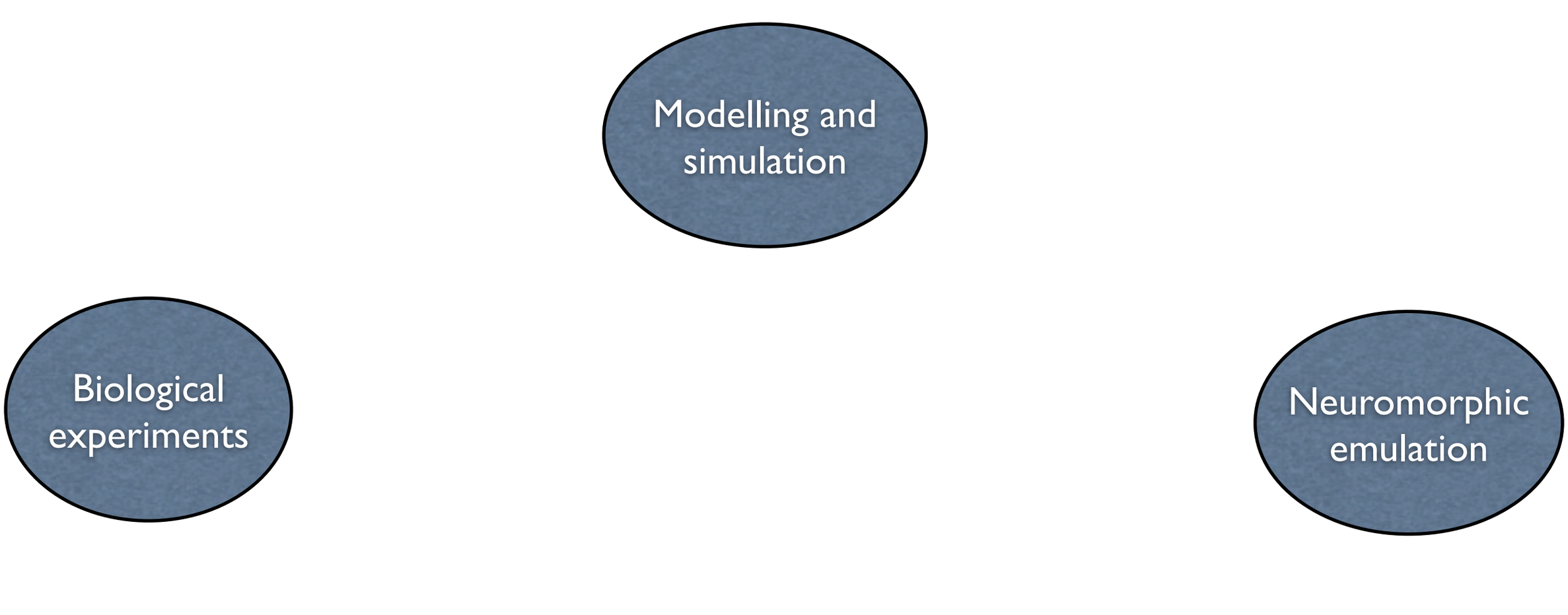
Integration in FACETS

a collaborative software tool-chain for neuromorphic computation

Andrew Davison
UNIC, CNRS,
Gif sur Yvette,
France
http://www.andrewdavison.info

Frontiers in Neuromorphic Computation. Paris. 3-4 June 2010

We heard today and yesterday about the range of activities going on in FACETS, from biological experiments, through numerical simulations to neuromorphic hardware development.
Now I'm going to talk about some of the interactions between these different activities.



There are three fields represented within FACETS (and in neuromorphic computation in general). Most of the interaction between these domains takes place by exchange of knowledge, where I define knowledge as what we find in the Conclusions section of a paper. All three domains contribute to our collective knowledge about neuroscience and about computation. In turn, we use this knowledge in designing new experiments, new models, in designing new systems.

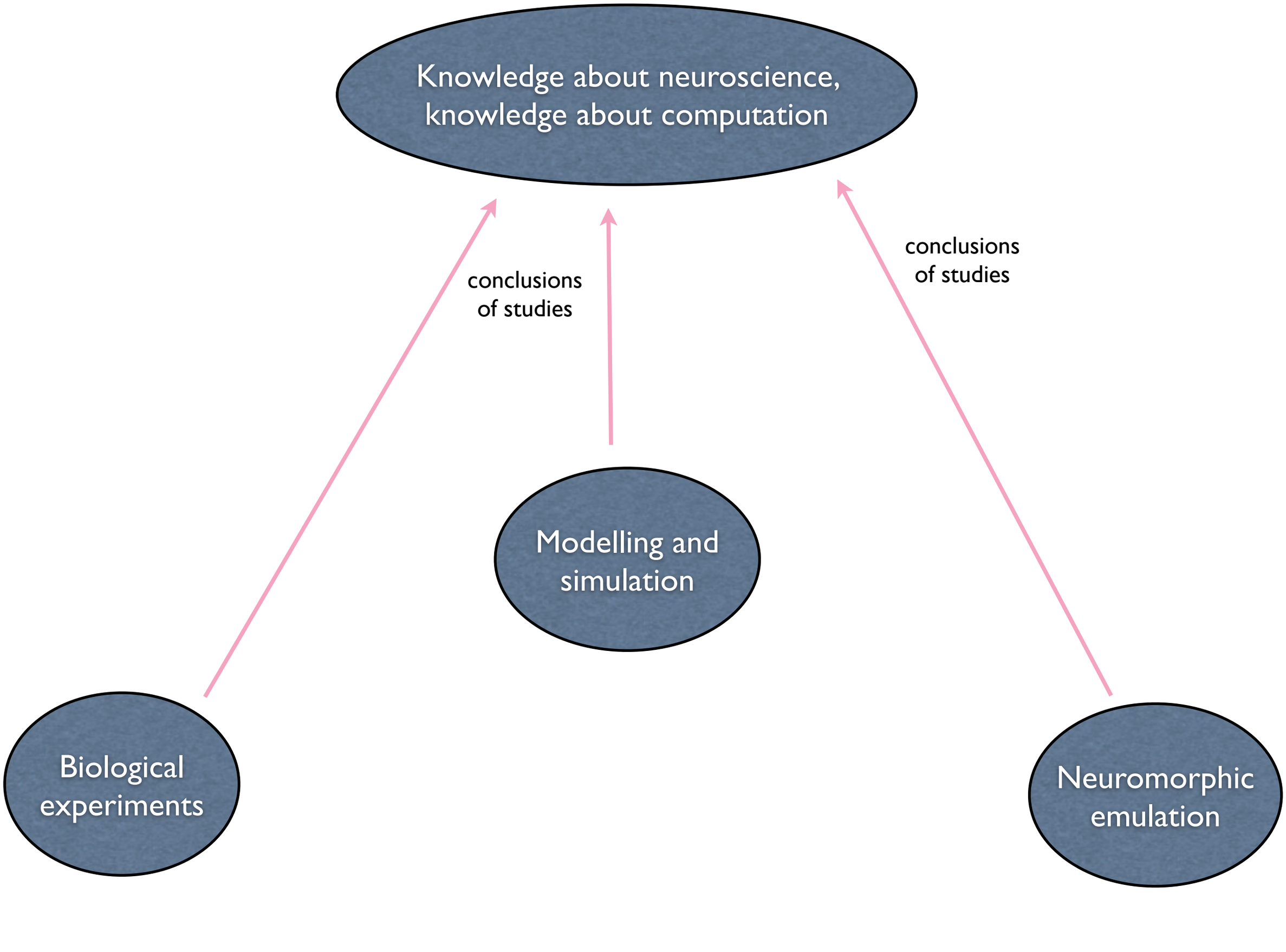
Knowledge about neuroscience,
knowledge about computation

Modelling and simulation

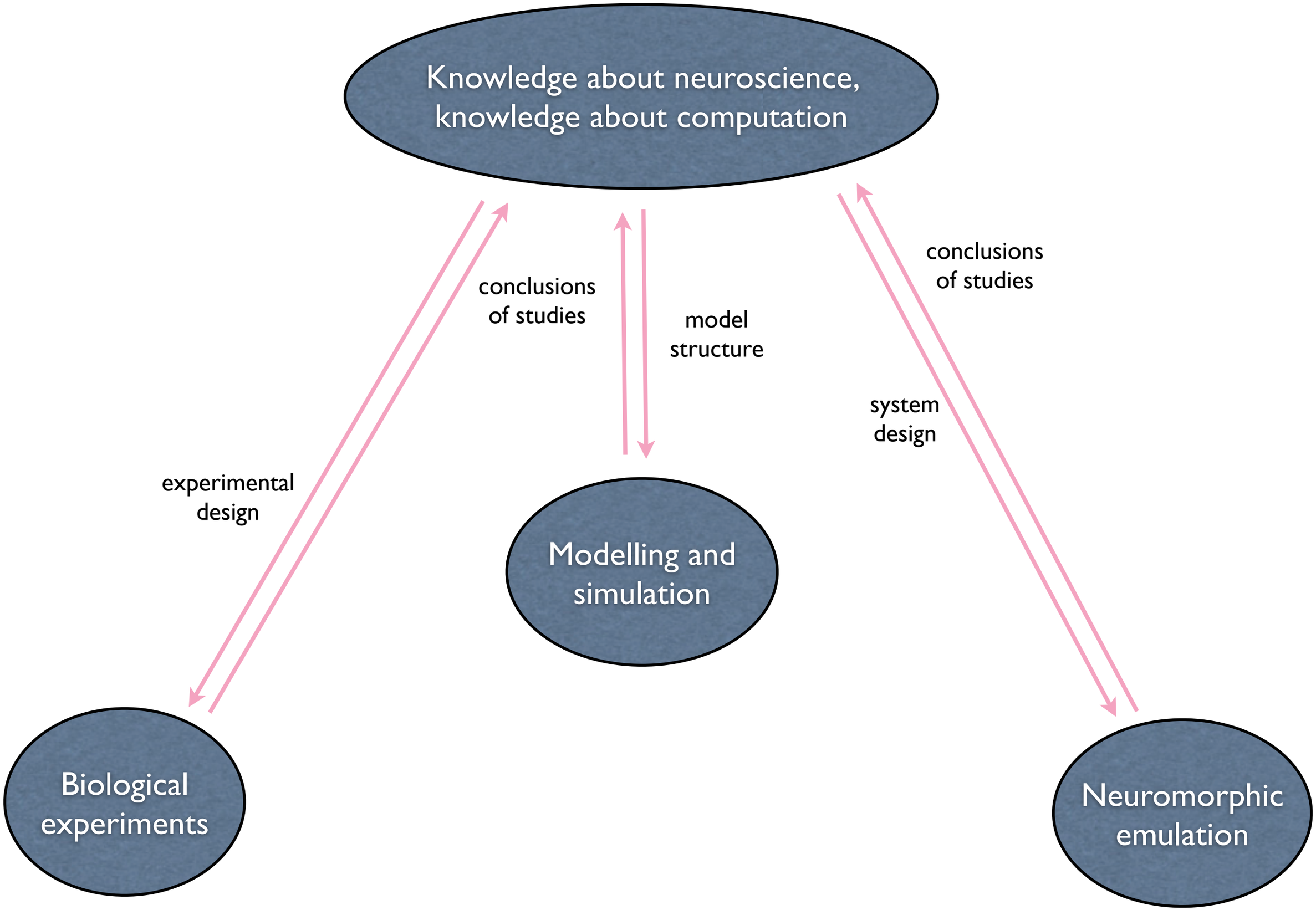
Biological experiments

Neuromorphic emulation

There are three fields represented within FACETS (and in neuromorphic computation in general). Most of the interaction between these domains takes place by exchange of knowledge, where I define knowledge as what we find in the Conclusions section of a paper. All three domains contribute to our collective knowledge about neuroscience and about computation. In turn, we use this knowledge in designing new experiments, new models, in designing new systems.

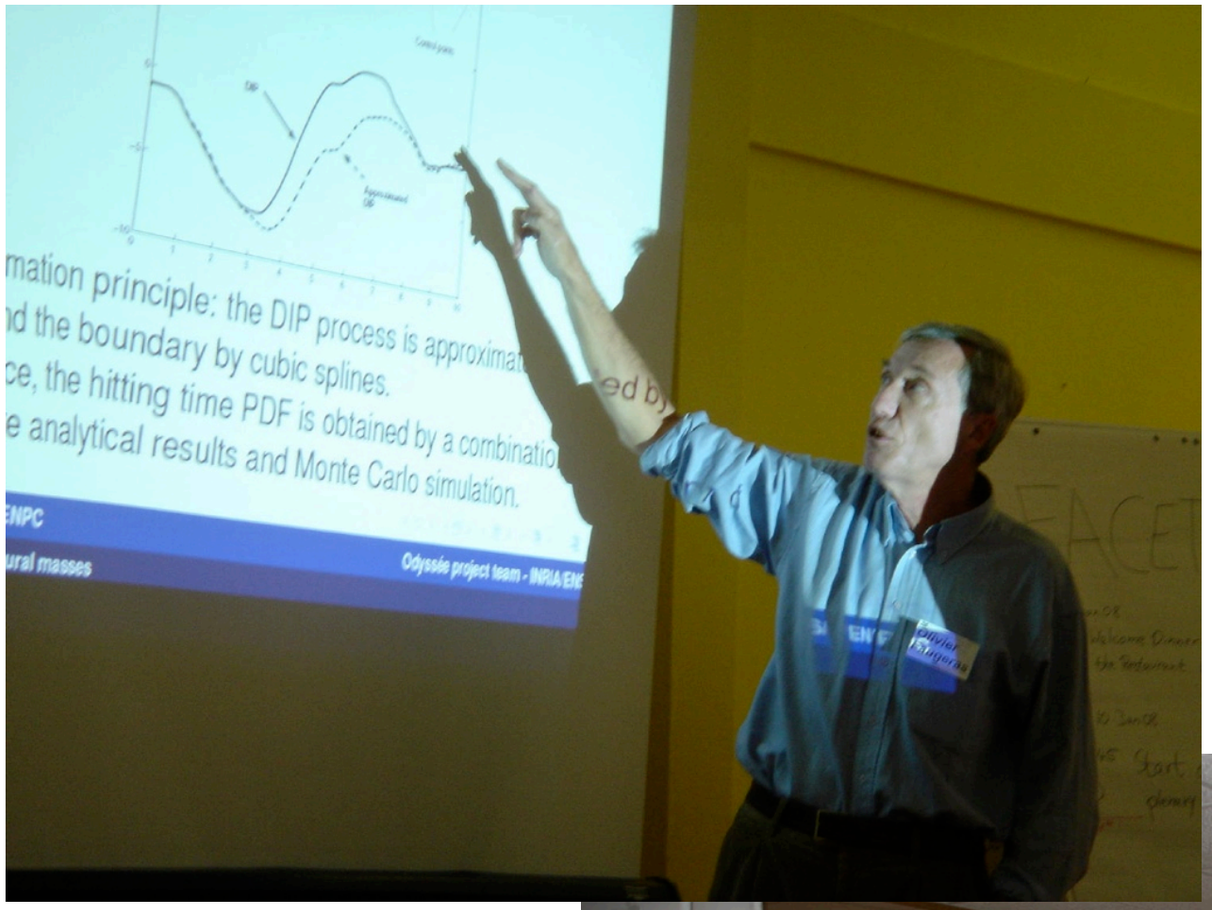


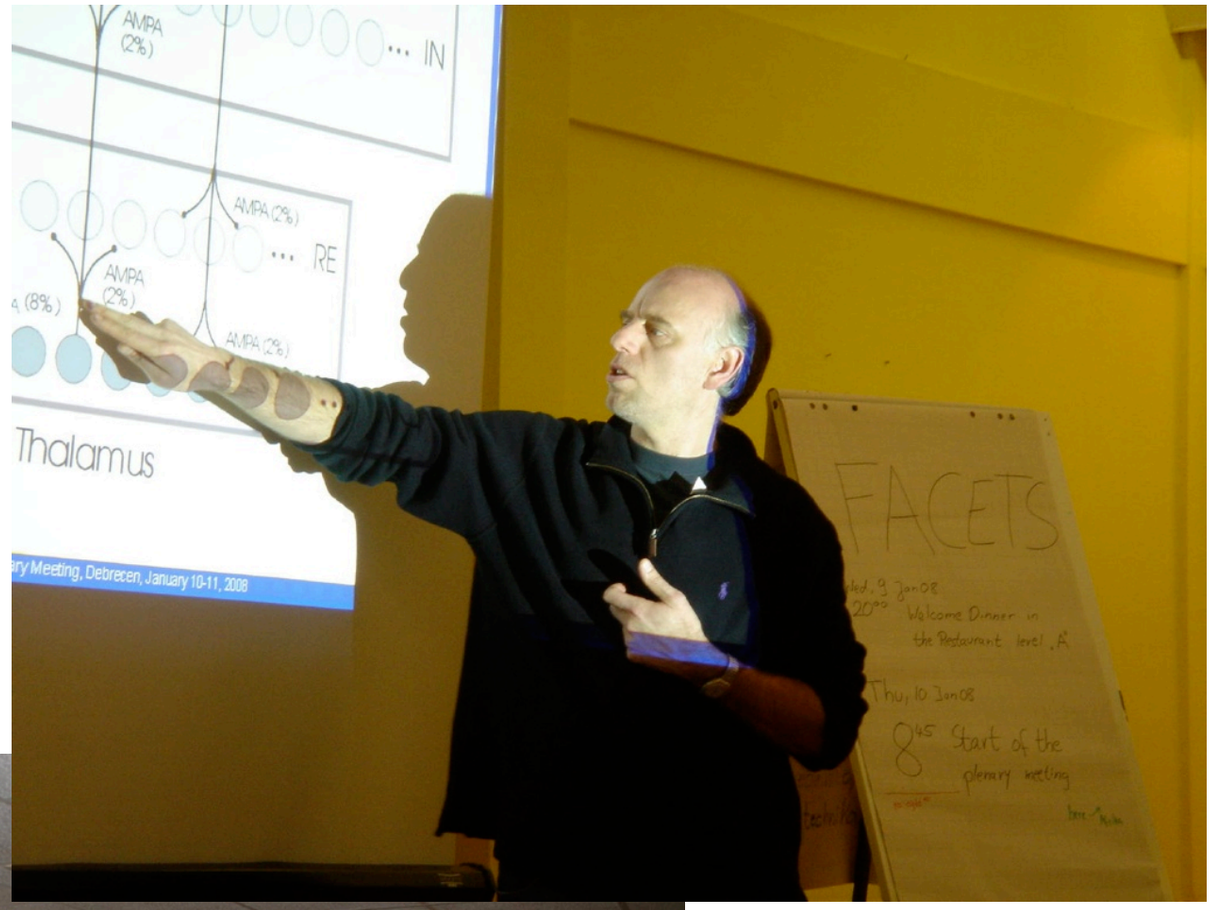
There are three fields represented within FACETS (and in neuromorphic computation in general). Most of the interaction between these domains takes place by exchange of knowledge, where I define knowledge as what we find in the Conclusions section of a paper. All three domains contribute to our collective knowledge about neuroscience and about computation. In turn, we use this knowledge in designing new experiments, new models, in designing new systems.



