Supplementary Information: Probing bacterial cell biology using image cytometry

Julie A. Cass[a], **Stella Stylianidou**[a], **Nathan J. Kuwada**[b], **Beth Traxler**[c], **and Paul A. Wiggins**[a,c,d,1]

[a]Departments of Physics; [c]Microbiology and ; [d]Bioengineering, University of Washington, Seattle, WA, 98195; [b]Department of Physics, Central Washington University, Ellensburg, WA, 98926

Software Availability and Documentation

The website for the SuperSegger software (Stylianidou et al., 2016), which contains the analysis tool `gateTool`, can be found at http://mtshasta.phys.washington.edu/website/SuperSegger.php. The software can be downloaded at the GitHub repository https://github.com/wiggins-lab/SuperSegger/. The GitHub wiki (found by following the 'Wiki' tab of the Github page) contains a detailed outline of how to use the SuperSegger software; this includes instructions for downloading, and details for both GUI and command-line use of the segmentation (`SuperSegger`) and `gateTool` analysis tool. Tutorials and sample datasets for these processes are also available at http://mtshasta.phys.washington.edu/website/tutorials.php.

The "View gateTool tutorial" link brings you to a scroll-through html version of the tutorial. The version begins with an electronic Table of Contents, so that you can jump directly to a section of interest. The "Download Tutorial" link allows you to download the tutorial script and all associated files, so that you may run through the tutorial yourself. This interactive method allows you to select your own gates by clicking to draw gates on 1D and 2D plots, just as you would when using the `gateTool` in your analysis.

Feature	Description
SuperSegger	Automated MATLAB-based trainable image cell segmentation, fluorescence quantification and analysis suite, well suited for high-throughput time lapse fluorescence microscopy of *in vivo* bacterial cells
`gateTool`	MATLAB-based image cytometry tool, designed to be part of the same complete package as *SuperSegger*, but able to be used independently; intakes *Clists* for plotting, gate-setting through interactive plots, and other analysis
Clist	Matrix-structured summary of single-valued cell descriptors for all cells
3D Clist	3D matrix-structured summary of cell descriptors for all cells, throughout the cell-cycle
Single Cell Towers	Time-lapse images of a single cell, stacked in chronological order, with consistent pole orientation
Consensus Images	Mean cell-cycle dependent fluorescence localization pattern from sets of single cell towers, spatially scaled to the same cell length and width

Table 1. Summary of *SuperSegger* and its analysis features.

Clist structure

All *Clist* information is collected in a single structure-array; it possesses structure-array fields for holding single-valued and time-dependent *Clist* data, cell descriptor definitions, and gates. These fields are defined as follows:

- **`data`**: stores single-valued measurements for all cells; each row represents an individual cell tracked through the time-course and the columns represent a subset of the > 70 cell descriptors.
- **`def`**: defines the `data` column number where each cell descriptors is stored.
- **`data3D`**: stores the time-dependent *3D Clist* data; it contains measurements for a subset of cell descriptors at each frame. The matrix contains Not-a-Number (NaNs) in frames where the cell does not exist.
- **`def3D`**: defines the `data3D` column number where each time-dependent cell descriptor is stored.
- **`gate`**: holds information about which cell descriptors have been gated (stored as field `x`) and what the gate values are (stores as field `ind`).

Clist definition fields

As shown in Section II, the `def` and `def3D` fields hold definitions of the cell descriptors stored in `data` and `data3D`, respectively. (For all fluorescent focus cell descriptors, FocusX(Y) refers to focus number Y found in channel X, and FluorZ refers to fluorescence channel Z.) Below are the `def` field definitions for the *Clist*:

