

```

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["C:/Users/garrett/Desktop/Garrett_Local/", 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["C:/Users/garrett/Desktop/Garrett_Local/", date,
    "/", mouse, "/Session", ToString[sessionNum], "/dFoverF0TimeSeries/"]]]];

In[*]:= 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}];

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

In[*]:= 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];

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, 1, numROIs}];

In[*]:= (**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[ ]:= stimProps = Take[stimProps, Length[vStimOnsets]];

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

In[ ]:= sfs = Round /@ ((Part[#, 4] & /@ stimProps) * 100);

In[ ]:= tfs = Round /@ ((Part[#, 3] & /@ stimProps) * 10);

In[ ]:= stimProps = Partition[Riffle[sfs, tfs], 2];

In[ ]:= sFTFPairs = Part[#, 1] & /@ Sort[Tally[stimProps]];

In[ ]:= speeds = Table[N@stimProps[[n, 2]] / stimProps[[n, 1]], {n, 1, Length[stimProps]}];

In[ ]:= possibleSpeeds = Part[#, 1] & /@ Sort[Tally[speeds]];

```

```

In[ ]:= possibleSFs = {2, 4, 8, 16, 32};

In[ ]:= possibleTFs = {5, 10, 20, 40, 80, 150, 240};

In[ ]:= (**Find the positions at which each of the 35 different SF/TF combos was presented**)

In[ ]:= pos2sf5tf = Flatten[Position[stimProps, {2, 5}]];

In[ ]:= pos4sf5tf = Flatten[Position[stimProps, {4, 5}]];

In[ ]:= pos8sf5tf = Flatten[Position[stimProps, {8, 5}]];

In[ ]:= pos16sf5tf = Flatten[Position[stimProps, {16, 5}]];

In[ ]:= pos32sf5tf = Flatten[Position[stimProps, {32, 5}]];

In[ ]:= (**)

In[ ]:= pos2sf10tf = Flatten[Position[stimProps, {2, 10}]];

In[ ]:= pos4sf10tf = Flatten[Position[stimProps, {4, 10}]];

In[ ]:= pos8sf10tf = Flatten[Position[stimProps, {8, 10}]];

In[ ]:= pos16sf10tf = Flatten[Position[stimProps, {16, 10}]];

In[ ]:= pos32sf10tf = Flatten[Position[stimProps, {32, 10}]];

In[ ]:= (**)

In[ ]:= pos2sf20tf = Flatten[Position[stimProps, {2, 20}]];

In[ ]:= pos4sf20tf = Flatten[Position[stimProps, {4, 20}]];

In[ ]:= pos8sf20tf = Flatten[Position[stimProps, {8, 20}]];

In[ ]:= pos16sf20tf = Flatten[Position[stimProps, {16, 20}]];

In[ ]:= pos32sf20tf = Flatten[Position[stimProps, {32, 20}]];

In[ ]:= (**)

In[ ]:= pos2sf40tf = Flatten[Position[stimProps, {2, 40}]];

In[ ]:= pos4sf40tf = Flatten[Position[stimProps, {4, 40}]];

In[ ]:= pos8sf40tf = Flatten[Position[stimProps, {8, 40}]];

In[ ]:= pos16sf40tf = Flatten[Position[stimProps, {16, 40}]];

In[ ]:= pos32sf40tf = Flatten[Position[stimProps, {32, 40}]];

In[ ]:= (**)

In[ ]:= pos2sf80tf = Flatten[Position[stimProps, {2, 80}]];

In[ ]:= pos4sf80tf = Flatten[Position[stimProps, {4, 80}]];

In[ ]:= pos8sf80tf = Flatten[Position[stimProps, {8, 80}]];

In[ ]:= pos16sf80tf = Flatten[Position[stimProps, {16, 80}]];

```

```

In[ ]:= pos32sf80tf = Flatten[Position[stimProps, {32, 80}]];

In[ ]:= (**)

In[ ]:= pos2sf150tf = Flatten[Position[stimProps, {2, 150}]];

In[ ]:= pos4sf150tf = Flatten[Position[stimProps, {4, 150}]];

In[ ]:= pos8sf150tf = Flatten[Position[stimProps, {8, 150}]];

In[ ]:= pos16sf150tf = Flatten[Position[stimProps, {16, 150}]];

In[ ]:= pos32sf150tf = Flatten[Position[stimProps, {32, 150}]];

In[ ]:= (**)

In[ ]:= pos2sf240tf = Flatten[Position[stimProps, {2, 240}]];

In[ ]:= pos4sf240tf = Flatten[Position[stimProps, {4, 240}]];

In[ ]:= pos8sf240tf = Flatten[Position[stimProps, {8, 240}]];

In[ ]:= pos16sf240tf = Flatten[Position[stimProps, {16, 240}]];

In[ ]:= pos32sf240tf = Flatten[Position[stimProps, {32, 240}]];

In[ ]:= (**)

In[ ]:= (***Find the positions at which each of the 5 different spatial freqs was presented,
regardless of temporal freq. Note values are
multiplied by 100 to make them integers (e.g. 2 = 0.02 cpd)***)

In[ ]:= pos2sf = Flatten[Position[stimProps, {2, _}]];

In[ ]:= pos4sf = Flatten[Position[stimProps, {4, _}]];

In[ ]:= pos8sf = Flatten[Position[stimProps, {8, _}]];

In[ ]:= pos16sf = Flatten[Position[stimProps, {16, _}]];

In[ ]:= pos32sf = Flatten[Position[stimProps, {32, _}]];

In[ ]:= (***Find the positions at which each of the 7 different temporal freqs was presented,
regardless of spatial freq. Note values are
multiplied by 10 to make them integers (e.g. 5 = 0.5 cps)***)

In[ ]:= pos5tf = Flatten[Position[stimProps, {_, 5}]];

In[ ]:= pos10tf = Flatten[Position[stimProps, {_, 10}]];

In[ ]:= pos20tf = Flatten[Position[stimProps, {_, 20}]];

In[ ]:= pos40tf = Flatten[Position[stimProps, {_, 40}]];

In[ ]:= pos80tf = Flatten[Position[stimProps, {_, 80}]];

In[ ]:= pos150tf = Flatten[Position[stimProps, {_, 150}]];

In[ ]:= pos240tf = Flatten[Position[stimProps, {_, 240}]];