There are three fields represented within FACETS (and in neuromorphic computation in general). Most of the interaction between these domains takes place by exchange of knowledge, where I define knowledge as what we find in the Conclusions section of a paper. All three domains contribute to our collective knowledge about neuroscience and about computation. In turn, we use this knowledge in designing new experiments, new models, in designing new systems.

Knowledge flow





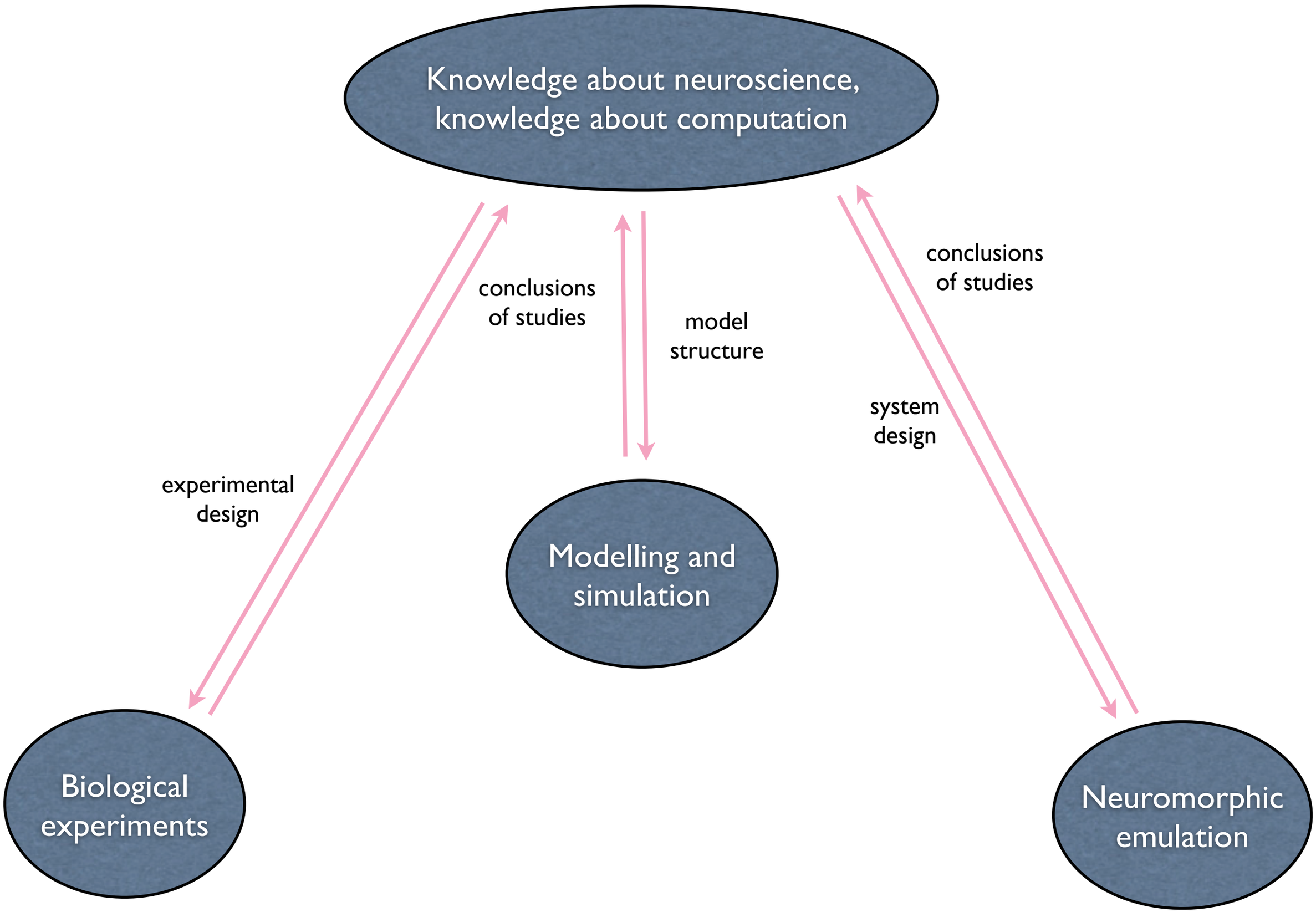
So how does this exchange of knowledge take place - of course it takes place from human-to-human, via meetings, papers, presentations. How well does the exchange work?

Knowledge flow: works fairly well*

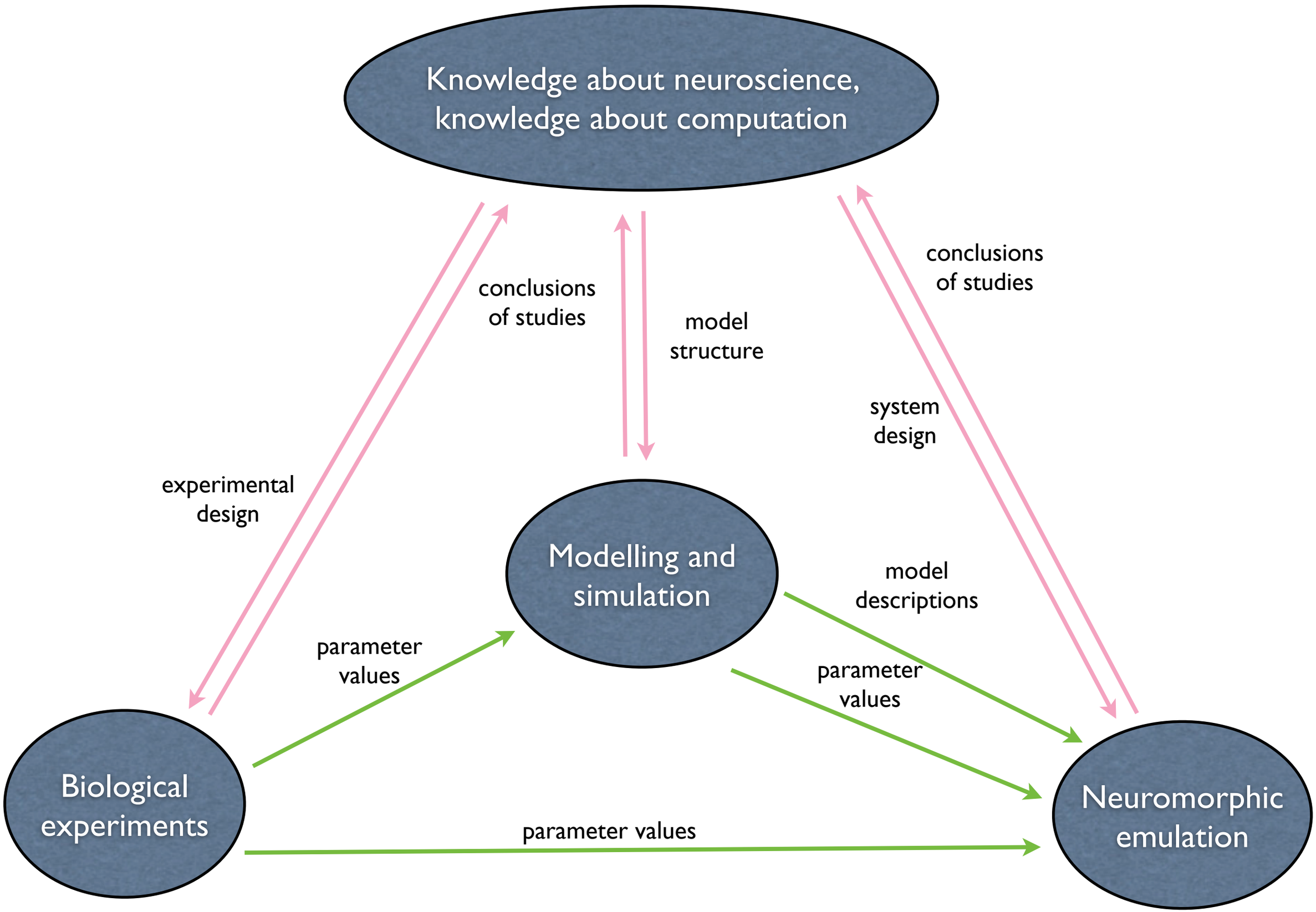
I 210 & 57 freeways by kla4067
http://www.flickr.com/photos/84263554@N00/2078587948/

* but see Science 2.0

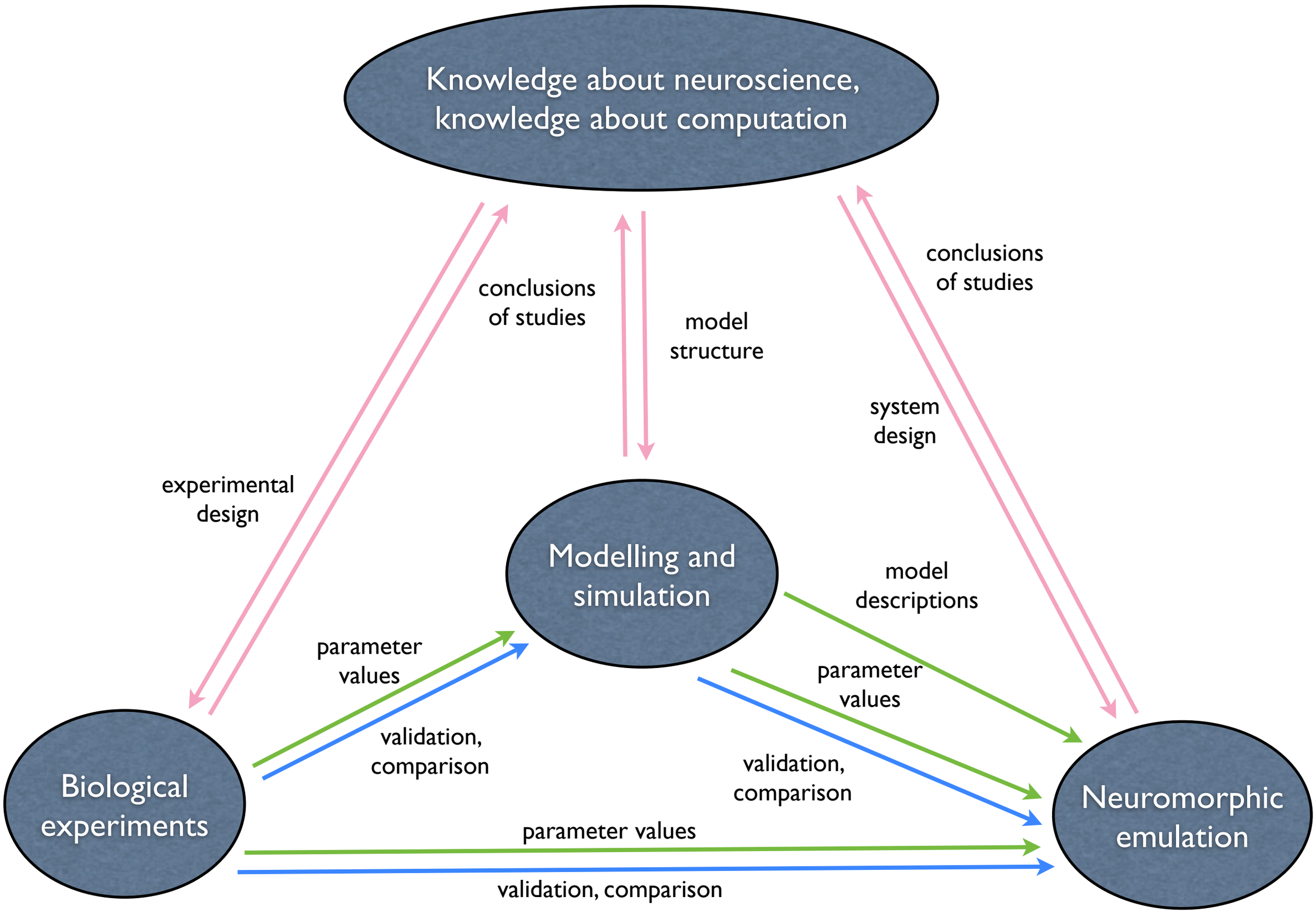
In general, it works fairly well – very well in the FACETS project – knowledge gets moved around fairly fast and effectively, even if some people think it could work a lot better (open-access publishing, semantic web, blog-based publishing, etc.)



But as well as the flow of knowledge, we also have flows of information (green lines) and of data (blue), where I define information as what goes into the Methods section of a paper: metadata might be a better word. So biological experiments can provide parameter values for models, and we can use data from experiments to test and attempt to validate our models and hardware.



But as well as the flow of knowledge, we also have flows of information (green lines) and of data (blue), where I define information as what goes into the Methods section of a paper: metadata might be a better word. So biological experiments can provide parameter values for models, and we can use data from experiments to test and attempt to validate our models and hardware.



But as well as the flow of knowledge, we also have flows of information (green lines) and of data (blue), where I define information as what goes into the Methods section of a paper: metadata might be a better word. So biological experiments can provide parameter values for models, and we can use data from experiments to test and attempt to validate our models and hardware.

Information and data flow

of the underlying V_m. We also find that the nonlinearities of the CRF are present at the level of V_m. Finally, we find that the variation in firing rates among cells can be accounted for by the different slopes of the linear relationships between V_m and the spike rate.

Materials and Methods

Surgical protocol. Experiments were conducted in accordance with the ethical guidelines of the National Institutes of Health and with the approval of the Institutional Animal Care and Use Committee of the University of Pennsylvania. Adult cats (2.5–3.5 kg) were anesthetized with an initial intraperitoneal injection of thiopental (25 mg/kg). Supplementary halothane (2–4% in a 70:30 mixture of N_2O and O_2) permitted the placement of two venous catheters. Subsequently, deep anesthesia was maintained during surgery with intravenous thiopental as needed and maintained for the duration of the experiment (14–16 hr) with a continuous infusion (3–10 mg/hr). Atropine sulfate (0.05 mg/kg, i.m.) was administered to prevent secretions and dexamethasone (4 mg, i.m.) to prevent cerebral edema. Lidocaine (2%) was generously applied to all skin incisions and pressure points. The animal was paralyzed with gallamine triethiodide (Flaxedil) by an initial injection of 60 mg and maintained with continuous intravenous infusion (20 mg/hr). The level of anesthesia was determined by continuously monitoring the EEG and the heart rate. Because the thiopental is infused continuously, we obtained very stable patterns of anesthesia throughout the experiment. The end-tidal CO_2 concentration was kept at 3.7 ± 0.2%, and the rectal temperature was kept at 37–38°C with a heating pad.

The surface of the visual cortex was exposed with a craniotomy centered at Horsley Clarke posterior 4.0, lateral 2.0 and bathed in mineral oil to prevent desiccation. The stability of the recordings was ensured by performing a bilateral pneumothorax, drainage of the cisterna magna, hip suspension, and by filling the cranial defect with a solution of 4% agar.

Visual stimulation. The corneas were protected with neutral contact lenses after dilating the pupils with 1% ophthalmic atropine and retracting the nictitating membranes with phenylephrine (Neosynephrine). Spectacle lenses were chosen by the tapetal reflection technique to optimize the focus of stimuli on the retina. The position of the monitor was adjusted with an *x–y*-stage so that the area centralae were well centered on the screen and their coordinates entered into the computer for tracking receptive field (RF) positions in retinal coordinates.

