Dataset Structure and File Specification

NINAI

July 2023

1 Two Photon Processed Data and Metadata

The scan and stack metadata, synchronized stimulus movies, synchronized behavioral traces, cell segmentation masks, calcium traces, and inferred spikes are stored in DataJoint tables (https://datajoint.io/, Yatsenko et al. (2015)). This file contains a Docker image in a tar archived uncompressed format (.tar) that has a MYQL v5.7 database and schema preloaded with processed data. The attributes of each table are defined below, with primary keys entered in bold and dependent attributes in plain text, along with the data type and a short description. A second Docker image is prebuilt with DataJoint (Python) installed for access via Jupyter notebook. Please see https://github.com/cajal/microns-nda-access for instructions on setup.

The following data types are represented:

int	32-bit integer number, ranging from -2,147,483,648 to 2,147,483,647.
int unsigned	32-bit positive integer, ranging from 0 to 4,294,967,295.
smallint	16-bit integer number, ranging from -32,768 to 32,767.
tinyint	8-bit integer number, ranging from -128 to 127.
float	single-precision floating-point number.
double	double-precision floating-point number.
decimal(N,F)	fixed-point number with N total decimal digits and F fractional digits.
longblob	arbitrary numeric array (e.g. matrix, image, structure), up to 4 GiB in size. Numeric arrays are compatible between MATLAB and Python (NumPy).
char(N)	a character string up to N characters (but always takes the entire N bytes to store).
varchar(N)	text string of arbitrary length up to N characters.

The schema structure is described below, and can also be found at: https://github.com/cajal/microns_phase3_nda

1.1 Scan

Scan metadata including filename, frame count, imaging field count, and temporal frequency.

session	smallint	session ID
scan_idx	smallint	scan ID
nframes	int	number of frames per scan
nfields	tinyint	number of fields per scan
fps	float	frames per second (Hz)

1.2 ScanInclude

Collection of 14 preferred scans of higher quality routinely used for analysis.

session	smallint	session ID
scan_idx	smallint	scan ID

1.3 Field

Scan field metadata including location, shape, and spatial resolution. Field motor coordinates refer to microscope inputs used to target field location, and vary slightly from session to session. XYZ coordinates here refer to the field center. Increasing X = posterior to anterior. Increasing Y = lateral to medial. Increasing Z = pia to white matter.

session	smallint	session ID
scan_idx	smallint	scan ID
field	smallint	field number
px_width	smallint	field pixels per line
px_height	smallint	lines per field
um_width	float	field width (microns)
um_height	float	field height (microns)
field_x	float	field x motor coordinates (microns)
field_y	float	field y motor coordinates (microns)
field_z	float	field z motor coordinates (microns)

1.4 RawManualPupil

Video of the left eye and face was collected at ~20 Hz, and were semi-automatically segmented to label pixels containing the pupil. An ellipse was fit to the pupil, and the center, major and minor radii were extracted. Increasing X = anterior to posterior. Increasing Y = inferior to superior. Pixel x and y coordinates vary between sessions due to differences in cropping, as well as small variations in camera placement.

session	smallint	session ID
scan_idx	smallint	scan ID
pupil_min_r	longblob	vector of pupil minor radii (pixels)
pupil_maj_r	longblob	vector of pupil major radii (pixels)
pupil_x	longblob	vector of pupil x positions (pixels)
pupil_y	longblob	vector of pupil y positions (pixels)
pupil_times	longblob	vector of times relative to scan start (seconds)

1.5 ManualPupil

Traces from RawManualPupil were low-pass filtered with a hamming window to the scan frame rate and linearly interpolated to field 1 scan frame times.

session	smallint	session ID
scan_idx	smallint	scan ID
pupil_min_r	longblob	vector of pupil minor radii synchronized with field 1 frame times (pixels)
pupil_maj_r	longblob	vector of pupil major radii synchronized with field 1 frame times (pixels)
pupil_x	longblob	vector of pupil x positions synchronized with field 1 frame times (pixels)
pupil_y	longblob	vector of pupil y positions synchronized with field 1 frame times (pixels)