1. Cell ID
2. Region num birth
3. Region num death
4. Cell birth time
5. Cell death time
6. Cell age
7. Old pole age
8. Error frame
9. stat0
10. Long axis (L) birth
11. Long axis (L) death
12. Short axis birth
13. Short axis death
14. Area birth
15. Area death
16. Region score birth
17. Region score death
18. X position birth
19. Y position birth
20. Fluor1 sum
21. Fluor1 mean
22. Fluor2 sum
23. Fluor2 mean
24. Num of neighbors
25. Region gray value
26. Focus1(1) long axis birth
27. Focus1(1) short axis birth
28. Focus1(1) score birth
29. Focus1(1) intensity birth
30. Focus1(2) long axis birth
31. Focus1(2) short axis birth
32. Focus1(2) score birth
33. Focus1(2) intensity birth
34. Focus1(2) long axis birth
35. Focus1(3) short axis birth
36. Focus1(3) score birth
37. Focus1(3) intensity birth
38. Focus1(4) long axis birth
39. Focus1(4) short axis birth
40. Focus1(4) score birth
41. Focus1(4) intensity birth
42. Focus1(5) long axis birth
43. Focus1(5) short axis birth
44. Focus1(5) score birth
45. Focus1(5) intensity birth
46. Focus1(1) long axis pole align
47. Focus1(1) long axis norm pole align
48. Focus1(1) long axis normalized
49. Focus1(2) long axis normalized
50. Focus1(3) long axis normalized
51. Focus1(4) long axis normalized
52. Focus1(5) long axis normalized
53. Focus1(1) short axis normalized
54. Focus1(2) short axis normalized
55. Focus1(3) short axis normalized
56. Focus1(4) short axis normalized
57. Focus1(5) short axis normalized
58. Focus1(1) gaussian fit width
59. Focus1(2) gaussian fit width
60. Focus1(3) gaussian fit width
61. Mother ID
62. Daughter1 ID
63. Daughter2 ID
64. dL max
65. dL min
66. L death / L birth
67. Fluor1 sum death
68. Fluor1 mean death
69. Fluor2 sum death
70. Fluor2 mean death
71. Focus1(1) long axis death
72. Focus1(1) short axis death
73. Focus1(1) score death
74. Focus1(1) intensity death
75. Focus1(2) long axis death
76. Focus1(2) short axis death
77. Focus1(2) score death
78. Focus1(2) intensity death
79. Focus1(2) long axis death
80. Focus1(3) short axis death
81. Focus1(3) score death
82. Focus1(3) intensity death
83. Focus1(4) long axis death
84. Focus1(4) short axis death
85. Focus1(4) score death
86. Focus1(4) intensity death
87. Focus1(5) long axis death
88. Focus1(5) short axis death
89. Focus1(5) score death
90. Focus1(5) intensity death
91. Focus1(1) gaussian fit width death
92. Focus1(2) gaussian fit width death
93. Focus1(3) gaussian fit width death
94. Long axis/Short axis birth
95. Long axis/Short axis death
96. Neck width
97. Maximum width
98. Cell dist to edge
99. Growth Rate

Below are the `def3D` field definitions for the *3D Clist*:

```
1. Cell ID                    10. Focus1(1) short axis
2. Long axis (L)              11. Focus1(1) score
3. Short axis                 12. Focus1(1) intensity
4. Area                       13. Focus1(2) long axis
5. Fluor1 sum                 14. Focus1(2) short axis
6. Fluor1 mean                15. Focus1(2) score
7. Fluor2 sum                 16. Focus1(2) intensity
8. Fluor2 mean                17. Focus1(1) long axis pole align
9. Focus1(1) long axis        18. Focus1(1) long axis norm pole align
```

The cell descriptors included in both the *Clist* and *3D Clist* are easily customizable. Users may do this by editing the *SuperSegger* script `trackOptiClist.m`, or simply by using the `add` and `add3d` commands of the `gateTool` (See Table 2).

`gateTool` Implementation

The `gateTool` function allows the user to manage all *Clist* modification and visualization by passing commands through a single `gateTool` function. Table 2 gives a description of all `gateTool` commands for modifying the *Clist*; this includes options such as merging *Clists*, loading data from the *Clist*, and setting gates. Table 3 gives a description of all `gateTool` commands for viewing the *Clist* data; this includes options such as producing histograms, dot plots, KDE plots, and setting axis scaling. Finally, Table 4 gives a description of all additional `gateTool` commands for such options as saving the *Clist* and displaying the *Clist* channel definitions. To use these commands, the user simply passes in the *Clist* (or a set of *Clists*), along with the command and any accompanying arguments (i.e. `[output] = gateTool(Clist, 'command', argument);`). Several commands can (and some commands *must*) be used together in a single call to `gateTool`. A tutorial for using the `gateTool` can be found at `http://mtshasta.phys.washington.edu/website/gatetoolTutorial/sample.html`.

If you prefer to use a GUI's, you can open an easy-to-use `gateTool` GUI using the script `gateToolGUI`. An image of this GUI is shown in Figure 1.

