```
In[•]:= (***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[•]:= (***For the first movie,
        import the neuropil-corrected calcium signal from each ROI chosen for the session***)

In[•]:= neuropilCorrFsMov1 = Import[
           StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/", mouse, "/Session",
             ToString[sessionNum], "/Ftraces/", "Movie1/F_neuropilsubtracted.txt"], "Table"];

In[•]:= numROIs = Dimensions[neuropilCorrFsMov1][[2]];

In[•]:= (***Do the same for all the other movies in the session***)

In[•]:= numAdditionalMovies =
          Length[FileNames["*", File[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/",
               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[•]:= Table[Evaluate@ToExpression[StringJoin["neuropilCorrFsMov", ToString[n]]] =
            Import[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/",
              date, "/", mouse, "/Session", ToString[sessionNum], "/Ftraces/",
              If[n ≥ 10, StringJoin["file_000", ToString[n], "_00001"],
               StringJoin["file_0000", ToString[n], "_00001"]], "_moco40_ref1-200",
              "/F_neuropilsubtracted.txt"], "Table"];, {n, 2, numAdditionalMovies + 1}];

In[•]:= (***Extract the calcium trace for each ROI***)

In[•]:= Table[Table[Evaluate@
              ToExpression[StringJoin["neuropilCorrFsMov", ToString[n], "ROI", ToString[m]]] =
             (ToExpression[StringJoin["neuropilCorrFsMov", ToString[n]]])[[All, m]];,
           {n, 1, numAdditionalMovies + 1}], {m, 1, numROIs}];

In[•]:= (***Concatenate the calcium traces for each ROI***)

In[•]:= Table[Evaluate@ToExpression[StringJoin["neuropilCorrFsFullSessionROI", ToString[m]]] =
            Catenate[Table[(ToExpression[StringJoin["neuropilCorrFsMov", ToString[n], "ROI",
                  ToString[m]]]), {n, 1, numAdditionalMovies + 1}]];, {m, 1, numROIs}];

In[•]:= (***Import the frame times for the 2P images and calculate the frame rate***)

In[•]:= tpFrameTimes =
          Drop[Drop[(Import[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date,
               "/", mouse, "/Session", ToString[sessionNum], "/", date, "_", mouse, "_",
               "Session", ToString[sessionNum], "_2PFrameTimes.txt"], "List"]), 16], -1];

        (***Verify that length of tpFrameTimes = length of tiff series, exactly***)

In[•]:= Length[tpFrameTimes]

In[•]:= Length[neuropilCorrFsFullSessionROI1]
```

```
In[∘]:= (***Number of frames per discrete acquisition***)

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

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

In[∘]:= (***Partition the neuropil-
         corrected fluorescence trace of each ROI into acquisition bouts***)

In[∘]:= Table[Evaluate@
           ToExpression[StringJoin["neuropilCorrFsFullSessionAcqBoutsROI", ToString[n]]] =
          Partition[(ToExpression[StringJoin["neuropilCorrFsFullSessionROI", ToString[n]]]),
           numFramesPerAcq];, {n, 1, numROIs}];

In[∘]:= (***For each ROI, compute F0 as a 1-minute moving 10th percentile
          of the fluorescence distribution.  Compute F0 per acquisition bout***)

In[∘]:= window = Round[tpFrameRate * 60];

In[∘]:= Table[Evaluate@ToExpression[StringJoin["f0ROI", ToString[n]]] =
          Flatten[Table[MovingMap[Quantile[#, 1 / 10] &,
              (ToExpression[StringJoin["neuropilCorrFsFullSessionAcqBoutsROI", ToString[n]]])[[
               i]], window, "Reflected"], {i, 1, Length[(ToExpression[StringJoin[
                  "neuropilCorrFsFullSessionAcqBoutsROI", ToString[n]]])]}]];, {n, 1, numROIs}];

In[∘]:= (***For each ROI, compute dF/F0***)

In[∘]:= Table[Evaluate@ToExpression[StringJoin["dFoverF0ROI", ToString[n]]] =
          ((ToExpression[StringJoin["neuropilCorrFsFullSessionROI", ToString[n]]]) -
              (ToExpression[StringJoin["f0ROI", ToString[n]]])) /
           (ToExpression[StringJoin["f0ROI", ToString[n]]]);, {n, 1, numROIs}];

In[∘]:= (***For each ROI, generate dF/F0 time series***)

In[∘]:= Table[Evaluate@ToExpression[StringJoin["dFoverF0ROIts", ToString[n]]] =
          Partition[Riffle[tpFrameTimes, ToExpression[StringJoin["dFoverF0ROI", ToString[n]]]],
           2];, {n, 1, numROIs}];

In[∘]:= (***Export dF/F0 time series for each ROI***)

In[∘]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/", mouse,
          "/Session", ToString[sessionNum], "/dFoverF0TimeSeries/", date, "_", mouse, "_",
          "Session", ToString[sessionNum], "_dFoverF0ts_", "ROI", ToString[n], ".txt"],
         ToExpression[StringJoin["dFoverF0ROIts", ToString[n]]]], {n, 1, numROIs}];

In[∘]:= Manipulate[ListLinePlot[ToExpression[StringJoin["dFoverF0ROIts", ToString[n]]],
          PlotRange → All], {n, 1, numROIs, 1}]
```