ACS Synthetic Biology



Technical Note

# TASBE Flow Analytics: A Package for Calibrated Flow Cytometry Analysis

Jacob Beal, Cassandra Overney, Aaron Adler, Fusun Yaman, Lisa Tiberio, and Meher Samineni

*ACS Synth. Biol.*, **Just Accepted Manuscript** • DOI: 10.1021/acssynbio.8b00533 • Publication Date (Web): 03 May 2019



ACS Publications

is published by the American Chemical Society. 1155 Sixteenth Street N.W., Washington, DC 20036

# TASBE Flow Analytics: A Package for Calibrated Flow Cytometry Analysis

Jacob Beal,*,† Cassandra Overney,‡,† Aaron Adler,† Fusun Yaman,† Lisa Tiberio,† and Meher Samineni¶,†

†*Raytheon BBN Technologies*

‡*Olin College*

¶*University of Utah*

E-mail: jakebeal@ieee.org

Abstract

Flow cytometry is a powerful method for high-throughput precision measurement of cell fluorescence and size. Effective use of this tool for quantification of synthetic biology devices and circuits, however, generally requires careful application of complex multi-stage workflows for calibration, filtering, and analysis with appropriate statistics. The TASBE Flow Analytics package provides a free, open, and accessible implementation of such workflows in a form designed for high-throughput analysis of large synthetic biology datasets. Given a set of experimental samples and controls, this package can process them to output calibrated data, quantitative analyses and comparisons, automatically generated figures, and detailed debugging and diagnostic reports in both human-readable and machine-readable forms. TASBE Flow Analytics can be used through a simple user-friendly interactive Excel interface, as a library supporting Matlab, Octave, or Python interactive sessions, or as a component integrated into automated workflows.

Keywords: flow cytometry, unit calibration, software tools, batch analysis, user interface

Flow cytometry is a powerful method for acquiring large volumes of high-precision singlecell size and fluorescence measurements. Most flow cytometry analysis software, however, has tended to focus on screening, sorting, and categorization of cell types (e.g., FlowJo,[1] Cytospec,[2] FlowingSoftware,[3] FlowCytometryTools,[4] OpenCyto,[5] FlowPy[6]). Many synthetic biology applications, however, such as engineering devices and circuits, instead need precise quantification of expression levels of fluorescent proteins or other fluorophores. Moreover, in order to compare and compose measurements (and derived models) across instruments and laboratories, these measurements must be calibrated to reproducible units (e.g., fluorescence into molecules of equivalent fluorescein (MEFL), forward scatter into equivalent $\mu m$ (E$\mu$m) diameter). Methods for precise quantification and calibration are well established, but generally require careful application of complex multi-stage workflows. Prior tools do not provide these workflows, making it difficult for any but expert users to obtain precise, calibrated data.

TASBE Flow Analytics provides a solution for this problem: modular calibration and analysis workflows usable via Excel, Matlab, Octave, or Python, distributed under a free and open license (https://tasbe.github.io/, https://github.com/TASBE/). This package was initially developed internally at BBN as part of the Toolchain to Accelerate Synthetic Biology Engineering (TASBE) project,[7] for purposes of characterizing transcriptional repressor devices with equivalent units on inputs and outputs, then refined across a number of projects, including the prediction of repressor circuits[8] and replicon systems,[9] the development of novel CRISPR-based repressors[10] and promoter insulators,[11] and interlaboratory studies on flow cytometry calibration.[12] Now publicly released, it has also inspired two related packages, FlowCal[13] and CytoFlow,[14] each providing a subset of its functionality (a detailed feature comparison is provided in Supporting Table S1).

The remainder of this document provides more details on TASBE Flow Analytics and the workflows it implements. First, we review methods for calibrated interpretation of flow cytometry data. Next, we present the architecture of TASBE Flow Analytics and its use either with

a user-friendly interactive interface or as part of automation-assisted workflows. Finally, we discuss ongoing development and how this package can serve the larger synthetic biology community.

## Calibrated Interpretation of Flow Cytometry Data

Flow cytometers operate by streaming particles from a sample past one or more interrogating lasers, measuring each particle's signal across multiple optical channels. Flow cytometry files contain this raw information, measured in arbitrary units determined by instrument configuration, settings, and time-dependent performance. To effectively quantify expression of a fluorescent protein or other marker, this data must be processed to filter out non-cell events, remove background fluorescence and "bleed-over" from other channels, and translate arbitrary units into reproducible, comparable units. TASBE Flow Analytics provides a modular workflow (illustrated in Figure 1) for creating "color models" (i.e., data structures containing the transformation functions for mapping raw flow cytometry data to filtered and calibrated unit data) from a standardized collection of process controls, and for applying these color models to the calibrated interpretation of flow cytometry data.