```

```

In[ ]:= (**Find the positions at which each distinct speed (TF/SF) was presented**)

In[ ]:= pos3speed = Flatten[Position[speeds, 0.3125]];

In[ ]:= pos6speed = Flatten[Position[speeds, 0.625]];

In[ ]:= pos13speed = Flatten[Position[speeds, 1.25]];

In[ ]:= pos25speed = Flatten[Position[speeds, 2.5]];

In[ ]:= pos50speed = Flatten[Position[speeds, 5.]];

In[ ]:= pos100speed = Flatten[Position[speeds, 10.]];

In[ ]:= pos200speed = Flatten[Position[speeds, 20.]];

In[ ]:= pos400speed = Flatten[Position[speeds, 40.]];

In[ ]:= pos1200speed = Flatten[Position[speeds, 120.]];

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

In[ ]:= Table[{StringJoin["ROI", 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[ ]:= (*****SF/TF Combos*****)

(**For each ROI, collect the peri-stimulus dF/F traces for each SF/TF combo**)

In[ ]:= Table[
    Table[Evaluate@ToExpression[StringJoin["periStimDFFsf", ToString[j], "tf", ToString[k],
        ToString[n]]] = (ToExpression[StringJoin["evokedRawDFFsTS", ToString[n]]][[
        ToExpression[StringJoin["pos", ToString[j], "sf", ToString[k], "tf"]]]];,
        {n, 1, numROIs}];, {j, {2, 4, 8, 16, 32}}, {k, {5, 10, 20, 40, 80, 150, 240}}];

In[ ]:= Table[Table[Evaluate@ToExpression[
    StringJoin["periStimDFFblSubsf", ToString[j], "tf", ToString[k], ToString[n]]] =
    (ToExpression[StringJoin["blSubtractEvokedDFFsTS", ToString[n]]][[
        ToExpression[StringJoin["pos", ToString[j], "sf", ToString[k], "tf"]]]];,
        {n, 1, numROIs}];, {j, {2, 4, 8, 16, 32}}, {k, {5, 10, 20, 40, 80, 150, 240}}];

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

```

```

In[ ]:= Table[Table[Evaluate@ToExpression[
    StringJoin["periStimZscoredDFFsf", ToString[j], "tf", ToString[k], ToString[n]] =
    (ToExpression[StringJoin["zScoredDFFsTS", ToString[n]]][[
        ToExpression[StringJoin["pos", ToString[j], "sf", ToString[k], "tf"]]]];,
    {n, 1, numROIs}];, {j, {2, 4, 8, 16, 32}}, {k, {5, 10, 20, 40, 80, 150, 240}}];

In[ ]:= (**Calculate significance of visual
    responsiveness for each ROI. To be significantly responsive,
    at least one SF/TF combo must drive the cell to significance,  $p/35 < 0.05$ **)

In[ ]:= Table[Evaluate@ToExpression[
    StringJoin["evokedRawDFFs", "sf", ToString[j], "tf", ToString[k], ToString[n]] =
    Table[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]][[i],
        {i, vStimOnsets[[x]] - 1, vStimOnsets[[x]] + 5, 1/tpFrameRate}],
        {x, ToExpression[StringJoin["pos", ToString[j], "sf", ToString[k], "tf"]]]];,
    {n, 1, numROIs}, {j, {2, 4, 8, 16, 32}}, {k, {5, 10, 20, 40, 80, 150, 240}}];

In[ ]:= Table[Evaluate@ToExpression[StringJoin["visRespPVal",
    "sf", ToString[j], "tf", ToString[k], "ROI", ToString[n]] =
    Quiet[TTest[{Mean /@ (Take[#, tpFrameRate] & /@ (ToExpression[StringJoin[
        "evokedRawDFFs", "sf", ToString[j], "tf", ToString[k], ToString[n]]])),
        Mean /@ (Take[#, {tpFrameRate + 1, (tpFrameRate + 1) + (2 * tpFrameRate)}]) & /@
        (ToExpression[StringJoin["evokedRawDFFs", "sf", ToString[j], "tf",
            ToString[k], ToString[n]]])),}, AlternativeHypothesis -> "Less"]];,
    {n, 1, numROIs}, {j, {2, 4, 8, 16, 32}}, {k, {5, 10, 20, 40, 80, 150,
        240}}];

In[ ]:= Table[Evaluate@ToExpression[StringJoin["pValList", "ROI", ToString[n]] =
    Flatten@Table[ToExpression[StringJoin["visRespPVal",
        "sf", ToString[j], "tf", ToString[k], "ROI", ToString[n]]],
        {j, {2, 4, 8, 16, 32}}, {k, {5, 10, 20, 40, 80, 150, 240}}];, {n, 1, numROIs}];

In[ ]:= sigRespROIs =
    DeleteCases[Table[If[AnyTrue[ToExpression[StringJoin["pValList", "ROI", ToString[n]]],
        # < (0.05 / 35) &], n, Null], {n, 1, numROIs}], Null];

In[ ]:= (*****Speed tuning*****)

(**For each ROI, collect the peri-stimulus dF/F traces for each speed**)

In[ ]:= Table[Table[Evaluate@
    ToExpression[StringJoin["periStimDFF", ToString[m], "SpeedROI", ToString[n]] =
    (ToExpression[StringJoin["evokedRawDFFsTS", ToString[n]]][[
        ToExpression[StringJoin["pos", ToString[m], "speed"]]]];,
    {n, 1, numROIs}];, {m, {3, 6, 13, 25, 50, 100, 200, 400, 1200}}];

In[ ]:= Table[Table[Evaluate@ToExpression[
    StringJoin["periStimDFFblSub", ToString[m], "SpeedROI", ToString[n]] =
    (ToExpression[StringJoin["blSubtractEvokedDFFsTS", ToString[n]]][[
        ToExpression[StringJoin["pos", ToString[m], "speed"]]]];,
    {n, 1, numROIs}];, {m, {3, 6, 13, 25, 50, 100, 200, 400, 1200}}];