Stimuli were presented on an Image Systems (Minnetonka, MN) model M09LV monochrome monitor operating at 125 frames per second at a spatial resolution of 1024 × 786 pixels and a mean luminance of 47 cd/m^2. The screen subtends 36 by 27° (28.7 pixels per degree), and lookup tables were linearized for a contrast range of ±100%. Stimuli were synthesized using custom software by means of the framestore portion of a Cambridge Research Systems (Cambridge, UK) VSG card mounted in a conventional personal computer. Programs provide for stimulus control, online displays of acquired signals (V_m and spikes), and a graphical user interface for controlling all stimulus parameters. In addition to this online control, all data were stored on a Nicolet Vision, and it was from these records that offline analyses were performed. V_m and stimulus marks were sampled at 10 kHz with 16 bit analog-to-digital converters.

Computer-assisted hand plotting routines were used with every cell to estimate quickly and accurately the optimal orientation, direction, and spatial and temporal frequencies and to determine the receptive field position and dimensions. Contrast response functions were generated by presenting sinusoidal gratings of optimal orientation, direction, and spatial frequency, drifting within a patch limited to the receptive field. Mean luminance and all parameters of the stimuli were held constant except for

presentations at each contrast, and 5–15 passes wer
contrasts used were always 0, 2, 4, 8, 16, 32, and 64

Simple cells were distinguished from complex
modulation of their spike trains. If the fundamenta
temporal frequency of the grating) equaled or exceed
rate (the DC), the cell was classified as simple (S
otherwise it was classified as complex.

Intracellular recording procedures. Intracellular
tained from the visual cortex as close as possible to
the area centralis (P4, L2). Intracellular recordings
glass micropipettes filled with 2 M potassium acetat
otin added). The depth of the cells was estimated
reading, which was calibrated by comparing thos
depths of cells filled with Neurobiotin ($n = 12$) and
error. After beveling, pipettes had final resistances

Statistical analysis. Contrast response functions w
using MatLab (MathWorks, Natick, MA). Spike firi
mined from the Nicolet records, and PSTHs were
spike counts per bin ($n = 100$) evenly spaced ove
contrast. F1 and DC response components were ext
stimulus histograms (PSTHs) at each contrast on
Spikes were also removed from the records of V_m (
tion), and cyclegrams were generated of V_m for ea
components were extracted from the V_m cyclogram
F1 terms and seven DC terms were obtained for bo
every cell. Each set of 4 × 7 observations was fit to
tions using the Levenberg–Marquardt method to m
between the observations and the candidate functio
bines the steepest-descent method and a Taylor ser
obtain a fast, reliable technique for nonlinear opti
the lead of Albrecht and Hamilton (1982), the four
are:

Linear $R(C) = a + b*C$,
Log $R(C) = a + b*\log_{10}(C)$,
Power $R(C) = a + b*C^c$,
Hyperbolic ratio $R(C) = R_{max}*C^n/(C_{50}{}^n + C^n)$,

where $R(C)$ denotes response as a function of co
activity (or resting V_m) was subtracted from the data
The parameters of the hyperbolic ratio function
Results.

In most instances, the groups being compared ar
nonparametric statistics are used unless otherwise

Results

Our goal was to characterize quantitatively the
trophysiologically defined cell classes in prima
a function of the contrast of visual stimuli.
recording *in vivo*, we measured the responses
dal gratings of optimal orientation and spatiot
presented at logarithmically spaced contrasts
fied electrophysiologically with intracellular
and contrast response functions (CRFs) wer
and spike rates (in Hertz). The CRFs were cha
tatively by least-squares fits to four mathemat
ear, logarithmic, power, and hyperbolic ratio
Methods). The parameters of these fits were
pare the CRFs obtained simultaneously for s
and to summarize and compare the responses
classes. We emphasize the differences betwe
because they constitute the great majority of
hibitory cells in the neocortex, but we also sh

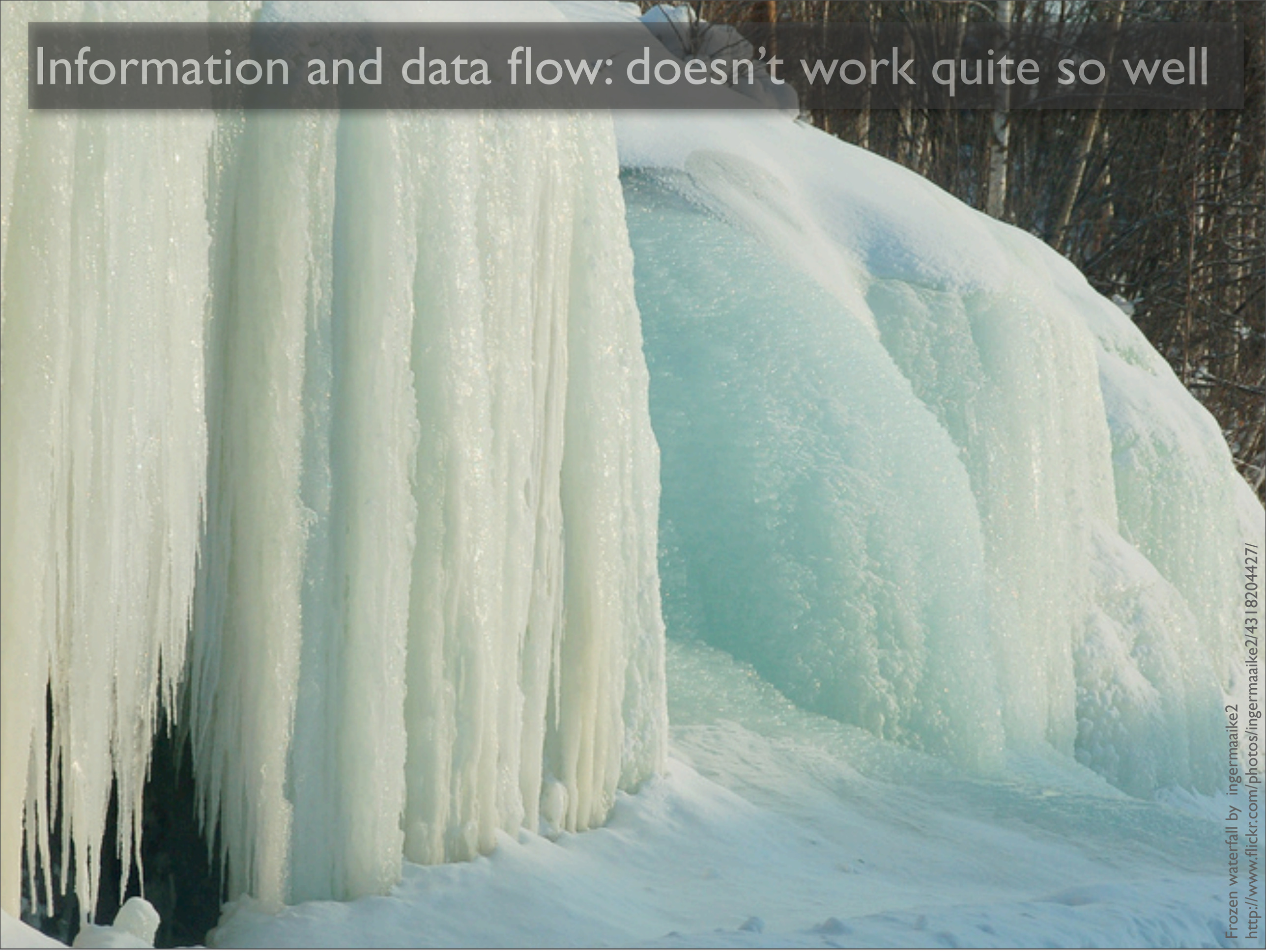
Now, at the moment, data flows between scientists mostly via zip files, e-mail attachments, DVDs. Information – metadata – mostly flows via the same route as knowledge, via PDFs.

Information and data flow: doesn't work quite so well

Horse & Cart at the Muslim Cemetery, Tetovo by themanwithsalthair
http://www.flickr.com/photos/themanwithsalthair/3038240771/

This is not so good. We have format problems, data going missing, multiple slightly different copies... (see http://www.phdcomics.com/comics.php?f=1323)

Information and data flow: doesn't work quite so well

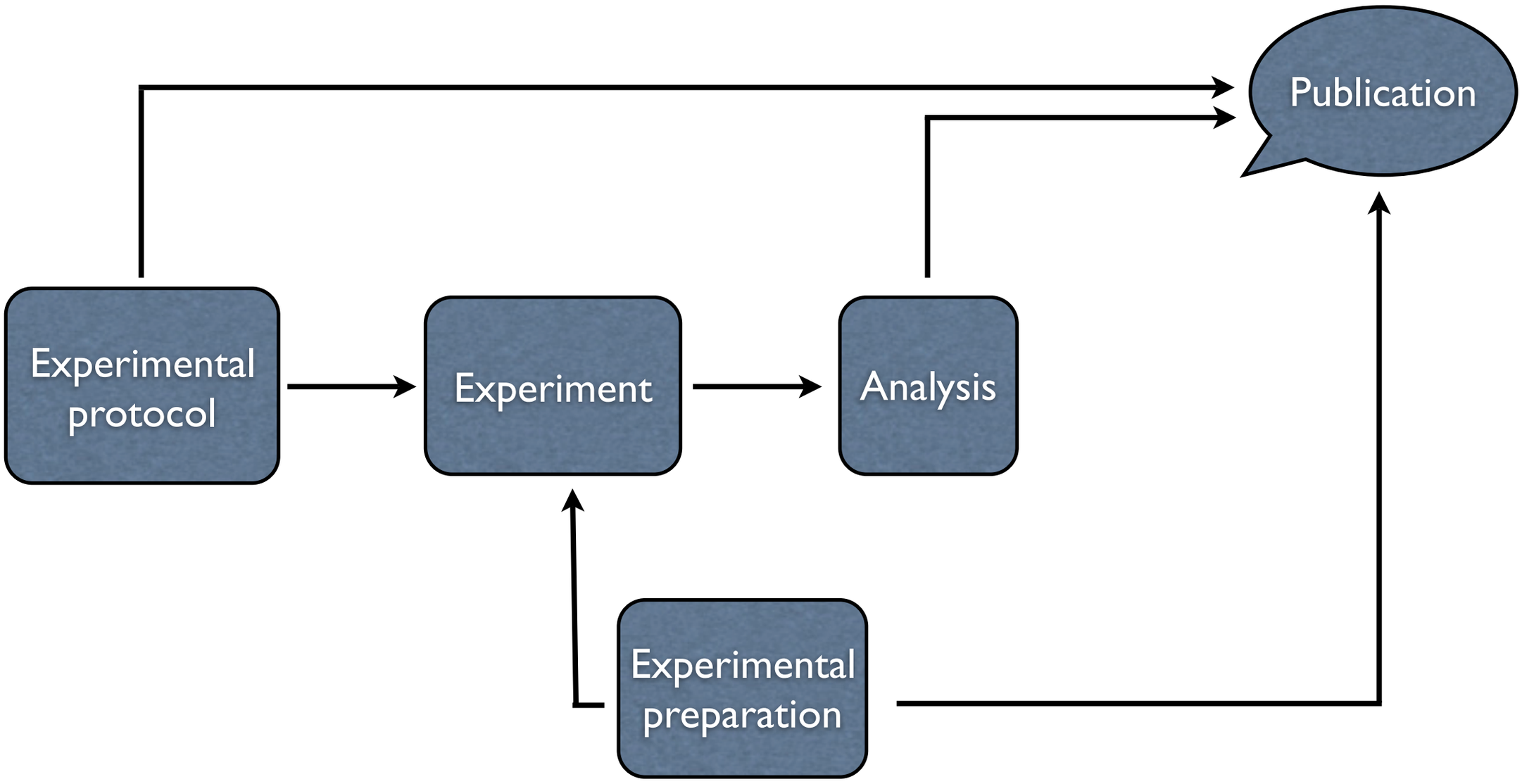


Frozen waterfall by ingermaaike2
http://www.flickr.com/photos/ingermaaike2/4318204427/

and information doesn't flow at all; Or only flows via the knowledge pathways (papers, PDFs) and with very lossy transmission.

To recap: neuromorphic computation requires effective and rapid communication of knowledge, information, and data between biologists, physicists, engineers, ... but while knowledge flow works well, there are problems with exchanging information and data.
So one of the goals of FACETS was to improve the flow of data and of information between the different groups and disciplines within the project.

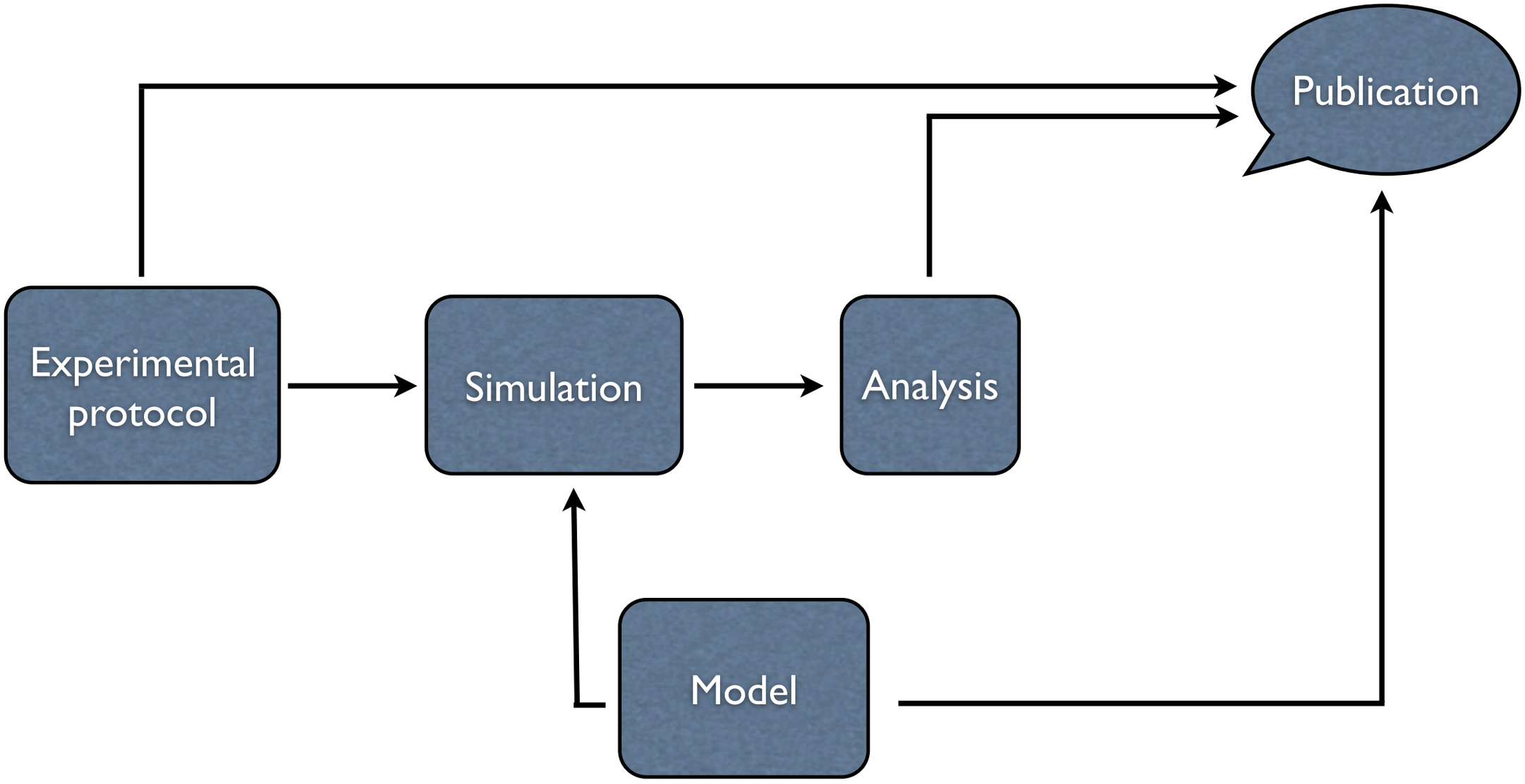
Experimental workflow



Now I'm going to present some of the tools we developed for improving data/information flow in FACETS.
To be more precise about the types of data and information we're interested in, I want to consider some typical workflows in FACETS.