In particular, for complete calibration (whether with TASBE Flow Analytics or with any other tool), one needs the following standardized process controls:

- Variable fluorescence beads, for calibration of fluorescent channels to standard units of intensity.
- Variable fluorescence beads and variable-diameter beads, for calibration of forward scatter to standard sizing units.
- A wild-type or null transfection, for autofluorescence compensation and automationassisted gating.

- Strong constitutive expression of each fluorescent protein separately, for compensation for spectral overlap.
- Equivalent co-expression of multiple fluorescent proteins, for conversion of fluorescence channels to comparable units.

More information about how each is used in calibration is provided below, along with circumstances where some of these may be omitted. Failure or unexpected behavior for any of these controls is also a valuable indicator of issues in a protocol or instrument.

## Automation-Assisted Gating

Many events captured by a flow cytometer are not cells, but debris, cell-fragments, or pairs and clumps of cells. Unless such non-cell events are under study, they should be eliminated by some gating function. Typically gates are based on arbitrary thresholds set by analysts based on personal judgement, creating high variability across practitioners and instruments, even if gates are shared.[15] As has been rediscovered multiple times, this can be greatly improved by fitting gates from Gaussian mixture models (GMMs).[16–18] TASBE Flow Analytics provides a GMM gating implementation, defaulting to a two-component model for aggressive removal of non-cell events, but configurable for more permissive operation or in various ways to handle data peculiarities. An effective gate can typically be constructed from a wild-type or null transfection control, but others may need to be used if the protocol causes strong changes in cell morphology. Such issues can typically be detected as strong fluctuations from sample to sample in the fraction of events being filtered by a gate, which is one of the statistics reported by TASBE Flow Analytics. Gating may also be omitted (to study non-cell distributions) or replaced with alternative functions, e.g., asymmetric distributions.[19,20]

