



HAL
open science

A long journey into reproducible computational neuroscience research

Meropi Topalidou, Arthur Leblois, Thomas Boraud, Nicolas P. Rougier

► To cite this version:

Meropi Topalidou, Arthur Leblois, Thomas Boraud, Nicolas P. Rougier. A long journey into reproducible computational neuroscience research. Fourth International Symposium on Biology of Decision Making (SBDM 2014), May 2014, Paris, France. hal-01090648

HAL Id: hal-01090648

<https://inria.hal.science/hal-01090648>

Submitted on 3 Dec 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

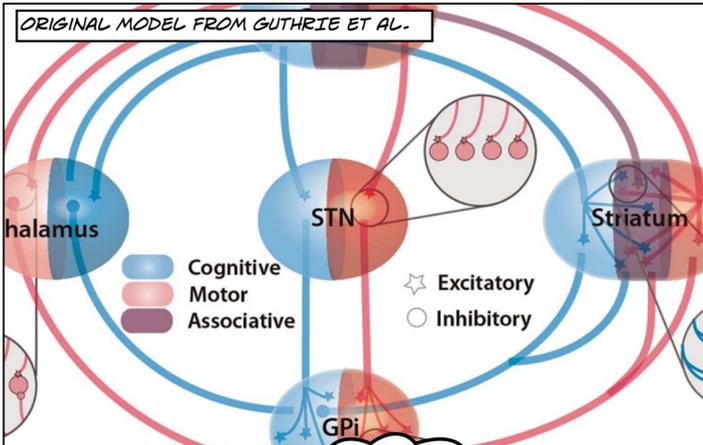
L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A LONG JOURNEY INTO REPRODUCIBLE COMPUTATIONAL NEUROSCIENCE

100% PYTHON!
SPIKE FREE!

MEROPI TOPALIDOU^{1,2} - ARTHUR LEBLOIS³
THOMAS BORAUD¹ - NICOLAS ROUGIER^{1,2}

¹ INSTITUTE OF NEURODEGENERATIVE DISEASES - UNIVERSITÉ VICTOR SEGALEN - BORDEAUX 2
² CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE, UMR 5293, BORDEAUX
³ LABORATORY OF NEUROPHYSICS AND PHYSIOLOGY UNIVERSITE PARIS DESCARTES
CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE, UMR 8119, PARIS



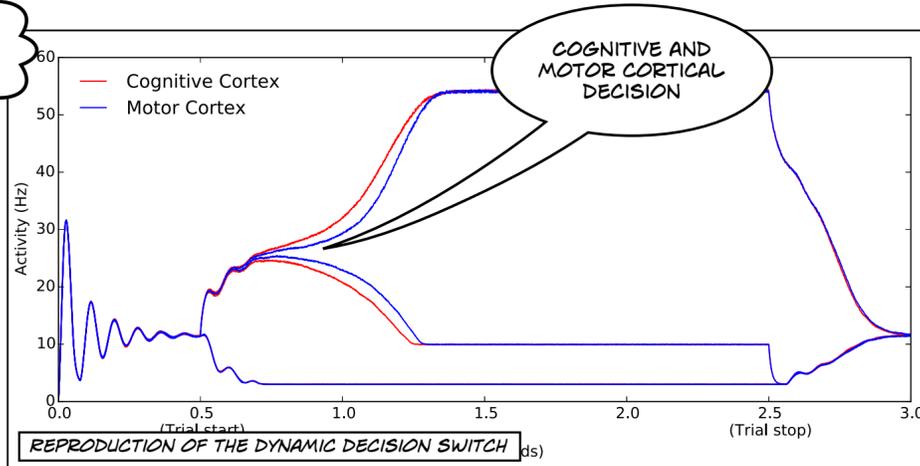
IN A PREVIOUS MODELING STUDY, LEBLOIS ET AL. (2006) DEMONSTRATED AN ACTION SELECTION MECHANISM IN CORTICO-BASAL GANGLIA LOOPS BASED ON COMPETITION BETWEEN THE POSITIVE FEEDBACK, DIRECT PATHWAY THROUGH THE STRIATUM AND THE NEGATIVE FEEDBACK, HYPERDIRECT PATHWAY THROUGH THE SUBTHALAMIC NUCLEUS.