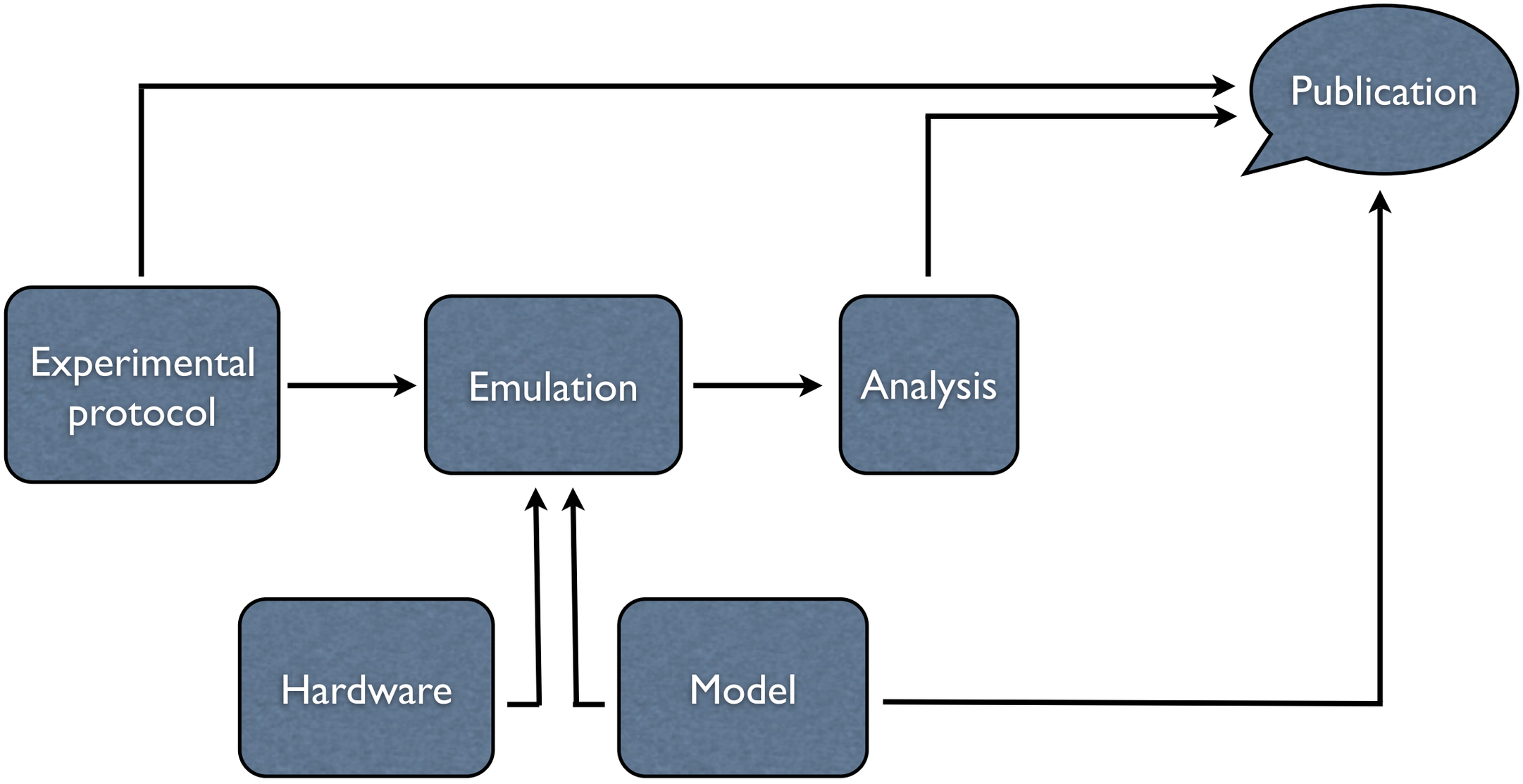
The workflows for biological experiments, ...

Software simulation workflow

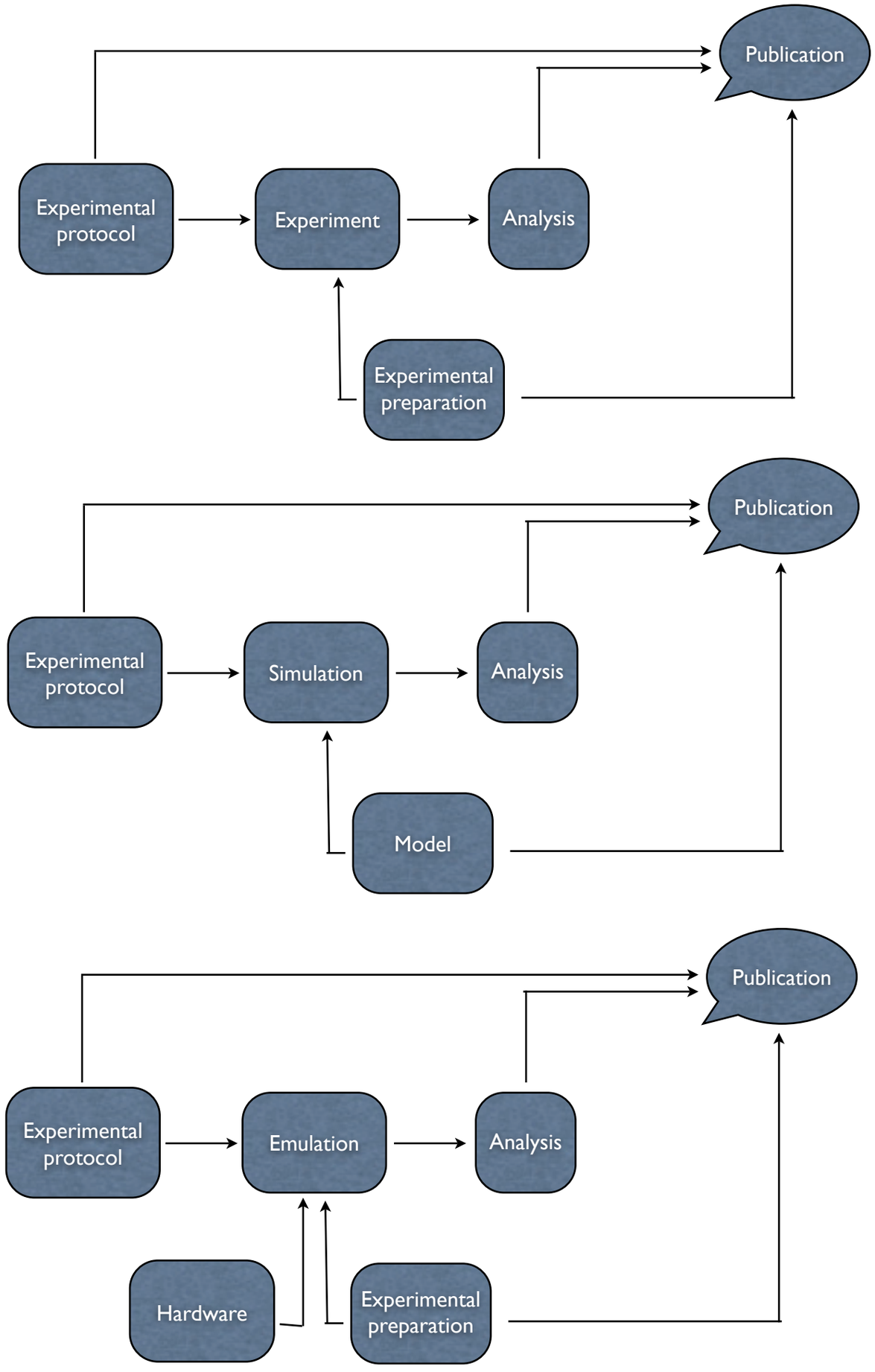


... for software simulations, ...

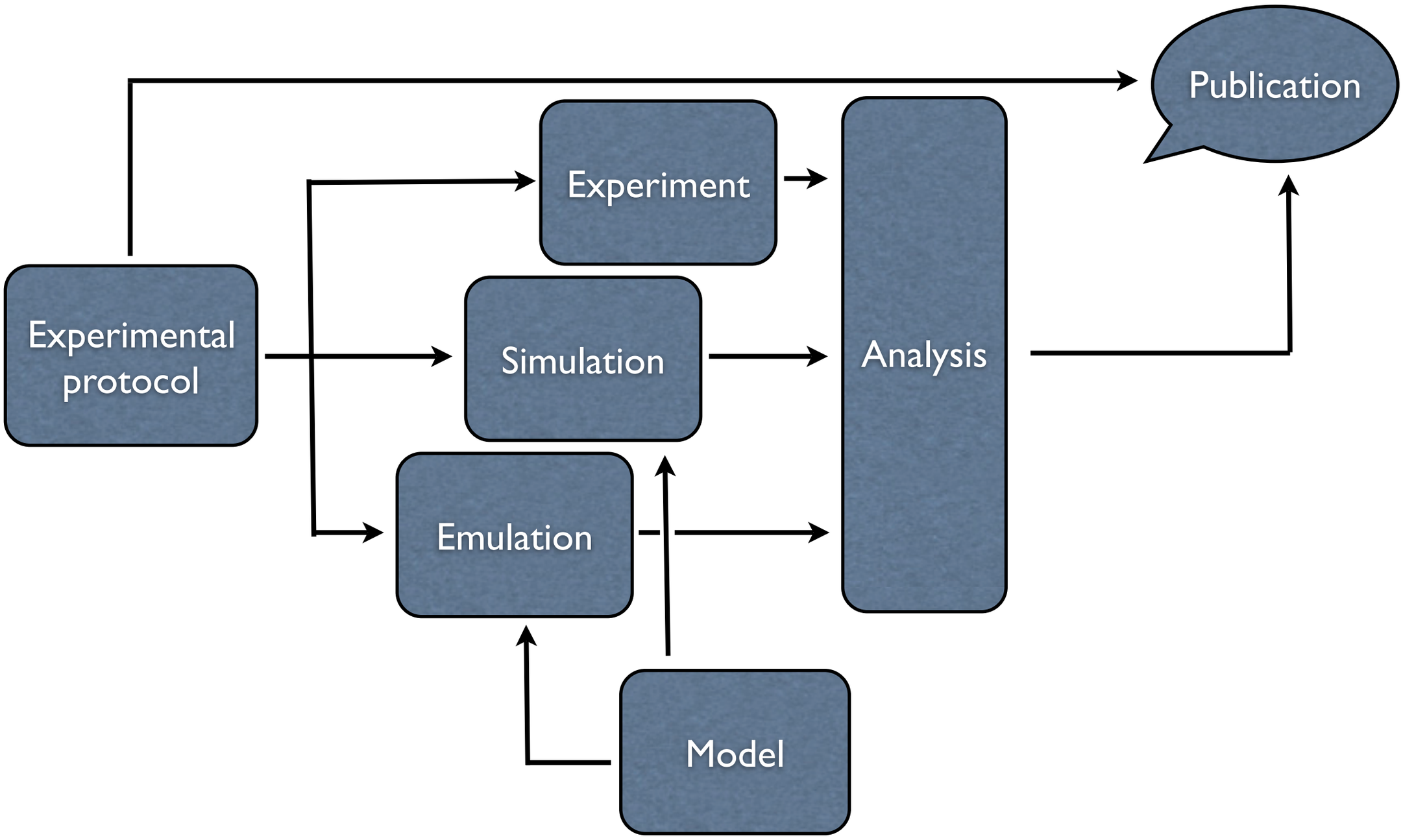
Hardware emulation workflow



... and for running experiments on the hardware (as Daniel Brüderle showed yesterday) are all very similar.

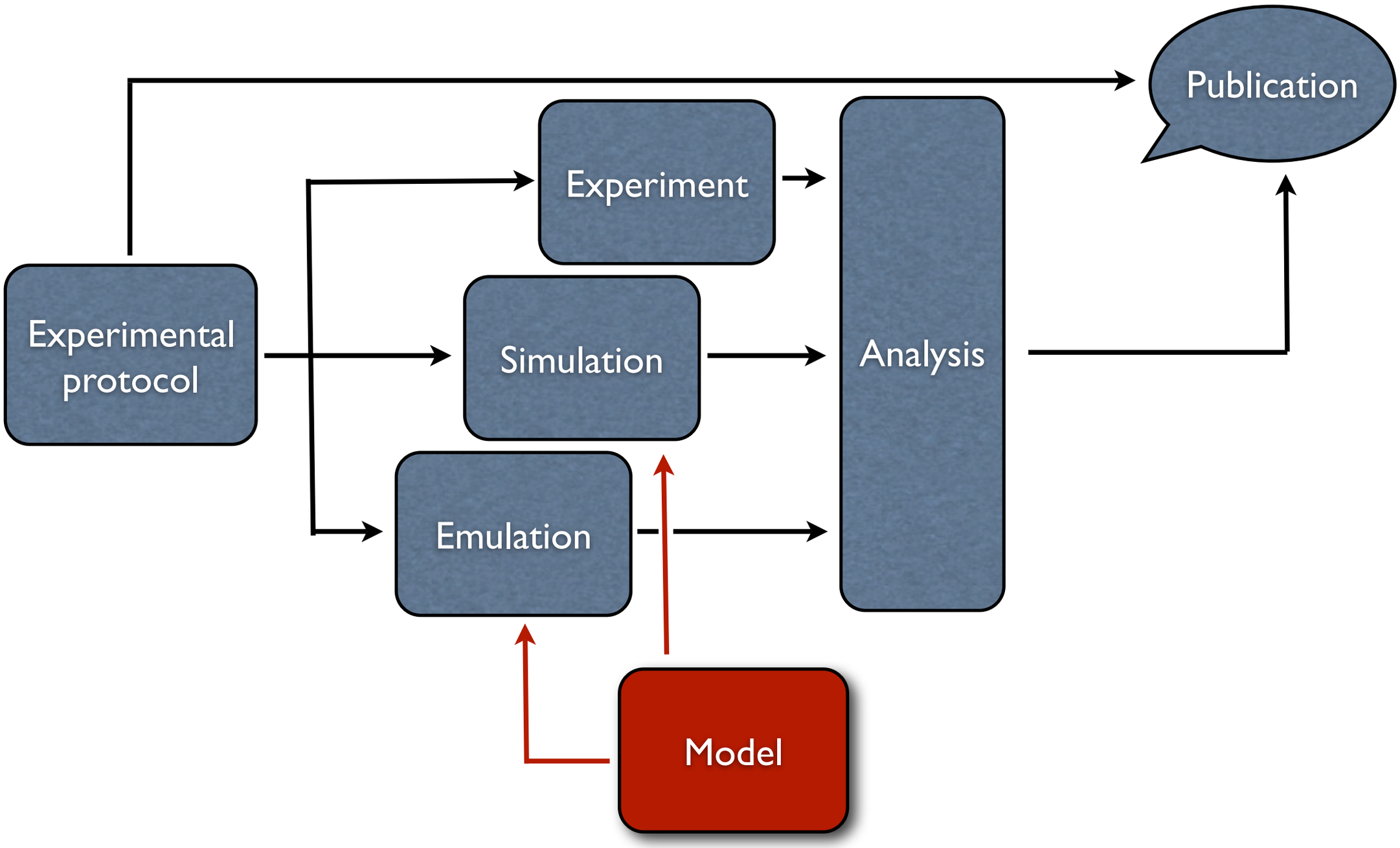


So there are evident redundancies here, and if we wish to compare experiments, software simulations and hardware emulations, there are several barriers.



One of our goals therefore is to merge these workflows, to be able to apply the same experimental protocol to biology, simulations and hardware, use the same model for both software and hardware, use the same analyses across all three types of experiment.

A common interface for model descriptions



The first part of this is to be able to use the same model for both software simulations and with neuromorphic hardware

In neuroscience, models often live in a walled garden

- not reproducible from published descriptions
- only run on a single simulator
- hence not testable or reusable

Walled garden, Palazzo Medici-Riccardi by Robert Scarth
http://www.flickr.com/photos/robert_scarth/1384386471/

The broader context for this is that there are many problems with sharing numerical models in neuroscience, even between software simulators, never mind between software and hardware.

Solution 1: improve published descriptions

- improve the papers:
  - E. Nordlie and H. E. Plesser. Visualizing neuronal network connectivity with connectivity pattern tables. Front. Neuroinform., 3:39, 2010. DOI 10.3389/neuro.11.039.2009.
  - E. Nordlie, M.-O. Gewaltig, and H. E. Plesser. Towards reproducible descriptions of neuronal network models. PLoS Comput Biol, 5(8):e1000456, Aug 2009. DOI 10.1371/journal.pcbi.1000456.
- publish to a database:
  - machine-readable, declarative descriptions
  - widely used in systems biology (SBML, CellML, SED-ML, BioModels database)
  - preliminary attempts in neuroscience: NeuroML, NineML.

Walled garden, Palazzo Medici-Riccardi by Robert Scarth
http://www.flickr.com/photos/robert_scarth/138438647/

There are 2 ways to improve this situation, and we need both.

One way is to improve the published descriptions.

Solution 2: get the code

SenseLab ModelDB NeuronDB SimToolDB

ModelDB provides an accessible location for storing and efficiently retrieving computational neuroscience models. ModelDB is tightly coupled with NeuronDB. Models can be coded in any language for any environment. Model code can be viewed before downloading and browsers can be set to auto-launch the models. About model sharing in general and ModelDB in particular.

Submit a new model entry | Model entry tutorial | Help

Find models by
- model name
- first author
- each author
- Region(circuits)

Find models for
- Cell type
- Current
- Receptor
- Transmitters
- Topic
- Simulators
- Methods

Find models of
- Networks
- Neurons

Synapses
- Electrical synapses (gap junctions)
- Chemical synapses

- Ion channels
- Neuromuscular junctions
- Axons

Search for models by author name or accession number [Search]

Search for SenseLab models using Google [Search] Hints

Search for models containing specific combinations of keywords

Search for publications in ModelDB or in PubMed

Register for an account
Login to access your models

Walled garden, Palazzo Medici-Riccardi by Robert Scarth
http://www.flickr.com/photos/robert_scarth/138438647/

A more pragmatic, and much faster solution is just to get hold of the code, if you can. Fortunately, things have been improving in this area, for example with the ModelDB database at SenseLab (http://senselab.med.yale.edu/modeldb) ...