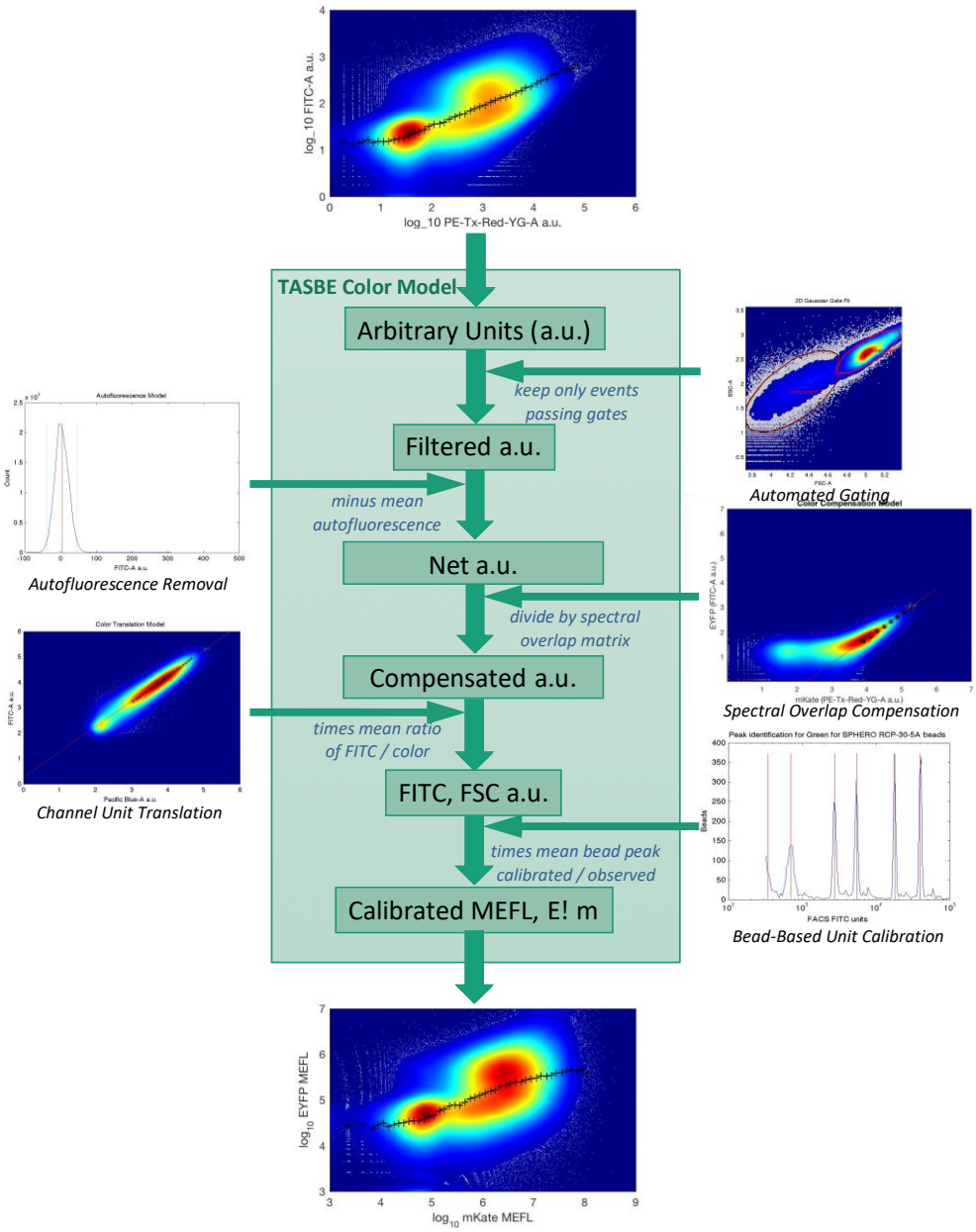


Figure 1: TASBE Flow Analytics provides a modular "color model" workflow for calibrated interpretation of flow cytometry data (top to bottom): raw data is filtered to remove noncell events using a gate function that can be automatically generated from a wild-type control. Background fluorescence and "bleed-over" from other channels is then removed using autofluorescence and spectral overlap models based on non-fluorescent and strong single-fluorescence controls, respectively. If multiple fluorescent channels are used, their values are translated to a single channel (preferably the FITC channel) using co-expression controls (scatter channels are not translated). Finally, calibration beads are used to scale to standard units—preferably molecules of equivalent fluorescein (MEFL) and equivalent $\mu m$ diameter (E$\mu$m).

## Compensation for Autofluorescence and Spectral Overlap

Fluorescence measurements capture more than just the intended molecule. Cells have some autofluorescence, and fluorescent molecules have overlapping excitation and emission spectrums, such that each fluorescent channel "bleeds over" into other channels—especially with fluorescent proteins, many of which have particularly broad spectrums. Both autofluorescence and spectral overlap can be compensated with linear transformations,[21,22] subtracting autofluorescence and dividing by a matrix of overlap percentages. TASBE Flow Analytics color models compute and apply such compensation (subject to user configuration), quantifying autofluorescence from non-fluorescent controls (e.g., wild-type or null transfection) and spectral overlap from strong single-positive controls. Optionally, for assays needing to include autofluorescence, its contribution can be restored after spectral overlap is removed.

## Unit Calibration

Reproducible scientific measurements generally require comparable units. Since flow cytometer measurements depend on both instrument optical configuration and settings for a particular run, units typically must be determined through calibration against reference samples. While synthetic biologists often compare only with reference strains,[23,24] interlaboratory study demonstrates that independent calibrants can greatly reduce variability.[25] The standard independent calibrant for flow cytometry fluorescence is beads with a known intensity[26–30]—typically implemented as a mixed population with multiple intensity values that can be seen as peaks in a histogram of flow cytometry events. Similarly, variable diameter beads can calibrate forward scatter to quantify particle size,[31,32] although the relationship is more complex and the method not yet as rigorously validated. TASBE Flow Analytics can use both fluorescence and size calibration beads, looking them up in an included catalog of calibration values for all currently known commercially available bead models and lots. With multiple fluorescent channels, color models can also convert fluorescent channels to equivalent units by

reference to multi-color co-expression controls, e.g., allowing simultaneous measurement of inputs and outputs in comparable units.[8,10] This technique requires fluorescent proteins with equivalent expression levels and degradation rates, which is simple in some systems (e.g., transient co-transfection of pHef1a-[fluorescent protein] plasmids in mammalian cells, as in[8,10]), but significantly more difficult in others. Optionally, when calibration is unavailable or somehow undesirable, TASBE Flow Analytics can also leave channels in arbitrary units. Combined with compensation and gating, this completes specification of color models to transform raw flow cytometry data into calibrated quantification of cellular fluorescence.

## Example Application to Experimental Data

To better explain the operation of these processes, consider Figure 1, which illustrates the creation of a TASBE color model and its application to analysis of an experimental flow cytometry file. This example is drawn from the examples and templates in the TASBE Flow Analytics tutorial,[1] which in turn uses an excerpt of flow cytometry data in the experiments leading to one of the papers using TASBE Flow Analytics.[8]

In this example, the experimental data is from a genetic circuit experiment with three fluorescent channels of interest: PE-Tx-Red-YG-A, FITC-A, and Pacific Blue-A, respectively reporting mKate (red) fluorescence from a constitutive transfection marker, EYFP (yellow) fluorescence for the circuit output, and EBFP2 (blue) fluorescence for the circuit input. The plots show a density map of constitutive vs. output fluorescence, with black showing parametric analysis of mean output vs. constitutive expression, which in turn is indicative of the level of transient transfection in cells.

