Traject3d User Manual

Contents

1

Overview

Traject3d is an analysis pipeline, based on the KNIME data analysis platform, intended to identify, quantify and plot patterns of behaviour from 3D culture that may be occurring in parallel within a well. It relies upon a number of previous actions by a user:

- The successful 3D culture of cells and imaging of many such cultures over multiple days using phase contrast imaging. We have tested and extensively used the IncuCyte system for this purpose.
- The successful segmentation and tracking of these 3D cultures and extraction of their shape, size and movement features over time, which can be achieved using CellProfiler.
- In some instances (as described below), user-defined classification of 3D structures can be integrated into the analysis, which can be achieved by CellProfiler Analyst.

While an IncuCyte system and CellProfiler do not *a priori* have to be used to generate images of 3D culture, Traject3d requires data structure and outputs in the format that CellProfiler provides. We provide example time-lapse image sequences of 3D cultures and successful CellProfiler segmentation and analysis pipelines in the Traject3d GitHub repository. Traject3d is therefore not a system to segment images or track cells, but rather is a pipeline to analyse the data after segmentation, tracking and feature extraction from long-term time-lapse imaging to uncover heterogeneity in 3D cultures.

2

Before starting

2.1 Data required and folder structure

The first step in operating Traject3d is to assemble the data to be processed in a standardised format and directory structure. These data files are described below. An essential feature of Traject3d is that data format and directory structure must be followed exactly for proper operation. In turn, this means that when structured as described below, Traject3d provides a rapid, standardised analysis for complex data. We provide extensive documentation below to help the user.

Prior to analysis using this KNIME pipeline, you will need to have the following files:

- *Data:* Your data as a CSV file named “PhaseGrayCysts”, in which each column is either metadata or a measured morphological feature, and each row corresponds to a single object, from a single image. This file is generated by CellProfiler with this name and format when using the provided CellProfiler pipelines and automatically added to the Data subdirectory.
- *Experiment Key:* A custom Microsoft Excel spreadsheet, that we provide as a template, containing relevant metadata including, sample and experiment names, and well IDs. This template is provided in the Traject3d GitHub repository. In the example spreadsheet provided ensure the following columns are completed in the “Experiment Key” sheet (tab); “Metadata_ExperimentName”, “Metadata_Plate”, “Metadata_Well”, “Condition”. The remaining columns in this sheet will be auto-populated using this information. The other sheets in the Excel workbook will also be automatically filled in. The contents of “Metadata_ExperimentName” should be unique to each experimental/biological replicate, and “Condition” naming (e.g. cell line and/or manipulation) should be consistent (in terms of case, spelling, symbols or spaces used) between, each experimental/biological and technical replicate.
- *Phase:* The original phase images from time-lapse imaging, in RGB (3-channel) TIFF format. Filenames should be in the following format to embed metadata in the filename: ExperimentName_Plate_Well_Site_Date_Time. Please see the sample data provided for an example. This format ensures that such metadata can be extracted from the filename by CellProfiler, and that this can be matched to the user description in the Experiment Key above. Image and filename formatting is typically set at the point of export from the imaging system utilised. This is automatically outputted in this format from the IncuCyte system, but files could also be renamed from other imaging systems in this format.
- *PhaseGrayCystOutlines:* Binary (white on black background) images of segmented object outlines, as PNGs. These will be pseudocoloured by state classification during analysis and overlaid onto the phase images for visual inspection. There should be one corresponding image containing object outlines for each phase image that the outlines are derived from in the “Phase” folder. The outline images should each have the same file name as the corresponding phase image, but are stored in the PhaseGrayCystOutlines subdirectory. These outline images are generated by CellProfiler in the provided accompanying pipelines and automatically added to the PhaseGrayCystOutlines subdirectory.

The computer you will be running the KNIME analysis from should have either access to, or local copy of, the directory containing your phase images, and output from CellProfiler. We have run analysis both on local copy and across network drives successfully. The latter will depend on the speed and stability of network connection.

An Experiment Key and directory structure template (SampleData) can be found in the Traject3d GitHub repository. We recommend browsing these files to familiarise yourself with the data. Copy exactly to your appropriate location the file structure described below from the GitHub examples. For reference, this directory should have the following structure:

1. DatasetName
   - Experiment___1
     - Data
     - Experiment Key
     - Phase
     - PhaseGrayCystOutlines
   - Experiment___n
   - Output

Experimental/biological replicate folders are named "Experiment___n", where *n* corresponds to the replicate number (please note double underscore).

The default output directory for the KNIME analysis will be the “Output” subdirectory of this folder.

2.2 KNIME installation

Traject3d uses the KNIME data analytics platform. We have used KNIME v4.0.2, which can be downloaded from here. Newer versions of the KNIME software have not been tested by us with the Traject3d pipeline. There will be three steps in setting up KNIME to enable you to run the provided pipeline: installing KNIME and its functionality extensions (1.2.1), followed by setting up integrations of R (1.3), and Python (1.4) with KNIME. Once these are set up, you are ready to run your analysis. Although the instructions provided below allow for installation of the required analysis tools without needing to know how to code, please ask your bioinformatics or IT support teams if you are unclear on the instructions.

2.2.1 Notes for KNIME workflow

The KNIME software base installation doesn’t come with all possible functionalities. As such, additional functionality required for Traject3d needs to be activated by the user through installation of extensions to the base software. The KNIME extensions used by the Traject3d pipeline are listed below. These can be installed from within the KNIME software, by following *File → Install KNIME Extensions...* in the toolbar at the top of the user interface.

Required KNIME version and extensions:

Name	Version
KNIME Analytics Platform	4.0.2.v201909300912
KNIME Core	4.0.2.v201909300912
KNIME Data Generation	4.0.0.v201905311239
KNIME Distance Matrix	4.0.2.v201909260824
KNIME Excel Support	4.0.1.v201908131226
KNIME File Handling Nodes	4.0.1.v201908131226
KNIME HCS Tools	4.0.0.v201906200802
KNIME Image Processing	1.8.0.201911140609
KNIME Interactive R Statistics Integration	4.0.1.v201908131226
KNIME Math Expression (JEP)	4.0.2.v201909242005
KNIME Python Integration	4.0.0.v201906241606
KNIME Quick Forms	4.0.2.v201909242005
KNIME SVG Support	4.0.1.v201908131226
KNIME Virtual Nodes	4.0.0.v201905311239
Vernalis KNIME Nodes	1.27.0.qualifier

2.3 R Integration

The Traject3d KNIME workflow uses an R integration for some steps of the analysis. The workflow was built using R version 4.2.0, which is available here: Windows, macOS.

2.3.1 Windows

First install Rtools by following the instructions in sections 2b-c of this guide. Rtools is software that enables building of R packages from source code. During installation of Rtools the installer might not present you with the tick box to automatically "Add Rtools to system PATH" as shown in section 2b of the linked guide. If this is the case, after installation of Rtools, proceed to section 2c and manually add the path to Rtools to your environment variables; this path will have been indicated to you during the installation process. Additionally, as described in section 2c of the guide, add the path to the R bin folder if it isn't listed. The path to R installation on your system can be found from within the R application using the following command:

```
normalizePath(R.home())
```

The path to the bin folder is created by appending "\bin" to this path. Note that the paths required in the system PATH are separated using "\", not "\\" as returned by R.

Finally, install Rserve v1.8-10 by opening the R application and running the following line of code:

```
install.packages("Rserve", repos = "https://cran.r-project.org", type="win.binary")
```

When installing R packages as a non-admin user you may not have the user privileges to install packages in the default R library. If this is the case, R will suggest an alternative location in which to install the packages. Accepting this is fine, and does not affect access to the packages from the R integration in KNIME. You can check the location of this local (to your user account) library using the command:

```
.libPaths()
```

2.3.2 macOS

Instructions for setting up the R integration can be found here.

2.3.3 All users

To finish setting up the integration you next must define where KNIME can find the R installation on your system. In the toolbar (top of window) of KNIME Analytics Platform user interface, go to *File* → *Preferences*. From the list on the left, under "KNIME", select "R". In "Path to R Home" enter the path indicating the location where R is installed on your system. This path can be found from within the R application by typing:

```
normalizePath(R.home())
```

Again, note that the required path is separated using "\", not "\\" as returned by R.

Set the "Rserve receiving buffer size limit" to 0.

The Traject3d workflow also utilises a number of packages, which are not included in the base R installation. When Traject3d R integrations are run for the first time, the R scripts within the KNIME workflow should automatically download and install any missing R packages that are required for the analysis. A dialogue box may come up asking you to select where the packages should be installed from (a CRAN mirror). If so, select whichever location is geographically closest to you.

