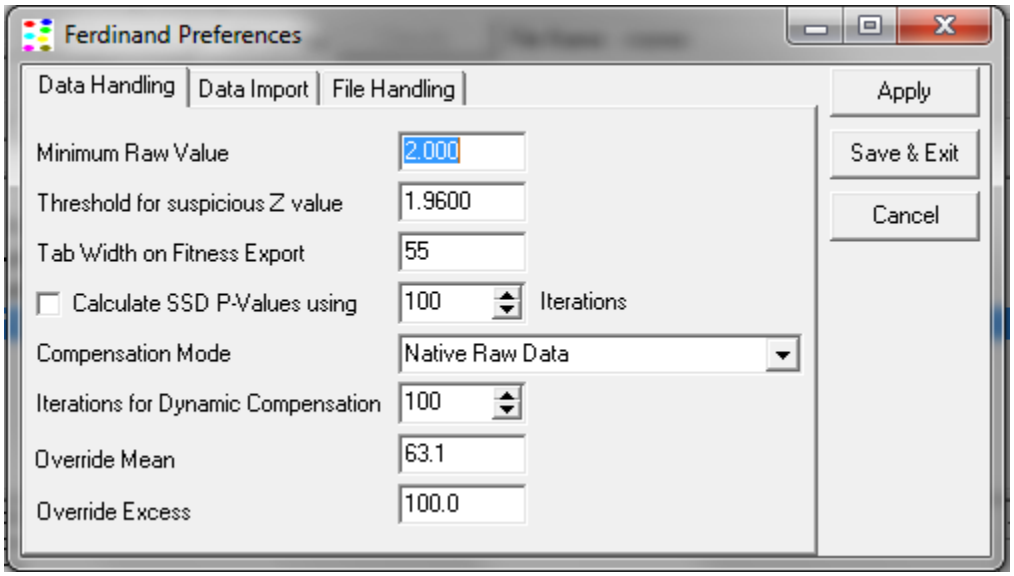
Save Button. In this case, you will be queried for a file name; the data will be saved to that file and will not be sent to a file viewer of any sort.

Preferences Form

The Preferences Form is available from the File Menu. Here, one can set some useful parameters. Some controls are disabled for typical users and can be safely ignored.

Preferences | Data Handling

Under the Data Handling Tab, one controls a number of setting that affect how data are processed. After values are changed, press the Apply Button to read those values, or the Save & Exit Button to apply them and close the form. Press the Cancel Button to close the form without reading the new values.



Minimum Raw Value. Set this threshold for the minimum value that must occur on any fluorescent Channel for that event to be saved. Use this to cull noise points which fail to have any meaningful value on any Channel.

Threshold for suspicious Z value. This controls a report flag in the classification summary if values for individual files in multi-file analyses deviate significantly from the average for all files. IT can be safely set to values in the 1.9 – 2.5 range. Lower values will trigger the flag too often to be useful.

Tab width on fitness export. Sets the tab width; this is used when formatting the data summary for fitness comparisons.

Calculate SSD P-Values. This controls the number of iterations on a bootstrap analysis for curve fits. This is extremely esoteric and if you don't know what it is, then don't check it; it will just slow you down.

Compensation Mode. This toggles between Native, baseline-corrected and Dynamic compensation modes. This is available for developers only.

Iterations for Dynamic Compensation. Controls the sampling depth; this is available for developers only.

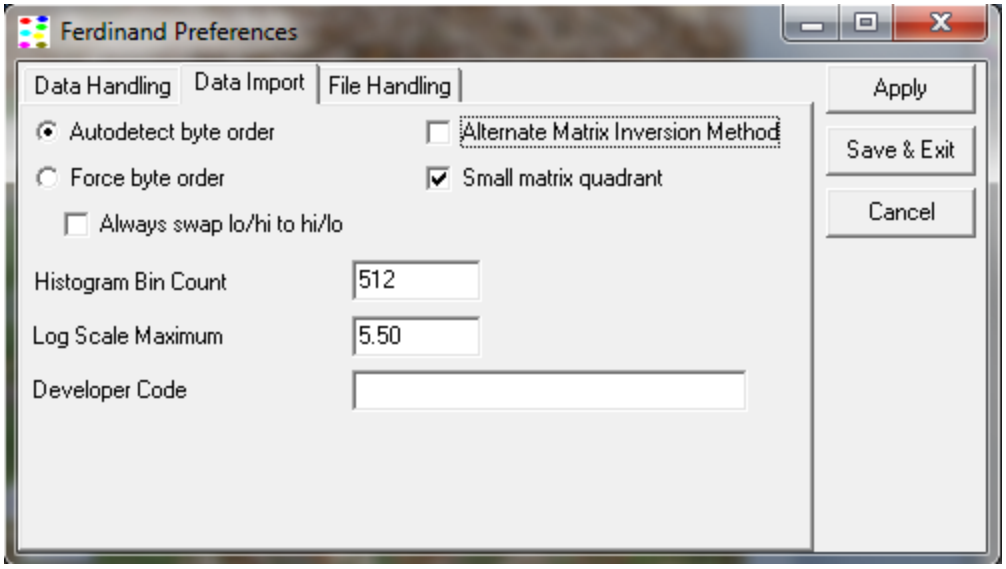
Override Mean. This is a crude control for overriding event classification. Events can be ignored when compensated signal on an alternate Channel exceeds the compensated signal on a Channel used for the maximum-likelihood Genotype (used when the assigned Channel has an extremely low mean). This value sets the threshold where Ferdinand starts paying attention to where this is a "weak-signal" Channel. Values below 100 are typical.

Override Excess. This sets the threshold for the excess on an alternative Channel that will reject this event. In the setting above, a weak Channel is one where mean signal in less than 63.1. For events wherein fluorophores detected by this

Channel are most likely to have produced the observed signal, events are not classified if compensated signal on an alternate Channel exceeds the compensated signal on this Channel by more than 100.

Preferences | Data Import

Under the Data Import Tab, one controls a number of setting that affect how data are read from FCS files.



Autodetect by order. Flow cytometers store data as 16-bit or 32-bit numbers (either 2 bytes of 4 bytes). If a number needs 4 bytes, how are those bytes stored? There are two ways; for a 32-bit number those are 1,2,3,4 or 4,3,2,1. The FCS file itself notes how these numbers are stored. So, this Check Box should always be true. For safety, it has been disabled for users; only developers can alter this to test how data are being imported.

Force byte order. You really want to do this? You know, if you're wrong all your data will be nonsense because the stored numbers will be read incorrectly (bytes in the wrong order). What are you thinking? This is disabled except for program testers.

Always swap lo/hi to hi/lo. See above.

Alternate Matrix Inversion Methods. Compensation and fitness calculations both require matrix inversion, which is kind of tricky. There are two methods encoded by Ferdinand and this selects between them. This is disabled except for developers.

Small Matrix Quadrant. See above.

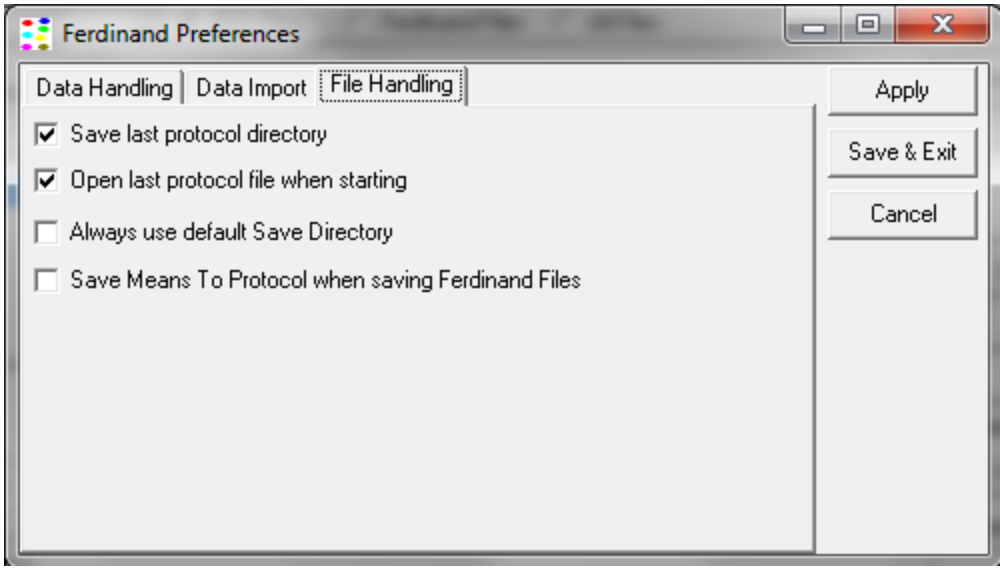
Histogram Bin Count. This sets the number of bins in distributions, which affects how Gaussian curves are fit. More bins means a better fit when large numbers of events are analyzed. Smaller numbers of bins are appropriate when fewer events are available. Summit software typically uses 256 bins. That works here. We find that 512 bins works somewhat better, assuming more than 100,000 events are being evaluated. With large data files, 1024 bins works nicely.

Log Scale Maximum. This is provided for back-compatibility only and affects only users using older (before version 2.5) versions of Ferdinand when reading files saved with older Protocols. This sets the number of decades used in Channels and Gates with log display (typically 4.0). It is not used in any Ferdinand version 3.0 data analyses.

Developer Code. Enter the access code to uncover Developer functions and controls. If you don't know what it is, you shouldn't be doing this.

Preferences | File Handling

Under the File Handling Tab, one controls a number of setting that affect how files are opened and saved.



Check the Save last protocol directory Check Box to allow rapid reading of available protocols into the Protocol Selector on the Protocol Tab. By default, Protocols are saved in the Ferdinand program directory; unless you have a good reason to change this, keep on doing it and don't uncheck this box.

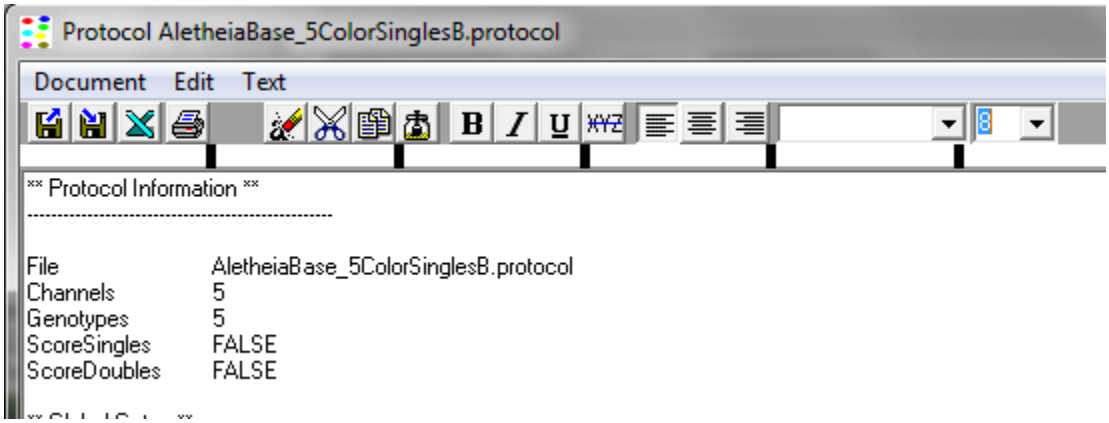
When the Open last protocol file when starting Check Box is selected, your last active protocol is loaded when the program starts. This speeds up analyses since you don't need to go do it. Not much of a reason to uncheck it unless the program is used by many users with different protocols.

Select the Always use default Save Directory Check Box to force saving Ferdinand files to its default data directory, which is a subdirectory of the directory containing the program itself.

Select the Save Means To Protocol when saving Ferdinand Files Check Box to overwrite Gaussian values in the Protocols with those in the current detectors.

File Viewer

The File Viewer Form is the standard avenue to display text output from Ferdinand. An example is shown below, produced by exporting a Protocol from the Protocol | Export Menu Item.



The title bar at the top gives an indication as to what the contents of the viewer are; here it is labeled as Protocol AletheiaBase-5ColorSinglesB.protocol. That is noting that this is a Protocol, and then it provides the file name of the protocol.

The File Viewer Form is separate from the main program window, so both may be viewed at the same time. However, its program thread is spawned from the Ferdinand program, so that when a user closes Ferdinand, the File Viewers will close as well.

File Viewer Menu

The File Viewer has three menus which offer commands to manipulate files.

The Document Menu offers commands to manipulate files; most commands are replicated with a button on the Toolbar as well:

Open	Open a text file; here, one can read a file created elsewhere
Save	Save the existing file to disk
Save as Plain	When checked, the data are saved as plain text. When not checked, rich text (rtf) files are created.
Print	Send the text to the printer
Quit	Close the File Viewer

The Edit Menu offers Windows standard commands to manipulate text; most commands are replicated using a button on the Toolbar as well. In addition, one may show or hide the Toolbar and Status Bar using commands under this menu.

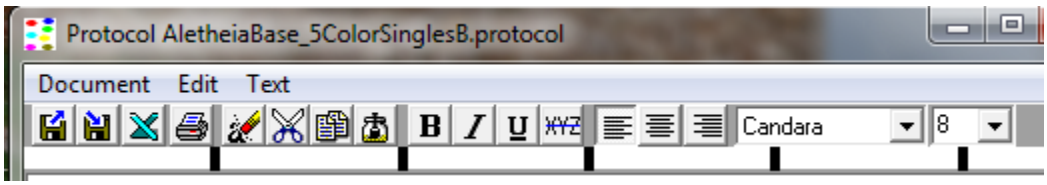
Undo	Under the previous editing command
Cut	Cut the selected text and place it on the Windows clipboard
Copy	Copy the selected text and place it on the Windows clipboard
Paste	Paste text from the Windows clipboard at the position of the cursor; if the clipboard does not contain text, nothing is pasted
Clear	Delete text but do not place it on the windows clipboard.
Select All	Select all the text in the File Viewer window

The Text Menu offers standard mark-up commands that accomplish the following; most commands are replicated on the Toolbar as well:

Bold	Make the selected text **Bold** in font style; mark-up is saved only when document is saved as Plain Text, as set in the Document Menu.
Italic	Make the selected text *italic* in font style
Underline	Make the selected text <u>underlined</u> in font style
Strike-through	Make the selected text ~~strike-through~~ in font style
Left-Justify	Justify the paragraph to the left
Center Justify	Justify the paragraph in the center
Right Justify	Justify the paragraph to the right
Word Wrap	When checked, text within paragraph wrap to the next line. When unchecked, a horizontal scroll bar is needed to follow text that will extend beyond the right-hand edge of the window.

File Viewer Toolbar

The Toolbar is located immediate below the Main Menu; it offers many of the commands located within the menus for easy access with a mouse.



Commands are organized into 5 horizontal sections of Buttons:

File Commands: Open, Save, Export to Excel, Print

Edit Commands: Undo, Cut, Copy, Paste

Format Commands: Bold, Italic, Underline, Strike-Though

Justification Commands: Left, Center, Right

Font Commands: Font Name Selector, Font Size Selector

File Viewer Tab Bar

The Tab Bar is located immediate below the Toolbar. For the paragraph selected, one may place a tab stop (all left tabs) at the position of the mouse cursor by double clicking or by right clicking and selecting the Add Tab Menu Item from the pop-up menu. Above, five default tab stops are shown. Tab stops may be removed by right-clicking the area near the tab stop and selecting the Delete Tab Menu Item from the pop-up menu.

File Viewer Status Bar

The Status Bar appears at the bottom of the File Viewer form; it shows the current row and column of the cursor, the status of the file (either *Unsaved* in a red box, or the file name in a green box), and the content hint.

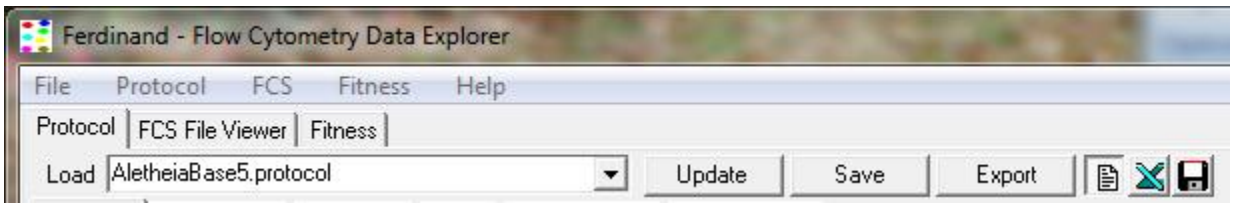
Ferdinand Protocols

What are Protocols?

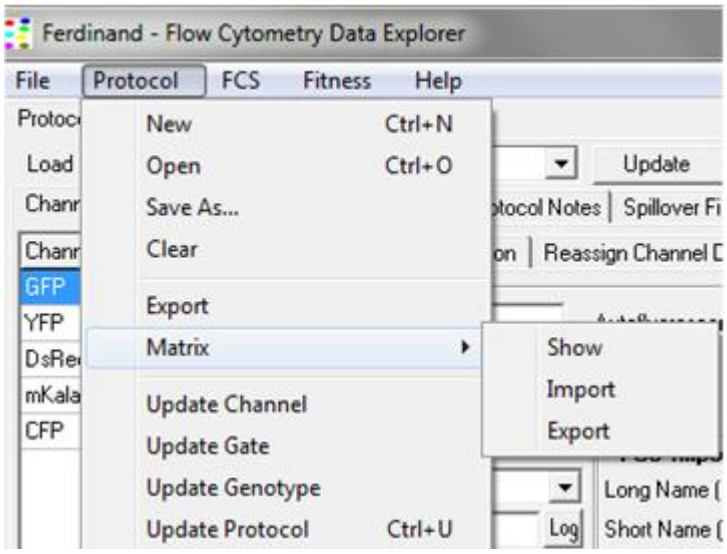
Protocols are instructions that tell Ferdinand how to read and manipulate data. It can open FCS files and allow one to view their contents without a Protocol. But filtering and scoring fluorescent events requires a Protocol. Protocols can be made *de novo* using the Protocol | New Menu command, but this is tedious and for experts only. A more robust method is to derive a protocol from an FCS file as described above.

Managing Protocols

Available protocols are listed in the Protocol List Box, located at the top of the Protocol Tab. Clicking on the arrow at the right-hand side of the box produces a list of Protocols that have been saved to the program file directory; selecting one of the entries automatically loads that Protocol. The name of the current Protocol is then listed at the base of the program window in the Status Bar. In the example above, the Protocol file AletheiaBase5.protocol has been loaded.



Adjacent to the Protocol List Box are three Buttons to assist in Protocol management. The Update Button saves on-screen information to the Protocol. The Save Button saves the Protocol to its *.protocol file; you will be prompted for a name if it has not been saved, otherwise the old version will be overwritten. The Export Button will export the data to the format specified (see the Export Options pages in the Overview section).



Protocols can also be managed from the Protocols Menu. Protocols can be saved to, and loaded from, *.protocol files. These are typically stored in the same directory as the Ferdinand program itself. These are binary files stored with a proprietary file structure.

The Protocol | New command creates a blank protocol. You will be prompted for the number of Channels and Gates. This command is for experts only, as getting everything right is not easy. It is safer to create a Protocol from a sample FCS file, as described in the Inspecting FCS Files | Creating Protocols from FCS Files section. As noted in the figure to the left, this Menu Command may be invoked by pressing the Ctrl-N Keyboard Shortcut.

The Protocol | Open Menu Item calls a File Open Dialog Box, allowing the user to specify a Protocol file to load. This is useful if the Protocol is not located in the same directory as the Ferdinand program file; otherwise, it's faster to select the *.protocol file from the Protocol List Box. As noted in the figure to the above, this Menu Command may be invoked by pressing the Ctrl-O Keyboard Shortcut.

The Protocol Clear command purges the current protocol from memory and erases on-screen content. This does not affect the *.protocol files.

The Protocol | Save as… command calls a dialog box to save the current Protocol. Different versions of Protocols are needed to interpret data from different flow cytometers, from the same flow cytometer which has save different data sets, or from the same cytometer but with different seed values for Gaussian curve fitting.

The Protocol | Export… command exports the current Protocol as a text file into the File Viewer or to Excel as specified by the Export Options Buttons at the top of the Protocol Tab. This is the only way to save Protocol information in a human-readable form; this text file cannot replace a proper *.protocol file for importing data into Ferdinand.

The commands under the Protocol | Matrix Menu Item performs commands specific to the Protocol | Matrix Tab. See the Protocol | Matrix section for details.

The Update Channel, Update Gate, Update Genotype and Update Protocol commands commit the information typed on-screen to the Protocol saved in memory.

The Protocol | Channel Tab

The Channel Tab is used to manage Channels, which correspond to parameters reporting signal strength on fluorescence detectors in flow cytometers. That is, the values on each Channel correspond to the signal reported by different detectors each optimized to detect a particular fluorophore. Often Channels are named after the fluorophore the cognate detector was designed to sense.

Two roles for Channel Data

One critical set of values stored in the Channel are the mean and variances for Gaussian curves fit to the distributions of fluorescence signal strength; these values are used in Z-Scoring. As discussed below, they are used for two purposes.

> ***Initial protocols***. Initial Protocols are ones used to read FCS files; here, the means and variances are used as seed values in finding Gaussian curves that fit the data distributions in the FCS file just opened. Without these seed values, Ferdinand would take a very long time exploring parameter space to find optimal fits.
>
> ***Final protocols***. Final Protocols are ones saved within *.Ferdinand files. Here, the means and variances of Gaussian curves are precisely those found above, and correspond exactly to the data set within the *.Ferdinand file.