[1] In particular: the color model is that generated by the template file https://github.com/TASBE/TASBEFlowAnalytics-Tutorial/template_colormodel/make_color_model.m from one of the sets of example controls in https://github.com/TASBE/TASBEFlowAnalytics-Tutorial/example_controls/, and the experiment file analyzed is https://github.com/TASBE/TASBEFlowAnalytics-Tutorial/example_ assay/LacI-CAGop_C4_P3.fcs

The color model is built and applied in five stages:

1. Automation-assisted gating, working from the blank control, identifies a high-density component in forward scatter vs. side-scatter morphology map that are likely to be single-cell events (upper right red oval in "automated gating" image of Figure 1). For the experimental data, the color models applies this model as a gate to filter out all events not contained within the identified component—in this case 20.6% of approximately 180,000 events in the raw file.

2. The same blank control is used to build a model of autofluorescence on each channel, identifying mean and two standard deviations (e.g., the FITC-A autofluorescence model has a mean of 3.3 a.u. shown by the solid red line in the "autofluorescence removal" image of Figure 1). The color model corrects the experimental data for autofluorescence by subtracting the mean autofluorescence from each channel—though in this particular case autofluorescence is extremely low so the correction is negligible.

3. The three single-color constitutive controls are used to build spectral overlap models, from which a 3x3 overlap matrix is extracted (e.g., a calculated effective overlap of 0.57% from PE-Tx-Red-YG-A a.u. to FITC-A a.u. is shown by the red line in the "spectral overlap compensation" image of Figure 1). The color model corrects the experimental data for spectral overlap by the multiplying each event by the inverse of the overlap matrix.

4. Channel-to-channel color translations are computed from a control containing an equal cotransfection of the three plasmids used in the single-color controls, all of which are identical except for the particular fluorescent protein used (e.g., a calculated multiplier of 2.17 to transform Pacific Blue-A a.u. to FITC-A a.u. is shown by the red line in the "channel unit translation" image of Figure 1). The color model maps compensated Pacific Blue-A and PE-Tx-Red-YG-A data into equivalent FITC-A values by multiplying by these translation multipliers.

5. Finally, the unit scaling from compensated FITC-A a.u. to MEFL is computed from a bead sample (in this case SpheroTech RCP-30-5A beads, lot AA02) by identifying sharp "peaks" in the FITC-A histogram (for this particular dataset, only the single brightest peak is resolvable due to an extreme voltage setting, so Figure 1 illustrates a more typical case from another dataset, in which six of the eight peaks for RCP-305A beads are visible and identified, as indicated by the vertical red lines). The color model applies this translation to the experimental data by multiplying by the unit scaling factor.

The result is a filtered and transformed data set, as illustrated by the visible differences between the top "raw" experimental data plot and the bottom calibrated experimental data plot. In particular, notice that the units have changed from arbitrary units of each color to comparable MEFL units on both channels, filtering has enriched the relative density of data points with high values, and compensation has rounded out the bottom right portion of the distribution, which was previously distorted upward and cut off sharply along a diagonal due to spectral overlap.

## TASBE Architecture and Applications

TASBE Flow Analytics makes calibrated interpretation workflows available through multiple interfaces (illustrated in Figure 2), implemented with Octave-compatible Matlab code that can also be wrapped and executed from Python (e.g., via Oct2Py). This package is supplemented with tutorials, sample data, templates for invoking workflows, and web-based documentation. Entry points are thus provided for users at varying levels of sophistication: for less computational users, the web-based documentation and Excel interface provide a quick start without need for deep understanding of the tools or process; for those ready to engage more deeply with data processing, the tutorials provide an interactive Matlab/Octave walk-through that explains and illustrates key concepts with the aid of sample data and bundled papers for supplementary reading, and the workflow templates contain internal documentation making them easy to copy

and adapt to use on a user's own projects; and for sophisticated users each function in the API has been documented following typical Matlab/Octave conventions.

The TASBE package architecture is designed to enable analysis of large data volumes, including automated (re)execution of analyses. Notably, this includes "warn but comply" exception handling to better support batch processing of datasets that include problematic samples. In this exception-handling paradigm, all problems are reported in warnings and/or log files but execution continues through a batch of data to the greatest extent possible, even if it means that results may be invalid; the user then inspects the warnings post-run to determine whether the batch needs to be adjusted and re-executed or whether some samples should simply be marked as invalid (e.g., due to culturing or collection failures). Likewise, workflows default to a set of recommended best practices, but are highly modular and configurable to support a wide range of unconventional experiment goals, salvaging value from flawed data, and otherwise adapting to unexpected circumstances and results.

