```
(***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[•]:= dilConsOnsetPairsLargeQuiNoWhisk =
        ToExpression /@ Import[StringJoin["S:/Imaging/Garrett/FMB208_2PRig/", date, "/", mouse,
            "/Session", ToString[sessionNum], "/Pupil/", date, "_", mouse, "_", "Session",
            ToString[sessionNum], "_quisecentDilationConstrictionTimePairs.txt"], "List"];

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

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

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

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[•]:= tpFrameRate = Round[Length[tpFrameTimes] / (Last[tpFrameTimes] - First[tpFrameTimes])];

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["dFFtimeseriesInt", ToString[n]]] = Interpolation[
            ToExpression[StringJoin["dFFtimeseries", ToString[n]]]];, {n, 1, numROIs}];

In[•]:= Table[Evaluate@ToExpression[StringJoin["dFFAllBaselines", ToString[roi]]] =
            Table[Table[(ToExpression[StringJoin["dFFtimeseriesInt", ToString[roi]]])[t],
              {t, lowPupilPeriods[[n, 1]], lowPupilPeriods[[n, 2]], 1 / tpFrameRate}],
            {n, 1, Length[lowPupilPeriods]}];, {roi, 1, numROIs}];

In[•]:= Table[Evaluate@ToExpression[StringJoin["quietMean", ToString[roi]]] =
            Mean[Flatten[(ToExpression[StringJoin["dFFAllBaselines", ToString[roi]]])]];,
          {roi, 1, numROIs}];
```

```mathematica
In[ ]:= Table[Evaluate@ToExpression[StringJoin["quietSD", ToString[roi]]] =
         StandardDeviation[Flatten[(ToExpression[
             StringJoin["dFFAllBaselines", ToString[roi]]])]];, {roi, 1, numROIs}];
```

```mathematica
In[ ]:= Table[Evaluate@ToExpression[StringJoin["dFFts", ToString[n]]] =
         Partition[Riffle[(ToExpression[StringJoin["dFFtimeseries", ToString[n]]])[[All, 1]],
           ((ToExpression[StringJoin["dFFtimeseries", ToString[n]]])[[All, 2]] -
              ToExpression[StringJoin["quietMean", ToString[n]]]) /
            ToExpression[StringJoin["quietSD", ToString[n]]]], 2];, {n, 1, numROIs}];
```

```mathematica
In[ ]:= (***Bandpass filter the dF/F0 traces from 0.1 to 1 Hz***)
```

```mathematica
In[ ]:= fftFilter[data_, sR_, band_] := Module[{dur, fRes, f1, f2, rSpan, f}, dur = Length@data / sR;
        (*Total duration of data.*)fRes = 1 / dur;
        (*Resolution of FFT is 1/duration.*)
        (*frequencies in terms of Fourier part.1=DC*){f1, f2} = 1 + Quotient[#, fRes] & /@ band;
        rSpan = If[f1 == 1, Span @@ {2, -1}, Span @@ {1, -1}];
        f = Fourier[data, List /@ Range[f1, f2]];
        f = PadLeft[f, f2];
        f = Join[PadRight[f, (Length@data) - f2 + 1], Conjugate@Reverse@(f[[rSpan]])];
        f = PadRight[f, Length@data];
        Re@InverseFourier[f]]
```

```mathematica
In[ ]:= Table[Evaluate@ToExpression[StringJoin["dFFbpf", ToString[n]]] =
         fftFilter[Part[#, 2] & /@ ToExpression[StringJoin["dFFts", ToString[n]]],
          tpFrameRate, {0.1, 1}];, {n, 1, numROIs}];
```

```mathematica
In[ ]:= Table[Evaluate@ToExpression[StringJoin["dFFbpfTS", ToString[n]]] =
         Partition[Riffle[Part[#, 1] & /@ ToExpression[StringJoin["dFFts", ToString[n]]],
           ToExpression[StringJoin["dFFbpf", ToString[n]]]], 2];, {n, 1, numROIs}];
```

```mathematica
In[ ]:= Table[Evaluate@ToExpression[StringJoin["dFFbpfInt", ToString[n]]] =
         Interpolation[ToExpression[StringJoin["dFFbpfTS", ToString[n]]]];, {n, 1, numROIs}];
```

```mathematica
In[ ]:= (***For each dilation-constriction interval for each ROI,
    extract the bandpass-filtered dF/F0 trace in that interval.  Assign phase values -
     Pi to 0 for these intervals***)
```

```mathematica
In[ ]:= Table[Evaluate@ToExpression[StringJoin["dFFvsDilConsPhase", ToString[n]]] =
         Flatten[Table[Partition[Riffle[
            Table[n, {n, -Pi, 0, N@Pi / (Length[Table[(ToExpression[StringJoin["dFFbpfInt",
                    ToString[n]]])[t], {t, dilConsOnsetPairsLargeQuiNoWhisk[[i, 1]],
                  dilConsOnsetPairsLargeQuiNoWhisk[[i, 2]], N@1 / tpFrameRate}]] - 1)}],