If, for any reason, the automatic download and installation is unsuccessful, any missing packages can be installed manually from within the R application (externally to KNIME). With the exception of two packages in R (cytofkit2 and Rserve; the latter of which will have already been installed as described above), all other packages can be installed manually within the R application (note: not in KNIME) by running the following command in R:

```
install.packages(PACKAGE_NAME)
```

On the other hand, manual installation of cytofkit2 requires running the following code (again, within the R application):

```
if(!require(devtools)) install.packages('devtools')
devtools::install_github("JinmiaoChenLab/cytofkit2", dependencies=TRUE)
```

Traject3d utilises the following packages:

Package	Version
ClassDiscovery	3.4.0
circlize	0.4.14
cytofkit2	0.99.80
DescTools	0.99.44
factoextra	1.0.7
FactoMineR	2.4
fields	13.3
ggdendro	0.1.23
ggimage	0.3.1
ggnewscale	0.4.7
ggplot2	3.3.6
ggseqlogo	0.1
imputeTS	3.2
MASS	7.3.51.4
pheatmap	1.0.12
plyr	1.8.7
png	0.1-7
RColorBrewer	1.1-3
reshape2	1.4.4
Rserve	1.8-10
Rtsne	0.16
vcd	1.4-9
viridis	0.6.2

2.4 Python Integration

The KNIME workflow requires Python in order to run and algorithm for subsampling data, called GeoSketch.

- First, install Anaconda, which can be found here.
- A preconfigured Python 3 environment (“py3_knime.yaml”) is provided on the Traject3d Github repository within the “KNIME” directory. Download this environment.
- Import the downloaded environment into Anaconda using the Anaconda Navigator application (installed as part of the earlier step). To do this, navigate to the “Environments” tab on the left hand side of the Anaconda Navigator user interface. At the bottom of the displayed environments list, select Import. This will bring up an Import Environment dialog box. Type a descriptive name for the environment you will be creating; we recommend using the existing environment name (“py3_knime”). Select the folder icon to navigate to and choose the previously downloaded environment (“py3_knime.yaml”). Select “Import”. Your newly imported environment will appear in the Environments list.
- Finally, follow the instructions under “Option 1” here to select the Python 3 environment (“py3_knime.yaml”) within your KNIME Python preferences.

Package	Version
GeoSketch	1.0

3

KNIME

3.1 Using KNIME

KNIME Analytics Platform is free, open-source software that can be used to design modular data science workflows using a graphical user interface, without any need for coding. It also allows integration with other tools such as R and Python in a single workflow. We recommend familiarising yourself with KNIME using the helpful introductory documentation, which can be found here.

Our workflow for Traject3d (Traject3d_v1.knwf) can be found on the Traject3d Github repository. In brief, to load the workflow into the KNIME Analytics Platform: in the main toolbar at the top of the KNIME user interface select *File → Import KNIME Workflow* and browse to select the workflow file from wherever in the local file directory it was downloaded to. Click "Finish" to import the selection. The Traject3d_v1 workflow should then open in the Local Workspace, viewed in the KNIME Explorer pane on the left of the user interface. Double-click on the workflow (Traject3d_v1) within this pane to open it in the Workflow Editor.

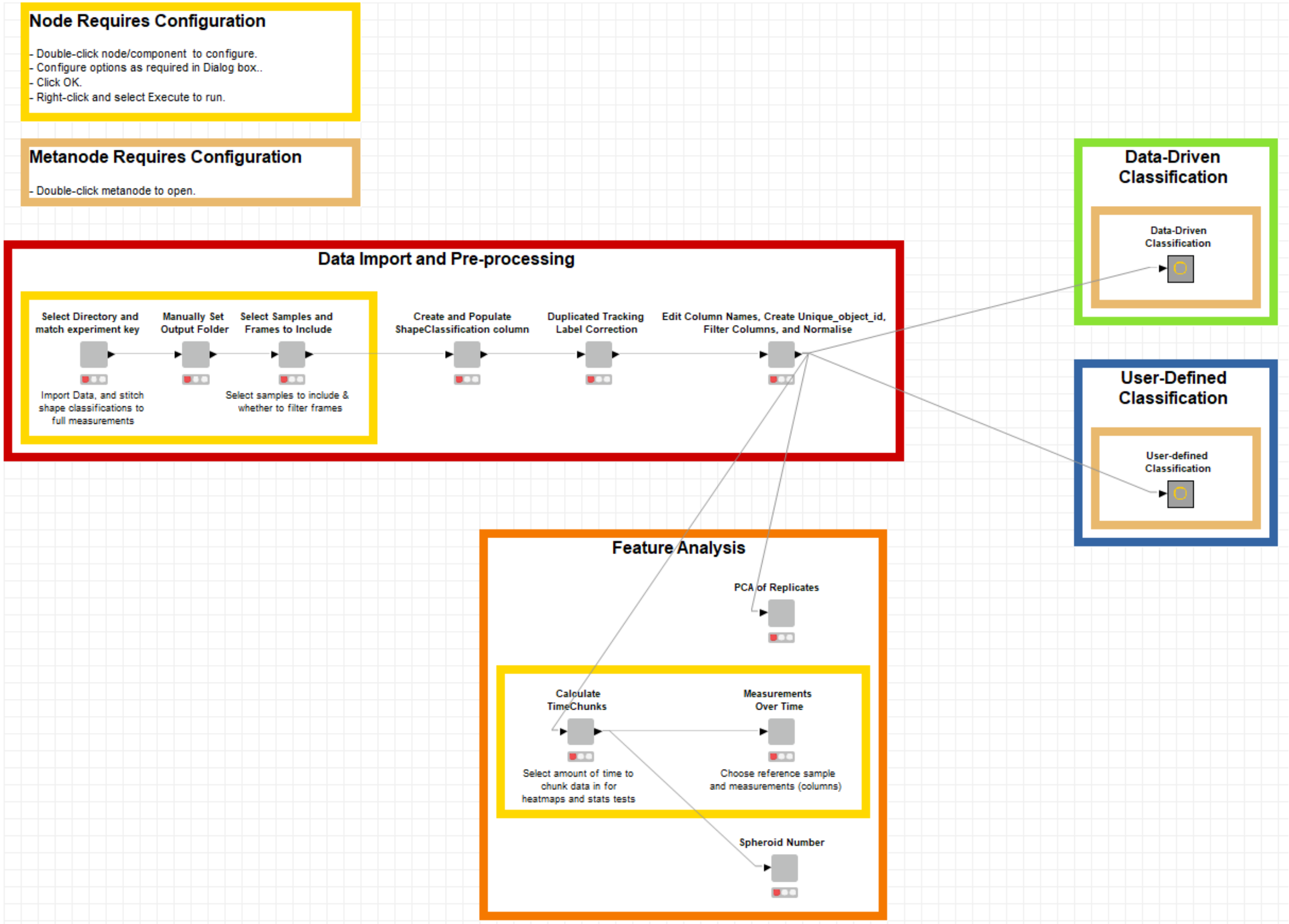
A KNIME workflow is comprised of three types of building block (defined below): nodes, metanodes, and components. The most basic of these are nodes, each of which represents a function being performed on the data. For simplicity, we have encapsulated nodes into components. These act as a wrapper, breaking the workflow into logical steps in the analysis, and condensing any user input required by the respective section into a dialog box. For example, each grey element within the "Data Import and Pre-processing" (red) section in the image below is a component. We have then used metanodes as an additional wrapper, enabling sections of the workflow (components) to be collapsed and hidden until required by the user. For example, two metanodes in the workflow are "User-Defined Classification" and "Data-Driven Classification", shown in blue and green, respectively, in the image below.

Tips for those new to KNIME:

- Double-click on a node/metanode/component. If the element is a node/component, a dialog box will open for configuration. Configure as appropriate and click OK to apply the changes. If no configuration is required, the dialog box will be empty. Conversely, if the element is a metanode, double-clicking will cause it to open revealing the nodes/components within.
- Dialog boxes for configuration of components include tick boxes labelled "Change". These toggle whether edits can be made to the corresponding field by the user, but do not affect whether a function will be performed. Tick this box to alter an input field.
- After configuration, right-click on a node/component and select "Execute" to run it.
- To open a component, right-click on it and select *Component → Open*. This will reveal the contents (nodes) of the component within a new tab in the KNIME user interface.
- Nodes that fail to run will be indicated by a red X to aid troubleshooting.
- Error messages are printed in the Console panel in the bottom right of the KNIME user interface to aid troubleshooting.
- The panel on the right-hand side of the KNIME user interface provides additional information about a selected node/metanode/component.
- After a node/component has run successfully, right-click and select the bottom-most option (usually called "Port 1" for components) to view the processed data at this stage. This can aid visual inspection for quality

control of data as it is processed, as well as for troubleshooting if there are any problems with the analysis.

3.2 Workflow Structure



Node Requires Configuration

- Double-click node/component to configure.
- Configure options as required in Dialog box..
- Click OK.
- Right-click and select Execute to run.

