

```

ln[ ]:= (**Input identifying information**)

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

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

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

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

ln[ ]:= 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];

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

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

ln[ ]:= numROIs =
  Length[FileNames["*", File[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date,
    "/", mouse, "/Session", ToString[sessionNum], "/dFoverF0TimeSeries/"]]]];

ln[ ]:= Table[Evaluate@ToExpression[StringJoin["dFFts", ToString[n]]] =
  ToExpression /@ Import[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/",
    date, "/", mouse, "/Session", ToString[sessionNum], "/dFoverF0TimeSeries/",
    date, "_", mouse, "_Session", ToString[sessionNum], "_",
    "dFoverF0ts_ROI", ToString[n], ".txt"], "List"]; {n, 1, numROIs}];

ln[ ]:= (**Import the Spike2 file indicating onset times of the noise movies**)

ln[ ]:= vStimOnsets =
  Drop[Drop[(Import[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date,
    "/", mouse, "/Session", ToString[sessionNum], "/", date, "_", mouse, "_",
    "Session", ToString[sessionNum], "_VisStimOnsetTimes.txt"], "List"], 16], -1];

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

ln[ ]:= Table[Evaluate@ToExpression[StringJoin["dFFInterpFunc", ToString[n]]] =
  Interpolation[ToExpression[StringJoin["dFFts", ToString[n]]]]; {n, 1, numROIs}];

ln[ ]:= (**For each ROI,
calculate the raw evoked dF/F0 for around each stimulus presentation onset,
which includes 1 s prior stimulus onset and 3 s post stimulus offset**)
Table[Evaluate@ToExpression[StringJoin["evokedRawDFFs", ToString[n]]] =
  Table[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]][i],
    {i, vStimOnsets[[x]] - 1, vStimOnsets[[x]] + 5, 1/tpFrameRate}],
    {x, 1, Length[vStimOnsets]}]; {n, 1, numROIs}];

```

```

In[ ]:= (**For each ROI, calculate the baseline (0.5 s pre-stim)-
        subtracted evoked dF/F0 for each stimulus presentation**)
Table[Evaluate@ToExpression[StringJoin["blSubtractEvokedDFFs", ToString[n]]] =
  Table[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]][i] -
    (Mean[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]][j],
      {j, vStimOnsets[[x]] - 1, vStimOnsets[[x]]}]]),
    {i, vStimOnsets[[x]] - 1, vStimOnsets[[x]] + 5, 1/tpFrameRate}],
    {x, 1, Length[vStimOnsets]}];, {n, 1, numROIs}];

In[ ]:= (**For each ROI,
        calculate the z-scored dF/F0 for around each stimulus presentation onset,
        which includes 1 s prior stimulus onset and 3 s post stimulus offset**)

In[ ]:= Table[Evaluate@ToExpression[StringJoin["baselineMeans", ToString[n]]] =
  Table[Mean[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]][k],
    {k, vStimOnsets[[x]] - 1, vStimOnsets[[x]], 1/tpFrameRate}]]],
    {x, 1, Length[vStimOnsets]}];, {n, 1, numROIs}];

In[ ]:= Table[Evaluate@ToExpression[StringJoin["baselineSDs", ToString[n]]] = Table[
  StandardDeviation[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]][
    k], {k, vStimOnsets[[x]] - 1, vStimOnsets[[x]], 1/tpFrameRate}]]],
    {x, 1, Length[vStimOnsets]}];, {n, 1, numROIs}];

In[ ]:= Table[Evaluate@ToExpression[StringJoin["evokedZscoredDFFs", ToString[n]]] =
  Table[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]][i] -
    (ToExpression[StringJoin["baselineMeans", ToString[n]]][[x]]) /
    ((ToExpression[StringJoin["baselineSDs", ToString[n]]][[x]]),
    {i, vStimOnsets[[x]] - 1, vStimOnsets[[x]] + 5, 1/tpFrameRate}],
    {x, 1, Length[vStimOnsets]}];, {n, 1, numROIs}];

In[ ]:= evokedTimeVals = Table[n, {n, -1, 5, 1/tpFrameRate}];