Managing Channels

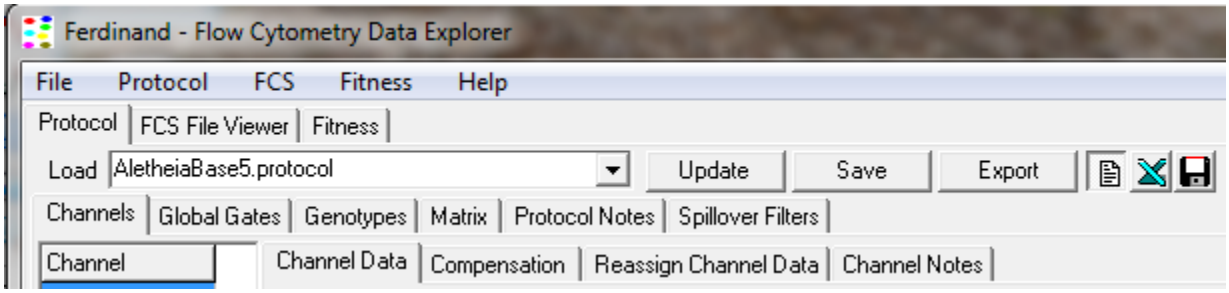
The number of Channels is specified at the time the Protocol is created; Channels cannot be added or removed after a Protocol is created. Each Channel is listed along the left-hand side of the Channels Tab in the Channel List. In the example on the left, there are five Channels, each named after the fluorescent protein they were optimized to detect.

There is no limit to the number of Channels that may be included, although there must be at least one; one Channel is not terribly useful, though, since Channels are used to discriminate among events with different fluorescent properties. Events may express more than one fluorescent protein, and therefore have strong signal on more than one Channel. The classification of events based on which Channels that have strong signal is accomplished through the use of Genotypes, discussed below.

Navigate among Channels using the mouse or arrow keys. Moving to a different Channel will display that Channel's information on screen. If on-screen values are changed, they are saved to the Protocol by either navigating to a new Channel or by pressing the Update Channel Button, which is located at the bottom of the Channel List, or by selecting the Update Channel Menu Item from the Protocol Menu.

A Wealth of Channel Data

Because Channels are central to the discrimination among fluorescent events, there are a large number of controls associated with their manipulation. These controls are spread across four panes within the Protocol | Channel Tab : Channel Data, Compensation, Reassign Channel Data, and Channel Notes. Click on the tabs to navigate to these subsections.

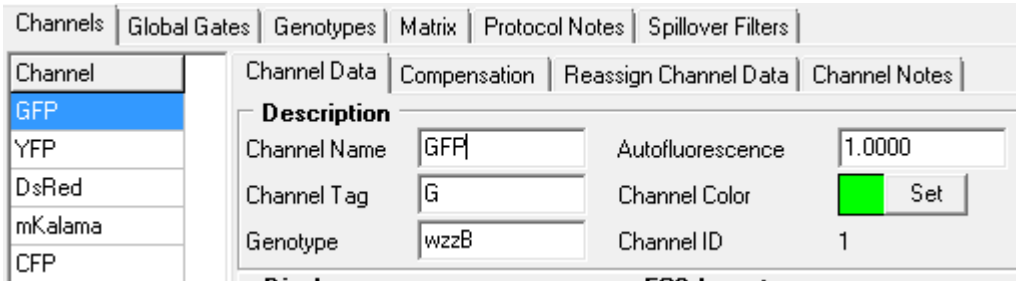


The Protocol | Channel | Channel Data Tab

This pane contains the primary descriptive information on the selected Channel, including lists of Gaussian curves fit to the distribution of data. These values are used as seed values when fitting curves, and contain final values of the mean and variance for curves fit to particular data sets for Protocols saved within *.Ferdinand files.

Channel Data | Description box

The Description Box allows access to values that control the display of Channel information. The **Channel Name** is used in reporting classification, plotting histograms and appearing as Menu Items for drawing plots. The Channel Name should be brief and descriptive. When Channels are created automatically from FCS files, the **Channel Name** is set initially to the **LongName** field of the cognate parameter.



The **Channel Tag** is a single character that is used in reporting classifications; make it one that will make sense.

The Genotype Edit Box can be used to accelerate the creation of Genotypes within the Protocol; see the Protocol | Genotype section below for a detailed description. Values entered here will be used to automatically create proper Genotypes for cells bearing single fluorescent proteins. This field has no meaning for Protocols used in the analysis of cells with multiple fluorescent proteins, where each Genotype has strong signal on multiple Channels.

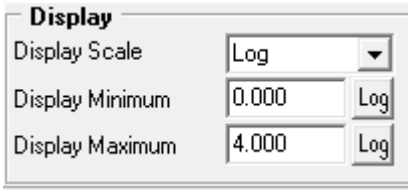
The Autofluorescence Edit Box is used to enter the **Autofluorescence** value for this Channel; this is used in advanced compensation algorithms currently available only for developers. This is disabled for users that have not entered the proper release code on the Preferences Form.

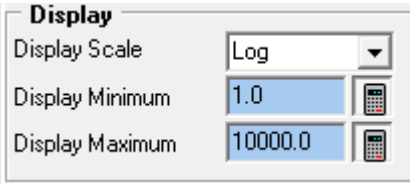
The **Channel Color** is used in plotting the distribution of values for events; this is done in the FCS File Viewer | Distribution Tab. To set the **Channel Color**, press the Set Button to the right of the rectangle and choose a color from the dialog box. The new color will be displayed in the rectangle. In the example above, distributions for the GFP Channel will be shown in green; this helps to minimize confusion when comparing distributions between Channels.

The **ChannelID** is an internal number used to control screen updates.

Channel Display Box

This box controls how values are displayed when histograms are being plotted. Fluorescence signal strength shows a Gaussian distribution when plotted on a log scale; therefore, it is recommended that the **Display Scale** always be set to "Log".





One the scale has been defined, the minimum and maximum values for display can be set. These do not affect data filtering or analysis in any way, solely the display of data on histograms on the FCS File Viewer | Distribution Tab. For each of comparison between files, it should be set to an objective range, no necessarily corresponding to the minimum and maximum values within any particular data file. In the example here, the values range from 1 to 10,000, or 10^0 to 10^4. Because the data

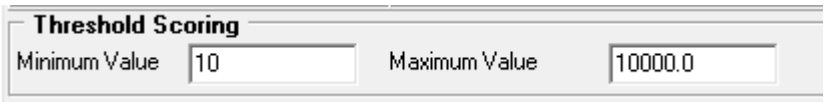
are displayed on a log scale, the values are set at 0.0 and 4.0. For those not comfortable entering values on a log scale, Ferdinand has a built-in Calculator. Press the Log Button next to the appropriate Edit Box and enter the value on a linear scale. In the example above, both Buttons were pressed, and values were converted to a linear scale. New values could now be entered on a linear scale, pressing the Calculator Button when finished. AS shown above, edit boxes with Calculators will always be shown with blue backgrounds when they are displaying values on a linear scale.

Channel Data | FCS Import Box

This box contains information which maps this Channel onto the cognate field of parameters within FCS files. There are two fields within the parameter that can be used to map a parameter to a Channel. The first is the **LongName**, encoded by the $PnS field; the second is the **ShortName**, encoded by the $PnN field (see the Inspecting FCS Files section for details). If a Protocol is created manually, inspect the FCS file header (see FCS File Viewer | Header Tab) to identify these fields in each parameter. If a Protocol is created automatically, these are completed for you. It is critical not to change these; critically, if these fields do not match between the Channel and a parameter in the FCS file, then the file cannot be processed.

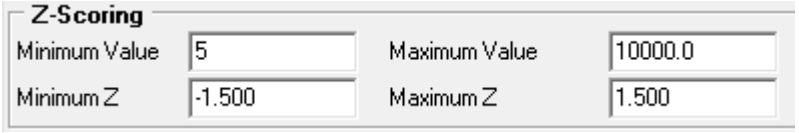
Some flexibility in matching Channels to parameters is afforded; if an exact match is required, check the appropriate Exact Check Box next to the field; both are checked in the example above. If the parameters are named slightly differently between FCS files (*e.g.,* different machines write slightly different names), uncheck the Exact Check Box. Here, the string entered must be contained within the **Long Name** or **Short Name** of the parameter field. For example, if the GFP detector is labeled "LogGFP" by one machine and "GFPLog" in another, uncheck the Exact Check Box and use "Log" for the **Long Name** in the Gate, which will match both; in both cases here, "GFP" is the **ShortName**. The combination of **LongName** and **ShortName** should allow precise matching.

Channel Data | Threshold Scoring Box



This box controls the range of values that are acceptable for the Channel. Values lower than that set in the Minimum Gate Edit Box, or higher than that set in the Maximum Gate Edit Box, will not be scored as being "positive" for this Channel during threshold scoring (see the Classifying Events section for details). These values *do not* affect Z-scoring. Typically, the threshold values are the same between Channels.

Channel Data | Z-Scoring Box



This box controls the range of values that are considered during Z-scoring. That is, events whose values lie outside these ranges cannot be classified into Genotypes defined using this Channel.

Similar to the values for Threshold Scoring above, absolute **Minimum Z Gate** and **Maximum Z Gate** values should be provided; is the events has a value for this Channel that lies outside this range, it cannot be classified as a Genotype defined using this Channel. These need not be rigorous, as the heavy lifting in terms of filtering events will fall on the **Minimum Z** and **Maximum Z** values.

The **Minimum Z** and **Maximum Z** values define the range of acceptable Z-scores for Genotypes defined using this Channel. The Z-score itself is calculated from the mean and variance of a Gaussian curve fit to the distribution of signal strength for events that pass all filters imposed by the Gates (see the Protocol | Gates section below). The compensated signal of a fluorescent event on each Channel is converted to a standard Z-score for event *I* on detector *j* as:

$$Z_{ij} = (x_{ij} - \mu_j)/\sigma_j$$

where x_{ij} is the compensated signal strength on Channel *j*, and μ_j and σ_j are the mean and standard deviation for events on Channel *j* originating from cells expressing the appropriate fluorescent protein (*i.e.*, not spillover).

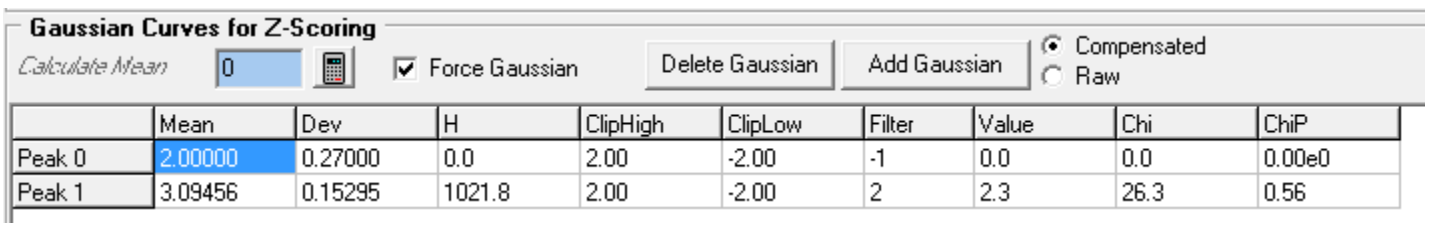
For Z-scoring to be valid, the **Minimum Z** and **Maximum Z** values should be the same across all Channels; otherwise, different fractions of events will be counted. If this is true, the proportions of each Genotype will be to be adjusted to compensate for this difference. That is really not worth the effort; it's best to keep these all the same.

Channel Data | Gaussian Curves for Z-Scoring Box

This box includes controls for managing the Gaussian curves that have been – or will be – fit to data. Two type of Gaussian curves as listed, and both have their uses.

Peak 0 is the first Gaussian curve; in the example below for DsRed, its values for mean and standard deviation – μ=2.0, or log(100) and σ=0.27 – are used as seed values when fitting curves to novel data; we know this curve was not generated from actual data since the maximum height of the curve (the column labeled "H") is 0.0. This Gaussian is useful because this protocol can be used to process data from large numbers of FCS files where the signal strength for DsRed is somewhere around 100.

There is another curve listed, Peak 1, which was fit to data; we know this was fit to actual data since the height ("H") is 1021.8 and non-zero. This has a very different mean, more than 1000 (log more than 3.0), with a narrower standard deviation. This curve was fit to actual data; notice the χ^2 goodness-of-fit value (in the Chi column) is low, with a non-significant P-value of 0.56. This indicates that this was a good fit. This curve would be used by the Genotype to evaluate Z-scores (see Protocol | Genotype section below).



	Mean	Dev	H	ClipHigh	ClipLow	Filter	Value	Chi	ChiP
Peak 0	2.00000	0.27000	0.0	2.00	-2.00	-1	0.0	0.0	0.00e0
Peak 1	3.09456	0.15295	1021.8	2.00	-2.00	2	2.3	26.3	0.56

Gaussian curves can be added manually by pressing the Add Gaussian Button; you may use the built-in calculator – shown with its usual blue edit box – to convert linear values to log values. To remove unwanted curves, press the Delete Gaussian Button.

All curves for Z-Scoring will be fit to compensated data, so the Compensated Radio Button should be selected. Developers working on native likelihood methods will have an active Raw Radio Button to fit curves to raw data.

The Force Gaussian Check Box is selected when the Gaussian curve is used to calculate Z-scores. If left unchecked, then Ferdinand will use the last Gaussian listed as seed values to fit curves to the data. The Force Gaussian Check Box should be unselected in *Initial Protocols* were the Gaussian values are seeds used to fit curves. The Force Gaussian Check Box should be checked in *Final Protocols* where the Gaussian values represent curves already fit.

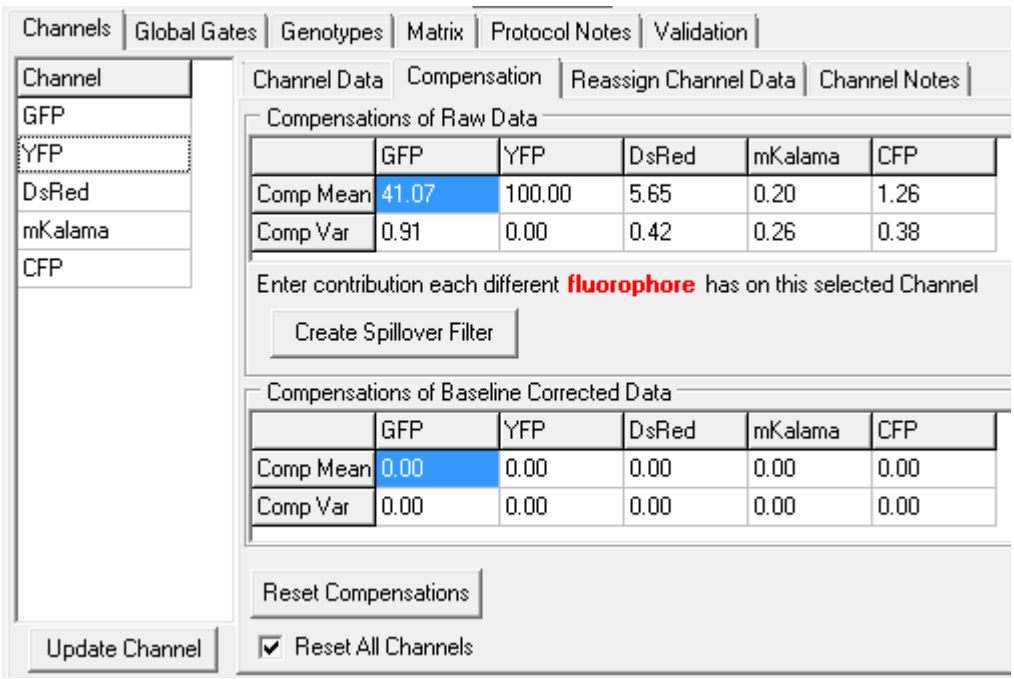
The Protocol | Channel | Compensation Pane

The Compensation Pane allows users to edit Spillover Coefficients and create Spillover Filters (see Protocol | Spillover Filters section below). There are two type of compensation matrices calculated by Ferdinand.

Compensation of Raw Data Box

The values shown in the Compensations of Raw Data Box – as shown in the example below – are those used in Z-Scoring. These can be entered by hand, imported *en masse* on the Protocol | Matrix Tab, or calculated by Ferdinand in the FCS File Viewer | Distributions Tab.

Each spillover is reported with a mean and variance. The columns represent fluorophores whose light can detected on this Channel. The maximal value is set to 100 and should correspond to the fluorophore which this Channel was designed to report. The other values should range between 0 and 100; values closer to zero indicate that the fluorophore does not shed light on this detector.



Compensation of Baseline Corrected Data Box

The values shown in the Compensations of Baseline Corrected Data Box are used in Baseline Compensation; this compensation Mode is available for developers. Developers should enter the Release Code in the Preferences Form to activate this feature.

In the example to the right, spillover coefficients have been entered for the YFP Channel in the Compensations of Raw Data Box; each value has a mean and a variance. The values for the Baseline Corrected Data Box are empty since this was not used in this Protocol.

Creating Spillover Filters

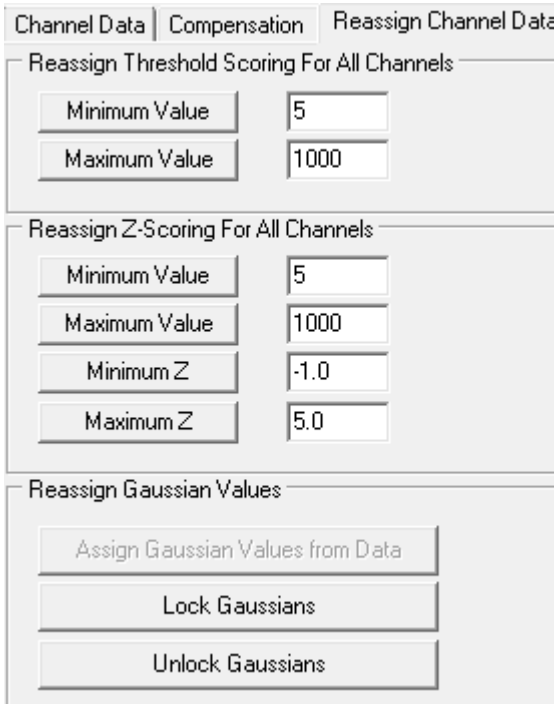
Ferdinand can use the mean and variance of the spillover coefficients to create Spillover Filters. These are used to examine events with strong signal on a detector to determine if they originated from the fluorophore that Channel was designed to report. In the example above, we see that the GFP fluorophore spills over onto the Channel designed here for YFP, specifically at an average of 41% of its strength on the GFP Channel. We can use this information to determine if strong signals on the GFP Channel indeed correspond to GFP, as we would expect these events to show about 41% raw signal strength on the YFP detector.

Resetting Compensations

To erase these values and being anew, press the Reset Compensations Button; this erases all data for this Channel – or for all Channels if the Reset All Channels Check Box is selected – and provides you with a blank slate. There is no way to reverse this action.

The Protocol | Channel | Reassign Channel Data Pane

These controls allow certain fields to be reset with every Channel; this is very useful for ensuring uniformity among Channels, especially when exploring classification schemes.



Assigning New Values

New values should be entered in the appropriate Edit Box; these values are supplied to each Channel once the Button to the left of the Edit Box is pressed.

In the example on the right, we can set the **Minimum Value** for Threshold Scoring among all Channels to 5.0; the value of 5 has been entered into the Edit Box to the right of the Minimum Value Button within the Reassign Threshold Gating For All Channels Group Box. Pressing the Minimum Value Button would assign this value to all Channels, not simply the Channel currently selected.

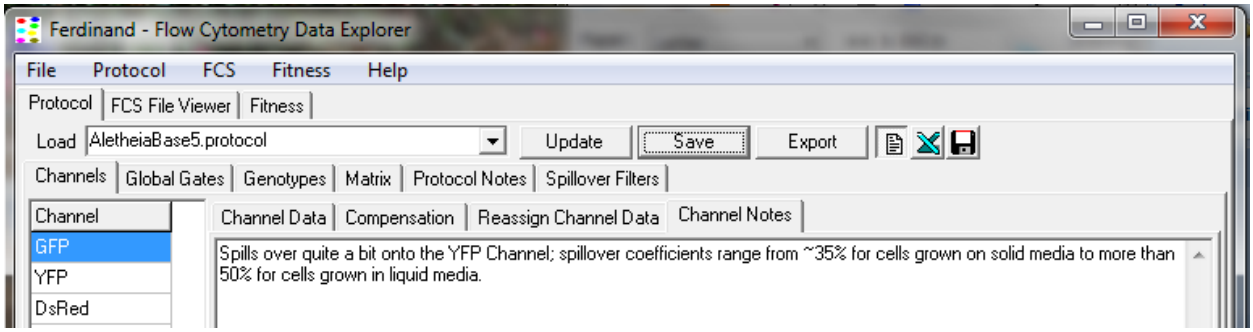
Using the controls on this pane, one can reset both the **Minimum Value** and **Maximum Value** for threshold scoring, thus providing uniform threshold across all Channels. One may also set the **Minimum Value** and **Maximum Value,** as well as the critical **Minimum Z** and **Maximum Z** values for Z-scoring; uniform values for **Minimum Z** and **Maximum Z** are required for Z-scoring to be accurate in Ferdinand; this is the best way to adjust those parameters.

Reassigning Gaussian Data

The Buttons in the Reassign Gaussian Values Group Box manipulate the Gaussian values in each Channel. The Assign Gaussian Values from Data Button will store the mean and variance from each distribution generated from the current FCS or Ferdinand file – which is visible on the FCS File Viewer | Distribution Tab. This is useful for storing these automatically-generated values if one is not using the Custom Detector to refine these fits. There must be an FCS or Ferdinand file loaded for this Button to be enabled; in the example above, there is no file loaded so the Button is disabled. One may also lock or unlock all Gaussians at once (see Channel Data | Gaussian Curves for Z-Scoring Box section for more information).