```

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

```
ln[ ]:= Table[
  Table[Evaluate@ToExpression[StringJoin["periStimZscoredDFF", ToString[m], "SpeedROI",
    ToString[n]]] = (ToExpression[StringJoin["zScoredDFFsTS", ToString[n]]][[
      ToExpression[StringJoin["pos", ToString[m], "speed"]]]];,
    {n, 1, numROIs}];, {m, {3, 6, 13, 25, 50, 100, 200, 400, 1200}}];
```

```
ln[ ]:= (**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**)
```

```
ln[ ]:= Table[Evaluate@ToExpression[StringJoin["respAt", ToString[m], "SpeedROI", ToString[n]]] =
  ((Mean[Part[#, 2] & /@
    Mean[Table[N@Take[(ToExpression[StringJoin["periStimDFF", ToString[m],
      "SpeedROI", ToString[n]]][[k]], {tpFrameRate + 1, (tpFrameRate + 1) +
      (2 * tpFrameRate)}], {k, 1, Length[(ToExpression[StringJoin[
      "periStimDFF", ToString[m], "SpeedROI", ToString[n]]][[k]]]}]] -
    (Mean[Part[#, 2] & /@ Mean[Table[N@Take[(ToExpression[StringJoin["periStimDFF",
      ToString[m], "SpeedROI", ToString[n]]][[k]], tpFrameRate],
      {k, 1, Length[(ToExpression[StringJoin["periStimDFF", ToString[m],
      "SpeedROI", ToString[n]]][[k]]]}]]]) /
    (Mean[Part[#, 2] & /@ Mean[Table[N@Take[(ToExpression[StringJoin["periStimDFF",
      ToString[m], "SpeedROI", ToString[n]]][[k]], tpFrameRate],
      {k, 1, Length[(ToExpression[StringJoin["periStimDFF", ToString[
      m], "SpeedROI", ToString[n]]][[k]]]}]]])];,
    {n, 1, numROIs}, {m, {3, 6, 13, 25, 50, 100, 200,
      400,
      1200}}];
```

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

```
ln[ ]:= Table[Evaluate@
  ToExpression[StringJoin["zScoredRespAt", ToString[m], "SpeedROI", ToString[n]]] =
  (Mean[Part[#, 2] & /@ Mean[Table[N@Take[(ToExpression[StringJoin[
    "periStimZscoredDFF", ToString[m], "SpeedROI", ToString[n]]][[k]],
    {tpFrameRate + 1, (tpFrameRate + 1) + (2 * tpFrameRate)}],
    {k, 1, Length[(ToExpression[StringJoin["periStimZscoredDFF",
      ToString[m], "SpeedROI", ToString[n]]][[k]]]}]]];,
    {n, 1, numROIs}, {m, {3, 6, 13, 25, 50, 100, 200, 400, 1200}}];
```

```
ln[ ]:= (**For each ROI,
  calculate the z-scored mean response to the presentation of each SF/TF combo**)
```



```

In[ ]:= Table[
  Evaluate@ToExpression[StringJoin["zScoredRespAt", ToString[m], "SF", ToString[p], "TF",
    "ROI", ToString[n]]] = (Mean[Part[#, 2] & /@
    Mean[Table[N@Take[(ToExpression[StringJoin["periStimZscoredDFFsf",
      ToString[m], "tf", ToString[p], ToString[n]]][[k]],
      {tpFrameRate + 1, (tpFrameRate + 1) + (2 * tpFrameRate)}],
      {k, 1, Length[(ToExpression[StringJoin["periStimZscoredDFF",
        ToString[m], "SpeedROI", ToString[n]]][[k]]}]]]]];,
  {n, 1, numROIs}, {m, {2, 4, 8, 16, 32}}, {p, {5, 10, 20, 40, 80,
    150,
    240}}];

```

```

In[ ]:= (**Speed tuning curves for each ROI**)

```

```

In[ ]:= Table[Evaluate@ToExpression[StringJoin["respVsSpeedROI", ToString[n]]] =
  {{3, ToExpression[StringJoin["respAt3SpeedROI", ToString[n]]]},
  {6, ToExpression[StringJoin["respAt6SpeedROI", ToString[n]]]},
  {13, ToExpression[StringJoin["respAt13SpeedROI", ToString[n]]]},
  {25, ToExpression[StringJoin["respAt25SpeedROI", ToString[n]]]},
  {50, ToExpression[StringJoin["respAt50SpeedROI", ToString[n]]]},
  {100, ToExpression[StringJoin["respAt100SpeedROI", ToString[n]]]},
  {200, ToExpression[StringJoin["respAt200SpeedROI", ToString[n]]]},
  {400, ToExpression[StringJoin["respAt400SpeedROI", ToString[n]]]}, {1200,
  ToExpression[StringJoin["respAt1200SpeedROI", ToString[n]]]}}; {n, 1, numROIs}];

```

```

In[ ]:= Table[Evaluate@ToExpression[StringJoin["zScoredRespVsSpeedROI", ToString[n]]] =
  {{3, ToExpression[StringJoin["zScoredRespAt3SpeedROI", ToString[n]]]},
  {6, ToExpression[StringJoin["zScoredRespAt6SpeedROI", ToString[n]]]},
  {13, ToExpression[StringJoin["zScoredRespAt13SpeedROI", ToString[n]]]},
  {25, ToExpression[StringJoin["zScoredRespAt25SpeedROI", ToString[n]]]},
  {50, ToExpression[StringJoin["zScoredRespAt50SpeedROI", ToString[n]]]},
  {100, ToExpression[StringJoin["zScoredRespAt100SpeedROI", ToString[n]]]},
  {200, ToExpression[StringJoin["zScoredRespAt200SpeedROI", ToString[n]]]},
  {400, ToExpression[StringJoin["zScoredRespAt400SpeedROI", ToString[n]]]},
  {1200, ToExpression[
    StringJoin["zScoredRespAt1200SpeedROI", ToString[n]]]}}; {n, 1, numROIs}];

```

```

In[ ]:= (*****Spatial frequency tuning*****

```

```

  (**For each ROI, collect the peri-stimulus dF/F traces for each SF**)

```

```

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

```

```

In[ ]:= Table[Table[Evaluate@
  ToExpression[StringJoin["periStimDFFblSub", ToString[m], "SFRROI", ToString[n]]] =
    (ToExpression[StringJoin["blSubtractEvokedDFFsTS", ToString[n]]][[ToExpression[
      StringJoin["pos", ToString[m], "sf"]]]]; {n, 1, numROIs}]; {m, possibleSFs}];

