

Release notes for the Baxter Algorithms

Klas Magnusson
klasma@kth.se

Created: December 3, 2018

1.4

This release serves as the initial reference point. Before this release, intermediate unlabeled versions were sometimes used for processing. After version 1.4, all new versions have a number.

1.4.2

This version introduces functions to analyze centrally located nuclei in fluorescently stained muscle sections.

1.4.3

This version introduces a function to plot the number of live cells, dead cells, and ghost cells over time. Ghost cells are cells that would have been created if the dead cells had never died. This plot can be used to analyze proliferation in a way which is not biased by differences in cell death and the time between divisions.

1.5

While version 1.4 was an unfinished software under development, version 1.5 is the first release which can be seen as a finished software. The software still needs additional user testing before it can be released publicly, but a user should be able to use all functionalities of the software after reading the user guide and watching the video tutorials. The functions on the "development" level in the menu system are exceptions to this rule. Those functions have not gone through the same thorough testing as the other functions, and are not meant to be used for analysis of biological data. The following major changes have been made since version 1.4.3.

1. A large number of bugs have been fixed. Most of them were in the graphical user interfaces (GUIs). I will not list bug-fixes here unless they had a significant impact on the software.

2. The entire software is now documented with comments in the code.
3. The folder structure of the source code has been rearranged.
4. Development code and project specific code has been removed.
5. A startup script has been added. The script adds all necessary paths if MATLAB is started in the program directory. This is convenient if you are developing your own code, but otherwise it does not matter, as the Baxter Algorithms adds the paths automatically when you start the program.
6. The image stabilization module has been re-implemented in MATLAB. Previously, the image stabilization was run through ImageJ. The new code produces almost identical results, with comparable runtime. The new code has support for GPU processing, and therefore it is about 3 times faster on a computer with a good graphics card than on a computer without a graphics card. The new image stabilization also has the option to crop the images instead of extrapolating missing pixel values. This gets rid of a lot of segmentation artifacts. Another improvement is that the new code can handle image sequences with multiple microscopy channels.
7. A segmentation algorithm based on template matching has been added. You can define your own templates in the segmentation GUI.
8. A tophat filter can now be used to remove non-uniform illumination.
9. It is now possible to load 3D segmentations from 16 bit label images. This allows you to perform the segmentation using other programs. Previously, this was only possible for 2D data.
10. The lists of tracking scores that are used for track linking now have 1-based indexing instead of 0-based indexing. MATLAB uses 1-based indexing for everything and therefore it makes sense to have 1-based indexing in the MATLAB code and then convert to 0-based indexing in the C++ code.
11. Track linking of false positive tracks is now done using the same track linking algorithm as the linking of the cell tracks. Previously, it was done using an algorithm based on frame by frame matching.
12. The code which locates files and folders in the imaging experiments has been made faster. This makes the user interfaces respond faster when you are working with large datasets.
13. The computation of mitosis scores has been re-implemented in a faster way. This has a significant impact on the runtime for image sequences with hundreds of cells.

14. Caching of cell objects has been removed from the function which loads tracking results. This got rid of some bugs and also made the program faster.
15. The thumbnails in the main program are now updated if you change settings which determine how images are displayed.
16. All buttons and other controls have been given tooltip messages.
17. Message boxes with instructions have been added to many GUIs. The messages are displayed when the GUIs are opened, but can be turned off by pressing "Do not show again" in the message boxes or by pressing "Help/Information dialogs/Display none".
18. A new GUI for loading of settings has been created. The new GUI has descriptions and sample images for all settings files. It is still possible to load settings directly from csv-files using the menu option "Load Settings (browse for file)".
19. The GUI for cropping of circular microwells can now handle datasets with multiple microscopy channels.
20. The step lengths of the time sliders in players have been changed. Clicking on the slider moves 10 % of the image sequence forward or backward. The interval between played frames now applies to both playing and stepping from frame to frame using the arrow keys. Clicking on the arrow buttons of the slider will however still move only one frame.
21. It is now possible to go to a specific frame in players by entering the frame index in a textbox.
22. It is now possible to pan after you have zoomed in, by pressing "m" or holding down "space".
23. Bugs related to keyboard shortcuts of control objects have been fixed. Previously, pressing a button in a player could both affect a control object and trigger a keyboard shortcut. For example, pressing the up-key in a dropdown menu in the segmentation GUI would both change the value in the dropdown menu and start playing the image sequence. This has now been changed so that keyboard shortcuts take precedence unless the control is a textbox. In textboxes, the keyboard shortcuts are not executed.
24. The plotting of holes in segmented regions during segmentation editing has been improved. Previously, the outlines of the holes did not follow the pixel boundaries.

25. Manually corrected tracking results now have log files with user notes and information about the program version that was used to save the results.
26. You can now train your own classifiers for cell counts, mitosis, and apoptosis. This used to be an unfinished development feature, but the code has now been cleaned up and tested.
27. The analysis of fluorescence intensities in the analysis GUIs is now fully functional.
28. All plotting functions in the Plot GUI have been cleaned up and tested. This GUI contains all advanced plotting functions which do more than just visualize parameters of individual cells. The entire GUI used to be a development feature.
29. The Scatter Plot Analysis GUI has been added. In this GUI, cell parameters can be plotted against each other in a 2-dimensional coordinate system.
30. The setting `authorStr` now works as intended. The setting allows you to enter a list of authors which will be used in all exported pdf documents that are created using an image sequence. The first image sequence to be processed decides the author list if multiple author lists have been defined for the processed image sequences.

