3D Dataset Information

# Introduction

The data is provided as part of the Gigascience publication by members of the 3D-Massomics consortium. The data is provided in two formats, both raw and processed. Specifics regarding each of the two file types are provided.

# Sample information

Slices from a single colorectal adenocarcinoma were analysed by desorption electrospray ionisation (DESI) mass spectrometry imaging (MSI).

## Preparation

Four neighbouring slices were analysed on each DESI slide, with numbering for each slice based on the depth at which the slice was cut. The first analysed slice is numbered *10* and subsequent slices increase by 10 each time (e.g. 10,20,30,etc). Slices were cut at 10 µm depth with every tenth being imaged. Thus the first slice occurs at a depth of 100 µm and the second at 200 µm. As the *x* and *y* spatial resolution is 100 µm, the three-dimensional voxels are consequentially cubic. The slices are arranged according to the figure below:

n + 10

n

n + 20

n + 30

# Raw data

The raw imzML (and associated ibd) files thus consist of 4 individual slices. The files are named such that the z-position of each slice is stated explicitly in terms of location on the slice, e.g. *120TopL, 90TopR, 110BottomL, 100BottomR-centroid.imzML*. The optical images are provided with different filenames; these start with the number of the upper-rightmost slice proceeding clockwise.

# Processing stages

The various stages involved in conversion of the raw data are summarised below.

1. Alignment of *m/z* values
2. Co-registration of optical and MS images (slides of 4 samples)
3. Separation of each slide into 4 individual slices
4. Alignment of individual slices from the front to back
5. Variable normalisation
6. File export

## Alignment of m/z values

Each pixel of each file has a separate series of *m/z* values. The alignment is performed such that each file has a common *m/z* vector and new MS images are produced for each file, with the same x/y dimensions as in the original file but a (no doubt larger) z dimension that is the same across all files.

## Co-registration

In order to be able to properly divide the slides into separate slices, the optical and MS images are aligned by means of overlap between tissue object pixels in MS and optical images. The aligned optical image is thus a warped form of the original (the MS image remains static).

## Separation

Four polygons were drawn over the newly aligned optical image. The coordinates of these regions were used to export each slice to an individual file.

## Global alignment

The individual files need to be aligned to each other. By default, the procedure was started with the first slice (i.e. slice number 10) which is used as the template image and is the only image that is does not change.

The procedure is for the optical image of a subsequent slice to be co-registered with the optical image of the preceding slice (fixed) and the required transformation is applied to both MS and optical images. These newly transformed images thus form the template for the subsequent slice. The process is continued until the last slice is reached. Following alignment, all optical and MS images have the same dimensions.

## Normalisation

Probabilistic quotient normalisation was performed across all variables. This procedure is described in the literature.

## Export

The fully processed dataset was exported to an HDF5 file, the structure of which is subsequently detailed.

# Processed data

The processed data is presented in a single H5 / HDF5 file. It contains optical and MS images for each tissue slice as well as the *z*-axes position. In addition, there is a single *m/z* vector (common for all slices) and some metadata parameters.

A function (import3dh5.m) to import the data into Matlab has been uploaded to the GitHub repository at <https://github.com/jsmckenzie/3DMassomics> which can be downloaded and modified. Additional functions for the visualisation and analysis will be provided later.

H5 files essentially store data in groups (akin to folders/directories on a computer) and datasets (the actual files). The location of a dataset may be written as */group1/group2/dataset* where the variable *dataset* is stored within the two groups labelled *group1* and *group2*. This simple approach allows for specific parts of the file to be imported. The structure within the H5 file is summarised below:

|  |  |  |
| --- | --- | --- |
| **Location** | **Contents** | **Description** |
| /  (i.e. root directory) | Metadata | Name/value pairs for the metadata |
| /mz | *m/z* vector |  |
| /data |  | Group containing each slice’s data |
| /data/s1 |  | Group containing the data from the 1st slice |
| /data/s2 |  | Group containing the data from the 2nd slice |
| /data/s2/op | Optical image |  |
| /data/s2/x | MS image (in three dimensions) |  |
| /data/s2/zPosition | Slice number/depth |  |

The number of slices is stored with the metadata (named *numSlices*). Note that the sequence of s1, s2, s3 is continuous with the final number being equal to *numSlices*. Matlab provides various functions for the reading and writing of H5 files. A full list can be accessed from here (<http://www.mathworks.co.uk/help/matlab/hdf5-files.html>) but those necessary for reading are: h5readatt, h5read and h5info.