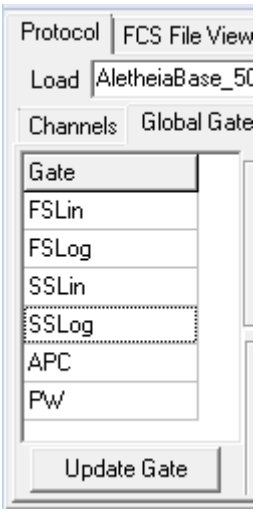
The Protocol | Channel | Channel Notes Pane

Use the Notes Memo on the Channel Notes Pane to record comments regarding particular Channels; you may also enter notes in the Protocol Notes Memo. As with other Notes Memos, there is no limit to the text that can be entered; however, only plain text is saved to the *.protocol file; all text formatting will be lost. An example of a **Note** is shown below:



The Protocol | Global Gates Tab

The Gates Tab is used to manage the Global Gates, which filter events to remove those not likely to result from cells of interest. Gates often correspond to optical parameters within FCS files, although fluorescent parameters (such as signal strength on detectors designed for nucleic acids stains) can be used. Critically, Gates are not used to discriminate among classes of fluorescent event; there are only used to cull events with atypical properties.



Managing Gates

The number of Gates is specified at the time the Protocol is created; Gates cannot be added or removed after a Protocol is created. Each Gate is listed along the left-hand side of the Gate Tab in the Gate List. In the example on the left, six Gates have been defined in this Protocol. Note that the fifth Gate is labeled "APC;" this Gate corresponds to signal strength on the fluorescence detector responsive to the Syto-62 nucleic acid strain. Since this is used to discriminate among cellular and non-cellular events – and is NOT used to discriminate among different classes of cellular events – it is used as a Gate, not a Channel.

There is no limit to the number of Gates that may be included, although there must be at least one. Including more Gates does increase the size of Ferdinand files that are saved, and increases the memory usage. Therefore, adding large number of Gates – especially if they are not used to filter unwanted events – is not recommended.

Navigate among Gates using the mouse or arrow keys. Moving to a different Gate will display that Gate's information on screen. If on-screen values are changed, they are saved to the Protocol by either navigating to a new Gate or by pressing the Update Gate Button, which is located at the bottom of the Gate List (see image above) , or by selecting the Update Gate Menu Item from the Protocol Menu.

Gate | Description Box

This box contains values that are used to display Gate information.

The **Gate Name** is a label used in all Tables, Reports, Graphs and Menus. It should be description but no overly long. When creating a Protocol from an FCS file (see the FCS File Viewer | Header Tab), this will be set from the *LongName* field of the parameter used to create the Gate.

The **Gate Color** is used in plotting the distribution of values for events; this is done in the FCS File Viewer | Distribution Tab. To set the **Gate Color**, press the Set Button to the right of the rectangle and choose a color from the Color Selection Dialog Box. The new color will be displayed in the rectangle.

The **GateID** is an internal number used to control screen updates.

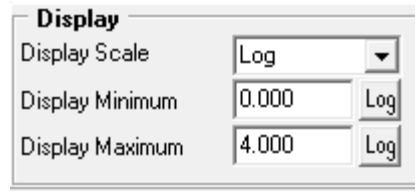
The Active Gate Check Box is used to indicate whether a Gate is used actively in filtering events. If this box is unchecked, the data are still stored in *.Ferdinand Files, and the distributions may be examined, but the values for the Gate for each event are not used to exclude them from analysis. Disabling a Gate in this fashion is to ignore its values in the analysis.

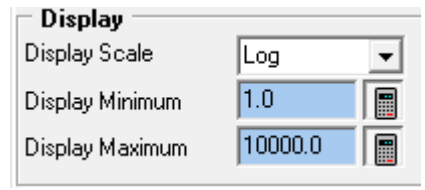
The Cull Gate Check Box is used to cull unwanted events from FCS files upon reading. Importantly there can be only ONE cull Gate in a Protocol (it's the Highlander of Gates). If value fall outside of the specified range (on the FCS File Viewer |

Load Tab), then the event is discarded. This is useful in culling large numbers of unwanted events when reading large files. Therefore, it should be set to the Gate with the largest discriminatory power. We often set it to the APC Gate, which read signal strength from the nucleic acid stain, thereby eliminating non-staining particles (dirt and agar).

Gate | Display Box

This box controls how values are displayed. The major decision is whether the values should be shown on a linear scale or a logarithmic scale. One can experiment here. Some values display well on a linear scale (*e.g.,* Pulse Width); other require a log scale to be useful at all. In the example on the left, the values are displayed on a log scale.

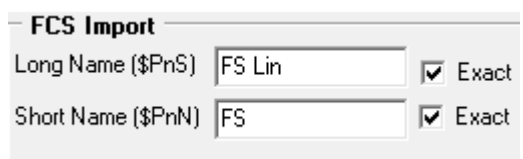




One the scale has been defined, the minimum and maximum values for display can be set. These do not influence data filtering or analysis in any way, solely the display of data on histograms on the FCS File Viewer | Distribution Tab. For each of comparison between files, it should be set to an objective range, no necessarily corresponding to the minimum and maximum values within any particular data file. In the example here, the values range from 1 to 10,000, or 10^0 to 10^4. Because the data are displayed on a log scale, the values are set at 0.0 and 4.0. For those not comfortable entering values on a log scale, Ferdinand has a built-in Calculator. Press the Log Button next to the appropriate Edit Box and enter the value on a linear scale. In the example above, both buttons were pressed, and values were converted to a linear scale. New values could now be entered on a linear scale, pressing the Calculator Button when finished. AS shown above, edit boxes with Calculators will always be shown with blue backgrounds when they are displaying values on a linear scale.

Gate | FCS Import Box

This box contains information which maps this Gate onto the cognate field of parameters within FCS files. There are two fields within the parameter that can be used to map a parameter to a Gate. The first is the **LongName**, encoded by the $PnS field; the second is the **ShortName**, encoded by the $PnN field. If a Protocol is created manually, inspect the FCS file header (see FCS File Viewer | Header Tab) to identify these fields in each parameter. If a Protocol is created automatically, these are completed for you. It is critical not to change these**; if these fields do not match between the Gate and a parameter in the FCS file, the then file cannot be processed**.



Some flexibility in matching Gates to parameters is afforded; if an exact match is required, check the appropriate Exact Check Box next to the field; both are checked in the example above. If the parameters are named slightly differently between FCS files (*e.g.,* different machines write slightly different names), uncheck the Exact Check Box. Here, the string entered must be contained within the **LongName** or **ShortName** of the parameter field. For example, if the Forward Scatter parameter is called FSLog by one machine and FS_Log in another, uncheck the Exact Check Box and use "FS" for the **Long Name** in the Gate, which will match both. The combination of **LongName** and **ShortName** should allow precise matching.

Gate | Threshold Gating Box

This box controls the range of values that are acceptable. Values lower than that set in the Minimum Gate Edit Box, or higher than that set in the Maximum Gate Edit Box, will be filtered out. That is, these events will not be considered when

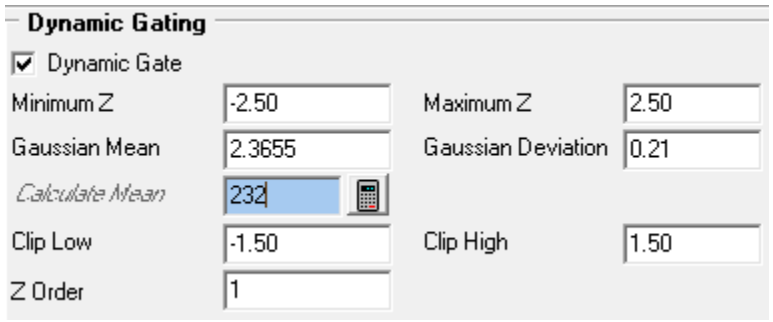
classifying events by the strength of their fluorescent signal. In the example below, events with Side scatter values lower than 75, or greater than 500, will be ignored.

Setting these parameters greatly affects the classification of fluorescent events, and these are the gatekeepers of valid events. If Gates are set too lax, one will be classifying noncellular events, such as cellular aggregates. Set to rigorous, and one could exclude important classes of cells; at best, one is reducing the number of events and therefore allowing stochastic variation to play a greater role in determining the ratios of classes of fluorescent events.

Gate| Dynamic Gating Box

There is more than one way to use values on Gates to filter unwanted events. The threshold gating scheme above uses set values that are applied; this is useful when the range of values is similar across many files (*e.g.,* Forward Scatter), although it could be tweaked. However, it is not useful if the range of value could vary greater between FCS files being examined using this Protocol. An example of such a variable Gate would be nucleic acid staining; fresh stain can yield string signal, whereas older stain, or sample exposed to light during transport, will show weaker signal. Static thresholds set above may be less useful.

In these cases, a dynamic Gate may be used. Here, A Gaussian curve is fit to the distribution of Gate values; needless to say, this is only useful is the distribution of values is Gaussian; this is true of signal strength from nucleic acid stains. Filtering by the dynamic Gate occurs in addition to filtering by the threshold parameters set above.



To enable dynamic Gating, select the Dynamic Gate Check Box, as is shown in the example above. The **mean** and **standard deviation** of this curve should be specified, as well as the **minimum Z** and **maximum Z** values that define the range of acceptable values. Events which fall outside this range of Z values (give the mean and variance defined), will be excluded from analyses.

The values for **mean** and **standard deviation** provided in the Protocol (as shown above), are used as seed values when Ferdinand fits a Gaussian distribution to the distribution of values form the FCS file is analyzes. The **Clip Low** and **Clip High** values specify the range of Z values used in χ^2 goodness of fit tests when finding the optimal Gaussian distribution; see the section on Custom Detector for a detailed explanation of how Gaussian curves are fit.

Finally, events excluded by a Dynamic Gate will affect the distribution of events that will be used to fit curves to the next dynamic Gate (if any); this is because the dynamic Gate may exclude events, which will affect distributions fit by the next dynamic Gate. Specify the order in which Gaussian curves dynamic Gates are fit by the Z Order field; lower numbers are fit first.

The Protocol | Genotypes Tab

The Genotype Tab manages the classes of fluorescent cells that are identified by Z-scoring. The likelihood of each Genotype in creating the set of compensated signals set for each event is calculated; the Genotype with the maximum likelihood – assuming the Z-scores for the Channels are within the scorable rage – is assigned to that event.

If classification is attempted prior to the creation of Genotypes, Ferdinand will assume that cells express single fluorophore, and a Genotype will be created that expresses the protein detected by each Channel. That is, if you have 5 Channels, then 5 Genotypes will be created, one for each Channel.

The list of Genotypes is shown at the left-hand side of the Genotype Tab. In the example shown to the left, there are five Genotypes, one for each Channel. The buttons below the list manage Genotypes.

Managing Genotypes

To create a new Genotype, press the Add Button; this will create a new Genotype using the information on screen. To delete a Genotype, press the Delete Button.

As with Channels and Gates, changes to on-screen information are not automatically saved. If you change on-screen information, you may update the information in Ferdinand by navigating to a new Genotype via the Genotype List, or by pressing the Update Genotype Button.

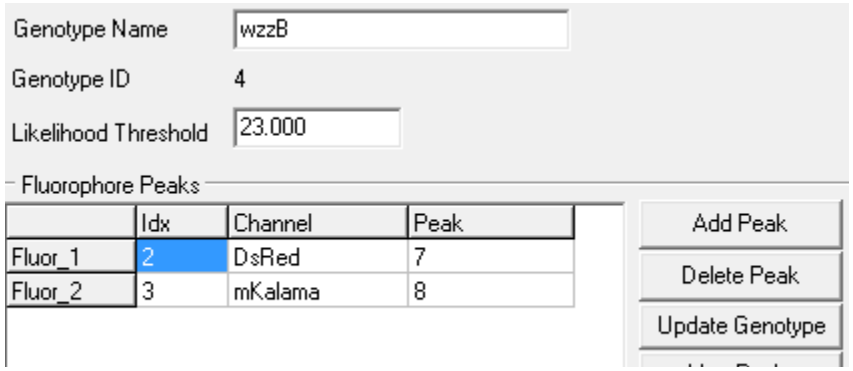
You may change the order of Genotypes by using the Up Button and Down Button. The order of Genotypes must be the same for calculating fitness via reciprocal experiments; for more information, see the Fitness section.

Genotypes may be imported as text form the Windows clipboard. Copy the Genotype to the clipboard and press the Import Genotype Button. Data should be organized as comma-separated values as `Name, Channel_Index, Peak_Index`. This is a rapid way to import a long list of Genotypes.

Managing Peaks

Fundamentally, a Genotype indicates which Gaussian distributions on which Channels are those produced by cells in its class. For cells with a single fluorescent protein, this should be a single Gaussian distribution on a single Channel; each Genotype should point to a different Channel. For cells bearing two fluorescent proteins, each Genotype will have two peaks, each pointing to a specific Gaussian distribution on a different Channel.

The example to the right shows the peak map for the *wzzB* Genotype in cells labeled with two fluorescent proteins. Here, two peaks have been added, indicating that *wzzB* cells express both the DsRed and mKalama proteins. More importantly, the Genotype indicates which Gaussian distribution on each Channel is appropriate to calculate the likelihood of this Genotype. As these are



doubly-labeled cells, each Channel will have many Gaussian distributions defined.

To add a peak to a Genotype, press the Add Peak Button; a new Peak will be added to the end of the list of Peaks. By default, this is assigned to the first Gaussian (Peak=0) of the first Channel. Change these values, where the first Channel is number zero (0) and the first Gaussian in any Channel in peak zero(0). After changing these values, either navigate to the new Genotype to update the values, or press the Update Genotype Button.

To delete a peak, press the Delete Peak Button.

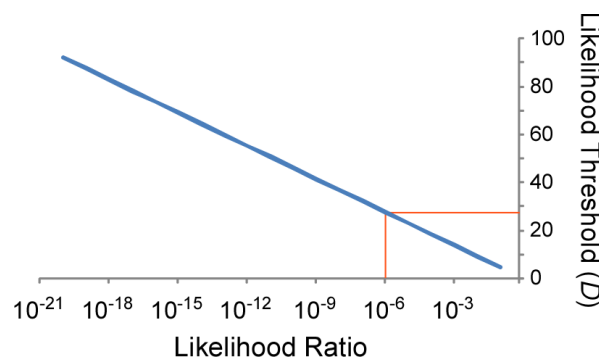
To ease the creation of Genotype for doubly-labeled cells, where each Channel will have many Gaussians defined, the Map Peaks Button can be useful. After indicating which Channels are used, the appropriate Gaussian curves are identified automatically. Here, the Gaussian in each Channel are search for the one most recently added that uses the other indicated Channel as a filter (see Custom Detector for details). In the example above, the Gaussian in the Channel 2 (DsRed) would be search for the one most recently added which was filtered by Channel 3, mKalama), and vice-versa.

Setting the Likelihood threshold

The **Likelihood Threshold** is used when classifying Genotypes by Z-Scoring. Events are typically assigned to the Genotype with the highest likelihood of generating the observed data. The **Likelihood Threshold** is used to determine if this Genotype is significantly more likely to produce the pattern of fluorescent signal than the other Genotypes. This is accomplished through a likelihood ratio test, where the likelihood ratio, ***D***, is calculated as:

$$D = -2*\mathrm{Ln}(L_1/L_0)$$

Where L_0 is the maximum likelihood and L_1 is the next-best likelihood among the genotypes examined. If the likelihoods are equal, then D is zero. As shown in the figure to the right, increasing values of the threshold (D) will require a greater difference in the difference between maximum and next-best likelihoods for the event to be scored.



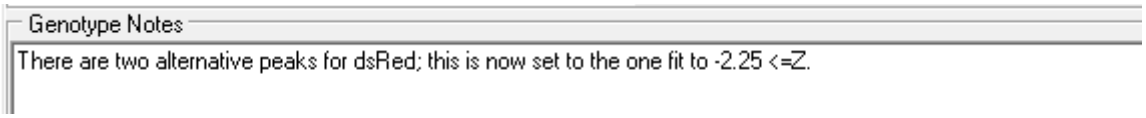
The appropriate threshold is up to the user. Because the models that are being tested have the same number of free parameters (they are two different genotypes), we cannot use a χ^2 distribution to assess the significance of D as we would in a conventional likelihood ratio test of nested models. Rather, a set threshold is chosen and events which fail to reach that threshold are excluded. As shown in the graph above, a threshold of 27.63 demands that the maximum likelihood be one million times more likely (likelihood ration of 10^{-6}) to be accepted; this is rather stringent. By default, the **Likelihood Threshold** is set to 0.0.

Genotypes for singly-labeled cells

As noted above, Genotypes are automatically created when none are defined by the user, where Ferdinand assumes cells are labeled with single fluorophores. Here, each Genotype uses the most recently-created Gaussian in each Channel. This, of course, may not be correct. Therefore, if users have entered multiple peaks, they should ensure that the Genotypes – which are created the first time that events are classified (see below) – point to the correct peaks. This is especially important is Gaussians are fit after the Genotypes are created; they must be updated to point to the proper Gaussians in each Channel.

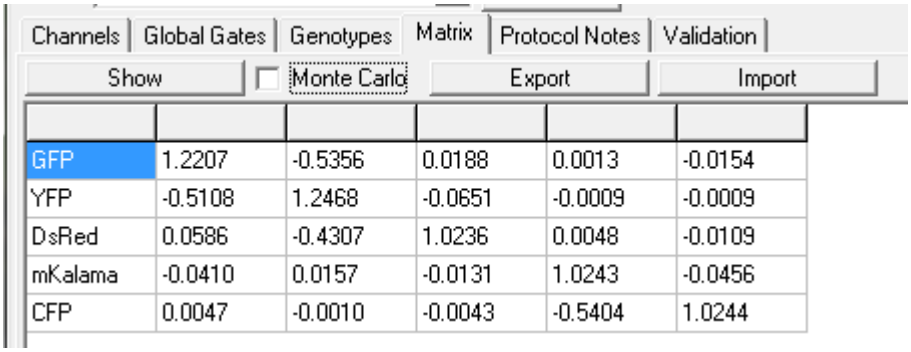
Genotype Notes

As with Protocols and Channels, users may maintain notes on each Genotype. This is useful when noting idiosyncrasies of which peaks are used, are shown in the example below.



The Protocol | Matrix Tab

This tab allows users to view the compensation matrix, which is calculated by matrix inversion from the matrix of spillover coefficients. To show the compensation matrix, press the Show Button. The matrix may be exported to Excel by pressing the Export Button. An example is shown on the right (these are the values shown for the compensation matrix in the example provided in the Overview section).



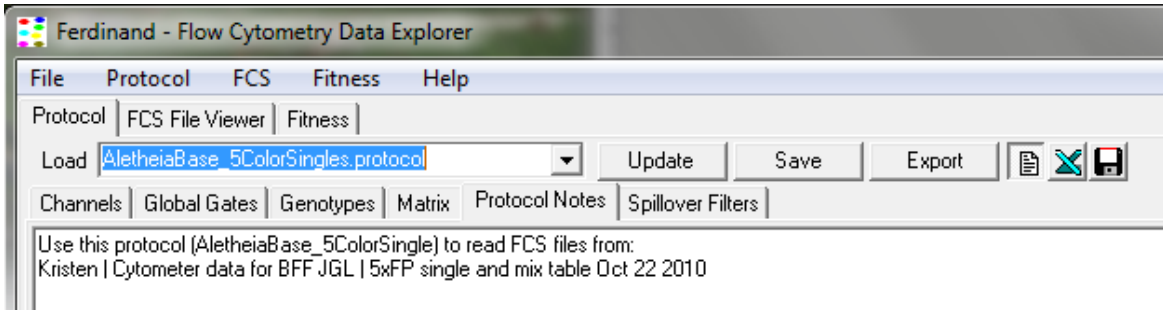
GFP	1.2207	-0.5356	0.0188	0.0013	-0.0154
YFP	-0.5108	1.2468	-0.0651	-0.0009	-0.0009
DsRed	0.0586	-0.4307	1.0236	0.0048	-0.0109
mKalama	-0.0410	0.0157	-0.0131	1.0243	-0.0456
CFP	0.0047	-0.0010	-0.0043	-0.5404	1.0244

If the Monte Carlo Check Box is selected, a variant of the compensation matrix will be selected by varying spillover coefficients according to their variances. This is what is done during dynamic compensation, which explores randomly generated compensation matrices.

A Spillover matrix may be imported from an outside source (*e.g.,* one generated by an outside program, or exported from another Protocol) using the Matrix Tab. Copy the matrix to the Windows clipboard as tab-separated values (which is the format when copying from Excel, for example). Copy the values using the Import Button on the Matrix Tab. Do not include any row or column headers; the text should be only an *NxN* matrix of floating point numbers, which each Channel as a column, as is found in the exported Protocols described above.

The Protocol | Protocol Notes Tab

Use the Notes Panel to save notes to the *.protocol file. You may type anything you wish into the Notes Memo; there is no limit to the text that may be entered.