Metanode Requires Configuration

- Double-click metanode to open.

Data Import and Pre-processing
Select Directory and match experiment key
Manually Set Output Folder
Select Samples and Frames to Include
Create and Populate ShapeClassification column
Duplicated Tracking Label Correction
Edit Column Names, Create Unique_object_id, Filter Columns, and Normalise
Import Data, and stitch shape classifications to full measurements
Select samples to include & whether to filter frames
Data-Driven Classification
Data-Driven Classification
User-Defined Classification
User-defined Classification
Feature Analysis
PCA of Replicates
Calculate TimeChunks
Measurements Over Time
Select amount of time to chunk data in for heatmaps and stats tests
Choose reference sample and measurements (columns)
Spheroid Number

The KNIME workflow is comprised of four parts, each addressed in one section of this manual:

Data Import and Pre-processing: Essential for all subsequent parts of the workflow. This part allows import of data and merging with experimental keys, filtering to exclude any time intervals or treatment groups if required, parsing of user-defined classifications, correction of duplicated tracking labels, and data normalisation.

Feature Analysis: Generates heatmaps of size, shape and movement features over time with statistical comparison between control and treatment groups. Can also generate counts of objects analysed and perform PCA of experimental/biological and technical replicates. The outputs from this section can be useful to gain a general sense of how average features of interest across a sample (such as size, shape and movement) change over time. This section is not required for subsequent steps.

User-Defined Classification: Generates heatmaps of user-defined states over time with statistical comparison between control and treatment groups. Can also calculate means of all measured features and generate representative outlines for each state. Phase images are also overlaid with outlines colour coded by user-defined classification. Dimensionality reduction using t-SNE can be performed for data visualisation.

Data-Driven Classification: Dataset may be subsampled to identify data-driven states, from which trajectories are subsequently identified. Both are quantified as heatmaps. Trajectory motifs and transition plots are created to summarise trajectories. Representative object outlines are identified to represent each data-driven state classification, and representative tracked objects are identified to represent each trajectory. Phase images overlaid with outlines colour coded by state classification are also generated.

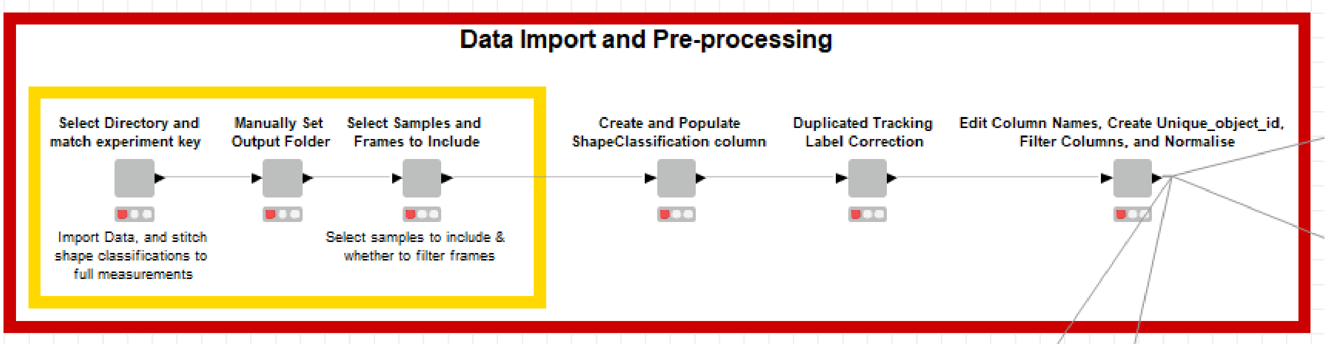
NOTE: We have indicated in each section of this user manual and, where necessary, on the pipeline itself (yellow boxes) which nodes/components require configuration by the user. This usually means that the user has to make a decision, through the selection of options in dialog boxes to provide input.

3.3 Demo Videos

We have recorded video demonstrations showing sample data being loaded into and processed using the Traject3d KNIME pipeline. This can be found in the Traject3d Github repository in *KNIME* → *Demo Videos*. Note that the timings of the video are not representative; to reduce computation time in these videos we reduced subsampling depths compared to those we would normally use, and the videos were subsequently trimmed to reduce wait time.

4

Data Import and Pre-processing


Data Import and Pre-processing
Select Directory and match experiment key
Manually Set Output Folder
Select Samples and Frames to Include
Create and Populate ShapeClassification column
Duplicated Tracking Label Correction
Edit Column Names, Create Unique_object_id, Filter Columns, and Normalise
Import Data, and stitch shape classifications to full measurements
Select samples to include & whether to filter frames

4.1 Select Directory and match experiment key

Configuration required.

This component reads in the CSV files output by CellProfiler and combines them with their respective experiment keys.

Double-click to configure the node and select Browse to set the root directory containing all your experimental/biological replicate folders in the format described in Section 1.1 “Data required and folder structure”.

We have also included functionality wherein if there is something that needs to be changed in your input files, for example a misspelling in one experiment name but not another, this can be completed within this node. In most cases, if data input has been followed exactly, this should not be necessary. This only happens when the following input fields are populated. Otherwise, this will not occur. To do so, type your search term in the ‘Input Search String’ box and your replacement term in the ‘Input Replacement strings’ box. Next select which column you wish to apply this find-and-replace function to. If you are including machine learning classification, select ‘Select Column for String Replace’. If no machine learning, select ‘Select Column to Change Strings (If no machine learning)’. Once all required options are selected, right-click and select “Execute” to run the component.

4.2 Manually Set Output Folder

Configuration required.

This is where you select the directory to which your output will be written.

Traject3d will provide a default output directory (“Output” subdirectory of your root directory). This can be altered if required.

First, double-click to configure the component. If you wish to override the default output directory, select “Yes”, and “Browse” to choose an alternative directory for output. Otherwise, select “No” and this will use the default output directory. Apply changes and close the window. Right-click the component and select “Execute” to run.

4.3 Select Samples and Frames to Include

Configuration required.

The user may wish to define the time period of imaging from a time-lapse series. This can include only select samples to be analysed or only data from a certain period in the time-series.

For instance, if different experimental replicates were imaged for different lengths of time (e.g. Experiment 1 was imaged for 95 hours, but Experiment 2 was imaged for 96 hours). Analysis for 1-95 hours should be set across both experiments to ensure appropriate comparisons. This component applies filtering for desired sample(s), image frame(s), and the lifetime of objects in the dataset.

Double-click to configure the component.

To filter samples: Samples to be retained for analysis should be in the green “Include” box, while all others in the red “Exclude” box. This can be used to remove unwanted samples or to select certain sample(s) for test runs of analysis.

To filter by frame number (time point): In order to include all frames present, whatever their imaging length, set “Filter by timepoints” to “No”. Otherwise, set it to “Yes”, and set the “Start Frame Number” and “Stop Frame Number” values to the lower- and upper-bound frames of interest, respectively. As described above, this can ensure that despite potential differences in the imaging length of experiments, only imaging intervals that are present in all datasets are analysed.

To filter by lifetime: Set “Filter objects by lifetime” to “Yes”, and set the maximum and minimum lifetime values, where lifetime is the number of frames an object was tracked for by CellProfiler. This is used to only allow objects tracked over the selected time interval to be included.

Right-click and select “Execute” to run the component.

4.4 Create and Populate ShapeClassification column

No configuration required. Collects shape classifications, when performed, into a singular column. Regardless of whether shape classification was performed in CellProfiler, right-click and select “Execute”.

If user-defined shape classifications are present, this component parses the classification columns from CellProfiler into a new “ShapeClassification” column. The new column is populated with the classification names as defined in CellProfiler. Note, these classifications would be included in the ‘PhaseGrayCysts.csv’ file in the data folder if this function was performed in CellProfiler. Otherwise, a lack of these columns is ignored.

4.5 Duplicated Tracking Label Correction

No configuration required. Just right-click and select “Execute”.

In CellProfiler, when a tracked object splits into two, or more, new objects, the software assigns the tracking label of the parent to the daughter objects. This creates conflicts downstream in the KNIME workflow, whereby multiple objects have the same tracking ID. This component ensures that the tracked objects in each Experiment, Plate, Well, and Site combination have unique tracking labels. Where duplicate tracking labels are present, the object that is largest at the initiation of the duplication retains this label and the smaller object(s) is assigned a new tracking label. For more information on this please refer to the Traject3d manuscript.

