## Modifications of u-track since the publication of the paper (Jaqaman et al., Nat. Methods 2008):

### 12 December 2008

**(1)** An additional Kalman filter function (default: `kalmanReverseLinearMotion`) to time-reverse Kalman filter information for the second and third rounds of frame-to-frame linking.

---

### 26 June 2009

**(2)** Diagnostics:

**(2A)** Gap closing time window: If the additional field "gapCloseParam.diagnostics" is set to 1, the software will plot in the end of tracking a histogram of the closed gap lengths. This will help with assessing the quality of gap closing. Generally speaking, longer gaps should be less frequent than shorter gaps; thus, a gap length histogram with a plateau might be indicative of a too large gap closing time window.

**(2B)** Maximum search radius: The frame-to-frame linking cost function has the additional input "parameters.diagnostics", through which the code will output histograms of frame-to-frame linking distances at the specified frames. For example, if parameters.diagnostics = [2 35], then the histogram of linking distances between frames 1 and 2 will be plotted, as well as the overall histogram of linking distances for frames 1->2, 2->3, ..., 34->35. The histograms can be plotted at any frame except for the first and last frame of a movie. To not plot, enter [ ].

**(3)** Gap length penalty: The gap closing cost function has the additional input "parameters.gapPenalty" to penalize longer gaps. If parameters.gapPenalty has the value $p$, then the penalty for a gap of length p will be $p$^$n$. Note that for $p > 1$ longer gaps are penalized, for $p = 1$ there is no gap length penalty, while for $p < 1$ longer gaps are favored.

**(4)** Resolution limit: The gap closing cost function has the additional input "parameters.resLimit", representing the resolution limit in pixels. The resolution limit is generally the Airy disk radius, but it could be smaller when iterative Gaussian mixture-model fitting is used for detection. The resolution limit is used to expand the merge/split search radius if it is found to be smaller than the resolution limit.

---

### 25 April 2010

**(5)** Detection code bug fix: The previous version of the detection code was overestimating the position uncertainties by a factor of (PSF sigma)^2.

**(6)** Code modifications to handle larger datasets: Both the detection and tracking codes have been modified to better handle larger datasets. This has no effect on their input or output though.

**(7)** New cost functions costMatLinearMotionLink2 and costMatLinearMotionCloseGaps2:

These new cost functions are similar to their predecessors, with a few “behind-the-scenes” changes.

**(7a)** Distance cost scaling in costMatLinearMotionCloseGaps2: For gap closing, merging and splitting, the part of the cost based on distance is now scaled by the average frame-to-frame displacement of the track segments involved. This avoids punishing particles that are more mobile relative to those that are less mobile.

**(7b)** Alternative costs in costMatLinearMotionCloseGaps2: Previously, the gap closing alternative cost was taken as the 90th percentile of the costs of all potential assignments. Furthermore, the merging and splitting alternative cost was determined on a case-by-case basis. In the new cost function, all alternative costs are assigned the same value, taken as the X percentile of the distribution of potential assignment costs, where X is calculated from the structure of the matrix of potential assignments (in particular, it takes into account the number of potential assignments each track segment has).

**(7c)** Auxiliary (lower right) block costs in costMatLinearMotionLink2 and costMatLinearMotionCloseGaps2: Those were previously assigned to be the smallest costs in the cost matrix, the goal being that they should not influence the LAP outcome. But, in retrospect, having the lowest costs might favor them, thus influencing the LAP outcome. Thus, they have been modified to be equal to the alternative costs, so that they are truly neutral and do not influence the LAP outcome.

---

**1 April 2011**

**(8)** Detection – using absolute background information: The detection code can take additional input arguments that supply it with images of “absolute background” and an alpha-value to compare local maxima to this absolute background.

“Absolute background” is usually a cropped subpart from the original images, where the cropped area lies outside of the cell. Since this area tends to be quite dimmer than inside the cell, one can use a stricter alpha-value, for example 0.001, to compare local maxima to this background area. The use of a stricter alpha-value for comparison with background outside of the cell and a less strict alpha-value for comparison with local background inside the cell minimizes false positives outside of the cell AND false negative inside the cell, where the objects of interest are located.

If this option is used, there must be an absolute background image for each original image.

**(9)** costMatLinearMotionCloseGaps2 – explicit definition of power for scaling search radius with time:

In the previous version of costMatLinearMotionCloseGaps2, the user only defined "timeReachConfB" and "timeReachConfL." Given these parameters, the code internally scaled the search radius with the square root of time before "timeReachConfB" and "timeReachConfL," and with time^0.01 after.

In the new version, the user defines these powers explicitly, to give more flexibility. Setting the new parameters brownScaling and linScaling to [0.5 0.01] is equivalent to the old settings.

---

**21 December 2011**

**(9)** Tracking code modifications to improve speed: The tracking code has been internally modified to enhance its speed; it runs faster and no longer slows down almost exponentially with increase movie length. No effect on input or output.

**(10)** New cost functions costMatRandomDirectedSwitchingMotionLink and costMatRandomDirectedSwitchingMotionCloseGaps:

These new cost functions are similar to their predecessors with one added motion model option, namely moving in a directed manner in a certain direction without the possibility of immediate direction reversal as was the case before. These cost functions have 3 options for motion propagation (instead of 2):

linearMotion = 0: one motion model, namely random (Brownian) motion.
linearMotion = 1: two motion models, namely random motion and movement with constant velocity.
linearMotion = 2: two motion models, namely random motion and movement along a straight line but with the possibility of immediate direction reversal.

What was linearMotion = 1 in the previous cost functions corresponds to what is linearMotion = 2 in the new cost functions. Everything else is the same.

Examples: Motor-driven movement is best tracked with linearMotion = 1. One-dimensional diffusion is best tracked with linearMotion = 2. Diffusive movement without any drift or directionality is best tracked with linearMotion = 0.

---