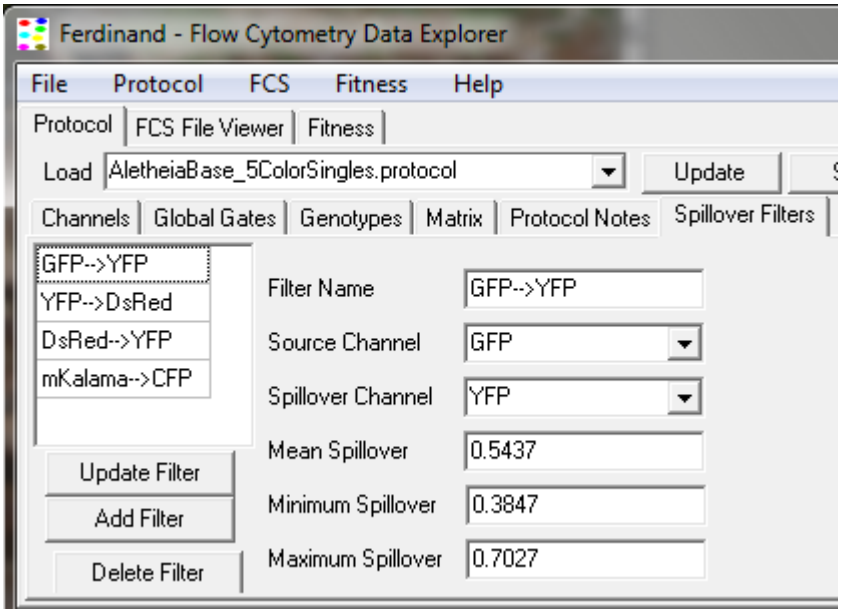
The Notes Memo saves plain text only; formatted text (*e.g.,* from an RTF or Word document) pasted into it will lose its formatting. Useful notes could include the date or creation, the intended use, notes on how the spillover coefficients were calculated, or notes comparing it to other similar Protocols. An example is shown above.

The Protocol | Spillover Filters Tab

Use the controls on the Spillover Filters Tab to create filters using the spillover of signal from one fluorophore onto alternate Channels.

Creating Spillover Filters

Ferdinand can use the mean and variance of the spillover coefficients to create Spillover Filters(see the Protocol | Channel | Compensation Pane section above). These filters are used to examine events with strong signal on a detector to determine if they originated from the fluorophore that Channel was designed to report. In the example discussed in the section above, the GFP fluorophore spilled over onto the Channel designed for YFP. We can use this information to determine if strong signals on the GFP Channel indeed correspond to GFP.



In the Channel | Compensation Pane, we could press the Create Spillover Button to generate a Spillover Filter. The advantage to that method was that the range of acceptable values was generated automatically from the mean and variance of the spillover measured from a particular file (see the FCS File Viewer | Distributions section for details on generating spillover coefficients from data).

We may also create or edit Spillover Filters here. One filter has been selected in the example on the right. This one describes the spillover from YFP onto the DsRed Channel. The **Mean Spillover** is 37.1%, which is the value used to create the compensation matrix. The important values for effective use of this filter are the **Minimum Spillover** and **Maximum Spillover** values; these encompass the acceptable range of values we expect for different events; these were set to represent 95% of events, as calculated from the variances of the spillover coefficients. One may set these values manually as well.

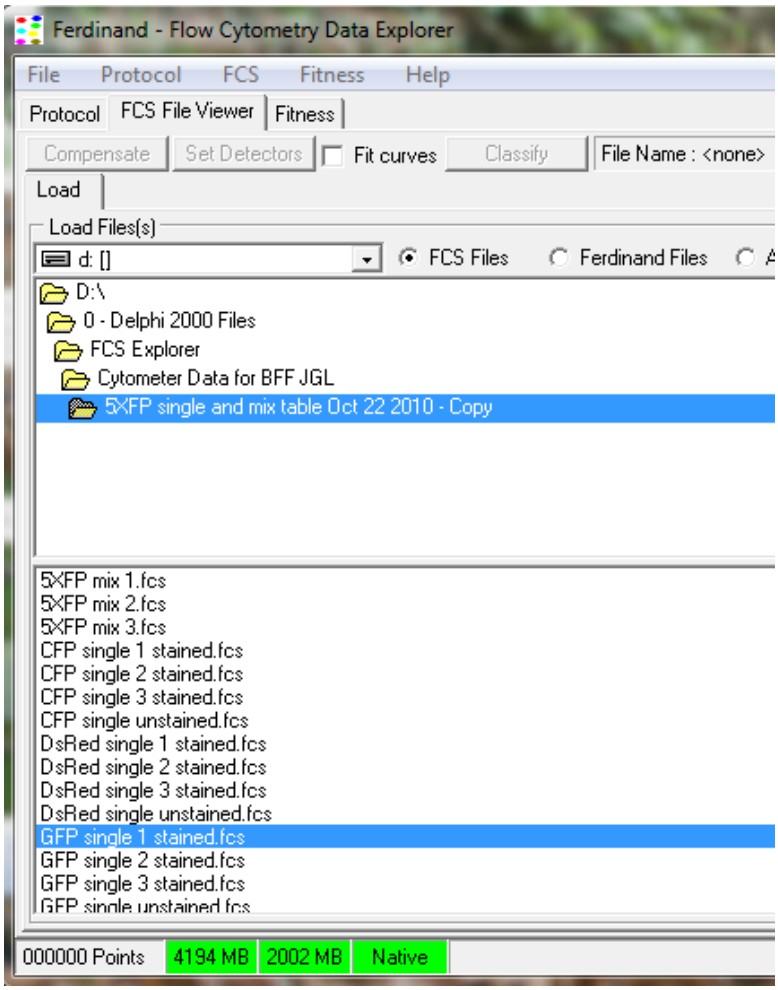
Managing Spillover filters

After editing values for a Spillover Filter on screen, press the Update Filter Button to save those values to the Protocol. You may create a new filter manually by changing the on-screen information and pressing the Add Filter Button. Select the source and spillover Channels, enter the appropriate range of values and you are set. Unwanted filters may be removed by pressing the Delete Filter Button.

Managing FCS Files

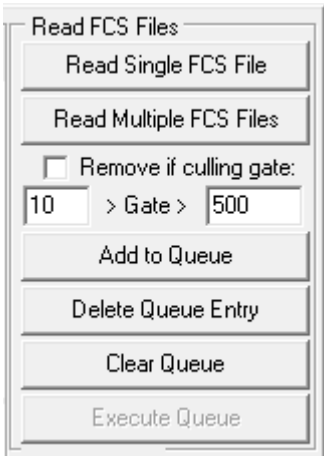
Reading a single FCS file : The FCS File Viewer | Load Tab

Before Ferdinand can analyze FCS files, it needs a Protocol that allows it to decipher the FCS data. This Protocol is made from FCS files themselves. To begin, read an FCS file. Move to the FCS File Viewer pane in the main program window and navigate to the Load Tab within it. When the program is first run, and no data have been analyzed yet, the Load Tab is the only tab visible. Other tabs will be uncovered when data are read and analyzed.



Select the Drive and Directory containing the FCS files you wish to examine. First choose the drive using the Drive Drop-Down Box. After a drive is selected, a directory tree will be displayed below it. Navigate the Directory Tree by double-clicking the folder icons. AS each directory is chosen, a list of files is displayed in the File List Box below the Directory Tree. Select the file of interest by clicking its name with the mouse. To the right, the figure shows a listing of FCS files with one highlighted, ready to be loaded.

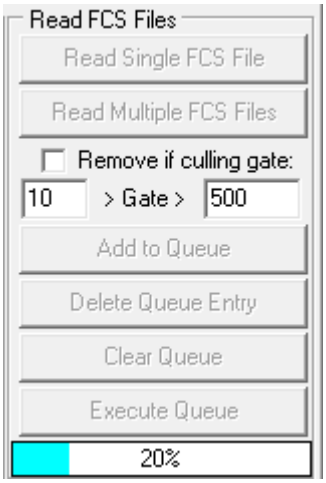
Be sure to click the FCS Files Radio Button so that the appropriate files are listed. This filters the files listed so that only those with the *.fcs file extension are shown. If you've saved your files using another file extension, click the All Files Radio Button instead. Ferdinand saves the last directory from which you loaded an FCS file, so it will automatically navigate to this directory when you run the program again.



You may then load this file by double-clicking the file name itself, or by clicking the Read Single FCS File button on the control panel to the right; these buttons are shown in the figure on the left. There are a lot of other controls there, and we will discuss them below. If you are not interesting, skip ahead to the next section to learn how to make Protocols from FCS files after opening one.

Reading Multiple FCS files

Ferdinand can read data from multiple FCS files and analyze them *en masse*. Simply select multiple files by holding the shift key down as you select. To select blocks of files, hold the Alt button as you click. When all files have been selected, press the Read Multiple FCS Files button. As shown to the right, all control buttons will be inactivated while the files are being read and a small Gauge will show progress through reading the set of files; here, it is reading the first of 5 files.



Reading multiple files is useful for analyzing replicate data. Ferdinand retains the file of origin for each event and, as discussed below, classifies events within each individual file as well as for the data set as a whole. In addition, Ferdinand can save the entire data collection in a single file.

Because reading multiple FCS files can lead to very large data sets, users may wish to remove events that are clearly not of interest. One way is to cull values that fall outside a desired range for a specified culling gate. This is activated by checking the Remove if Culling Gate Check Box. Consult the Protocol | Gates section to learn more. In the example above, the Remove if Culling Gate Check Box is not checked, so no events are culled. If it were checked, then events whereby the value for the specified Gate were less than 10, or greater than 500, then the event would have been discarded.

Saving Ferdinand files

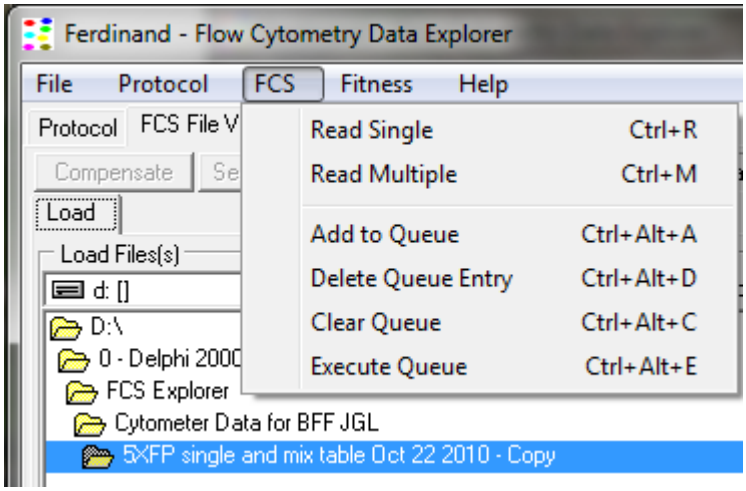
FCS file(s) that have been read by Ferdinand can be stored in a compact fashion in *.Ferdinand files. See the Managing Ferdinand Files section below on for more information. Files are saved via the File | Save As… Menu Item.

As detailed below – see the Managing Ferdinand Files section – Ferdinand files afford several advantages for storing data, including:

- The Protocol is saved with the data
- The data are saved only for a subset of events, those with useful data.
- Data from multiple files are combined.
- Data are compressed.

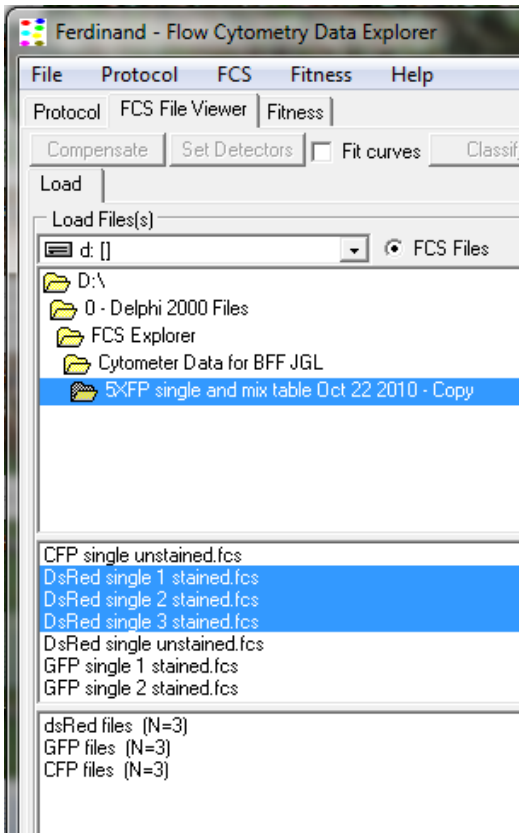
The FCS Menu

You may also manage reading FCS files from the File Menu and the FCS Menu. The FCS Menu contains commands for reading FCS files, either single files, multiple files, or as a queue of separate groups of files (see the Reading Series of FCS files via the Ferdinand Queue section below). The FCS Menu is shown to the right. All commands are available as keyboard shortcuts (see the Program Interface | Keyboard Shortcuts section for details). The File Menu allows you to save your FCS data in *.Ferdinand files (see the File Menu section below).



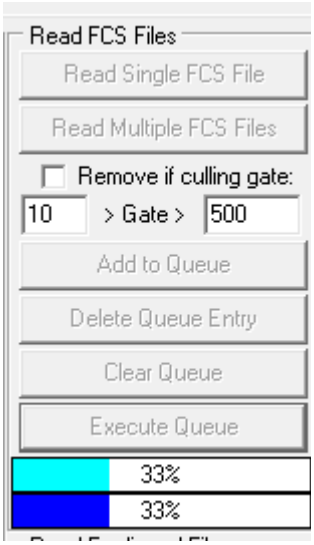
Reading Series of FCS files via the Ferdinand Queue

Large numbers of FCS files may be read and analyzed at once using the Ferdinand Queue. Effective use of the queue requires a Protocol to be used. Here, sets of files are read, filtered and saved to individual Ferdinand files. This is very useful if large numbers of files are to be analyzed. The task of reading them, filtering unwanted events and analyzing the combined data sets – a task which may take many hours – may be automated with the Ferdinand queue.



First, select the files you wish to analyze as described above for Reading Multiple FCS Files. Do not open them. Specify the Protocol as described below. Then click the Add to Queue button. You will be asked for a name of this queue entry. After it has been named, it is added to the Ferdinand queue, and a list of queued jobs appears below the file list. You may then continue to select sets of files – each associated with the appropriate Protocol – and add them to the queue. In the figure to the right, the queue has three entries, each with three sets of files being read.

To remove a queue entry, highlight it on the list of queue entries and click the Delete Queue Entry button. This will remove this list of files from the queue. To remove all queue entries, click the Clear Queue button. This will also hide the list of queued jobs.



Once all queue entries have been specified, press the Execute Queue button; this button is only enabled when the job queue has one or more entries. Ferdinand will read each set of files in turn, filter events as described below, and analyze the data using the Protocol specified when the queue entry was created.

As the queue runs, a second Gauge will appear showing progress through the queue itself. As above, the Cyan gauge show progress through the list of files in that queue entry. The Navy gauge show progress through the queue as a whole. The figure to the left shows that the first of three files is being read from the first of three queue entries. Gauges will be hidden when the queue has been read completely.

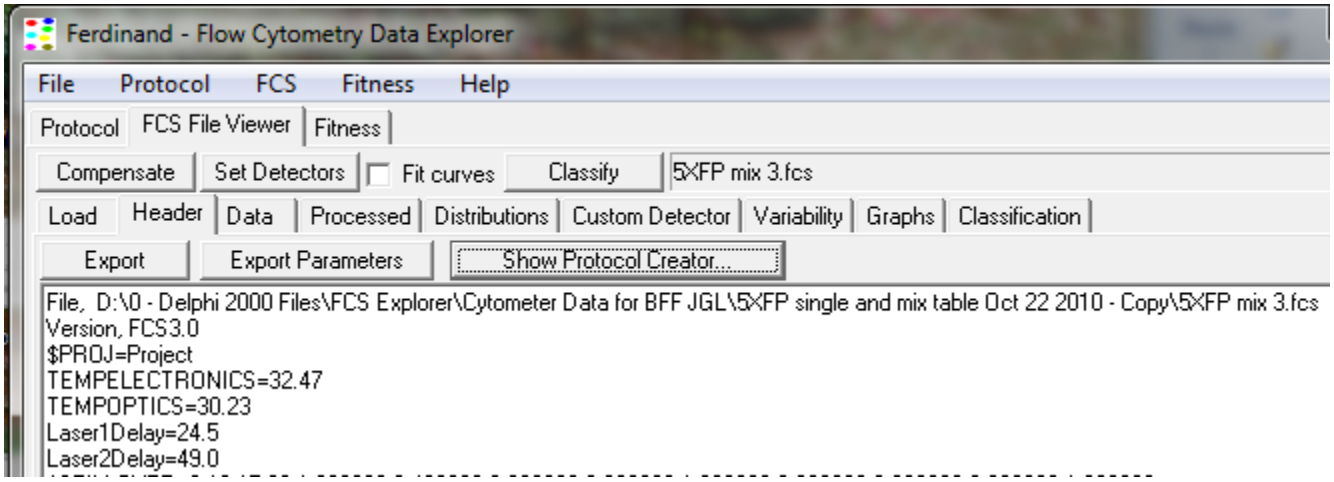
When the queue is complete, an output file will be generated which has the results of the analysis of all of the queue entries. This is shown in the File Viewer, which is described above.

Inspecting FCS Files

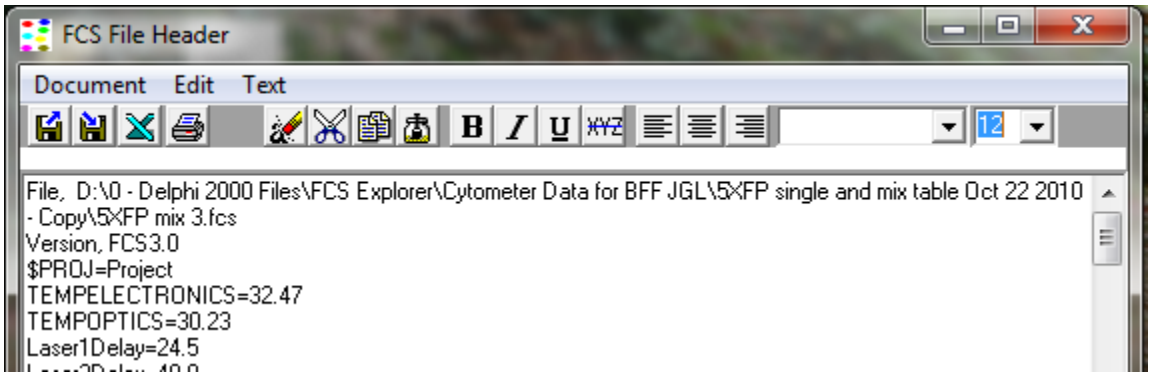
When a single FCS file is read, its header and internal data is displayed. This is done even when a Ferdinand Protocol is not available. When a file is read, additional tabs appear under the FCS File Viewer Pane.

The FCS File Viewer | Header Tab

Below, we have selected the Header Tab. The first line shows the file that is being displayed (here, 5XFP mix 3.fcs), as well as a large amount of rather esoteric information. Unless you are interested in FCS files, this is of no interest. What is important, though, is that this information is used to create an initial Ferdinand Protocol



The Export Button sends this data to a File Viewer, from which one can save or print the data. An example is shown below:



The buttons can be used to save or print the text data. This is a common File Viewer Form that is used for all text data.

Each parameter is described by several key properties: **LongName** ($PnS), **ShortName**, **Bytes** ($PnB), **Range**, **Scale** and **decades**($PnE), **Filter** ($PnF), **Gain** ($PnG), **Wavelength** ($PnL), **Power** ($PnO), **PercentLight** ($PnP), **DetectorType** ($PnT) and **Voltage** ($PnV), all of which are coded within the header. These are parsed by Ferdinand to create a Protocol.

The Export Parameters Button saves a complete parameter description as a comma-separated value (CSV) file. This can be read by many programs (including Microsoft Excel), and used to create protocols. Below, the data have been exported to Excel.

A1 | fx | Idx

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Idx	LongNam	ShortNam	Bytes	Range	Scale	Minimum	Decades	Filter	Gain	Wavelen	Power	PercentLi	Detector1	Voltage
2	1	Time LSW	TIME	16	65536	Linear	na	na		0	0	1	0		400
3	2	Time MSW	Time MSW	16	65536	Linear	na	na		0	0	1	0		400
4	3	Pulse Wic	Pulse Wic	16	512	Linear	na	na		0	0	1	0		400
5	4	FS	FS Lin	16	65536	Linear	na	na		488	0	30	0		20
6	5	FS	FS Area	16	65536	Linear	na	na		488	0	30	0		0
7	6	FS	FS Log	16	65536	Log	0	4		488	0	30	0		0
8	7	SS	SS Lin	16	65536	Linear	na	na		488	0	10	0		659
9	8	SS	SS Area	16	65536	Linear	na	na		488	0	10	0		659
10	9	SS	SS Log	16	65536	Log	0	4		488	0	10	0		659
11	10	GFP	FL 1 Lin	16	65536	Linear	na	na		488	0	1	0		470
12	11	GFP	FL 1 Area	16	65536	Linear	na	na		488	0	1	0		470
13	12	GFP	FL 1 Log	16	65536	Log	0	4		488	0	1	0		470

The LongName and ShortName fields for each parameter are supplied by the Flow Cytometer itself, typically through the software used to operate the machine. The number of parameters, their ranges and other properties are set by the software used to operate the flow cytometer and generate the FCS file itself. These cannot be changed by Ferdinand.

The Show Protocol Creator… Button is used to create protocols using the parameter list of this FCS file. The resulting Protocol can be used to analyze and FCS file which has similar parameters. See the next section for detailed instructions.

Creating Protocols From FCS Files

The Show Protocol Creator… button is used to create protocols using the parameter list of this FCS file. Press the button to display the Protocol Creation Panel within the Header Tab of the FCS File Viewer Tab. An example is shown to the right. The **LongName** and **ShortName** fields for each parameter are listed. You must specify at least one Gate and at least one Channel to create a valid Protocol.