1.6 RawTreadmill

Cylindrical treadmill rostral-caudal position was extracted with a rotary optical encoder at ~60-100 Hz and converted into velocity.

session	smallint	session ID
scan_idx	smallint	scan ID
treadmill_velocity	longblob	vector of treadmill velocities (cm/s)
treadmill_timestamps	longblob	vector of times relative to scan start (seconds)

1.7 Treadmill

Treadmill velocities were low-pass filtered with a hamming window to the scan frame rate then linearly interpolated to scan frame times

session	smallint	session ID
scan_idx	smallint	scan ID
treadmill_velocity	longblob	vector of treadmill velocities synchronized with field 1 frame times (cm/s)

1.8 ScanTimes

Scan frame collection times for the first pixel of the first imaging field, relative to the start of the scan.

session	smallint	session ID
scan_idx	smallint	scan ID
frame_times	longblob	stimulus frame times for field 1 of each scan, relative to scan start (length = nframes)
ndepths	smallint	number of imaging depths recorded for each scan

1.9 Stimulus

Stimulus movies were constructed as in Stimulus Movies, but were low-pass filtered with a hamming window to the scan frame frequency, then interpolated at scan frame times.

session	smallint	session ID
scan_idx	smallint	scan ID
movie	longblob	stimulus images synchronized with field 1 frame times (H x W x F matrix)

1.10 Trial

A stimulus condition is a unit of movie stimulus, and a stimulus trial is an instance of that condition being shown to the animal, as well as the activity and behavior that are recorded during that period. This table contains the stimulus condition type and start / stop index for each trial in the full stimulus movie stored in the nda.Stimulus table. It also contains the presentation time of each frame of the stimulus movie relative to the scan start (frame_times), as well as the first and last frame times (start_frame_time, end_frame_time). Each unique condition has its own condition_hash. Detailed information for each condition_hash is located in the corresponding Clip table, Monet2 table, or Trippy table, depending on the condition type.

session	smallint	session ID
scan_idx	smallint	scan ID
trial_idx	smallint	index of trial within stimulus
type	varchar(16)	type of stimulus trial
start_idx	int unsigned	index of field 1 scan frame at start of trial
end_idx	int unsigned	index of field 1 scan frame at end of trial
start_frame_time	double	start time of stimulus frame relative to scan start (seconds)
end_frame_time	double	end time of stimulus frame relative to scan start (seconds)
stim_times	longblob	full vector of stimulus frame times relative to scan start (seconds)
condition_hash	char(20)	120-bit hash (The first 20 chars of MD5 in base64)

1.11 Clip

This stimulus type is composed of 10 second clips from cinematic releases, Sports-1M dataset, or custom rendered first person POV videos in 3D environment in Unreal Engine. Notably, since the dependent attributes have been reorganized for this release, rehashing these attributes will no longer yield the same condition_hash. All clips included here are 30 fps.

condition_hash	char(20)	120-bit hash (The first 20 chars of MD5 in base64)
movie_name	char(250)	full clip source
duration	decimal(7,3)	duration of clip (seconds)
clip	longblob	clip used for stimulus (T x H x W)
short_movie_name	char(15)	clip type (cinematic, sports1m, rendered)
fps	float	clip framerate during presentation

1.12 Monet2

This stimulus type is generated from smoothened Gaussian noise and a global orientation and direction component. Entries in this table differ in rng_seed and as a result differ in the directions vector (order in which directions were shown) and in the final movie. All other parameters (fps, duration, etc) are shared across all conditions. Directions are stored in degrees and progress clockwise, with 0 degrees indicating upwards motion and 90 degrees indicating rightward motion.

