Dimitri Yatsenko, Jacob Reimer, Edgar Walker
(Andreas Tolias' Lab)

DataJoint

Intro

2016-12-09

The DataJoint framework

Aim: Efficiency, integrity, and collaboration

- Large volumes of multimodal, dynamic data with evolving structure
- Explicit, systematic, self-documenting data organization
- Solid theoretical foundations: a strict and streamlined variant of the relational data model to make easy to understand and use
- MATLAB and Python interoperability
- Data integrity
- Precise, fast, interactive queries
- Data sharing with fine access control
- Distributed computation

DataJoint history

- In continuous use and development since **2009** in Andreas Tolias lab but never as its own project.
- Used by several other labs since **2011**.
- **2016:** a major data organization tool for the MICrONS project.
- **Oct 2016:** Dr. Paul Schrater at U. of Minnessota was awarded $50k supp funding to adopt DataJoint and to write tutorials
- **Nov 2016:** Dimitri Yatsenko et al founded Vathes LLC to apply for Phase I small-business funding for enhanced hosting and networking services based on DataJoint.

DataJoint

data pipelines for the science lab
with MATLAB and Python

Dimitri Yatsenko

In-depth guide to be released on March 1, 2017
https://leanpub.com/datajoint

Introductory tutorials online

Big Data? NoSQL?

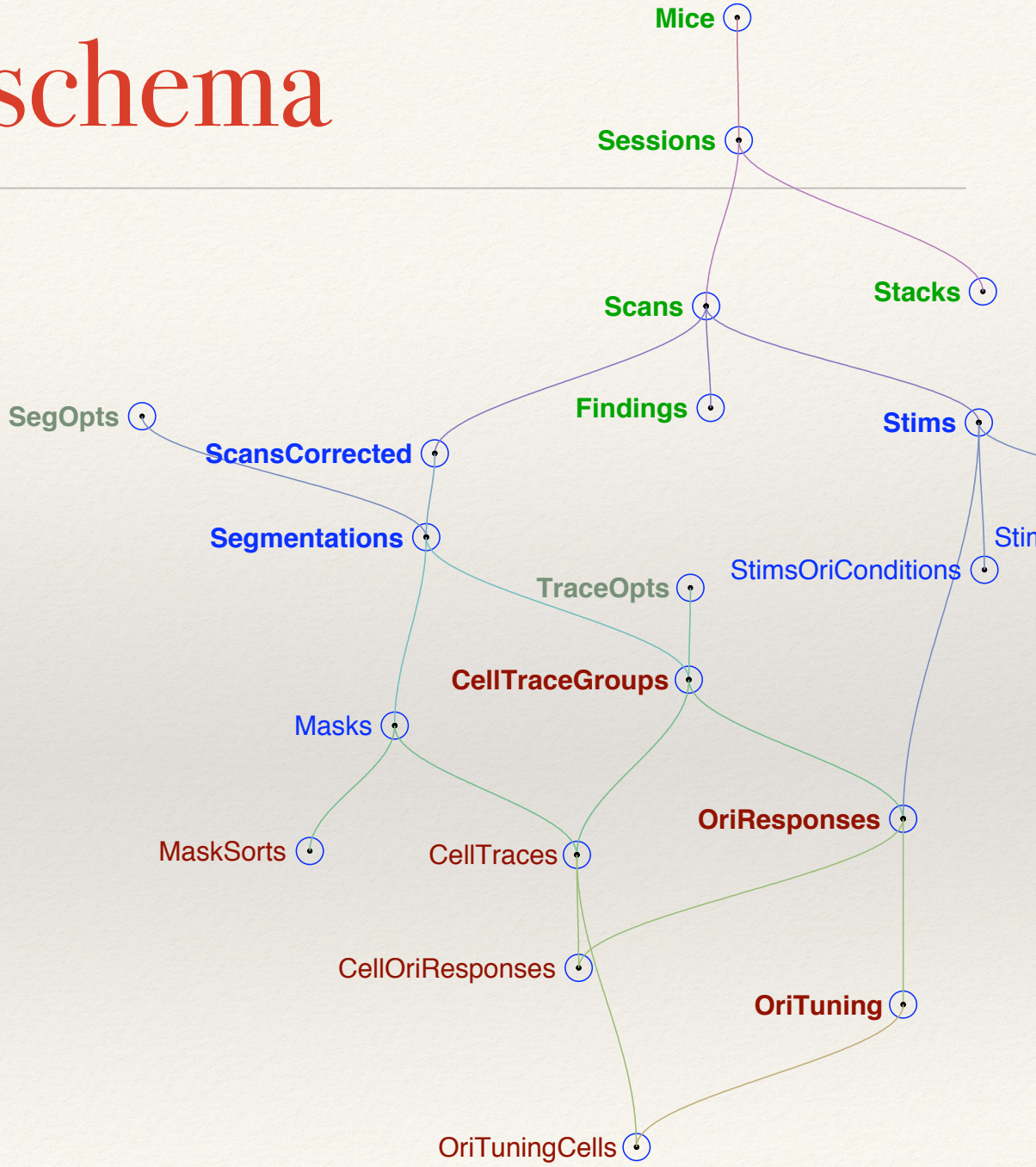
Big Data often describes data without predefined structure

- structure is learned from data itself
- in this sense, DataJoint is **not** Big Data as it enforces explicit structure

The trendy **NoSQL** databases abandon structure and consistency for the sake of flexible distributed hosting (scalability)

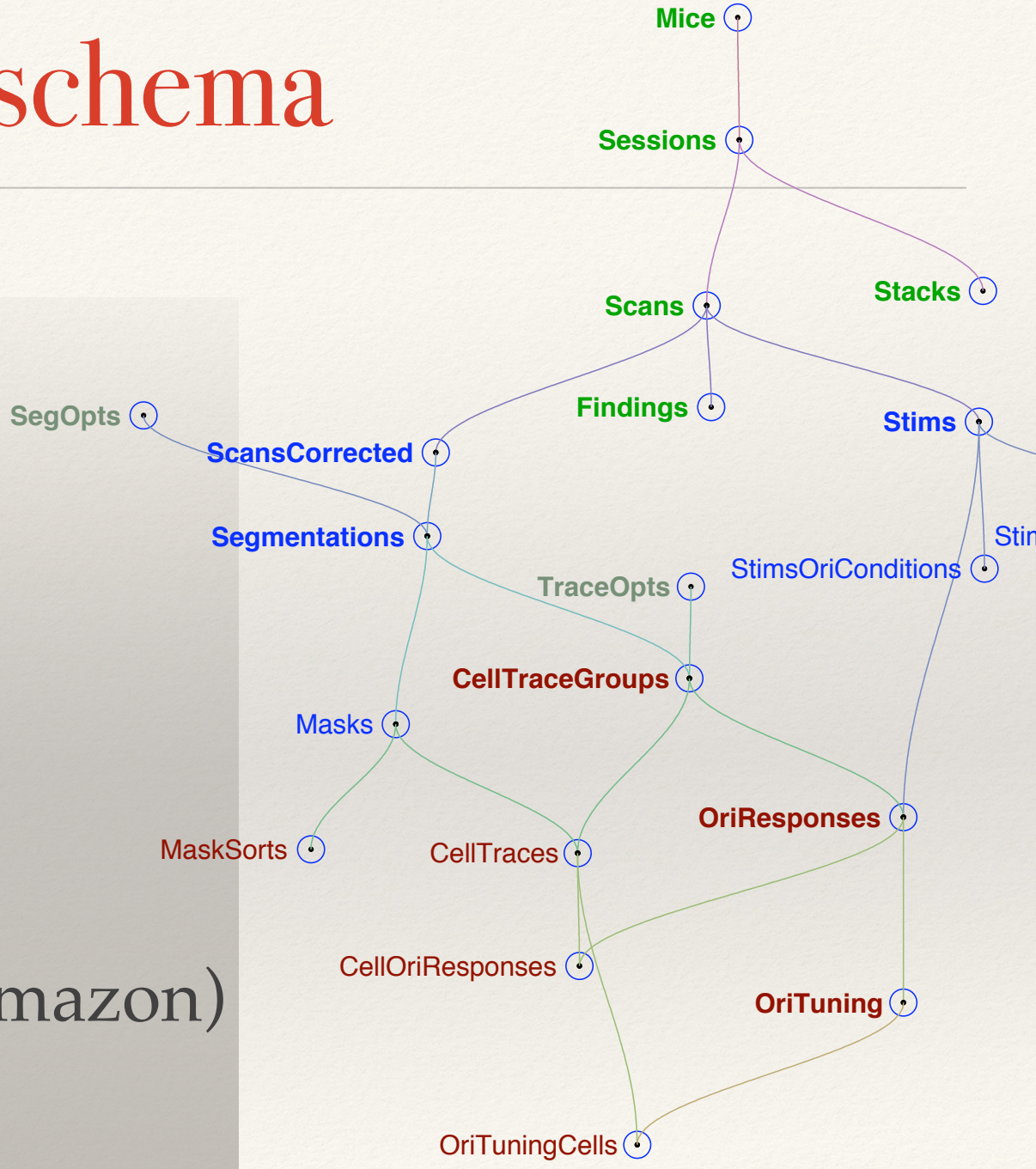
- in this sense, DataJoint is **not** NoSQL
- DataJoint strives to achieve scalability while maintaining consistency

A DataJoint schema



A DataJoint schema

- Backed by MySQL
- Hosting
  - personal computer
  - lab server
  - cloud server (e.g. Amazon)



What is a DataJoint database?