In[ ]:= (**Import the csv file containing the stimulus
        properties for each stimulus presentation in the session**)

In[ ]:= stimProps = Import[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/",
  date, "/", mouse, "/Session", ToString[sessionNum], "/", date, "_",
  mouse, "_", "Session", ToString[sessionNum], "_StimProps.csv"]];

In[ ]:= {Length[stimProps], Length[vStimOnsets], Length[stimProps] == Length[vStimOnsets]}

In[ ]:= (**Extract grating orientation for each stimulus presented**)

In[ ]:= orients = Round /@ (Part[#, 2] & /@ stimProps);

orientList = Part[#, 1] & /@ Sort[Tally[orients]];

In[ ]:= (**Find the positions at which each of the 12 orientations was presented**)

In[ ]:= pos0deg = Flatten[Position[orients, 0]];

In[ ]:= pos32deg = Flatten[Position[orients, 32]];

In[ ]:= pos64deg = Flatten[Position[orients, 64]];

```

```

In[ ]:= pos96deg = Flatten[Position[orients, 96]];

In[ ]:= pos128deg = Flatten[Position[orients, 128]];

In[ ]:= pos160deg = Flatten[Position[orients, 160]];

In[ ]:= pos192deg = Flatten[Position[orients, 192]];

In[ ]:= pos224deg = Flatten[Position[orients, 224]];

In[ ]:= pos256deg = Flatten[Position[orients, 256]];

In[ ]:= pos288deg = Flatten[Position[orients, 288]];

In[ ]:= pos320deg = Flatten[Position[orients, 320]];

In[ ]:= pos352deg = Flatten[Position[orients, 352]];

In[ ]:= (**For each ROI, make a time series of raw evoked dF/F0,
baseline-subtracted evoked dF/F0, and mean baseline-subtracted evoked dF/F0**)

In[ ]:= Table[Evaluate@ToExpression[StringJoin["evokedRawDFFsTS", ToString[n]]] =
  Table[Partition[Riffle[evokedTimeVals,
    (ToExpression[StringJoin["evokedRawDFFs", ToString[n]]][[m]]], 2], {m, 1, Length[
    (ToExpression[StringJoin["evokedRawDFFs", ToString[n]]][[m]]]}];, {n, 1, numROIs}];

In[ ]:= Table[Evaluate@ToExpression[StringJoin["blSubtractEvokedDFFsTS", ToString[n]]] =
  Table[Partition[Riffle[evokedTimeVals,
    (ToExpression[StringJoin["blSubtractEvokedDFFs", ToString[n]]][[m]]], 2],
    {m, 1, Length[(ToExpression[StringJoin["blSubtractEvokedDFFs", ToString[n]]][[m]])]}];,
    {n, 1, numROIs}];

In[ ]:= Table[Evaluate@ToExpression[StringJoin["meanBLSubtractEvokedDFFsTS", ToString[n]]] =
  Mean[Table[Partition[Riffle[evokedTimeVals,
    (ToExpression[StringJoin["blSubtractEvokedDFFs", ToString[n]]][[m]]], 2],
    {m, 1, Length[(ToExpression[StringJoin["blSubtractEvokedDFFs",
    ToString[n]]][[m]])]}];, {n, 1, numROIs}];

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

In[ ]:= Table[Evaluate@ToExpression[StringJoin["zScoredDFFsTS", ToString[n]]] =
  Table[Partition[Riffle[evokedTimeVals,
    (ToExpression[StringJoin["evokedZscoredDFFs", ToString[n]]][[m]]], 2],
    {m, 1, Length[(ToExpression[StringJoin["evokedZscoredDFFs", ToString[n]]][[m]])]}];, {n,
    1, numROIs}];

In[ ]:= Table[Evaluate@ToExpression[StringJoin["meanZscoredDFFsTS", ToString[n]]] =
  Mean[Table[Partition[Riffle[evokedTimeVals,
    (ToExpression[StringJoin["evokedZscoredDFFs", ToString[n]]][[m]]], 2],
    {m, 1, Length[(ToExpression[StringJoin["evokedZscoredDFFs",
    ToString[n]]][[m]])]}];, {n, 1, numROIs}];

```

```

In[ ]:= (**To quantify visual responsiveness of each ROI,
perform a T test comparing the mean dF/F 1 s before stimulus onset and the mean
dF/F during the 2 s stimulus presentation, across all stimulus presentations**)

In[ ]:= Table[Evaluate@ToExpression[StringJoin["visRespPValROI", ToString[n]]] =
  Quiet[TTest[{Mean /@ (Take[#, tpFrameRate] & /@
    (ToExpression[StringJoin["evokedRawDFFs", ToString[n]]])),
    Mean /@ (Take[#, {tpFrameRate + 1, (tpFrameRate + 1) + (2 * tpFrameRate)}]) & /@
    (ToExpression[StringJoin["evokedRawDFFs", ToString[n]]])),
    AlternativeHypothesis -> "Less"]];, {n, 1, numROIs}];

