Simple cell tracking algorithm

May 24, 2016

Modern devices have been used to get data from many biological processes that occur with cells. Much of this data are 3D/4D images. Computer techniques have been important in studies of cells. The understanding of many of these biological processes involves segmentation and tracking of cells. General applications that can process several different cells and processes are required to keep increasing the understanding of this field.

The goal of this simple project is try to understand all of the problems to be solved in these cells studies. This first version works with 2D videos. For the first step, a simple algorithm was developed to process basic situations, where a smaller number of events can happen. We consider here a perfect scenario the case where all cells were correctly segmented in each frame of the video, the number of cells from the frame on the time t to $t+1$ is the same and no overlap, no problem with too close cells or high velocity happens. However, some events are allowed, the size changing and n number of division of cells. So in this case, only one event can change the number of cells from the frame on the time t to $t+1$. This first simple version of the algorithm is described below.

Data: A set I of 2D images, where each image represents the features of the cells on the time t.
Result: A set L of lists, where each element t of each list L_i have the position of the cell i on the time t

$I \leftarrow loadVideo()$;
$T \leftarrow$ number of elements in I ;
$t \leftarrow 1$;
$P_t \leftarrow$ set of particles in the image I_t ;
for *each list i in* L **do**
  add the element i of P_t to L_i
end
for *t=2 to t=T* **do**
  $P_t \leftarrow$ set of particles in the image I_t ;
  for *each element i in* L **do**
    $c \leftarrow$ the element of P_t where the distance is the smallest to element t-1 in L_i
    add c in L_i
  end
  if *any element left in* P_t **then**
    $R \leftarrow$ list of elements left in P_t ;
    for *each element k of* R **do**
      $c \leftarrow$ the element t-1 of any list L_i where the distance is the smallest to element k in R
      $e \leftarrow$ the element t of the list L_i ;
      remove the element t from L_i ;
      create 2 new list L_{i+1} and L_{i+2} in L ;
      add the element e to L_{i+1} ;
      add the element k of R to L_{i+2}
    end
  end
end

Algorithm 1: Simple tracking cells