4.6 Edit Column Names, Create Unique_object_id, Filter Columns, and Normalise

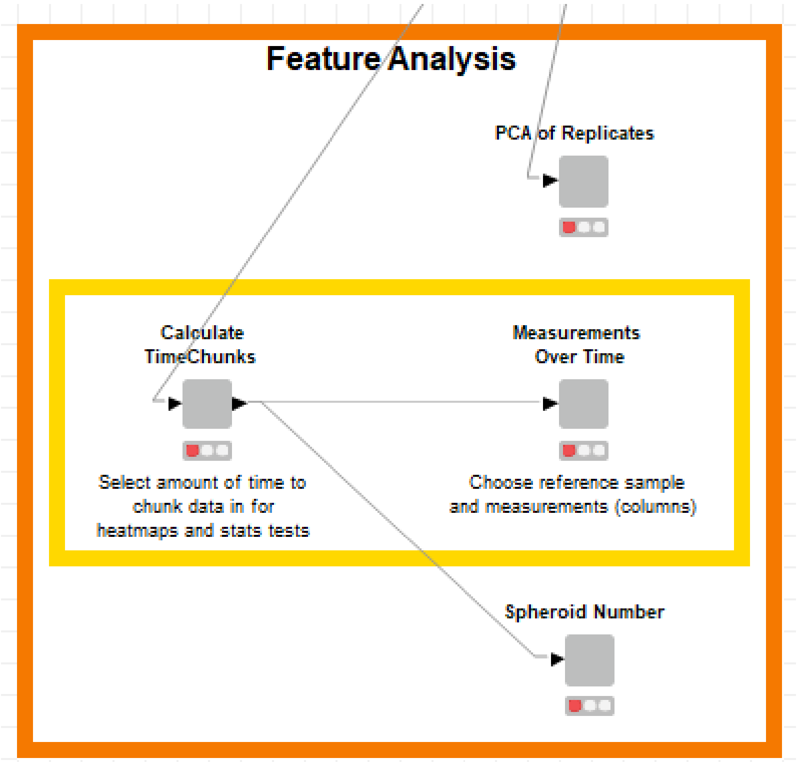
No configuration required. Just right-click and select “Execute”.

This component removes unnecessary prefixes from column names. The AspectRatio is then calculated. Unique IDs are created for each tracked object. Finally, the filename of the originating image is extracted and measurement columns are Z-score normalised.

5

Feature Analysis

The analysis from this section can be useful to gain a general sense of how average features of interest across a sample (such as size, shape and movement) change over time. This section does not require for user-defined classification to have been performed in CellProfiler, and the output from this section is not required for analysis of user-defined or data-driven classifications using Traject3d.


Feature Analysis
PCA of Replicates
Calculate TimeChunks
Measurements Over Time
Select amount of time to chunk data in for heatmaps and stats tests
Choose reference sample and measurements (columns)
Spheroid Number

5.1 PCA of Replicates

No configuration required. Just right-click and select "Execute".

This component will generate a plot to examine differences between samples, displayed as either the average effect of all parameters across experimental or technical replicates. This can be used to gain a broad overview of similarity between experimental and technical replicates across each sample.

This component performs Principal Component Analysis (PCA) on the dataset replicates, and outputs results for two PCAs: one in which all wells containing a sample are displayed as individual points in the plot ("PCAofReplicates_perWell.pdf"), and another in which each sample is a singular point from each exper-

iment (“PCAofReplicates_perExperiment.pdf”). A plot of PCA Loadings is also output for each analysis: “PCAofReplicates_perWell_Loadings.pdf”, “PCAofReplicates_perExperiment_Loadings.pdf”.

5.2 Calculate TimeChunks

Configuration required.

Visualisation of the change in a variable over time can be complex to display when many timepoints are present. This for example can occur when images are taken hourly over many days, such as 4 days/96 hours. To simplify this, data can be presented in grouped segments of time, rather than at individual timepoints. For example, behaviours that occur in 12 hour intervals can be grouped, resulting in 8 data points from 96 hours of imaging. This component allows definition of potential time interval chunking of the input data table.

Double-click to open dialog box. Configure the component to indicate how many timepoints (frames) to include in each timechunk. If chunking is not required, set this value to 1 (indicating one frame = one interval). Right-click and select “Execute” to run the component.

5.3 Measurements Over Time

Configuration required.

This component generates a heatmap showing change in measurements of area, shape, and movement over time for a set of user-specified measurements. For each sample, the average value of the measurement is calculated at each of the previously defined timechunk intervals. For the purposes of presentation, resulting values are Z-score normalised per measurement. T-tests are performed to compare samples at each time interval to the specified reference sample. A Bonferroni adjustment is applied to adjust for multiple testing. Two files are generated in the main Output directory: “MeasurementsOverTime_CONTROLSAMPLEControl.pdf”, “MeasurementsOverTime_CONTROLSAMPLEControl.csv”

Double-click to open dialog box. Select “Reference Sample” from the drop-down menu to set a reference sample for statistical comparison. Measurements to be plotted in the heatmap should be included in the green “Include” box - all others should remain in the red “Exclude” box. Descriptions of each of the measurements generated by CellProfiler can be found in the CellProfiler documentation. Selecting multiple measurements will generate one heatmap with all values on the same colour scale. Selecting one measurement at a time will result in maximum and minimum values set for each heatmap independently.

Finally, indicate how the rows (samples) in the heatmap should be ordered: alphabetically or in user-specified ordering. If a user-specified order is to be applied, use the text box to define this order (from top row of the heatmap, to bottom). Sample names should be written as they appear in the Experiment Key, each entry separated by a comma. The easiest way to ensure this is done correctly is by using keyboard shortcuts (control/command + c, control/command + v) to copy and paste sample names directly from the Experiment Key.

Tip: Be aware of any leading or trailing spaces in sample names in the Experiment Key, and if present, ensure not to delete them when configuring this text box.

5.4 Spheroid Number

No configuration required. Just right-click and select “Execute”.

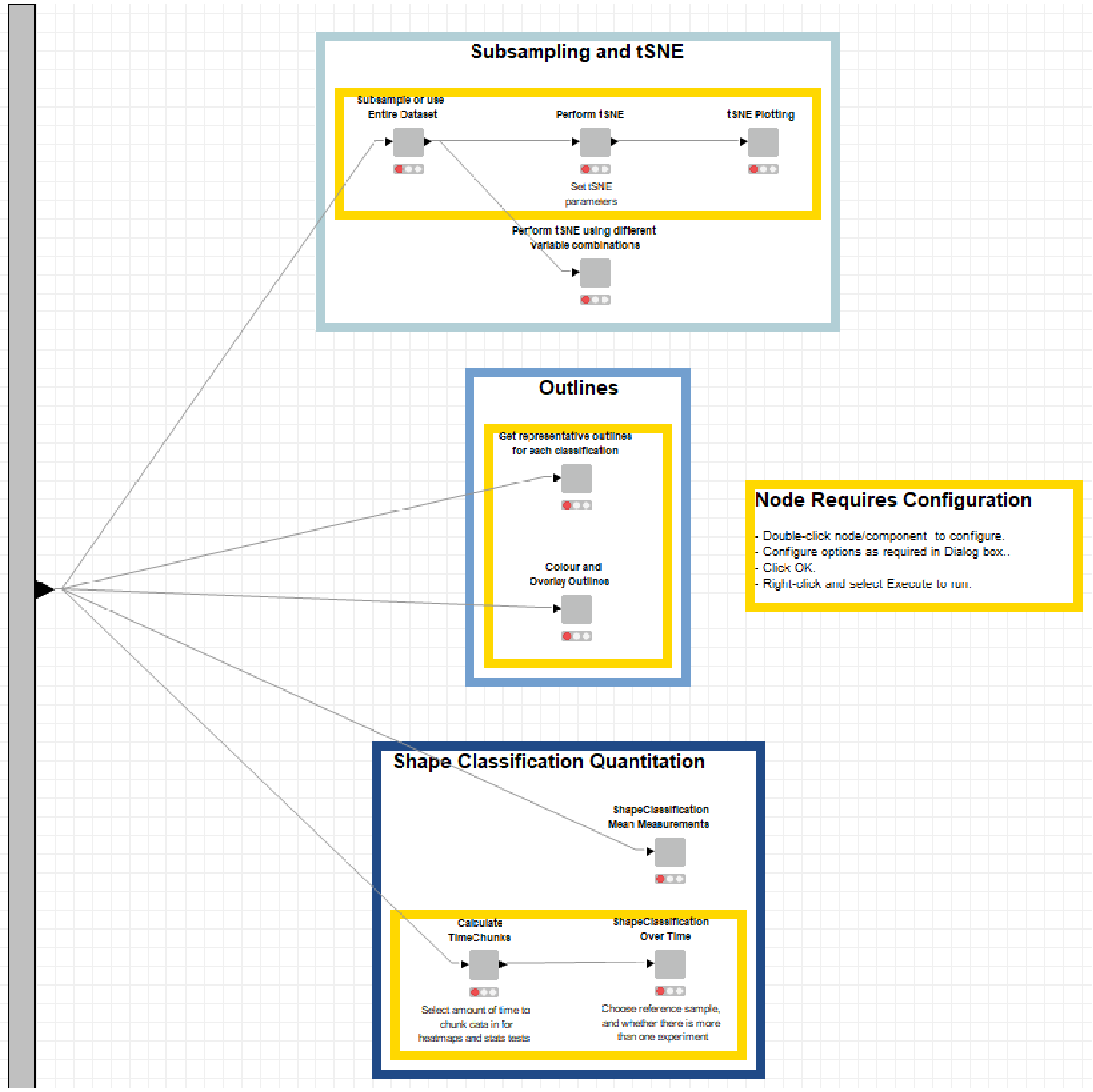
This component outputs the initial counts of spheroids in each treatment/sample per experiment (“InitialSpheroidNumbersPerExperiment.csv”), and the relative change from this initial count over time (“SpheroidNumberOverTime.csv” and “SpheroidNumberOverTime_LinePlot.pdf”).