- Nodes are tables of data

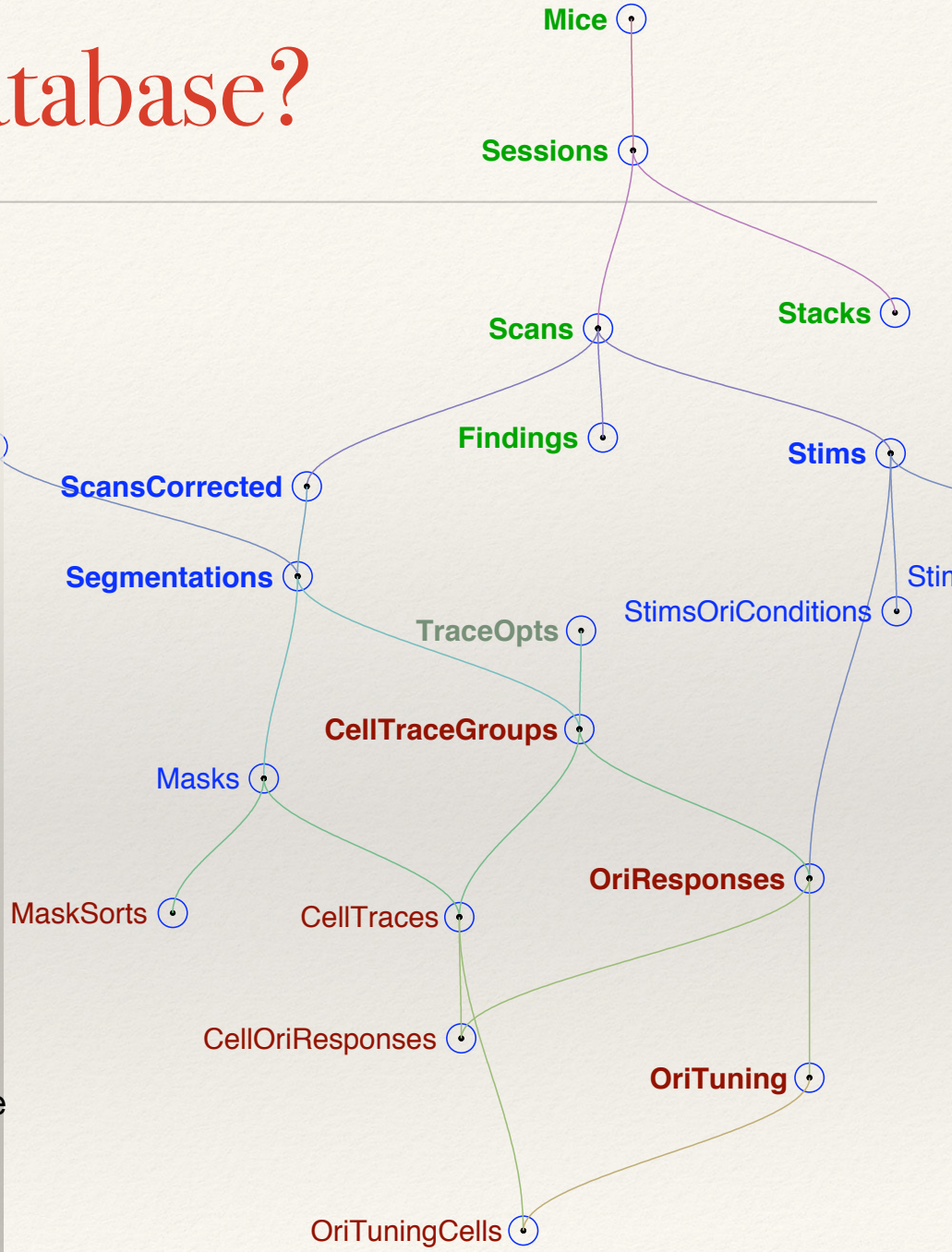
primary key

attribute name

id	scan	scan_date	operator
2	1	2014-01-01	Edgar
2	2	2014-06-01	Edgar
2	3	2015-01-01	George
3	1	2015-05-01	Cathryn
3	2	2015-05-08	Shan
3	3	2015-05-15	Shan

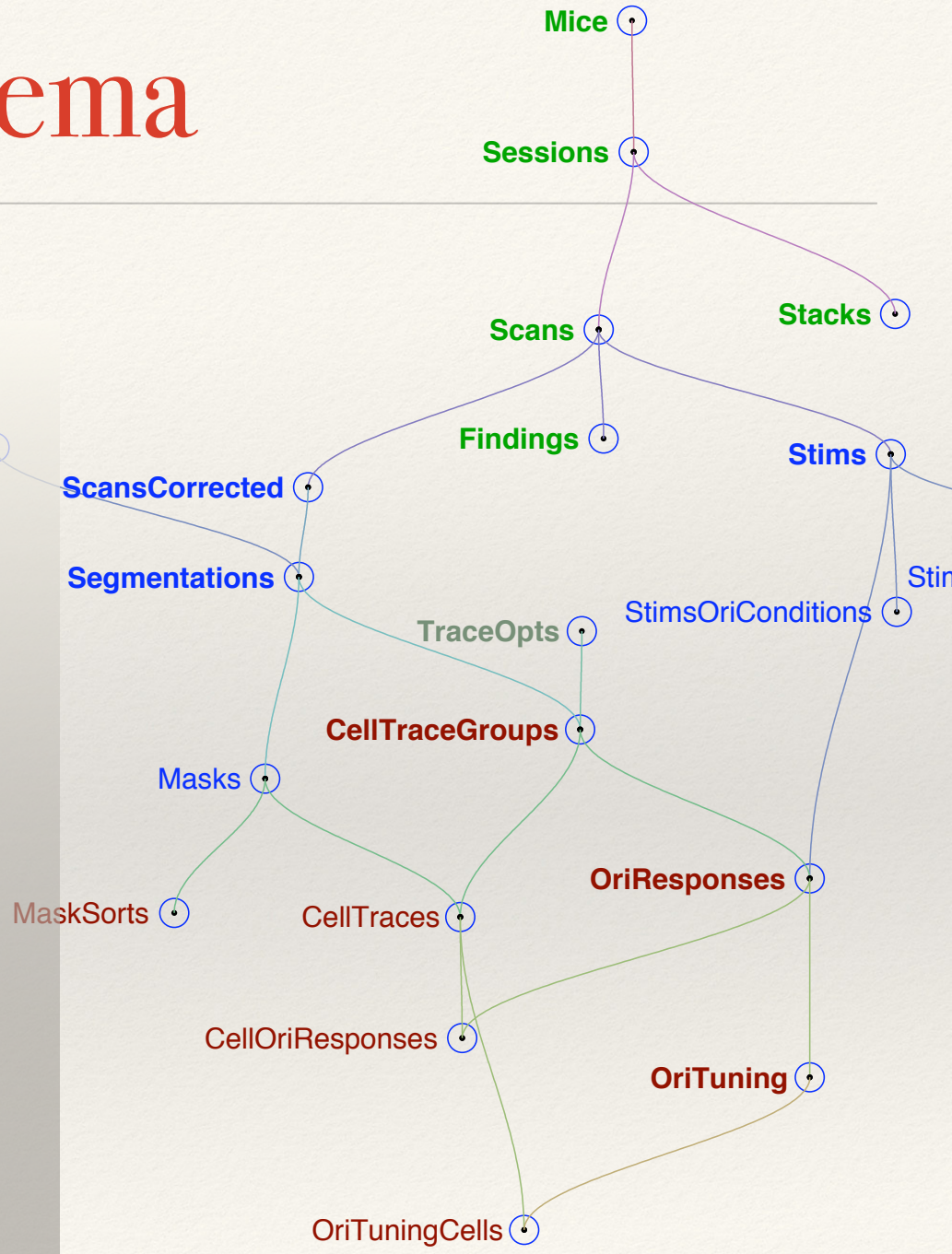
tuple

attribute value



A DataJoint schema

- Nodes are classes in MATLAB or Python
- Nodes are also tables in a relational database
- Data echelons:
  - manual
  - imported
  - computed