In[ ]:= pVallist = Table[ToExpression[StringJoin["visRespPValROI", ToString[n]]], {n, 1, numROIs}];

In[ ]:= (**Extract the ROIs with significant responses**)

In[ ]:= sigRespROIs = DeleteCases[Table[If[pVallist[[n]] < 0.01, n, Null], {n, 1, numROIs}], Null];

In[ ]:= (**For each ROI,
visualize the p-value and overall evoked responses relative to baseline**)

In[ ]:= Table[{StringJoin["ROI", ToString[n]],
  ToExpression[StringJoin["visRespPValROI", ToString[n]]],
  ListLinePlot[ToExpression[StringJoin["meanBLSubtractEvokedDFFsTS", ToString[n]]]}],
  {n, 1, numROIs}] // TableForm

In[ ]:= (**For each ROI, calculate a new dF/F value that indicates
its overall visual responsiveness across all stimulus presentations,
where F0 is the mean dF/F 1 s before stimulus onset and F is the mean dF/F during the 2-
s stimulus presentation**)

In[ ]:= Table[
  Evaluate@ToExpression[StringJoin["overallVisDFF", ToString[n]]] = (Mean[Part[#, 2] & /@
    Take[Mean[(ToExpression[StringJoin["evokedRawDFFsTS", ToString[n]]]),
      {tpFrameRate + 1, (tpFrameRate + 1) + (2 * tpFrameRate)}]) - Mean[Part[#, 2] & /@
    Take[Mean[(ToExpression[StringJoin["evokedRawDFFsTS", ToString[n]]]),
      tpFrameRate]]] / Mean[Part[#, 2] & /@
    Take[Mean[(ToExpression[StringJoin["evokedRawDFFsTS", ToString[n]]]),
      tpFrameRate]]];, {n, 1, numROIs}];

In[ ]:= (**For each ROI, calculate an additional measure of overall visual
responsiveness across all stimulus presentations, which, instead of a new dF/F,
is the mean z-scored dF/F during the 2 s of stimulus presentation**)

In[ ]:= Table[Evaluate@ToExpression[StringJoin["overallVisZScore", ToString[n]]] = Mean[
  Part[#, 2] & /@ Take[Mean[(ToExpression[StringJoin["zScoredDFFsTS", ToString[n]]]),
    {tpFrameRate + 1, (tpFrameRate + 1) + (2 * tpFrameRate)}]]];, {n, 1, numROIs}];

In[ ]:= (**For each ROI, collect the peri-stimulus dF/F traces for each orientation**)

In[ ]:= Table[Table[Evaluate@
  ToExpression[StringJoin["periStimDFF", ToString[m], "DegROI", ToString[n]]] =
  (ToExpression[StringJoin["evokedRawDFFsTS", ToString[n]]][ToExpression[
    StringJoin["pos", ToString[m], "deg"]]]];, {n, 1, numROIs}];, {m, orientList}];

```

```

In[ ]:= Table[Table[Evaluate@
    ToExpression[StringJoin["periStimDFFb1Sub", ToString[m], "DegROI", ToString[n]]] =
    (ToExpression[StringJoin["b1SubtractEvokedDFFsTS", ToString[n]]])[[ToExpression[
        StringJoin["pos", ToString[m], "deg"]]]];, {n, 1, numROIs}];, {m, orientList}];

In[ ]:= (**For each ROI,
    collect the peri-stimulus z-scored dF/F traces for each orientation**)

In[ ]:= Table[Table[Evaluate@ToExpression[
    StringJoin["periStimZscoredDFF", ToString[m], "DegROI", ToString[n]]] =
    (ToExpression[StringJoin["zScoredDFFsTS", ToString[n]]])[[ToExpression[
        StringJoin["pos", ToString[m], "deg"]]]];, {n, 1, numROIs}];, {m, orientList}];

In[ ]:= (**For visualization purposes (orientation):**)

In[ ]:= Table[Evaluate@ToExpression[StringJoin["maxRespValOrient", ToString[n]]] =
    Max[Flatten[{Part[#, 2] & /@
        Mean[(ToExpression[StringJoin["periStimDFFb1Sub0DegROI", ToString[n]]])],
        Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
            "periStimDFFb1Sub32DegROI", ToString[n]]])], Part[#, 2] & /@
        Mean[(ToExpression[StringJoin["periStimDFFb1Sub64DegROI", ToString[n]]])],
        Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
            "periStimDFFb1Sub96DegROI", ToString[n]]])], Part[#, 2] & /@
        Mean[(ToExpression[StringJoin["periStimDFFb1Sub128DegROI", ToString[n]]])],
        Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
            "periStimDFFb1Sub160DegROI", ToString[n]]])], Part[#, 2] & /@
        Mean[(ToExpression[StringJoin["periStimDFFb1Sub192DegROI", ToString[n]]])],
        Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
            "periStimDFFb1Sub224DegROI", ToString[n]]])], Part[#, 2] & /@
        Mean[(ToExpression[StringJoin["periStimDFFb1Sub256DegROI", ToString[n]]])],
        Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
            "periStimDFFb1Sub288DegROI", ToString[n]]])], Part[#, 2] & /@
        Mean[(ToExpression[StringJoin["periStimDFFb1Sub320DegROI", ToString[n]]])],
        Part[#, 2] & /@ Mean[(ToExpression[StringJoin["periStimDFFb1Sub352DegROI",
            ToString[n]]])]]];, {n, 1, numROIs}];

```

```

In[ ]:= Table[Evaluate@ToExpression[StringJoin["minRespValOrient", ToString[n]]] =
  Min[Flatten[{Part[#, 2] & /@
    Mean[(ToExpression[StringJoin["periStimDFFblSub0DegROI", ToString[n]])],
    Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
      "periStimDFFblSub32DegROI", ToString[n]])], Part[#, 2] & /@
    Mean[(ToExpression[StringJoin["periStimDFFblSub64DegROI", ToString[n]])],
    Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
      "periStimDFFblSub96DegROI", ToString[n]])], Part[#, 2] & /@
    Mean[(ToExpression[StringJoin["periStimDFFblSub128DegROI", ToString[n]])],
    Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
      "periStimDFFblSub160DegROI", ToString[n]])], Part[#, 2] & /@
    Mean[(ToExpression[StringJoin["periStimDFFblSub192DegROI", ToString[n]])],
    Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
      "periStimDFFblSub224DegROI", ToString[n]])], Part[#, 2] & /@
    Mean[(ToExpression[StringJoin["periStimDFFblSub256DegROI", ToString[n]])],
    Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
      "periStimDFFblSub288DegROI", ToString[n]])], Part[#, 2] & /@
    Mean[(ToExpression[StringJoin["periStimDFFblSub320DegROI", ToString[n]])],
    Part[#, 2] & /@ Mean[(ToExpression[StringJoin["periStimDFFblSub352DegROI",
      ToString[n]])]]]]]; {n, 1, numROIs}];

In[ ]:= (**For each ROI, calculate the mean response to the presentation of each orientation,
  which is a new trial-averaged dF/F, where F0 for each trial is the mean dF/F 1 s before
  stimulus onset and F is the mean dF/F during the 2 s stimulus presentation**)

In[ ]:= Table[Evaluate@ToExpression[StringJoin["respAt", ToString[m], "DegROI", ToString[n]]] =
  ((Mean[Part[#, 2] & /@ Mean[Table[N@Take[(ToExpression[StringJoin["periStimDFF",
    ToString[m], "DegROI", ToString[n]])][[k]], {tpFrameRate + 1,
    (tpFrameRate + 1) + (2 * tpFrameRate)}], {k, 1, Length[(ToExpression[
    StringJoin["periStimDFF", ToString[m], "DegROI", ToString[n]])]}]]]) -
  (Mean[Part[#, 2] & /@ Mean[Table[N@Take[(ToExpression[StringJoin["periStimDFF",
    ToString[m], "DegROI", ToString[n]])][[k]], tpFrameRate],
    {k, 1, Length[(ToExpression[StringJoin["periStimDFF", ToString[m],
    "DegROI", ToString[n]])]}]]])]) /
  (Mean[Part[#, 2] & /@ Mean[Table[N@Take[(ToExpression[StringJoin["periStimDFF",
    ToString[m], "DegROI", ToString[n]])][[k]], tpFrameRate],
    {k, 1, Length[(ToExpression[StringJoin["periStimDFF", ToString[m],
    "DegROI", ToString[n]])]}]]])]); {n, 1, numROIs}, {m, orientList}];

In[ ]:= (**For each ROI,
  calculate the z-scored mean response to the presentation of each orientation**)

```

```

In[ ]:= Table[
  Evaluate@ToExpression[StringJoin["zScoredRespAt", ToString[m], "DegROI", ToString[n]]] =
    (Mean[Part[#, 2] & /@ Mean[Table[N@Take[(ToExpression[
      StringJoin["periStimZscoredDFF", ToString[m], "DegROI", ToString[n]]][[
        k]], {tpFrameRate + 1, (tpFrameRate + 1) + (2 * tpFrameRate)}],
      {k, 1, Length[(ToExpression[StringJoin["periStimZscoredDFF", ToString[m],
        "DegROI", ToString[n]]])}]]]]]; {n, 1, numROIs}, {m, orientList}];

In[ ]:= (**For visually selective ROIs, compute orientation selectivity indices,
