In[5]:=
```
<< Eidomatica`
SetDirectory[NotebookDirectory[]];
$HistoryLength = 0;
```

```
Revisions: {Eidomatica Package: 347, libEidomatica: Unknown}
```

Import data

Directories

In[ ]:=
```
sample1 = "../results/sample-1/plots/";
sample2 = "./results/sample-2/plots/";
sample3 = "./results/sample-3/plots/";
directories = {sample1, sample2, sample3}
```

Out[ ]=
```
{../results/sample-1/plots/,
 ./results/sample-2/plots/, ./results/sample-3/plots/}
```

Cell Position Data

```
sample1 = Import["../test-data/sample-1/data.mx"];
sample2 = Import["../test-data/sample-2/data.mx"];
sample3 = Import["../test-data/sample-3/data.mx"];
data = {sample1, sample2, sample3};
```

Tracks

In[8]:=
```
sample1Tracks = Import["../test-data/sample-1/tracks.wdx"];
sample2Tracks = Import["../test-data/sample-2/tracks.wdx"];
sample3Tracks = Import["../test-data/sample-3/tracks.wdx"];
tracks = {sample1Tracks, sample2Tracks, sample3Tracks};
```

Densities

```
sample1Bonne = Import["../test-data/sample-1/densities250-bonne.mx"];
sample2Bonne = Import["../test-data/sample-2/densities250-bonne.mx"];
sample3Bonne = Import["../test-data/sample-3/densities250-bonne.mx"];
bonne = {sample1Bonne, sample2Bonne, sample3Bonn2};

sample1Densities = Import["../test-data/sample-1/densities250.mx"];
sample2Densities = Import["../test-data/sample-2/densities250.mx"];
sample3Densities = Import["../test-data/sample-3/densities250.mx"];
densities = {sample1Densities, sample2Densities, sample3Densities};
```

Colors

In[12]:=
```
colors = ColorData[109, #] & /@ {14, 13, 15}
```

Out[12]= {■, ■, ■}

Cell counts

Absolute cell counts

```
data // Dimensions
```

{3, 3, 420, 3}

```
data[[All, All, All, 1]] // Dimensions
```

{3, 3, 420}

```
counts = Map[Length, data[[All, All, All, 1]], {3}];
```

```
{counts, Total /@ counts} // Dimensions
```

{2, 3}

Relative cell counts

```
totals = Total /@ counts;
relatives = Table[# / totals[[i]] & /@ counts[[i]], {i, 1, Length@counts}];
```

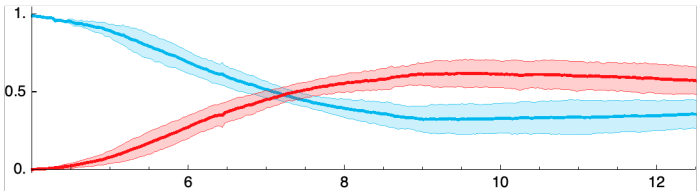
```
Dimensions@relatives
```

{3, 3, 420}

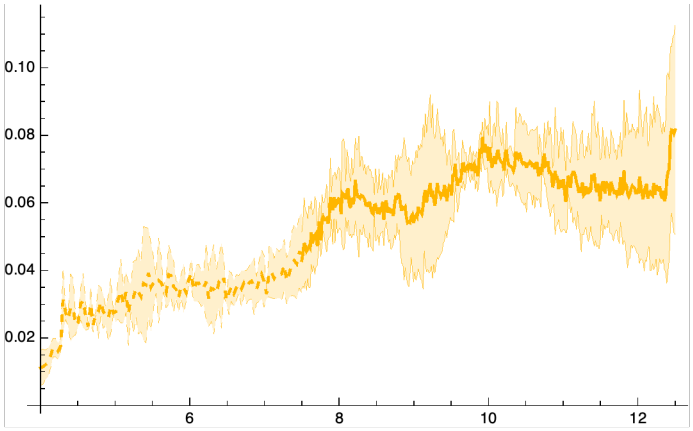
```
SetOptions[ConfidencePlot, BaseStyle -> {8, FontFamily → "Helvetica"}];
ticks = Table[If[EvenQ[i], {i, i, {.01, 0}}, {i, "", {.005, 0}}], {i, 5, 12}]
```

