```
In[◦]:= (***Need a separate script for cell bodies because cell body ROIs aren'
         t initially chosen based on their activity levels like axon ROIs are***)

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

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

In[◦]:= rois = Import[StringJoin["S:/Imaging/Garrett/FMB208_2PRig/", date, "/", mouse,
            "/Session", ToString[sessionNum], "/", "LocomotionData/", date, "_", mouse,
            "_", "Session", ToString[sessionNum], "_locModROIs", ".txt"], "List"];

In[◦]:= Table[Evaluate@ToExpression[StringJoin["dFFts", 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, rois}];

In[◦]:= (***Import the walk bout start and end times***)

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

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[◦]:= (***)

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

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

In[◦]:= acqBoutIntervals =
          Interval /@ ({First[#], Last[#]} & /@ (Partition[tpFrameTimes, numFramesPerAcq]));

In[◦]:= (****Test which locomotion onset times are within at least 15 s after the start
          of an acquisition bout and 6 s before the end of an acquisition bout****)
```

```
In[•]:= acceptedLocOnsetTest = Table[AnyTrue[acqBoutIntervals,
             IntervalMemberQ[#, Interval[{walkBouts[[n, 1]] - 15, walkBouts[[n, 1]] + 6}]] &],
           {n, 1, Length[walkBouts]}];

In[•]:= acceptedLocOnsetPositions = Position[acceptedLocOnsetTest, True];

In[•]:= (****Test which locomotion offset times are within at least 6 s after the start
          of an acquisition bout and 15 s before the end of an acquisition bout****)

In[•]:= acceptedLocOffsetTest = Table[AnyTrue[acqBoutIntervals,
             IntervalMemberQ[#, Interval[{walkBouts[[n, 2]] - 6, walkBouts[[n, 2]] + 15}]] &],
           {n, 1, Length[walkBouts]}];

In[•]:= acceptedLocOffsetPositions = Position[acceptedLocOffsetTest, True];

In[•]:= (***Only consider walk bouts that fulfill the above criteria***)

In[•]:= walkBoutsForOnset = Table[walkBouts[[(Flatten[acceptedLocOnsetPositions][[n]])]],
           {n, 1, Length[acceptedLocOnsetPositions]}];

In[•]:= walkBoutsForOffset = Table[walkBouts[[(Flatten[acceptedLocOffsetPositions][[n]])]],
           {n, 1, Length[acceptedLocOffsetPositions]}];

In[•]:= locOnsetTimes = walkBoutsForOnset[[All, 1]];

In[•]:= locOffsetTimes = walkBoutsForOffset[[All, 2]];

In[•]:= (***Get interpolation functions for the dF/F0 time series for each ROI***)

In[•]:= Table[Evaluate@ToExpression[StringJoin["dFFInterpFunc", ToString[n]]] =
             Interpolation[ToExpression[StringJoin["dFFts", ToString[n]]]];, {n, rois}];

In[•]:= (*********************************Z-
         SCORING FROM LOC ONSET************************************)

In[•]:= (***For each ROI, calculate the z-scored dF/F0 for around each onset,
        using the 10 to 15 seconds before onset as the baseline***)

In[•]:= Table[Evaluate@ToExpression[StringJoin["baselinePeriOnsetMeans", ToString[n]]] =
             Table[Mean[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]])[k],
                 {k, locOnsetTimes[[x]] - 15, locOnsetTimes[[x]] - 10, 1/tpFrameRate}]],
               {x, 1, Length[locOnsetTimes]}];, {n, rois}];

In[•]:= Table[Evaluate@ToExpression[StringJoin["baselinePeriOnsetSDs", ToString[n]]] = Table[
              StandardDeviation[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]])[
                 k], {k, locOnsetTimes[[x]] - 15, locOnsetTimes[[x]] - 10, 1/tpFrameRate}]],
               {x, 1, Length[locOnsetTimes]}];, {n, rois}];

In[•]:= Table[Evaluate@ToExpression[StringJoin["periOnsetZscoredDFFs", ToString[n]]] =
             Table[Table[(((ToExpression[StringJoin["dFFInterpFunc", ToString[n]]])[i]) -
                   (ToExpression[StringJoin["baselinePeriOnsetMeans", ToString[n]]])[[x]]) /