and visualize tuning curves. Orientation selectivity is computed as 1-CircVariance**)

In[ ]:= Table[Evaluate@ToExpression[StringJoin["respVsOrientROI", ToString[n]]] =
  {{0 Degree, ToExpression[StringJoin["respAt0DegROI", ToString[n]]]},
  {32 Degree, ToExpression[StringJoin["respAt32DegROI", ToString[n]]]},
  {64 Degree, ToExpression[StringJoin["respAt64DegROI", ToString[n]]]},
  {96 Degree, ToExpression[StringJoin["respAt96DegROI", ToString[n]]]},
  {128 Degree, ToExpression[StringJoin["respAt128DegROI", ToString[n]]]},
  {160 Degree, ToExpression[StringJoin["respAt160DegROI", ToString[n]]]},
  {192 Degree, ToExpression[StringJoin["respAt192DegROI", ToString[n]]]},
  {224 Degree, ToExpression[StringJoin["respAt224DegROI", ToString[n]]]},
  {256 Degree, ToExpression[StringJoin["respAt256DegROI", ToString[n]]]},
  {288 Degree, ToExpression[StringJoin["respAt288DegROI", ToString[n]]]},
  {320 Degree, ToExpression[StringJoin["respAt320DegROI", ToString[n]]]}, {352 Degree,
  ToExpression[StringJoin["respAt352DegROI", ToString[n]]]}}; {n, 1, numROIs}];

In[ ]:= Table[Evaluate@ToExpression[StringJoin["LorientROI", ToString[n]]] =
  Abs[(Sum[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 2]] *
    Exp[2 * I * (ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 1]]],
    {k, 1, Length[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]])}]] /
    (Sum[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 2]],
    {k, 1, Length[(ToExpression[
      StringJoin["respVsOrientROI", ToString[n]]])}]])]], {n, 1, numROIs}];

In[ ]:= (**Sqrt[(Sum[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 2]] *
  Sin[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 1]]],
  {k, 1, Length[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]])}]])] /
  (Sum[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 2]],
  {k, 1, Length[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]])}]]))^2 +
  ((Sum[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 2]] *
  Cos[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 1]]],
  {k, 1, Length[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]])}]])] /
  (Sum[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 2]], {k, 1,
  Length[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]])}]]))^2] ***)

```

```

In[ ]:= Table[Evaluate@ToExpression[StringJoin["LdirROI", ToString[n]]] =
  Abs[(Sum[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 2]] *
    Exp[I * (ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 1]]],
    {k, 1, Length[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 2]]]})) /
  (Sum[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 2]],
    {k, 1, Length[(ToExpression[
      StringJoin["respVsOrientROI", ToString[n]]][[k, 2]]]}))], {n, 1, numROIs}];

In[ ]:= (**For visually selective ROIs,
compute orientation selectivity indices, and visualize tuning curves
(FROM Z-SCORED DATA). Orientation selectivity is computed as 1-CircVariance**)

In[ ]:= Table[Evaluate@ToExpression[StringJoin["zScoredRespVsOrientROI", ToString[n]]] =
  {{0 Degree, ToExpression[StringJoin["zScoredRespAt0DegROI", ToString[n]]],
  {32 Degree, ToExpression[StringJoin["zScoredRespAt32DegROI", ToString[n]]],
  {64 Degree, ToExpression[StringJoin["zScoredRespAt64DegROI", ToString[n]]],
  {96 Degree, ToExpression[StringJoin["zScoredRespAt96DegROI", ToString[n]]],
  {128 Degree, ToExpression[StringJoin["zScoredRespAt128DegROI", ToString[n]]],
  {160 Degree, ToExpression[StringJoin["zScoredRespAt160DegROI", ToString[n]]],
  {192 Degree, ToExpression[StringJoin["zScoredRespAt192DegROI", ToString[n]]],
  {224 Degree, ToExpression[StringJoin["zScoredRespAt224DegROI", ToString[n]]],
  {256 Degree, ToExpression[StringJoin["zScoredRespAt256DegROI", ToString[n]]],