This can be helpful to observe whether initial seeding of 3D cultures is comparable. It can also help to identify whether unusual data patterns may result from low or high numbers of spheroids, or an unexpected change over time.

6

User-defined Classification

The analysis from this section can be useful for visualisation of user-defined state classification, and how their frequency may change over time. This section requires user-defined classification to have been performed in CellProfiler. However, the output from this section is not required for analysis of data-driven classifications using Traject3d.


Subsampling and tSNE
Subsample or use Entire Dataset
Perform tSNE
tSNE Plotting
Set tSNE parameters
Perform tSNE using different variable combinations
Outlines
Get representative outlines for each classification
Colour and Overlay Outlines
Node Requires Configuration
Shape Classification Quantitation
ShapeClassification Mean Measurements
Calculate TimeChunks
ShapeClassification Over Time
Select amount of time to chunk data in for heatmaps and stats tests
Choose reference sample, and whether there is more than one experiment

- Double-click node/component to configure.
- Configure options as required in Dialog box..
- Click OK.
- Right-click and select Execute to run.

6.1 Subsample or use entire dataset

Configuration required.

Selects the appropriate fraction of the data to use to generate tSNE plots to display variation within each parameter.

Subsampling may be used to lower computation time by selecting a representative fraction of the data for use in subsequent steps. This component first ensures equal sample size. Then, if the data is to be subsampled, this is done to user specifications. Otherwise, all the remaining data is retained. Note: this section only subsamples the data for the nodes and plots that are connected directly after this step. Where other subsampling is required, such as for data-driven analyses, subsampling will be done separately at that point.

Double-click to open dialog box. Configure to indicate whether data is to be subsampled from the SubsamplingChoice dropdown menu: Subsample is default choice, Whole dataset is alternate option. If subsampling, select which method to use (SubsamplingTypeChoice dropdown menu: Random vs GeoSketch [Geometric Sketching]; note default is GeoSketch), and input the number of objects to subsample in the SubsamplingDepthChoice field.

To view the total number of objects and inform on how many objects to subsample, right-click on the previous component in the workflow (last node in Data Import and Pre-processing section, 'Edit Column Names, Create Unique…) and select "Port 1"; the total number of objects is the "Rows" shown at the top of the tab in the resulting window, and the subsampling choice should be lower than this value. After configuration, right-click the component and select "Execute" to run. As a starting point, although values need to be determined by the user, we suggest 10-20% of your total object number.

6.2 Perform tSNE

Configuration required.

If you will also be running the data-driven analysis, skip the following tSNE sections and instead run the equivalent components in section 7. This will ensure matching tSNE plots of your user-defined classes and the identified data-driven states.

Performs tSNE using Area, Zernike features, Displacement, and Distance Travelled.

Double-click to open dialog box. Configure the component to set values for perplexity, theta, the number of iterations to perform, and to indicate whether PCA should be performed first. If unsure of what values to set for these parameters, run the component called "Perform tSNE using different variable combinations" in section 6.4, and use the resulting plots to inform variable selection. Right-click and select "Execute" to run the component.

Optional Customisation: We use Area, Zernike features, Displacement and DistanceTraveled for analysis (see Traject3d manuscript Results section for rationale). If you would like to change the set of features used for analysis, right-click on this component and select Component → Open. Customise the selection of features (green box) within the "Column Filter" node highlighted in yellow.

6.3 tSNE Plotting

Configuration required.

After ensuring equal sample size and subsampling as required by the user, this component produces tSNE plots of the input data table. Plots are produced in which points are: coloured by sample, both in individual plots and all together; coloured by their value for each measurement; for each sample, coloured by point density; and coloured by user-defined classification. Plots are saved to Output → tSNE, within a folder whose name summarises the tSNE parameters as set by the user.

Double-click to open dialog box. Configure component to indicate whether to subsample the data further prior to plotting. This value should be lower than the number of objects used for tSNE analysis. This can be used to adjust the number of points to ensure optimal plotting (i.e. to make sure that the plot is neither too sparse nor too dense). Additionally, select a colour scheme, and point size to be used for plotting. Right-click and select "Execute" to run the component.

6.4 Perform tSNE using different variable combinations

No configuration required. Right-click and select "Execute" to run the component.

tSNE is performed (on Area, Zernike features, Displacement, and DistanceTravelled) using combinations of different values for all parameters. Plots for these are coloured by user-defined classification and saved within Output → tSNE. These are intended to aid in selecting the appropriate values to use for best possible visualisation.

perplexity: 25, 50, 100

theta: 0.25, 0.5

iterations: 5000

PCA: TRUE

Optional Customisation: We use Area, Zernike features, Displacement and DistanceTraveled for analysis (see manuscript Results section for rationale). If you would like to change the set of features used for analysis, right-click on this component and select Component → Open. Customise the selection of features (green box) within the "Column Filter" node highlighted in yellow, making sure to include the "ShapeClassification" column.

6.5 Get representative outlines for each classification

Configuration required.

Traject3d identifies different behaviour subtypes in the data. To enable intuitive understanding of what these different subtypes represent in terms of spheroid morphology, we provide the most representative outline for each sub-type. Note that this is optional to run.

This component selects and outputs a representative outline for each user-defined classification. This is selected from outlines of x (where x = user-selected number) objects nearest (in terms of Euclidian distance) to the mean measurements of the group. Outlines are saved in the ShapeClassification → RepresentativeOutlines subdirectory. Plots and the selected representative object outline are subsequently saved in ShapeClassification → RepresentativeOutlines → Plots subdirectory.

Double-click to open dialog box. Configure to indicate the number of outlines from which to select a representative per classification group. Right-click and select "Execute" to run the component. A larger selection may result in refined example, but the trade-off is increased computation time. We recommend starting with 30 outlines (the default) and adjusting as needed.

6.6 Colour and Overlay Outlines

Configuration required.

This component overlays object outlines, coloured by their user-defined classification, onto the original phase images. The resulting images are saved in Output → ShapeClassification → OutlineOnPhase. Note that this is an optional function to run.

Double-click to open dialog box. Configure this component to select an experiment, plate, and site (imaging position within a well is multiple sites were imaged) for which to overlay outlines. Next indicate whether this is to be performed for all wells or a user-defined subset. If the latter, populate the associated text input with the wells of interest, separated by commas but no spaces. Finally, indicate whether to process frames for the duration of the whole experiment, or for a user-defined subset of timepoints/frames. If a subset is to be used, fill in the appropriate text input to indicate the frames of interest, again separated by commas but no spaces.

In the Traject3d manuscript, we use user-defined classes of 'Round', 'Spread' and 'Spindle'. If you are not using these classes you will need to adjust this component to assign colours to your specific classifications. First, open the component (right-click, select Component → Open). There will be a number of nodes (don't be scared!). Navigate to the node highlighted with a yellow box called "Color Manager". Double-click to configure the "Color Manager" node, and set the colours to be used for each classification; note the colours that are used for each shape classification in the tSNE plots etc, if you want these to match.

Right-click and select "Execute" to run the component.

6.7 ShapeClassification Mean Measurements

No configuration required. Right-click and select "Execute" to run the component.

This component generates a CSV file containing the mean values of measurements for each user-defined class. The resulting file is named "ShapeClassificationMeanMeasurements.csv" and is located in Output → ShapeClassification. The output generated by this component can be useful for getting an idea of what an average object in each user-defined classification looks like in terms of the size, shape, and movement features. This can help for understanding how more abstract aspects of object morphology contribute to an observed phenotype.

6.8 Calculate TimeChunks

Configuration required.

Visualisation of the change in a variable over time can be complex to display when many timepoints are present. To simplify this, data can be presented in grouped segments of time, rather than at individual timepoints. This component calculates time interval chunks for the input data table.

Double-click to open dialog box. Configure the component to indicate how many timepoints (frames) to include in each timechunk. If chunking is not required, set this value to 1. Right-click and select “Execute” to run the component. See section 5.2 for expanded explanation of TimeChunks.

6.9 ShapeClassification Over Time

Configuration required.