A DataJoint schema

DataJoint infrastructure

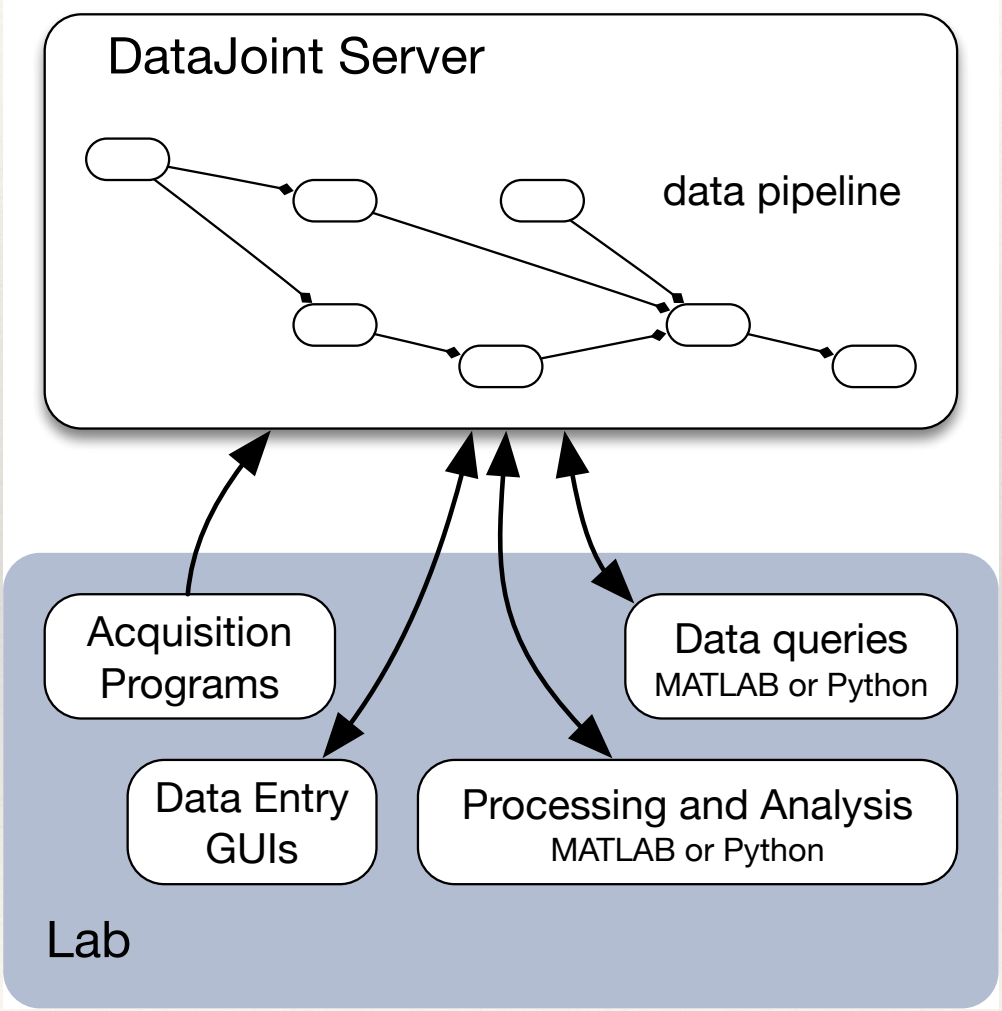
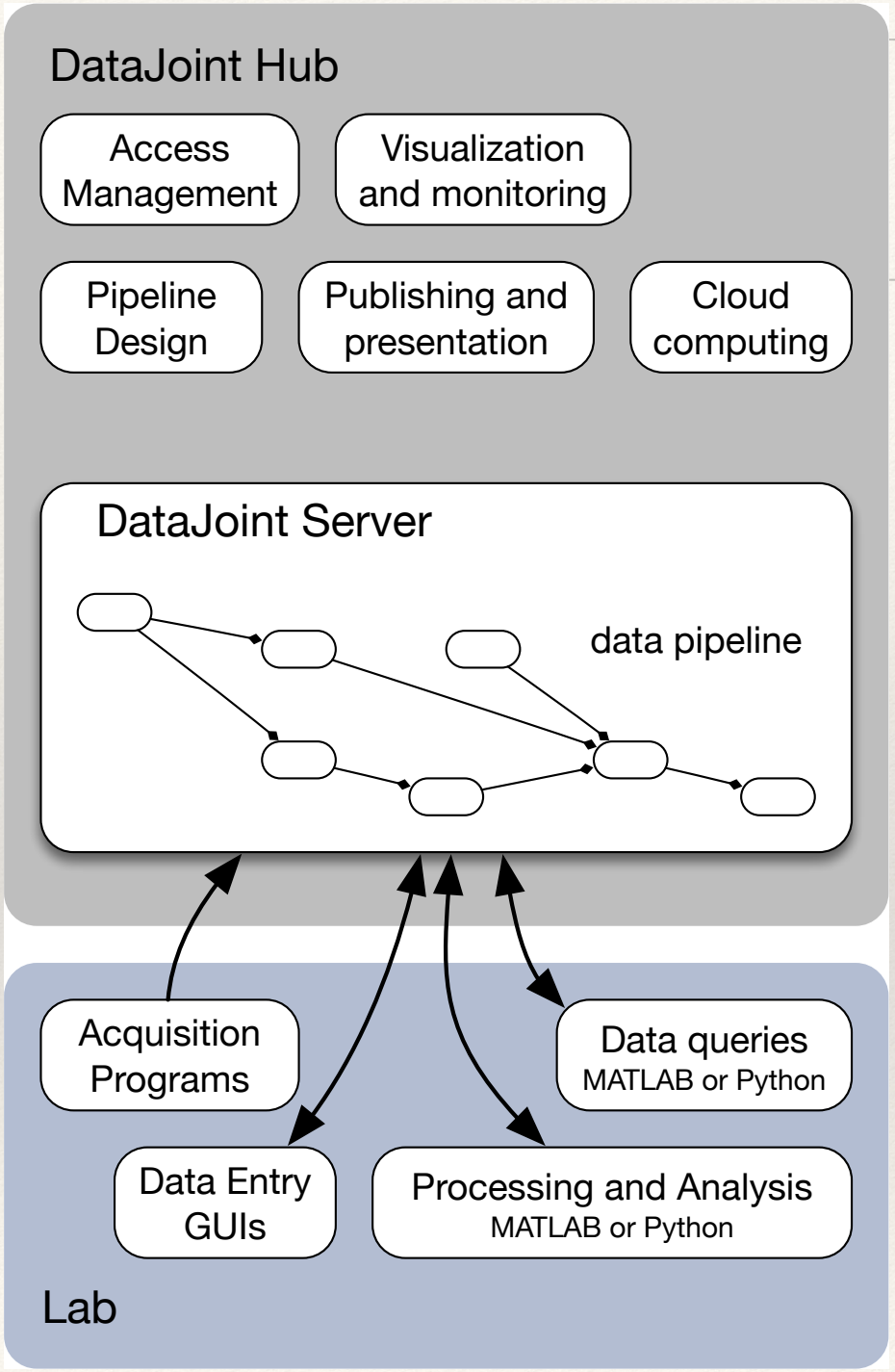


Figure 1: Basic DataJoint architecture.



DataJoint infrastructure

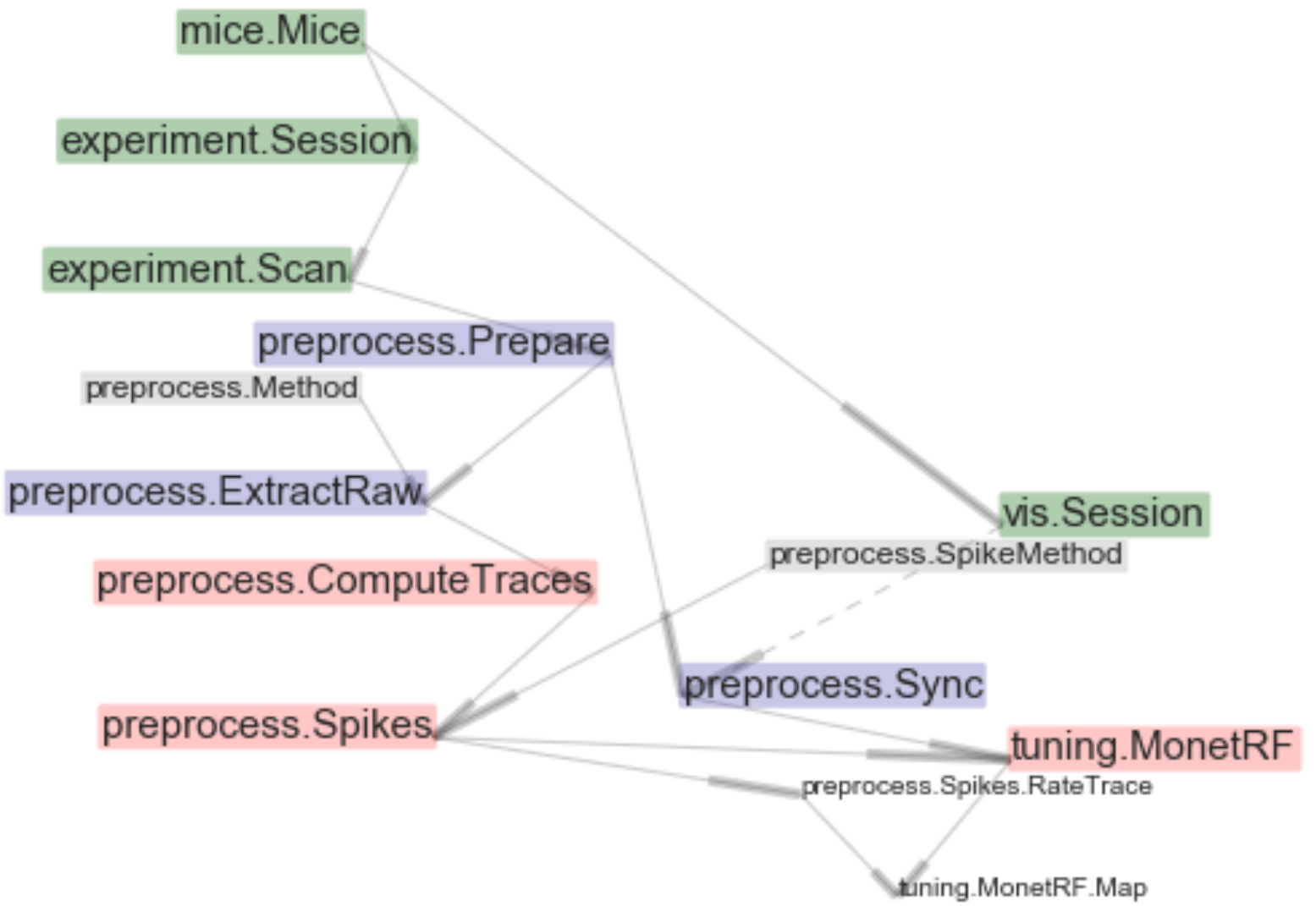
(planned in 2017)

