**Scripts used for Nichols et al. Science 2017.**

Nichols, A. L. A., Eichler, T., Latham, R., & Zimmer, M. (2017). A global brain state underlies C. elegans sleep behavior. Science (New York, NY), 356(6344), eaam6851. <http://doi.org/10.1126/science.aam6851>

Disclaimer: The following is a description of which scripts were used for what data analysis in Nichols, *et al*. 2017. I taught myself Matlab during my PhD and some of these scripts may not be easily accessible or as elegant as they could be. I’ve done my best to comment the sections. Even after my best efforts to avoid them, mistakes are uncovered, the ones which I’m aware of are included in a section at the end. If need you can contact me on my permanent email address: [annika.nichols@uqconnect.edu.au](mailto:annika.nichols@uqconnect.edu.au)

Neuronal time series from whole brain imaging datasets can be found at: <https://osf.io/kbf38/>

For behavioural data, please contact Manuel Zimmer (manuel.zimmer@univie.ac.at).

**Pharynx tracking (Figure 1A,B, Sfigure 1A,B,C,D)**

Data is in the form of tiff stacks. The following scripts were used to generate the data for the designated figures.

*PharynxTrackerBatch.m* uses *pharynxTracker\_convex.m* and runs over all folders (each folder has a single avi movie which was made from inversing the black/white colours of the original tiff stack using FIJI) and outputs a mat file for each folder with: *'Tracks', 'LevelThresh','Levelbackground','options'.*

Uses *BackgroundProduction\_V4* to generate background (commented out).

*pharynxTrackerSpeed\_Slide.m* gets data across all recordings in one condition and creates a ‘.*mat’* file with: *'Speed','BinSpeed','AllSlideSpeed','binning','fps','SlideBin','MotionState'*

Motionstate is what was used in Figure 1B.

*PharynxPlotHist.m* was used to generate the data for the histogram plots in Sfigure 1A-D. Used *HistXedges* and *nanmean(BinnedSpeedLogEd)* and plotted them in Prism7*.*

*PharynxMovieMaker.m* was used to make Smovie1

Data: /zimmer/Annika\_Nichols/Nichols\_2017\_Science/\_1-Sleeping\_in\_clumps\_F1AB-SF1AB

**Worm behavioural scripts:**

Figure 1CD, figure 2A,C,E, and scatter/whisker plots: Figure 1EF, figure 2B,D,F. Sfigure 1E-M, Sfigure 2, Sfigure 3A,B, Sfigure 5A-H, Sfigure 7.

Tracking: These scripts we not written by me but many other people.

I used the same compiled scripts as Tomas, and when I asked for them he sent me this: /zimmer/Annika\_Nichols/Nichols\_2017\_Science/MATLAB/\_\_This isTheTrackingScript!

More information on this can be found in this folder: “Behavioural tracking additional information”

Analysis: Many of these scripts were based on scripts written by Manuel Zimmer and Tomas Eichler (Tomas also asked some colleagues to help with some points).

The versions which I used for the paper can be found in the folder: “SleepQuantScripts”

1. Make background using: BackgroundProduction\_V5
2. Submit to cluster to be tracked. This outputs als files (more detailed information can be found in the folder: “Support information on scripts and experiments”
3. Use *SleepActivityPlot.mat* to plot lethargus curves as used in figures: Sfigure J,K,L.
4. als files were then manually divided up into preletharus, lethargus or postlethargus, or not included for the subsequent analysis.
5. Responses for each of these three states were averaged using: *AveragingSleepResponses.mat*. As seen in figure 1E, 2A,C,E, Sfigure 2A,D,G, 3A, 5A,C,D
6. To find the measurements of the behaviour in certain periods, *SleepQuantStats.mat* was used. Which was then used by: *SleepStats\_MeanReadout.mat* to get the measurement for each experiment. Figure 1F, 2B,D,F, Sfigure 2B, 3B, 5B,H, 7AB

For the *npr-1* control experiments, I used *SleepQuantStats\_input.mat* and then defined which *npr-1* experiments to use as the control (see: ‘*Data\_and\_NPR1\_ID\_Controls.xlsx’*).

I copied the readouts from *SleepStats\_MeanReadout.mat* to prism for plotting (before moving to illustrator) and to analyse the results.

Speed and turn plots are also generated by *AveragingSleepResponses.mat*. For the statistics I used *SleepStats\_MeanReadout\_FTspd* to readout the RealMeansTurnNonLet (or Let), RealMeansSpdNonLet (or Let) and firstBinAbovemeantwentypcNonLet (or Let) and then used Prism.

TA: anything with this means that it is “time adjusted”. I realised that the movies are not 90min long but slightly shorter, meaning that subsequent oxygen shifts are slight delayed. See: *Time adjustments.docx*. This did not make much difference in the statistics or plots, it’s only adjusted for figure S2.