Solution 2: get the code...and then translate it

SenseLab

ModelDB

Find Models by Simulation Environment

Click on a link to show a list of models implemented in that simulation environment or programming language.

Simulation Environment	Homepage	Number of models
BioPAX (web link to model)		1
Brian		4
C or C++ program		34
C or C++ program (web link to model)		19
CONTENT		1
CSIM		1
CSIM (web link to model)		3
CalC Calcium Calculator		1
CalC Calcium Calculator (web link to model)		7
Catacomb (web link to model)		1
CellExcite (web link to model)		1
CellML		0
CellML (web link to model)		1
Chemesis		2
Dynamics Solver		1
Emergent/PDP++		3
FORTRAN		4
FORTRAN (web link to a model)		1
GNUstep NeXTStep/OpenStep		1
Genesis		13
Genesis (web link to model)		7

Walled garden, Palazzo Medici-Riccardi by Robert Scarth
http://www.flickr.com/photos/robert_scarth/138438647/

... although even then, if you want to solve a model on a different simulator to the original one, or combine two models developed for different simulators, you're in for a tedious and difficult translation task.

Simulator diversity: problem and opportunity

Cons

- Considerable difficulty in translating models from one simulator to another...
- ...or even in understanding someone else's code.
- This:
  - impedes communication between investigators,
  - makes it harder to reproduce other people's work,
  - makes it harder to build on other people's work.

Pros

- Each simulator has a different balance between efficiency, flexibility, scalability and user-friendliness → can choose the most appropriate for a given problem.
- Any given simulator is likely to have bugs and hidden assumptions, which will be revealed by cross-checking results between different simulators → greater confidence in correctness of results.

There large number of simulators that are used in computational neuroscience is both a problem and an opportunity, and the same can be said for the diversity of approaches to developing neuromorphic hardware.

Having your cake and eating it

Simulator-independent environments for developing neuroscience models:

- keep the advantages of having multiple simulators or hardware devices
- but remove the translation barrier.

PyNN

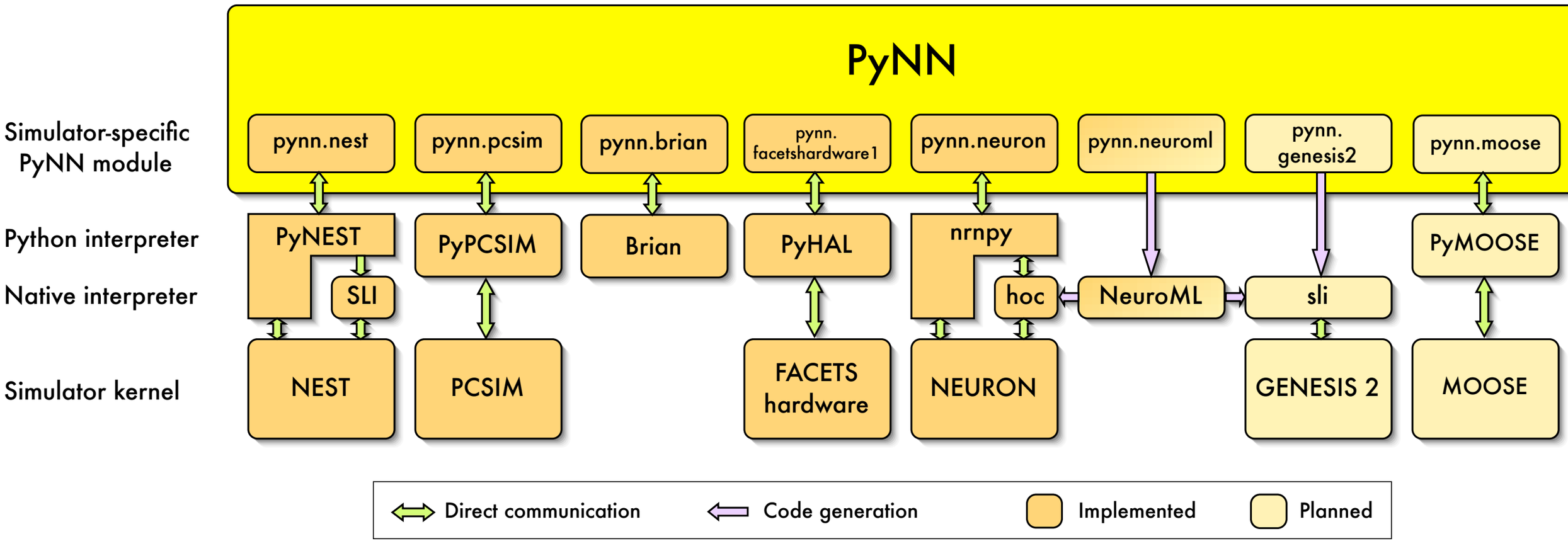
Three (complementary) approaches:

- GUI (e.g. neuroConstruct)
- XML-based language (e.g. NeuroML)
- interpreted language (e.g. Python)

Cawan Cake by Nono Fara http://www.flickr.com/photos/n-o-n-o/3280580620/

So can we keep the pros and get rid of the cons? The third approach listed on the slide is perhaps the most flexible, and is the one we've taken in FACETS

PyNN: write the code for a simulation once, run it on any supported simulator or hardware device *without modification*.



http://neuralensemble.org/PyNN

PyNN is both a definition of a common Python API for spiking network simulations and an implementation of that API for a number of commonly-used simulators.

```
sim.setup(timestep=0.1)
cell_parameters = {”tau_m”: 12.0, ”cm”: 0.8, ”v_thresh”: -50.0,
                   ”v_reset”: -65.0}
pE = sim.Population((100,100), sim.IF_cond_exp, cell_parameters,
                    label=”excitatory neurons”)
pI = sim.Population((50,50), sim.IF_cond_exp, cell_parameters,
                    label=”inhibitory neurons”)
input = sim.Population(100, sim.SpikeSourcePoisson)
rate_distr = random.RandomDistribution(”normal”, (10.0, 2.0))
input.rset(”rate”, rate_distr)
background = sim.NoisyCurrentSource(mean=0.1, stdev=0.01)
pE.inject(background)
pI.inject(background)
DDPC = sim.DistanceDependentProbabilityConnector
weight_distr = random.RandomDistribution(”uniform”, (0.0, 0.1))
connector = DDPC(”exp(-d**2/400.0)”, weights=weight_distr,
                 delays=”0.5+0.01d”)
TMM = sim. TsodyksMarkramMechanism
depressing = sim.DynamicSynapse(fast=TMM(U=0.5,tau_rec=800.0))
e2e = sim.Projection(pE, pE, connector, target=”excitatory”,
                     synapse_dynamics=plasticity)
e2i = sim.Projection(pE, pI, connector, target=”excitatory”)
i2e = sim.Projection(pI, pE, connector, target=”inhibitory”)
```


Here is an example of a PyNN script for a fairly simple network, with excitatory and inhibitory neurons connected through dynamic synapses, with a Gaussian connectivity profile, receiving both Poisson spiking input and noisy current injection.

```
import pyNN.neuron as sim
sim.setup(timestep=0.1)
cell_parameters = {”tau_m”: 12.0, ”cm”: 0.8, ”v_thresh”: -50.0,
                   ”v_reset”: -65.0}
pE = sim.Population((100,100), sim.IF_cond_exp, cell_parameters,
                    label=”excitatory neurons”)
pI = sim.Population((50,50), sim.IF_cond_exp, cell_parameters,
                    label=”inhibitory neurons”)
input = sim.Population(100, sim.SpikeSourcePoisson)
rate_distr = random.RandomDistribution(”normal”, (10.0, 2.0))
input.rset(”rate”, rate_distr)
background = sim.NoisyCurrentSource(mean=0.1, stdev=0.01)
pE.inject(background)
pI.inject(background)
DDPC = sim.DistanceDependentProbabilityConnector
weight_distr = random.RandomDistribution(”uniform”, (0.0, 0.1))
connector = DDPC(”exp(-d**2/400.0)”, weights=weight_distr,
                 delays=”0.5+0.01d”)
TMM = sim. TsodyksMarkramMechanism
depressing = sim.DynamicSynapse(fast=TMM(U=0.5,tau_rec=800.0))
e2e = sim.Projection(pE, pE, connector, target=”excitatory”,
                     synapse_dynamics=plasticity)
e2i = sim.Projection(pE, pI, connector, target=”excitatory”)
i2e = sim.Projection(pI, pE, connector, target=”inhibitory”)
```


This is how you run it with NEURON.

```
import pyNN.nest as sim
sim.setup(timestep=0.1)
cell_parameters = {”tau_m”: 12.0, ”cm”: 0.8, ”v_thresh”: -50.0,
                   ”v_reset”: -65.0}
pE = sim.Population((100,100), sim.IF_cond_exp, cell_parameters,
                    label=”excitatory neurons”)
pI = sim.Population((50,50), sim.IF_cond_exp, cell_parameters,
                    label=”inhibitory neurons”)
input = sim.Population(100, sim.SpikeSourcePoisson)
rate_distr = random.RandomDistribution(”normal”, (10.0, 2.0))
input.rset(”rate”, rate_distr)
background = sim.NoisyCurrentSource(mean=0.1, stdev=0.01)
pE.inject(background)
pI.inject(background)
DDPC = sim.DistanceDependentProbabilityConnector
weight_distr = random.RandomDistribution(”uniform”, (0.0, 0.1))
connector = DDPC(”exp(-d**2/400.0)”, weights=weight_distr,
                 delays=”0.5+0.01d”)
TMM = sim. TsodyksMarkramMechanism
depressing = sim.DynamicSynapse(fast=TMM(U=0.5,tau_rec=800.0))
e2e = sim.Projection(pE, pE, connector, target=”excitatory”,
                     synapse_dynamics=plasticity)
e2i = sim.Projection(pE, pI, connector, target=”excitatory”)
i2e = sim.Projection(pI, pE, connector, target=”inhibitory”)
```


This is how you run it with NEST.

```
import pyNN.hardware.facets.stage1 as sim
sim.setup(timestep=0.1)
cell_parameters = {”tau_m”: 12.0, ”cm”: 0.8, ”v_thresh”: -50.0,
                   ”v_reset”: -65.0}
pE = sim.Population((100,100), sim.IF_cond_exp, cell_parameters,
                    label=”excitatory neurons”)
pI = sim.Population((50,50), sim.IF_cond_exp, cell_parameters,
                    label=”inhibitory neurons”)
input = sim.Population(100, sim.SpikeSourcePoisson)
rate_distr = random.RandomDistribution(”normal”, (10.0, 2.0))
input.rset(”rate”, rate_distr)
background = sim.NoisyCurrentSource(mean=0.1, stdev=0.01)
pE.inject(background)
pI.inject(background)
DDPC = sim.DistanceDependentProbabilityConnector
weight_distr = random.RandomDistribution(”uniform”, (0.0, 0.1))
connector = DDPC(”exp(-d**2/400.0)”, weights=weight_distr,
                 delays=”0.5+0.01d”)
TMM = sim. TsodyksMarkramMechanism
depressing = sim.DynamicSynapse(fast=TMM(U=0.5,tau_rec=800.0))
e2e = sim.Projection(pE, pE, connector, target=”excitatory”,
                     synapse_dynamics=plasticity)
e2i = sim.Projection(pE, pI, connector, target=”excitatory”)
i2e = sim.Projection(pI, pE, connector, target=”inhibitory”)
```


This is how you run it on the FACETS neuromorphic hardware.

So once you’ve defined your model, you can choose to run it on the simulator that fits it best, and you can check that different simulators give the same result, and, as Daniel Brüderle demonstrated yesterday, you can also transfer your model trivially to neuromorphic hardware.

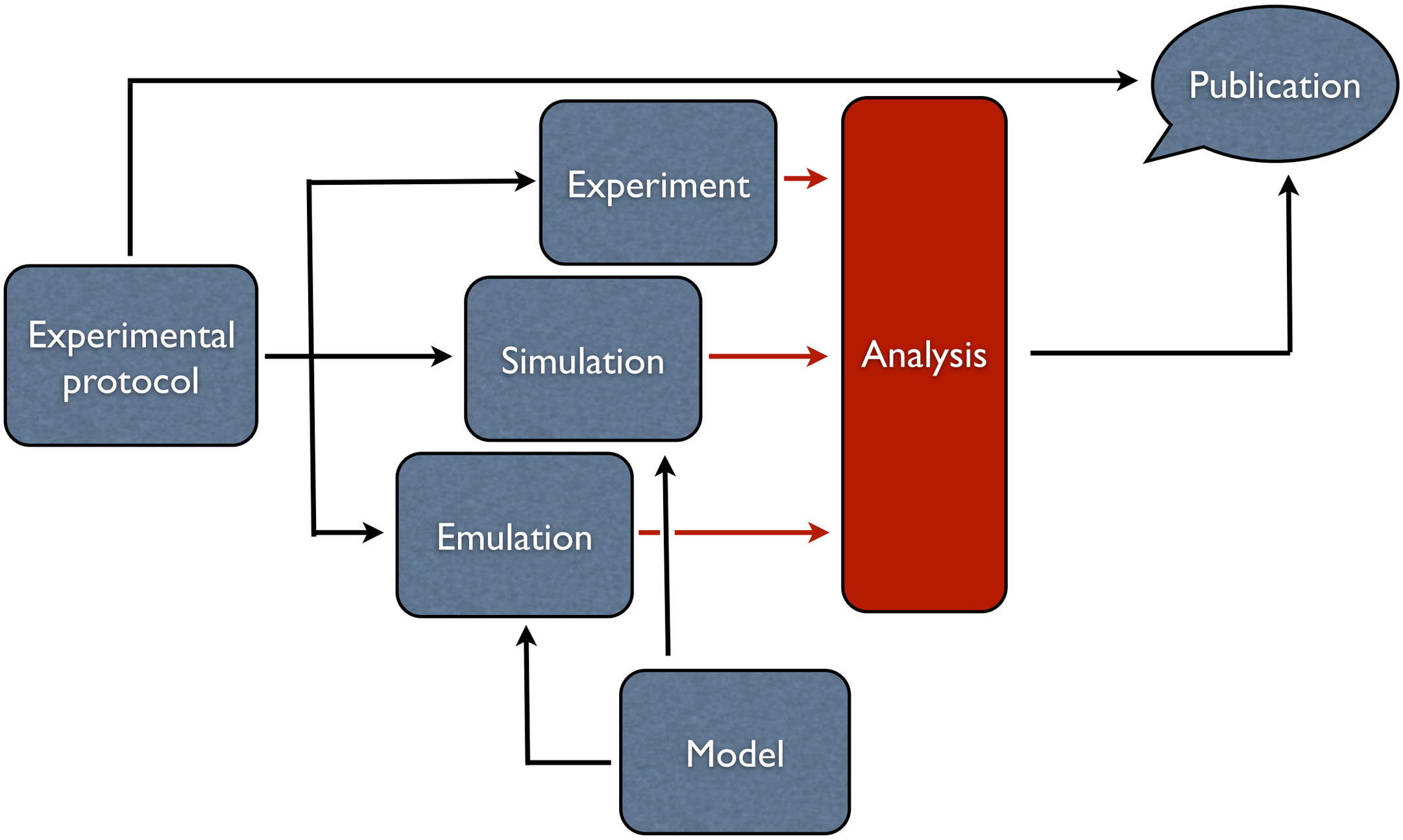
http://neuralensemble.org/PyNN

Davison A.P., Brüderle D., Eppler J.M., Kremkow, J., Muller E., Pecevski D.A., Perrinet L. and Yger P. (2009) PyNN: a common interface for neuronal network simulators. *Frontiers in Neuroinformatics* 2:11: doi:10.3389/neuro.11.011.2008

Brüderle D., Muller E., Davison A., Muller E., Schemmel J. and Meier K. (2009) Establishing a Novel Modeling Tool: A Python-based Interface for a Neuromorphic Hardware System. *Frontiers in Neuroinformatics* 3:17: doi:10.3389/neuro.11.017.2009

PyNN is open-source, and we welcome contributions. For more information, see either of these two articles. Full documentation is on the website.

A shared toolbox for analysis and visualisation



I've talked about sharing model descriptions, can we do the same for analysis and visualisation of results?

Parallel reimplementation of analysis routines by generations of PhD students...

We've talked a lot about parallelism in this meeting.....and while it is undoubtedly educational to implement your own analysis routines, it's also rather wasteful and error prone.