DataJoint Hub (datajoint.io)

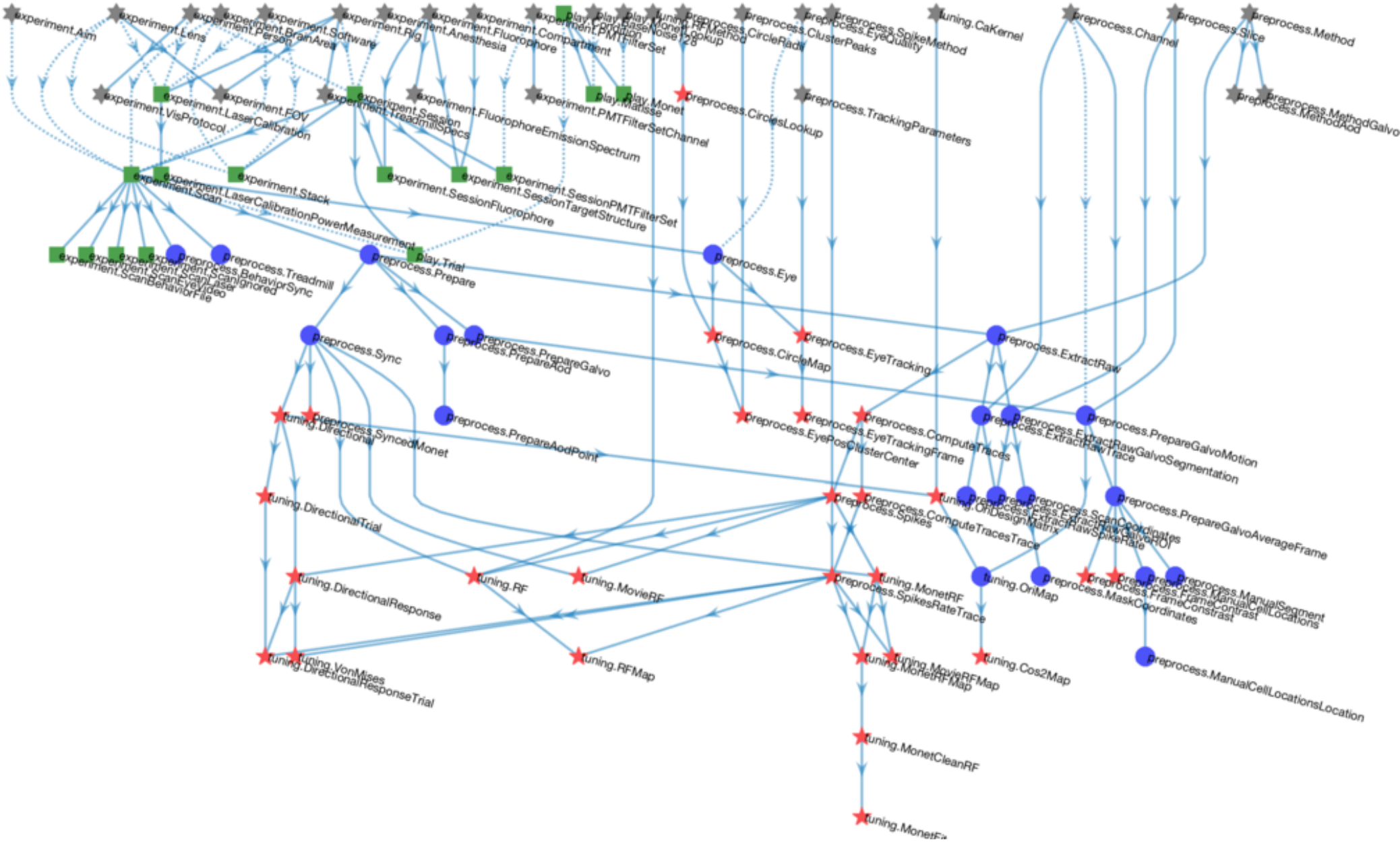
- pipeline hosting (free or subscription)
- distributed cloud computing

MICrONS pipeline (fragment)

https://github.com/cajal/pipeline

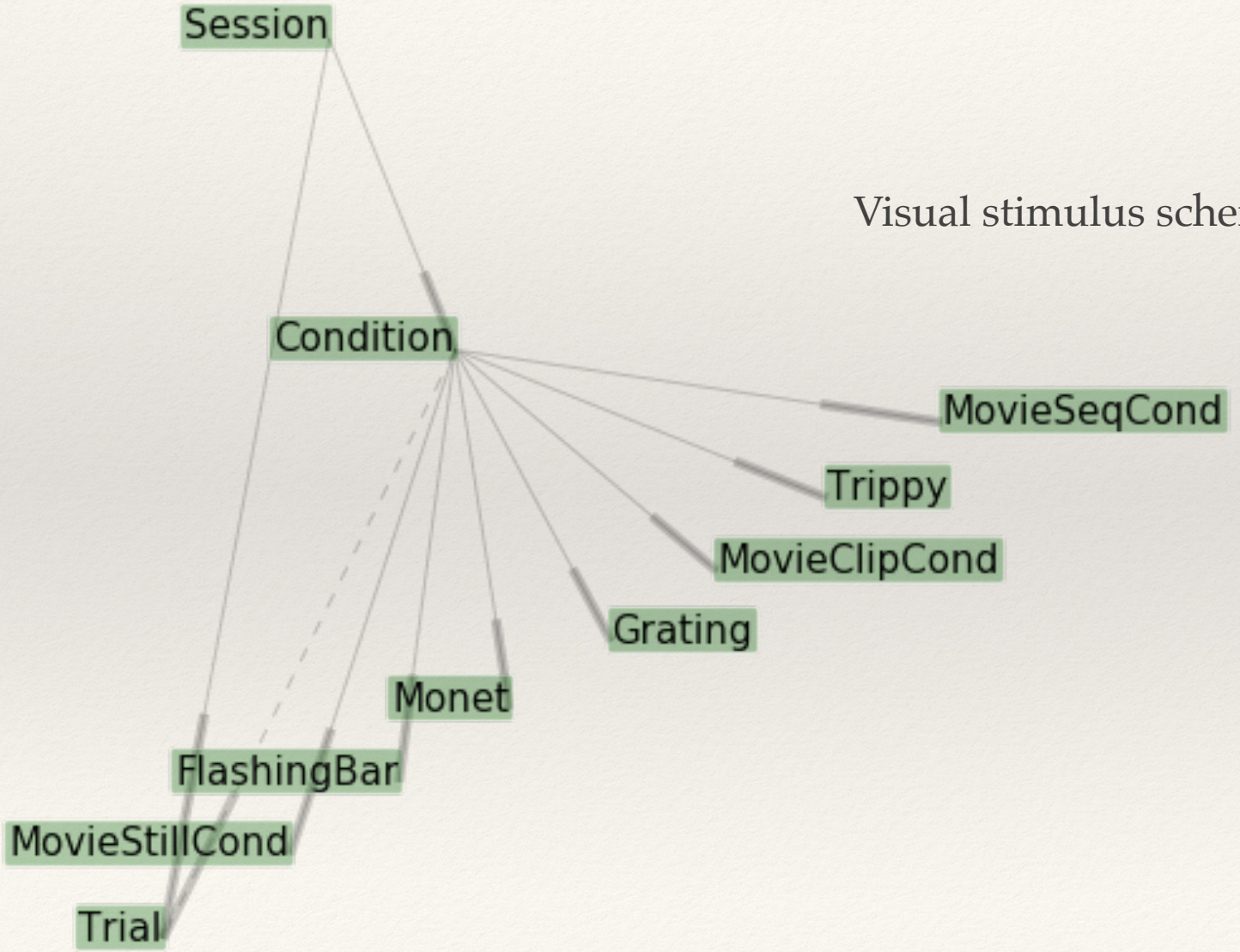


MICrONS pipeline



MICrONS pipeline (fragment)