```

```

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

```

```

In[ ]:= Table[Table[Evaluate@
  ToExpression[StringJoin["periStimZscoredDFF", ToString[m], "SFR0I", ToString[n]]] =
  (ToExpression[StringJoin["zScoredDFFsTS", ToString[n]]])[[ToExpression[
    StringJoin["pos", ToString[m], "sf"]]]]; {n, 1, numROIs}]; {m, possibleSFs}];

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], "SFR0I", ToString[n]]] =
  ((Mean[Part[#, 2] & /@ Mean[Table[N@Take[(ToExpression[StringJoin["periStimDFF",
    ToString[m], "SFR0I", ToString[n]]])[[k]], {tpFrameRate + 1,
    (tpFrameRate + 1) + (2 * tpFrameRate)}], {k, 1, Length[(ToExpression[
    StringJoin["periStimDFF", ToString[m], "SFR0I", ToString[n]]])}]]]] -
  (Mean[Part[#, 2] & /@ Mean[Table[N@Take[(ToExpression[StringJoin["periStimDFF",
    ToString[m], "SFR0I", ToString[n]]])[[k]], tpFrameRate],
    {k, 1, Length[(ToExpression[StringJoin["periStimDFF", ToString[m],
    "SFR0I", ToString[n]]])}]]]])) /
  (Mean[Part[#, 2] & /@ Mean[Table[N@Take[(ToExpression[StringJoin[
    "periStimDFF", ToString[m], "SFR0I", ToString[n]]])[[k]], tpFrameRate],
    {k, 1, Length[(ToExpression[StringJoin["periStimDFF", ToString[m],
    "SFR0I", ToString[n]]])}]]]]]; {n, 1, numROIs}, {m, possibleSFs}];

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

In[ ]:= Table[
  Evaluate@ToExpression[StringJoin["zScoredRespAt", ToString[m], "SFR0I", ToString[n]]] =
  (Mean[Part[#, 2] & /@ Mean[Table[N@Take[(ToExpression[
    StringJoin["periStimZscoredDFF", ToString[m], "SFR0I", ToString[n]]])[[
    k]], {tpFrameRate + 1, (tpFrameRate + 1) + (2 * tpFrameRate)}],
    {k, 1, Length[(ToExpression[StringJoin["periStimZscoredDFF", ToString[m],
    "SFR0I", ToString[n]]])}]]]]]; {n, 1, numROIs}, {m, possibleSFs}];

In[ ]:= (**SF tuning curves for each ROI**)

In[ ]:= Table[Evaluate@ToExpression[StringJoin["respVsSFR0I", ToString[n]]] =
  {{0.02, ToExpression[StringJoin["respAt2SFR0I", ToString[n]]]},
  {0.04, ToExpression[StringJoin["respAt4SFR0I", ToString[n]]]},
  {0.08, ToExpression[StringJoin["respAt8SFR0I", ToString[n]]]},
  {0.16, ToExpression[StringJoin["respAt16SFR0I", ToString[n]]]},
  {0.32, ToExpression[StringJoin["respAt32SFR0I", ToString[n]]]}}; {n, 1, numROIs}];

In[ ]:= Table[Evaluate@ToExpression[StringJoin["zScoredRespVsSFR0I", ToString[n]]] =
  {{0.02, ToExpression[StringJoin["zScoredRespAt2SFR0I", ToString[n]]]},
  {0.04, ToExpression[StringJoin["zScoredRespAt4SFR0I", ToString[n]]]},
  {0.08, ToExpression[StringJoin["zScoredRespAt8SFR0I", ToString[n]]]},
  {0.16, ToExpression[StringJoin["zScoredRespAt16SFR0I", ToString[n]]]},
  {0.32, ToExpression[StringJoin["zScoredRespAt32SFR0I", ToString[n]]]}}; {n, 1, numROIs}];

```

```

In[ ]:= (*****Temporal frequency tuning*****)

(***For each ROI, collect the peri-stimulus dF/F traces for each TF***)

In[ ]:= Table[Table[
  Evaluate@ToExpression[StringJoin["periStimDFF", ToString[m], "TFROI", ToString[n]]] =
    (ToExpression[StringJoin["evokedRawDFFsTS", ToString[n]]]) [[ToExpression[
      StringJoin["pos", ToString[m], "tf"]]]];, {n, 1, numROIs}];, {m, possibleTFs}];

In[ ]:= Table[Table[Evaluate@
  ToExpression[StringJoin["periStimDFFblSub", ToString[m], "TFROI", ToString[n]]] =
    (ToExpression[StringJoin["blSubtractEvokedDFFsTS", ToString[n]]]) [[ToExpression[
      StringJoin["pos", ToString[m], "tf"]]]];, {n, 1, numROIs}];, {m, possibleTFs}];

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

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

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

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

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

In[ ]:= (***TF tuning curves for each ROI***)