Sfig1ef, 2D histograms were generated by: *normSpdEccenHist2Sleep.m*

Sfig1hi, plots were made by: *PlotSpdEccenMS\_perTrack.m*

Data: /zimmer/Annika\_Nichols/Nichols\_2017\_Science/\_3-Lethargus assays

**Whole-brain-imaging analysis**

For Figure 3A,B *WormHeadPlot\_v2* (but use *WormHeadPlot\_v3* as it’s slightly updated) was used. Need to specify start point (secToPlot) and the time from that point to include in the MIP (secRange).

Figure 3C-F top panels used *MakeHeatMap.m*, go through each section as needed. Bottom panels used *awbRMSAveragePlot.m*. It runs on whatever analysis you input, as generated by: *awbRMSAveragePlot.m*. Make sure you load the analysis struct and then call it awbRMSAverage. Check that the analysis is the one you want, I used global mean. Can use *awbBrainStateBar.m* to plot a bar with the state colours.

I used the *wb* toolkit written by Saul Kato for tracking, measurement and identification of the neurons. I also used many of its tools including the *wbStateTraceAnnotator* to determine the 4 states of the neurons. See also Kato, et al. 2015, Cell. <https://doi.org/10.1016/j.cell.2015.09.034>. There is a private github repsoistory: <https://github.com/saulkato/wb>. I included a copy of this folder (wb\_11Feb2016) in the Matlab folder.

Figure 4A,B

Quiescence classification: *awbQAstateClassifier.m* is used to generate *QuiescentState.mat* in the Quant folder where it is run. This classification was used for many subsequent analyses. Plotted using: *QuiesceStatePlotMultiple.m.* Also, Sfig6.

Figure 4c,d, calculating the fraction quiescent in each bin used: *BatchPercentQorA.m.* Then used Prism7

Figure 4e,f

Mean distribution calculation, use *BatchawbPowerDist.m* (uses *awbPowerDist.m)* and then copied to prism. Used *PermuteCumRMS.m* to conduct permutation test. Could also use the output of this to make the graphs.

Figure 4g,h,i,j,k,l,m, Sfig11 distributions and the distances are calculated by awbActivityDistFQR.m (equivalent to: awbActivityDistFQRmean.m, but this one doesn’t have the plotting stuff I was using beneath it). Data was copied to Prism7 for plotting except for the distribution graphs which were produced directly by this script. Permutation was performed by: *awbPermuteActivityDistFQR.m*, also gives out true distance as: trueDistances.

Figure 5 a-d (also, Sfig 6)

PCA was calculated using: *wbComputePCA\_demo\_AN\_v2\_2016.m* which uses Saul’s wb PCA computation method. Phaseplots were plotted using: *awbPhasePlotFigure*.m.

Figure 5e,f,g,h Sfig12

To extracting transitions use: *awbStateTransBatch.m* check all the input parameters. This generates: StateTrans\_AVALrisetriggered\_0sTo1080s\_15sThres\_QFoff\_.mat (or something like it). There’s a lot of different data transformations you can do at the bottom of this script. I then transferred the data to prism for plotting. Used *vectorDistPermuteBatch.m* to do the permutations (table S1).

Note: for the transition analyses, Anton Parinov has adapted and extended many of these scripts.

Figure 5i-p, Used: *awb3StateTransFrequency.m*. which also plots the graph (e.g. ForwardTransProb10pc\_log3bin\_Turn.ai) and then *permuteTSTransFrequency.m* to run permutation.

Sfigure 8 Oxygen sensation, used *BatchawbNeuronFull.m* to get and interpolate the traces of the neurons. You can find the generated structs here: /zimmer/Annika\_Nichols/Nichols\_2017\_Science/\_5-imaging analysis/

Also, used: *awbNeuronFull\_peakProcess.m* for processing and *awbPlotNeuronSEM.m* for plotting.

Sfigure 9 Oxygen sensation – single traces used: *QuickPlotIndividualTraces.m*

Sfigure 10. Quantitation of reversals. *awbBatchReversals.m* uses *awbReversals.m* to find AVA rise to determine the number of reversal events across time.

Movie S2, used: MakePhasePlotHeatMapMovie\_v2.m

**Uncovered mistakes:**

Anton P. uncovered that I had made a wrong call in the Find function of *FindTransition.m*. It therefore took only the first 49 transitions instead of all of them. This shouldn’t have made much of a difference to my results as its only effect was to exclude transitions over the 49th and in my case most neurons had much less than 49 transitions.

Problem (mistake underlined):

Transitions=find(Rise(:,aaa),'1');

Should be:

Transitions=find(Rise(:,aaa));

This mistake occurs in a couple of places in the script.