## Analytical Workflows

The core of TASBE Flow Analytics is two workflows: construction of calibration "color models" from process controls and application of color models to analyze experimental data, both following the calibrated interpretation process described above. Color model construction takes a configuration and set of process control FCS files and produces a color model object, plus diagnostic logs noting any potential issues detected and figures showing how parameters were extracted from each sample.

Experiment analysis follows one of several workflows based on common experimental structures, producing not only logs and figures but also statistical values computed from calibrated sample data. In addition to direct conversion of flow cytometry FCS files to pointclouds of calibrated values, three main workflows are currently supported: batch analysis of sets of replicates (e.g., collections of candidate insulators[11]), comparison of contrasting

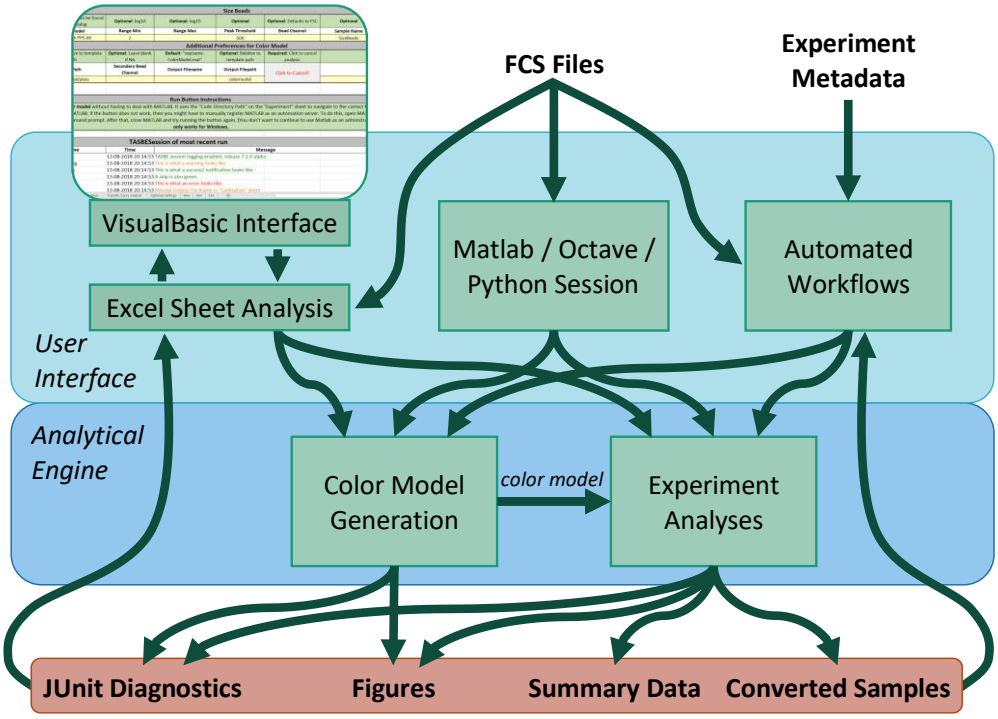


Figure 2: Architecture of TASBE Flow Analytics package: processing of flow cytometry FCS files can be invoked through three interfaces: a user-friendly Excel spreadsheet; via interactive sessions using the package as a library in Matlab, Octave, or Python; or as a component in high-throughput automated workflows. Through these interfaces, two main analytical workflows are invoked: construction of calibration "color models" from process controls and application of color models to analyze experimental data. Both workflows output diagnostic logs in JUnit format and graphical figures, while the experimental analysis workflow also outputs summaries of experimental data and (optionally) conversions of samples into point clouds.

conditions (e.g., "plus" vs. "minus" conditions[10]), and computation of input/output transfer curves (e.g., repressor models[8]). The outputs include geometric statistics, as most gene expression distributions are heavy-tailed (typically either log-normal[33,34] or gamma[35]), plus analysis of signal-to-noise relations as appropriate.[36]

## Interactive User Interface

TASBE Flow Analytics can be used straightforwardly as a library through standard interactive sessions in Matlab, Octave, or Python. For those who prefer a simpler spreadsheet environment, a user-friendly Excel interface is also provided. This interface is an Excel workbook template with blanks for information about an experiment, independent variables, sets of samples and

replicates, and how they are to be used in color models and experimental analyses. Visual Basic code embedded into the workbook then allows TASBE Flow Analytics to be run from Excel (on Windows machines at least), invoking the library and using multithreading to incrementally collect execution results back into Excel. The Matlab back-end is thus hidden from the user, who can manage TASBE Flow Analytics workflows entirely from within Excel.