IN GUTHRIE ET AL. (2013), AUTHORS INVESTIGATED HOW MULTIPLE LEVEL ACTION SELECTION COULD BE PERFORMED BY THE BASAL GANGLIA. TO DO THIS, THE MODEL IS EXTENDED IN A MANNER CONSISTENT WITH KNOWN ANATOMY AND ELECTRO-PHYSIOLOGY IN THREE MAIN AREAS. FIRST, TWO-LEVEL DECISION MAKING HAS BEEN INCORPORATED, WITH A COGNITIVE LEVEL SELECTING BASED ON CUE SHAPE AND A MOTOR LEVEL SELECTING BASED ON CUE POSITION. WE SHOW THAT THE DECISION MADE AT THE COGNITIVE LEVEL CAN BE USED TO BIAS THE DECISION AT THE MOTOR LEVEL. WE THEN DEMONSTRATE THAT, FOR ACCURATE TRANSMISSION OF INFORMATION BETWEEN DECISION-MAKING LEVELS, LOW EXCITABILITY OF STRIATAL PROJECTION NEURONS IS NECESSARY, A GENERALLY OBSERVED ELECTROPHYSIOLOGICAL FINDING. SECOND, INSTEAD OF PROVIDING A BIASING SIGNAL BETWEEN CUE CHOICES AS AN EXTERNAL INPUT TO THE NETWORK, WE SHOW THAT THE ACTION SELECTION PROCESS CAN BE DRIVEN BY REASONABLE LEVELS OF NOISE. FINALLY, WE INCORPORATE DOPAMINE MODULATED LEARNING AT CORTICOSTRIATAL SYNAPSES. AS LEARNING PROGRESSES, THE ACTION SELECTION BECOMES BASED ON LEARNED VISUAL CUE VALUES AND IS NOT INTERFERED WITH BY THE NOISE THAT WAS NECESSARY BEFORE LEARNING.

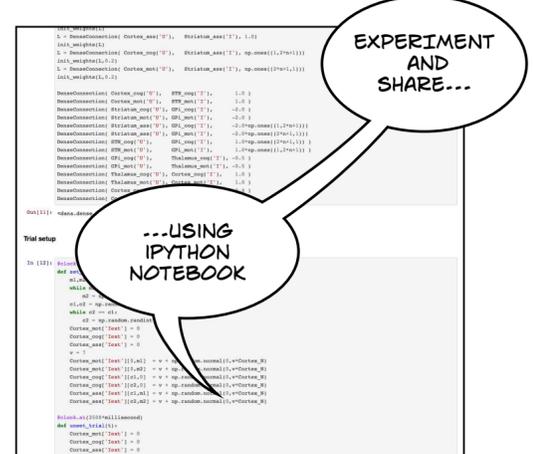
HOWEVER, THE MODEL WAS NOT REPRODUCIBLE FROM THE ARTICLE DESCRIPTION...



GITHUB.COM/ROUGIER/NEUROSCIENCES



REPRODUCTION OF THE DYNAMIC DECISION SWITCH (ds)



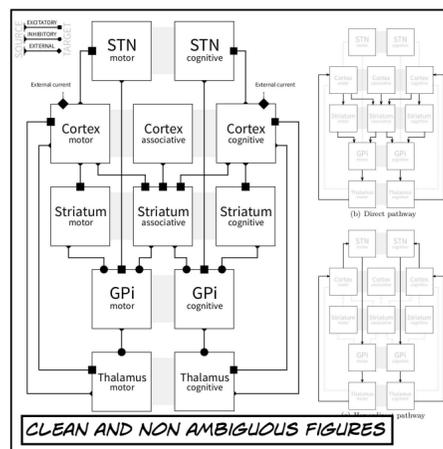
IF REPRODUCIBILITY IS THE HALLMARK OF SCIENCE, NON-REPRODUCIBILITY SEEMS TO BE THE HALLMARK OF COMPUTATIONAL NEUROSCIENCES. GUTHRIE ET AL. (2013) IS A PROTOTYPIC CASE OF SUCH NON-REPRODUCIBLE COMPUTATIONAL NEUROSCIENCE RESEARCH EVEN THOUGH THE PROPOSED MODEL GIVES A FAIR ACCOUNT OF DECISION MAKING IN THE BASAL GANGLIA COMPLEX.

WHILE TRYING TO REPLICATE RESULTS STARTING FROM THE ARTICLE DESCRIPTION, WE SOON REALIZED SOME INFORMATION WERE UNDISCLOSED, SOME OTHER WERE AMBIGUOUS AND THERE WERE ALSO SOME FACTUAL ERRORS. EVEN AFTER ACCESSING THE ORIGINAL SOURCES (GOOD LINES OF PASCAL), WE WERE STILL UNABLE TO UNDERSTAND HOW THE MODEL WORKED. IN THE END, ONLY THE ORIGINAL MATERIAL (A WINDOWS EXECUTABLE) ALLOWED US TO ACCESS THE MISSING INFORMATION AND AFTER TWO MONTHS OF INTENSIVE REFACTORED, WE WERE FINALLY ABLE TO REPLICATE RESULTS USING ONLY 200 LINES OF PYTHON.