  {288 Degree, ToExpression[StringJoin["zScoredRespAt288DegROI", ToString[n]]],
  {320 Degree, ToExpression[StringJoin["zScoredRespAt320DegROI", ToString[n]]],
  {352 Degree, ToExpression[
    StringJoin["zScoredRespAt352DegROI", ToString[n]]}}], {n, 1, numROIs}];

In[ ]:= Table[Evaluate@ToExpression[StringJoin["LorientZscoreROI", ToString[n]]] =
  Abs[(Sum[(ToExpression[StringJoin["zScoredRespVsOrientROI", ToString[n]]][[k, 2]] *
    Exp[2 * I * (ToExpression[StringJoin["zScoredRespVsOrientROI", ToString[n]]][[
      k, 1]]], {k, 1, Length[
      (ToExpression[StringJoin["zScoredRespVsOrientROI", ToString[n]]][[k, 2]])})) /
  (Sum[(ToExpression[StringJoin["zScoredRespVsOrientROI", ToString[n]]][[k, 2]],
    {k, 1, Length[(ToExpression[
      StringJoin["zScoredRespVsOrientROI", ToString[n]]][[k, 2]]]}))], {n, 1, numROIs}];

In[ ]:= (**Sqrt[(Sum[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 2]] *
  Sin[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 1]]],
  {k, 1, Length[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 2]]]})) /
  (Sum[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 2]],
  {k, 1, Length[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 2]]]}))]^2+
  ((Sum[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 2]] *
    Cos[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 1]]],
    {k, 1, Length[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 2]]]})) /
  (Sum[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 2]], {k, 1,
    Length[(ToExpression[StringJoin["respVsOrientROI", ToString[n]]][[k, 2]]]}))]^2)**)

```



```

In[ ]:= Table[Evaluate@ToExpression[StringJoin["LdirZscoreROI", ToString[n]]] =
  Abs[(Sum[(ToExpression[StringJoin["zScoredRespVsOrientROI", ToString[n]]][[k, 2]] *
    Exp[I * (ToExpression[StringJoin["zScoredRespVsOrientROI", ToString[n]]][[
      k, 1]]], {k, 1, Length[
        (ToExpression[StringJoin["zScoredRespVsOrientROI", ToString[n]]][[]]) /
        (Sum[(ToExpression[StringJoin["zScoredRespVsOrientROI", ToString[n]]][[k, 2]],
          {k, 1, Length[(ToExpression[
            StringJoin["zScoredRespVsOrientROI", ToString[n]]][[]])])], {n, 1, numROIs}];

In[ ]:= (**Visualize the orientation responses and orientation tuning curves for each ROI**)

```

```

In[ ]:= Manipulate[
  GraphicsGrid[{{ListLinePlot[Mean[ToExpression[StringJoin["periStimDFFb1Sub0DegROI",
    ToString[ROI]]]], PlotLabel → "0 Degrees",
    PlotRange → {All, {ToExpression[StringJoin["minRespValOrient", ToString[ROI]]],
      ToExpression[StringJoin["maxRespValOrient", ToString[ROI]]}}}], ListLinePlot[
    Mean[ToExpression[StringJoin["periStimDFFb1Sub32DegROI", ToString[ROI]]]],
    PlotLabel → "32 Degrees",
    PlotRange → {All, {ToExpression[StringJoin["minRespValOrient", ToString[ROI]]],
      ToExpression[StringJoin["maxRespValOrient", ToString[ROI]]}}}], ListLinePlot[
    Mean[ToExpression[StringJoin["periStimDFFb1Sub64DegROI", ToString[ROI]]]],
    PlotLabel → "64 Degrees",
    PlotRange → {All, {ToExpression[StringJoin["minRespValOrient", ToString[ROI]]],
      ToExpression[StringJoin["maxRespValOrient", ToString[ROI]]}}}], ListLinePlot[
    Mean[ToExpression[StringJoin["periStimDFFb1Sub96DegROI", ToString[ROI]]]],
    PlotLabel → "96 Degrees",
    PlotRange → {All, {ToExpression[StringJoin["minRespValOrient", ToString[ROI]]],
      ToExpression[StringJoin["maxRespValOrient", ToString[ROI]]}}}], ListLinePlot[
    Mean[ToExpression[StringJoin["periStimDFFb1Sub128DegROI", ToString[ROI]]]],
    PlotLabel → "128 Degrees",
    PlotRange → {All, {ToExpression[StringJoin["minRespValOrient", ToString[ROI]]],
      ToExpression[StringJoin["maxRespValOrient", ToString[ROI]]}}}], ListLinePlot[
    Mean[ToExpression[StringJoin["periStimDFFb1Sub160DegROI", ToString[ROI]]]],