```

```
ln[ ]:= Table[Evaluate@ToExpression[StringJoin["respVsTFROI", ToString[n]]] =
  {{0.5, ToExpression[StringJoin["respAt5TFROI", ToString[n]]]},
   {1, ToExpression[StringJoin["respAt10TFROI", ToString[n]]]},
   {2, ToExpression[StringJoin["respAt20TFROI", ToString[n]]]},
   {4, ToExpression[StringJoin["respAt40TFROI", ToString[n]]]},
   {8, ToExpression[StringJoin["respAt80TFROI", ToString[n]]]},
   {15, ToExpression[StringJoin["respAt150TFROI", ToString[n]]]},
   {24, ToExpression[StringJoin["respAt240TFROI", ToString[n]]]}}]; {n, 1, numROIs}};
```

```
ln[ ]:= Table[Evaluate@ToExpression[StringJoin["zScoredRespVsTFROI", ToString[n]]] =
  {{0.5, ToExpression[StringJoin["zScoredRespAt5TFROI", ToString[n]]]},
   {1, ToExpression[StringJoin["zScoredRespAt10TFROI", ToString[n]]]},
   {2, ToExpression[StringJoin["zScoredRespAt20TFROI", ToString[n]]]},
   {4, ToExpression[StringJoin["zScoredRespAt40TFROI", ToString[n]]]},
   {8, ToExpression[StringJoin["zScoredRespAt80TFROI", ToString[n]]]}, {15,
   ToExpression[StringJoin["zScoredRespAt150TFROI", ToString[n]]]}, {24, ToExpression[
   StringJoin["zScoredRespAt240TFROI", ToString[n]]]}}]; {n, 1, numROIs}};
```

```
ln[ ]:= (**For visualization purposes (SF/TF combos):**)
```

```
ln[ ]:= Table[Evaluate@ToExpression[StringJoin["maxRespValSFTF", ToString[n]]] =
  Max[Flatten[{Part[#, 2] & /@
    Mean[ToExpression[StringJoin["periStimDFFblSubsf2tf5", ToString[n]]]],
    Part[#, 2] & /@ Mean[ToExpression[StringJoin["periStimDFFblSubsf2tf10",
      ToString[n]]]], Part[#, 2] & /@ Mean[ToExpression[
      StringJoin["periStimDFFblSubsf2tf20", ToString[n]]]], Part[#, 2] & /@
    Mean[ToExpression[StringJoin["periStimDFFblSubsf2tf40", ToString[n]]]],
    Part[#, 2] & /@ Mean[ToExpression[StringJoin[
      "periStimDFFblSubsf2tf80", ToString[n]]]], Part[#, 2] & /@
    Mean[ToExpression[StringJoin["periStimDFFblSubsf2tf150", ToString[n]]]],
    Part[#, 2] & /@ Mean[ToExpression[StringJoin[
      "periStimDFFblSubsf2tf240", ToString[n]]]], Part[#, 2] & /@
    Mean[ToExpression[StringJoin["periStimDFFblSubsf4tf5", ToString[n]]]],
    Part[#, 2] & /@ Mean[ToExpression[StringJoin[
      "periStimDFFblSubsf4tf10", ToString[n]]]], Part[#, 2] & /@
    Mean[ToExpression[StringJoin["periStimDFFblSubsf4tf20", ToString[n]]]],
    Part[#, 2] & /@ Mean[ToExpression[StringJoin[
      "periStimDFFblSubsf4tf40", ToString[n]]]], Part[#, 2] & /@
    Mean[ToExpression[StringJoin["periStimDFFblSubsf4tf80", ToString[n]]]],
    Part[#, 2] & /@ Mean[ToExpression[StringJoin[
      "periStimDFFblSubsf4tf150", ToString[n]]]], Part[#, 2] & /@
    Mean[ToExpression[StringJoin["periStimDFFblSubsf4tf240", ToString[n]]]],
    Part[#, 2] & /@ Mean[ToExpression[StringJoin[
      "periStimDFFblSubsf8tf5", ToString[n]]]], Part[#, 2] & /@
    Mean[ToExpression[StringJoin["periStimDFFblSubsf8tf10", ToString[n]]]],
    Part[#, 2] & /@ Mean[ToExpression[StringJoin[
      "periStimDFFblSubsf8tf20", ToString[n]]]], Part[#, 2] & /@
    Mean[ToExpression[StringJoin["periStimDFFblSubsf8tf40", ToString[n]]]],
    Part[#, 2] & /@ Mean[ToExpression[StringJoin[
```

```

periStimDFFb1Subsf8tf80", ToString[n]]], Part[#, 2] & /@
Mean[(ToExpression[StringJoin["periStimDFFb1Subsf8tf150", ToString[n]])]],
Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
"periStimDFFb1Subsf8tf240", ToString[n]])]], Part[#, 2] & /@
Mean[(ToExpression[StringJoin["periStimDFFb1Subsf16tf5", ToString[n]])]],
Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
"periStimDFFb1Subsf16tf10", ToString[n]])]], Part[#, 2] & /@
Mean[(ToExpression[StringJoin["periStimDFFb1Subsf16tf20", ToString[n]])]],
Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
"periStimDFFb1Subsf16tf40", ToString[n]])]], Part[#, 2] & /@
Mean[(ToExpression[StringJoin["periStimDFFb1Subsf16tf80", ToString[n]])]],
Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
"periStimDFFb1Subsf16tf150", ToString[n]])]], Part[#, 2] & /@
Mean[(ToExpression[StringJoin["periStimDFFb1Subsf16tf240", ToString[n]])]],
Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
"periStimDFFb1Subsf32tf5", ToString[n]])]], Part[#, 2] & /@
Mean[(ToExpression[StringJoin["periStimDFFb1Subsf32tf10", ToString[n]])]],
Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
"periStimDFFb1Subsf32tf20", ToString[n]])]], Part[#, 2] & /@
Mean[(ToExpression[StringJoin["periStimDFFb1Subsf32tf40", ToString[n]])]],
Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
"periStimDFFb1Subsf32tf80", ToString[n]])]], Part[#, 2] & /@
Mean[(ToExpression[StringJoin["periStimDFFb1Subsf32tf150", ToString[n]])]],
Part[#, 2] & /@ Mean[(ToExpression[StringJoin["periStimDFFb1Subsf32tf240",
ToString[n]])]]];]; {n, 1, numROIs}];];