UNFORTUNATELY, SUCH LOOSE DESCRIPTION IS NOT AN ISOLATED CASE !!!

TO BE CONTINUED...

BORING BUT INCREDIBLY USEFUL!



CLEAN AND NON AMBIGUOUS FIGURES

A Model Summary					
Populations	Twelve: Cortex (motor, associative & cognitive), Striatum (motor, associative & cognitive), GPi (motor & cognitive), STN (motor & cognitive), Thalamus (motor & cognitive)				
Topology	-				
Connectivity	one to one, one to many (divergent), many to one (convergent)				
Neuron model	Dynamic rate model				
Channel model	-				
Synapse model	Linear synapse				
Plasticity	Reinforcement learning rule				
Input	External current in cortical areas (motor, associative & cognitive)				
Measurements	Firing rate				
B Populations					
Name	Elements	Size	Threshold (h)	Noise	Initial state
Cortex motor	Linear neuron	1 x 4	-3	1.0%	0.0
Cortex cognitive	Linear neuron	4 x 1	-3	1.0%	0.0
Cortex associative	Linear neuron	4 x 4	-3	1.0%	0.0
Striatum motor	Sigmoidal neuron	1 x 4	0	0.1%	0.0
Striatum cognitive	Sigmoidal neuron	4 x 1	0	0.1%	0.0
Striatum associative	Sigmoidal neuron	4 x 4	0	0.1%	0.0
GPi motor	Linear neuron	1 x 4	+10	3.0%	0.0
GPi cognitive	Linear neuron	4 x 1	+10	3.0%	0.0
STN motor	Linear neuron	1 x 4	-10	0.1%	0.0
STN cognitive	Linear neuron	4 x 1	-10	0.1%	0.0
Thalamus motor	Linear neuron	1 x 4	-40	0.1%	0.0
Thalamus cognitive	Linear neuron	4 x 1	-40	0.1%	0.0
Values (V _i)	Scalar	4	-	-	0.5
C Connectivity					
Source	Target	Pattern	Weight (W)	Gain (G)	Plastic
Cortex motor	Thalamus motor	(i, i) → (i, i)	1.0	0.4	No
Cortex cognitive	Thalamus cognitive	(i, i) → (i, i)	1.0	0.4	No
Cortex motor	STN motor	(i, i) → (i, i)	1.0	1.0	No
Cortex cognitive	STN cognitive	(i, i) → (i, i)	1.0	1.0	No
Cortex motor	Striatum motor	(i, i) → (i, i)	N(0.5, 0.005)	1.0	Yes
Cortex cognitive	Striatum cognitive	(i, i) → (i, i)	N(0.5, 0.005)	1.0	Yes
Cortex motor	Striatum associative	(i, i) → (*, i)	N(0.5, 0.005)	0.2	Yes
Cortex cognitive	Striatum associative	(i, i) → (i, *)	N(0.5, 0.005)	0.2	Yes
Cortex associative	Striatum associative	(i, j) → (i, j)	N(0.5, 0.005)	1.0	Yes
Thalamus motor	Cortex motor	(i, i) → (i, i)	1.0	1.0	No
Thalamus cognitive	Cortex cognitive	(i, i) → (i, i)	1.0	1.0	No
GPi motor	Thalamus motor	(i, i) → (i, i)	1.0	-0.5	No
GPi cognitive	Thalamus cognitive	(i, i) → (i, i)	1.0	-0.5	No
STN motor	GPi motor	(i, i) → (i, i)	1.0	1.0	No
STN cognitive	GPi cognitive	(i, i) → (i, i)	1.0	1.0	No
Striatum motor	GPi motor	(i, i) → (i, i)	1.0	-2.0	No
Striatum cognitive	GPi cognitive	(i, i) → (i, i)	1.0	-2.0	No
Striatum associative	GPi motor	(*, i) → (i, i)	1.0	-2.0	No
Striatum associative	GPi cognitive	(i, *) → (i, i)	1.0	-2.0	No