condition_hash	char(20)	120-bit hash (The first 20 chars of MD5 in base64)
fps	decimal(6,3)	display refresh rate (Hz)
duration	decimal(6,3)	trial duration (seconds)
rng_seed	double	random number generator seed
blue_green_saturation	decimal(4,3)	0 = grayscale, 1=blue/green
pattern_width	smallint	width of generated pattern (pixels)
pattern_aspect	float	aspect ratio of generated pattern
temp_kernel	varchar(16)	temporal kernel type (hamming, half-hamming)
temp_bandwidth	decimal(4,2)	temporal bandwidth of the stimulus (Hz)
ori_coherence	decimal(4,2)	1=unoriented noise. pi / ori_coherence = bandwidth of orientation kernel.
ori_fraction	float	fraction of stimulus with coherent orientation vs unoriented
ori_mix	float	mixin-coefficient of orientation biased noise
n_dirs	smallint	number of directions
speed	float	units/s motion component, where unit is display width
directions	longblob	computed directions of motion (deg)
onsets	blob	computed direction onset (seconds)
movie	longblob	rendered uint8 movie (H x W x 1 x T)

1.13 Trippy

This stimulus type is generated using the cosine of a smoothened noise phase movie with content that varies locally in orientation, direction, spatial frequency, and temporal frequency. Entries in this table differ in the rng_seed and as a result differ in the underlying randomly generated phase movie (packed_phase_movie) and in the final movie. All other parameters (fps, duration, etc) are shared across all conditions.

condition_hash	char(20)	120-bit hash (The first 20 chars of MD5 in base64)
fps	decimal(6,3)	display refresh rate (Hz)
rng_seed	double	random number generate seed
packed_phase_movie	longblob	phase movie before spatial and temporal interpolation
tex_ydim	smallint	texture height (pixels)
tex_xdim	smallint	texture width (pixels)
duration	float	trial duration (seconds)
xnodes	tinyint	x dimension of low-res phase movie
ynodes	tinyint	y dimension of low-res phase movie
up_factor	tinyint	spatial upscale factor
temp_freq	float	temporal frequency if the phase pattern were static (Hz)
temp_kernel_length	smallint	length of Hanning kernel used for temporal filter. Controls the rate of change of the phase pattern.
spatial_freq	float	approximate max. The actual frequencies may be higher. (cy/point)
movie	longblob	rendered movie (H x W x T)

1.14 RasterCorrection

Raster phase correction applied to each scan field.

session	smallint	session ID
scan_idx	smallint	scan ID
field	smallint	field number
raster_phase	float	difference between expected and recorded scan angle

1.15 MotionCorrection

Motion correction applied to each scan frame.

session	smallint	session ID
scan_idx	smallint	scan ID
field	smallint	field number
y_shifts	longblob	y motion correction shifts (pixels)
x_shifts	longblob	x motion correction shifts (pixels)
y_std	float	standard deviation of y shifts (um)
x_std	float	standard deviation of x shifts (um)

1.16 MeanIntensity

Average of field pixel intensity per scan frame.

session	smallint	session ID
scan_idx	smallint	scan ID
field	smallint	field number
intensities	longblob	mean intensity

1.17 SummaryImages

Summary images for each scan field. Average calculated per pixel across scan frames. Correlation image calculated from the normalized cross-correlation between each pixel and its 8 neighbors. All summary images calculated after correction for field brightness.

session	smallint	session ID
scan_idx	smallint	scan ID
field	smallint	field number
correlation	longblob	correlation image
average	longblob	average image

1.18 Stack

Stack metadata including location, shape, spatial resolution, and z coordinates of the first slice. Stack motor coordinates refer to microscope inputs used to target the stack location. XYZ coordinates refer to the volume center. Increasing X = posterior to anterior. Increasing Y = lateral to medial. Increasing Z = pia to white matter, with negative values corresponding to distance above the pia.

stack_session	smallint	session ID
stack_idx	smallint	stack ID
motor_z	float	center of volume in the motor coordinate system (microns, cortex at z=0)
motor_y	float	center of volume in the motor coordinate system (microns)
motor_x	float	center of volume in the motor coordinate system (microns)
px_depth	smallint	number of slices
px_height	smallint	lines per frame
px_width	smallint	pixels per line
um_depth	float	depth (microns)
um_height	float	height (microns)
um_width	float	width (microns)
surf_z	float	depth of first slice - half a z step (microns, cortex is at z=0)

