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

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

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

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

In[ ]:= (**Import the raw pixel intensities from the whisker pad**)
whiskPixIntensities =
  Part[#, 2] & /@ (Drop[Import[StringJoin["S:/Imaging/Garrett/FMB208_2PRig/", date, "/",
       mouse, "/Session", ToString[sessionNum], "/", date, "_", mouse, "_", "Session",
       ToString[sessionNum], "_Whiskers/WhiskerPadPixelIntensity/", "Results.csv"]], 1]);

In[ ]:= (***Import the frame times from the camera***)

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

In[ ]:= (***Calculate camera frame rate***)
frameRate = Round[Length[frameTimes] / (Last[frameTimes] - First[frameTimes])]

(***Verify that length of frameTimes = length of whiskPixIntensities, +/- 2***)

In[ ]:= Length[whiskPixIntensities]

In[ ]:= Length[frameTimes]

In[ ]:= (***Match up the length of the cam sync pulses with the length of the image frames***)

In[ ]:= If[Length[frameTimes] > Length[whiskPixIntensities],
  frameTimes = Take[frameTimes, Length[whiskPixIntensities]];,
  whiskPixIntensities = Take[whiskPixIntensities, Length[frameTimes]];];

In[ ]:= (***Calculate the motion energy of the whisker pad***)

In[ ]:= whiskMotionEnergy = Abs /@ Differences[whiskPixIntensities];

In[ ]:= whiskMotionEnergyTS = Partition[Riffle[Drop[frameTimes, 1], whiskMotionEnergy], 2];

In[ ]:= whiskMotionInterp = Interpolation[whiskMotionEnergyTS];

In[ ]:= whisker10Hz = Table[whiskMotionInterp[t],
    {t, First[whiskMotionEnergyTS][[1]], Last[whiskMotionEnergyTS][[1]], 0.1}];

In[ ]:= whisker10HzTimeSeries = Partition[Riffle[Table[n, {n, First[whiskMotionEnergyTS][[1]],
      Last[whiskMotionEnergyTS][[1]], 0.1}], whisker10Hz], 2];

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[•]:= (***Lowpass filter dF/F 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[•]:= (***Import time periods with no locomotion***)

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[•]:= (***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[•]:= (*************************************************************************
         ************************************)
        (***In this part, calculate cross-correlograms between dF/F0 and whisker motion***)
        (*************************************************************************
         ************************************)

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

In[•]:= whisker10HzInt = Interpolation[whisker10HzTimeSeries];

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

In[•]:= Table[Evaluate@ToExpression[StringJoin["dFFwhiskCrossCorr", ToString[roi]]] =
          Mean[DeleteCases[Table[whisker = Table[whisker10HzInt[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[whisker, RotateRight[neuron, -n]], {n, 0, 80, 1}]],
                Drop[Table[Correlation[whisker, RotateRight[neuron, n]], {n, 0, 80, 1}], 1]],
               Null], {i, 1, Length[quiescentPeriodsLong]}], Null]];, {roi, 1, numROIs}];

In[•]:= (***For each ROI, make a time series of cross-correlation***)

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

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

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

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

In[•]:= (***Also calculate a summary cross correlation value for each ROI***)
```

```
In[ ]:= Table[Evaluate@ToExpression[StringJoin["summaryDffWhiskCorr", ToString[roi]]] = Mean[
          Table[Correlation[Table[ToExpression[StringJoin["dFFlpfInt", ToString[roi]]][t],
             {t, quiescentPeriods[[n, 1]], quiescentPeriods[[n, 2]], 0.1}], Table[
             whisker10HzInt[t], {t, quiescentPeriods[[n, 1]], quiescentPeriods[[n, 2]], 0.1}]],
           {n, 1, Length[quiescentPeriods]}]];, {roi, 1, numROIs}];

In[ ]:= (***Export data***)

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

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

    whiskModROIs = Range[numROIs];

In[ ]:= Export[StringJoin["S:/Imaging/Garrett/FMB208_2PRig/", date, "/", mouse,
        "/Session", ToString[sessionNum], "/", "WhiskerData/", date, "_", mouse, "_",
        "Session", ToString[sessionNum], "_WhiskModROIs", ".txt"], whiskModROIs];
```