D1 Neuron Model	
Name	Linear neuron
Type	Rate model
Membrane Potential	$\tau dV/dt = -V + I_{syn} + I_{ext} - h$ $U = \max(V, 0)$
D2 Neuron Model	
Name	Sigmoidal neuron
Type	Rate model
Membrane Potential	$\tau dV/dt = -V + I_{syn} + I_{ext} - h$ $U = V_{min} - (V_{max} - V_{min}) / (1 + e^{-\frac{V-V_{c}}{V_c}})$
E Synapse	
Name	Linear synapse
Type	Weighted sum
Output	$I_{syn} = \sum A \in sources (G_{A \rightarrow B} W_{A \rightarrow B} U_A)$
F Plasticity	
Name	Reinforcement learning
Type	Delta rule
Delta	$\Delta W_{A \rightarrow B} = \alpha \times PE \times U_B$ $PE = Reward - V_i$ $\alpha = 0.01$ if $PE < 0$ (LTD), $\alpha = 0.02$ if $PE > 0$ (LTP)
G Input	
Type	Cortical input
Description	A trial is preceded by a settling period (500ms) and followed by a reset period. At time $t = 0$, two shapes are presented in cortical cognitive area ($I_{ext} = 7$ at $\{i_1, i_2\}$) at two different locations in cortical motor area ($I_{ext} = 7$ at $\{j_1, j_2\}$) and the cortical associate area is updated accordingly ($I_{ext} = 7$ at $\{i_1, i_2\} \times \{j_1, j_2\}$).
Timing	Trial start Stimulus onset Stimulus offset Reset -500ms 0 2500 ms 3000 ms
H Measurements	
Site	Cortical areas
Data	Activity in cognitive and motor cortex Cortico-striatal weights
I Environment	
OS	OSX 10.9 (maverick)
Language	Python 2.7.6 (brew installation)
Libraries	Numpy 1.8.1 (pip installation) SciPy 0.13.3 (pip installation) IPython 1.2.1 (pip installation) Matplotlib 1.3.0 (pip installation) DANA 0.5.0 (pip installation)
Tools	Safari browser (native)