https://github.com/cajal/pipeline

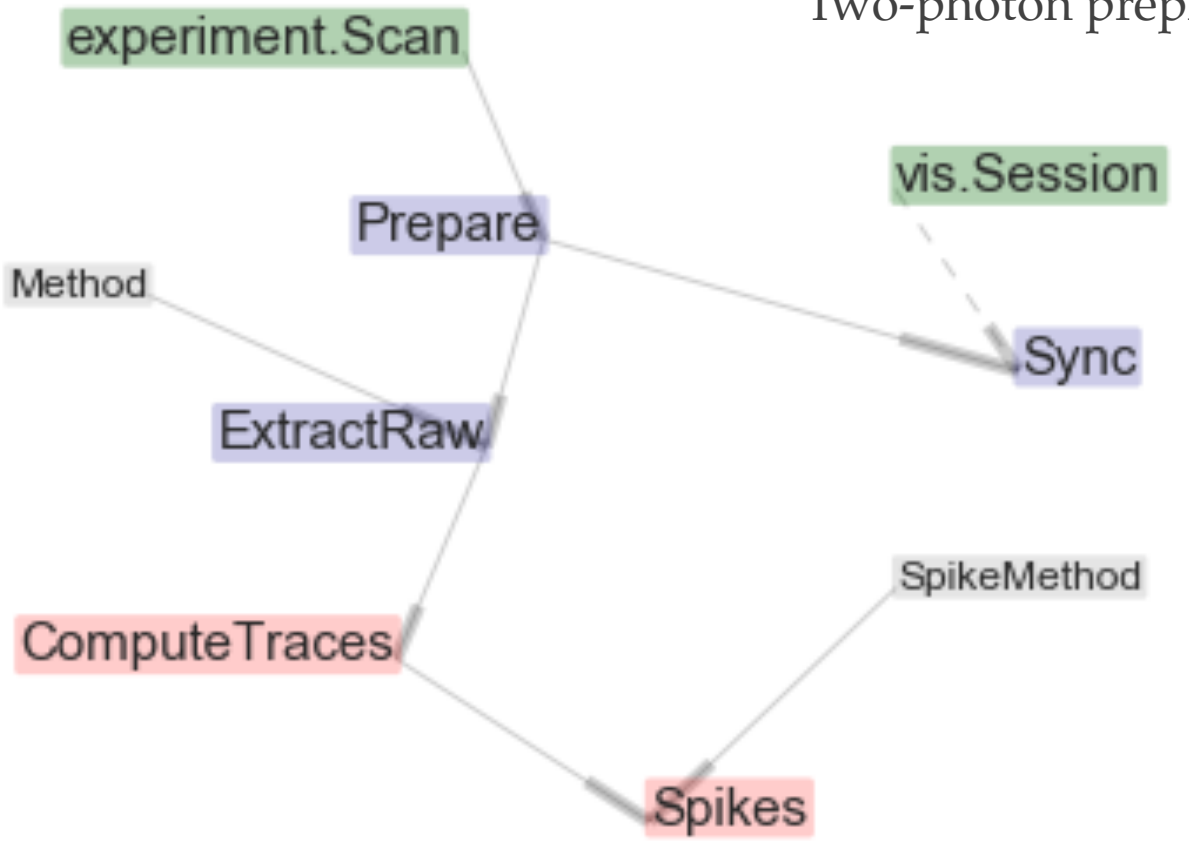


Visual stimulus schema `vis`

MICrONS pipeline (fragment)

https://github.com/cajal/pipeline

Two-photon preprocessing schema `preprocess`



A DataJoint node

Python

```
@schema
class Scan(dj.Manual):
    definition = """   # scanimage scan info
    -> Session
    scan_idx : smallint # scanimage-generated index
    -----
    surfz                :float  # (um) z-coord at pial surface
    depth=0              :int  # manual depth measurement
    laser_wavelength     :float    # (nm)
    laser_power          :float    # (mW) to brain
    cortical_area="V1" :enum('other','unknown','V1','LM','AL','PM')
    scan_notes = ""     :varchar(4095)  #  free-notes
    scan_ts = CURRENT_TIMESTAMP : timestamp   # don't edit
    """
```

A DataJoint node

MATLAB

```
%{
common.Scan (manual) # scanimage scan info
->common.Session
scan_idx : smallint # scanimage-generated index
-----
surfz                :float  # (um) z-coord at pial surface
depth=0              :int  # manual depth measurement
laser_wavelength     :float    # (nm)
laser_power          :float    # (mW) to brain
cortical_area="V1" :enum('other','unknown','V1','LM','AL','PM')
scan_notes = ""     :varchar(4095)  #  free-notes
scan_ts = CURRENT_TIMESTAMP : timestamp   # don't edit
%}

classdef Scan < dj.Relvar
end
```

A DataJoint node

MATLAB

```
%{
common.Scan (manual) # scanimage scan info
->common.Session
scan_idx : smallint # scanimage-generated index
-----
surfz                :float  # (um) z-coord at pial surface
depth=0              :int  # manual depth measurement
laser_wavelength     :float    # (nm)
```

animal_id	tp_session	scan_idx	surfz	depth	laser_wavelength	laser_power	cortical_ar
3625	2	5	0	57	920	40	V1
3626	1	1	0	80	920	34	V1
3626	1	3	0	80	920	34	AL
3626	1	5	0	120	920	34	AL
3626	1	7	0	90	920	34	V1

```
classdef Scan < dj.Relvar
end
```

A DataJoint node (computed)

Python

```
@schema
class Power(dj.Computed):
    definition = """  # LFP power in each frequency band
    -> LFP
    -> FrequencyBand
    -----
    power :float  # mV^2 Hz -- averge power in the band
    """

    def _make_tuples(self, key):
        # fetch required data
        lo, hi = (FrequencyBand() & key).fetch1['freq_lo', 'freq_hi']
        signal, dt = (LFP() & key).fetch1['voltage', 'dt']
        # compute
        signal = band_pass_filter(signal, lo, hi, dt)
        power = compute_average_power(signal, dt)
        # submit
        self.insert1(dict(key, power=power))
```

A DataJoint node (computed)

MATLAB

```
%{
example.Power (computed)  # LFP power in each frequency band
-> example.LFP
-> example.FrequencyBand
-----
power :float  # mV^2 Hz -- average power in the band
%}

classdef Power < dj.Relvar & dj.AutoPopulate
    methods(Access=protected)
        function makeTuples(self, key)
            % fetch required data
            [lo, hi] = fetch1(example.FrequencyBand & key, 'freq_lo', 'freq_hi');
            [signal, dt] = fetch1(example.LFP & key, 'voltage', 'dt');
            % compute
            signal = band_pass_filter(signal, lo, hi, dt);
            key.power = compute_average_power(signal, dt);
            % submit
            self.insert(key)
        end
    end
end
```

Automated computation

matlab

```
>> populate(example.Power)
```

python

```
>>> example.Power.populate()
```

Distributed computation

matlab

```
>> parpopulate(example.Power)
```

python

```
>>> example.Power().populate(reserve_jobs=True)
```

Query language

- Formulate and refine your query before retrieving data
- Three operators:
  - restrict A & cond
  - difference A - cond
  - join A * B
  - union A + B
  - project A.proj(*attr-list*)
  - aggregate A.aggr(B, *computations*)

Query language

Restrict A&B

A & B = all rows in A that have matching rows in B

id	species	sex	date_of_birth
1	mouse	F	2015-04-01
2	monkey	M	2011-12-01
3	mouse	M	2015-05-08
4	mouse	F	2015-05-08

&

id	scan	image
2	1	(BLOB)
2	2	(BLOB)
2	3	(BLOB)
3	1	(BLOB)
3	2	(BLOB)
3	3	(BLOB)

=

id	species	sex	date_of_birth
2	monkey	M	2011-12-01
3	mouse	M	2015-05-08

Query language

Join

A * B = all combinations of matching rows from A and B

id	species	sex	date_of_birth
1	mouse	F	2015-04-01
2	monkey	M	2011-12-01
3	mouse	M	2015-05-08
4	mouse	F	2015-05-08

*

id	scan	image
2	1	(BLOB)
2	2	(BLOB)
2	3	(BLOB)
3	1	(BLOB)
3	2	(BLOB)
3	3	(BLOB)

=

id	scan	species	sex	date_of_birth	image
2	1	monkey	F	2011-12-01	(BLOB)
2	2	monkey	F	2011-12-01	(BLOB)
2	3	monkey	F	2011-12-01	(BLOB)
3	1	mouse	M	2015-05-08	(BLOB)
3	2	mouse	M	2015-05-08	(BLOB)
3	3	mouse	M	2015-05-08	(BLOB)

Query language

Project:

Select, rename, or compute columns

id	scan	date_of_birth	image
2	1	2011-12-01	(BLOB)
2	2	2011-12-01	(BLOB)
2	3	2011-12-01	(BLOB)
3	1	2015-05-08	(BLOB)
3	2	2015-05-08	(BLOB)
3	3	2015-05-08	(BLOB)

.proj(‘id -> animal_id’, ‘image’**) =**

animal_id	scan	image
2	1	(BLOB)
2	2	(BLOB)
2	3	(BLOB)
3	1	(BLOB)
3	2	(BLOB)
3	3	(BLOB)

Expressions

“all the two-photon sessions using the **25× lens** that have scans that have **not yet been synchronized** with the **visual stimulus**”

```
Session & 'lens="25x"' & (Scan - Sync)
```

Relational algebra (combinations of operators)

"all the two-photon sessions using the **25× lens** that have scans that have **not yet been synchronized** with the **visual stimulus**"

```
Session & 'lens="25x"' & (Scan - Sync)
```

translated into SQL

```
SELECT *
FROM `tp`.`session`
WHERE lens="25x"  AND (`animal_id`,`session_id`) IN (
    SELECT `animal_id`,`session_id`
    FROM `tp`.`scan`
    WHERE ((`animal_id`,`scan_id`,`session_id`) NOT IN (
        SELECT `animal_id`,`scan_id`,`session_id`
        FROM `tp`.`_sync`)))
```