{{5, , {0.005, 0}}, {6, 6, {0.01, 0}}, {7, , {0.005, 0}}, {8, 8, {0.01, 0}},
 {9, , {0.005, 0}}, {10, 10, {0.01, 0}}, {11, , {0.005, 0}}, {12, 12, {0.01, 0}}}

```
first = Show[MapThread[ConfidencePlot[Transpose@#1, Color → #2,
      PlotRange → {{4., 12.5}, All}, DataRange → {4, 18}, AxesOrigin → {4., 0}] &,
    Most /@ {Transpose[relatives], colors}], AspectRatio → 1 / 4,
  Ticks → {Automatic, {0., .5, 1.}}]
```



```
last = Show[{
      ConfidencePlot[Transpose@(Take[#, 105] & /@ #[[1]]), Color → #[[2]],
       LineStyle → Dashed, DataRange → {4., 7.5}, AxesOrigin → {4., 0}],
      ConfidencePlot[Transpose@(Drop[#, 105] & /@ #[[1]]), Color → #[[2]],
       DataRange → {7.5, 12.5}, AxesOrigin → {4., 0}]}, PlotRange → All
   ] &@(Last /@ {Transpose[relatives], colors})
```




```
gr = Show[first, last];
```

```
Export["../tresults/relative_cell_counts+1,96*standard_error.pdf", gr]
```

```
ListLinePlot[Mean /@ Transpose@relatives, PlotStyle → colors,
 PlotRange → {{4., 12.5}, All}, DataRange → {4, 18}, AxesOrigin → {4., 0}]
```

```
MapThread[
 ListLinePlot[Transpose@#1, PlotStyle → #2, PlotRange → {{4., 12.5}, All},
   DataRange → {4, 18}, AxesOrigin → {4., 0}] &, {Transpose[relatives], colors}]
```

```
Export["../results/relative-cell_counts.pdf", Show[
  MapThread[ConfidencePlot[Transpose@#1, Color → #2, PlotRange → {All, All}] &,
   {Transpose[relatives], colors}]]]
```

Radii

```
radii = Map[Norm@Last@# &, data[[All, All, All, 1]], {4}];
```

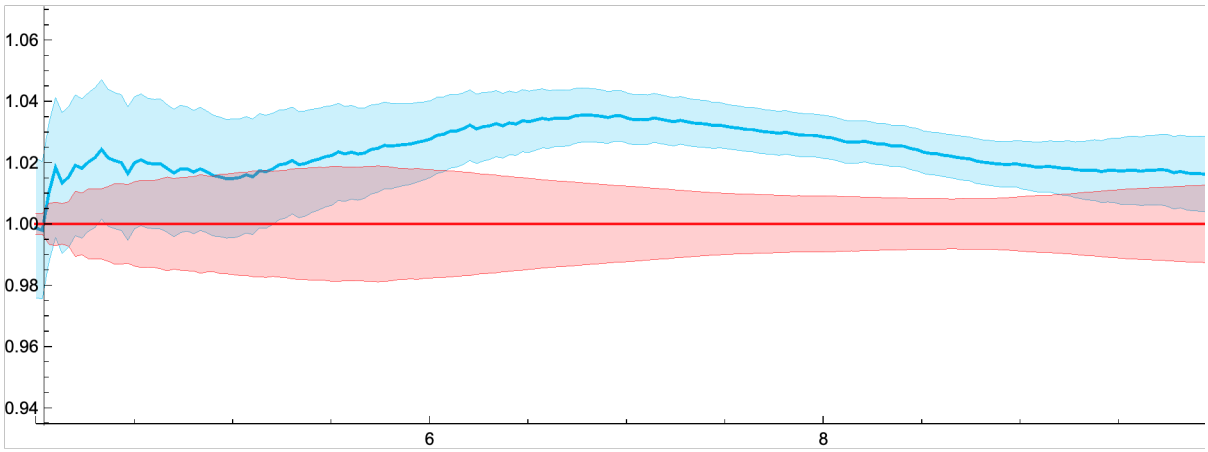
```
sample1Mean = Map[Mean, radii[[1]], {2}];
```

```
radii // Dimensions
```

