Change Children: Allows the user to correct the parents during the mitosis stage.

By: Maria Enokian

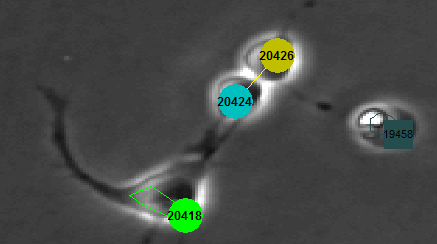


Fig 1. 20418 should be the parent of 20424 and 20426

During regimentation sometimes it will hiccup when two cells are close to each other, or one cell moves randomly to that particular area. That’s when a mix up of parents occurs. Here are steps to fixing this issue with the Change Children tab.

1. Click on the desired cell either 20424 or 20426 in this case and a menu will show up. Then click on the “Change Children” tab.

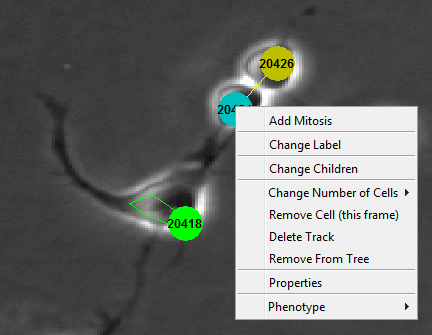


Fig 2. Menu shown when user right clicks on desired cell

1. Type the desired cell parent you want to swap these children with. For this case the parent will be 20418.

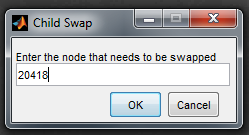


Fig 3. Menu shown when user clicks on child swap and types desired parent.

1. Finished the mitosis is corrected with the correct parent.

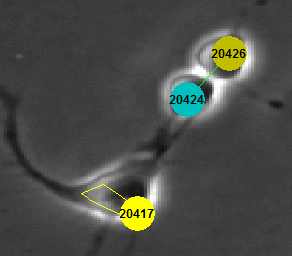


Fig 4. This is the corrected parent for the mitosis creating the children 20424 and 20426

Programmers Guide:

These are the Files that are affected while using the change child action. This shows the path it take in order to get to do the action correctly:

**UI.CreateContextMenuCalls.m**

uimenu(Figures.cells.contextMenuHandle,...

'Label', 'Change Children',... % will display Change Children in the right click menu

'CallBack',@changeChildren,... % route the code to the changeChildren function further down the code

'Separator','on'); % will add separator

On line 47-50 on the UI.CreateContextMenuCalls.m creates the tab for users to right click in the cell menu.

function changeChildren(src,evnt)

global Figures

% Will Switch two nodes together followed by the children and the rest of

% the tree.

[hullID trackID] = UI.GetClosestCell(0);

if(isempty(trackID)),return,end

Editor.ContextChangeChildren(Figures.tree.familyID,Figures.time,trackID);

UI.DrawTree(Figures.tree.familyID);

end

On line 140-148 on the UI.CreateContextMenuCalls.m will direct the code to the Editor.ContextChangeChildren which takes three inputs. The inputs are FamilyID, time, and trackID. The familyID is the tree ID of the tree that the user is using. The time is the current time frame, and the trackID is the cell that was right clicked by the user.

**Editor.ContextChangeChildren.m**

global CellTracks

global Figures

% This will prompt the user and ask for what child should be swapped.

newTrackID = inputdlg('Enter the node that needs to be swapped','Child Swap',1,{num2str(trackID)});

if(isempty(newTrackID)),return,end;

newTrackID = str2double(newTrackID(1));

% If there isn't a parent ID, will send error. Example: Root Node

parentTrackID = CellTracks(trackID).parentTrack;

if (isempty(parentTrackID))

warndlg('This is the root node and will not be able to switch parents');

return;

end

% If the track doesn't exist it will send a warning.

if ( newTrackID > length(CellTracks) )

warn = sprintf('Track %d does not exist, use "Remove from Tree" instead.',newTrackID);

warndlg(warn);

return;

end

% if the cell doesn't exist in the current frame it will send a warning.

if(isempty(CellTracks(newTrackID).hulls))

warn = sprintf('Track %d does not exist, cannot switch children',newTrackID);

warndlg(warn);

return

end

% if the cells are the same it will send a warning.

