BrainBuilder

Luis Riquelme, Mike Gevaert

Human Brain Project

September 15, 2015

Scope

In Scope:

- Workflow breakdown into modules
- Modules are swappable

Out of Scope:

- Not showing boiler-plate
- Implementations are not scientifically accurate
- Not showing implementations

Modularity in BlueBuilder

Each Module is

- Self-contained
  - Need only to take required parameters
  - Need only to output data in a specified format
- Replaceable, individually or en-masse
- Usually created a 'Random' version, and a more complex to verify the inputs and outputs made sense

Modules in BrainBuilder

From previous work, and help from Eilif and Jean-Denis, these are the current set of modules:

- Build.Region: Select/Create Region of Interest (ROI)
- Build.Cells: Cell Positions
- Build.EI: E-I ratios
- Build.Composition.ME: METype for Soma
- Build.Placement: Morphology assignment
- Build.S2F: Functional Synapses (TBD)
- Build.NF: Channel Distribution (TBD)
- Build.SF: Synapse Functionality (TBD)

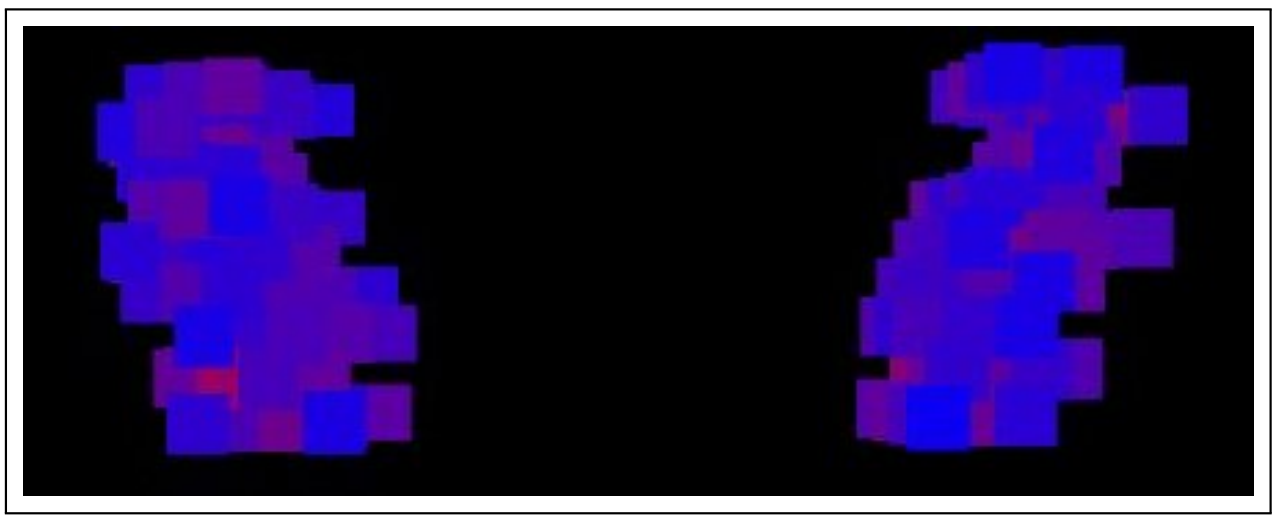
Build.Region: Region of Interest

Select region of interest by brain region or geometric primitives
Example:

```
from brainbuilder.select_region import select_region

region_name = 'Primary somatosensory area, lower limb'
roi = select_region(aibs_annotation, aibs_full_density,
    aibs_hierarchy, region_name)
```

Outputs:

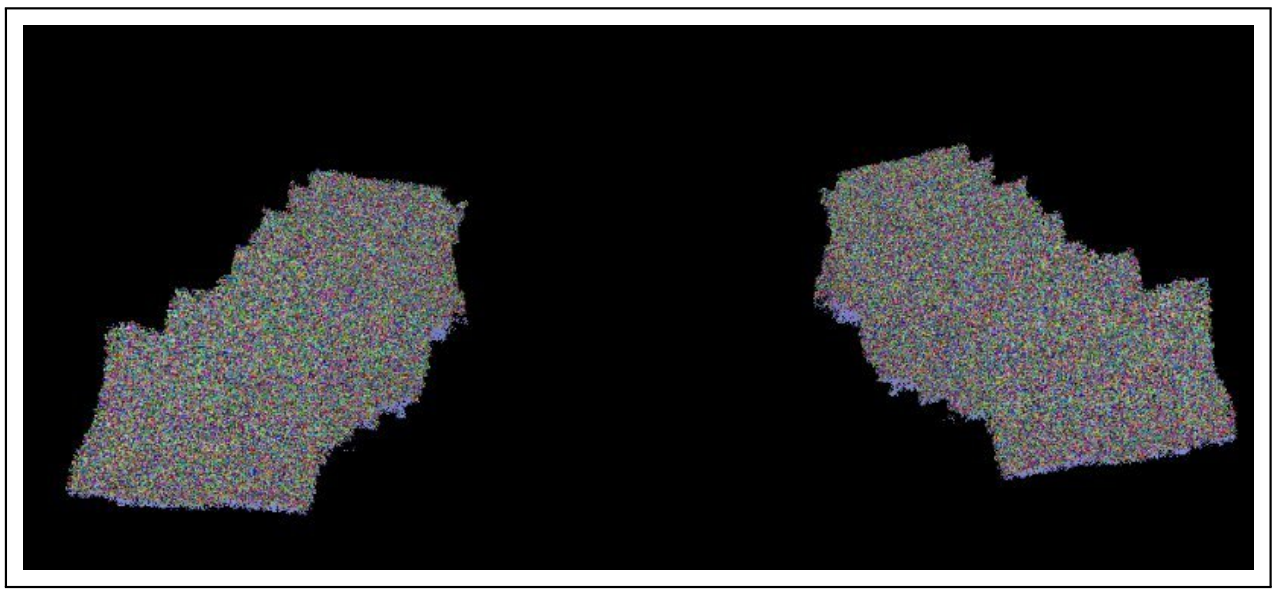


Build.Cells: Cell Positions

Generate a list of points within the ROI
Example:

```
from brainbuilder.cell_positioning import cell_positioning
positions = cell_positioning(roi, voxel_dimensions,
    total_cell_count)
```

Outputs:

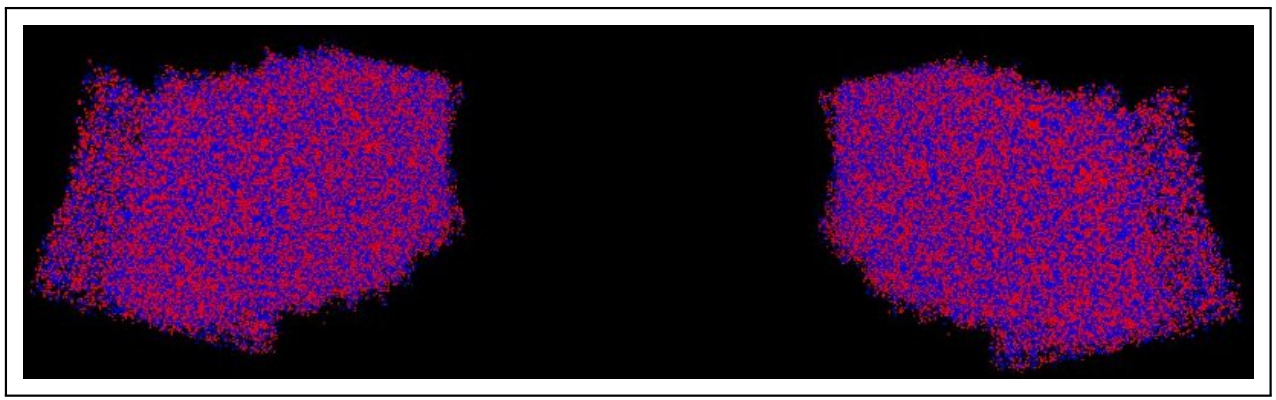


Build.EI: E-I Assignment - Random Assignment from Ratio

Assign each cell position it's E-I-ness

```
from brainbuilder.assignment_synapse_class import
    assign_synapse_class_randomly
chosen_synapse_class =
    assign_synapse_class_randomly(positions,
    inhibitory_fraction=0.5)
```

Outputs:

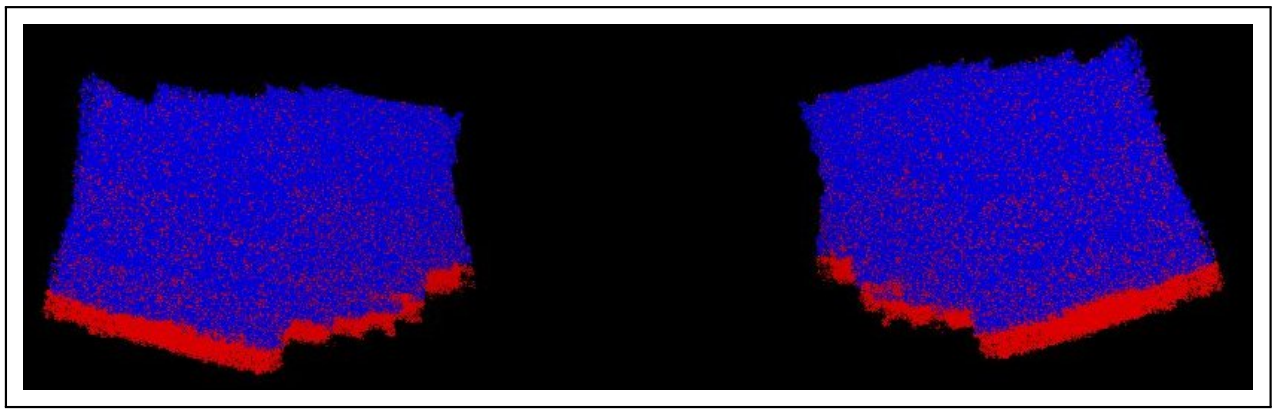


Build.EI: E-I Assignment - From Spatial Distribution

Example:

```
from brainbuilder.assignment_synapse_class import
    assign_synapse_class_from_spatial_dist
chosen_synapse_class =
    assign_synapse_class_from_spatial_dist(positions,
    sclass_sdist, voxel_dimensions)
```

Outputs:

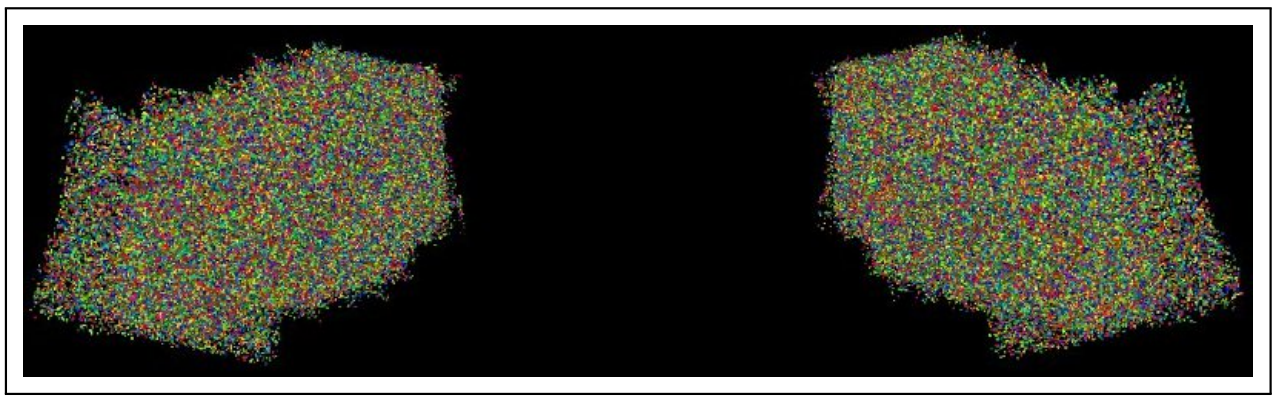


Build.Composition.ME: METype for Soma - Random

Assign each cell position ME-type

```
from brainbuilder.assignment_metype import
    assign_metype_random
chosen_me = assign_metype_random(positions, mtypes,
    etypes)
```