    PlotLabel → "160 Degrees",
    PlotRange → {All, {ToExpression[StringJoin["minRespValOrient", ToString[ROI]]],
      ToExpression[StringJoin["maxRespValOrient", ToString[ROI]]}}}], ListLinePlot[
    Mean[ToExpression[StringJoin["periStimDFFb1Sub192DegROI", ToString[ROI]]]],
    PlotLabel → "192 Degrees",
    PlotRange → {All, {ToExpression[StringJoin["minRespValOrient", ToString[ROI]]],
      ToExpression[StringJoin["maxRespValOrient", ToString[ROI]]}}}], ListLinePlot[
    Mean[ToExpression[StringJoin["periStimDFFb1Sub224DegROI", ToString[ROI]]]],
    PlotLabel → "224 Degrees",
    PlotRange → {All, {ToExpression[StringJoin["minRespValOrient", ToString[ROI]]],
      ToExpression[StringJoin["maxRespValOrient", ToString[ROI]]}}}], ListLinePlot[
    Mean[ToExpression[StringJoin["periStimDFFb1Sub256DegROI", ToString[ROI]]]],
    PlotLabel → "256 Degrees",
    PlotRange → {All, {ToExpression[StringJoin["minRespValOrient", ToString[ROI]]],
      ToExpression[StringJoin["maxRespValOrient", ToString[ROI]]}}}], ListLinePlot[
    Mean[ToExpression[StringJoin["periStimDFFb1Sub288DegROI", ToString[ROI]]]],
    PlotLabel → "288 Degrees",
    PlotRange → {All, {ToExpression[StringJoin["minRespValOrient", ToString[ROI]]],
      ToExpression[StringJoin["maxRespValOrient", ToString[ROI]]}}}], ListLinePlot[
    Mean[ToExpression[StringJoin["periStimDFFb1Sub320DegROI", ToString[ROI]]]],
    PlotLabel → "320 Degrees",
    PlotRange → {All, {ToExpression[StringJoin["minRespValOrient", ToString[ROI]]],
      ToExpression[StringJoin["maxRespValOrient", ToString[ROI]]}}}], ListLinePlot[
    Mean[ToExpression[StringJoin["periStimDFFb1Sub352DegROI", ToString[ROI]]]],
    PlotLabel → "352 Degrees",
    PlotRange → {All, {ToExpression[StringJoin["minRespValOrient", ToString[ROI]]],
      ToExpression[StringJoin["maxRespValOrient", ToString[ROI]]}}}],
    ImageSize → Full], {ROI, sigRespROIs}]

```

```

In[ ]:= Manipulate[ListPolarPlot[ToExpression[StringJoin["respVsOrientROI", ToString[ROI]]],
  Joined → True, PolarGridLines → Automatic, PolarAxes → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotLabel → StringJoin["Lorient = ", ToString[
    ToExpression[StringJoin["LorientROI", ToString[ROI]]]]], {ROI, sigRespROIs}]

In[ ]:= (*****Table[Evaluate@
  ToExpression[StringJoin["dFFdataPoints", ToString[deg], "Deg", "ROI", ToString[roi]]] =
  Table[( (Mean[Part[#, 2] & @Take[ (ToExpression[StringJoin["periStimDFF",
    ToString@deg, "Deg", "ROI", ToString@roi]])[[n]],
    {tpFrameRate, tpFrameRate + (tpFrameRate*2)}]] -
    (Mean[Part[#, 2] & @Take[ (ToExpression[StringJoin["periStimDFF",
    ToString@deg, "Deg", "ROI", ToString@roi]])[[n]], tpFrameRate]])) /
    (Mean[Part[#, 2] & @Take[ (ToExpression[StringJoin["periStimDFF",
    ToString@deg, "Deg", "ROI", ToString@roi]])[[n]], tpFrameRate]]),
  {n, 1, Length[(ToExpression[StringJoin["periStimDFF", ToString@deg, "Deg",
    "ROI", ToString@roi]])]}], {roi, 1, numROIs}, {deg, orientList}]***)

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

In[ ]:= (**Export the P-values for visual responsiveness of each ROI**)

In[ ]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/",
  mouse, "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse,
  "_Session", ToString[sessionNum], "_", "visResp_pVal", ToString[n], ".txt"],
  ToExpression[StringJoin["visRespPValROI", ToString[n]]]], {n, 1, numROIs}];

In[ ]:= (**Export the list of the identities of the significantly responsive ROIs**)

In[ ]:= Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/", mouse,
  "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse, "_Session",
  ToString[sessionNum], "_", "sigResponsiveGratingROIs", ".txt"], sigRespROIs]

Out[ ]:= C:/Users/garrett/Desktop/Garrett_Local/090522/Mouse23015/Session2/VisStimResults/090522