curHull = CellTracks(newTrackID).hulls(1);

if ( newTrackID == parentTrackID )

warndlg('These are the same cells');

return;

end

% case with cells that already split and cells that aren't in the

% current frame time.

if ( 0 == Tracks.GetHullID(time, newTrackID) )

warndlg('This cell does not exist in the current frame.');

return;

end

% if the cell does not exist untill later on in the tree it will send a

% warning message.

if ( time < CellTracks(newTrackID).startTime )

warn = sprintf('Cannot switch children from %d to %d, track %d does not exist until frame %d.',parentTrackID, newTrackID,newTrackID, CellTracks(newTrackID).startTime);

warndlg(warn);

return

end

% if all the errors are good it will prompt the Editor.ChangeChildren

% function as a replayable action.

bErr = Editor.ReplayableEditAction(@Editor.ChangeChildren,familyID,trackID, newTrackID, time);

if ( bErr )

return;

end

% if the swap is successful in the global Log stack it will display

% the words 'Children Swapped' in the action field

Error.LogAction('Children Swapped',parentTrackID,newTrackID);

newTrackID = Hulls.GetTrackID(curHull);

Tracker.UpdateHematoFluor(time);

UI.DrawTree(CellTracks(newTrackID).familyID);

UI.DrawCells();

end

Editor.ContextChangeChildren will prevent many errors from happening during the switching process before it switches. This is because the tree has to be protected and the user doesn’t want the program to crash. When all the errors are checked it will take you where the swap happens. This is the Editor.ChangeChildren file which takes four variables. These are familyID, trackID,newTrackID, and time. This is where new track ID is the desired parent that is going to be swapped with the trackID’s parent.

**Editor.ChangeChildren.m**

% historyAction = ChangeChildren(Family ID, trackA, trackB, time)

% Edit Action:

% Switches Children

% Toggles lock on familyID

% Context Swap Labels

% Toggle lock on family ID

% Maria Enokian

function historyAction = ChangeChildren(familyID,trackID,newTrackID, time)

% Initialized variables

global CellFamilies CellTracks Figures

trackID = CellTracks(trackID).parentTrack;

trackA = trackID;

trackB = newTrackID;

Ntime = CellTracks(trackID).endTime;

Tracker.GraphEditSetEdge(trackA, trackB, Ntime);

Tracker.GraphEditSetEdge(trackB, trackA, Ntime);

% This will unlock the tree and if the cell family is locked it will

% unlock it but otherwise it will lock it.

if ( isempty(CellFamilies(familyID).bLocked) )

CellFamilies(familyID).bLocked = 0;

end

bIsLocked = CellFamilies(familyID).bLocked;

Helper.SetTreeLocked(familyID, ~bIsLocked);

% This will swap the two children together to correct the mitosis

bLocked = Helper.CheckTreeLocked([trackA trackB]);

if ( any(bLocked) )

Tracks.LockedSwapLabels(trackA, trackB, Ntime);

else

Tracks.SwapLabels(trackA, trackB, Ntime);

end

% This will unlock the tree and if the cell family is locked it will

% unlock it but otherwise it will lock it.

if ( isempty(CellFamilies(familyID).bLocked) )

CellFamilies(familyID).bLocked = 0;

end

bIsLocked = CellFamilies(familyID).bLocked;

Helper.SetTreeLocked(familyID, ~bIsLocked);

UI.DrawTree(Figures.tree.familyID);

historyAction = 'Push';

end

Editor.ChangeChildren has three main steps. These go as follows:

1. Unlock the tree
2. Swap the two labels
3. Relock the tree

Unlocking the tree is necessary in order to have the change happen throughout the entire tree structure. Next, The trackID has to be the parents trackID in order to do that we reinitialize the variable trackID with new parameters.

trackID = CellTracks(trackID).parentTrack;

This code snippit above: allows the user to find the parent track of the track that was chosen.

After, what happens is that because sometimes the user could forget to click beforehand on the cell during mitosis to fix that a variable has to be set for Ntime. This is shown in the code snippit below.

Ntime = CellTracks(trackID).endTime;

This variables gets the last frame trackID has before it splits to mitosis.

Then the swapping labels allow the two tracks to switch places, and also it allows the user to have this mitosis corrected for them. Finally when the two labels are successfully swapped the tree is relocked to protect the tree from further changes, and then pushed to the undo stack.