### Workflow Integration

Finally, TASBE Flow Analytics has also been designed for use as a component of more complex automated workflows. Supporting Octave (and through it Python) simplifies deployment for headless execution on shared server or cloud environments, as does "warn but comply" exception handling. In addition, TASBE Flow Analytics can be configured from JSON files, providing a simple path for configuring workflows from experiment metadata. Complementarily, execution traces can write into JUnit format for tooling-friendly reporting of workflow successes, errors, and warnings. Finally, TASBE Flow Analytics is maintained following industrial software engineering best-practices, including use of git-flow development patterns, continuous integration and validation via Travis-CI, and semantic versioning, simplifying integration and maintenance as a software dependency. Together, these facilitate integration of TASBE Flow Analytics into automated workflows such as those enabled by SBOL[37] and linking to data and metadata storage systems such as SynBioHub[38] or EDD.[39]

## Discussion

TASBE Flow Analytics provides a free and open solution for calibrated interpretation of flow cytometry data. Multiple user interfaces support easy integration into synthetic biology workflows, from individual practitioners with a handful of files to complex multi-institution projects. Given the critical value of comparable units in science, and their proven value to synthetic biology engineering in particular, we hope the availability of this package will help

facilitate wider adoption of reproducible units and standard process controls throughout the synthetic biology community.

Development and maintenance of the TASBE Flow Analytics package is ongoing, driven by the various projects using the package, and as more people continue to adopt the package across a broader range of applications and organisms, we anticipate that it will continue to be refined to better serve the needs of the community.

## Author Contributions

J.B., C.O., A.A., F.Y., L.T., and M.S. wrote the software, J.B. wrote the manuscript, and J.B., C.O., A.A., F.Y., L.T., and M.S. edited the manuscript.

## Acknowledgement

Development of TASBE Flow Analytics has been partially supported by DARPA under contracts HR0011-10-C-0168, W911NF-11-054, FA8750-17-C-0231, and W911NF-17-2-0098 and by NSF Award #1522074 and #8600021241. All views and conclusions embodied in this distribution are those of the authors and not DARPA, NSF, or the U.S. Government. This document does not contain technology or technical data controlled under either U.S. International Traffic in Arms Regulation or U.S. Export Administration Regulations.

Additional code contributors to TASBE Flow Analytics are listed in the AUTHORS file of its distribution: at the time of this writing, these additional contributors are James Gentile and Jed Singer.

# Supporting Information Available

Table S1: Feature comparison between free and open calibrated flow cytometry tools: TASBE Flow Analytics, FlowCal, and CytoFlow.

| **Feature** (as of version evaluated) | **TASBE Flow Analytics** 7.6 | **CytoFlow** 1.0 | **FlowCal** 1.2.1 |
|---|---|---|---|
| License | Permissive Usage | GPL | Permissive |
| Matlab / Octave | Yes | No | No |
| Python / Jupyter | via Oct2Py | Yes | Yes |
| Excel interface | Yes | No | Yes |
| Point and click UI | No | Yes | No |
| FCS import | Yes | Yes | Yes |
| CSV import | Yes | Yes | No |
| Automatic gating | Yes | Yes | Yes |
| Fluorescence unit calibration | Yes | Yes | Yes |
| Size unit calibration | Yes | No | No |
| Built-in bead catalog | Yes | Partial | No |
| Autofluorescence correction | Yes | Yes | No |
| Spectral overlap compensation | Yes | Yes | No |
| Channel-to-channel unit conversion | Yes | Yes | No |
| High-throughput analysis templates | Yes | Yes | Yes |
| Bulk statistics | Yes | Yes | Yes |
| Histogram analysis | Yes | Yes | Yes |
| Gaussian mixture model analysis | Yes | Yes | No |
| Parametric analysis | Yes | Yes | No |
| Plus/minus comparative analysis | Yes | No | No |
| Signal-to-noise ratio analysis | Yes | No | No |
| Excluded data analysis | Yes | No | No |
| High-throughput exception architecture | Yes | No | Yes |
| Metadata validation | Yes | No | Yes |
| Machine-readable analysis logs | Yes | No | No |
| Load/save configuration state | Yes | No | No |
| Continuous integration & testing | Yes | Disabled | No |

# References

(1) FlowJo. https://www.flowjo.com/, Retrieved, Dec. 2018.

(2) Purdue University Cytometry Laboratories, Cytospec. http://www.cyto.purdue.edu/Purdue_software, Retrieved, Dec. 2018.

(3) Terho, P. Flowing Software. Turku, Finland: Cell Imaging Core of the Turku Centre for Biotechnology, 2012.