FIND

Finding Information in Neural Data

Data Acquisition and Generation

Analysis Tools

Individual Data-Format Import

commercial software

fixed and rigid analysis programs

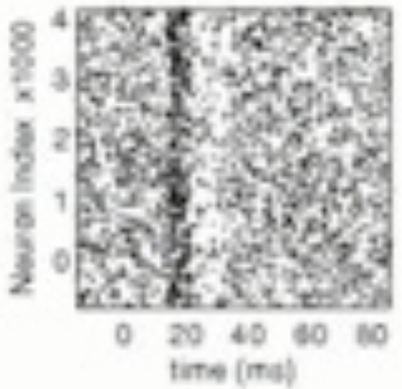
in house only tools

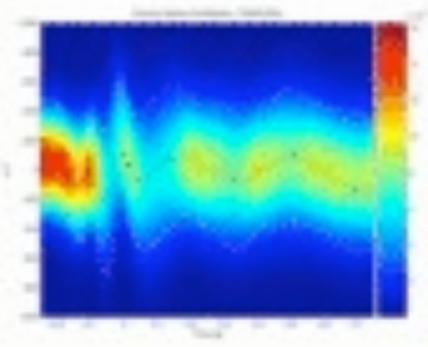
ill documented software

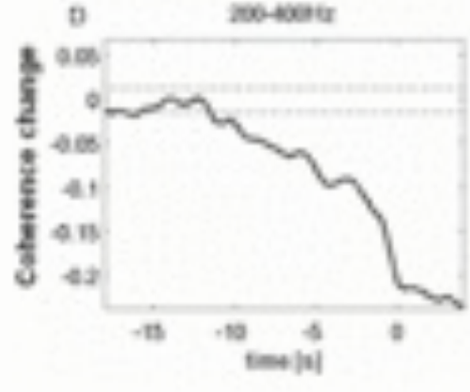
dead source code

Individual Data-Analysis

Results







Ralph Meier, BCCN Freiburg

There have been several activities in FACETS working on toolboxes for neuronal analyses. In the Matlab environment, we have FIND, developed in Freiburg. This project started before FACETS, but developed during the FACETS period.

FIND

Finding Information in Neural Data using NeuroShare

Data Acquisition and Generation

Alpha Omega

Cambridge Electronic Design

NeuroExplorer

Plexon

R.C. Electronics Inc

Tucker-Davis Technologies

Cyberkinetics Inc

Multi Channel Systems

NEST

NeuroShare

Analysis Tools

commercial software

fixed and rigid analysis programs

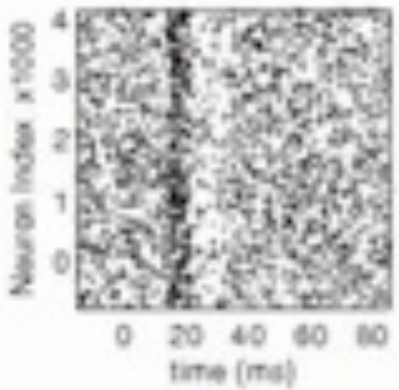
in house only tools

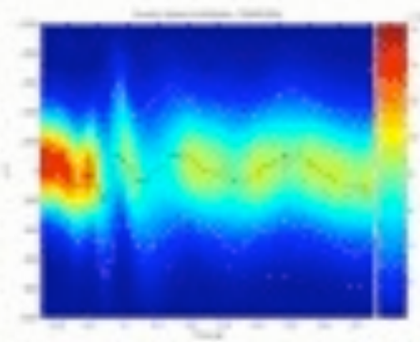
ill documented software

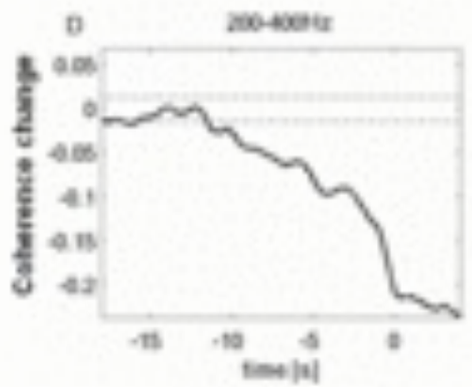
dead source code

Individual Data-Analysis

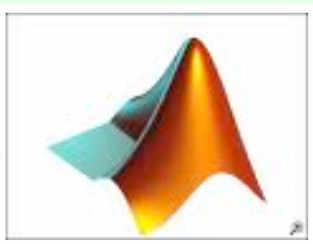
Results







Ralph Meier, BCCN Freiburg



There have been several activities in FACETS working on toolboxes for neuronal analyses. In the Matlab environment, we have FIND, developed in Freiburg. This project started before FACETS, but developed during the FACETS period.

FIND

Finding Information in Neural Data – Toolbox using NeuroShare

Data Acquisition and Generation

Alpha Omega

Cambridge Electronic Design

NeuroExplorer

Plexon

R.C. Electronics Inc

Tucker-Davis Technologies

Cyberkinetics Inc

Multi Channel Systems

NEST

NeuroShare

Analysis Tools

open source

flexible & adaptable analysis modules

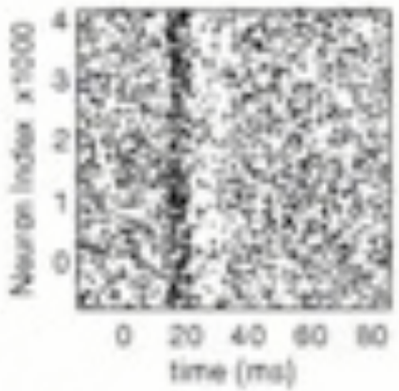
comparable across labs

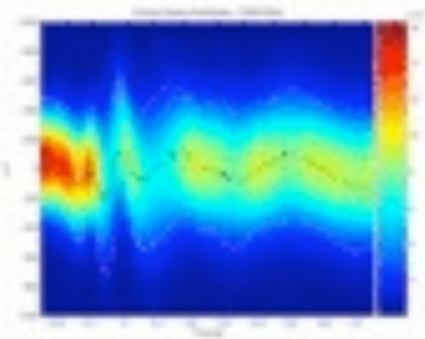
software documented by programmer

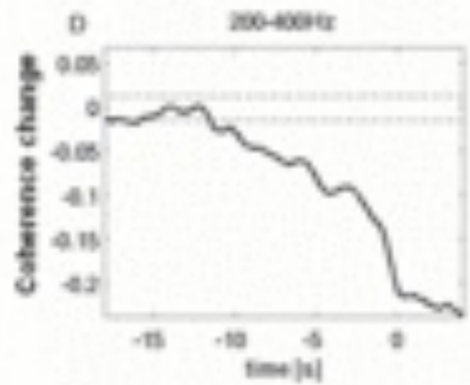
code "lives"

FIND Toolbox

Results







http://find.bccn.uni-freiburg.de

Ralph Meier, BCCN Freiburg

There have been several activities in FACETS working on toolboxes for neuronal analyses. In the Matlab environment, we have FIND, developed in Freiburg. This project started before FACETS, but developed during the FACETS period.

http://neuralensemble.org/NeuroTools

- signals: manipulation of and calculations with spike trains and analog signals.
- parameters: management of large, hierarchical parameter sets
- analysis: miscellaneous analysis functions
- stgen: various stochastic process generators relevant for Neuroscience (Ornstein-Uhlenbeck, Poisson, inhomogenous gamma, ...).
- plotting: tools for plotting and image processing, based on Matplotlib and the Python Imaging Library.
- datastore: intelligent caching of intermediate results

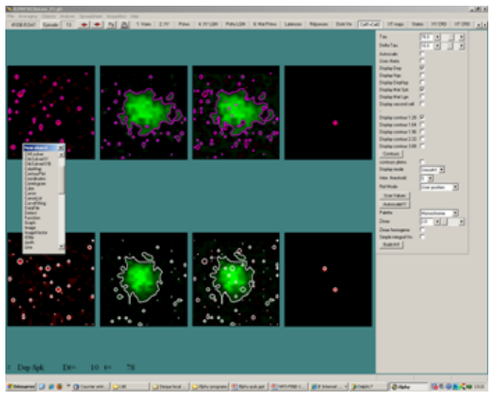
While FIND is mainly aimed at experimental data, most modellers in FACETS have tended to use Python. The simulators all use Python as well, so it is convenient to plug things together this way. NeuroTools is a toolbox for simulation projects, containing not just analysis routines, but tools for signal generation, parameter management, etc. NeuroTools is fully open-source, and accepts contributions from anyone.

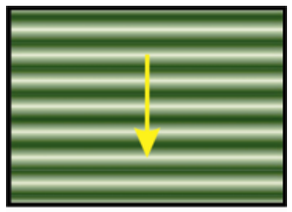
Elphy

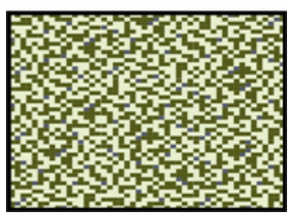
- Programmable data acquisition and analysis environment for Windows
- Used for a large fraction of biological experiments within FACETS

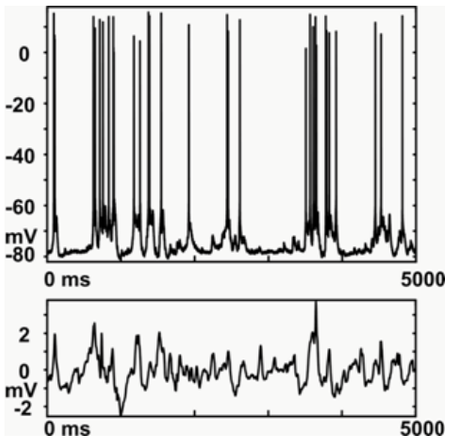
Object Oriented Programming

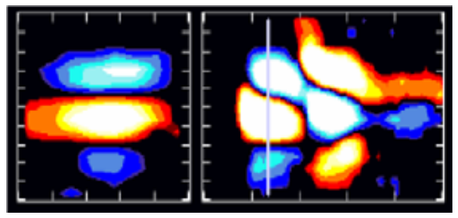
User Friendly Graphic Interface	Visual Stimulation	Data Acquisition and Control of Experiments	Data Analysis
Multi-page display Owner designed pages Mouse handling of objects	DirectX Hardware control Standard graphic adapter	Interfaces from Axon, ITC, MCC, UEI On line analysis	120 dedicated objects 2000 functions Event detection, correlations, Fourier analysis, etc…

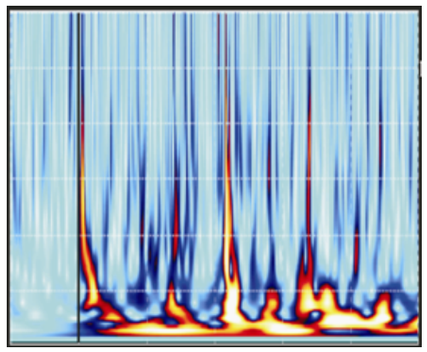


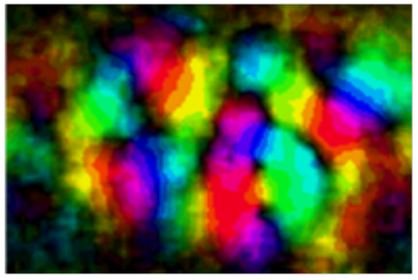










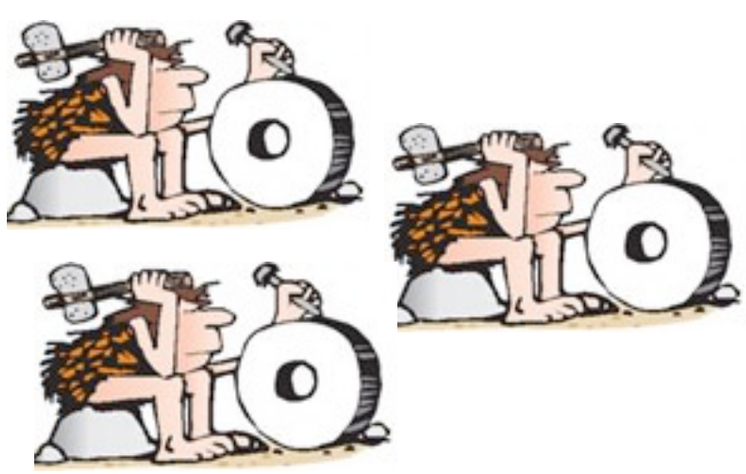


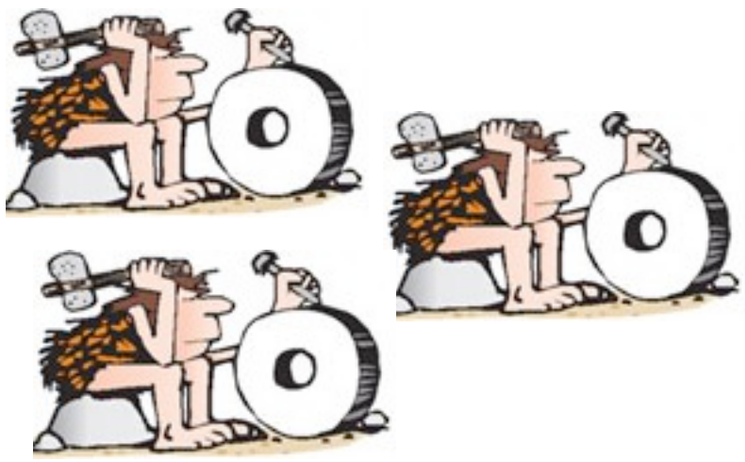
G. Sadoc
C. Monier
Team: Y. Frégnac

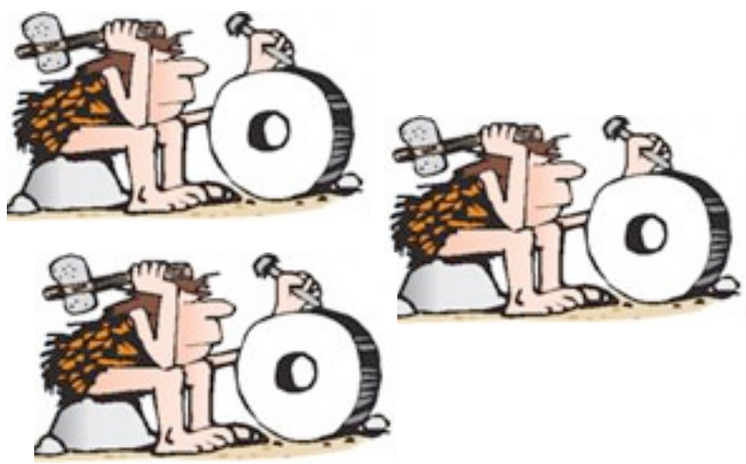
http://www.unic.cnrs-gif.fr/software.html

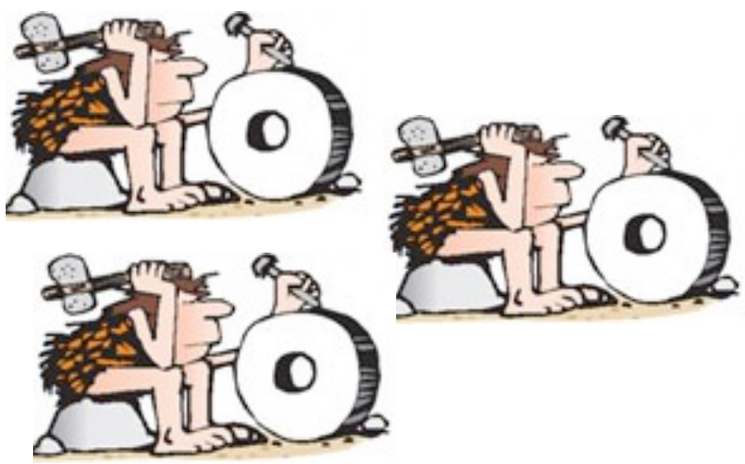
and then we have Elphy, which is a highly flexible Windows programme for data acquisition and analysis, and is developed in Gif sur Yvette by Gérard Sadoc. Elphy has been developed for many years now, and was used for a large fraction of the biological experiments within FACETS, including all the results shown by Yves Frégnac earlier.

Parallel reimplementation of analysis toolboxes by generations of EC-funded projects?