This component compares, between a reference and non-reference sample, the proportions of each user-defined classification at every timechunk interval. A heatmap and CSV file of results is output to Output → ShapeClassification, named: ShapeClassification_ShapeChangeOverTimeHeatmap_StatsTest_REFERENCE SAMPLE NAME, where StatsTest represents: Cochran-Mantel-Haenszel test (CMH), Chi-Squared test (ChiSq).

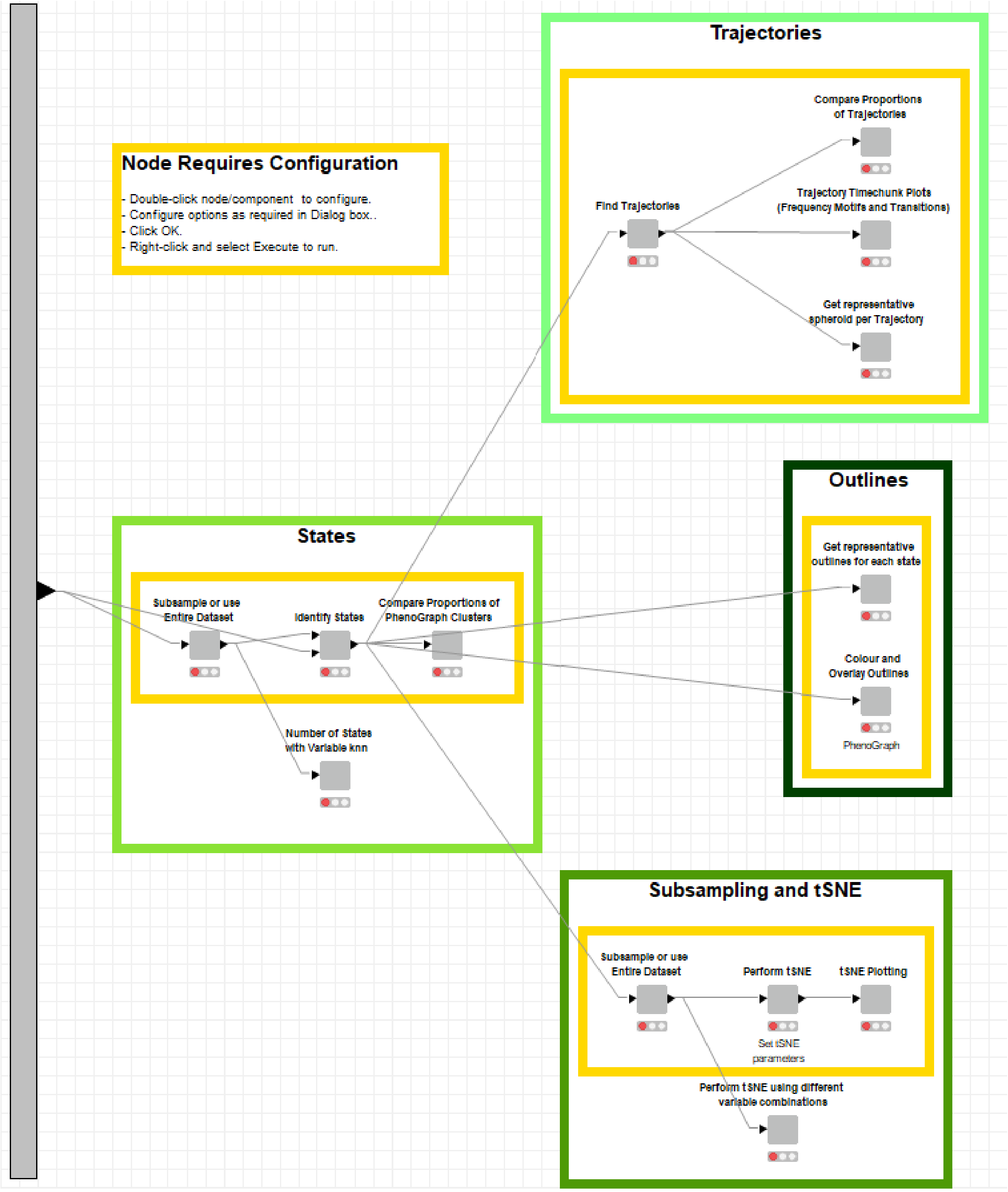
Double-click to open dialog box. Configure the component to set a reference sample for statistical comparison. Select the statistical test to be used. Generally, Cochran-Mantel-Haenszel is used when comparing across multiple experimental/biological replicates, whereas Chi-squared is used when there is only one replicate. Finally, indicate how rows (samples) in the heatmap should be ordered: alphabetically, dendrogram clustered, or by user-specified ordering. If dendrogram clustering is used, the dendrogram is output as a separate file (ShapeClassification_ShapeChangeOverTimeHeatmap_Dendrogram_REFERENCE SAMPLE NAME). If a user-specified order is to be used, use the text box to define this order (from top row of the heatmap, to bottom). Sample names should be written as they appear in the Experiment Key, separated by only commas. The easiest way to ensure this is done correctly is by using keyboard shortcuts (control/command + c, control/command + v) to copy and paste sample names directly from the Experiment Key.

Tip: Be aware of any leading or trailing spaces in sample names in the Experiment Key, and if present, ensure not to delete them when configuring this text box. Right-click and select “Execute” to run the component.

7

Data-Driven Classification

Identifies data-driven state classifications based on the size, shape, and movement features output by CellProfiler. These classifications are then analysed temporally to identify trajectories of change over time. This can be a useful approach for identification of classes *de novo*, when the user either may not know or not want to define classifications of interest. This section subsequently outputs visualisations for understanding what phenotypes the classes represent biologically, and how class frequency varies across samples.



Node Requires Configuration

- Double-click node/component to configure.
- Configure options as required in Dialog box..
- Click OK.
- Right-click and select Execute to run.

Trajectories
Compare Proportions of Trajectories
Find Trajectories
Trajectory Timechunk Plots (Frequency Motifs and Transitions)
Get representative spheroid per Trajectory
Outlines
Get representative outlines for each state
Colour and Overlay Outlines
PhenoGraph
States
Subsample or use Entire Dataset
Identify States
Compare Proportions of PhenoGraph Clusters
Number of States with Variable knn
Subsampling and tSNE
Subsample or use Entire Dataset
Perform tSNE
tSNE Plotting
Set tSNE parameters
Perform tSNE using different variable combinations

7.1 Subsample or use Entire Dataset

Configuration required.

Selects the appropriate fraction of the data to use to generate tSNE plots to display variation within each parameter.

Subsampling may be used to lower computation time by selecting a representative fraction of the data for use in subsequent steps. This component first ensures equal sample size. Then, if the data is to be subsampled, this is done to user specifications. Otherwise, all the remaining data is retained. Note: this section only subsamples the data to be used in sections 7.2-3 "Number of states with Variable knn" and "Identify States". Where other subsampling is required, such as for user-defined analyses, subsampling will be done separately at that point.

Double-click to open dialog box.

Configure to indicate whether data is to be subsampled from the SubsamplingChoice dropdown menu: Subsample is default choice, Whole dataset is alternate option. If subsampling, select which method to use (SubsamplingTypeChoice dropdown menu: Random vs GeoSketch [Geometric Sketching]; note default is GeoSketch), and input the number of objects to subsample in the SubsamplingDepthChoice box.

To view the total number of objects and inform on how many objects to subsample, right-click on the previous component in the workflow (last node in Data Import and Pre-processing section, 'Edit Column Names, Create Unique. . .) and select "Port 1"; the total number of objects is the "Rows" shown at the top of the tab in the resulting window, and the subsampling choice should be lower than this value. After configuration, right-click the component and select "Execute" to run. As a starting point, although values need to be determined by the user, we suggest 10-20% of your total object number.

7.2 Number of States with Variable knn

No configuration required. Right-click and select "Execute" to run the component.

In section 7.3, the user needs to define the value of the k-nearest neighbours parameter used by the PhenoGraph algorithm to determine data-driven states in the dataset. This component is optional but its output can be used to identify the key value for your data required to run 7.3.

The PhenoGraph clustering algorithm is performed on the Area, Zernike features, Displacement, and Distance Travelled columns of the input data table. To do this the k-nearest neighbours parameter must be set, which defines how many 'nearest neighbours' in phenotypic space must be identified for each object.

This component varies the knn value between 20-100 to help in determination of potential knn values for section 7.3. The number of identified states across knn iterations is output as a line plot (PDF) named "VaryingKNN_LinePlot" within in Output → PhenoGraph.

Optional Customisation: We use Area, Zernike features, Displacement and DistanceTraveled for analysis (see Traject3d manuscript Results section for rationale). If you would like to change the set of features used for analysis, right-click on this component and select Component → Open. Customise the selection of features (green box) within the "Column Splitter" node highlighted in yellow.

7.3 Identify States

Configuration required.