(4) Friedman, J.; Yurtsev, E. FlowCytometryTools. https://pypi.org/project/FlowCytometryTools/, Retrieved, Dec. 2018.

(5) Finak, G.; Frelinger, J.; Jiang, W.; Newell, E. W.; Ramey, J.; Davis, M. M.; Kalams, S. A.; De Rosa, S. C.; Gottardo, R. OpenCyto: an open source infrastructure for scalable, robust, reproducible, and automated, end-to-end flow cytometry data analysis. *PLoS Computational Biology* 2014, *10*, e1003806.

(6) Saxena, S.; Sastry, M. V. S. R.; Kumar, R. S.; Mehta, T.; Bose, B. FlowPy. http://flowpy.wikidot.com/, Retrieved, Dec. 2018.

(7) Beal, J.; Weiss, R.; Densmore, D.; Adler, A.; Appleton, E.; Babb, J.; Bhatia, S.; Davidsohn, N.; Haddock, T.; Loyall, J.; Schantz, R. An end-to-end workflow for engineering of biological networks from high-level specifications. *ACS Synthetic Biology* 2012, *1*, 317–331.

(8) Davidsohn, N.; Beal, J.; Kiani, S.; Adler, A.; Yaman, F.; Li, Y.; Xie, Z.; Weiss, R. Accurate predictions of genetic circuit behavior from part characterization and modular composition. *ACS Synthetic Biology* 2014,

(9) Beal, J.; Wagner, T. E.; Kitada, T.; Azizgolshani, O.; Parker, J. M.; Densmore, D.; Weiss, R. Model-driven Engineering of Gene Expression from RNA Replicons. *ACS Synthetic Biology* 2014,

(10) Kiani, S.; Beal, J.; Ebrahimkhani, M. R.; Huh, J.; Hall, R. N.; Xie, Z.; Li, Y.; Weiss, R. CRISPR transcriptional repression devices and layered circuits in mammalian cells. *Nature Methods* 2014, *11*, 723–726.

(11) Carr, S. B.; Beal, J.; Densmore, D. M.; Mutalik, V.; Gao, Y.; Arkin, A. Reducing DNA context dependence in bacterial promoters. *PLoS One* 2017, *12*, e0176013.

(12) Beal, J.; DeLateur, N.; Teague, B.; Weiss, R.; Sexton, J.; Castillo-Hair, S.; Tabor, J. J. Toward Quantitative Comparison of Fluorescent Protein Expression Levels via Fluorescent Beads. 9th International Workshop on Bio-Design Automation (IWBDA). 2017.

(13) Castillo-Hair, S. M.; Sexton, J. T.; Landry, B. P.; Olson, E. J.; Igoshin, O. A.; Tabor, J. J. FlowCal: A user-friendly, open source software tool for automatically converting flow cytometry data from arbitrary to calibrated units. *ACS Synthetic Biology* 2016, *5*, 774–780.

(14) Teague, B. Cytoflow. http://bpteague.github.io/cytoflow/, Retrieved, Dec. 2018.

(15) Spidlen, J.; Leif, R. C.; Moore, W.; Roederer, M.; Brinkman, R. R. Gating-ML: XMLbased gating descriptions in flow cytometry. *Cytometry Part A: The Journal of the International Society for Analytical Cytology* 2008, *73*, 1151–1157.

(16) Demers, S.; Kim, J.; Legendre, P.; Legendre, L. Analyzing multivariate flow cytometric data in aquatic sciences. *Cytometry: The Journal of the International Society for Analytical Cytology* 1992, *13*, 291–298.

(17) Chan, C.; Feng, F.; Ottinger, J.; Foster, D.; West, M.; Kepler, T. B. Statistical mixture modeling for cell subtype identification in flow cytometry. *Cytometry Part A: The Journal of the International Society for Analytical Cytology* 2008, *73*, 693–701.

(18) Boedigheimer, M. J.; Ferbas, J. Mixture modeling approach to flow cytometry data. *Cytometry Part A: The Journal of the International Society for Analytical Cytology* 2008, *73*, 421–429.

(19) Lo, K.; Brinkman, R. R.; Gottardo, R. Automated gating of flow cytometry data via robust model-based clustering. *Cytometry Part A: The Journal of the International Society for Analytical Cytology* 2008, *73*, 321–332.

(20) Lo, K.; Hahne, F.; Brinkman, R. R.; Gottardo, R. flowClust: a Bioconductor package for automated gating of flow cytometry data. *BMC Bioinformatics* 2009, *10*, 145.

(21) Roederer, M. Spectral compensation for flow cytometry: visualization artifacts, limitations, and caveats. *Cytometry Part A: The Journal of the International Society for Analytical Cytology* 2001, *45*, 194–205.

