Oct 30, 2020

MAPS image analysis

jessechao [1]

[1]1. Department of Cellular and Physiological Sciences, Life Sciences Institute, University of British Columbia, Vancouver, Canada, V 6T1Z3

1 Works for me dx.doi.org/10.17504/protocols.io.bn7dmhi6

University of British Columbia

jessechao

DOI

dx.doi.org/10.17504/protocols.io.bn7dmhi6

PROTOCOL CITATION

jessechao 2020. MAPS image analysis. **protocols.io** https://dx.doi.org/10.17504/protocols.io.bn7dmhi6

CREATED

Oct 29, 2020

LAST MODIFIED

Oct 30, 2020

PROTOCOL INTEGER ID

43973

Setting up required environment and packages

1 Navigate to https://github.com/jessecanada/MAPS, download the repository.

2 The following commands assume that you are using conda (or Anaconda) to manage your python packages and that you are on a Unix type system (Linux or OSX).

2.1 Use the following commands in terminal:

create virtual environment

conda create -n myenv python=3

Create a new virtual python environment using conda. Replace "myenv" with your own preferred name.

Mac OS any

activate virtual environment

conda activate myenv

Mac OS any

install required packages

conda install --file requirements.txt

Mac OS

exit virtual environment

conda deactivate

Use this command at the end of your work session.

Mac OS

Now you are ready to rock and roll. For additional details, see https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html

Step 1 - image QC

3 Open "MAPS_1_img-QC.ipynb" in Jupiter Notebook and follow the steps in the notebook to remove blurry images.

Step 2 - cell detection

4 Follow this quick start guide to build your first object detection model on Azure. (link: https://docs.microsoft.com/en-us/azure/cognitive-services/custom-vision-service/get-started-build-detector)

Note that you can augment the training images you create on Azure. To do this, follow the optional training augmentation step below.

Take note of the endpoint, training key, prediction key, resource ID and project ID.

4.1 (optional) Once you have labeled ~100 training images, you can perform training augmentation to boost the performance of your model. Note that it is always better to label more training images.

Open "MAPS_2.1_training_augmentation.ipynb". We recommend opening this notebook on Google Colab so that you don't need to install additional packages

4.2 Open "MAPS_2_cell-detection.ipynb". We would recommend opening this notebook on Google Colab. Follow the steps in the notebook to crop out individual cells.

Step 3 - phenotype discovery

5 There are two options here. To use the method in Figure 5A, open "MAPS_3_conv_stacking.ipynb" and follow the

Citation: jessechao (10/30/2020). MAPS image analysis. https://dx.doi.org/10.17504/protocols.io.bn7dmhi6

instructions in the notebook to generate cell galleries. Alternatively, to use the method in Figure 5C (i.e. deep autoencoder), open "MAPS_3_autoencoder.ipynb" and follow the steps in the notebook. Again, we recommend opening the notebooks in Colab.

5.1 After generating cell galleries for wild type and a few variants, carefully inspect them to decide how many classes of localizations you need to classify.

Step 4 - phenotype classification

6 Gather the cropped individual cell images. Open "MAPS_4_phenotype_classification.ipynb" and follow the instructions in the notebook to make predictions and classify phenotypes. Again, we recommend opening this notebook on Colab. In the end you will get a .csv file containing predictions for each cell.

Congratulations, you've made it to the end!

Citation: jessechao (10/30/2020). MAPS image analysis. https://dx.doi.org/10.17504/protocols.io.bn7dmhi6