

JSON file content, structure and usage

JSON file containing copies information are organized in a dictionary and subdictionary structure as follows.

A single JSON file contains a set of copies.

Every copy is identified by an unique label like this:

```
neuronLabel = "95817008_iadap_0_Idep_1_Idep0_1_c_1_d_1_R_0_0_0_0_0_0_p_0.6"
```

So, if we are interested in retrieving this copy, we need to ask for the label (Python):

```
neuronCopiesDatabase[neuronLabel]
```

Labels are structured as follows:

95817008_iadap_0_Idep_1_Idep0_1_c_1_d_1_R_0_0_0_0_0_0_p_0.6

Neuron name from which the copy has originated;

95817008_**iadap_0_Idep_1_Idep0_1_c_1_d_1_R_0_0_0_0_0_0_p_0.6**

parameter name. Following nomenclature holds:

- iadap => $\max I_{adap}^{start}$
- Idep => I_{dep}^{start}
- Idep0 => I_{dep}^0
- c => c
- d => η
- R => firing block parameters
- P => p

95817008_iadap_**0_Idep_1_Idep0_1_c_1_d_1_R_0_0_0_0_0_0_p_0.6**

Parameter values. In particular:

iadap_**0_**: multiplier for $\max I_{adap}^{start}$. Original neurons values is 0

Idep_**1_**: multiplier for I_{dep}^{start} . Original neurons values is 1

Idep0_**1_**: multiplier for I_{dep}^0 . Original neurons values is 1

c_**1_**: multiplier for c . Original neurons values is 1

d_**1_**: multiplier for η . Original neurons values is 1

R_**0_0_0_0_0_0_0**: perturbation value for firing block parameters. These parameters are organized in this order:

$$J_{block}^{sup} - \varepsilon_A^{sup} - \varepsilon_B^{sup} - J_{block}^{inf} - \varepsilon_A^{inf} - \varepsilon_B^{inf}$$

Where J is such that $I_{copy} = I_{original} + J$.

`p_0.6`: multiplier for p . Original neurons values is 1.

In this example:

`95817008_iadap_0_Idep_1_Idep0_1_c_1_d_1_R_0_0_0_0_0_0_p_0.6`

this label refers to a copy where the only parameter that differs from the original neuron is p .

JSON structure

For each copy the following fields are present:

- name
- type
- spike_currents
- parameters
- block_line_params
- spike_times_exp
- spike_times_sim

In order to retrieve one of this information the following command holds (Python):

```
neuronLabel = "95817008_iadap_0_Idep_1_Idep0_1_c_1_d_1_R_0_0_0_0_0_0_p_0.6"
```

```
neuronCopiesDatabase[neuronLabel]["name"]
```

```
>> 95817008_copy
```

```
neuronCopiesDatabase[neuronLabel]["block_line_params"]
```

```
>> [700,0.15,0.25,300,0.4,25.8]
```

Parameters

`parameters` key contains a subdictionary of all the numerical parameters necessary for AGLIF simulations. Here the list of the parameters:

- EL
- Vres
- VTM
- Iadap
- Cm
- Ith
- tao
- sc
- alpha
- bet
- delta1

- Idep_ini
- Idep_ini_vr
- psi
- time scale
- A
- B
- C
- alphaD

So, in order to retrieve the parameter value, the following syntax holds (Python):

```
neuronCopiesDatabase[neuronLabel]["parameters"]["Iadap"]
```

```
>> 1.18
```