TABULAR DESCRIPTION OF THE MODEL FROM NORDLIE ET AL. (2009)

```

# Parameters
N = 12
Nc = 4
Nm = 4
Na = 4
Nstn = 4
Nth = 4
Ngt = 4
Ngc = 4
Ngs = 4
Ngtm = 4
Ngtc = 4
Ngtas = 4

# Weights
Wc = 1.0
Wm = 1.0
Wa = 1.0
Wstn = 1.0
Wth = 1.0
Wgt = 1.0
Wgc = 1.0
Wgs = 1.0
Wgtm = 1.0
Wgtc = 1.0
Wgtas = 1.0

# Synapses
S = {}
S['cortex_motor'] = {'source': 'cortex_motor', 'target': 'thalamus_motor', 'weight': 1.0, 'gain': 0.4, 'plastic': False}
S['cortex_cognitive'] = {'source': 'cortex_cognitive', 'target': 'thalamus_cognitive', 'weight': 1.0, 'gain': 0.4, 'plastic': False}
S['cortex_stn'] = {'source': 'cortex_motor', 'target': 'stn_motor', 'weight': 1.0, 'gain': 1.0, 'plastic': False}
S['cortex_stn'] = {'source': 'cortex_cognitive', 'target': 'stn_cognitive', 'weight': 1.0, 'gain': 1.0, 'plastic': False}
S['cortex_striatum'] = {'source': 'cortex_motor', 'target': 'striatum_motor', 'weight': 1.0, 'gain': 1.0, 'plastic': True}
S['cortex_striatum'] = {'source': 'cortex_cognitive', 'target': 'striatum_cognitive', 'weight': 1.0, 'gain': 1.0, 'plastic': True}
S['cortex_striatum'] = {'source': 'cortex_motor', 'target': 'striatum_associative', 'weight': 1.0, 'gain': 0.2, 'plastic': True}
S['cortex_striatum'] = {'source': 'cortex_cognitive', 'target': 'striatum_associative', 'weight': 1.0, 'gain': 0.2, 'plastic': True}
S['cortex_striatum'] = {'source': 'cortex_associative', 'target': 'striatum_associative', 'weight': 1.0, 'gain': 1.0, 'plastic': True}
S['thalamus_striatum'] = {'source': 'thalamus_motor', 'target': 'striatum_motor', 'weight': 1.0, 'gain': 1.0, 'plastic': False}
S['thalamus_striatum'] = {'source': 'thalamus_cognitive', 'target': 'striatum_cognitive', 'weight': 1.0, 'gain': 1.0, 'plastic': False}
S['gpi_striatum'] = {'source': 'striatum_motor', 'target': 'gpi_motor', 'weight': 1.0, 'gain': -2.0, 'plastic': False}
S['gpi_striatum'] = {'source': 'striatum_cognitive', 'target': 'gpi_cognitive', 'weight': 1.0, 'gain': -2.0, 'plastic': False}
S['gpi_striatum'] = {'source': 'striatum_associative', 'target': 'gpi_motor', 'weight': 1.0, 'gain': -2.0, 'plastic': False}
S['gpi_striatum'] = {'source': 'striatum_associative', 'target': 'gpi_cognitive', 'weight': 1.0, 'gain': -2.0, 'plastic': False}
S['stn_gpi'] = {'source': 'stn_motor', 'target': 'gpi_motor', 'weight': 1.0, 'gain': 1.0, 'plastic': False}
S['stn_gpi'] = {'source': 'stn_cognitive', 'target': 'gpi_cognitive', 'weight': 1.0, 'gain': 1.0, 'plastic': False}

# External currents
I_ext = {}
I_ext['cortex_motor'] = 7.0
I_ext['cortex_cognitive'] = 7.0
I_ext['cortex_associative'] = 7.0
I_ext['striatum_motor'] = 7.0
I_ext['striatum_cognitive'] = 7.0
I_ext['striatum_associative'] = 7.0
I_ext['thalamus_motor'] = 7.0
I_ext['thalamus_cognitive'] = 7.0
I_ext['gpi_motor'] = 7.0
I_ext['gpi_cognitive'] = 7.0
I_ext['stn_motor'] = 7.0
I_ext['stn_cognitive'] = 7.0
I_ext['thalamus_motor'] = 7.0
I_ext['thalamus_cognitive'] = 7.0
I_ext['gpi_motor'] = 7.0
I_ext['gpi_cognitive'] = 7.0
I_ext['stn_motor'] = 7.0
I_ext['stn_cognitive'] = 7.0

# Initial states
V = {}
V['cortex_motor'] = 0.0
V['cortex_cognitive'] = 0.0
V['cortex_associative'] = 0.0
V['striatum_motor'] = 0.0
V['striatum_cognitive'] = 0.0
V['striatum_associative'] = 0.0
V['gpi_motor'] = 0.0
V['gpi_cognitive'] = 0.0
V['stn_motor'] = 0.0
V['stn_cognitive'] = 0.0
V['thalamus_motor'] = 0.0
V['thalamus_cognitive'] = 0.0
V['gpi_motor'] = 0.0
V['gpi_cognitive'] = 0.0
V['stn_motor'] = 0.0
V['stn_cognitive'] = 0.0

# Simulation
t = 0
while t < 3000:
    # Update membrane potentials
    for neuron in neurons:
        V[neuron] = max(V[neuron], 0)
        dV = -V[neuron] + I_ext[neuron] + I_syn[neuron] - h[neuron]
        V[neuron] += dt * dV

    # Update synaptic weights
    for synapse in synapses:
        U_A = V[synapse.source]
        U_B = V[synapse.target]
        PE = Reward - V[synapse.target]
        delta_W = alpha * PE * U_B
        W[synapse] += delta_W

    # Update external currents
    I_ext['cortex_motor'] = 7.0
    I_ext['cortex_cognitive'] = 7.0
    I_ext['cortex_associative'] = 7.0
    I_ext['striatum_motor'] = 7.0
    I_ext['striatum_cognitive'] = 7.0
    I_ext['striatum_associative'] = 7.0
    I_ext['thalamus_motor'] = 7.0
    I_ext['thalamus_cognitive'] = 7.0
    I_ext['gpi_motor'] = 7.0
    I_ext['gpi_cognitive'] = 7.0
    I_ext['stn_motor'] = 7.0
    I_ext['stn_cognitive'] = 7.0
    I_ext['thalamus_motor'] = 7.0
    I_ext['thalamus_cognitive'] = 7.0
    I_ext['gpi_motor'] = 7.0
    I_ext['gpi_cognitive'] = 7.0
    I_ext['stn_motor'] = 7.0
    I_ext['stn_cognitive'] = 7.0

    t += dt
    
```

DANA RULES!

DANA IS A PYTHON FRAMEWORK FOR DISTRIBUTED, ASYNCHRONOUS, NUMERICAL AND ADAPTIVE COMPUTING. THE COMPUTATIONAL PARADIGM SUPPORTING THE DANA FRAMEWORK IS GROUNDED ON THE NOTION OF A UNIT THAT IS ESSENTIALLY A SET OF ARBITRARY VALUES THAT CAN VARY ALONG TIME UNDER THE INFLUENCE OF OTHER UNITS AND LEARNING. EACH UNIT CAN BE CONNECTED TO ANY OTHER UNIT (INCLUDING ITSELF) USING A WEIGHTED LINK AND A GROUP IS A STRUCTURED SET OF SUCH HOMOGENEOUS UNITS.

HTTP://DANA.LORIA.FR