2 Channels
3 Gates

Create Protocol
☐ Save after creation
☐ Process FCS Data immediately

Long Name	Short Name	Gate/Channel
Time MSW	Time MSW	--
Pulse Width	Pulse Width	GATE
FS	FS Lin	--
FS	FS Area	--
FS	FS Log	GATE
SS	SS Lin	--
SS	SS Area	--
SS	SS Log	GATE
GFP	FL 1 Lin	--
GFP	FL 1 Area	--
GFP	FL 1 Log	CHANNEL
YFP	FL 2 Lin	--
YFP	FL 2 Area	--
YFP	FL 2 Log	CHANNEL

Selecting Gates

Select the cell in the Gate/Channel column of the parameter of interest and double-click it to specify that parameter as a Gate. In the example to the right, three Gates have been specified: pulse width, the log of Forward Scatter and the log of Side Scatter. All events have valid data for each Gate; these values may be used to filter all events. Gates may be optical data like Forward Scatter, or fluorescence data that would be applicable to all cells, such as a fluorescence Channel recording the SYTO-62 nucleic acid stain. The number of Gates created is listed at the top of the Protocol Creation Panel.

Selecting Channels

Select the cell in the Gate/Channel column of the parameter of interest and double-click it ***twice*** to specify that parameter as a Channel (in practice, this toggles the parameter first to a Gate, then to a Channel). In contrast to Gates, Channels specify fluorescence data that vary between classes of cells. In the example above, two Channels have been specified, once for the log of peak height on a detector designed for GFP, and one for a detector designed for YFP. Events are classified by their signal strength on Channels, so that different classes of events show different patterns of

signal on different Channels, even though their signal on Gates may be nearly identical. This is what makes Channels very different from Gates. The number of Channels created is listed at the top of the Protocol Creation Panel.

Deselecting Gates and Channels

To deselect a Gate or a Channel, continue to double-click the appropriate cell until it is blank. Cell status rotates through the states of *unchecked-Gate-Channel-unchecked* as one continues to double-click it.

Generating the protocol

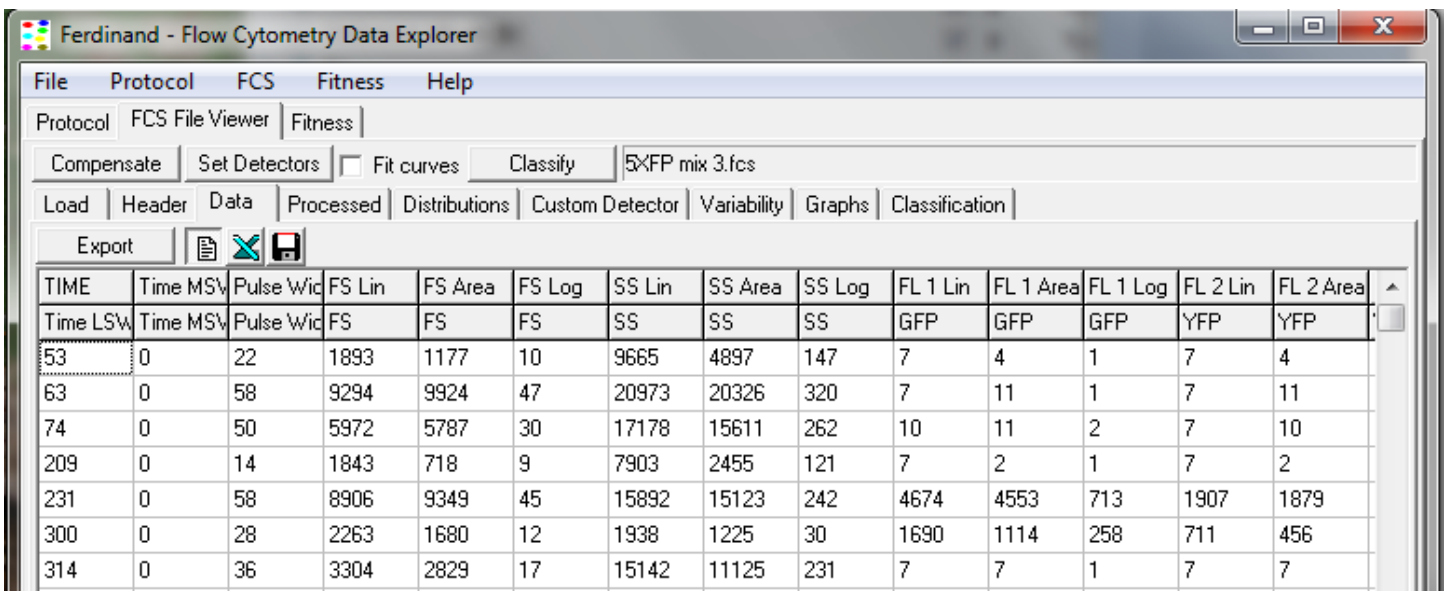
After the Gates and Channels have been selected, press the Create Protocol button; this will be active only when parameters used to define at least one Gate and one Channel have been specified. This will create a Protocol using the settings for the parameters selected.

You may save the Protocol automatically by selecting the Save after creation Check Box before generating the Protocol. Otherwise, you may save the Protocol from the Protocol Tab after editing.

If you wish to have the protocol immediately applied to the FCS data, select the Process FCS data immediately Check Box before generating the Protocol. In this way, the distributions of values – see the FCS File Viewer | Distributions Tab section for details – may be used to define the limits of the Gates and establish the compensation matrix using spillover coefficients calculated by Ferdinand.

The FCS File Viewer | Data Tab

Below we have selected the Data Tab. Here, the data for all parameters for each event is listed. The top two rows indicate the **LongName** and **ShortName** for each of the parameters. Each event is shown on a single row.



TIME	Time MSV	Pulse Wid	FS Lin	FS Area	FS Log	SS Lin	SS Area	SS Log	FL 1 Lin	FL 1 Area	FL 1 Log	FL 2 Lin	FL 2 Area
Time LSV	Time MSV	Pulse Wid	FS	FS	FS	SS	SS	SS	GFP	GFP	GFP	YFP	YFP
53	0	22	1893	1177	10	9665	4897	147	7	4	1	7	4
63	0	58	9294	9924	47	20973	20326	320	7	11	1	7	11
74	0	50	5972	5787	30	17178	15611	262	10	11	2	7	10
209	0	14	1843	718	9	7903	2455	121	7	2	1	7	2
231	0	58	8906	9349	45	15892	15123	242	4674	4553	713	1907	1879
300	0	28	2263	1680	12	1938	1225	30	1690	1114	258	711	456
314	0	36	3304	2829	17	15142	11125	231	7	7	1	7	7

Press the Export Button to send these data to a spreadsheet, or to an external program for analysis. Choose the destination using the Output Options Selector Buttons to the right of the Export Button. Keep in mind that the fluorescent signal strength (*e.g.,* the columns for GFP and YFP above) are raw data and have not been compensated for spillover. See the What is spillover? section for more information.

Managing Ferdinand Files

What are Ferdinand Files?

Ferdinand stores cytometric data in native *.Ferdinand files. This is a proprietary, compressed binary file format storing all analytical data. There are many advantages to using Ferdinand files to store cytometric data:

1. The Protocol is saved with the data. In this way, the correct Protocol – including Gaussian curve fits, is saved right with the data. Moreover, reanalyzing the data is faster since the Gaussian curves do not need to be found again; their values are already stored. This is especially important when analyzing cells with multiple fluorescent proteins, where many Gaussian curves are fit for each Channel.
2. The data is saved for each event, but it is a smaller data set than saved in the FCS file. FCS files often bear large numbers of parameters. Only a subset of these parameters is stored in a Ferdinand files as Gate and Channel data. Other data are excluded.
3. Only a subset of events is saved. First, events with minimal signal are culled; this threshold is set in the Preferences. This is set low enough to avoid discarding any remotely useful event, but high enough so that true noise events (*e.g.,* those stemming from dust) are discarded. With noisy data sets, the majority of events may be discarded. Second, subsets of events which fail a Culling Gate are also discarded, further reducing file size.
4. Data from multiple files are combined. Once merged into a single Ferdinand file, data management is easier.
5. Data are compressed. Ferdinand files are naturally data compressed, thereby saving disk space. These features combined can often reduce file sizes to less than 3% of the space taken by native FCS files, without the loss of any significant data.

Saving data in Ferdinand files

After reading FCS files and processing them, data may be saved via the File | Save As… Menu Command. Files can be stored in any location. Ferdinand remembers the location of the last *.Ferdinand file read, and uses this to control the directory listing on the FCS File Viewer | Load Tab.

Loading Ferdinand files

Ferdinand files may be loaded two ways. First, one can locate the file using the File | Load Menu Command. Alternatively, one may navigate to the location of the Ferdinand file using the file listing on the FCS File Viewer | Load Tab. Navigate to this tab and, as shown to the right, select Ferdinand Files from the list of radio buttons. This will force the File List Box to show only *.Ferdinand files. Next, locate the file and open it by either double-clicking on the file name, or press the Load Ferdinand File Button located in the control panel to the right of the File List Box.

Appending Ferdinand files

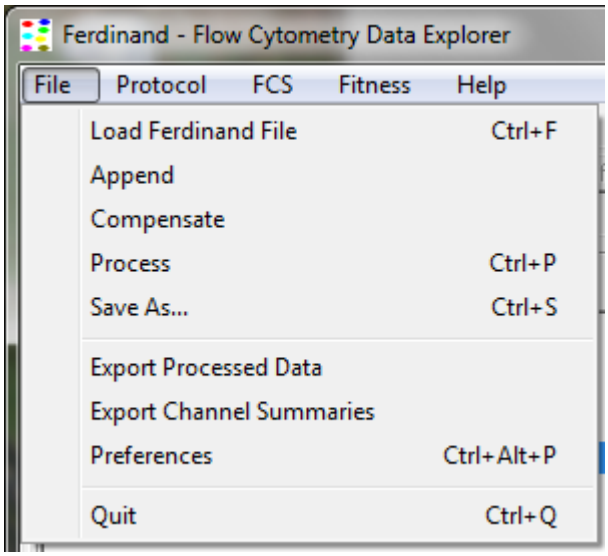
Rather than loading a Ferdinand file alone, one may append its data into the data set already open, bypassing the Protocol in the saved Ferdinand file and using the current Protocol instead. To append data from a saved Ferdinand file to the current data set, select the Ferdinand file as above and click the Append Ferdinand Files Button.

Peeking into Ferdinand files

Opening large Ferdinand files can take some time. If the only details of interest are Channel summaries, one may export these from the embedded Protocol using the Export Channel Summaries Button. This is useful for scanning libraries of Ferdinand files to locate one with desired properties.

The File Menu

The File Menu allow you to Load, Save and Append *.Ferdinand files. In addition, you may process, compensate and export the data that has been loaded from either a *.Ferdinand file or an FCS files.



In addition to these data handling functions, you may access the Preferences Form to set global Ferdinand settings (see the Preferences Form section).

Many of the common menu commands may also be invoked through their keyboard shortcuts.

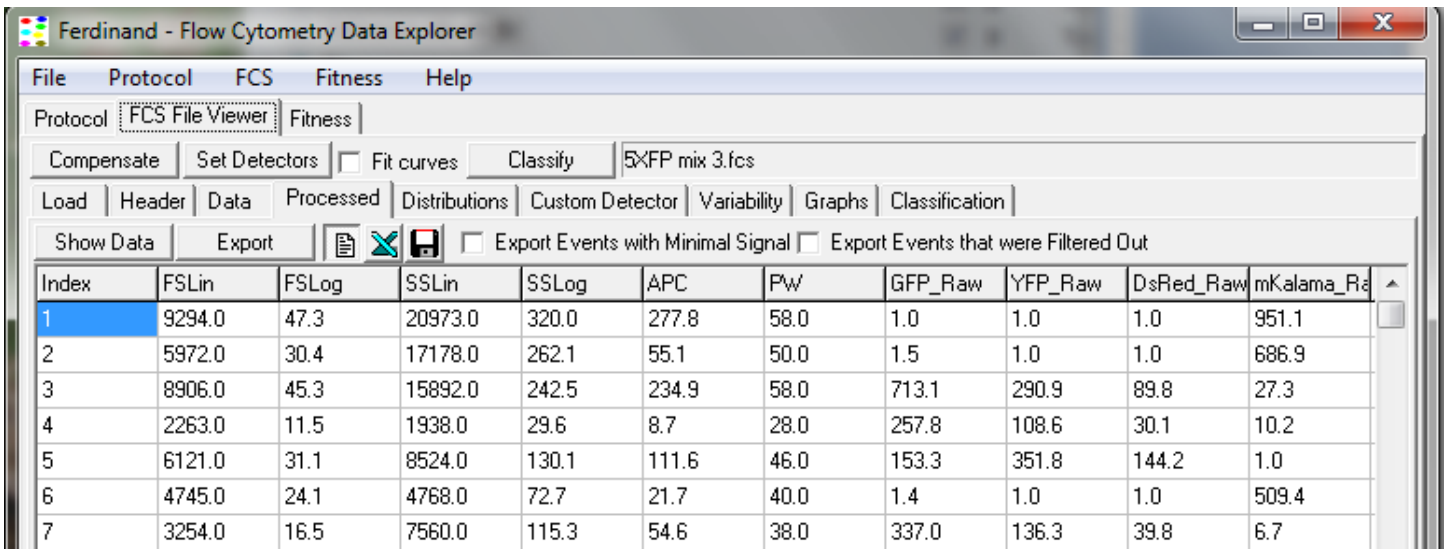
Processing FCS & Ferdinand Files

Is a Protocol is loaded and active, the data read from an FCS file will be processed. That is, the compensation matrix will be applied, compensated signals will be assessed, and Genotypes will be classified accordingly. Data may be inspected in a number of ways.

The FCS File Viewer | Processed Tab

The processed data is shown on the Processed Tab. Whereas the Data Tab showed values for all parameters in an FCS file, only the values for Gates and Channels are shown here.

An example is shown below. The first column shows the index of the data point. After this, the data for each Gate is shown; in this example, six Gates were specified: FSLin, FSLog, SSLin, SSLog, APC (for the Syto-62 dye), and PW (pulse width). After these, values for raw and compensated signal on each Channel is show, first raw values then compensated values. There are five Channels specified in this Protocol: GFP, YFP, DsRed, mKalama and CFP.



Index	FSLin	FSLog	SSLin	SSLog	APC	PW	GFP_Raw	YFP_Raw	DsRed_Raw	mKalama_Ra
1	9294.0	47.3	20973.0	320.0	277.8	58.0	1.0	1.0	1.0	951.1
2	5972.0	30.4	17178.0	262.1	55.1	50.0	1.5	1.0	1.0	686.9
3	8906.0	45.3	15892.0	242.5	234.9	58.0	713.1	290.9	89.8	27.3
4	2263.0	11.5	1938.0	29.6	8.7	28.0	257.8	108.6	30.1	10.2
5	6121.0	31.1	8524.0	130.1	111.6	46.0	153.3	351.8	144.2	1.0
6	4745.0	24.1	4768.0	72.7	21.7	40.0	1.4	1.0	1.0	509.4
7	3254.0	16.5	7560.0	115.3	54.6	38.0	337.0	136.3	39.8	6.7

Data are not automatically shown when Ferdinand files are read. If not, press the Show Data button to show the data on-screen.

Data can be exported to a text for, or to Microsoft Excel, for further analysis. Press the Export Button to do this; choose the destination using the Output Options Selector Buttons to the right of the Export Button. You may use the exported data to classify events, create figures, or simply stare at it in amazement.

The events being exported can be filter in two ways. First, events within minimal signal can be excluded. These are events where the fluorescent signal strength on all Channels is below a minimum value specified in the Preferences. These are simply noise events (*e.g.,* dust) of no interest and are typically excluded from exported data sets to save space. IF you wish to see these events, select the Export Events with Minimum Signal Check Box. Second, one can exclude events that were filtered out by using their values for Gates; these events are not classified. To see these events, select the Export Events that were Filtered Out Check Box.

The FCS File Viewer | Distributions Tab

This is where the distribution of signal strength for each Gate and Channel (raw and compensated) can be viewed. The Control Panel appears on the right-hands side of the screen.

Histogram Display Selector

These are a set of Radio Buttons that allow users to select which data series to display: Gates, Channel raw values, or Channel compensated values. In the figure to the right, the values for the GFP Channel (raw values) has been selected.

Once a data series has been selected, a histogram of values will be constructed and displayed. The minimum and maximum values for the histogram are set in the Protocol. The number of bins is set in the Preferences.

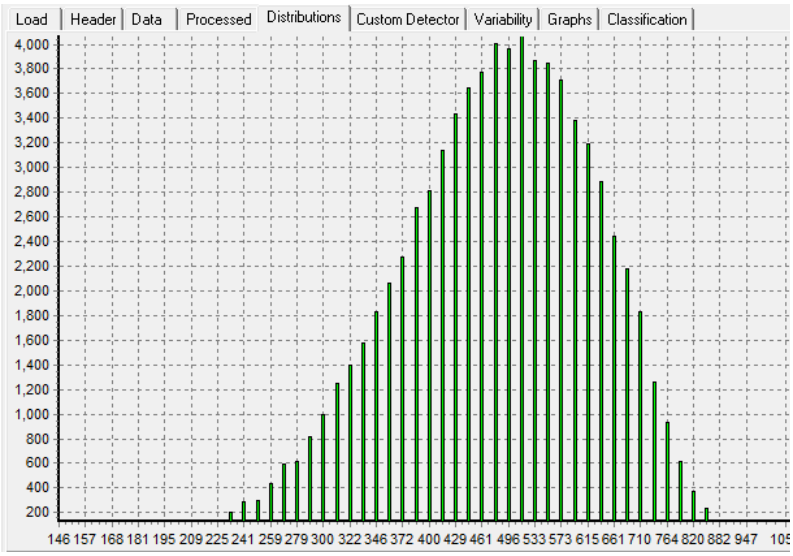
Display Subset Buttons

The subset of events tabulated for each histogram is selected by the set of 3 buttons at the top left of the control panel. One may select any or all of these data sets, where each is a subset of the previous set.

All Button. This button produces a histogram constructed from all events.

Threshold Button. This button produces a histogram from events remaining after applying static threshold of all Gates. That is, values that lay above the maximum, or below the minimum, acceptable value for any Gate results in filtering that event out. This will be a subset of the events reported using the All Button above.

Global Z Button. This button produces a histogram of events after removing those failing the static thresholds as above, but also that that fail any dynamic Gate. That is, values falling outside the acceptable range of Z values for Gaussian curves fit to their distribution, will be excluded. This is only useful is the distribution of gating values follows a Gaussian distribution (*e.g.,* signal strength of a Syto-62 dye). If none of the Gates use dynamic gating as an additional filtering step, then this histogram will be identical to that produced by the Threshold Button above.



The colors used for each histogram are specified in the Protocol for each Gate or Channel. In the example to the left, the raw data for the GFP Channel are shown, colored green. Users may zoom in by clicking and dragging down and right before releasing. Click and drag up and to the left to zoom out.

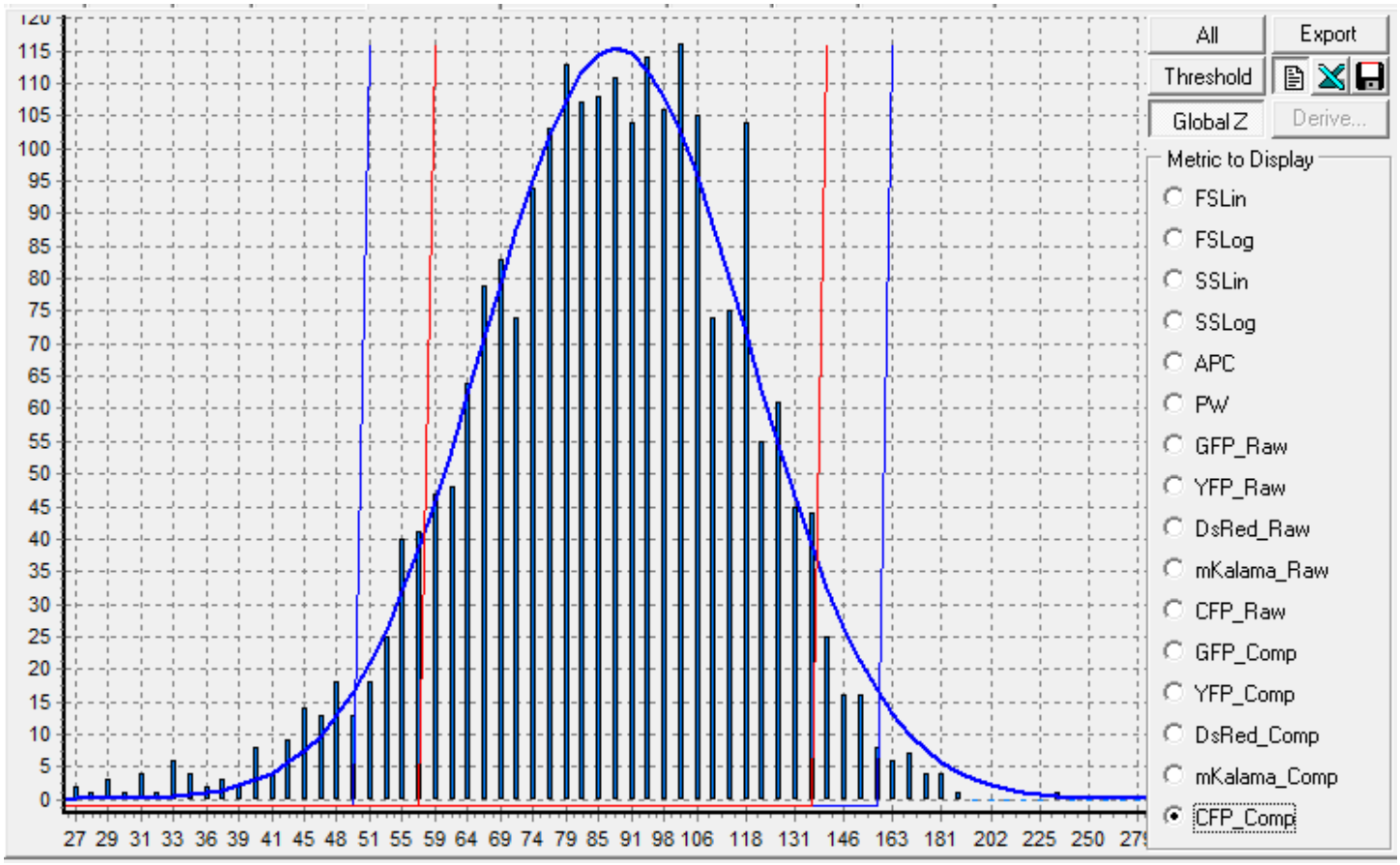
Derive Button. This button become enabled when a raw data series for a Channel is selected; it is used to create a compensation matrix, as described below.