1.19 Registration

Scan fields were registered into the structural two-photon stack using an affine transformation matrix estimated via gradient ascent on the correlation between the sharpened average image and the predicted location in the stack.

stack_session	smallint	session ID
stack_idx	smallint	stack ID
session	smallint	session ID
scan_idx	smallint	scan ID
field	smallint	field number
a11	float	row 1, column 1 of the affine matrix (microns)
a21	float	row 2, column 1 of the affine matrix (microns)
a31	float	row 3, column 1 of the affine matrix (microns)
a12	float	row 1, column 2 of the affine matrix (microns)
a22	float	row 2, column 2 of the affine matrix (microns)
a32	float	row 3, column 2 of the affine matrix (microns)
reg_x	float	x translation (microns)
reg_y	float	y translation (microns)
reg_z	float	z translation (microns)
score	float	cross-correlation score (-1 to 1)
reg_field	longblob	extracted field from the stack in the specified position

1.20 Coregistration

3D transformation solutions between two-photon stack and EM stack, generated by the Allen Institute (https://github.com/AllenInstitute/em_coregistration/tree/phase3). Functions for applying the transformation and tutorial for how and when to use each transform_type included at https://github.com/cajal/microns_phase3_nda.

stack_session	smallint	session ID
stack_idx	smallint	stack ID
transform_id	int	id of the transform
version	varchar(256)	coordinate framework
direction	varchar(16)	direction of the transform (EMTP: EM to 2P, TPEM: 2P to EM)
transform_type	varchar(16)	linear (more rigid) or spline (more nonrigid)
transform_args	longblob	parameters of the transform
transform_solution	longblob	transform solution

1.21 Segmentation

Scans were raster and motion corrected, and segmentation by constrained non-negative matrix factorization was performed via the CaImAn package (https://github.com/flatironinstitute/CaImAn, Giovannucci et al. (2019)). Mask_ids are unique per field.

session	smallint	session ID
scan_idx	smallint	scan ID
field	smallint	field number
mask_id	smallint	mask ID, unique per field
pixels	longblob	indices into the image in column major (Fortran) order
weights	longblob	weights of the mask at the indices above

1.22 Fluorescence

Fluorescence traces per segmentation mask, before spike extraction.

session	smallint	session ID
scan_idx	smallint	scan ID
field	smallint	field number
mask_id	smallint	mask ID, unique per field
trace	longblob	fluorescence trace

1.23 ScanUnit

Functional unit metadata, including unit_id (unique per scan). Motor coordinates inherited from scan field motor coordinates. Unit ms_delay represents the time difference from the first pixel of the first field and the time that unit's fluorescence is recorded, and is approximated by linear attribution of the inter-frame period (1 / frame rate) to pixels in each field in acquisition order, then taking the weighted average using the segmentation mask pixel weights.

session	smallint	session ID
scan_idx	smallint	scan ID
unit_id	int	unique per scan
field	smallint	field number
mask_id	smallint	mask ID, unique per field
um_x	smallint	centroid x motor coordinates (microns)
um_y	smallint	centroid y motor coordinates (microns)
um_z	smallint	centroid z motor coordinates relative to surface of the cortex (microns)
px_x	smallint	centroid x pixel coordinate in field (pixels)
px_y	smallint	centroid y pixel coordinate in field (pixels)
ms_delay	smallint	delay from start of frame (field 1 pixel 1) to recording of this unit (milliseconds)

1.24 UnitHash

Assigns hash and semantic string to each unique session - scan_idx - unit_id triplet.

session	smallint	session ID
scan_idx	smallint	scan ID
unit_id	int	unique per scan
hash	varchar(64)	unique hash per unit
str_key	varchar(64)	session, scan_idx and unit_id as string

1.25 Activity

Activity was inferred from fluorescence traces using noise constrained deconvolution, as in (Giovannucci et al., 2019).

session	smallint	session ID
scan_idx	smallint	scan ID
unit_id	int	unique per scan
trace	longblob	spike trace

