

Dataset_to_plate.py script changes.

Changes added to the script allow for more flexible placement of wells on the plate when converting dataset to a plate.

The screenshot shows the 'Dataset To Plate.py' script interface. It includes a title bar, a description box, and various input fields and buttons. The description box contains the following text: 'Take a Dataset of Images and put them in a new Plate, arranging them into rows or columns as desired. Optionally add the Plate to a new or existing Screen. See <http://www.openmicroscopy.org/site/support/omero4/users/client-tutorials/insight/insight-util-scripts.html> Authors: William Moore, OME Team Contact: ome-users@lists.openmicroscopy.org.uk Version: 4.3.2'. Below the description box are several input fields and buttons: 'Data Type: * Dataset' (dropdown), 'Remove From Dataset: ☒' (checkbox), 'IDs: *' (text input), 'Filter Names: ' (text input), 'First Axis: * column' (dropdown), 'First Axis Count: * 12' (text input) with 'Min: 1' (text input), 'Column Names: * number' (dropdown), 'Row Names: * letter' (dropdown), 'Sorting: * natural' (dropdown), 'Wells To Skip: ' (text input), 'Column Offset: 0' (text input), 'Row Offset: 0' (text input), and 'Screen: ' (text input). At the bottom right, there is a yellow tooltip that says 'Offset to first acquire'. At the bottom left, there is a 'View Script' button. At the bottom right, there are 'Cancel' and 'Run Script' buttons.

List of changes:

- Row and column offset. If acquired data does not start at column 1 or row 1, images can be offset to start at arbitrary row or column.
- Well skipping. Allows to leave empty wells if data for a given well is not provided.
- Natural sorting. Option added to place images in correct wells for different naming conventions.

When data available does not start at row 1 and/or column 1 user can input offset values to **Row Offset** and **Column Offset** edit boxes which will leave Offset number of rows and/or columns empty.

When not all of the images are present and some wells should be left empty user can input list of wells to skip to **Wells To Skip** edit box in following format:
Row:Column,Row:Column,Row:Column,

If images are named using natural naming convention: Image 1, Image 2, ... , Image 10, ... , natural sorting option should be selected from **Sorting** combo box as Alphabetical sorting will result in incorrect image order: Image 1, Image 10, , Image 2,

Example Use.

Example dataset can be found at:

https://github.com/emilroz/ExampleData/tree/master/Dataset_to_Plate.py This data set was created to test new functionality.

1. Running script without new functionalities for this dataset results in wrong plate layout.

Settings (emulate previous version of the script):

- First Axis: Row (images are labeled row:column)
- Axis Count: 8 (highest row number)
- Sorting: alphanumeric
- Wells to skip:
- Column offset: 0
- Row offset: 0

test 59

- ...2013-r1.nd [Stage1 "r1c3"].tif
- ...013-r1.nd [Stage10 "r2c4"].tif
- ...013-r1.nd [Stage11 "r3c4"].tif
- ...013-r1.nd [Stage12 "r4c4"].tif
- ...013-r1.nd [Stage13 "r5c4"].tif
- ...013-r1.nd [Stage14 "r6c4"].tif
- ...013-r1.nd [Stage15 "r7c4"].tif
- ...013-r1.nd [Stage16 "r8c4"].tif
- ...013-r1.nd [Stage17 "r1c5"].tif
- ...013-r1.nd [Stage18 "r2c5"].tif
- ...013-r1.nd [Stage19 "r3c5"].tif
- ...2013-r1.nd [Stage2 "r2c3"].tif
- ...013-r1.nd [Stage20 "r4c5"].tif
- ...013-r1.nd [Stage21 "r5c5"].tif
- ...013-r1.nd [Stage22 "r6c5"].tif
- ...013-r1.nd [Stage23 "r7c5"].tif
- ...013-r1.nd [Stage24 "r8c5"].tif
- ...013-r1.nd [Stage25 "r1c6"].tif
- ...013-r1.nd [Stage26 "r2c6"].tif
- ...013-r1.nd [Stage27 "r3c6"].tif
- ...013-r1.nd [Stage28 "r4c6"].tif
- ...013-r1.nd [Stage29 "r5c6"].tif
- ...2013-r1.nd [Stage3 "r3c3"].tif
- ...013-r1.nd [Stage30 "r6c6"].tif
- ...013-r1.nd [Stage31 "r7c6"].tif
- ...013-r1.nd [Stage32 "r8c6"].tif
- ...013-r1.nd [Stage33 "r1c7"].tif
- ...013-r1.nd [Stage34 "r2c7"].tif
- ...013-r1.nd [Stage35 "r3c7"].tif

Plate Layout (8x9 grid):

Row 1: A1 (red box), A2 (green box), A3, A4, A5, A6, A7, A8, A9

Row 2: B1, B2, B3, B4, B5, B6, B7, B8, B9

Row 3: C1, C2, C3, C4, C5, C6, C7, C8, C9

Row 4: D1, D2, D3, D4, D5, D6, D7, D8, D9

Row 5: E1, E2, E3, E4, E5, E6, E7, E8, E9

Row 6: F1, F2, F3, F4, F5, F6, F7, F8, F9

Row 7: G1, G2, G3, G4, G5, G6, G7, G8, G9

Row 8: H1, H2, H3, H4, H5, H6, H7, H8, H9

File 1 (red box): 2032013-r1.nd [Stage1 "r1c3"].tif
IMAGE ID: 6496 WELL ID: 1461

File 2 (green box): 2032013-r1.nd [Stage10 "r2c4"].tif
IMAGE ID: 6497 WELL ID: 1462

Alphabetical sorting used in Dataset_to_plate.py will result in wrong placement of images (listed on the left) on the plate (see below - images placed on plate along rows).

2. Running script with correct settings.

Settings:

- First Axis: Row (images are labeled row:column)
- Axis Count: 8 (highest row number)
- Sorting: natural (this will properly sort images 1,2,3,4,5,6,7,8,9,10,...)
- Wells to skip: (format rowA:columnA,rowB:colB, ...)
1:10,2:10,3:10,4:10,5:10,6:10,1:11,2:11,3:11,4:11,5:11,6:11,7:11
- Column offset: 2 (data starts at column 3)
- Row offset: 0

