I started by making sure the code replicates the data we had produced for S1 and S1naive - it does, although the stochastic nature of the tests mean that results aren’t identical.

Fig 1 shows new equivalent of fig4a from the paper:

|  |  |  |
| --- | --- | --- |
| S1 (289 neurons) | S1naive (318 neurons) | PPC (129 neurons) |
| Fig1 PAIRS measure for different brain areas | | |

Fig 2 shows Equivalent of fig 4b from the paper. These use less neurons than the fig4a PAIRS tests, because selection criteria is that: for each neuron, trials removed so that ratios of LICK/NOLICK is at least 25%/75%. If that then results in less than 100 trials, the neuron is discarded from analysis.

Black dots: neurons with significant performance on both lick versus no lick and GO versus NOGO.

Blue: neurons with significant performance on lick versus no-lick.

Purple: neurons with significant performance on GO versus NOGO.

Grey: neurons with significant performance on neither

(Neurons were deemed to support classification at a significant level of performance if they performed better than 95 of the 100 surrogates).

These charts are produced automatically by the python script, but the data is also stored in the excel sheet so that we can produce them manually and play about with them.

|  |  |  |
| --- | --- | --- |
| S1 (284 neurons)    Note that the spreadsheet has data for only 272 neurons. This is because we decided to remove a further 12 neurons which had a small number of trials, suggesting that something had gone wrong in the data collection process: sessionID 406/20180409. | S1naive (193 neurons)  All 193 neurons contained in spreadsheet | PPC (76 neurons)  Following original rules, but seem below for comment |
| Fig 2 Classification performance (percent correct) supported by individual neurons | | |

The results for PPC produce a large number of neurons whose lick accuracies are ~75%. These all come the first session of mouse 588, session 20181003. Investigating the reason for this, I find that there is an issue with the classification accuracy algorithm:

* Recall that the classification accuracy is calculated as follows: “Training was performed using the sklearn SVM library and classification performance assessed using 5-fold cross validation. A linear kernel was used, and several regularisation parameters tried (.001, 0.01, 0.1, 1) before choosing the one with best cross validated score. The training procedure was repeated for 100 surrogates generated by shuffling trial labels. Neurons were deemed to support classification at a significant level of performance if they performed better than 95 of the surrogates ”. So for each neuron we should produce something along the lines of fig3a, allowing us to calculate a rank.
* For these particular neurons, if the regularisation parameter is 0.001, the classifier is coming up with the same lick accuracy for all 100 shuffles i.e. it is failing to act as a classifier and rank of unshuffled data is 50. So we end up with a distribution like fig3b. However, this regularisation parameter gives the highest score and so this is the classifier which is chosen by the python script. I’m not sure exactly what is going wrong but it is clearly doing something wrong since a) many neurons are coming up with the same lick accuracy and b) these neurons come up with the same accuracy across all shuffles. [I haven’t been able to examine exactly what is going on, but it is worth noting that, in the sklearn implementation, the regularisation parameter C is applied to the loss function – so low values of C mean a small penalization for any misclassified data and will result in a simple model, possibly underfitted.]
* To resolve this, I simply restrict the allowed regularisation parameters to be only (0.01, 0.1, 1). [Alternatively, I could also have run with original allowed parameters but rejected any configuration where shuffling fails to change the classification accuracy.] Having done this, we now get the classification performance as show in fig 4a

|  |  |
| --- | --- |
|  |  |
| Fig3a. for a single neuron, comparing trial classification accuracy of data with genuine labels to shuffled labels | Fig3b. example distribution of classifiers where the same score is obtained across all 100 shuffles. |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Fig 4a. PPC classification performance after rejecting classifiers which fail to discern between shuffles – 76 neurons | Fig 4b PPC classification performance – mouse 588 only 58 neurons | Fig 4c PPC classification – excluding the first session of mouse 588, which is 14 neurons. So 62 neurons |

When we met, we discussed different criteria for which PPC neurons to include:

* We could follow the same rule as for the other brain areas (make sure there is at least a 25%/75% split and the only include if that results in more than 100 trials). This results in 76 neurons being accepted (this is what is shown in the fig4a). Note that this uses all the sessions from mouse 588 and 2 sessions from mouse 756
* Alternatively just use mouse 588 sessions – fig4b. This would reduce number of neurons from 76 to 58 and gives the chart in fig4b
* I’ve put the data into excel and implemented a way for you to decide which neurons to include or exclude in the plot. Hopefully it should be clear how to operate it.
* It’s interesting to note that practically all the neurons with high GO accuracies (ie purple or black colouring in the charts) are in the first session of mouse 588. So if these are ignored we get chart fig4c. I’m not sure whether this implies there is something wrong with the classifier (recall that this is the session which was affected by low regularisation parameter) or whether there is something strange about that session? This needs further consideration:
  + Would we expect PPC to classify GO and LICK well in the first session but then deteriorate in later sessions?
  + Being the first session, was there something strange about the protocol on that day?
  + Was the mouse already trained at that session?
  + Like almost all the PPC data, it is concerning that the mouse licked on a very high proportion of trials (about 85%) , suggesting it didn’t know what was going on.