



Nov 29, 2023

Brain image simulation protocol

DOI

<https://dx.doi.org/10.17504/protocols.io.4r3l22bxql1y/v1>

Bin Fu¹

¹University of Cambridge



Bin Fu

University of Cambridge

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.4r3l22bxql1y/v1>

Protocol Citation: Bin Fu 2023. Brain image simulation protocol. **protocols.io**

<https://dx.doi.org/10.17504/protocols.io.4r3l22bxql1y/v1>

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: November 29, 2023



Last Modified: May 31, 2024

Protocol Integer ID: 91567

Keywords: ASAPCRN, brain image, protocol, large feature

Funders Acknowledgements:

Aligning Science Across Parkinson's

Grant ID: ASAP-000509

Abstract

This protocol details the method for simulating a brain image with small and large features

Attachments



protocol_simulation....

13KB

Materials

File required: 'imageSimulation' folder

Toolbox required: Statistics and Machine Learning Toolbox, image processing toolbox

- 1 1. Prepare **negative control images** where no fluorescent puncta were labelled. These images are considered as background images.
- 2 2. Select one or several cropped images containing **large aggregates** in the library and apply a **sigmoid function** to the selected cropped image where x in the sigmoid function is determined by **the distance to the large aggregate in the cropped image**.
- 3 3. Simulate **small puncta** with 2D gaussian distribution on a **blank image** (same image size as the negative control image), the number of which should be determined from **real images**. The sigma of puncta should be determined from **real images** as well.
- 4 4. **Add** cropped **large aggregates** onto the **background images** at a random location. Then **add** the **simulated puncta** onto **the background image** to form a simulated image with **both small and large features**.
- 5 5. Record position, intensity and background information per small puncta expect for those overlapping with the large aggregates in the simulated image and save the results.
- 6 6. **Repeat** step2 to 5 again for other background images