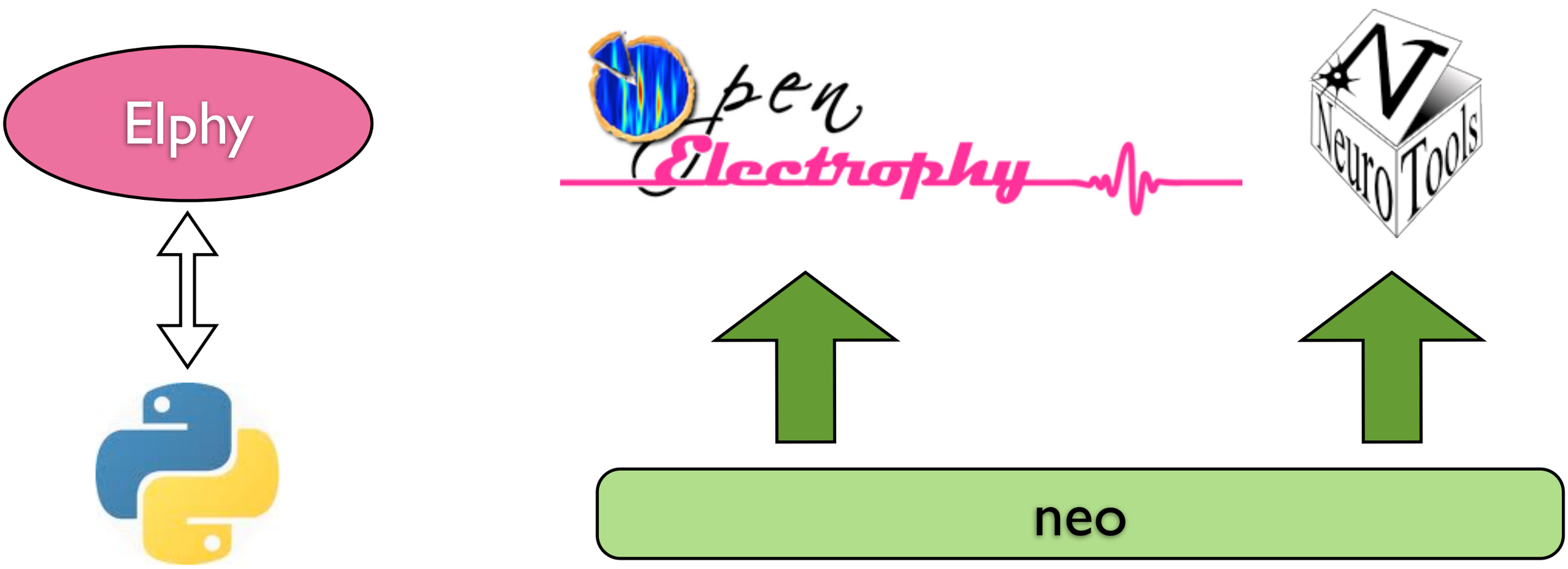






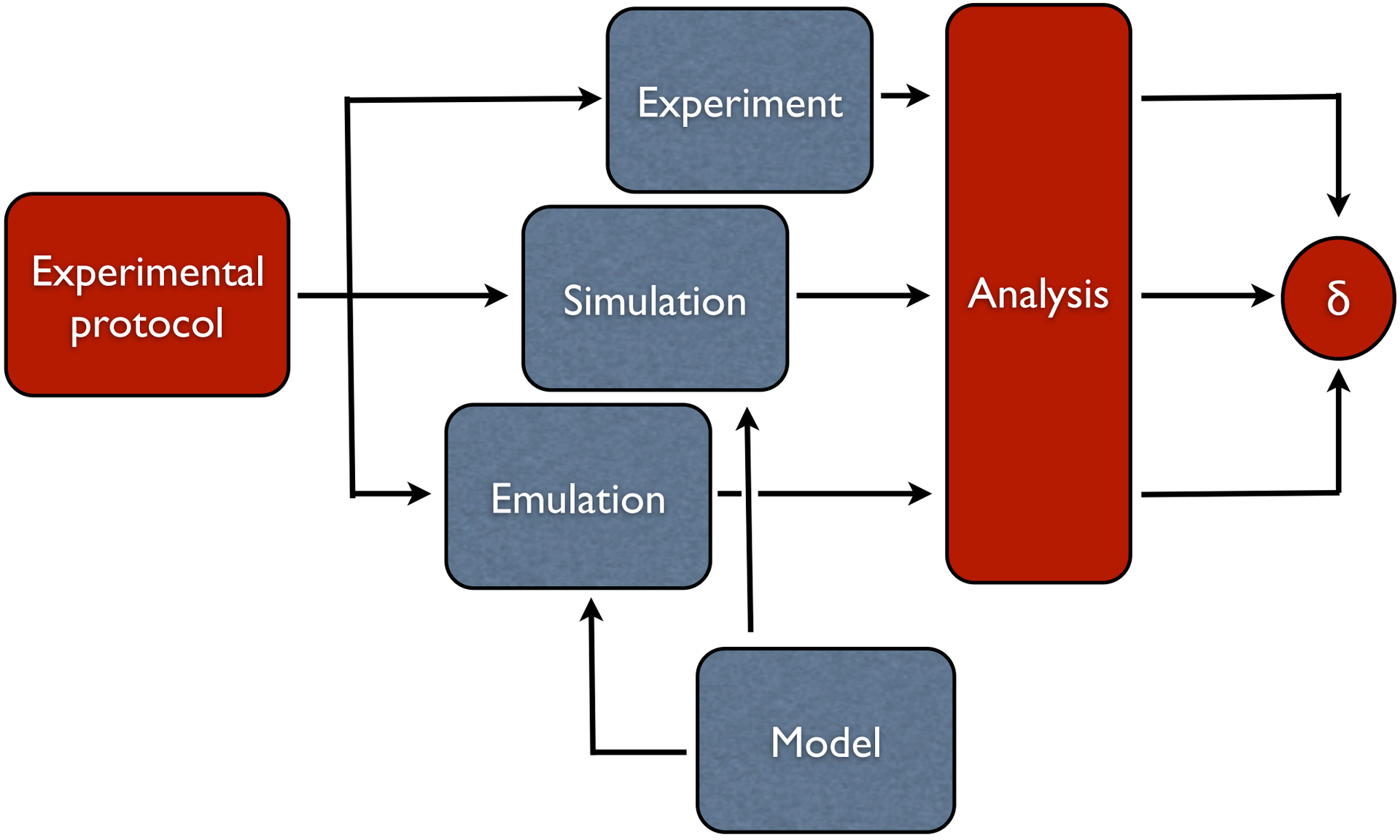


So you might argue that we haven't moved forward very much in being able to share and communicate analysis workflows.
Except...at least these guys can communicate, and find each others' bugs,



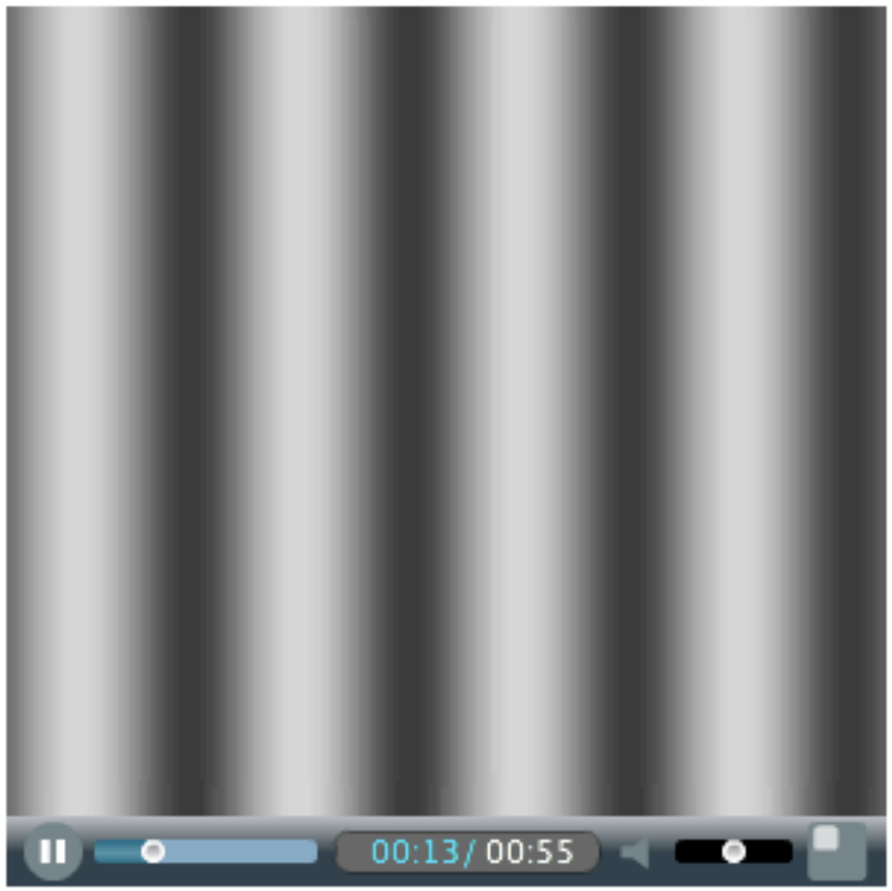
and in fact we've started to have good communication between Matlab, Python and Elphy, with the development of a Python interface to Elphy running as a server, and the merging of the data storage layers of NeuroTools and a non-FACETS tool, OpenElectrophy, developed by Samuel Garcia from the Université Claude Bernard, Lyon.

Comparing simulations to experiment



Since one of the things we want to do in FACETS is compare biological experiments with the results of simulating model systems, in software or hardware, we need to go the next step and use the same experimental protocols, and use exactly the same analysis methods for real and simulated data, and we need to automate the whole comparison process, because complex models need a lot of data to properly constrain them.

Stimulus *multiple_gratings_20080804-1034.zip at 94 cd/m2*



Reference #	4
Duration (ms)	56000.0
Frame duration (ms)	50.0
Size (pixels)	100,100
Pixels/degree	50.0
Centre (degrees)	3.0,3.0
Max luminance (cd/m²)	94.0
Background luminance (cd/m²)	47.0
Variables	contrast, orientation, spatial_freq
Created by	apdavison
Last modified	Fri 12 Dec 2008
Status	public

To achieve this automated comparison we need a standard format for specifying stimuli ...

filter-by-preferred (variable = 'orientation, spatial_freq')

merge trials ()

tuning curve (method = 'mean', variable = 'contrast')

curve fitting (curvetype = 'hyperbolic ratio', method = 'Levenberg-Marquardt', normalization = 'subtract background')

extract parameter (name = 'n')

histogram (binwidth = 0.3, maximum = 7.95, minimum = 0.15)

and we need to break analyses down into standard blocks so we can then build-up full workflows in different systems.

Encode Materials and Methods in a machine-readable format

6936 • The Journal of Neuroscience, July 30, 2003 • 23(17):6936–6945

Behavioral/Systems/Cognitive

Response to Contrast of Electrophysiologically Defined Cell Classes in Primary Visual Cortex

Diego Contreras and Larry Palmer
Department of Neuroscience, University of Pennsylvania School of Medicine, Philadelphia, Pennsylvania 19106

Information processing in the visual cortex is critically dependent on the input–output relationships of its component neurons. The transformation of synaptic inputs into spike trains depends in turn on the host of intrinsic membrane properties expressed by neurons, which define established electrophysiological cell classes in the neocortex. Here we studied, with intracellular recordings *in vivo*, how the electrophysiological cell classes in the primary visual cortex transform an increasing input, represented by stimulus contrast, into membrane depolarization and trains of action potentials. We used contrast as input because, regardless of their stimulus selectivity, primary visual cortical cells increase their firing rates in response to increases in luminance contrast. We found that both the spike rate response and the membrane potential response are best described by the hyperbolic ratio function when compared with linear, power, and logarithmic functions. In addition, both responses show similar parameter values and similar residual variance from the fits to all four functions. We also found that changes in membrane potential are similar, but firing rates differ strongly, between the established electrophysiological cell classes: fast spiking neurons show the highest firing rates, followed by fast rhythmic bursting, and regular spiking (RS) cells. In addition, among complex cells, RS cells from supragranular layers fired at higher rates than RS cells from infragranular layers. Finally, we show that the differences in firing rates between cell classes arise from differences in the slope of the relationship between membrane potential and spike rate.

Key words: **contrast; visual cortex; intrinsic properties; intracellular; *in vivo*; simple; complex**

Introduction

A critical step in understanding the operations of local cortical networks is to determine the input–output relations of its com-

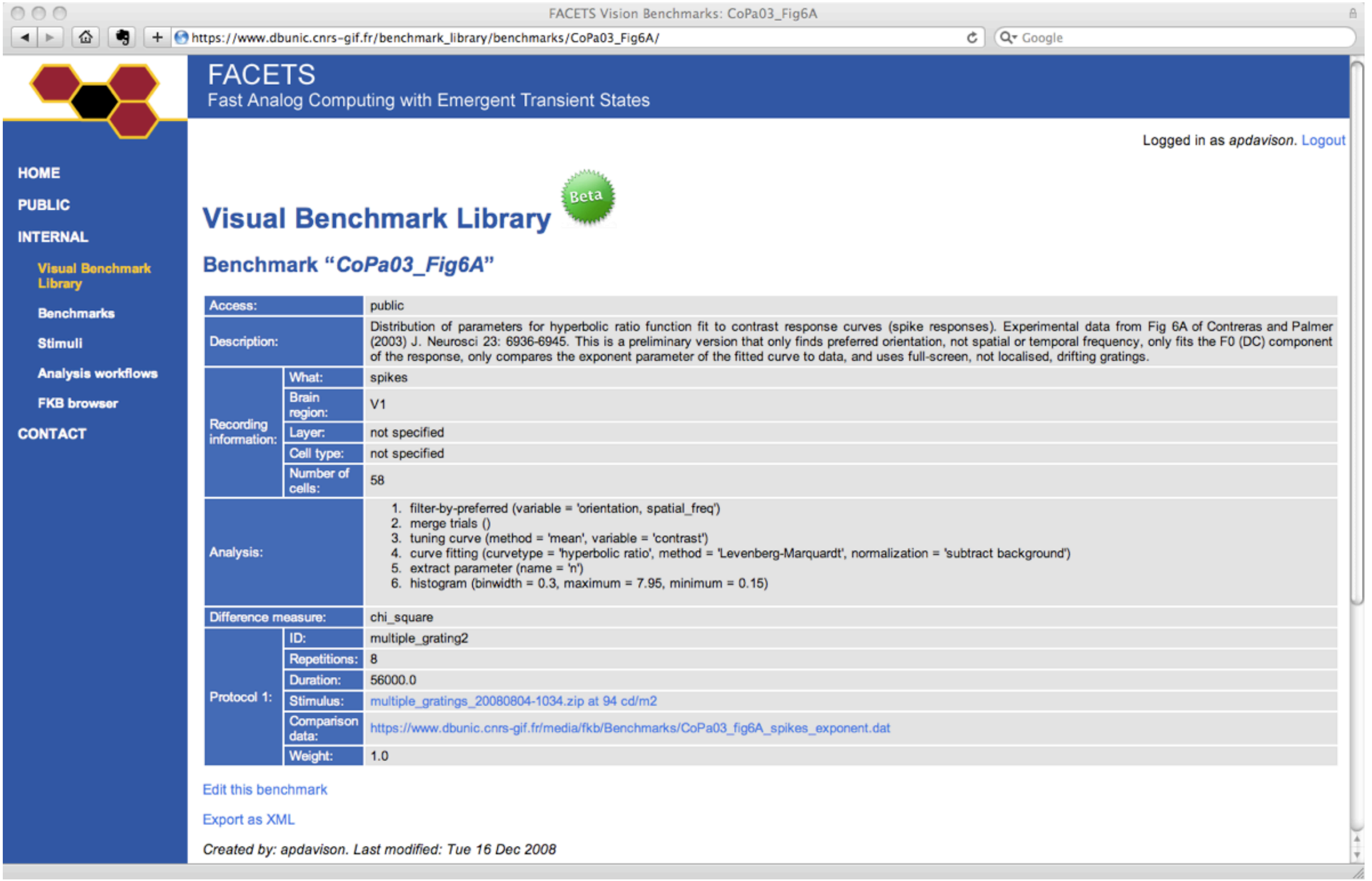
are generally GABAergic inhibitory interneurons, whereas regular spiking (RS) cells are glutamatergic excitatory cells. Therefore, for a functional understanding of cortical operations it is critical

We've started by taking experiments from the literature, and converting them into a machine-readable format, so we can automate the process of running simulations, doing the analysis and comparing to experimental data.

```
<?xml version="1.0" encoding="utf-8"?>
<benchmark id="CoPa03_Fig6A" version="n/a">
  <description>
    Distribution of parameters for hyperbolic ratio function fit to contrast response curves (spike responses). Experimental data from Fig 6A of
Contreras and Palmer (2003) J. Neurosci 23: 6936-6945. This is a preliminary version that only finds preferred orientation, not spatial or temporal
frequency, only fits the F0 (DC) component of the response, only compares the exponent parameter of the fitted curve to data, and uses full-screen,
not localised, drifting gratings.
  </description>
  <recording>
    <measureable>
      spikes
    </measureable>
    <location>
      <brain-region name="V1">
        <layer name="not specified">
          <cell-type name="not specified" number="58"/>
        </layer>
      </brain-region>
    </location>
  </recording>
  <analysis type="filter-by-preferred">
    <parameter name="variable" value="orientation, spatial_freq"/>
  </analysis>
  <analysis type="merge trials"/>
  <analysis type="tuning curve">
    <parameter name="method" value="mean"/>
    <parameter name="variable" value="contrast"/>
  </analysis>
  <analysis type="curve fitting">
    <parameter name="curvetype" value="hyperbolic ratio"/>
    <parameter name="method" value="Levenberg-Marquardt"/>
    <parameter name="normalization" value="subtract background"/>
  </analysis>
  <analysis type="extract parameter">
    <parameter name="name" value="n"/>
  </analysis>
  <analysis type="histogram">
    <parameter name="binwidth" units="" value="0.3"/>
    <parameter name="maximum" units="" value="7.95"/>
    <parameter name="minimum" units="" value="0.15"/>
  </analysis>
  <difference-measure type="χ²"/>
  <protocol duration="56000.0" id="multiple_grating2" repetitions="8" weight="1.0">
    <stimulus background-luminance="47.0" img="https://www.dbunic.cnrs-gif.fr/media/stimuli/multiple_gratings_20080804-1034.zip" max-luminance="94.0"
scale-factor="50.0" variables="contrast, orientation, spatial_freq"/>
    <comparison-data url="https://www.dbunic.cnrs-gif.fr/media/fkb/Benchmarks/CoPa03_fig6A_spikes_exponent.dat"/>
  </protocol>
</benchmark>
```


This shows our prototype XML format for describing experiments in a machine-readable format, such that the experiments can be automatically applied to different models...

