Apr 01, 2024 Version 1

🌐 CODA (part 2): calculate registration on low-resolution tissue images | HuBMAP | JHU-TMC V.1

Kyu Sang Han[1], Pei-Hsun Wu[1], Joel Sunshine[2], Ashley Kiemen[2], Sashank Reddy[2], Denis Wirtz[1,2]

[1]Johns Hopkins University; [2]Johns 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]

Protocol status: Working
We use this protocol and it's working

Created: March 28, 2024

Last Modified: April 01, 2024

Protocol Integer ID: 97508

Keywords: resolution tissue image, tmc coda, hubmap, registration, tissue, coda, resolution, tmc, image, low resolution

Abstract

CODA workflow part 2. Calculate registration on low-resolution tissue images

Troubleshooting

Calculating the tissue space

1. For most images, a resolution of 1x should be sufficient. If the 1x results are not well registered, or registration of small tissues (such as needle biopsies) is desired, try calculating registration at a higher resolution such as 2x or 4x.

2. The quality of the registration depends on first calculating the tissue space in each image, as this allows removal of noise (shadows and lines that aren't tissue within the whole slide image).

3. The presence of noise in the images during registration can affect the codes determination of registration accuracy (QC step), as this is determined through pixel-to-pixel cross correlation.

4. Create logical images indicating locations containing tissue with the function **calculate_tissue_ws**. This function has two methods of calculating the tissue space.

5. First try with method 1:
   calc_style=1;
   calculate_tissue_ws(pth1x, calc_style);

   This function will create logical images with the same filenames as the images in pth1x, saved in a subfolder named 'TA'. Check that these tissue images look correct (are not white over background, nontissue space in the image, only white on tissue regions).

6. If they do not look correct, delete the images inside the folder named 'TA' and try again, using method 2:
   calc_style=2;
   calculate_tissue_ws(pth1x, calc_style);

Apply the CODA nonlinear image registration to the low-resolution tiff images

7. Next, apply the CODA nonlinear image registration to the low-resolution tif images. Note, this function can be applied without first creating the tissue masks with **calculate_tissue_ws.** The registration code will by default use method 1 to create the tissue masks. But it is best practice to first calculate the tissue masks and validate that they are accurate to ensure the best registration quality is achieved.
   calculate_image_registration(pth1x);

8. This function will create output folders containing the globally registered images, the elastically registered images, and the registration transformation metadata:

   pth1xG=[pth1x,'registered'];
   pth1xE=[pth1x,'registered\elastic registration'];

pthdata=[pth1x,'registered\elastic registration\save_warps\'];

9   Validate the image registration by loading the elastically aligned, downsampled z-stack in ImageJ. These validation images will be saved in a subfolder within the folder containing the low-resolution tif images:

pth_validate_regisration=[pth1x,'registered\elastic_registration\check'];

10   Notes on debugging registration: If your results are bad and you would like to re-register with different parameters, first delete the 'registered' folder (inside pth1x) created by **calculate_image_registration**. If you do not delete this subfolder the registration code will skip all images that were previously registered.

11   If the elastically registered images are too jiggly, try reducing szE and/or diE inside **calculate_image_registration**

12   If the elastically registered images are too smeared, try increasing szE and/or diE inside **calculate_image_registration**

13   If the registration is taking too long for one image (>5 min), try reducing the resolution of the images, reduce the szE or diE inside **calculate_image_registration** and/or try a computer with higher RAM