Export Button. This allows an export of all histograms to a spreadsheet; these data are sent

to the destination selected by the Output Options Selector Buttons located immediate below the Export Button. This allows data to be replotted for the creation of publication-quality figures.

Gaussian Curve fits

Gaussian curves are fit automatically to the distributions of compensated data on each Channel for data sets have passed GlobalZ filtering on all Gates; Gaussian curves are not fit for raw data, or for compensated data for **All** data, or data passing **static thresholds** for all Gates (see above). These curves may be refined using the Custom Detector (see below). As example is shown below for the compensated date on the CFP Channel; aside from the blue histogram bars, we see a Gaussian curve fit to the data. As discussed in the Protocols section, Ferdinand uses a seed mean and variance for this Gaussian distribution which is stored in the Protocol.

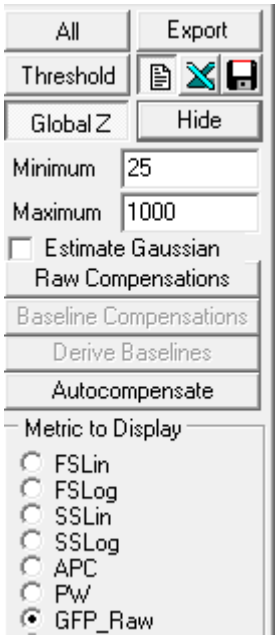


In addition to the curve fit, there are four vertical lines, two in red and two in blue. The Two blue lines indicate the bounds for curve Clipping; that is, the curve is only fit to value between these two lines. Values outside this range are ignored during curve fitting. Here, the clipping values are set to -2.0 <= Z <= 2.0. If the curve does not fit well, the range of clipping values should be altered (see the Custom Detector section for more information).

The two red lines indicate the range of Z values used in Z-Scoring. This range should not lie outside the range of clipping values. That is, if the range does not allow good curve fitting, it should not be used for Z-Scoring.

To export all data for this channel – all subsets and curve fits – right click the graph and choose Export.

Deriving a Compensation Matrix



A compensation matrix may be constructed from one or more FCS files. Here, the raw data on each Channel is used to calculate spillover coefficients. A Protocol must be active to create a compensation matrix within each of its Channels. Either read a Protocol, or create one from an FCS file as described in the Creating a protocol section above.

Next, read an FCS file that you wish to use to create the compensation matrix for one or more Channels. The ideal way is to have separate FCS files where all events have signal that is optimized by one detector and spill over onto all other detectors. For example, if your Protocol has three channels (*e.g.,* GFP, YFP and DsRed), you would have three data files, one each with only GFP, one with only YFP and one with only DsRed events.

In this example, we have opened an FCS file made from only GFP-expressing cells. When a Raw data series is chosen in the Histogram Display Selector (as GFP_Raw is selected in the example to the left), this will enabled the Derive Button. Press the Derive Button to reveal the Compensation Constructor Buttons. Press the Hide Button (which is the Derive Button renamed) to hide these buttons once again.

Press the Raw Compensations Button to calculate Spillover Coefficients from events on this Channel onto other detectors and store these values in the Compensation Matrix of the current Protocol. In this example, we have read a file with GFP events; any signal on other channels (YFP and DsRed) would represent spillover. This button will calculate those mean spillover coefficient for values with raw GFP signal between 25 and 1000, the values specified in the edit boxes (see figure above). Set these values to exclude noise events. Critically, this button assumes that all events express GFP; this should not be a file with cells expressing different fluorescent proteins. Repeat this procedure with files containing cells expressing each fluorescent protein in turn. In this example, we would read a separate file with YFP-expressing cells and use it to calculate spillover from YFP onto the GFP and DsRed detectors. After that, we would then read a file with DsRed-expressing cells. These three files allow all spillover coefficients to be calculated from all three Channels. This is the best way to calculate a compensation matrix.

If separate files – each with a different fluorescent cell type – are not available, one can use a mixture of all fluorescent cells. This is less robust in that a single range of raw value is used (the range 25-1000 is used above). After reading this FCS file (or combination of multiple FCS files as explain above), select a Raw channel to reveal the Compensation Constructor Buttons as above. Here, press the Autocompensate Button. This will attempt to classify events according to their compensated signal (using a crude compensation matrix); it then uses the appropriate subset of events to calculate spillover. This works reasonably well, but its values are not as good as those constructed from individual files.

The Baseline Compensation Button and Derive Baseline Button are used to construction alternative compensation matrices that account for baseline background signal. This function is available to developers only.

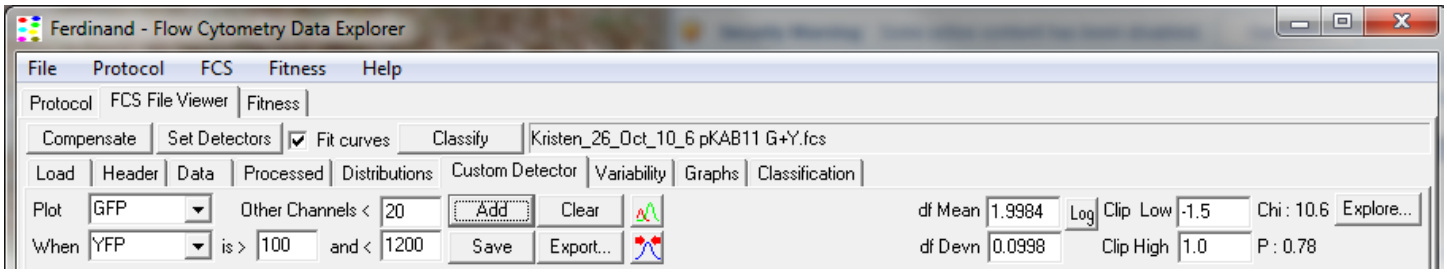
Creating Seed Gaussian Distributions

The events read to calculate spillover coefficients can also be used to estimate values for the seed Gaussian Curve, whose values are used to fit Gaussian curves to actual data. By default, Ferdinand uses a Gaussian with a mean of 250 (log=2.39) and a standard deviation of 0.27. This works for many distributions of actual data, but it will be slow and will fail if the mean for actual data is far from that value. If you wish to save the mean and standard deviation from the data used to calculate Spillover Coefficients, select the Estimate Gaussian Check Box. These data are saved to the Protocol.

The FCS File Viewer | Custom Detector Tab

Gaussian distributions are automatically created from the distribution of compensated signal each Channel, as described above. However, these fits may not be optimal for three reasons. First, the Clipping values – range of Z-values over which to evaluate the curve fit via a χ^2 goodness-of-fit test – are specified in the Protocol. This is useful for getting approximate curve fits, but optimal fits may require adjustment of these values. Second, noise may confound curve fits. Lastly, signal may be produced from multiply-fluorescent cells. For example, GFP signal may arise from both GFP+YFP and GFP+DsRed expressing cells; the Gaussian curve for these two distributions will not be identical, and fits to all events with strong compensated signal on the GFP Channel will be suboptimal.

These problems can be solved by use of the Custom Detector, which allow for optimal fit of a Gaussian curve to the distribution of compensated signal strength for each Channel. The control panel is situated at the top of the Custom Detector Tab.



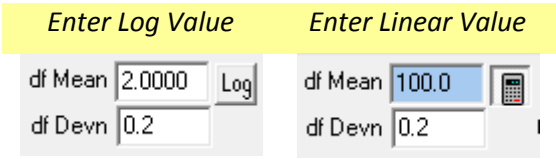
Selecting Primary and Filtering Channels

To begin, specify which Channel to analyze; above we are analyzing compensated signal on the GFP Channel. We are filtering events in two ways. First, we are specifying that signal on other Channels is low (less than 20 in this case). That way, clumps of cells with multiple strong signals can be discarded. Second we are specifying a filtering Channel, in this case YFP. That is, we are only examining GFP signal from events with reasonable signal strength (between 100 and 1200) on YFP. This is used to define a Gaussian curve for GFP in cell expressing both GFP *and* YFP. IF we do not want a filtering Channel (*e.g.*, we want to examine GFP signal without regard to another Channel), we would set the filter to pass GFP events over a minimum value, as shown to the right.

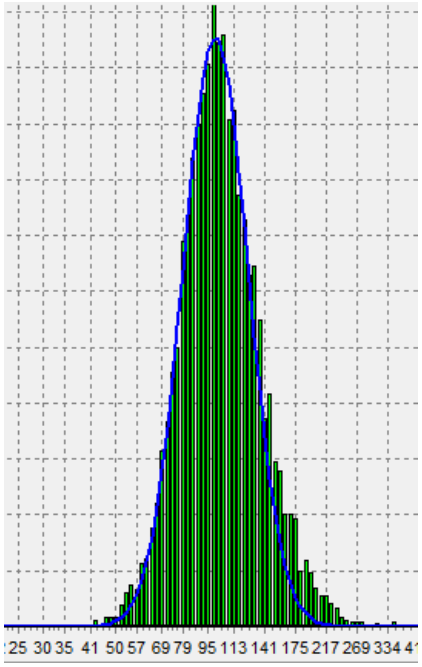
Example of setting a simple, 1-Channel filter.

Setting seed mean and variance

We then supply seed value for the mean and standard deviation; these should be supplied as log-transformed values. Inspection of the GFP curve for this file shows a mean signal strength of about 100, or a log of about 2.0; a standard deviation of 0.2 was given as a seed value. These are supplied in the df Mean Edit Box (default mean) and df Devn Edit Box (default deviation).



Values should be entered on a log scale. For those not accustomed to calculating log transformations on the fly, Ferdinand includes a built-in calculator. In the figure on the right, we see a value of 2.0. The button to the right of the Edit Box read "Log," indicating that this is on the appropriate log scale. To calculate this value, depress the Log Button. The Edit Box will turn blue (as shown in the figure above) and the button will show a calculator. It has transformed the value of 2.0 to read 100.0. We could alter this to read something else (say 217.5), then press the Calculator button. This would transform this value onto a log scale. This convention of toggling between linear and log scales – with a accompanied change in the sharing of the Edit Box – is used in several locations in Ferdinand.



Setting clipping range

Initial Clipping values were set as -1.5 <= Z <= 1.0; these were entered into the Clip Low and Clip High Edit Boxes.

Fitting a curve

After all these parameters are entered, we press the Add Button. This fits the curve, which is shown to the left. The mean and standard deviation (log-transformed) are reported in the edit boxes used to supply the initial seed values. Above we can see that the curve being fit has a mean of 1.9984 and a standard deviation of 0.0998.

The goodness-of-fit is calculated as a χ^2 between observed values and those predicted by the Gaussian curve across the range of Z values specified in the Clipping controls. The value of the χ^2 statistic, as well as the P-value, are reported on the upper right. For this fit, the is 10.6, with a P-value of 0.78. This is not significant; therefore, we do not reject the hypothesis that the Gaussian curve is a good fit to the observed data across the range being analyzed. Significant values for the χ^2 statistic indicate a poor fit.

Curve fitting algorithm

Curves are fit via nested iteration according to the following algorithm. Means and variances are explored

```
Begin with default mean and variance
Set Mean Step = 500% Bin Width; variance multiplier = 10%; Explore Means and variances
Set Mean Step = 50% Bin Width; variance multiplier = 2%; Explore Means and variances
If Refining then
  Set Mean Step = 0.5% Bin Width; variance multiplier = 0.1%; Explore Means and variances

Exploration routine:
  repeat
     Increase or decrease mean by Mean Step
     repeat
        Increase or decrease variance by Variance Multiplier
        Fit curve;
        Calculate sum of squared deviation of fit
     until 10 cycles of variance adjustment show no improvement
  until 10 consecutive cycles of mean adjustment show no improvement
```

Using Gaussian Distributions

To save the curve found, press the Save Button. This will save the data to the Protocol under the specified Channel.

To export a summary of the data shown, press the Export Button; this will send the data to a File Viewer.

To export the data in the histograms, right-click the graph and press the Export Menu Item.

To clear the data, press the Clear Button.

Refine and Filter Buttons unselected:

Refine and Filter Buttons selected:

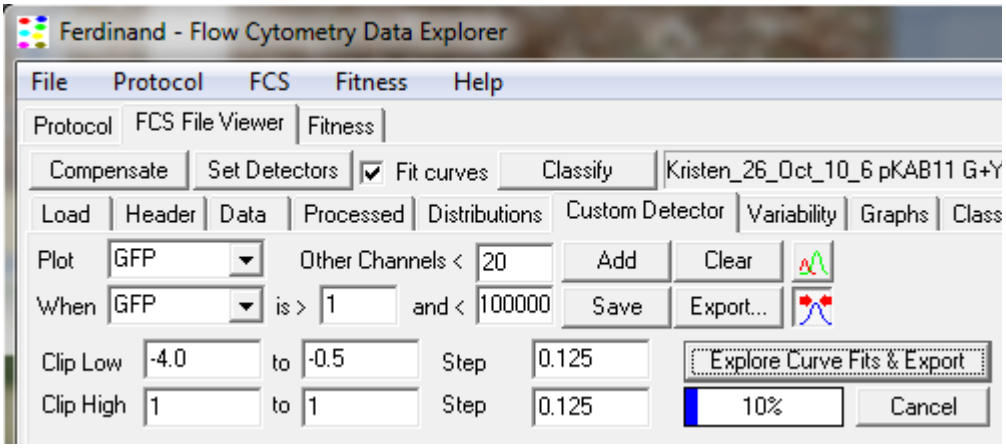
Better fits can be obtained by refining the curve fits; this explores fine-scale changes to the mean and variance values and can result in slightly better fits. This is achieved at the cost of lower speed, which can be an issue on slower computers. To include a refinement step, click the Refine Button so that it is down; this button is

located immediately to the right of the Export Button. This will add the refinement step to any curve being added from that point onward. To turn off refinement, click the button again.

Additional refinement can be gained by filtering events based on their raw spillover onto other Channels. Define spillover filters in the Protocol for each Channel. To activate any spillover filters for this channel, click the Filter By Spillover Button so that it is down; this is located above the Refine Button, to the right of the Clear Button.

Exploring ranges of clipping values

Optimal fits are obtained using appropriate Clipping values; when the Clipping range is set too large, noise events are included and the fit is poor. If the range is set too narrow, fewer data points are included and the fit can be suboptimal.



To explore ranges of values, click the Explore… button on the right-hand side of the Custom Detector control panel. This will reveal the Exploration panel, as to the left. Here, ranges of Clipping values can be explored; the values for Gaussian curve fits will then be automatically exported to a File Viewer form. In the example shown here, we are exploring minimal Z values between -4.0 and -0.5 in 0.125 increments; this will explore 29 values of Clip Low. In this example, the Clip High value is set to a constant of 1.0; it could, of course, be explored separately, or in combination with varying Clip Low values. Once values are set, press the Explore Curve Fits & Export Button. This will perform all curve fits and export a summary; as shown above, a progress gauge is shown along with a Cancel Button that will allow one to abort the analysis.

MinZ	MaxZ	Chi	P	Mean	Dev
-4	1	124.2	1.73E-12	100	0.103
-3.875	1	119.9	4.39E-12	100	0.103
-3.75	1	118.2	4.13E-12	100	0.103
-3.625	1	118.2	4.13E-12	100	0.103
-3.5	1	106.6	1.64E-10	100	0.103
-3.375	1	66.2	1.01E-04	100	0.103
-3.25	1	64.8	9.66E-05	100	0.103
-3.125	1	53.4	1.84E-03	100	0.103
-3	1	49.9	3.32E-03	100	0.103
-2.875	1	49.9	3.32E-03	100	0.103
-2.75	1	49	2.87E-03	100	0.103
-2.625	1	49.5	1.65E-03	99.6	0.101
-2.5	1	23.9	0.41	100	0.103
-2.375	1	15.2	0.85	100	0.103
-2.25	1	15.1	0.82	100	0.103
-2.125	1	13.1	0.87	100	0.102
-2	1	13.1	0.87	100	0.102
-1.875	1	13.1	0.83	100	0.102
-1.75	1	12.8	0.8	100	0.102
-1.625	1	12.8	0.75	100	0.102
-1.5	1	11.5	0.78	100	0.101
-1.375	1	11.4	0.72	100	0.1
-1.25	1	11.1	0.68	100	0.1
-1.125	1	11.1	0.68	100	0.1
-1	1	18.7	0.14	98.9	0.0932
-0.875	1	10.6	0.57	99.7	0.0932
-0.75	1	9.2	0.6	100	0.0932
-0.625	1	15.2	0.13	98.9	0.0932
-0.5	1	13.8	0.13	98.9	0.0932

The summary can be used to find optimal values for the Clipping parameters. The output for this analysis is shown on the left. The χ^2 goodness of fit test does not begin to reject fit of the Gaussian to the data until Clip Low values of Z=-2.625, indicating a very good fit to a wide range of values. For lower Clipping values, a Gaussian curve is a poor fit, and the good-of-fit tests reject these fits. Notably, the mean and variance for these Gaussian curves do not change significantly as the value for Clip Low is varied.

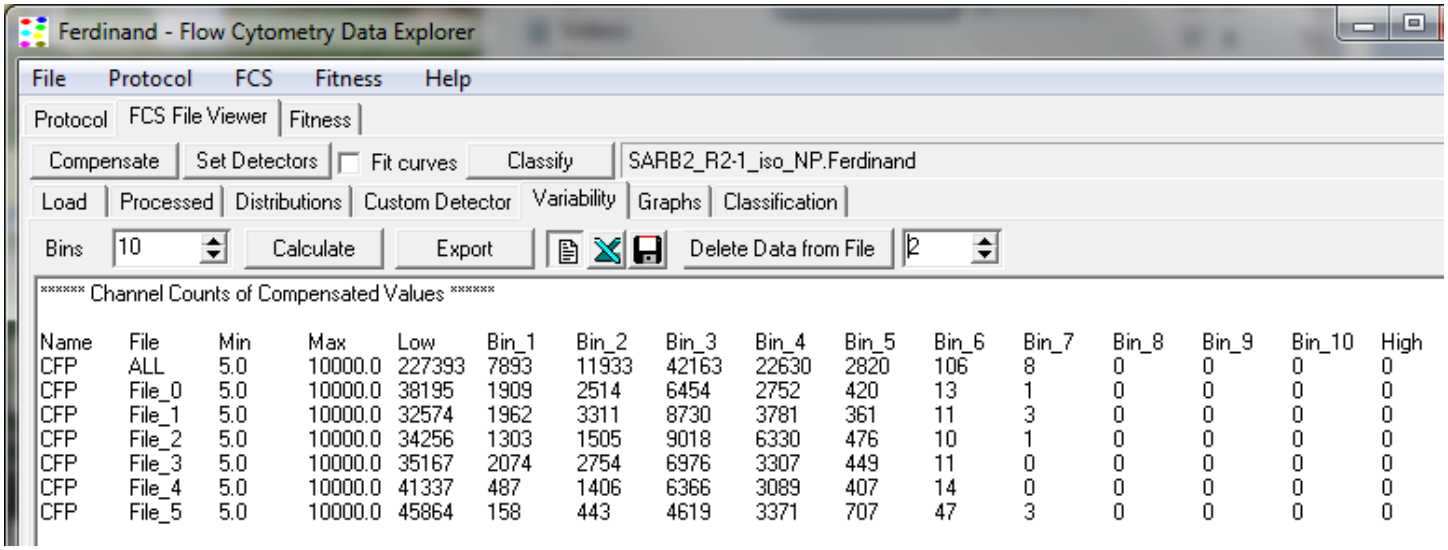
Once the exploration of Clipping values is complete, the data are exported to Excel and the Exploration Panel is automatically hidden once again.

The FCS File Viewer | Variability Tab

As described above, multiple FCS files may be analyzed *en masse*. While data are combined for curve fitting, classification is done separately. In the Variability Tab, the distribution of signal strength is examined across individual files. In this way, files with aberrant data can be identified and eliminated from the analysis.

Validating data across multiple files

The analysis works by sorting compensated data for each Channel into bins; the number of bins is set using the Spin Edit at the upper left side of the control bar. Once set, the data are analyzed by pressing the Calculate Button. Data will appear in the pane below (see example below.)



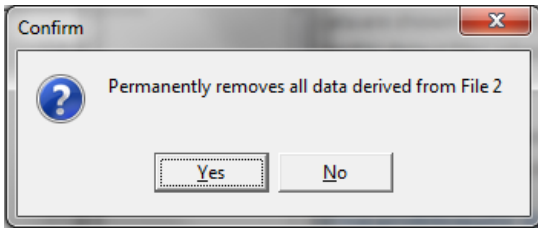
Name	File	Min	Max	Low	Bin_1	Bin_2	Bin_3	Bin_4	Bin_5	Bin_6	Bin_7	Bin_8	Bin_9	Bin_10	High
CFP	ALL	5.0	10000.0	227393	7893	11933	42163	22630	2820	106	8	0	0	0	0
CFP	File_0	5.0	10000.0	38195	1909	2514	6454	2752	420	13	1	0	0	0	0
CFP	File_1	5.0	10000.0	32574	1962	3311	8730	3781	361	11	3	0	0	0	0
CFP	File_2	5.0	10000.0	34256	1303	1505	9018	6330	476	10	1	0	0	0	0
CFP	File_3	5.0	10000.0	35167	2074	2754	6976	3307	449	11	0	0	0	0	0
CFP	File_4	5.0	10000.0	41337	487	1406	6366	3089	407	14	0	0	0	0	0
CFP	File_5	5.0	10000.0	45864	158	443	4619	3371	707	47	3	0	0	0	0

In this example, data were sorted into 10 bins. Above we see the data for the CFP Channel; the counts for each bin are shown for the combined all (File = "ALL"), then for each of the 5 separate files that were combined into this data set. These data are shown for each Channel, then summarized as proportions of the data in each bin. A χ^2 goodness-of-fit test is used to detect files whereby fractions of cells in each bin vary from what one would expect given the collection of files as a whole. This is useful in detecting replicates that show aberrant data.