But who wants to write XML?



but no-one wants to write XML, so we have a website that allows people to create these experiment representations by filling in a form.

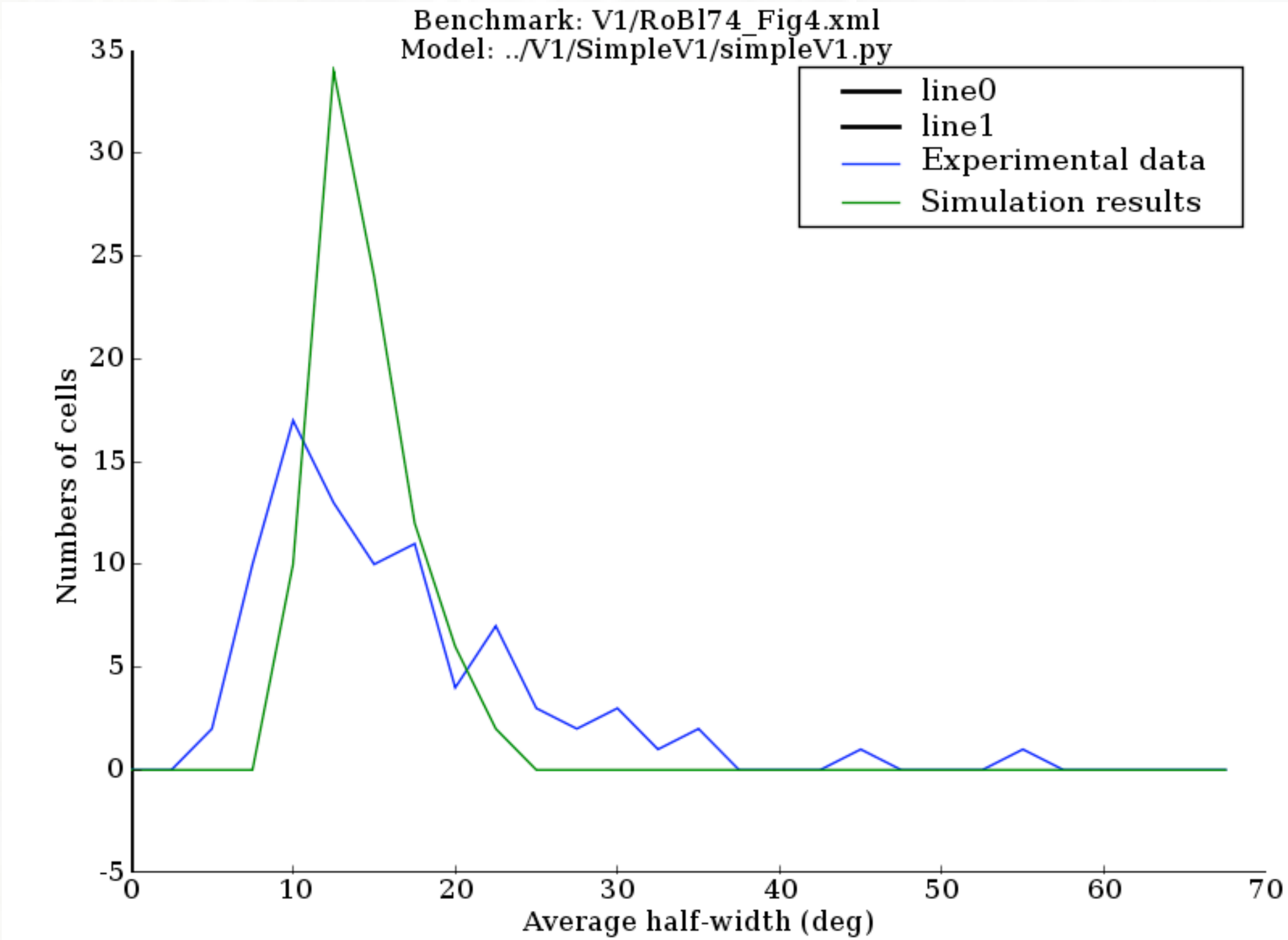
If you use Python...

```
python visionbenchmark.py
        spontaneous_firing_rate_dark.xml
        simpleV1.py
```

```
Id                    spontaneous-firing-rate-dark
Description           Spontaneous firing rate in the dark adapted cat.
Recording.Measureable spikes
Difference_Measure    absolute-difference
Analysis 1            mean firing rate

Protocol "dark_screen":
          repetitions:     1
          weight:          1.0
          duration:        10000.0 ms
          scale-factor:    0.1 pixels/degree
          stimulus:        srb://facets.inria.fr/WP5/Benchmarks/VisualStimuli/dark_screen.zip
          max-luminance:   1.0 cd/m²
          comparison-data: 4.0 s^-1

Running protocol "dark_screen":

Running retina model .
Analysing results...
    mean firing rate: {'not-specified': (0.0,)}
absolute-difference difference: 4.0
```


Then to run the simulation and do the comparison is a single command, specifying the experiment to run and the model to run it on.

Distribution of orientation tuning curve widths



In FACETS at least three different models of V1 cortex were developed, and we ran a small library of five benchmarks on each of them.
Of course, this is just a proof of concept, but we think we've made a good start in being able to automatically test models against a large library of biological datasets.

UNIC Helmholtz Database: List of experiments

https://www.dbunic.cnrs-gif.fr/helmholtz/vision/

Google

Helmholtz

Home | Vision | Logout | Profile | Help | Contact

List of experiments

[Add New Experiment]

Page 1 of 2

Ref.	Date	Animal	Block recordings
MANIP_2006_18	25/04/2006	Cat, M	6
MANIP_2006_12	22/03/2006	Cat, M	4
MANIP_2006_03	10/01/2006	Cat, FAM130, M	5
MANIP_2006_01	03/01/2006	Cat, ESU173	9
MANIP_2005_49	01/12/2005	Cat, EVP507, M	14
MANIP_2005_05	25/01/2005	Cat, F	2
MANIP_2004_51	13/12/2004		2
MANIP_2004_48	23/11/2004	Cat, EGT721, M	4
MANIP_2004_25	15/06/2004	Cat, DWE772, F, 156 weeks	20
MANIP_2004_22	24/05/2004	Cat, DWE770, F, 156 weeks	4
MANIP_2004_19	04/05/2004	Cat, DVL071, M, 261 weeks	12
MANIP_2004_17	19/04/2004	Cat, DUF306, 365 weeks	5
MANIP_2004_14	30/03/2004	Cat, DVL070, M, 313 weeks	3
MANIP_2004_12	16/03/2004	Cat, F, 108 weeks	6
MANIP_2004_08	17/02/2004	Cat, F	9
MANIP_2004_05	27/01/2004	Cat, 2778U, F, 521 weeks	5

In the prototype, we took data from the literature, but of course we would like to make more in-depth comparisons, and, as Wulfram Gerstner said in his talk, if you're going to compare models you need a training set and a testing set, so we would also like to compare to experimental recordings directly. To do this, and to promote the reuse of existing biological data more widely, we've started to develop a database of experimental data obtained during FACETS and previously, from several experimental labs.

Helmholtz

Home | Vision | MANIP_2004_25 Logout | Profile | Help | Contact

Experiment MANIP_2004_25 [Edit] - [Delete]

Date	15th June 2004
Setup	
Experimenters	

Animal

Species	Sex	Tattoo	Age	Weight	Eye Correction
Cat	Female	DWE772	157 weeks	3.2 kg	(dioptres)

General Notes	Medical Notes	Behaviour

Drug Injections

No data available

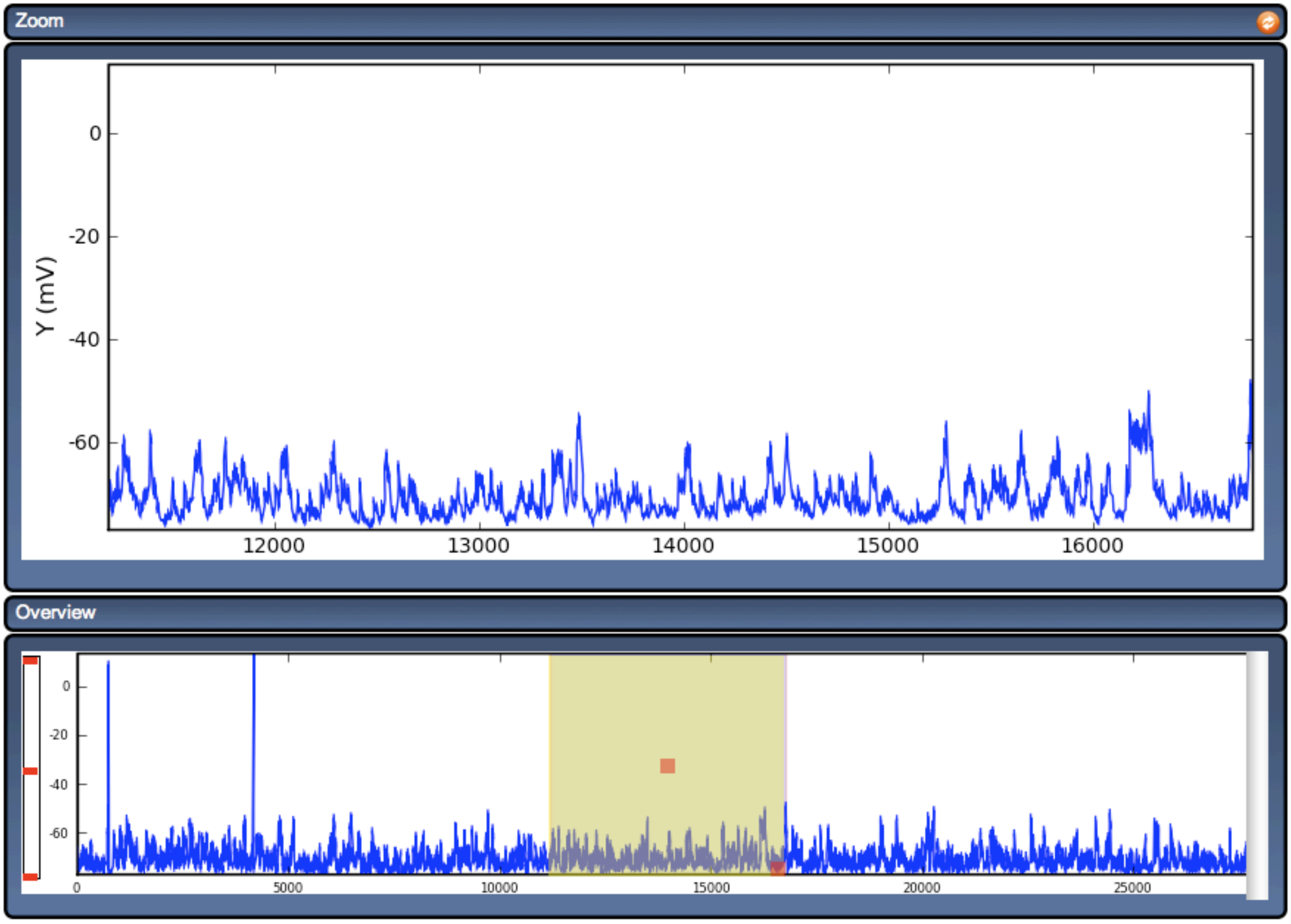
Drug Perfusions

No data available

Recording Blocks

Id	Duration (minutes)	Start time	End time	DAT Tapes	Recording method(s)	Elphy Files	Notes
0425A					sharp	0	
0425B					sharp	0	
0425C	45	15-06-2004 04:48	15-06-2004 05:33		sharp	3	
0425D	8	15-06-2004 05:33	15-06-2004 05:41		sharp	0	
0425E	167	15-06-2004 05:41	15-06-2004 08:28		sharp	11	
0425F					sharp	2	
0425G	120	15-06-2004 08:28	15-06-2004 10:28		sharp	9	

In this database, we aim to capture all the metadata that would be needed either to reproduce an experiment by a biologist, or to reproduce the experiment in a simulation.
This takes time, of course. Here there is data available for the anaesthesia and so forth, but it's all written down in lab notebooks, in bad handwriting, and it takes time to digitize all this stuff.



We can also access and download the data itself, of course.

To improve communication of data and information requires both social change and tool development

- social/process aspect: make it normal to digitize/share data + information
  - carrot: make datasets, etc. citable, count towards career progression
  - stick: funding agencies increasingly require it

- tools aspect: make it easy to digitize/share data + information

I'd like to finish by reflecting on what we've learnt from the experience of working together in FACETS.

I started by saying that we need to improve communication of data and metadata between groups. Doing this requires changing some of the norms within our scientific field, making data and information sharing normal and rewarded, and requires data sharing to be made easier, through development of appropriate tools.

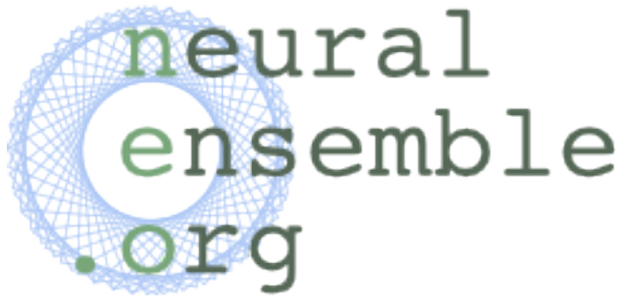
The best way to get these tools developed is via open collaboration

- large, well-funded, centralized projects
  - (BlueBrain Project, Allen Brain Atlas, ...)
  - have the manpower/resources to develop tools/resolve all these issues internally
- the rest of us
  - can't build all the pieces ourselves
  - need to collaborate, share components
  - either through formal collaborative projects like FACETS, or through informal collaborations
- this is really a false dichotomy
  - large centralised projects often very keen to share what they develop, and benefit from tools developed by others (*cf* contributions of IBM, etc. to Linux, Google to Python, ...)

These tools can't really be built by individual researchers or even individual labs. Formal collaboration through funded projects is not necessary – it's great to kick-start things, as FACETS has shown, but many people who have participated in the development of the tools I've shown today are from outside FACETS.
Also, even imperfect tools have value, and I think neuromorphic computation will benefit from the open-source philosophy of "release early, release often": if your tool is useful for you now, release it, don't wait to polish it to perfection first.

Collaborative tool development benefits from formal or informal coordination

- promote discussion
- develop standard interfaces



Even if the collaboration is informal, it can benefit from coordination, either through international organisations such as the INCF, or through grass-roots efforts such as the Neural Ensemble initiative, started by myself and Eilif Muller with the generous support of FACETS.

Acknowledgements

Funding
FET (FACETS)
CNRS

Benchmarking
Jens Kremkow
Klaus Schuch
Mikael Djurfeldt

FIND
Ralph Meier
Christian Garbers
Ulrich Egert

PyNN
Pierre Yger
Eilif Muller
Daniel Brüderle
Jochen Eppler
Jens Kremkow
Dejan Pecevski

Databases
Thierry Brizzi
Olivier Manette
Cyril Monier
Gérard Sadoc
Zoltan Kisvárday

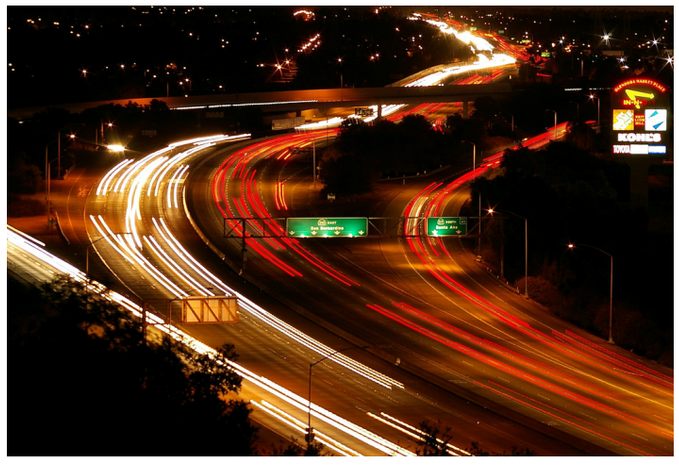
Elphy
Gérard Sadoc

NeuroTools
Laurent Perrinet
Pierre Yger
Daniel Brüderle
Eilif Muller
Michael Schmuker
Jens Kremkow
Samuel Garcia
Luc Estebanez

Support and advice
Yves Frégnac
Thierry Viéville
Alain Destexhe
Karlheinz Meier

The work presented here is due to a large number of people. I apologize if I've forgotten anyone.

Conclusions



- the best way to turn the cart-track into the superhighway
  and help make the complexity of neuromorphic computation manageable
  is software development in the open

- if you're interested in using and/or helping to develop PyNN, NeuroTools, etc.,
  please check out

http://neuralensemble.org/

- if you have a problem, an idea, or are seeking collaborators for your own software project,
  - check out http://groups.google.com/group/neuralensemble,
  - or come to the CodeJam http://neuralensemble.org/codejam4

http://www.andrewdavison.info/contact/
Twitter: @apdavison.