```
(***Input identifying information***)

In[•]:= date = ToString[Evaluate[Input["Input the date of the experiment"]]]

Out[•]= 071223

In[•]:= mouse = ToString[Evaluate[Input["Input the mouse identity (e.g. Mouse123)"]]]

Out[•]= Mouse23666

In[•]:= sessionNum = Evaluate[Input["Input the session number"]]

Out[•]= 2

In[•]:= (***Import pupil diameter time series***)

In[•]:= pupilTimeSeriesBef =
  ToExpression[Import[StringJoin["S:/Imaging/Garrett/FMB208_2PRig/", date, "/",
     mouse, "/Session", ToString[sessionNum], "/Pupil/", date, "_", mouse, "_",
     "Session", ToString[sessionNum], "_PupilRadiusTimeSeries.txt"], "List"]];

In[•]:= timeVals = Part[#, 1] & /@ pupilTimeSeriesBef;

In[•]:= (***Create pupil diameter time series,
excluding NaN elements from edge detection abnormalities***)

In[•]:= pupilTimeSeries =
  DeleteCases[Table[If[NumberQ@pupilTimeSeriesBef[[n, 2]], {pupilTimeSeriesBef[[n, 1]],
      pupilTimeSeriesBef[[n, 2]]}, Null], {n, 1, Length[pupilTimeSeriesBef]}], Null];

In[•]:= (***Interpolate the pupil diameter time series and resample at 10 Hz***)

In[•]:= pupilInterp = Interpolation[pupilTimeSeries];

In[•]:= pupil10Hz = Table[pupilInterp[t],
   {t, First[pupilTimeSeries][[1]], Last[pupilTimeSeries][[1]], 0.1}];

In[•]:= pupil10HzTimeSeries = Partition[Riffle[Table[n,
     {n, First[pupilTimeSeries][[1]], Last[pupilTimeSeries][[1]], 0.1}], pupil10Hz], 2];

In[•]:= (***Further,
only keep the dilations and constrictions that occur during non-locomotion,
non-whisking periods***)

In[•]:= quiescentPeriods =
  ToExpression[Import[StringJoin["S:/Imaging/Garrett/FMB208_2PRig/", date, "/",
     mouse, "/Session", ToString[sessionNum], "/LocomotionData/", date, "_", mouse,
     "_", "Session", ToString[sessionNum], "_QuiescentBouts.txt"], "List"]];

In[•]:= (***Import calcium fluorescence traces***)

In[•]:= numAdditionalMovies =
  Length[FileNames["*", File[StringJoin["S:/Imaging/Garrett/FMB208_2PRig/",
       date, "/", mouse, "/Session", ToString[sessionNum], "/Ftraces/"]]]] - 2;
 (*Subtract 2 because the first movie was already imported and the
 other file in the directory is an ROI list*)
```

```
In[•]:= tpFrameTimes =
          Drop[Drop[(Import[StringJoin["S:/Imaging/Garrett/FMB208_2PRig/", date, "/",
                mouse, "/Session", ToString[sessionNum], "/", date, "_", mouse, "_",
                "Session", ToString[sessionNum], "_2PFrameTimes.txt"], "List"]), 16], -1];

In[•]:= numFramesPerAcq = Round[Length[tpFrameTimes] / (numAdditionalMovies + 1)];

In[•]:= tpFrameRate = Round[Mean[numFramesPerAcq / Flatten[Differences /@
                ({First[#], Last[#]} & /@ (Partition[tpFrameTimes, numFramesPerAcq]))]]];

In[•]:= (***For each ROI picked for the session, upload the extracted dF/F0 time series***)

In[•]:= numROIs =
          Length[FileNames["*", File[StringJoin["S:/Imaging/Garrett/FMB208_2PRig/", date, "/",
              mouse, "/Session", ToString[sessionNum], "/dFOverF0TimeSeries/"]]]];

In[•]:= Table[Evaluate@ToExpression[StringJoin["dFFtimeseries", ToString[n]]] =
            ToExpression /@ Import[StringJoin["S:/Imaging/Garrett/FMB208_2PRig/", date,
                "/", mouse, "/Session", ToString[sessionNum], "/dFoverF0TimeSeries/",
                date, "_", mouse, "_Session", ToString[sessionNum], "_",
                "dFoverF0ts_ROI", ToString[n], ".txt"], "List"];, {n, 1, numROIs}];

In[•]:= Table[Evaluate@ToExpression[StringJoin["dFFts", ToString[n]]] =
            Partition[Riffle[(ToExpression[StringJoin["dFFtimeseries", ToString[n]]])[[All, 1]],
              Standardize[(ToExpression[StringJoin["dFFtimeseries", ToString[n]]])[[All, 2]]]],
             2];, {n, 1, numROIs}];

        (*****************************************************************************
         ************************************)
        (***In this part, calculate cross-correlograms between dF/F0 and pupil diameter***)
        (*****************************************************************************
         ************************************)