DataJoint in publications

http://datajoint.github.com/publications/

1. Baden T, Berens P, Franke K, Rezac M, Bethge M, and Euler T (2015). The functional diversity of retinal ganglion cells in the mouse. *Nature*, 529(7586), pp.345-350. link
2. Cadwell CR, Palasantza A, Jiang X, Berens P, Deng Q, Reimer J, Tolias K, Bethge M, Tolias AS (2015). Morphological, electrophysiological and transcriptomic profiling of single neurons using Patch-seq. *Nature Biotechnology* link
3. Jiang X, Shen S, Cadwell CR, Berens P, Sinz F, Ecker AS, and Tolias AS (2015). Principles of connectivity among morphologically defined cell types in adult neocortex. *Science*, 350(6264), aac9462. link
4. Yatsenko D, Josic K, Ecker AS, Froudarakis E, Cotton RJ, and Tolias AS (2015). Improved estimation and interpretation of correlations in neural circuits. *PLoS Comput Biol* 11(3): e1004083. doi:10.1371/journal.pcbi.1004083 link
5. Erisken S, Vaiceliunaite A, Jurjut O, Fiorini M, Katzner S, and Busse, L (2014). Effects of Locomotion Extend throughout the Mouse Early Visual System. *Current Biology*, 24(24), 2899-2907.
6. Reimer J, Froudarakis E, Cadwell CR, Yatsenko D, Denfield GH, Tolias AS (2014). Pupil fluctuations track fast switching of cortical states during quiet wakefulness. *Neuron*, 84(2), 355-

DataJoint in publications

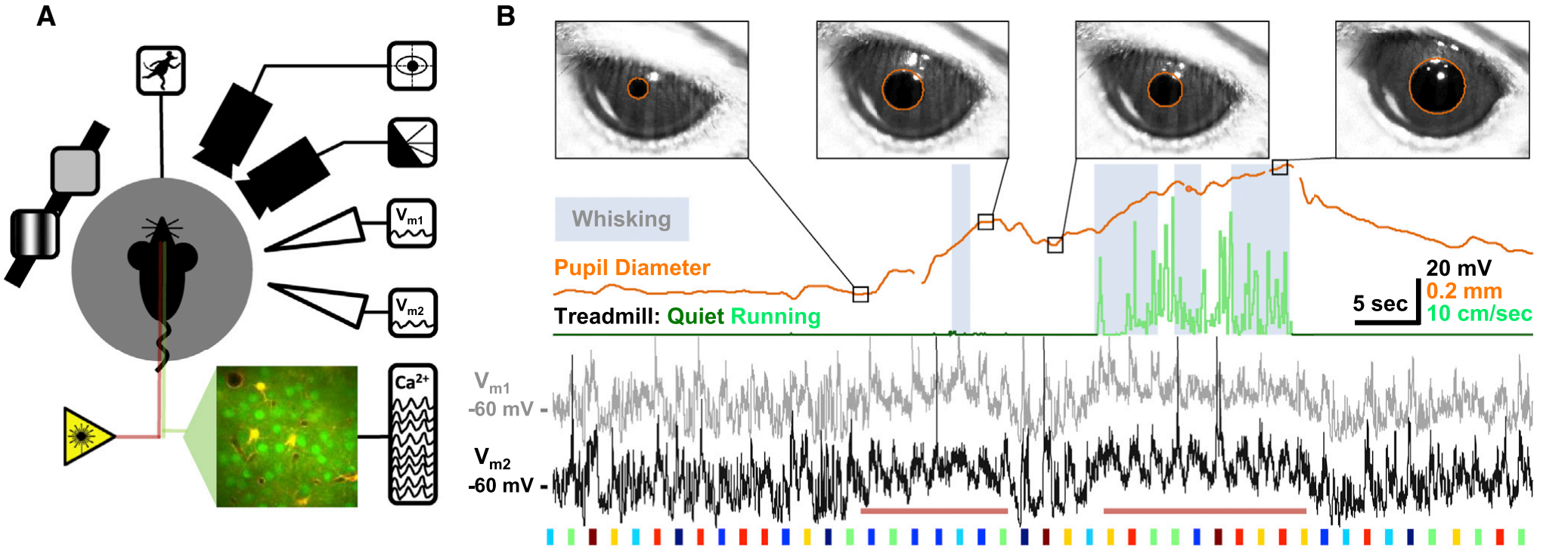
http://datajoint.github.com/publications/

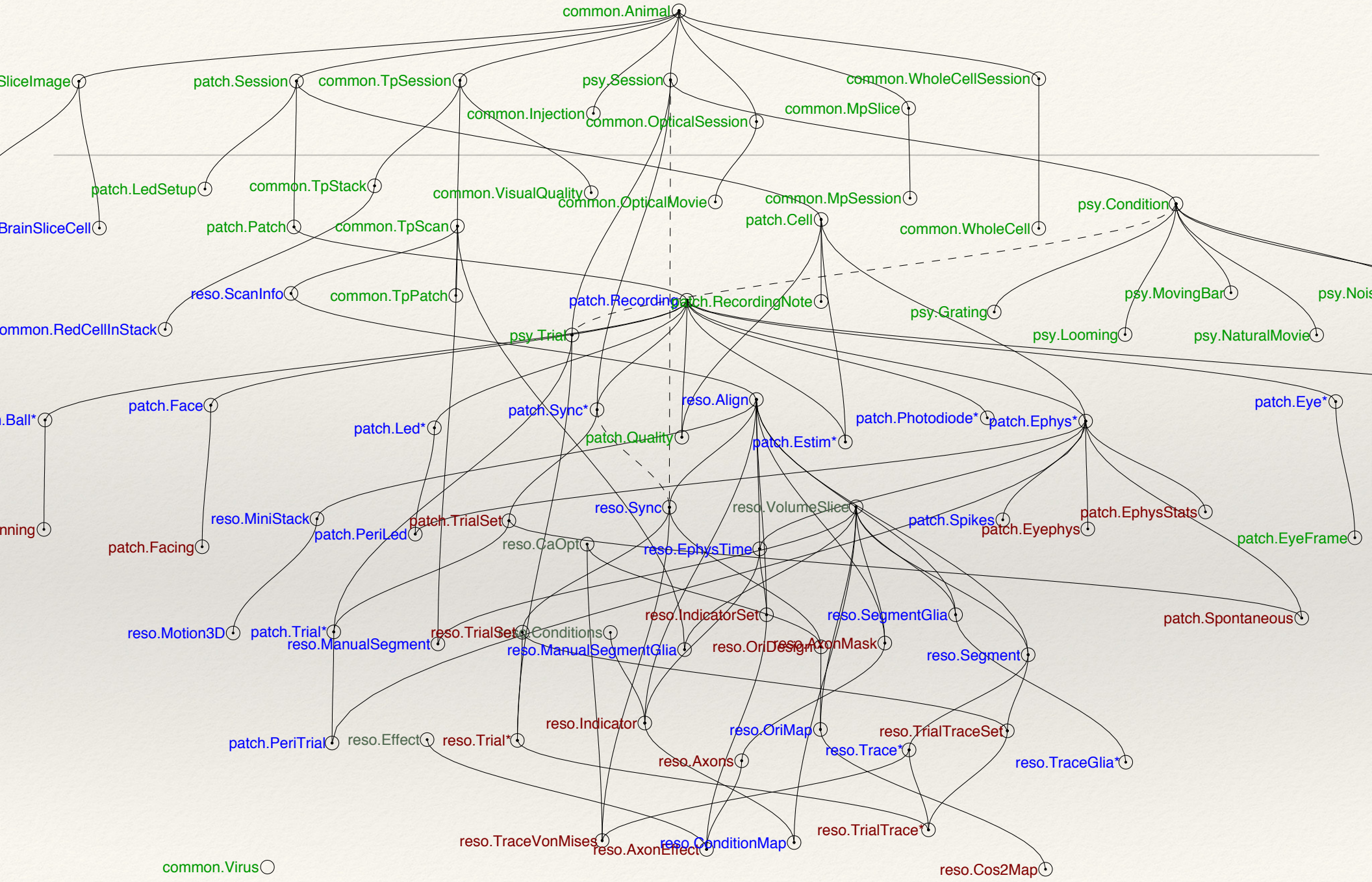
6. Reimer J, Froudarakis E, Cadwell CR, Yatsenko D, Denfield GH, Tolias AS (2014). Pupil fluctuations track fast switching of cortical states during quiet wakefulness. *Neuron*, 84(2), 355-362.
7. Erisken S, Vaiceliunaite A, Jurjut O, Fiorini M, Katzner S, and Busse L (2014). Effects of Locomotion Extend throughout the Mouse Early Visual System. *Current Biology*, 24(24), 2899-2907.
8. Froudarakis E, Berens P, Ecker AS, Cotton RJ, Sinz FH, Yatsenko D, Saggau P, Bethge M, and Tolias AS. Population code in mouse V1 facilitates readout of natural scenes through increased sparseness. *Nature neuroscience* (2014). link
9. Ecker AS, Berens P, Cotton RJ, Subramaniyan M, Denfield GH, Cadwell CR, Smirnakis SM, Bethge M, and Tolias AS (2014): State dependence of noise correlations in macaque primary visual cortex. Neuron 82(1). link code data pdf
10. Cotton RJ, Froudarakis E, Storer P, Saggau P, and Tolias AS (2013). Three-dimensional mapping of microcircuit correlation structure. *Frontiers in neural circuits*, 7. pubmed 24133414
11. Vaiceliunaite A, Erisken S, Franzen F, Katzner S, and Busse L (2013). Spatial integration in mouse primary visual cortex. *Journal of Neurophysiology*, 110(4), 964-972. pubmed 23719206