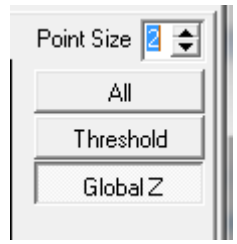
Data may be exported to a standard File Viewer Form, or sent to Excel, by pressing the Export Button. As always, the destination of the output may be chosen using the Output Options Selector Buttons to the right of the Export Button.

Removing problematic data

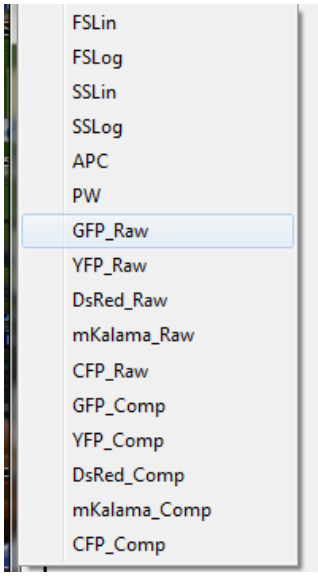
If a data file is deemed sufficiently suspect, its data may be removed from the Ferdinand file; this saves the efforts of locating the proper Protocol and reading the FCS files yet again. Specify the number of the atypical file using the spin edit at the right of the control bar (showing file 2 in the example above). Press the Delete Data from File Button; you will be asked to confirm this action (see example to the right). If you answer affirmatively, the data will be removed from the Ferdinand file (in this case from File_2). You may then save these remaining data as a separate Ferdinand file, thus retaining the original version.



The FCS File Viewer | Graphs Tab



The FCS File Viewer | Distributions Tab shows the distribution of values for a single Gate or Channel; it does not show the distribution of values for a Gate or Channel with respect to the value of another Gate or Channel for each event. The FCS File Viewer | Graphs Tab allows plotting of data for each event relating two Gates or Channels. To begin, use the small control panel at the right side of the panel to set the point size of the markers (smaller is better), and choose the subset of points. The All, Threshold and GlobalZ data sets are explained above in the FCS File Viewer | Distributions Tab section.

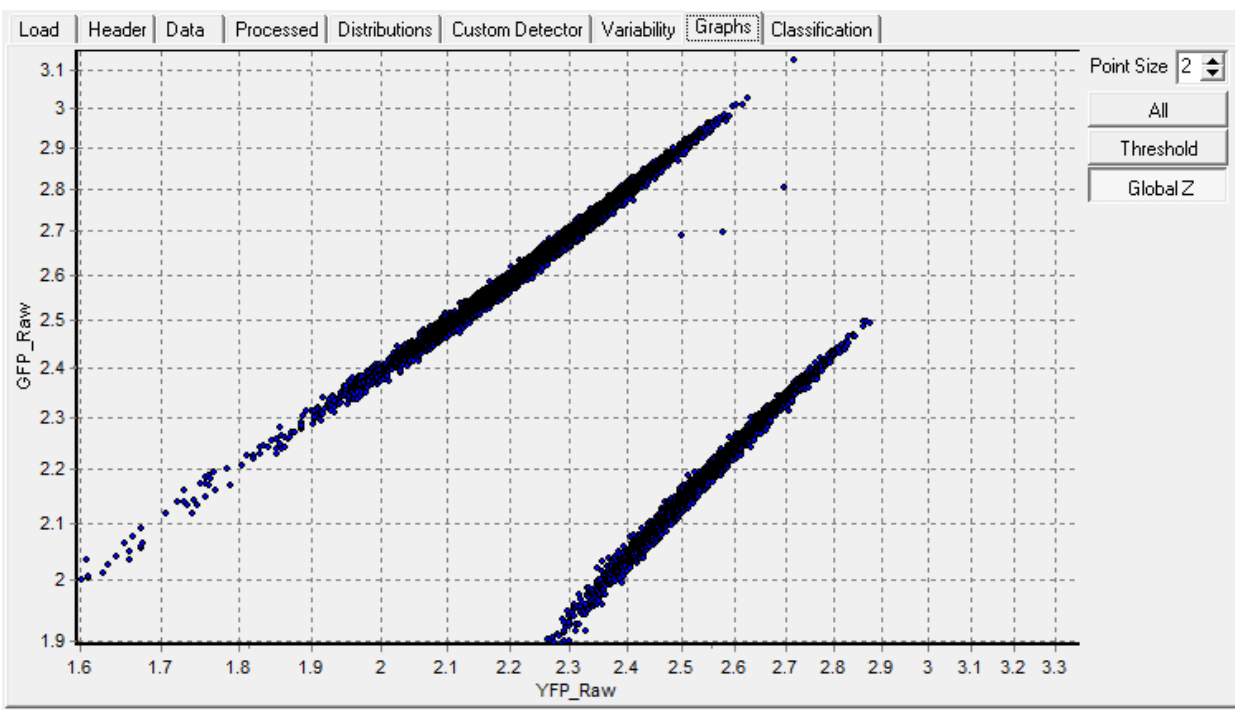


Next, choose the Gate or Channel to provide the data for the X- and Y-axes. Select each using a pop-up menu that appears when you right-click the graph itself.

To select values for the Y-axis, right-click along the left-hand 10% of the graph area, avoiding the bottom. A menu should appear. An example is shown to the left; the 6 global Gates are listed first, followed by the raw data for each Channel, then the compensated data for each Channel. This is the same listing as appears in the Histogram Display Selector Radio Buttons on the FCS File Viewer | Distributions Tab section. Choose a metric to display on the Y-axis by selecting a menu item.

To select values for the X-axis, right click on the bottom 10% of the graph area, avoiding the extreme left-hand side. A similar menu should appear. Choose a metric to display on the X-axis by selecting a menu item.

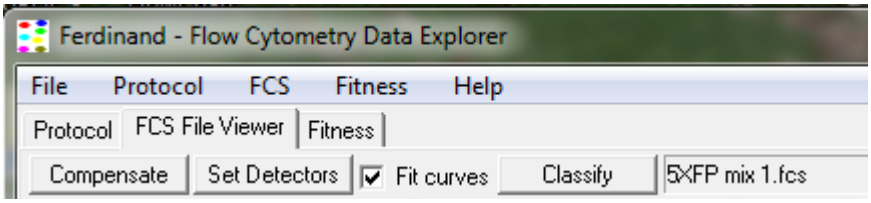
Once metrics for both axes have been chosen, a graph will appear; this can take quite some time if many (more than 200,000) values are being plotted. In the example below, raw values for the GFP and YFP Channels are plotted against each other; since each protein spills over onto the other detector, the two curves represent GFP-expressing (upper left) and YFP-expressing (lower right) cells. The lack of points between them bespeaks a low amount of noise.



Classifying Events

The FCS File Viewer | Classification Tab

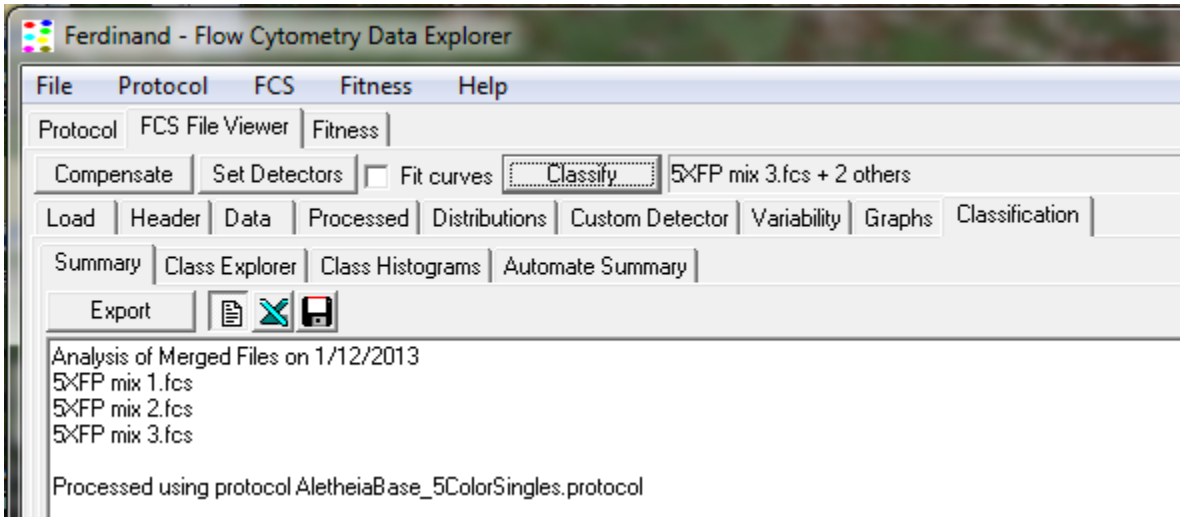
The FCS File Viewer | Classification Tab controls classification of fluorescent events. Events are classified according to Genotype, which is specified in the Protocol. Events are classified from FCS files in a three step process; these step are often run automatically, but can be re-run when the Protocol changes:



- First, an FCS file is opened using the current Protocol; alternatively, a Ferdinand file is opened, along with its embedded Protocol
- Next, raw data are compensated using the compensation matrix within the Protocol; this is true of Ferdinand files as well, as only the raw data are saved. If the compensation matrix is changed, then the data must be compensated again. Do this by using the Compensation Button, which is found in the upper left-hand side of the FCS File Viewer pane (see figure above).
- Next, the events are filtered and compensated values are calculated for the **GlobalZ** subset of events (see FCS File Viewer | Distributions Tab for a description). For Ferdinand files, the Gaussian curves stored in the Protocol are used; for FCS files, Gaussian curves are re-fit. If the data are re-filtered (by adjusting values for Gates in the Protocol), then the detectors must be set once again, using the Set Detectors Button above. This is not necessary if the Custom Detector is used to produce high-quality curve fits.
- Genotypes are specified in the Protocol
- Finally, events are classified. If Gaussian curves are adjusted, or is the ranges for Z-Scoring are adjusted in the Protocol, events must be classified again by pressing the Classify Button.
- The work flow proceeds from left to right in the button series shown above. If you press the Compensate Button, then the detectors are reset, and events are classified again.

The FCS File Viewer | Classification | Summary Pane

The Summary Pane shows the results of the classification of fluorescent events in the FCS or Ferdinand file given the parameters in the current Protocol; the results are produced anew every time the Classify Button is pressed. An example is shown in the figure below.



Exporting the data analysis

Data may be exported by pressing the Export Button at the top of the Classification | Summary pane. As always, the destination may be chosen by selecting the appropriate button among the Output Options Selector Buttons to the right of the Export Button.

Three sets of data analyses

The data are analyzed in three inter-related ways. A brief description follows; for a detailed presentation of the output file, consult the Anatomy of Ferdinand's Classification Output section below.

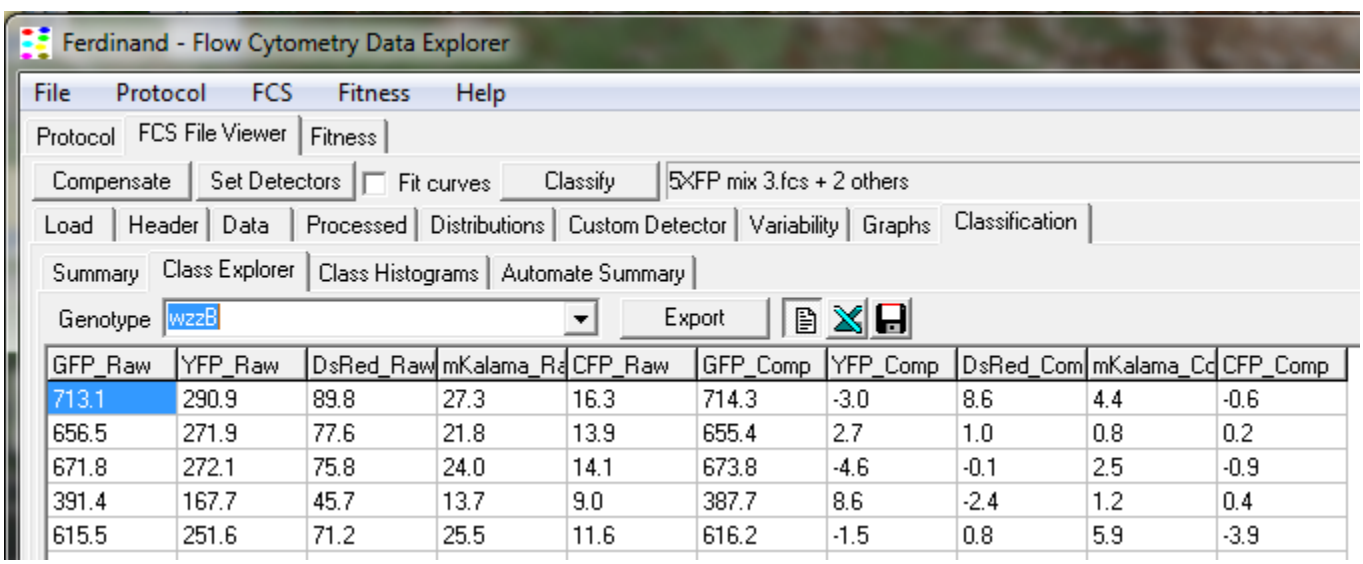
Threshold scoring. This classification uses simple thresholds to determine if an event is positive on a given Channel. Classes are defined as those being positive on certain Channel(s) and negative on all others. The numbers of events which are positive on all combinations of Channels are listed. Some of these may be biologically relevant (actually thought to exist in the sample), whereas other may not. For example, events may be scored as positive for both GFP and YFP, even though cells express only a single fluorescent protein.

Relaxed Z-Scoring. This is like threshold scoring, except when events which are positive on more than one Channel – and no such Genotype exists (such as the GFP+YFP example above). In these cases, the Channel with the strongest Z-score is marked as positive and the other is not. Essentially, the Z-scores are used to sort out false multiple-positives, reducing events scored with two or more positive Channels so that only the strongest Channel is considered. Unlike Threshold scoring – which enumerates events that are positive for each Channel – Relaxed Z-Scoring and Strict Z-scoring enumerate events which correspond to a set of Genotypes. See the Protocol | Genotypes section for more information.

Strict Z-Scoring. This implements a complete Z-Scoring algorithm, assessing events via a likelihood framework. Only events within the scorable range of Z values are considered.

The FCS File Viewer | Classification | Class Explorer Pane

The Class Explorer Pane summarizes each fluorescent event classified for each of the Genotypes by rigorous Z-Scoring. Events filtered by the Gates are not considered. Similar to the The FCS File Viewer | Processed Tab, the raw and compensated values for each Channel is shown; values for Gates are not shown.



GFP_Raw	YFP_Raw	DsRed_Raw	mKalama_Ra	CFP_Raw	GFP_Comp	YFP_Comp	DsRed_Com	mKalama_Co	CFP_Comp
713.1	290.9	89.8	27.3	16.3	714.3	-3.0	8.6	4.4	-0.6
656.5	271.9	77.6	21.8	13.9	655.4	2.7	1.0	0.8	0.2
671.8	272.1	75.8	24.0	14.1	673.8	-4.6	-0.1	2.5	-0.9
391.4	167.7	45.7	13.7	9.0	387.7	8.6	-2.4	1.2	0.4
615.5	251.6	71.2	25.5	11.6	616.2	-1.5	0.8	5.9	-3.9

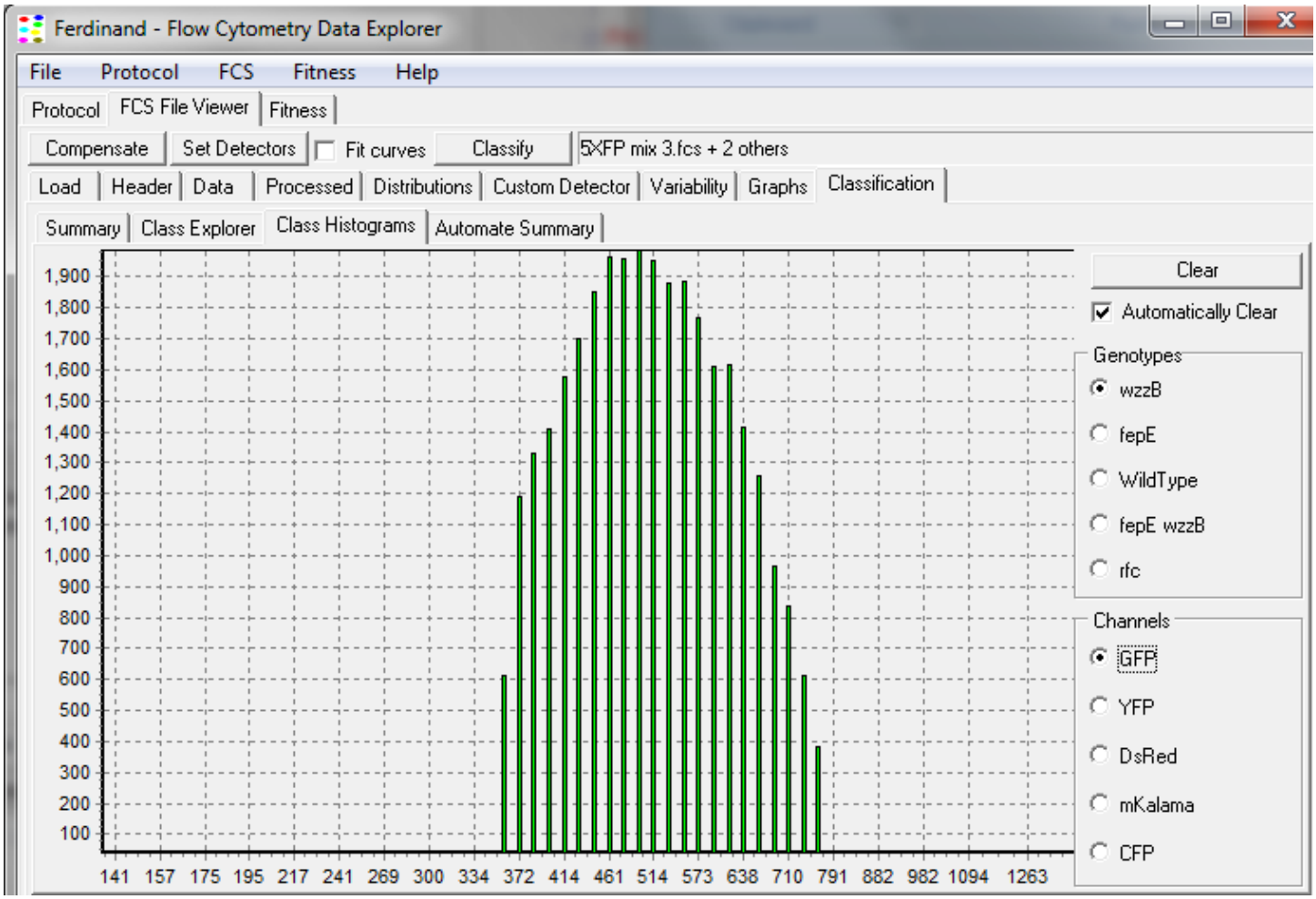
In the example above, the *wzzB* Genotype corresponds to cells labeled with GFP. As is evident from the first five events shown, they have strong signal on the GFP Channel and weak signal on the others. As expected, the compensated values

inferred from these data show only strong GFP signal. These data sets may be exported by pressing the Export Button located to the right of the Genotype Selector.

The FCS File Viewer | Classification | Class Histograms Pane

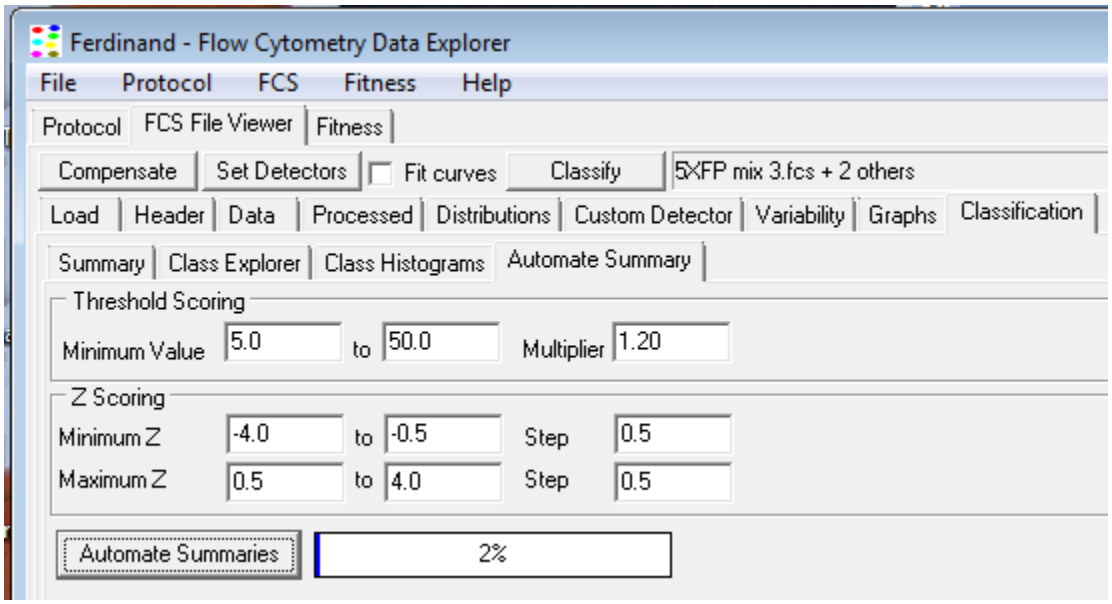
The Class Histograms Pane summarizes events classified into Genotypes by rigorous Z-scoring. Here, histograms of compensated signal strength on each Channel may be display for events classified into each Genotype.