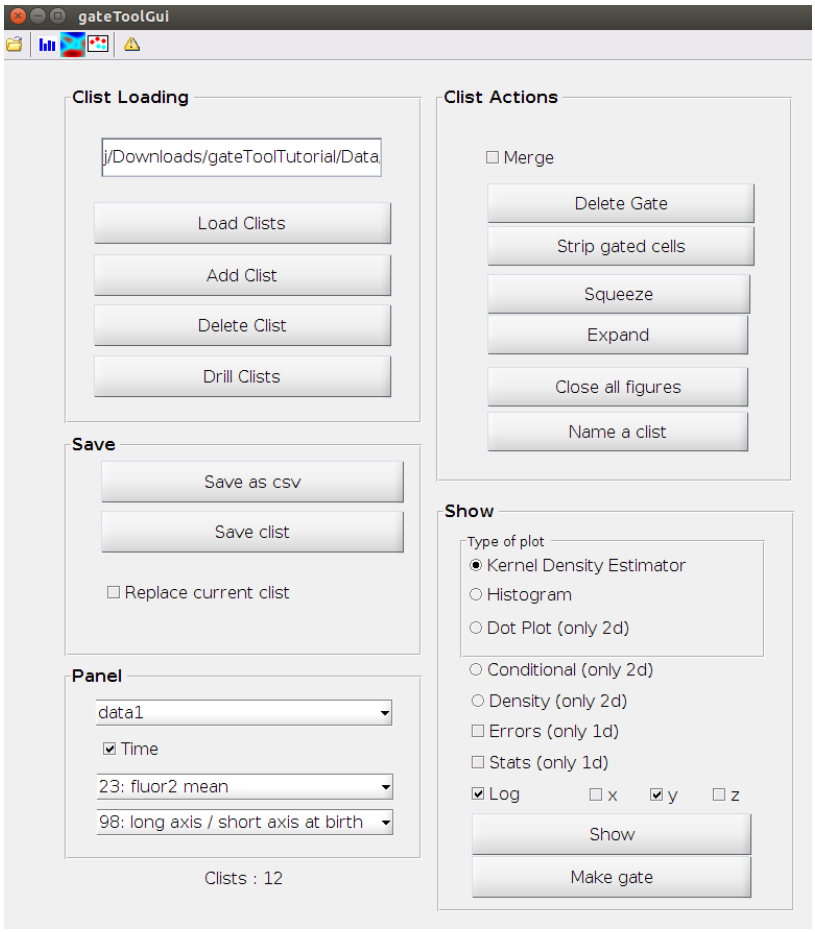

Fig. 1. Image of the `gateTool` GUI.

Command	Argument(s)	Description
`'merge'`		Merge and output all input *Clists* into a single *Clist*
`'strip'`		Remove all un-gated cells from the *Clist* data
`'squeeze'`		Treat all *Clist* entries identically
`'expand'`		Treat all *Clist* entries as different conditions
`'name'`	`name`	Provide a label for a *Clist* (appears in plot legends)
`'color'`	`cc`	Provide a color for a *Clist* (used for plotting *Clist* data)
`'add'`	`data, name`	Add a field (`name`) to the *Clist* data, using input values (`data`)
`'add3d'`	`data, name`	Add a field (`name`) to the 3D *Clist* data, using input values (`data`)
`'get'`	`ind`	Output data from data column `ind`
`'make'`	`ind`	Gate all input *Clists* on index (indices) in `ind`. (`ind` is a vector with either 1 or 2 elements.)
`'getgate'`	`ind`	Outputs the indexed gate

Table 2. *Clist* modification commands.