```

```
ln[*]:= Table[Evaluate@ToExpression[StringJoin["minRespValSFTF", ToString[n]]] =
Min[Flatten[{Part[#, 2] & /@
Mean[(ToExpression[StringJoin["periStimDFFblSubsf2tf5", ToString[n]]])],
Part[#, 2] & /@ Mean[(ToExpression[StringJoin["periStimDFFblSubsf2tf10",
ToString[n]]])], Part[#, 2] & /@ Mean[(ToExpression[
StringJoin["periStimDFFblSubsf2tf20", ToString[n]]])], Part[#, 2] & /@
Mean[(ToExpression[StringJoin["periStimDFFblSubsf2tf40", ToString[n]]])],
Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
"periStimDFFblSubsf2tf80", ToString[n]]])], Part[#, 2] & /@
Mean[(ToExpression[StringJoin["periStimDFFblSubsf2tf150", ToString[n]]])],
Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
"periStimDFFblSubsf2tf240", ToString[n]]])], Part[#, 2] & /@
Mean[(ToExpression[StringJoin["periStimDFFblSubsf4tf5", ToString[n]]])],
Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
"periStimDFFblSubsf4tf10", ToString[n]]])], Part[#, 2] & /@
Mean[(ToExpression[StringJoin["periStimDFFblSubsf4tf20", ToString[n]]])],
Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
"periStimDFFblSubsf4tf40", ToString[n]]])], Part[#, 2] & /@
Mean[(ToExpression[StringJoin["periStimDFFblSubsf4tf80", ToString[n]]])],
Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
"periStimDFFblSubsf4tf150", ToString[n]]])], Part[#, 2] & /@
```

```

Mean[ (ToExpression[StringJoin["periStimDFFblSubsf4tf240", ToString[n]])]],
Part[#, 2] & /@ Mean[ (ToExpression[StringJoin[
    "periStimDFFblSubsf8tf5", ToString[n]])]], Part[#, 2] & /@
Mean[ (ToExpression[StringJoin["periStimDFFblSubsf8tf10", ToString[n]])]],
Part[#, 2] & /@ Mean[ (ToExpression[StringJoin[
    "periStimDFFblSubsf8tf20", ToString[n]])]], Part[#, 2] & /@
Mean[ (ToExpression[StringJoin["periStimDFFblSubsf8tf40", ToString[n]])]],
Part[#, 2] & /@ Mean[ (ToExpression[StringJoin[
    "periStimDFFblSubsf8tf80", ToString[n]])]], Part[#, 2] & /@
Mean[ (ToExpression[StringJoin["periStimDFFblSubsf8tf150", ToString[n]])]],
Part[#, 2] & /@ Mean[ (ToExpression[StringJoin[
    "periStimDFFblSubsf8tf240", ToString[n]])]], Part[#, 2] & /@
Mean[ (ToExpression[StringJoin["periStimDFFblSubsf16tf5", ToString[n]])]],
Part[#, 2] & /@ Mean[ (ToExpression[StringJoin[
    "periStimDFFblSubsf16tf10", ToString[n]])]], Part[#, 2] & /@
Mean[ (ToExpression[StringJoin["periStimDFFblSubsf16tf20", ToString[n]])]],
Part[#, 2] & /@ Mean[ (ToExpression[StringJoin[
    "periStimDFFblSubsf16tf40", ToString[n]])]], Part[#, 2] & /@
Mean[ (ToExpression[StringJoin["periStimDFFblSubsf16tf80", ToString[n]])]],
Part[#, 2] & /@ Mean[ (ToExpression[StringJoin[
    "periStimDFFblSubsf16tf150", ToString[n]])]], Part[#, 2] & /@
Mean[ (ToExpression[StringJoin["periStimDFFblSubsf16tf240", ToString[n]])]],
Part[#, 2] & /@ Mean[ (ToExpression[StringJoin[
    "periStimDFFblSubsf32tf5", ToString[n]])]], Part[#, 2] & /@
Mean[ (ToExpression[StringJoin["periStimDFFblSubsf32tf10", ToString[n]])]],
Part[#, 2] & /@ Mean[ (ToExpression[StringJoin[
    "periStimDFFblSubsf32tf20", ToString[n]])]], Part[#, 2] & /@
Mean[ (ToExpression[StringJoin["periStimDFFblSubsf32tf40", ToString[n]])]],
Part[#, 2] & /@ Mean[ (ToExpression[StringJoin[
    "periStimDFFblSubsf32tf80", ToString[n]])]], Part[#, 2] & /@
Mean[ (ToExpression[StringJoin["periStimDFFblSubsf32tf150", ToString[n]])]],
Part[#, 2] & /@ Mean[ (ToExpression[StringJoin["periStimDFFblSubsf32tf240",
    ToString[n]])]]]]; {n, 1, numROIs}];

```

ln[]:= (***)For each ROI, calculate the mean response to the presentation of each orientation, which is a new trial-averaged dF/F , where F_0 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[i], "sf", ToString[j], "tf", ToString[n]]] =
  ((Mean[Part[#, 2] & /@ Mean[Table[N@Take[(ToExpression[StringJoin["periStimDFFsf",
    ToString[i], "tf", ToString[j], ToString[n]]])][[k]],
    {tpFrameRate + 1, (tpFrameRate + 1) + (2 * tpFrameRate)}],
    {k, 1, Length[(ToExpression[StringJoin["periStimDFFsf",
    ToString[i], "tf", ToString[j], ToString[n]]])]}]]]) -
  (Mean[Part[#, 2] & /@ Mean[Table[N@Take[(ToExpression[StringJoin["periStimDFFsf",
    ToString[i], "tf", ToString[j], ToString[n]]])][[k]], tpFrameRate],
    {k, 1, Length[(ToExpression[StringJoin["periStimDFFsf", ToString[i],
    "tf", ToString[j], ToString[n]]])]}]]]) /
  (Mean[Part[#, 2] & /@ Mean[Table[N@Take[(ToExpression[StringJoin["periStimDFFsf",
    ToString[i], "tf", ToString[j], ToString[n]]])][[k]], tpFrameRate],
    {k, 1, Length[(ToExpression[StringJoin["periStimDFFsf", ToString[
    i], "tf", ToString[j], ToString[n]]])]}]]])];,
{n, 1, numROIs}, {i, {2, 4, 8, 16, 32}}, {j, {5,
10,
20,
40,
80,
150,
240}}];

```

```

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

```

```

In[ ]:= Table[
  Evaluate@ToExpression[StringJoin["zScoredRespAt", ToString[i], "sf", ToString[j], "tf",
    ToString[n]]] = (Mean[Part[#, 2] & /@ Mean[Table[N@Take[(ToExpression[StringJoin[
    "periStimZscoredDFFsf", ToString[i], "tf", ToString[j], ToString[n]]])][[
    k]], {tpFrameRate + 1, (tpFrameRate + 1) + (2 * tpFrameRate)}],
    {k, 1, Length[(ToExpression[StringJoin["periStimZscoredDFFsf",
    ToString[i], "tf", ToString[j], ToString[n]]])]}]]])];,
{n, 1, numROIs}, {i, {2, 4, 8, 16, 32}}, {j, {5, 10, 20, 40,
80,
150,
240}}];

```

```

In[ ]:= (**Visualize the TF-SF space of responses**)

```

```

In[ ]:= Manipulate[GraphicsGrid[{{ListLinePlot[
  Mean[ToExpression[StringJoin["periStimDFFblSubsf32tf5", ToString[ROI]]]],
  PlotRange -> {All, {ToExpression[StringJoin["minRespValSFTF", ToString[ROI]]],
    ToExpression[StringJoin["maxRespValSFTF", ToString[ROI]]}}}}, ListLinePlot[
  Mean[ToExpression[StringJoin["periStimDFFblSubsf32tf10", ToString[ROI]]]],
  PlotRange -> {All, {ToExpression[StringJoin["minRespValSFTF", ToString[ROI]]],
    ToExpression[StringJoin["maxRespValSFTF", ToString[ROI]]}}}}, ListLinePlot[
  Mean[ToExpression[StringJoin["periStimDFFblSubsf32tf20", ToString[ROI]]]],
  PlotRange -> {All, {ToExpression[StringJoin["minRespValSFTF", ToString[ROI]]],
    ToExpression[StringJoin["maxRespValSFTF", ToString[ROI]]}}}}, ListLinePlot[
  Mean[ToExpression[StringJoin["periStimDFFblSubsf32tf40", ToString[ROI]]]],

```

[illegible]