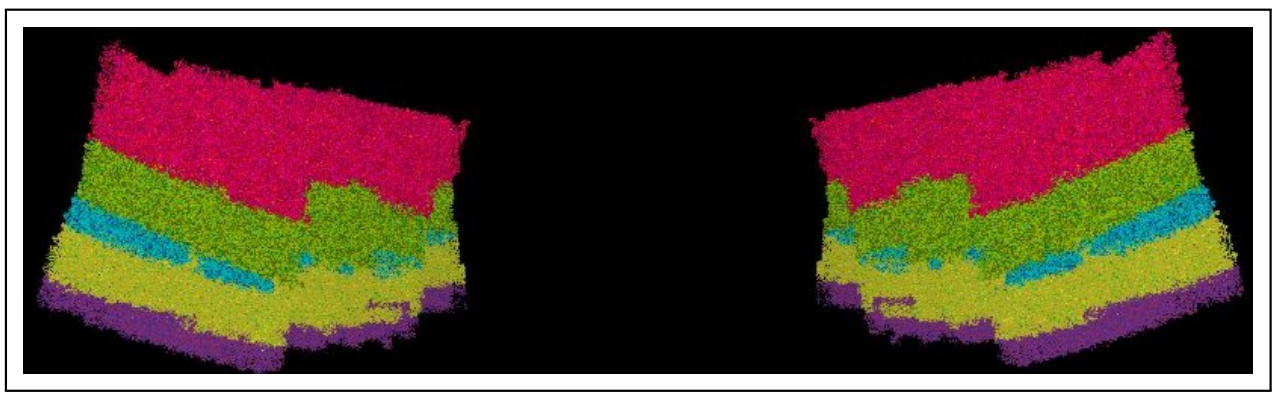
Outputs:



Build.Composition.ME: METype for Soma

```
from brainbuilder.assignment_metype import assign_metype
chosen_me = assign_metype(positions,
    chosen_synapse_class, recipe_sdist, voxel_dimensions)
```

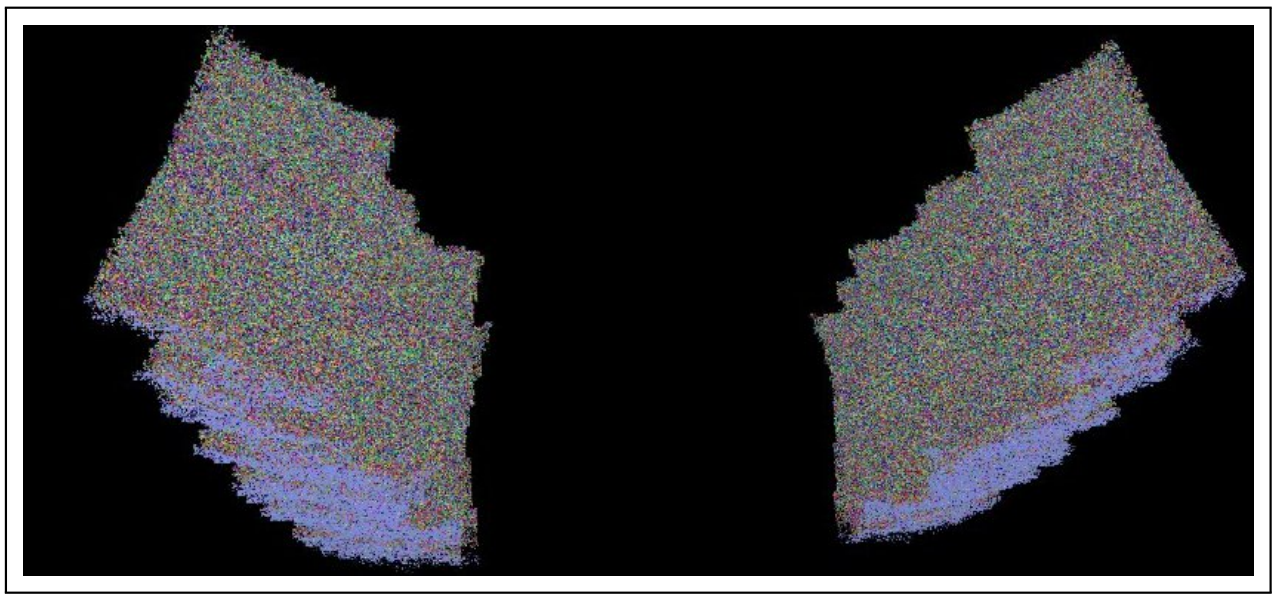
Outputs:



Build.Placement: Morphology assignment

```
from brainbuilder.assignment_morphology import
    assign_morphology
chosen_morphology = assign_morphology(positions,
    chosen_me, neuron_sdist, voxel_dimensions)
```

Outputs:



Roadmap

- Modules to be determined: need more scientific input
  - Build.S2F: Functional Synapses
  - Build.NF: Channel Distribution
  - Build.SF: Synapse Functionality
  - Note: MVD3
- Standalone version
- GUI Controlled Version, integrated with the Collab