Co-funded by
the European Union

Human Brain Project

Introduction to NEST 3

Håkon Mørk

Try it!

- https://github.com/nest/nest-simulator/tree/nest-3
- Docker:

```
docker run --rm -e LOCAL_USER_ID=`id -u $USER` \
-v $(pwd):/opt/data -p 8080:8080 nestsim/nest:3.0 notebook
```

Changes from NEST 2.x to NEST 3.0

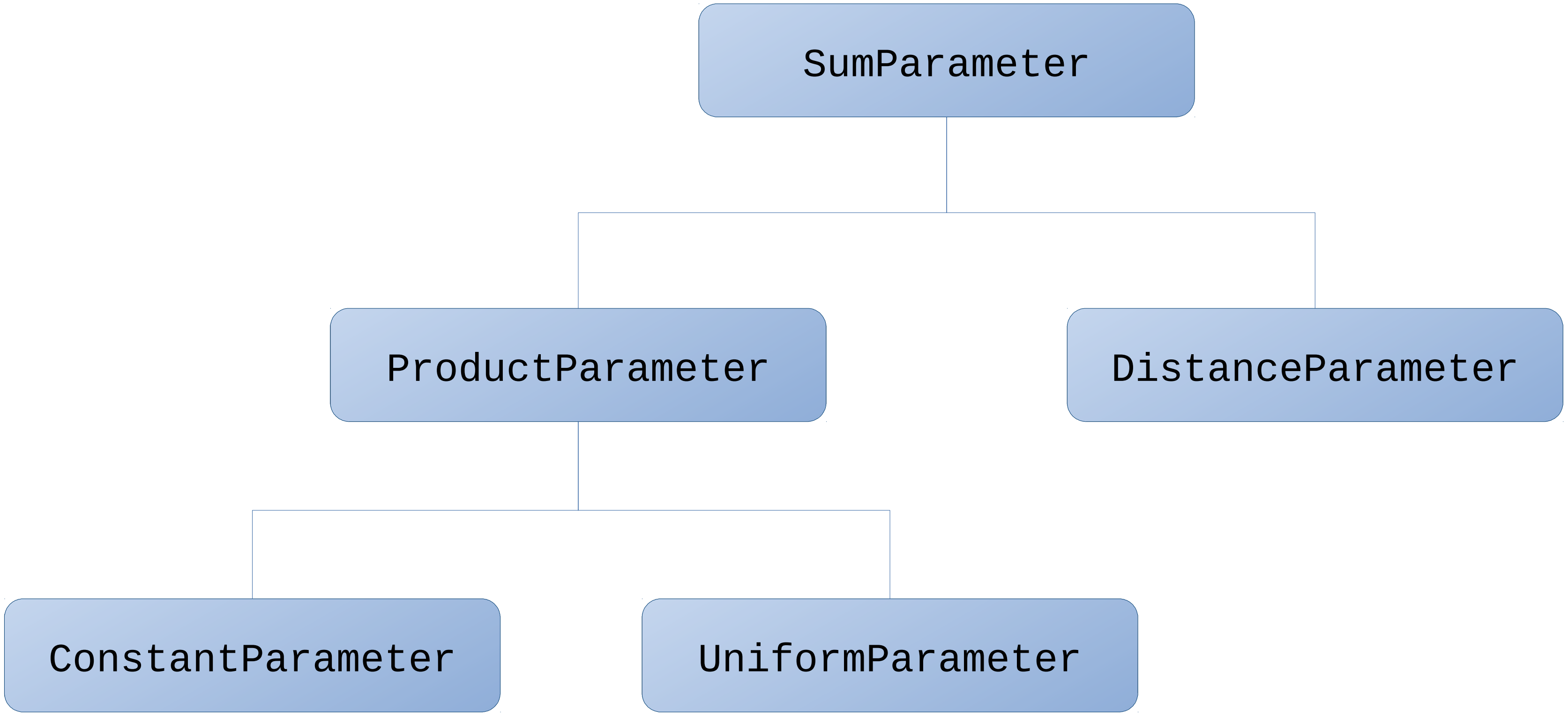
- Additions
  - NodeCollections for managing collections of nodes
  - SynapseCollections for managing connections
  - Object-oriented handling of NodeCollections and SynapseCollections
  - Parameters (e.g. `nest.random.uniform()`)
  - New recording backend (NESTio)
- Topology functions merged into NEST
- Removals
  - Subnets
  - `iaf_neuron` (use `iaf_psc_alpha`)

Parameters in NEST 3.0

- Parameters can be representations of
  - Values drawn from a random distribution
  - Values based on various spatial node parameters
- Parameters can be used to
  - Set node status
  - Create node positions
  - Define connection probabilities, weights and delays
- Can be combined in different ways
- Can be used with some provided mathematical functions

Structure of combined Parameters

```
parameter = 0.7 * nest.random.uniform() + nest.spatial.distance
```



Getting and setting node and connection attributes

NEST 2.x

```
nodes = nest.Create('iaf_psc_alpha', 10)
nest.SetStatus(nodes, {'V_m': -55.})
vm = nest.GetStatus(nodes, 'V_m')

conns = nest.GetConnections()
weights = nest.GetStatus(conns, 'weight')
```

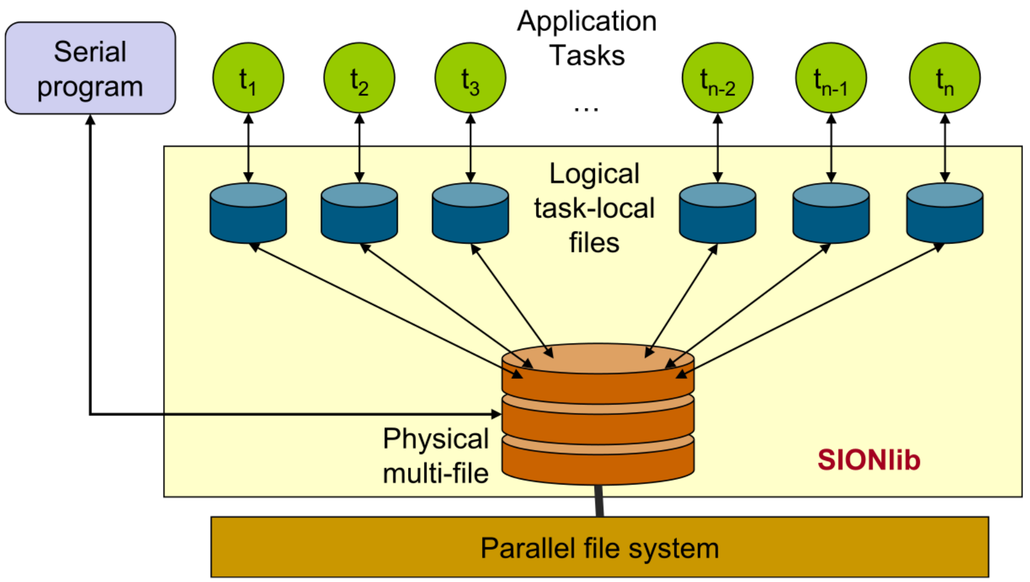
NEST 3.0

```
nodes = nest.Create('iaf_psc_alpha', 10)
nodes.V_m = -55.
vm = nodes.V_m

conns = nest.GetConnections()
weights = conns.weight
```

Recording backend saving to binary file (SIONlib)

Individual ranks write to task local streams that map into a physical multifile.



Wolfgang Frings, 2016

Data is a weakly ordered stream optimized for writing, with a final metadata block describing the devices, measures and neurons in the file.

NEST 3.0 demo

Jupyter notebooks in **doc/nest-3**