                 ((ToExpression[StringJoin["baselinePeriOnsetSDs", ToString[n]]])[[x]]),
                {i, locOnsetTimes[[x]] - 15, locOnsetTimes[[x]] + 6, 1/tpFrameRate}],
               {x, 1, Length[locOnsetTimes]}];, {n, rois}];

In[•]:= (***For each ROI, make a time series of z-scored dF/F0 and mean z-scored dF/F0,***)
```

```
In[∘]:= periOnsetTimeVals = Table[n, {n, -15, 6, 1/tpFrameRate}];

In[∘]:= Table[Evaluate@ToExpression[StringJoin["meanZscoredPeriOnsetDFFsTS", ToString[n]]] =
          Mean[Table[Partition[Riffle[periOnsetTimeVals,
                (ToExpression[StringJoin["periOnsetZscoredDFFs", ToString[n]]])[[m]]], 2],
             {m, 1, Length[(ToExpression[StringJoin["periOnsetZscoredDFFs",
                  ToString[n]]])]}]];, {n, rois}];

In[∘]:= (*******************************Z-
         SCORING FROM LOC OFFSET**********************************)

In[∘]:= (***For each ROI, calculate the z-scored dF/F0 for around each offset,
        using the 10 to 15 seconds before offset as the baseline***)

In[∘]:= Table[Evaluate@ToExpression[StringJoin["baselinePeriOffsetMeans", ToString[n]]] =
          Table[Mean[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]])[k],
              {k, locOffsetTimes[[x]] + 10, locOffsetTimes[[x]] + 15, 1/tpFrameRate}]],
           {x, 1, Length[locOffsetTimes]}];, {n, rois}];

In[∘]:= Table[Evaluate@ToExpression[StringJoin["baselinePeriOffsetSDs", ToString[n]]] = Table[
           StandardDeviation[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]])[
              k], {k, locOffsetTimes[[x]] + 10, locOffsetTimes[[x]] + 15, 1/tpFrameRate}]],
           {x, 1, Length[locOffsetTimes]}];, {n, rois}];

In[∘]:= Table[Evaluate@ToExpression[StringJoin["periOffsetZscoredDFFs", ToString[n]]] =
          Table[Table[(((ToExpression[StringJoin["dFFInterpFunc", ToString[n]]])[i]) -
                (ToExpression[StringJoin["baselinePeriOffsetMeans", ToString[n]]])[[x]]) /
              ((ToExpression[StringJoin["baselinePeriOffsetSDs", ToString[n]]])[[x]]),
             {i, locOffsetTimes[[x]] - 6, locOffsetTimes[[x]] + 15, 1/tpFrameRate}],
           {x, 1, Length[locOffsetTimes]}];, {n, rois}];

In[∘]:= (***For each ROI, make a time series of z-scored dF/F0 and mean z-scored dF/F0,***)

In[∘]:= periOffsetTimeVals = Table[n, {n, -6, 15, 1/tpFrameRate}];

In[∘]:= Table[Evaluate@ToExpression[StringJoin["meanZscoredPeriOffsetDFFsTS", ToString[n]]] =
          Mean[Table[Partition[Riffle[periOffsetTimeVals,
                (ToExpression[StringJoin["periOffsetZscoredDFFs", ToString[n]]])[[m]]], 2],
             {m, 1, Length[(ToExpression[StringJoin["periOffsetZscoredDFFs",
                  ToString[n]]])]}]];, {n, rois}];

        (***Visualize the z-scored dF/F of each ROI aligned to locomotion onset and offset***)

In[∘]:= Manipulate[{ListLinePlot[
           ToExpression[StringJoin["meanZscoredPeriOnsetDFFsTS", ToString[n]]], PlotRange → All],
          ListLinePlot[ToExpression[StringJoin["meanZscoredPeriOffsetDFFsTS", ToString[n]]],
           PlotRange → All]}, {n, rois}]

In[∘]:= (***Export results***)

In[∘]:= Table[Export[StringJoin["S:/Imaging/Garrett/FMB208_2PRig/", date, "/", mouse, "/Session",
          ToString[sessionNum], "/", "LocomotionData/", date, "_", mouse, "_", "Session",
          ToString[sessionNum], "_PeriOnsetZDFF_PreAndPostBaseline_ROI", ToString[n], ".txt"],
         ToExpression[StringJoin["meanZscoredPeriOnsetDFFsTS", ToString[n]]]], {n, rois}];
```

```
In[•]:= Table[Export[StringJoin["S:/Imaging/Garrett/FMB208_2PRig/", date, "/", mouse, "/Session",
            ToString[sessionNum], "/", "LocomotionData/", date, "_", mouse, "_", "Session",
            ToString[sessionNum], "_PeriOffsetZDFF_PreAndPostBaseline_ROI", ToString[n], ".txt"],
          ToExpression[StringJoin["meanZscoredPeriOffsetDFFsTS", ToString[n]]]], {n, rois}];
```