1.26 StackUnit

Each unit centroid is assigned 3D coordinates in the stack according to the field affine registration. The center of the stack in the motor coordinate system is inherited from acquisition and is stored in nda.Stack. Stack coordinates (stack_x, stack_y, stack_z) shift the origin to the top, back, left corner of the stack by subtracting the motor coordinate center and adding 1/2 of the height, width, depth to the appropriate axis. Both coordinate systems are in microns and differ only in a constant offset.

stack_session	smallint	session ID
stack_idx	smallint	stack ID
session	smallint	session ID
scan_idx	smallint	scan ID
field	smallint	field number
unit_id	int	unique per scan
motor_x	float	centroid x stack coordinates with motor offset (microns)
motor_y	float	centroid y stack coordinates with motor offset (microns)
motor_z	float	centroid z stack coordinates with motor offset (microns)
stack_x	float	centroid x stack coordinates (microns)
stack_y	float	centroid y stack coordinates (microns)
stack_z	float	centroid z stack coordinates (microns)

1.27 AreaMembership

Individual neurons inherit area labels based on registered xy stack coordinates relative to a manually annotated retinotopic map that was automatically registered into the stack.

session	smallint	session ID
scan_idx	smallint	scan ID
unit_id	smallint	unit id in scan
brain_area	char(10)	visual area membership of unit

1.28 MaskClassification

Segmented masks are classified as either soma or artifact using a Convolutional Neural Network from the CaImAn package (https://github.com/simonsfoundation/CaImAn).

session	smallint	session ID
scan_idx	smallint	scan ID
field	smallint	field number
mask_id	smallint	mask ID, unique per field
mask_type	varchar(16)	classification of mask as soma or artifact

1.29 Oracle

We used six natural movie conditions that were present in all scans and repeated 10 times per scan to calculate an "Oracle score" representing the reliability of the trace response to repeated visual stimuli. This score was computed as the jackknife mean of correlations between the leave-one-out mean across repeated stimuli with the remaining trial. Prior to calculation, segmentation masks were evaluated by a CNN classifier trained to predict probability false positive segmentation, and probable artifacts (~8%) were excluded from analysis (see MaskClassification).

session	smallint	session ID
scan_idx	smallint	scan ID
unit_id	int	unique per scan
trials	int	number of trials used
pearson	float	per unit oracle pearson correlation over all movies

1.29.1 Changelog

v7 (7/29/2021): first public version
v8 (2/8/2023): updates include:

- renamed FrameTimes table to ScanTimes
- updated some attribute descriptions for clarity
- added new tables RasterCorrection and MotionCorrection, containing the corrections applied to the raw functional scan data to produce the scan movies included in the release.
- removed references to L6 norm summary images from table SummaryImages.
- added new table UnitHash to offer unique hash or semantic string key uniquely representing each unique session - scan_idx - unit_id triplet
- concatenated stimulus movies in Stimulus table processed from updated v4 Stimulus movies
- frame_times in Trial table renamed to stim_times to avoid confusion with scan frame times, and offset of stim_times w/r/t first scan frame recomputed, resulting in small differences to v7 in frame time offset (1-10ms).
- start_index and end_index in Trial table recomputed to avoid sequential trials ending/starting on the same frame index.
- added fps attribute to Clip table for convenience (all values still 30 Hz).

References

A. Giovannucci, J. Friedrich, P. Gunn, J. Kalfon, B. L. Brown, S. A. Koay, J. Taxidis, F. Najafi, J. L. Gauthier, P. Zhou, B. S. Khakh, D. W. Tank, D. B. Chklovskii, and E. A. Pnevmatikakis. CaImAn: An open source tool for scalable calcium imaging data analysis. *Elife*, 8:e38173, 2019.

D. Yatsenko, J. Reimer, A. S. Ecker, E. Y. Walker, F. Sinz, P. Berens, A. Hoenselaar, R. J. Cotton, A. S. Siapas, and A. S. Tolias. DataJoint: managing big scientific data using MATLAB or python. Technical report, Bioinformatics, Nov. 2015.