Command	Argument(s)	Description
'show'	ind	Make a figure for viewing the *Clist*. ind is either and index or pair of indices specifying what is to be visualized. If no ind is passed then all gates are displayed on a single *Clist* input
'time'	ind	Make a temporal plot of single cell dynamics for one index.
'hist'	ind	Make a 1 or 2D histogram (depending on dimension of ind).
'dot'	ind	Make a 2D dot plot (dim. of ind must be 2).
'kde'	ind	Make a 1 or 2D KDE (depending on dimension of ind).
'bin'	bin	Set binning for hist, kde plots. (bin: (1D) scalar is number of bins, vector is set of bin centers, (2D) vector is set of bin numbers for each dim, cell array is two vectors of bin centers)
'den'		Normalize hist,kde plots to show probability density
'cond'		Normalize hist, kde plots to show conditional probability density
'rk'	rk	Set radius of the Gaussian kernel for kde plots
'rm'	rm	Set radius of the point mask (size of plotted point) for kde plots
'mult'	mult	Set resolution for kde plots (i.e. set the number of pixels in the image)
'err'		Show error in 1D hist, kde plots
'inv'		Invert 2D hist, kde plots for printing
'stat'		Show statistics for a 1D show command
'log'	axes	Set of axes to set with log scales. axes = [1,2,3] will set x, y and color axes to log scale
'no clear'		Do not run clear figure (clf) before drawing
'new fig'		Draws new figures for each input *Clist*

Table 3. *Clist* visualization commands. Note: these commands do not modify the *Clist*. All commands in this table must be used in conjunction with the 'show' command. All commands specifying aspects of 'kde' and 'hist' plots must be used in conjunction with the 'kde' and 'hist' commands.

Command	Argument(s)	Description
'def'		Show all *Clist* channel definitions at the command line
'def3D'		Show all *3D Clist* channel definitions at the command line
'xls'	filename	Export an excel document with *Clist* data (Excel must be installed)
'csv'	filename	Export a csv document with *Clist* data
'save'	filename	Save *Clist* to a .mat file
'units'	units	Set a multiplier for the data to set desired units
'drill'		Use recursive loading through a directory tree to any level

Table 4. Additional commands.

Strains and data collection

The analysis for this work was applied to data taken from a variety of new and previous studies. A list of all strains used is provided in Table 5. Data for the "Length as a proxy for cell age" and "Overexpression phenotypes" experiments were initially collected for previous studies (respectively, (Cass et al., 2016) and (Kuwada et al., 2015a; Kuwada et al., 2015b)). For all experiments, slides were prepared using pads of 0.2% LM- agarose (Cat. no. 16520-020). Cells are spotted onto the pads, covered with a coverslip and sealed using VaLP (1:1:1 vaseline, lanolin and paraffin). Imaging was completed using a Nikon Ti-E inverted wide-field fluorescence microscope with a large format sCMOS camera (Andor NEO) and controlled by NIS-Elements. Samples were kept at 30°C throughout the imaging process using an environmental chamber. Image processing and analysis was completed using custom MATLAB (Natick, MA) software

Bacterium	Strain	Vector	Experiment
A. baylyi	ADP1		Morphology gating
E. coli	AB1157	pZA12-GFP	Morphology gating
E. coli	MG1655	pALA2705	Length as age proxy
E. coli	K-12	ASKA	Overexpression phenotpyes
E. coli	MG1655		Cell-cycle duration

Table 5. Strains.

Kernel Density Estimates (KDEs)

Kernel density estimates provide a binning-independent, smoothed representation, or estimate, of the probability density of data. The histogram of values is normalized to provide the probability density, and the resulting values are convolved with a Gaussian kernel, smoothing the resulting values to prevent artifacts from binning. Two example 2D KDE plots are provided in Fig. 3.

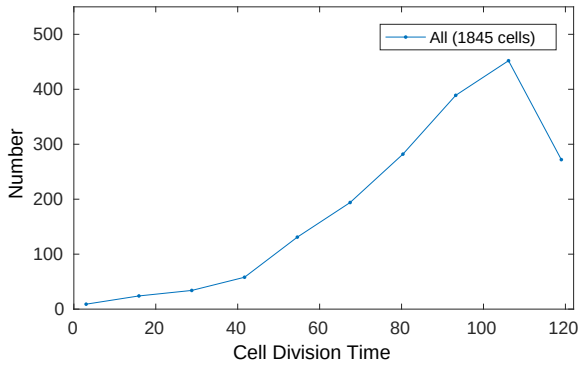

Fig. 2. Histogram of division times for cells in "Length as a proxy for cell age" experiment.