This node finds distinct states within your data.

The PhenoGraph clustering algorithm is performed on the Area, Zernike features, Displacement, and Distance Travelled columns of the input data table. When subsampling to decrease computation time, you likely took a subset of the data. The classifications are initially based on just this subset of data. The data previously excluded when ensuring equal sample size and during subsampling is then retrofitted to the identified states, using Euclidian distance (i.e. the same label as that of the nearest group is applied). This ensures that all data is classified into distinct states. Mean measurements (i.e. Area, AspectRatio) of the identified states are calculated, and output as a heatmap. The heatmap is saved as "PhenoGraph_MeanExpressionHeatmap.pdf" within Output → PhenoGraph .

Double-click to open dialog box. Configure the component to set a value for the PhenoGraph *k*-nearest neighbours parameter. If in doubt, use the plot generated by the component called "Number of States with Varying knn" (section 7.2) and select a value (x-axis) near the elbow point of the line plot. Right-click the component and select "Execute" to run.

Optional Customisation: We use Area, Zernike features, Displacement and DistanceTraveled for analysis (see Traject3d manuscript Results section for rationale). If you would like to change the set of features used for analysis, right-click on this component and select Component → Open. Customise the selection of features (green box) within the "Column Splitter" and "Similarity Search" nodes highlighted in yellow.

7.4 Compare Proportions of PhenoGraph Clusters

Configuration required.

This component will output statistical comparisons between samples.

This component compares the proportions of objects that belong to each PhenoGraph (data-driven state) classification. This is done pairwise between reference and non-reference samples. This, for example, allows statistical analysis of whether a subtype is enriched or depleted in a sample compared to the reference, such as gain or loss of a sub-population upon a drug treatment. Output files include a heatmap of the fold changes from control and statistical comparisons (PDF) and a CSV file containing the heatmap plot data, and are saved within Output → PhenoGraph as:

"PhenoGraph_TotalProportionHeatmap_StatsTest_REFERENCE SAMPLE NAME"

StatsTest represents: Cochran-Mantel-Haenszel test (CMH), Chi-Squared test (ChiSq).

Double-click to open dialog box. Configure the component to set a reference sample, and to select which statistical test to perform. Generally, Cochran-Mantel-Haenszel is used when comparing across multiple experimental/biological replicates, whereas Chi-squared is used when there is only one experimental replicate. The latter can be useful when taking a preliminary look at trends in a single experiment, while only the former is recommended for true analysis across appropriately controlled experimental replicates.

Next, indicate whether rows (states) and columns (samples) should be ordered alphabetically, clustered (by dendrogram based on Euclidian distance), or by a user-specified ordering. If a user-specified ordering is required, input the desired order into the appropriate text input, separated by commas without spaces. Sample names should be written as they appear in the Experiment Key. The easiest way to do this is by using keyboard shortcuts to copy and paste directly from the Experiment Key. If clustered ordering is selected, the associated dendrogram will be output as a separate PDF file in the location stated above. Right-click the component and select "Execute" to run.

7.5 Get representative outlines for each state

Configuration required.

Ensure this component is run prior to attempting section 7.9 "Trajectory Timechunk Plots (Frequency Motifs and Transitions)". This is because the representative outlines output by this component are utilised in the transition plots generated by section 7.9.

This component selects and outputs a representative outline for each identified state, providing a simple cartoon graphic of the shape of each subtype. This is selected from outlines of x (where x = the number selected by the user) objects nearest (in terms of Euclidian distance) to the mean measurements of the group. A larger selection may result in refined example, but the trade-off is increased computation time. We recommend starting with 30 outlines (the default) and adjusting as needed.

Outlines are saved in the PhenoGraph → RepresentativeOutlines subdirectory. Plots and the selected representative object outline are subsequently saved in PhenoGraph → RepresentativeOutlines → Plots subdirectory.

Double-click to open dialog box. Configure to indicate the number of outlines from which to select a representative per classification group. Right-click the component and select "Execute" to run.

Optional Customisation: We use Area, Zernike features, Displacement and DistanceTraveled (see Traject3d manuscript Results section for rationale) to identify data-driven states in the data. To select a representative outline for each state, all outlines are compared to the group mean for these same size, shape and movement features. If you would like to change the set of features used to compare between the group mean and each object, right-click on this component and select Component → Open. Customise the selection of features (green box) within the "Similarity Search" node highlighted in yellow, making sure to include the PhenoGraph column.

7.6 Colour and Overlay Outlines

Configuration required.

Note that this is optional to run.

This component overlays object outlines, pseudocoloured by their state classification, onto the original phase images. This allows the user to observe in the original images what behaviours are being labelled as distinct phenotypes by

Traject3d, providing a simple way to visualise the results of Traject3d. The resulting images are saved in Output → PhenoGraph → OutlineOnPhase.

Double-click to open dialog box. Configure this component to select an experiment, plate, and site (imaging position within a well is multiple sites were imaged) for which to overlay outlines. Next indicate whether this is to be performed for all wells, or a user-defined subset. If the latter, populate the associated text input with the wells of interest, separated by commas but no spaces. Finally indicate whether to process frames for the whole duration of the experiment, or for a user-defined subset of timepoints. If a subset is to be used, fill in the appropriate text input to indicate the frames of interest, again separated by commas but no spaces. Right-click and select "Execute" to run the component.

7.7 Find Trajectories

Configuration required.

This component strings the state classifications into a time sequence for each tracked object. After a series of filtering steps to remove objects that were only tracked for a short period of time, the PhenoGraph algorithm is used to identify recurring "trajectories" of state change over time. A heatmap of trajectories, as well as the heatmap data are saved in Output → PhenoGraph → Trajectory, as "TrajectoryHeatmap_knn.pdf", and "TrajectoryHeatmap_Data_knn.csv"

A CSV file containing the number/percentages of a) total values plotted in the heatmap, b) imputed values, and c) retrofitted PhenoGraph classifications are also saved ("TrajectoryHeatmap_DataSourceCounts.csv").

Line plots are output per measurement, of mean values for each identified trajectory: Output → PhenoGraph → Trajectory → MeasurementAverageOverTime

Line plots are output per trajectory, of normalised mean Area, Displacement, and DistanceTravelled: Output → PhenoGraph → Trajectory → AreaMovementPlots

Double-click to open dialog box. Configure the component to set a value for the PhenoGraph k-nearest neighbours parameter. In our datasets, a value of 40 has been a useful starting point. This may need to be adjusted according to your specific dataset. Generally, increasing this value will cause identified trajectory classes to merge, whereas decreasing the value will cause them to split; use visual inspection of "TrajectoryHeatmap_knn.pdf" to inform choice and adjust as necessary. Right-click the component and select "Execute" to run.

Note: for an in-depth description of how Trajectories are identified (including imputation and retrofitting), see the Traject3d manuscript.

7.8 Compare Proportions of Trajectories

Configuration required.

Used to quantify the proportions of objects displaying each trajectory and for statistical comparison between samples.

A key consideration in defining distinct trajectories is whether these occur repeatedly across independent experiments and whether these are changed in any of the comparisons chosen. This component compares the proportions of objects that belong to each Trajectory classification. This is done pairwise between reference and non-reference samples. Output files include a heatmap of the fold changes from control and statistical comparisons (PDF), and a CSV file containing the heatmap plot data. These are saved within Output → PhenoGraph → Trajectory as "Trajectory_TotalProportionHeatmap_StatsTest_REFERENCE SAMPLE NAME". StatsTest represents: Cochran-Mantel-Haenszel test (CMH), Chi-Squared test (ChiSq).

Double-click to open dialog box. Configure the component to set a reference sample, and to select which statistical test to perform. Generally, Cochran-Mantel-Haenszel is used when comparing across multiple experimental/biological replicates, whereas Chi-squared is used when there is only one replicate. Next, indicate whether rows (Trajectories) and columns (samples) should be ordered alphabetically, clustered (by dendrogram based on Euclidean distance), or by a user-specified ordering. If a user-specified ordering is required, input the desired order into the text input, separated by commas without spaces. Sample names should be written as they appear in the Experiment Key; the easiest way to do this is by using keyboard shortcuts (control/command + c, control/command + v) to copy and paste directly from the Experiment Key. If clustered row and/or column ordering is selected, the associated dendrograms will be output as separate PDF files in the location stated above. Right-click the component and select “Execute” to run.

7.9 Trajectory Timechunk Plots (Frequency Motifs and Transitions)

Configuration required.

Ensure you have run section 7.5 “Get representative outlines for each state” prior to attempting to run this component.

Used to plot the changes between and within trajectories over time.

Ensure the “Get representative outlines for each state” component has been completed, as these are required for the following functionality.