(22) Roederer, M. Compensation in flow cytometry. *Current Protocols in Cytometry* 2002, 1–14.

(23) Kelly, J. R.; Rubin, A. J.; Davis, J. H.; Ajo-Franklin, C. M.; Cumbers, J.; Czar, M. J.; de Mora, K.; Glieberman, A. L.; Monie, D. D.; Endy, D. Measuring the activity of BioBrick promoters using an in vivo reference standard. *Journal of Biological Engineering* 2009, *3*, 4.

(24) Nielsen, A. A. K.; Der, B. S.; Shin, J.; Vaidyanathan, P.; Paralanov, V.; Strychalski, E. A.; Ross, D.; Densmore, D.; Voigt, C. A. Genetic circuit design automation. *Science* 2016, *352*, aac7341.

(25) Beal, J.; Haddock-Angelli, T.; Baldwin, G.; Gershater, M.; Dwijayanti, A.; Storch, M.; de Mora, K.; Lizarazo, M.; Rettberg, R.; iGEM Interlab Study Contributors, Quantification of bacterial fluorescence using independent calibrants. *PLoS One* 2018, *13*, e0199432.

(26) Schwartz, A.; Gaigalas, A. K.; Wang, L.; Marti, G. E.; Vogt, R. F.; FernandezRepollet, E. Formalization of the MESF unit of fluorescence intensity. *Cytometry Part B: Clinical Cytometry* 2004, *57*, 1–6.

(27) Vogt Jr, R. F.; Marti, G. E.; Zenger, V. *Standardization and Quality Assurance in Fluorescence Measurements I*; Springer, 2008; pp 3–31.

(28) Wang, L.; Gaigalas, A. K.; Marti, G.; Abbasi, F.; Hoffman, R. A. Toward quantitative fluorescence measurements with multicolor flow cytometry. *Cytometry Part A: The Journal of the International Society for Analytical Cytology* 2008, *73*, 279–288.

(29) Hoffman, R. A.; Wang, L.; Bigos, M.; Nolan, J. P. NIST/ISAC standardization study: Variability in assignment of intensity values to fluorescence standard beads and in cross calibration of

standard beads to hard dyed beads. *Cytometry Part A: The Journal of the International Society for Analytical Cytology* 2012, *81*, 785–796.

(30) SpheroTech, *Measuring Molecules of Equivalent Fluorescein (MEFL), PE (MEPE) and RPE-CY5 (MEPCY) using Sphero Rainbow Calibration Particles*; 2001.

(31) Koch, A. L.; Robertson, B. R.; Button, D. K. Deduction of the cell volume and mass from forward scatter intensity of bacteria analyzed by flow cytometry. *Journal of Microbiological Methods* 1996, *27*, 49–61.

(32) Chandler, W. L.; Yeung, W.; Tait, J. F. A new microparticle size calibration standard for use in measuring smaller microparticles using a new flow cytometer. *Journal of Thrombosis and Haemostasis* 2011, *9*, 1216–24.

(33) Koch, A. L. The logarithm in biology. *Journal of Theoretical Biology* 1966, *12*, 276–290.

(34) Beal, J. Biochemical complexity drives log-normal variation in genetic expression. *Engineering Biology* 2017, *1*, 55–60.

(35) Friedman, N.; Cai, L.; Xie, X. S. Linking stochastic dynamics to population distribution: an analytical framework of gene expression. *Physical Review Letters* 2006, *97*, 168302.

(36) Beal, J. Signal-to-Noise Ratio Measures Efficacy of Biological Computing Devices and Circuits. *Frontiers of Bioengineering and Biotechnology* 2015, *3*.

(37) Myers, C. J.; Beal, J.; Gorochowski, T. E.; Kuwahara, H.; Madsen, C.; McLaughlin, J. A.; Mısırlı, G.; Nguyen, T.; Oberortner, E.; Samineni, M.; Wipat, A.; Zhang, M.; Zundel, Z. A standard-enabled workflow for synthetic biology. *Biochemical Society Transactions* 2017, *45*, 793–803.

(38) McLaughlin, J. A.; Myers, C. J.; Zundel, Z.; Mısırlı, G.; Zhang, M.; Ofiteru, I. D.; Goni-Moreno, A.; Wipat, A. SynBioHub: A Standards-Enabled Design Repository for Synthetic Biology. *ACS Synthetic Biology* 2018, *7*, 682–688.

(39) Morrell, W. C. et al. The experiment data depot: a web-based software tool for biological experimental data storage, sharing, and visualization. *ACS Synthetic Biology* 2017, *6*, 2248–2259.

## Graphical TOC Entry