```

PlotRange → {All, {ToExpression[StringJoin["minRespValSFTF", ToString[ROI]]],
  ToExpression[StringJoin["maxRespValSFTF", ToString[ROI]]]}},
{ListLinePlot[Mean[ToExpression[StringJoin["periStimDFFblSubsf4tf5", ToString[ROI]]]],
  PlotRange → {All, {ToExpression[StringJoin["minRespValSFTF", ToString[ROI]]],
    ToExpression[StringJoin["maxRespValSFTF", ToString[ROI]]]}}, ListLinePlot[
  Mean[ToExpression[StringJoin["periStimDFFblSubsf4tf10", ToString[ROI]]]],
  PlotRange → {All, {ToExpression[StringJoin["minRespValSFTF", ToString[ROI]]],
    ToExpression[StringJoin["maxRespValSFTF", ToString[ROI]]]}}, ListLinePlot[
  Mean[ToExpression[StringJoin["periStimDFFblSubsf4tf20", ToString[ROI]]]],
  PlotRange → {All, {ToExpression[StringJoin["minRespValSFTF", ToString[ROI]]],
    ToExpression[StringJoin["maxRespValSFTF", ToString[ROI]]]}}, ListLinePlot[
  Mean[ToExpression[StringJoin["periStimDFFblSubsf4tf40", ToString[ROI]]]],
  PlotRange → {All, {ToExpression[StringJoin["minRespValSFTF", ToString[ROI]]],
    ToExpression[StringJoin["maxRespValSFTF", ToString[ROI]]]}}, ListLinePlot[
  Mean[ToExpression[StringJoin["periStimDFFblSubsf4tf80", ToString[ROI]]]],
  PlotRange → {All, {ToExpression[StringJoin["minRespValSFTF", ToString[ROI]]],
    ToExpression[StringJoin["maxRespValSFTF", ToString[ROI]]]}}, ListLinePlot[
  Mean[ToExpression[StringJoin["periStimDFFblSubsf4tf150", ToString[ROI]]]],
  PlotRange → {All, {ToExpression[StringJoin["minRespValSFTF", ToString[ROI]]],
    ToExpression[StringJoin["maxRespValSFTF", ToString[ROI]]]}}, ListLinePlot[
  Mean[ToExpression[StringJoin["periStimDFFblSubsf4tf240", ToString[ROI]]]],
  PlotRange → {All, {ToExpression[StringJoin["minRespValSFTF", ToString[ROI]]],
    ToExpression[StringJoin["maxRespValSFTF", ToString[ROI]]]}},
{ListLinePlot[Mean[ToExpression[StringJoin["periStimDFFblSubsf2tf5", ToString[ROI]]]],
  PlotRange → {All, {ToExpression[StringJoin["minRespValSFTF", ToString[ROI]]],
    ToExpression[StringJoin["maxRespValSFTF", ToString[ROI]]]}}, ListLinePlot[
  Mean[ToExpression[StringJoin["periStimDFFblSubsf2tf10", ToString[ROI]]]],
  PlotRange → {All, {ToExpression[StringJoin["minRespValSFTF", ToString[ROI]]],
    ToExpression[StringJoin["maxRespValSFTF", ToString[ROI]]]}}, ListLinePlot[
  Mean[ToExpression[StringJoin["periStimDFFblSubsf2tf20", ToString[ROI]]]],
  PlotRange → {All, {ToExpression[StringJoin["minRespValSFTF", ToString[ROI]]],
    ToExpression[StringJoin["maxRespValSFTF", ToString[ROI]]]}}, ListLinePlot[
  Mean[ToExpression[StringJoin["periStimDFFblSubsf2tf40", ToString[ROI]]]],
  PlotRange → {All, {ToExpression[StringJoin["minRespValSFTF", ToString[ROI]]],
    ToExpression[StringJoin["maxRespValSFTF", ToString[ROI]]]}}, ListLinePlot[
  Mean[ToExpression[StringJoin["periStimDFFblSubsf2tf80", ToString[ROI]]]],
  PlotRange → {All, {ToExpression[StringJoin["minRespValSFTF", ToString[ROI]]],
    ToExpression[StringJoin["maxRespValSFTF", ToString[ROI]]]}}, ListLinePlot[
  Mean[ToExpression[StringJoin["periStimDFFblSubsf2tf150", ToString[ROI]]]],
  PlotRange → {All, {ToExpression[StringJoin["minRespValSFTF", ToString[ROI]]],
    ToExpression[StringJoin["maxRespValSFTF", ToString[ROI]]]}}, ListLinePlot[
  Mean[ToExpression[StringJoin["periStimDFFblSubsf2tf240", ToString[ROI]]]],
  PlotRange → {All, {ToExpression[StringJoin["minRespValSFTF", ToString[ROI]]],
    ToExpression[StringJoin["maxRespValSFTF", ToString[ROI]]]}},
ImageSize → Full], {ROI, sigRespROIs}]

```

In[]:= (**Visualize the SF, TF, and speed tuning curves**)

```

In[ ]:= Manipulate[
  {ListLogLinearPlot[ToExpression[StringJoin["zScoredRespVsSFROI", ToString[n]]],
    Joined → True], ListLogLinearPlot[
    ToExpression[StringJoin["zScoredRespVsTFROI", ToString[n]]], Joined → True],
  ListLogLinearPlot[ToExpression[StringJoin["zScoredRespVsSpeedROI", ToString[n]]],
    Joined → True]}, {n, sigRespROIs}]

In[ ]:= (**Fit log-Gaussian tuning curves to the SF
  and TF tuning plots of all significantly responsive ROIs**)

In[ ]:= Table[Evaluate@ToExpression[StringJoin["fittedTuningModelSF", ToString[i]]] =
  Quiet[NonlinearModelFit[ToExpression[StringJoin["zScoredRespVsSFROI", ToString[i]]],
    {rMaxSF * Exp[-((Log[2, x] - Log[2, sfPref])^2) / (2 * sigmaSF^2)], rMaxSF > 0,
    sfPref > 0, sigmaSF > 0}, {rMaxSF, sfPref, sigmaSF}, x]}; {i, sigRespROIs}];