This component generates summary plots for each identified Trajectory. Three types of plots are generated for each Trajectory: 1) Frequency motif of states at the defined timechunk interval. 2) Chord diagram of transitions between states, segmented by the defined timechunk interval. 3) Transitions plotted as line segments onto a PCA plot of states (one plot per timechunk). These are useful for PER TRAJECTORY 1) defining which states are most common during the defined timechunk interval, 2) displaying within each TimeChunk Interval how spheroids may transition (or not) between states, and 3) for states between which transitions are occurring, how they are related in phenotypic space. The files are saved in Output → PhenoGraph → Trajectory, in subdirectories named “Frequency Motifs”, and “Transition Plots”.

Double-click to open dialog box. Configuring this component is required to indicate the timechunk interval (1 = no chunking). Larger time intervals will result in less complexity in the motif, whereas smaller chunking will result in long and complex motifs. In our data, 12-hour intervals were useful in demonstrating trends over time. We recommend starting with this (the default) and increasing or decreasing as appropriate for the rate of change in your experiments.

A transition frequency (as a percentage) filter is required for both types of transition plot. This defines the threshold for the frequency that transitions must occur at (within a timechunk interval) to appear on the plots. The user needsto define what is an acceptable biological threshold for their experiments. In our data, a minimum threshold of 2% (the default) was useful in demonstrating trends over time. We recommend starting with this and increasing or decreasing as appropriate for the rate of transitions in your experiments.

Finally, a minimum line thickness is required for generating the PCA transition plots. In these plots, the line thickness is relative to the proportion of transitions between states. As each dataset will have different occupancy of phenotypic space, line thickness can be adjusted based on the data for visualisation purposes. This helps ensure lines do not interfere with display of states. In our data, a line thickness of 1.5 (the default) was useful in demonstrating trends over time. We recommend starting with this and increasing or decreasing as appropriate for the rate of transitions in your experiments.

Right-click the component and select “Execute” to run.

7.10 Get representative spheroid per Trajectory

Configuration required.

This component selects and outputs phase images of x (where x = user-defined number) “representative” spheroids for each trajectory. Images are saved as square crops around the selected spheroid, but can also be output as uncropped files. The resulting images are saved in Output → PhenoGraph → Trajectory → RepresentativeSpheroids.

Double-click to open dialog box. Configure to indicate how many representative spheroids should be output per trajectory and whether to save uncropped phase images (in addition to the cropped images). Right-click the component and select “Execute” to run.

These output images from this component are helpful in visualising different trajectories by highlighting the spheroids in the phase image that display each trajectory. As the spheroids are centered within the cropped images it can be difficult to appreciate any movement through the well. The equivalent uncropped images can be used to observe motility of the spheroids.

Note on the algorithm: First to identify a representative spheroid(s) for each trajectory a consensus sequence is determined. This is comprised of the most frequent state at each point in time, and matches the sequence appearing in the state frequency motifs generated in section 7.9. Next, the algorithm looks for the spheroid that over time has the closest match to this consensus sequence. This is done by examining the sequence of state classifications for every spheroid in a trajectory. Each tracked spheroid over time is compared to the 'consensus sequence' in that trajectory group, using a string comparison algorithm. This results in a similarity score. Images of x (where x = user-defined number) spheroids deemed most similar, in terms of this score, to the consensus sequence are retrieved. The spheroid outlines are overlayed onto the phase image, cropped, and output.

Note on the similarity score calculations: Only mismatches at equivalent timepoints are assessed, while insertion/deletion type alterations are not taken into consideration.

8

Troubleshooting Tips

We provide here some considerations for troubleshooting.

Problem: Running out of memory or crashing when I click 'execute all'.

Solution: KNIME allows for a user to click 'Execute All', which will run all nodes in the sequence of how they are connected. Some nodes may run in parallel. This occurs with default values. However, a number of nodes require optimisation of settings for subsequent steps. We therefore recommend executing each node in the order presented in this manual, ensuring the appropriate settings have been configured for your specific dataset. The ability to run multiple nodes in parallel will be highly dependent on your computer resources (i.e. CPU, RAM), and may cause execution errors.

Problem: Cannot install KNIME extensions, Conda (Anaconda) environments, or R packages as described in section 2.

Solutions: This may relate to an internet connection issue. KNIME, Anaconda, or R may be trying to access repositories from which to download extensions/packages, but without the correct configuration to access these. If you are setting the software up using an institute/university network connection, which is otherwise stable and functioning as expected, the problem may have to do with your institute network settings (i.e. your computer requires proxy server settings). Please see your IT department to help troubleshoot this.

Problem: Data won't load into "Select Directory and match experiment key" component.

Solutions: if folders or data are not in the format descripted in Section 2.1 "Folder Structure" this component will have trouble locating the necessary files. Please check the following:

- That you are configuring this node to direct to the Directory Folder, not an "Experiment__n" folder.
- That the Directory Folder has no spaces in the name.
- Each "Experiment__n" folder has double underscore before the number and no additional spaces/characters.
- That "Phase" folder contains all images in RGB (3-channel) TIFF format with filenames in following format; ExperimentName_Plate 1_Well_Site_Date_Time.
- That "PhaseGrayCystOutlines" folder contains binary images of object outlines as PNGs. This should be one image, with equivalent file name, per corresponding Phase image in "Phase" folder.
- That "Data" folder contains a CSV file named 'PhaseGrayCysts' in correct format with all relevant fields populated.
- That all fields in 'Experiment Key' and the appropriate 'Plate' tabs are filled in or auto-populated correctly.
- That value in "Metadata_ExperimentName" is different across experimental/biological replicates in each Experiment Key.
- That "Conditions" are consistent in Experiment Keys for both experimental/biological and technical replicates.
- Data described in Experiment Key corresponds to data analysed by CellProfiler and included in "PhaseGrayCysts" CSV file. Any data not included in analysis should be removed from the Experiment Key, and subsequently excluded using the component "Select Samples and Frames to Include" (Section 3.3).
- All data analysed by CellProfiler and included in "PhaseGrayCysts" CSV file is described in Experiment Key.

Problem: in "Select Samples and Frames to include" component there are more Sample names than there should be.

Solution: This can occur if mis-naming, errors, or variations in a sample name have occurred. For instance, 'Control' in one experiment is labelled 'Ctrl' in another experiment. Check that samples names are exactly the across for both experimental and technical replicates. This is most simply corrected by editing the "Condition" column of the Experiment Key Excel file for each experiment.

Problem: R Snippet node not running in various components.

Solution: As R Snippets are used to perform various processes in the workflow, this can occur for a variety of reasons. The KNIME console should print any errors from R to help inform on what the problem is. If the error indicates an issue with Rserve, or a missing package (i.e. "there is no package called 'x' ", or "could not find function 'x' "), in the first instance ensure the correct version of R and all relevant packages were installed correctly as detailed in Section 2.3 "R Integration". Manually install any packages that are missing. For additional help, see your IT/Bioinformatics support teams for R setup.

Note: Although KNIME allows for the possibility of multiple instances of R integration to be run in parallel, it is best to execute these nodes sequentially. The ability to run these in parallel will be highly dependent on your computer's resources. Some of the functions described in this manual, specifically 6.2 "Perform tSNE", 6.4 "Perform tSNE using different variable combinations" (as well as their equivalents in section 7), 7.2 "Number of States with Variable knn" and 7.3 "Identify States" are very computationally expensive to perform, so we do not recommend running them in parallel.

Problem: Error encountered when running statistical analysis using Cochran-Mantel_Haenszel test.

Solution: This test requires two or more experimental/biological replicates for *every* sample. If you have only one experimental/biological replicate, use the Chi-squared test instead. If you are certain you have multiple replicates, check that:

- Sample names are exactly the same for both experimental/biological and technical replicates in "Condition" field in Experimental Keys.
- Each experimental/biological replicate has a unique value in the "Metadata_ExperimentName" column of its respective Experiment Key.

Problem: Overlay outlines components not working.

Solution: Check that:

- Phase images are in 3-channel (RGB) TIFF format and saved within a folder called "Phase" (in the respective experimental replicate folder).
- Outline images show white outlines on black background are PNG format and contained within a folder called "PhaseGrayCystOutlines".
- The names of the phase and corresponding outlines images (described above) match exactly.
- User-defined classification component (section 6.6): ensure that you have set outline colours for your user-defined classes.

Problem: I changed the settings in CellProfiler because I wanted a different analysis than what you describe in this manual. Now Traject3d won't run.

Solution: All of the (current) CellProfiler output from the 'MeasureObjectSizeShape' component are parsed by Traject3d. Ensure that the columns to be included in analyses are selected for in each of the relevant nodes by executing nodes one by one according to the manual. In the Traject3d paper, the pipeline provided for CellProfiler (see Traject3d Github repository) allowed analysis of multiple cell lines and organoid 3D cultures. We recommend this pipeline for all image segmentation. Unfortunately, we cannot ensure compatibility with functionality of CellProfiler that we have not explored.