

Apr 01, 2024

Version 1

CODA (part 1): setting up environment and preparing sample dataset | HuBMAP | JHU-TMC V.1

DOI

<https://dx.doi.org/10.17504/protocols.io.q26g71rpkgwz/v1>

Kyu Sang Han¹, Pei-Hsun Wu¹, Joel Sunshine², Sashank Reddy², Ashley Kiemen², Denis Wirtz^{1,2}

¹Johns Hopkins University; ²Johns Hopkins Medicine

Human BioMolecular Atlas Program (HuBMAP) Method Development Community

Tech. support email: Jeff.spraggins@vanderbilt.edu



Kyu Sang Han

Johns Hopkins University

Create & collaborate more with a free account

Edit and publish protocols, collaborate in communities, share insights through comments, and track progress with run records.

Create free account

OPEN  ACCESS



DOI: <https://dx.doi.org/10.17504/protocols.io.q26g71rpkgwz/v1>

Protocol Citation: Kyu Sang Han, Pei-Hsun Wu, Joel Sunshine, Sashank Reddy, Ashley Kiemen, Denis Wirtz 2024. CODA (part 1): setting up environment and preparing sample dataset | HuBMAP | JHU-TMC. **protocols.io**

<https://dx.doi.org/10.17504/protocols.io.q26g71rpkgwz/v1>

Manuscript citation:

Kiemen, A.L., Braxton, A.M., Grahn, M.P. *et al.* CODA: quantitative 3D reconstruction of large tissues at cellular resolution. *Nat Methods* **19**, 1490–1499 (2022). <https://doi.org/10.1038/s41592-022-01650-9>

License: This is an open access protocol distributed under the terms of the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: Working

We use this protocol and it's working

Created: March 28, 2024

Last Modified: April 01, 2024

Protocol Integer ID: 97507

Keywords: CODA, deeplearning, tissuelabeling, segmentation, tmc coda, dataset, hubmap, coda, tmc

Funders Acknowledgements:

Institute of Arthritis and Musculoskeletal and Skin Diseases

Grant ID: U54AR081774

National Cancer Institute

Grant ID: U54CA143868

Abstract

CODA workflow part 1. setting up environment and preparing dataset

Software requirements

- 1 MATLAB [MATLAB \(mathworks.com\)](https://www.mathworks.com/).
Image processing toolbox [Image Processing Toolbox - MATLAB \(mathworks.com\)](https://www.mathworks.com/help/images/).
Deep learning toolbox [Deep Learning Toolbox - MATLAB \(mathworks.com\)](https://www.mathworks.com/help/deeplearning/).
MATLAB - Resnet50 model [MATLAB resnet50 \(mathworks.com\)](https://www.mathworks.com/help/deeplearning/ug/resnet50.html).
- 2 Aperio ImageScope [Aperio ImageScope \(leicabiosystems.com\)](https://leica-biosystems.com/aperio-imagescope/).
- 3 FIJI ImageJ [Fiji Downloads \(imagej.net\)](https://imagej.net/).

Download Source code

- 4 Codes are available at the following GitHub: **CODA Github Repository**.

Download Sample dataset

- 5 Here, we discuss application to a sample dataset “lungs”, containing 150 serial histological images. Download the sample dataset (serial images and sample annotations) here: [**Lung Sample Dataset on Google Drive**](#)
- 6 Images are .ndpi format and were scanned at 20x magnification (approximately 0.5 micron / pixel resolution), spaced 10 micron apart. Save the images in a local drive folder (e.g. **\\Users\\Ashley\\Documents\\lungs**).
- 7 Filenames for each image should be created such that tissue sections are read consecutively by Matlab. Therefore, include zero-padding in numerical indices.

CORRECT FILENAMES: lungs_001.ndpi, lungs_002.ndpi, ..., lungs_011.ndpi

INCORRECT FILENAMES (no zero padding): **lungs_1.ndpi, lungs_2.ndpi, ..., lungs_11.ndpi**

Create downsampled copies of high-resolution images

- 8 The function **create_downsampled_tif_images** will create downsampled copies of the .ndpi files by directly loading each high-resolution images in tiles and down sampling it to the desired pixel resolutions.



- 9 First, decide the resolution of the images you want to create. Here, we create images of 1 micron / pixel, 2 microns / pixel, and 10 micron / pixel resolution:

```
ds=[1 2 10];
```

- 10 Next, decide on the name of the output folders for each of the downsampled images you create. Here, we will save the images downsampled to 1 micron / pixel in a folder named "10x", the images downsampled to 2 micron / pixel in a folder named "5x," and the images downsampled to 10 micron / pixel in a folder named "1x."

```
subfolders=["10x" "5x" "1x"];
```

- 11 Finally, call the function: **create_downsampled_tif_images(pth,ds,subfolders);**

- 12 Using this function, you will make two subfolders within the original folder containing the .ndpi images. One subfolder named "10x" containing the 20x images downsampled by a factor of 2. The other subfolder named "1x" containing the 20x images downsampled by a factor of 20. Most calculations will be performed on these tif images. Note: here we use 10x and 1x for example, but other resolutions could be created as desired.

```
pth10x=[pth,'10x'];
```

```
pth1x=[pth,'1x'];
```

****Note: If this code fails due to memory constraints on your computer, try python Openslide.**