In[•]:= (***Interpolate the 10 Hz sampled pupil time series***)

In[•]:= pupil10HzInt = Interpolation[pupil10HzTimeSeries];

In[•]:= (***Lowpass filter the dF/F0 traces at 10 Hz***)

In[•]:= Table[
          Evaluate@ToExpression[StringJoin["dFFlpfTS", ToString[n]]] = LowpassFilter[TimeSeries@
              ToExpression[StringJoin["dFFts", ToString[n]]], 10 Hz];, {n, 1, numROIs}];

In[•]:= Table[Evaluate@ToExpression[StringJoin["dFFlpfInt", ToString[n]]] =
            Interpolation[ToExpression[StringJoin["dFFlpfTS", ToString[n]]]];, {n, 1, numROIs}];

In[•]:= (***Only keep quiescent periods at least 9 s long***)

In[•]:= quiescentPeriodsLongBef =
          DeleteCases[Table[If[quiescentPeriods[[n, 2]] - quiescentPeriods[[n, 1]] ≥ 9,
              quiescentPeriods[[n]], Null], {n, 1, Length[quiescentPeriods]}], Null];

In[•]:= (***For each acquisition bout, determine the quiescent periods
         contained within it and then make sure they're at least 9 s long***)
```

```
In[•]:= acqBoutIntervals =
          Interval /@ ({First[#] + 8, Last[#] - 8} & /@ (Partition[tpFrameTimes, numFramesPerAcq]));

In[•]:= testedIntervals = Table[MinMax /@ (IntervalIntersection[acqBoutIntervals[[n]], #] & /@
              (Interval /@ quiescentPeriodsLongBef)), {n, 1, Length[acqBoutIntervals]}];

In[•]:= acceptedQuiescentPeriods = Flatten[
          Table[DeleteCases[Table[If[NumberQ@(Differences[testedIntervals[[m, n]]][[1]]),
              testedIntervals[[m, n]], Null], {n, 1, Length[testedIntervals[[m]]]}],
            Null], {m, 1, Length[testedIntervals]}], 1];

In[•]:= quiescentPeriodsLong = DeleteCases[
          Table[If[acceptedQuiescentPeriods[[n, 2]] - acceptedQuiescentPeriods[[n, 1]] ≥ 9,
            acceptedQuiescentPeriods[[n]], Null],
           {n, 1, Length[acceptedQuiescentPeriods]}], Null];

In[•]:= (***Compute cross-correlations between pupil diameter and dF/F0 during quiescent
          periods.  Compute in a time window of 8 seconds with 100 ms resolution***)

In[•]:= Table[Evaluate@ToExpression[StringJoin["dFFpupilCrossCorr", ToString[roi]]] =
           Mean[DeleteCases[Table[pupil = Table[pupil10HzInt[t],
                 {t, quiescentPeriodsLong[[i, 1]], quiescentPeriodsLong[[i, 2]], 0.1}];
              neuron = Table[(ToExpression[StringJoin["dFFlpfInt", ToString[roi]]])[t],
                {t, quiescentPeriodsLong[[i, 1]], quiescentPeriodsLong[[i, 2]], 0.1}];
              If[quiescentPeriodsLong[[i, 2]] - quiescentPeriodsLong[[i, 1]] > 8,
               Join[Reverse[Table[Correlation[pupil, RotateRight[neuron, -n]], {n, 0, 80, 1}]],
                Drop[Table[Correlation[pupil, RotateRight[neuron, n]], {n, 0, 80, 1}], 1]], Null],
              {i, 1, Length[quiescentPeriodsLong]}], Null]];, {roi, 1, numROIs}];

In[•]:= crossCorrTimeVals = Table[n, {n, -8, 8, 0.1}];

In[•]:= Table[Evaluate@ToExpression[StringJoin["dFFpupilCrossCorrTS", ToString[n]]] =
           Partition[Riffle[crossCorrTimeVals, (ToExpression[
               StringJoin["dFFpupilCrossCorr", ToString[n]]])], 2];, {n, 1, numROIs}];

          (***Visualize pupil-dFF cross correlograms***)

In[•]:= Manipulate[ListLinePlot[ToExpression[StringJoin["dFFpupilCrossCorrTS", ToString[roi]]]],
           {roi, 1, numROIs, 1}]

In[•]:= (***Export dFF-pupil cross-correlation data data***)

In[•]:= Table[Export[StringJoin["S:/Imaging/Garrett/FMB208_2PRig/", date, "/", mouse,
             "/Session", ToString[sessionNum], "/", "Pupil/", date, "_", mouse, "_",
             "Session", ToString[sessionNum], "_dFFpupilCrossCorr_ROI", ToString[n], ".txt"],
            ToExpression[StringJoin["dFFpupilCrossCorrTS", ToString[n]]]], {n, 1, numROIs}];

In[•]:= (*********************************)

          pupilModROIs = Range[numROIs];

In[•]:= Export[StringJoin["S:/Imaging/Garrett/FMB208_2PRig/", date, "/", mouse,
            "/Session", ToString[sessionNum], "/", "Pupil/", date, "_", mouse, "_",
            "Session", ToString[sessionNum], "_pupilModROIs", ".txt"], pupilModROIs];
```