Custom Module: Combine Sample Metadata

Description: Combine logical queries on sample metadata to generate a binary column with results that can be used for filtering in later steps of analysis.

Setup:

|  |  |  |  |
| --- | --- | --- | --- |
| R version | CPU cores | RAM (GB) | Max run (h) |
| Any | 16 | 128+ | 4+ |

\* RAM and Max run time listed above are suggested minimums. Values may be increased (note that increasing RAM or Max run time may increase costs). If the module exceeds either value, it will immediately fail.

Variables:

Name must be spelled exactly as shown for script to work.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type | Name | Display name | Range | Default | Required |
| STRING | columnName | Name of new metadata column | 0-300 |  | Yes |
| STRING | column | Column(s) to query on | 0-1200 |  | No |
| STRING | logic | Logic for query (ex: ‘>’, ‘<=’, ‘=’) | 0-100 |  | No |
| STRING | value | Query comparison (ex: 5, ‘B-cell’) | 0-1200 |  | No |
| STRING | condition | If combining queries, AND or OR. If one query, leave blank | 2-3 | AND | No |
| STRING | fullCondition | Advanced feature: Full query condition. No error checking on this. Overrules other variables if available. | 0-2500 |  | No |
| STRING | trueResult | Advanced feature: value for true query result | 1-1200 | 1 | Yes |
| STRING | falseResult | Advanced feature: value for false query result | 1-1200 | 0 | Yes |

Packages: None

Outputs: Generates a new column called [columnName] in the metadata containing 0/1 depending on if the row meets the query criteria.

0/1 can be updated to anything using trueResult and falseResult but only 0/1 can be used for filtering in downstream modules.

Any quotes used in AtoMx MUST be single quotes. See below for details. Module will error immediately with no legible error message!

Tips:

* All values that contain multiple values need to be within c( ). See examples below. This is the [function](https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/c) that creates vectors in R.
* All STRING values within a vector need to be surrounded by quotes. Without the quotes, R assumes these values are variable names. There are innate quotes around STRING variables from the UI so adding quotes normally will prematurely close the quote. To avoid this, you will need single quotes ‘ around the values. This allows the script to keep the quote as intended.
  + UI Format = c(‘column1’, ‘column2’)
  + Backend format = “c(‘column1’, ‘column2’)”
    - If the normal double quotes are used this is what the script would see
    - c(column1, column2) yellow = string, clear = variable name

Example Query Creation:

Get the sample metadata column names using the [***GetSampleMetadata.R***](https://github.com/Nanostring-Biostats/CosMxDACustomModules/blob/main/SampleMetadata/GetSampleMetadata.R) custom module with ***metaSummary*** selected. This will get the exact column names and summary of what the column contains.

Basic Query:

**Let’s say you are looking for all cells that are in slide 2.**

This data is found in the metadata column “slide\_ID\_numeric”

There are two ways of creating this query within this custom module.

1. Setting individual variables and the script will generate the query.

This route allows for most basic queries to be generated.

A typical query might be written like this: slide\_ID\_numeric = 2

Yellow (slide\_ID\_numeric): column Blue (=): logic Pink (2): value

So for our example query our values will be:

**columnName**: newCol

**column**: slide\_ID\_numeric

**logic**: =

**value**: 2

**condition**: Can be left blank

1. Advanced Option: Provide the full query.

This is the advanced option and allows for more refined queries. But there is little to no oversight on this. There is no spell check on column names or valid options. Use at your own risk. This option overrules the basic query.

The equals logic in R is “==”.

The metadata object in this custom module is called `obs`.

**columnName**: newCol

**fullCondition**: obs$slide\_ID\_numeric == 2

A more advanced query is shown on the next page.

Advanced Query:

**Let’s say you are looking for all cells that are in slide 2 that have a high cell type probability and are in “niche6”.**

These data are found in the metadata columns “slide\_ID\_numeric”, “RNA\_nbclust\_[GUID]\_posterior\_probability”, and “spatialClusteringAssignment.”

There are two ways of creating this query within this custom module.

1. Setting individual variables and the script will generate the query.

This route allows for only AND or OR combination if there are multiple queries. Use the advanced query or run module multiple times for queries with AND and OR combination.

A typical query might be written like this: slide\_ID\_numeric = 2

Yellow (slide\_ID\_numeric): column Blue (=): logic Pink (2): value

So for our example query our values will be:

**columnName**: newCol

**column**: c( ‘slide\_ID\_numeric’, ‘RNA\_nbclust\_[GUID] \_posterior\_probability’, ‘spatialClusteringAssignments’)

**logic**: c(‘=‘, ‘>‘, ‘=‘)

**value**: c(2, 0.7, ‘niche6’)

**condition**: AND

1. Advanced Option: Provide the full query.

This is the advanced option and allows for more refined queries. But there is little to no oversight on this. There is no spell check on column names or valid options. Use at your own risk. This option overrules the basic query.

The equals logic in R is “==”.

The metadata object in this custom module is called `obs`.

**columnName**: newCol

**fullCondition**: obs$RNA\_nbclust\_[GUID]\_posterior\_probability > 0.7 & obs$slide\_ID\_numeric == 2 & obs$spatialClusteringAssignments == ‘niche6’