Benchmarking DeLTA tracking

To assess the performance of DeLTA to track individual cells over time and match daughter cells to their mother during divisions, you can perform manual tracking on a subset of your images and compare them to the results from DeLTA. For the manual tracking of cells I have been using the manual tracking workflow in Ilastik.

1. Preparing images for Manual tracking in Ilastik

For the manual tracking, we will need to export the label images from DeLTA as well as the binary segmentation mask. The following Python scripts exports both to two separate folders that subsequently can be used for the manual tracking workflow in Ilastik.

```
import delta
import os
from PIL import Image


# Load DeLTA data
dlt_data = delta.pipeline.Position(None,None,None)
dlt_data.load(os.path.join('delta_results', 'Position000000.pkl'))
label_stack = dlt_data.rois[0].label_stack

# Create directory for label stack images
ids_dir = os.path.join('delta_results', 'ids')
if not os.path.isdir(ids_dir):
    os.makedirs(ids_dir)

# Create directory for segmentation images
seg_dir = os.path.join('delta_results', 'seg')
if not os.path.isdir(seg_dir):
    os.makedirs(seg_dir)

# Save individual frames (labels and segmentations)
for i, frame in enumerate(label_stack):
    print(f'saving frame {i:03}')
    ids = Image.fromarray(frame)
    seg = Image.fromarray(frame>0)
    ids.save(os.path.join(ids_dir, f'ids_T{i:03}.png'))
    seg.save(os.path.join(seg_dir, f'seg_T{i:03}.png'))
```

2. Manual tracking in Ilastik

You can find a detailed tutorial for Ilasik workflows [here][manual tracking workflow in Ilastik](https://www.ilastik.org/documentation/tracking/tracking#sec_manual).

Here just a quick summery of the different steps of the workflow

1. Input Data: - **Raw Data:** select the directory with the exported label images (ids) - **Prediction Maps:** select the directory with the exported segmentation masks (seg) Add New… > Add a single 3D/4D Volume from Sequence… > Whole Directory > select the folder (either ids or seg)

2. Threshold and Size Filter: - Method: Simple - Input: 0 - Smoth: 0 0 - Threshold: 1 - Size Filter: Min=10, Max=1000000 - hit Apply

3. Manual Tracking: - Perform the tracking as described in the manual tracking workflow - Export Divisions as a CSV … (can be selected on the bottom of the top left panel)

4. Tracking Result Export: Export Settings - Source: Manual Tracking - Choose Export Image Settings… - Convert to Data Type: unsigned 16-bit - Format: png sequence - Export All - Move resulting manual tracking images to a folder named trk - move both the divisions.csv and the trk folder to the delta_results folder

3. Comparing tracking results

The following script uses the divisions.csv and the manual tracking images to check the tracking performed by DeLTA.

```python
import delta
import numpy as np
import os
import imageio.v3 as imgio
import pandas as pd

def get_cids(cell):
    '''
    Creates cell IDs based on the id and the generation
    '''
    cids = []
    gen = 0
    for d in cell['daughters']:
        if d != None:
            gen += 1
        cids.append(f'{cell["id"]}_{gen}')
    return(cids)

def get_did(cell):
    '''
    Returns a list of cell IDs of the daughter cells (None if no division)
    '''
    did1 = []
    did2 = []
    gen = len([d for d in cell['daughters'] if d!=None]) + 1
    did1_cur = None
    did2_cur = None
    for d in reversed(cell['daughters']):

        did1.append(did1_cur)
        did2.append(did2_cur)

        if d != None:
            gen -= 1
            did2_cur = f'{d}_0'
            did1_cur = f'{cell["id"]}_{gen}'

    return(list(reversed(did1)), list(reversed(did2)))



# Read the images from the manual tracking containing the individual cell IDs
# The same IDs are alos in the divisions.csv file
ila_stack = []
trk_imgs = sorted([f for f in os.listdir(os.path.join('delta_results', 'trk')) if f.endswith
for img in trk_imgs:
    ila_stack.append(imgio.imread(os.path.join('delta_results', 'trk', img)))

# Load the DeLTA results and get the individual cell data and the label stacks
dlt_data = delta.pipeline.Position(None,None,None)
dlt_data.load(os.path.join('delta_results', 'Position000000.pkl'))
cells = dlt_data.rois[0].lineage.cells
label_stack = dlt_data.rois[0].label_stack


# Match the cell id from DeLTA and the manual tracking
# 'cid_lut' contains echa cell id from DeLTA and the corresponding
# track id from the manual tracking, for each frame
cid_lut = {}
for cell in cells:
    cell['cid'] = get_cids(cell)
    cell['did1'], cell['did2'] = get_did(cell)
    cell['track_label'] = []
    for i, frame in enumerate(cell['frames']):
        if frame not in cid_lut.keys():
            cid_lut[frame] = {}
        cell_label = cell['id'] + 1
        trk_label = np.unique(ila_stack[frame][label_stack[frame] == cell_label])
        if len(trk_label) == 1:
            cell['track_label'].append(trk_label[0])
            cid_lut[frame][cell['cid'][i]] = trk_label[0]
        else:
            cell['track_label'].append(0)
            cid_lut[frame][cell['cid'][i]] = 0

# Checks for each cell if the track id stays the same over all frames
for cell in cells:
    cell['track_error'] = []
    track_label = cell['track_label'][0]
    cid = cell['cid'][0]
    for i, frame in enumerate(cell['frames']):
        if cell['cid'][i] != cid:
            cid = cell['cid'][i]
            track_label = cell['track_label'][i]

        if cell['track_label'][i] != track_label:
            cell['track_error'].append(True)
        else:
            cell['track_error'].append(False)

# Export cell information with a additional column indicating if there are tracking errors
df_cells_dlt = pd.concat([pd.DataFrame.from_dict(cell) for cell in cells])
df_cells_dlt.to_csv('delta_cells.csv')



# Open the manually tracked divisions.csv and check if the divisions from DeLTA
# correspond to the manually tracked divisions
df_divs_ils = pd.read_csv(os.path.join('delta_results', 'divisions.csv'))
divs_ils_list = list(zip(df_divs_ils['timestep_parent'],
                    df_divs_ils['track_id_parent'],
                    df_divs_ils['track_id_child1'],
                    df_divs_ils['track_id_child2']))


divs_dlt = {'timestep_parent' : [],
            'track_id_child1' : [],
            'track_id_parent' : [],
            'track_id_child2' : [],
            'correct' : []}
for cell in cells:
    for i, frame in enumerate(cell['frames']):
        if cell['daughters'][i] != None:
            daughter_track_labels = sorted([cid_lut[frame][cell['did1'][i-1]],
                                            cid_lut[frame][cell['did2'][i-1]]])
            divs_dlt['timestep_parent'].append(frame - 1)
            divs_dlt['track_id_parent'].append(cid_lut[frame-1][cell['cid'][i-1]])
            divs_dlt['track_id_child1'].append(daughter_track_labels[0])
            divs_dlt['track_id_child2'].append(daughter_track_labels[1])

            div_id = (divs_dlt['timestep_parent'][-1],
                        divs_dlt['track_id_parent'][-1],
                        divs_dlt['track_id_child1'][-1],
                        divs_dlt['track_id_child2'][-1])
            divs_dlt['correct'].append(div_id in divs_ils_list)

# Export a table of all division found by DeLTA and if they were
# also found in the manual tracking
df_divs_dlt = pd.DataFrame.from_dict(divs_dlt)
df_divs_dlt.to_csv('delta_divisions.csv')
```