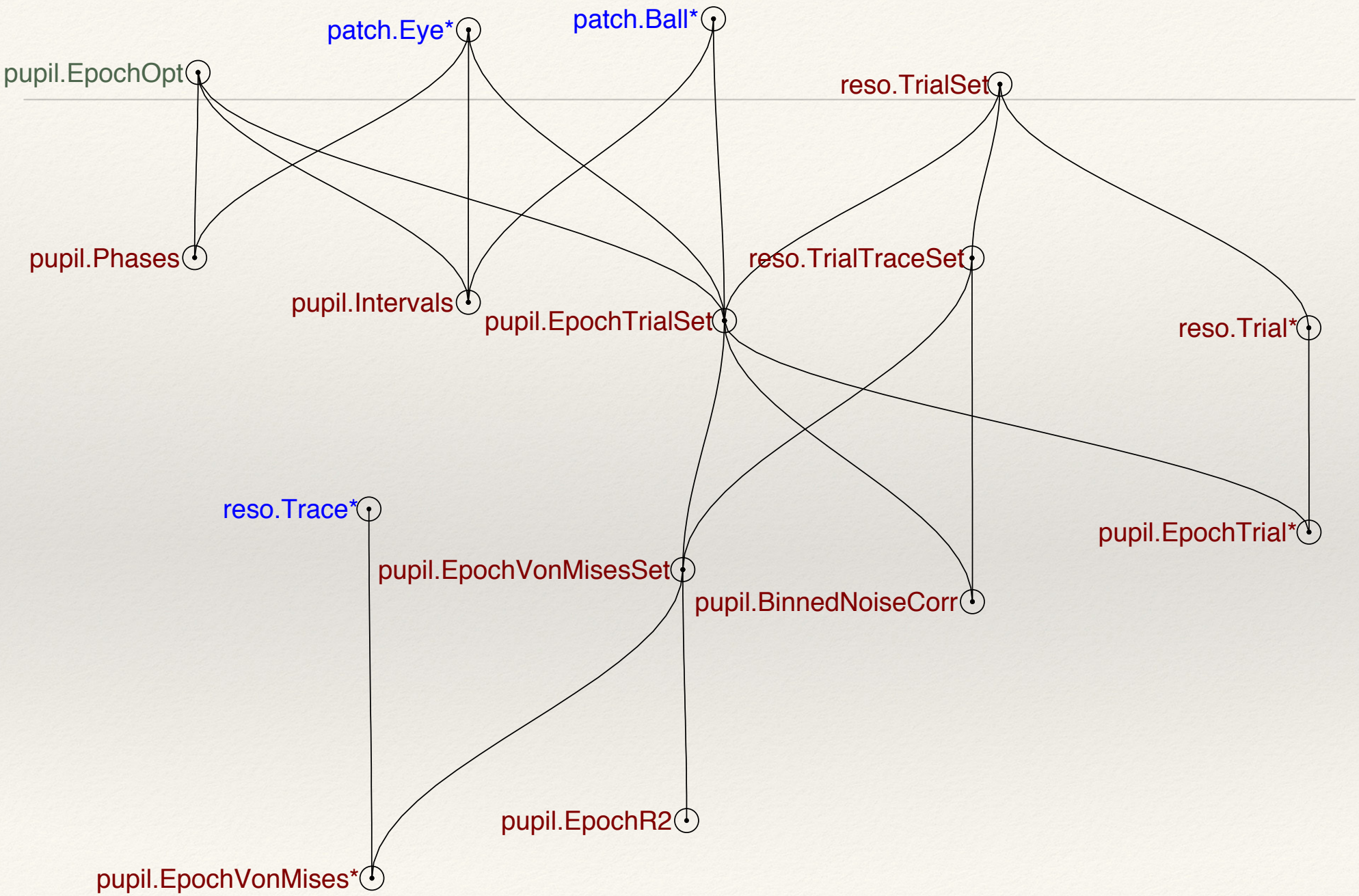
Pupil Fluctuations Track Fast Switching of Cortical States during Quiet Wakefulness

Jacob Reimer,[1,*] Emmanouil Froudarakis,[1] Cathryn R. Cadwell,[1] Dimitri Yatsenko,[1] George H. Denfield,[1] and Andreas S. Tolias[1,2,*]

Neuron *84*, 355–362, October 22, 2014



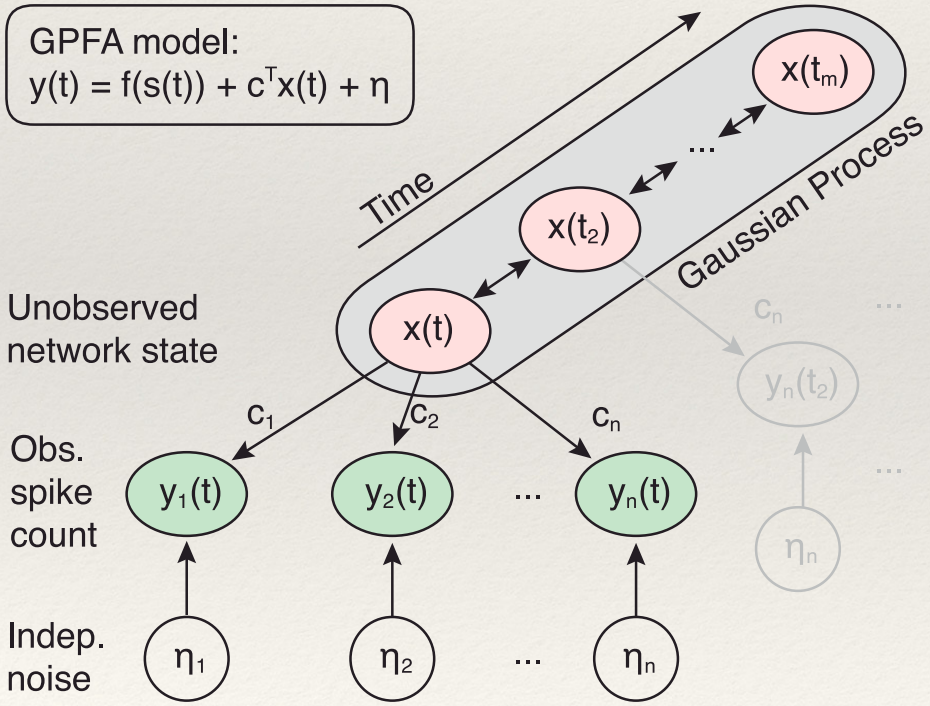

common.Animal
SliceImage
patch.Session
common.TpSession
psy.Session
common.WholeCellSession
common.Injection
common.OpticalSession
common.MpSlice
patch.LedSetup
common.TpStack
common.VisualQuality
common.OpticalMovie
common.MpSession
psy.Condition
BrainSliceCell
patch.Patch
common.TpScan
patch.Cell
common.WholeCell
reso.ScanInfo
common.TpPatch
patch.Recording
patch.RecordingNote
psy.Grating
psy.MovingBar
psy.Nois
ommon.RedCellInStack
psy.Trial
psy.Looming
psy.NaturalMovie
.Ball*
patch.Face
patch.Sync*
reso.Align
patch.Photodiode*
patch.Ephys*
patch.Eye*
patch.Led*
patch.Quality
patch.Estim*
reso.MiniStack
reso.Sync
reso.VolumeSlice
patch.Spikes
patch.EphysStats
nning
patch.TrialSet
patch.Eyephys
patch.Facing
patch.PeriLed
reso.CaOpt
reso.EphysTime
patch.EyeFrame
reso.IndicatorSet
reso.SegmentGlia
patch.Spontaneous
reso.Motion3D
patch.Trial*
reso.TrialSet*
reso.Conditions
reso.ManualSegment
reso.ManualSegmentGlia
reso.OriDesign
reso.AxonMask
reso.Segment
reso.Indicator
reso.OriMap
reso.TrialTraceSet
patch.PeriTrial
reso.Effect
reso.Trial*
reso.Trace*
reso.Axons
reso.TraceGlia*
reso.TrialTrace*
reso.TraceVonMises
reso.ConditionMap
reso.AxonEffect
common.Virus
reso.Cos2Map

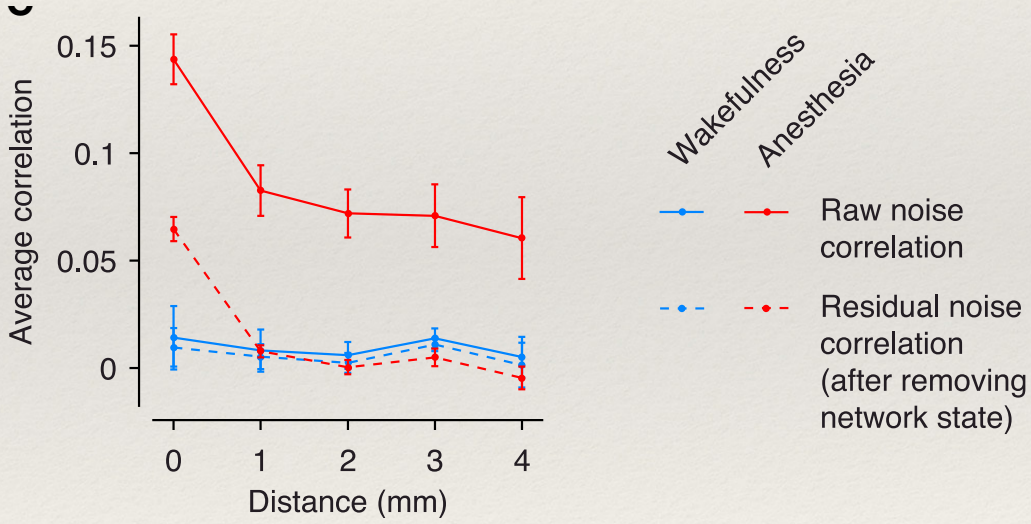

patch.Eye*
patch.Ball*
pupil.EpochOpt
reso.TrialSet
pupil.Phases
reso.TrialTraceSet
pupil.Intervals
pupil.EpochTrialSet
reso.Trial*
reso.Trace*
pupil.EpochTrial*
pupil.EpochVonMisesSet
pupil.BinnedNoiseCorr
pupil.EpochR2
pupil.EpochVonMises*

State Dependence of Noise Correlations in Macaque Primary Visual Cortex

Alexander S. Ecker,[1,2,3,5,*] Philipp Berens,[1,2,3] R. James Cotton,[1] Manivannan Subramaniyan,[1] George H. Denfield,[1] Cathryn R. Cadwell,[1] Stelios M. Smirnakis,[1,4] Matthias Bethge,[2,3,5] and Andreas S. Tolias[1,3,6,*]

Neuron *82*, 235–248, April 2, 2014 ©2014 Elsevier Inc.