In[ ]:= Table[Evaluate@ToExpression[StringJoin["fittedTuningModelTF", ToString[i]]] =
  Quiet[NonlinearModelFit[ToExpression[StringJoin["zScoredRespVsTFROI", ToString[i]]],
    {rMaxTF * Exp[-((Log[2, x] - Log[2, tfPref])^2) / (2 * sigmaTF^2)], rMaxTF > 0,
    tfPref > 0, sigmaTF > 0}, {rMaxTF, tfPref, sigmaTF}, x]}; {i, sigRespROIs}];

In[ ]:= (**Compute the adjusted R-
  Square value for the model fit for each significantly responsive ROI**)

In[ ]:= Table[Evaluate@ToExpression[StringJoin["rSquaredModelFitSF", ToString[i]]] =
  Quiet[(ToExpression[StringJoin["fittedTuningModelSF", ToString[i]]])["
    AdjustedRSquared"]]; {i, sigRespROIs}];

In[ ]:= Table[Evaluate@ToExpression[StringJoin["rSquaredModelFitTF", ToString[i]]] =
  Quiet[(ToExpression[StringJoin["fittedTuningModelTF", ToString[i]]])["
    AdjustedRSquared"]]; {i, sigRespROIs}];

In[ ]:= (**Identify which ROIs have acceptable model fits to the tuning plots (R^2 > 0.8)**)

In[ ]:= wellFitROIs = DeleteCases[
  Table[If[ToExpression[StringJoin["rSquaredModelFitSF", ToString[n]]] > 0.8 &&
    ToExpression[StringJoin["rSquaredModelFitTF", ToString[n]]] > 0.8,
    n, Null], {n, sigRespROIs}], Null];

In[ ]:= (**For ROIs with well-fit SF and TF tuning curves,
  extract the preferred spatial frequency, preferred temporal frequency,
  spatial frequency tuning width, and temporal frequency tuning
  width. Also extract the skewness of the temporal frequency tuning plots**)

In[ ]:= Table[Evaluate@ToExpression[StringJoin["sfPref", ToString[i]]] =
  Values[(ToExpression[StringJoin["fittedTuningModelSF", ToString[i]]])["
    BestFitParameters"]][[2]]]; {i, wellFitROIs}];

In[ ]:= Table[Evaluate@ToExpression[StringJoin["tfPref", ToString[i]]] =
  Values[(ToExpression[StringJoin["fittedTuningModelTF", ToString[i]]])["
    BestFitParameters"]][[2]]]; {i, wellFitROIs}];

In[ ]:= Table[Evaluate@ToExpression[StringJoin["sigmaSF", ToString[i]]] =
  Values[(ToExpression[StringJoin["fittedTuningModelSF", ToString[i]]])["
    BestFitParameters"]][[3]]]; {i, wellFitROIs}];

```

```

In[ ]:= Table[Evaluate@ToExpression[StringJoin["sigmaTF", ToString[i]]] =
  Values[(ToExpression[StringJoin["fittedTuningModelTF", ToString[i]]])["BestFitParameters"]][[3]]]; {i, wellFitROIs});

In[ ]:= Table[Evaluate@ToExpression[StringJoin["skewTF", ToString[i]]] = Skewness[Part[#, 2] & /@
  ToExpression[StringJoin["zScoredRespVsTFROI", ToString[i]]]]]; {i, wellFitROIs});

In[ ]:= (***Fit a 2-dimensional Gaussian to SF-
  TF combination responses. See Priebe et al. 2006***)

In[ ]:= Table[Evaluate@ToExpression[StringJoin["sfTFResponseVals", ToString[n]]] =
  Flatten[Table[{i * 0.01, j * 0.1, ToExpression[
    StringJoin["zScoredRespAt", ToString[i], "sf", ToString[j], "tf", ToString[n]]}],
    {i, {2, 4, 8, 16, 32}}, {j, {5, 10, 20, 40, 80, 150, 240}}, 1]; {n, wellFitROIs});

In[ ]:= Table[sfPref = ToExpression[StringJoin["sfPref", ToString[i]]];
  tfPref = ToExpression[StringJoin["tfPref", ToString[i]]];
  sigmaSF = ToExpression[StringJoin["sigmaSF", ToString[i]]];
  sigmaTF = ToExpression[StringJoin["sigmaSF", ToString[i]]];
  skewTF = ToExpression[StringJoin["skewTF", ToString[i]]];
  Evaluate@ToExpression[StringJoin["fittedSFTFgaussian", ToString[i]]] =
  Quiet[NonlinearModelFit[ToExpression[StringJoin["sfTFResponseVals", ToString[i]]],
    {Rmax * Exp[-(Log[2, x] - Log[2, sfPref])^2 / (2 * sigmaSF^2)] * (Exp[-(Log[2, y] -
      Log[2, (Power[2, xi * (Log[2, x] - Log[2, sfPref]) + Log[2, tfPref])])^2] /
      (2 * (sigmaTF + skewTF * (Log[2, y] - Log[2, (Power[2, xi * (Log[2, x] - Log[2,
        2, sfPref]) + Log[2, tfPref])])^2)] - Exp[-1 / skewTF^2])),
    Rmax > 0, -1 ≤ xi ≤ 1}, {Rmax, xi}, {x, y}]]; {i, wellFitROIs});

In[ ]:= (***Extract the speed tuning index, xi, from the 2-D Gaussian fits***)

In[ ]:= Table[Evaluate@ToExpression[StringJoin["speedTuningIndex", ToString[i]]] =
  Values[(ToExpression[StringJoin["fittedSFTFgaussian", ToString[i]]])["BestFitParameters"]][[2]]]; {i, wellFitROIs});

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], "_", "sigResponsiveROIs", ".txt"], sigRespROIs];

In[ ]:= (***Export the list of the identities of the ROIs with well-
  fit SF and TF tuning curves***)

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

```

```

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 spatial frequency 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], "_", "SFTuning_ROI", ToString[n], ".txt"],
    ToExpression[StringJoin["zScoredRespVsSFROI", ToString[n]]]], {n, 1, numROIs}];

In[ ]:= (**Export the temporal frequency 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], "_", "TFTuning_ROI", ToString[n], ".txt"],
    ToExpression[StringJoin["zScoredRespVsTFROI", ToString[n]]]], {n, 1, numROIs}];

In[ ]:= (**Export the speed 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], "_", "speedTuning_ROI", ToString[n], ".txt"],
    ToExpression[StringJoin["zScoredRespVsSpeedROI", ToString[n]]]], {n, 1, numROIs}];

In[ ]:= (**Export the speed tuning data index each ROI with well-
    fit SF and TF tuning curves**)

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

```