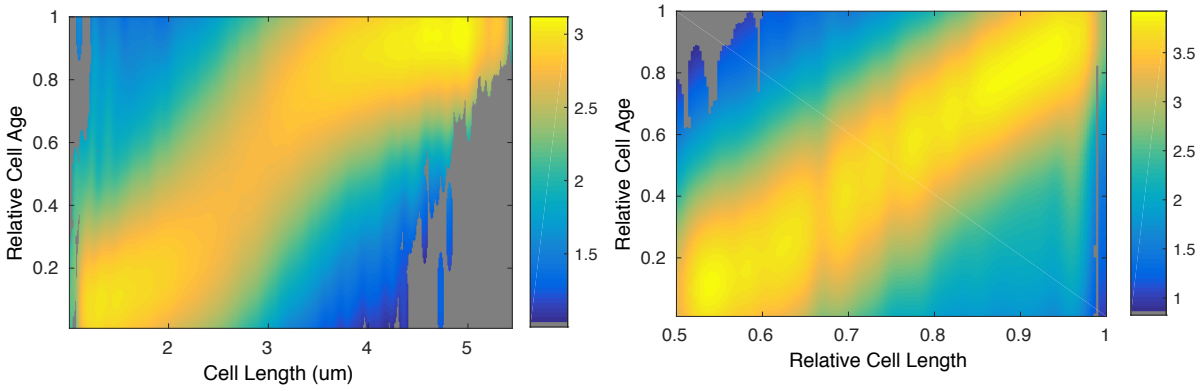

Fig. 3. Conditional Probabilities for Relative Variables. For the "Length as a proxy for cell age" experiment, we provide the conditional probabilities for cell length vs relative cell age (left) and for relative cell length vs relative cell age (right). Here "relative" means relative to the end-of-cell-cycle value (i.e. length and age at the time of cell division.

Computation of p-values for the determinants of growth rate

To compute the p-values for the different determinants of growth rate, we used the non-parameter Kolmogorov–Smirnov Test to compare the distributions and test whether they were generated from the same PDF. We used the MATLAB implementation `kstest2`. Due to the small number of determinants tested, we did not apply a Bonferroni correction to correct for multiplicity.

Estimation of mutual information

The mutual information is defined:

$$M(X,Y) \equiv \mathbb{E}_{\substack{X,Y \\ p(\cdot,\cdot)}} \log \frac{p(X,Y)}{p(X)\,p(Y)}, \quad [1]$$

where p are the pdf and joint pdf distributions for random variables X and Y. To estimate the probability distributions, we define the empirical probability distribution using a Kernel Density Estimate:

$$\hat{p}(x|x^N;\sigma) \equiv N^{-1} \sum_{i=1}^{N} K(x|x_i,\sigma), \quad [2]$$

where K is a Gaussian kernel centered on training observation x_i and σ is the kernel width. The leave-one-out-cross-validated mutual information estimate is:

$$\hat{M}(x^N,y^N;\sigma) \equiv N^{-1} \sum_{i=1}^{N} \log \frac{\hat{p}(x_i,y_i|x^{\neq i},y^{\neq i})}{\hat{p}(x_i|x^{\neq i})\,\hat{p}(y_i|y^{\neq i})}, \quad [3]$$

where $x^{\neq i}$ and $y^{\neq i}$ refer to the data set without the ith observation. Finally we maximize M with respect to the kernel density widths with respect to X and Y.

Descriptor	P-value
Phase Brightness	0.901
Dist. to Colony Edge	0.857
Number of Old Poles	0.012
Cell Length at Birth	0.026

Table 6. P-values for growth rate distributions gated by cell descriptor P-values are calculated for the growth distributions in Fig. 4E. For each cell descriptor, the growth rate distributions for the highest (+) and lowest (-) subpopulations are compared.

References

Cass, J. A., Kuwada, N. J., Traxler, B., and Wiggins, P. A. (2016). Escherichia coli chromosomal loci segregate from midcell with universal dynamics. *Biophysical Journal* **110**: 2597–2609.

Kuwada, N. J., Traxler, B., and Wiggins, P. A. (2015a). Genome-scale quantitative characterization of bacterial protein localization dynamics throughout the cell cycle. *Mol Microbiol* **95**: 64–79.

Kuwada, N. J., Traxler, B., and Wiggins, P. A. (2015b). High-throughput cell-cycle imaging opens new doors for discovery. *Curr Genet* **61**: 513–6.

Stylianidou, S., Brennan, C., Nissen, S. B., Kuwada, N. J., and Wiggins, P. A. (2016). Supersegger: robust image segmentation, analysis and lineage tracking of bacterial cells. *Mol Microbiol* **102**.