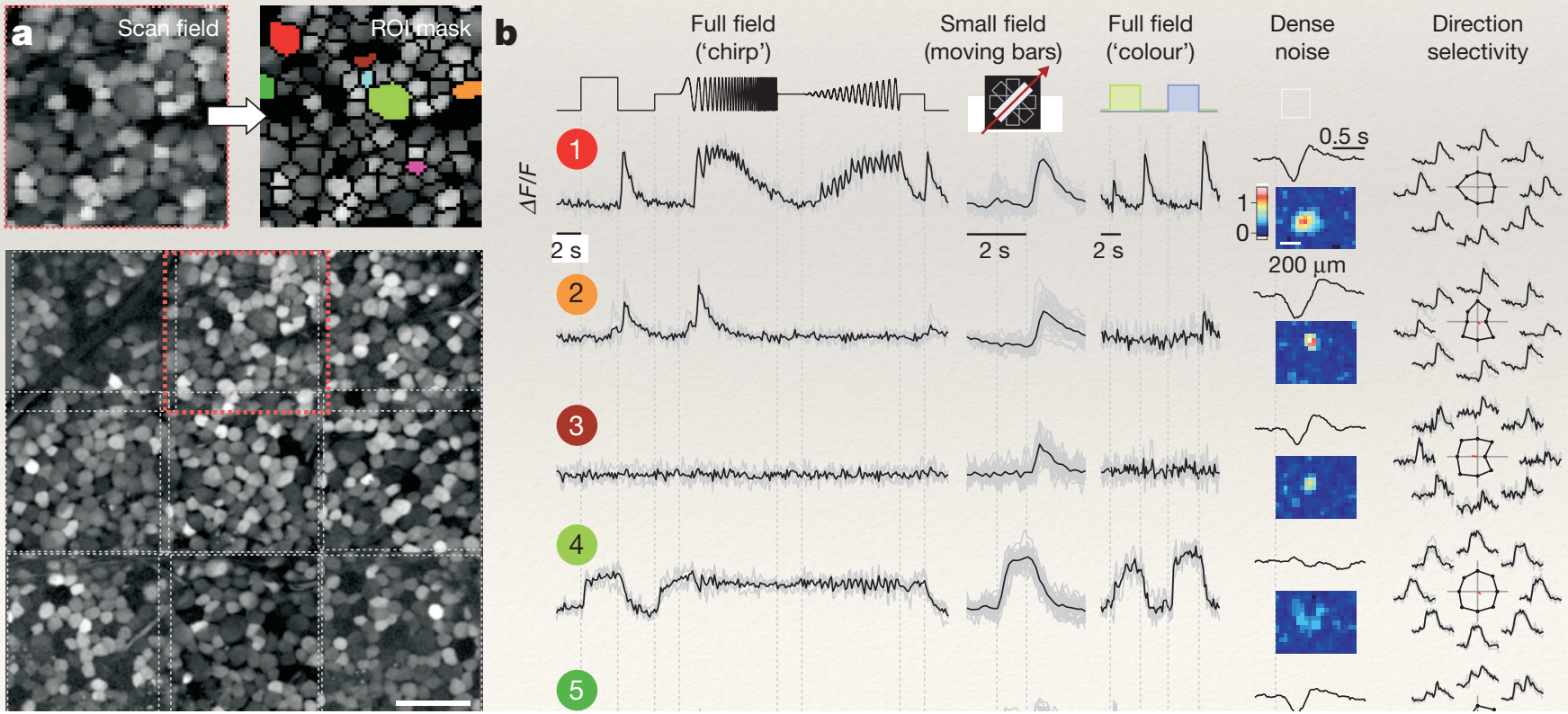


ARTICLE

doi:10.1038/nature16468

The functional diversity of retinal ganglion cells in the mouse

Tom Baden[1,2,3]*, Philipp Berens[1,2,3,4,5]*, Katrin Franke[1,2,3,6]*, Miroslav Román Rosón[1,2,3,6], Matthias Bethge[1,2,5,7] & Thomas Euler[1,2,3]

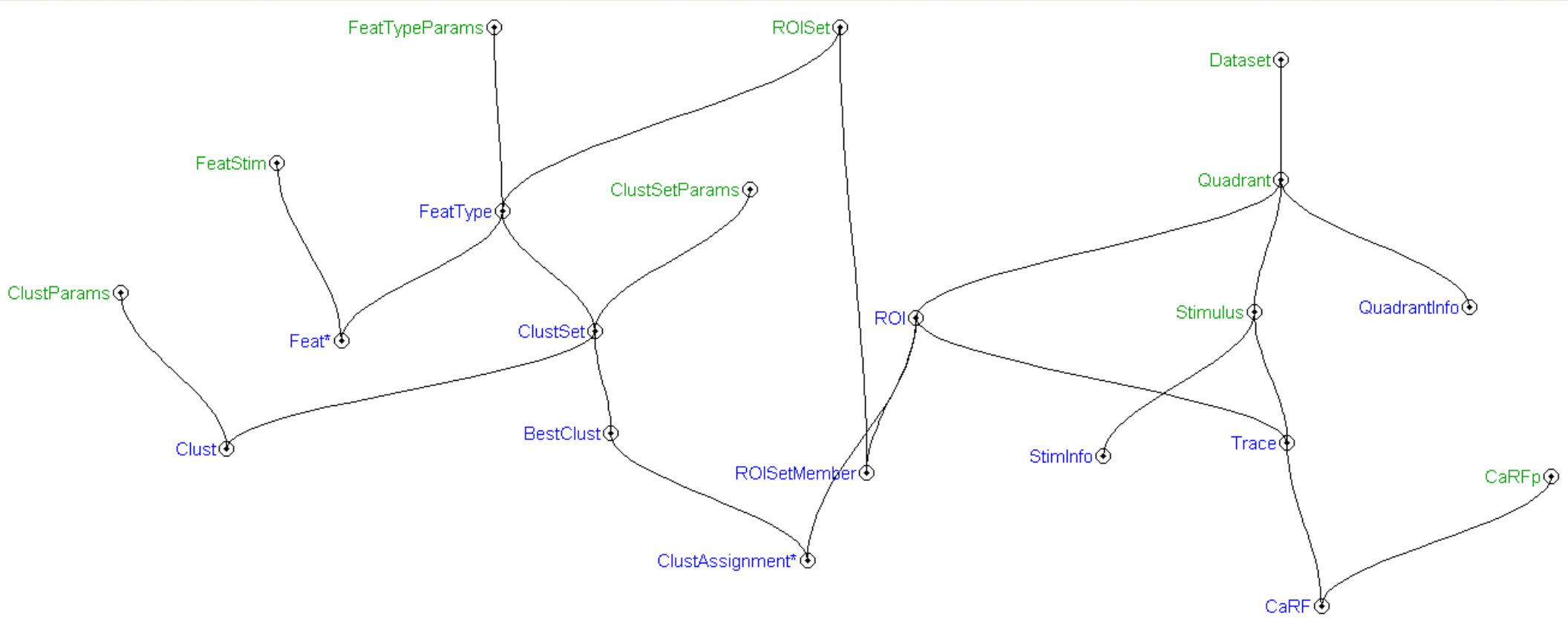


ARTICLE

doi:10.1038/nature16468

The functional diversity of retinal ganglion cells in the mouse

Tom Baden[1,2,3]*, Philipp Berens[1,2,3,4,5]*, Katrin Franke[1,2,3,6]*, Miroslav Román Rosón[1,2,3,6], Matthias Bethge[1,2,5,7] & Thomas Euler[1,2,3]



Resources

- **Reference:**

Yatsenko, D., Reimer, J., Ecker, A.S., Walker, E.Y., Sinz, F., Berens, P., Hoenselaar, A., Cotton, R.J., Siapas, A.S. and Tolias, A.S., 2015. **DataJoint: managing big scientific data using MATLAB or Python.** bioRxiv, p.031658.

- **General information**: datajoint.github.com
- **MATLAB**: github.com/datajoint/datajoint-matlab
- **Python**: github.com/datajoint/datajoint-python

Acknowledgements

Alex Ecker

Edgar Walker

Fabian Sinz

Andreas Hoenselaar

Philipp Berens

Jacob Reimer

Andreas Tolias

Thank you