            Table[(ToExpression[StringJoin["dFFbpfInt", ToString[n]]])[t],
             {t, dilConsOnsetPairsLargeQuiNoWhisk[[i, 1]],
              dilConsOnsetPairsLargeQuiNoWhisk[[i, 2]], N@1 / tpFrameRate}]], 2],
           {i, 1, Length[dilConsOnsetPairsLargeQuiNoWhisk]}], 1];, {n, 1, numROIs}];
```

```mathematica
In[ ]:= (***For each ROI, bin dF/F0 values into 32 bins from -Pi to 0***)
```

```mathematica
In[ ]:= dilPhaseBins = Partition[Table[n, {n, -Pi, 0, Pi / 32}], 2, 1];
```

```mathematica
In[ ]:= dilPhaseBinInts = Interval /@ dilPhaseBins;
```

```
In[ ]:= Table[Evaluate@ToExpression[StringJoin["meanDFFvsDilConsPhase", ToString[roi]]] =
        Mean /@ GatherBy[
          DeleteCases[Table[If[Length[Flatten[Position[IntervalMemberQ[#, (ToExpression[
                    StringJoin["dFFvsDilConsPhase", ToString[roi]]])[[n, 1]]] & /@
                dilPhaseBinInts, True]]] == 0, Null, {Take[Flatten[Position[
                IntervalMemberQ[#, (ToExpression[StringJoin["dFFvsDilConsPhase",
                      ToString[roi]]])[[n, 1]]] & /@ dilPhaseBinInts, True]], 1][[1]],
              (ToExpression[StringJoin["dFFvsDilConsPhase", ToString[roi]]])[[n, 2]]}],
            {n, 1, Length[(ToExpression[StringJoin["dFFvsDilConsPhase", ToString[roi]]])]}],
          Null], First];, {roi, 1, numROIs}];
```

```
In[ ]:= (***For each constriction-dilation interval for each ROI,
    extract the bandpass-filtered dF/F0 trace in that
      interval.  Assign phase values 0 to Pi for these intervals***)
```

```
In[ ]:= Table[Evaluate@ToExpression[StringJoin["dFFvsConsDilPhase", ToString[n]]] =
        Flatten[Table[Partition[Riffle[
            Table[n, {n, 0, Pi, N@Pi / (Length[Table[(ToExpression[StringJoin["dFFbpfInt",
                    ToString[n]]])[t], {t, consDilOnsetPairsLargeQuiNoWhisk[[i, 1]],
                  consDilOnsetPairsLargeQuiNoWhisk[[i, 2]], N@1 / tpFrameRate}]] - 1)}],
            Table[(ToExpression[StringJoin["dFFbpfInt", ToString[n]]])[t],
             {t, consDilOnsetPairsLargeQuiNoWhisk[[i, 1]],
              consDilOnsetPairsLargeQuiNoWhisk[[i, 2]], N@1 / tpFrameRate}]], 2],
          {i, 1, Length[consDilOnsetPairsLargeQuiNoWhisk]}], 1];, {n, 1, numROIs}];
```

```
In[ ]:= (***For each ROI, bin dF/F0 values into 32 bins from 0 to Pi***)
```

```
In[ ]:= consPhaseBins = Partition[Table[n, {n, 0, Pi, Pi / 32}], 2, 1];
```

```
In[ ]:= consPhaseBinInts = Interval /@ consPhaseBins;
```

```
In[ ]:= Table[Evaluate@ToExpression[StringJoin["meanDFFvsConsDilPhase", ToString[roi]]] =
        Mean /@ GatherBy[
          DeleteCases[Table[If[Length[Flatten[Position[IntervalMemberQ[#, (ToExpression[
                    StringJoin["dFFvsConsDilPhase", ToString[roi]]])[[n, 1]]] & /@
                consPhaseBinInts, True]]] == 0, Null, {Take[Flatten[Position[
                IntervalMemberQ[#, (ToExpression[StringJoin["dFFvsConsDilPhase",
                      ToString[roi]]])[[n, 1]]] & /@ consPhaseBinInts, True]], 1][[1]],
              (ToExpression[StringJoin["dFFvsConsDilPhase", ToString[roi]]])[[n, 2]]}],
            {n, 1, Length[(ToExpression[StringJoin["dFFvsConsDilPhase", ToString[roi]]])]}],
          Null], First];, {roi, 1, numROIs}];
```

```
In[ ]:= (***Finally, assign the phase values the
     bins to generate the final dF/F0 vs pupil phase data***)
```

```
In[ ]:= Table[Evaluate@ToExpression[StringJoin["dFFvsPupilPhase", ToString[roi]]] =
        Partition[Riffle[Join[First /@ dilPhaseBins, First /@ consPhaseBins], Join[
            Part[#, 2] & /@ ToExpression[StringJoin["meanDFFvsDilConsPhase", ToString[roi]]],
            Part[#, 2] & /@ ToExpression[StringJoin["meanDFFvsConsDilPhase", ToString[roi]]]]],
          2];, {roi, 1, numROIs}];
```

```
(***Visualize z-scored dFF aligned to pupil dilation-
 constriction phase for each ROI***)
```

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

In[ ]:= `(***Export dFF-pupil phase data***)`

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