1.5.1

This version has a number of bug-fixes that may be very important, depending on your application and MATLAB version.

1. The implementation of object oriented programming seems to have changed in MATLAB 2015b. This made callbacks in the manual correction GUI extremely slow for large datasets. To avoid this issue, the cell data is now stored using persistent variables in functions, and not as properties in the manual correction GUI class.
2. A work-around for a segfault, which occurred in the built in function `imdilate`, has been created. The segfault occurs when the first input to `imdilate` is a logical column vector. This only happened when blobs that had pixels in a single column were merged. This is now avoided by performing the operations on double arrays.
3. 5D data (3D image sequences with multiple channels) can now be processed and visualized properly. There was an error when properties related to the fluorescence intensity were computed, and maximum intensity projections were not computed properly.

4. The bipartite matching step, which is used to link fragments of clusters that have been broken, has now been improved. The new code removes impossible migrations when the Jaccard index is used to compute migration scores. This is important, as migrations were created randomly if the cluster breaking removed the overlap between objects that were linked in the original tracks. That rarely happened, but it looked very bad when it happened. The new code also allows the number of appearances and disappearances to change if that improves the overall score.
5. The lines representing the minimum and maximum fluorescence intensities are now displayed correctly when the GUI for specification of fluorescence settings is opened.

1.5.2

Optimized loading of segmentations from other software. The new code creates a file list for the folder with label images in the beginning of the loading procedure instead of doing it once for every loaded frame. This improves the performance significantly for large datasets and datasets on file servers.

1.5.3

Fixed a bug introduced in version 1.5.2. The bug caused an error in the segmentation GUI when the user tried to select loading of segmentation results from another software.

1.6

Version 1.6 is the first version of the software to be released publicly. The source code is released under the MIT license in a new repository without any history. A lot of side projects and other unused code has been excluded from the new repository. A lot of bugs and typos have been fixed, and many small improvements have been made. The software has now been thoroughly tested, except for the functionality under the development level in the GUI. The major changes compared to version 1.5.3 are described below.

1. The renaming of the program from "Baxter Algorithm" to "Baxter Algorithms" is now complete. All files and folders with "BaxterAlgorithm" in the name have now been renamed.
2. The software has been updated and tested to work in both MATLAB 2015b and 2018b. It should also work for all MATLAB releases in

between, but there is no guarantee that it will work in future releases, as backward compatibility is not always maintained between MATLAB versions.

3. It is now possible to perform automated optimization of segmentation parameters. The program uses coordinate ascent to optimize the SEG measure, which was used to measure segmentation performance in the cell tracking challenges. To achieve good results, the user needs to outline ~ 100 representative cells manually.
4. Tracks and associated segments can now be exported to the formats used in the cell tracking challenges. The output can be saved as tracking and segmentation results, tracking ground truths, or segmentation ground truths. Our implementation is an extension of the format used in the cell tracking challenges, which makes it possible to save information about death events and false positives.
5. Functions for evaluation of segmentation and tracking results have been added. The software now implements functions to evaluate the performance measures TRA/AOGM and SEG, which were used to evaluate tracking and segmentation performance respectively in the cell tracking challenges. We have also implemented a relaxed version of the SEG measure, which gives a more fair evaluation of segmentation algorithms that suffer from severe under-segmentation.
6. The zoom tool has been changed so that right-clicking zooms out to the view that was shown before the user zoomed in. This is more efficient and intuitive compared to always zooming out by a factor of 0.5 as was done previously.
7. The user has been given more control over training of custom classifiers.
8. The Cell Analysis GUI has been augmented with lineage trees where the branches are colored based on a cell property, such as the size or fluorescence intensity.
9. Lineage trees have been changed so that only cell divisions are marked with dots. Previously, beginnings of tracks and beginnings of track fragments after missing frames were marked with dots. This cluttered the lineage tree and made it harder to interpret.
10. The file format from the cell tracking challenges is now used as a fallback when saving to mat-files fails. Saving can fail due to an out-of-memory error when large tracking results are saved. This is because the compression used in mat-files requires a lot of RAM. Previously, v7.3 mat-files without compression were used as a fallback. We have

however seen examples where corrupt files were saved for extremely large tracking results. Therefore we switched to saving to our updated version of the file format used in the cell tracking challenges. These files take longer to load, but they are usually smaller than the v7.3 mat-files and they never become corrupted.

11. The directions of the z -axes in the XZ- and YZ-views of 3D projections have been changed so that the first z -slice is at the top in the XZ-view and to the left in the YZ-view. This is the conventional way to display 3D projections. It is more intuitive to see the bottom of the sample (relative to the objective) at the bottom in the XZ-view.
12. A bug has been fixed, so that the cell positions always coincide with the centroids of the blob regions. Previously, the positions were not always updated when blob regions were merged.
13. All spread sheets are now exported to csv-files instead Excel files. The program is therefore no longer dependent on Excel. The csv-files can however be opened in Excel.
14. In the fiber analysis GUI, it is now possible to select whether or not fibers that touch the image border should be included in the analysis.
15. The different GUIs for data analysis can now handle 3D datasets. Plots of axis ratio, total distance traveled, and other parameters which were previously only available for 2D datasets have been generalized for 3D datasets.
16. The time to first division is now only defined for cells which are present in the first image. Previously, it was defined for all cells which did not have a parent in the lineage tree. This incorrectly included all cells which entered the field of view after the beginning of the experiment.
17. The manual correction tool for continuous addition of cell positions can now add a cell position in the first frame that it is used in. This makes it possible to perform completely manual tracking very efficiently.