As shown in the example below, a list of Genotypes is shown to the right of the graph; here, we have selected the *wzzB* Genotype, which has strong signal on GFP. We may also select the Channel to display; here we are showing the GFP Channel. The distribution appears truncated by these events were classified into the *wzzB* Genotype only if they lay between -1.0 <= Z <= 1.5 as determined using the mean and variance of the Gaussian distribution fit to the compensated values on the GFP Channel.



The FCS File Viewer | Classification | Automate Summary Pane

The Automate Summary Pane allows an exploration of thresholds – both simple and Z-value threshold – that are used to classify events. These controls allow for many thresholds to be examined, thereby allowing assessment of the robustness of the proportions of cells assigned to each Genotype as a function of the threshold used to classify them. There is a parallel here to the exploration of different Z-value thresholds used to establish Clipping boundaries – the range of Z-scores used to fit Gaussian curves to the distribution of signal strengths in a Channel. Here, we are using the Gaussian distributions provided in the Protocol, but varying the Z-value threshold used to score events. The wider the range of Z-values that are allowed, the more events that will be scored.



Performing an automated survey or scoring thresholds

As shown in the figure above, we may set the range of values three parameters used in scoring:

- **Minimum Value** considered in threshold scoring. Here, the threshold value is incremented not by addition as above, but by multiplication. In the example above, the threshold is increased every iteration by 20% by multiplying its value by 1.20.
- **Minimum Z** considered in Z-Scoring. In the example above, we are stepping through all values of **Minimum Z** between -4.0 and -0.5 in additive increments of 0.5. The value of the step must be greater than 0.
- **Maximum Z** considered in Z-Scoring. In the example above, we are stepping through all values of **Maximum Z** between 0.5 and 4.0 in additive increments of 0.5. The value of the step must be greater than 0.
- As other explorations (*e.g.,* in the Custom Detector), the ranges are nested within each. That is, all values for the range of **Minimum Value** will be explored for all possible values of **Maximum Z**, which will be explored for all possible values of **Minimum Z**. This can lead to large numbers of analyses. It is advisable to explore ranges of one of these variables at a time.

After choosing the range of values to explore, press the Automate Summaries Button. This will begin the analysis; results will be exported to a File Viewer Form when complete. A gauge will be shown to note progress; in the example above, 2% of the analyses have been completed.

The Anatomy of Ferdinand's Classification Output

The output file created by classification by Ferdinand contains a wealth of information, much if which is useful for data validation. Here we review the content of this file. Begin by opening an FCS or *.Ferdinand file and classifying fluorescent events using an appropriate external or embedded protocol. The output can be seen the FCS File Viewer | Classification | Summary Tab.

The File Header

The output file header notes which files were analyzed. For FCS files this will include the name(s) of FCS files as well as the protocol file that was loaded during the analysis. For *.Ferdinand files, the protocol is internal, so it is simply named "FileStream". Note in the examples below that the Ferdinand file was created from a set of six experimental replicates.

Header for a single FCS file	Header for a *.Ferdinand file
Analysis of 5XFP mix 3.fcs on 1/9/2013 Processed using protocol AletheiaBase_5ColorSingles.protocol	Analysis of Merged Files on 1/9/2013 Kristen_17_Nov_10_2 iso SARB2 NP1.fcs Kristen_17_Nov_10_3 iso SARB2 NP2.fcs Kristen_17_Nov_10_4 iso SARB2 NP3.fcs Kristen_17_Nov_10_5 iso SARB2 NP4.fcs Kristen_17_Nov_10_6 iso SARB2 NP5.fcs Kristen_17_Nov_10_7 iso SARB2 NP6.fcs Processed using protocol FileStream

Simple Threshold Scoring : The Channel Summary

In the Simple Threshold Scoring, classes are scored according to the Channels on which events have signal above its defined **Minimum Value** – and below its **Maximum Value** – as defined in the Protocol. This section is prefaced by the Channel Summary, which indicates the number of events which lay in this scorable range for each Channel. In addition, the χ^2 values and P-values for curve fits to the distribution of compensated values are listed for the 4 subsets of events classified – All, Threshold filtered, Global Z filtered, and Fluorescent Z filtered. In the example below, curves were only fit to the Global Z filtered data, which are viewed in the FCS File Viewer | Distributions Tab.

Channel	Count	Chi_All	P	Chi_Thresh	P	Chi_GblZ	P	Chi_FlZ	P
GFP	76721	-1.0	na	-1.0	na	467.599	0.00e0	-1.0	na
YFP	175976	-1.0	na	-1.0	na	2219.346	0.00e0	-1.0	na
DsRed	22310	-1.0	na	-1.0	na	146.194	1.00e-13	-1.0	na
mKalama	29083	-1.0	na	-1.0	na	702.691	0.00e0	-1.0	na
CFP	84483	-1.0	na	-1.0	na	458.006	0.00e0	-1.0	na

These numbers cannot be used in classification as they report events which are positive on those Channels; since events may be positive on more than one Channel, their sum represents more than the number of events in the data files.

Simple Threshold Scoring : Class Analysis

After the Channel report, the fluorescent events are sorted into individual classes according to the Channels on which they registered a positive signal (above its **Minimum Value** and below its **Maximum Value**). These numbers are reported for all possible combination of the Channels defined in the Protocol. As shown in the example below, the total counts for each class are reported in the first data column, and additional columns show the break-down by the separate files that were analyzed together. If only a single file is read, then the two columns will be identical. The example below shows the results for 6 files that were merged.

Class	Count	File0	File1	File2	File3	File4	File5
-----	8473	2289	2354	1184	1873	401	372
G----	22854	3419	3326	3801	3395	4690	4223
-Y---	109037	19511	15707	15173	20272	18704	19670
--R--	6380	1194	799	595	1166	1424	1202
---K-	19891	2861	2513	2481	2798	4247	4991
----C	73181	12059	14326	14687	13314	10554	8241
GY---	44974	7623	4495	7460	4582	9237	11577
G-R--	935	74	192	195	105	148	221

G--K-	409	79	112	106	44	44	24
G---C	1175	116	354	353	138	90	124
-YR--	8565	722	2425	1513	778	1194	1933
-Y-K-	2726	363	691	722	363	291	296
-Y--C	4158	403	1326	1476	557	278	118
--RK-	269	39	85	71	29	28	17
--R-C	459	46	169	149	58	27	10
---KC	4603	604	919	1162	639	615	664
GYR--	5323	710	572	1277	501	978	1285
GY-K-	398	51	66	112	48	51	70
GY--C	247	18	60	108	16	27	18
G-RK-	22	1	8	12	0	0	1
G-R-C	15	0	6	3	1	1	4
G--KC	275	34	34	47	26	33	101
-YRK-	159	10	61	50	7	11	20
-YR-C	68	7	26	25	3	3	4
-Y-KC	224	19	73	90	16	17	9
--RKC	27	3	8	13	2	1	0
GYRK-	48	0	12	15	4	7	10
GYR-C	19	1	5	6	1	4	2
GY-KC	11	2	3	3	0	1	2
G-RKC	2	0	0	0	1	0	1
-YRKC	5	0	1	2	0	0	2
GYRKC	14	0	5	8	1	0	0

The Class is defined using the single-character **Channel Tag** field; this is described in the Protocol | Channels|Channel Data section. The class reported in the first row includes events which have no positive signal on any Channel. After that, we see the classes of events where there is positive signal on only one Channel, then two, then three, four and finally all 5 Channels defined in this Protocol.

Relaxed Z-Scoring : Class Analysis

The Relaxed Z-Scoring classification scheme is a conceptual hybrid between rigorous Z-Scoring and Threshold scoring. Among its many weaknesses, threshold scoring suffers from the phenomenon illustrated in the figure above, where many events are classified as being positive on more than one Channel, even though only one fluorophore is present in any one cell. These are misclassification events. Relaxed Z-Scoring uses the Z-Scores of each event – the relative distance of each events signal on each Channel from the mean of that Channel – to determine which Channel has the strongest relative signal. It then ignores the signal on other, weaker Channels when classifying cells. Essentially, this is threshold scoring using Z-Scores as a way to resolve classification ambiguities.

The classification of events by both Relaxed Z-Scoring and Rigorous Z-Scoring corresponds are reported in a similar fashion. We begin by reporting the number of events in each class. The classes represent Genotypes defined in the Protocol (see the Protocol |Genotypes section for details). For experiments using singly-tagged cells, this would appear as a subset of the classes reported by Threshold Scoring above, except multiply-fluorescent classes have been eliminated. An example is shown below.

Class	Count	File_0	File_1	File_2	File_3	File_4	File_5
-----	40049	12247	9202	4596	11402	1939	663
G----	20572	2429	2822	3726	2460	4690	4445
-Y---	145259	21257	18780	23601	19223	28899	33499
--R--	7663	953	732	785	899	1919	2375
---K-	17929	2122	2149	2471	2048	4124	5015
----C	83474	13250	17048	17720	14706	11535	9215

Below is a second example of Relaxed Z-Scoring used on a collection of 9 files, each containing a pure culture of cells expressing two fluorescent proteins. Here, we can see that the classes are defined as those bearing 2 fluorophores. This is not simply all combinations of two fluorophores. Rather, it is the combinations specified by the 9 Genotypes defined in the Protocol ; the combination of YFP and DsRed (`-YR--`) was not used in any Genotype and therefore does not appear in the Class List.

Class	Count	File_0	File_1	File_2	File_3	File_4	File_5	File_6	File_7	File_8
-----	26664	888	2003	2087	11374	181	195	5765	592	1635
-Y-K-	11848	11847	0	0	0	0	0	0	0	0
--R-C	10946	0	0	0	0	10946	0	0	0	0
---KC	10989	0	10988	0	0	0	0	0	0	0
--RK-	10838	0	0	0	0	0	10838	0	0	0
G---C	11664	0	0	0	0	0	0	0	11651	0
G--K-	12061	0	0	0	0	0	0	0	0	12061
-Y--C	10496	0	0	0	0	0	0	0	0	0
G-R--	10419	0	0	10419	0	0	0	0	0	0
GY---	10280	0	0	0	0	0	0	10276	0	0

Notably, classification is quite rigorous for these files. Each file has all events begin scored into a single class, and each class is comprised of events which originated from a single file. There are no misclassification events, outside of the possibility that the 26,664 events that were unclassified represent events that lay outside the range of acceptable values defined by the **Minimum Value** and **Maximum Value** fields for each Channel as defined in the Protocol.

Relaxed Z-Scoring : Proportion Analysis as a function of the Fluorophore

Following the reporting of the numbers of events in each class, Ferdinand supplies some analyses of these counts to assist in data validation. The first of these is an analysis of how the proportions of cells in each class vary as a function of the file or origin. Ideally, the proportion of cells should not vary between replicate files.

An example is shown below for the six-file example using singly-labeled cells that has been presented so far. The upper table shows the proportions of cells in each class for the aggregate data as well as for each individual file. There is variability across files. Notably, File_5 appears to be more variable than others, with far more cells of the DsRed (R) Genotype (60.4%) and far fewer of the CFP (C) Genotype (18%) than shown by other files. This is unlikely to reflect mere stochastic variation as this File has a larger number of events being classified than the others.

Class	Fraction	File_0	File_1	File_2	File_3	File_4	File_5
-----	0.7918	1.5523	0.9803	0.4627	1.3737	0.5864	0.4281
G----	0.0738	0.0600	0.0673	0.0744	0.0602	0.0879	0.0802
-Y---	0.5192	0.4869	0.4520	0.4909	0.4649	0.5573	0.6043
--R--	0.0273	0.0232	0.0135	0.0141	0.0172	0.0379	0.0473
---K-	0.0565	0.0407	0.0408	0.0436	0.0376	0.0682	0.0875
----C	0.3232	0.3893	0.4264	0.3770	0.4201	0.2487	0.1807
	SSD	0.006	0.016	0.004	0.013	0.007	0.029
File 0	0.005	0.000	0.003	0.000	0.001	0.026	0.061
File 1	0.009	0.003	0.000	0.004	0.000	0.044	0.087
File 2	0.005	0.000	0.004	0.000	0.003	0.022	0.054
File 3	0.007	0.001	0.000	0.003	0.000	0.040	0.081
File 4	0.022	0.026	0.044	0.022	0.040	0.000	0.007
File 5	0.048	0.061	0.087	0.054	0.081	0.007	0.000

The second table shows the pairwise comparisons of the Sums of Squared Deviations (SSD) of the proportions of cells in each class. The first column (labeled SSD) shows the SSD between each file and the data set as a whole; the proportions of unscored events are not considered. The file most different from the rest is clearly File 5, whose SSD of 0.048 is more than twice the next most variable file (File 4). The remainder of the lower table shows the SSD comparisons of each file's proportions of cells vs. every other file's proportions. As expected, the diagonals (*e.g.,* comparing File 0 *vs.* File 0) are 0.000, since those proportions are identical. This table is more illuminating, since it shows that File 5 and File 4 are actually somewhat similar to each other (SSD=0.007), while each is quite different from Files 0-3. This may lead one to believe that those two files are different from the others. This analysis would inform users of this aberrancy and inform further analyses to determine if these replicates could be excluded.

Relaxed Z-Scoring : Proportion Analysis as a function of the File

Next, the proportion of events in each class is calculated as a function of the file of origin. That is, if File 0 has ~25% of the events for one class of events (say, the GFP Genotype), then that should be true for other classes as well. IF a particular file is atypically enriched for a particular class of events, then something may be wrong with the data analysis or the experimental design.

An example of this analysis is shown below. The proportion of each class arising from each file is noted. The **mean** and **standard deviation** for these proportions (for all classes except the unscored class) are shown below. File 5 had the greatest share of events, with an average of 22.9% of events arising there, with File 4 providing the second-largest number of events.

```
Proportion of total cell counts as a function of the File

Class      Fraction   File0      File1      File2      File3      File4      File5
-----                 0.3058     0.2298     0.1148     0.2847     0.0484     0.0166
G----                 0.1181     0.1372     0.1811     0.1196     0.2280     0.2161
-Y---                 0.1463     0.1293     0.1625     0.1323     0.1989     0.2306
--R--                 0.1244     0.0955     0.1024     0.1173     0.2504     0.3099
---K-                 0.1184     0.1199     0.1378     0.1142     0.2300     0.2797
----C                 0.1587     0.2042     0.2123     0.1762     0.1382     0.1104


           Mean       0.1332     0.1372     0.1592     0.1319     0.2091     0.2293
           Dev        0.0184     0.0406     0.0418     0.0257     0.0437     0.0764

Z Values for each proportions relative to respective means

Class      Z Value    File0      File1      File2      File3      File4      File5
-----                 9.3906     2.2796     -1.0639    5.9499     -3.6785    -2.7846
G----                 -0.8214    -0.0010    0.5238     -0.4809    0.4319     -0.1737
-Y---                 0.7162     -0.1953    0.0777     0.0159     -0.2327    0.0166
--R--                 -0.4792    -1.0269    -1.3587    -0.5690    0.9457     1.0545
---K-                 -0.8060    -0.4274    -0.5122    -0.6893    0.4786     0.6591
----C                 1.3904     1.6506     1.2694     1.7233     -1.6236    -1.5566

All Z values are within the acceptable range
```

Variability across files is examined by looking at the Z-scores, of the variation of each class from its file mean. The second table shows the Z-scores for each class within each file, calculated from the mean and variance for each file.

IF the Z-Scores are too extreme (this threshold is set in the Preferences and has a default of 1.96), then the file will be flagged with the disclaimer:

```
**** THERE ARE SUSPICIOUSLY HIGH Z VALUES; DATA MAY BE SUSPECT ****
```

Otherwise, Ferdinand will report:

```
All Z values are within the acceptable range
```

The Z-Scores of the proportion of events in the null class (unscored) are not considered in this evaluation. IN the example above, all the Z-scores were within the acceptable range. But what is also clear is that the proportion of unscored events differs markedly from the distribution of scored events, with the largest fraction of unscored events coming from File 0.

Rigorous Z-Scoring

The same analyses discussed above for Relaxed Z-Scoring are repeated again for Rigorous Z-Scoring. Here, each Genotype is evaluated for its likelihood in producing the signal strength seen in its cognate Channels. The Genotype

with the highest likelihood is assigned to that event, provided that the signal strength lies in the acceptable range defined by **Minimum Value**, **Maximum Value**, **Minimum Z** and **Maximum Z** fields.

Channel Means Report

The classification of events by Rigorous Z Scoring is used to generate the Channel Means Report; here the strength of signal on each Channel is evaluated for events in each class. We would expect signal to be strong for a Genotype on its cognate Channels and be low on other Channels. This report is very useful in detecting over-compensation (means on alternate channels are strongly negative), under-compensation (means on alternate Channels are well above 10) or improper assignment of Gaussian distributions (events show strong signal on a non-cognate Channel).

Below is an example of the Channel Mean Report for singly-labeled cells. The Channels are listed as columns, with each Genotype class appearing as a row. The top table shows the mean of signal strength for events in that class, whereas the second table shows the standard deviation.

Classification is fairly robust, with the strongest signal in each class corresponding to its cognate Channel. It does appear that the spillover of CFP onto mKalama has been overestimated, as there is evidence of overcompensation, with the mean signal for CFP being well below zero (-61.4) for cells with strong mKalama signal (260.0). In addition, we may take a second look at the Gaussian curves for YFP, as a large number of the unclassified cells appear to have strong YFP signal; perhaps this Gaussian was ill-fitting.

Mean Channel Values

Class	GFP	YFP	DsRed	mKalama	CFP
----C	1.4	1.7	1.1	1.9	43.6
---K-	1.2	2.5	1.1	260.0	-61.4
--R--	1.0	5.8	328.7	1.5	1.1
-Y---	3.7	452.7	-3.2	1.3	1.1
G----	531.3	-18.6	-2.3	-11.1	-4.5
-----	21.1	136.2	3.2	9.8	6.7

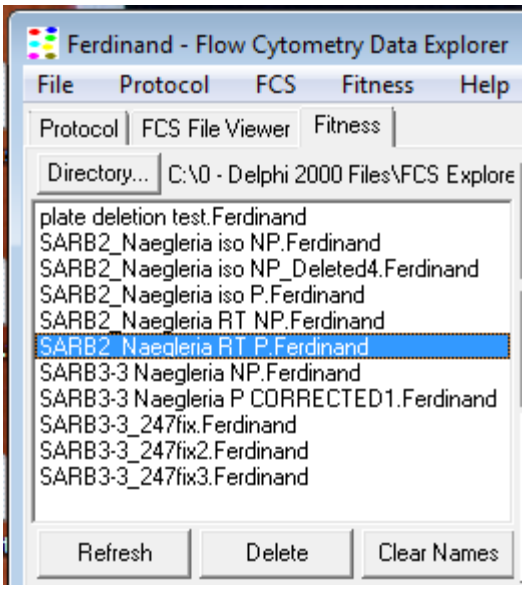
Mean Channel Deviation

----C	2.1	4.9	1.9	1.8	13.0
---K-	6.1	21.0	6.0	86.9	27.8
--R--	5.1	21.1	80.0	8.2	3.3
-Y---	10.0	127.4	11.8	7.3	2.7
G----	139.7	13.6	6.0	6.1	3.7
-----	129.9	305.2	45.5	62.4	31.3

Fitness Calculations

Ferdinand can calculate fitness values from sets of *.Ferdinand files.

This facility is only available to Ferdinand Developers.



Selecting Ferdinand Files

Full fitness analysis requires four *.Ferdinand files.

Troubleshooting

I read an FCS file and I don't see how to analyze the data.

You must have a Protocol loaded and active. IF one doesn't exist, create one from the FCS data itself. Consult the Creating a Protocol from an FCS File section.

I receive an error reading an FCS file

The file must be in FCS version 3.0 (or later) format. If it cannot be read at all, consult the software on your flow cytometer to see if the output format can be modified. Older file formats – *e.g.,* version 2.0 – are not supported.

I have a Protocol loaded, but I don't see any Channels on the Distributions Tab.

You may not have the right Protocol loaded. Export the parameter list from the FCS file (FCS File Viewer | Header Tab) and compare these to the **Long Name** and **Short Name** fields (in the FCS Import Group Box) for each Channel and Gate in your Protocol. If they all don't match, then the data cannot be interpreted.

The distributions of compensated data in the FCS File Viewer | Distribution Tab show many peaks.

You may not have compensated the data appropriately; this is especially true of the distributions for compensated data resemble the distributions for raw data. Consult the sections about making compensation matrices.

The Gaussian curves in the FCS File Viewer | Distribution Tab don't line up with my data.

The curves are fit using seed values from each Channel in the Protocol; if those values are far off, then Ferdinand will give up. Enter closer values for the Gaussian Curves for the Channels that are not fit well.

I see distributions when viewing all events, but I don't see any for Threshold or GlobalZ sets.

The thresholds set for filtering data via the Gates are probably too stringent. Look at the distributions for Gate values on the FCS File Viewer | Distributions Tab and compare them to the **Minimum Value and Maximum Value** fields for each Gate in the Protocol.