_Mouse23015_Session2_sigResponsiveGratingROIs.txt

In[ ]:= (**Export dF/F values for overall visual responsiveness for each ROI**)

In[ ]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/",
  mouse, "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse,
  "_Session", ToString[sessionNum], "_", "overallVisDFF_ROI", ToString[n], ".txt"],
  ToExpression[StringJoin["overallVisDFF", ToString[n]]]], {n, 1, numROIs}];

In[ ]:= (**Export z-scored dF/F values for overall visual responsiveness for each ROI**)

In[ ]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/", mouse,
  "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse, "_Session",
  ToString[sessionNum], "_", "overallVisDFFZScored_ROI", ToString[n], ".txt"],
  ToExpression[StringJoin["overallVisZScore", ToString[n]]]], {n, 1, numROIs}];

In[ ]:= (**Export the mean, baseline subtracted dF/F trace for each ROI**)

```

```

In[ ]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/", mouse,
"/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse, "_Session",
ToString[sessionNum], "_", "overallVisDFFtrace_ROI", ToString[n], ".txt"],
N@ToExpression[StringJoin["meanBLSubtractEvokedDFFsTS", ToString[n]]]], {n,
1, numROIs}];

In[ ]:= (**Export the mean, z-scored dF/F trace for each ROI**)

In[ ]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/", mouse,
"/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse, "_Session",
ToString[sessionNum], "_", "overallVisZscoredDFFtrace_ROI", ToString[n], ".txt"],
N@ToExpression[StringJoin["meanZscoredDFFsTS", ToString[n]]]], {n, 1, numROIs}];

In[ ]:= (**Export the orientation tuning data for each significantly responsive ROI**)

In[ ]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/",
mouse, "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse,
"_Session", ToString[sessionNum], "_", "orientTuning_ROI", ToString[n], ".txt"],
ToExpression[StringJoin["respVsOrientROI", ToString[n]]]], {n, 1, numROIs}];

In[ ]:= (**Export the Z-
SCORED orientation tuning data for each significantly responsive ROI**)

In[ ]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/", mouse,
"/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse, "_Session",
ToString[sessionNum], "_", "orientTuningZscored_ROI", ToString[n], ".txt"],
ToExpression[StringJoin["zScoredRespVsOrientROI", ToString[n]]]], {n, 1, numROIs}];

In[ ]:= (**Export the OSI for each significantly responsive ROI**)

In[ ]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/",
mouse, "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse,
"_Session", ToString[sessionNum], "_", "Lorient_ROI", ToString[n], ".txt"],
ToExpression[StringJoin["LorientROI", ToString[n]]]], {n, 1, numROIs}];

In[ ]:= (**Export the Z-SCORED OSI for each significantly responsive ROI**)

In[ ]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/",
mouse, "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse,
"_Session", ToString[sessionNum], "_", "LorientZscored_ROI", ToString[n], ".txt"],
ToExpression[StringJoin["LorientZscoreROI", ToString[n]]]], {n, 1, numROIs}];

In[ ]:= (**Export the DSI for each significantly responsive ROI**)

In[ ]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/",
mouse, "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse,
"_Session", ToString[sessionNum], "_", "Ldir_ROI", ToString[n], ".txt"],
ToExpression[StringJoin["LdirROI", ToString[n]]]], {n, 1, numROIs}];

In[ ]:= (**Export the Z-SCORED DSI for each significantly responsive ROI**)

In[ ]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/",
mouse, "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse,
"_Session", ToString[sessionNum], "_", "LdirZscored_ROI", ToString[n], ".txt"],
ToExpression[StringJoin["LdirZscoreROI", ToString[n]]]], {n, 1, numROIs}];

```