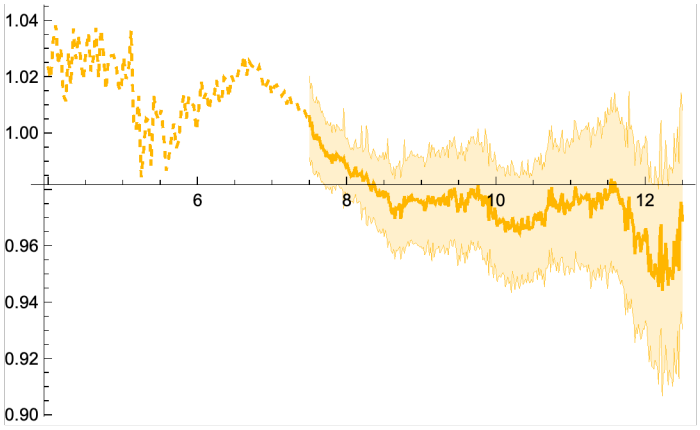
```
{3, 3, 420}
```

```
MapThread[StandardDeviationPlot[#1 / sample1Mean[[2]], Color → #2,
   DataRange → {4, 18}, PlotRange → {{4., 12.5}, All}] &, {radii[[1]], colors}]
```

```
first = Show[MapThread[StandardDeviationPlot[#1 / sample1Mean[[2]], Color → #2,
      DataRange → {4, 18}, PlotRange → {{4., 12.5}, All}, DeviationScaling → .3] &,
    Most /@ {radii[[1]], colors}], AspectRatio → 1 / 4]
```




```
last = Show[{
      StandardDeviationPlot[(Take[#[[1]] / sample1Mean[[2]], 105]), Color → #[[2]],
       LineStyle → Dashed, DataRange → {4., 7.5}, DeviationScaling → 0],
      StandardDeviationPlot[(Drop[#[[1]] / sample1Mean[[2]], 105]), Color →
        #[[2]], DataRange → {7.5, 12.5}, DeviationScaling → .3]}, PlotRange → All
   ] &@ (Last /@ {radii[[1]], colors})
```




```
Export["../results/radii+scaled-standard_deviation.pdf", Show[first, last]]
```

Population radii

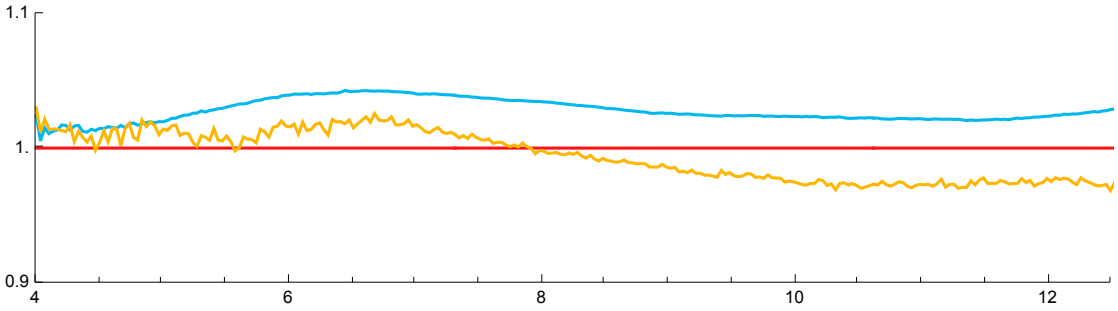
```
means = Map[Mean, radii, {3}];
```

```
means // Dimensions
```

```
{3, 3, 420}
```

```
normalization = Table[Mean@Flatten@means[[All, 2, i]], {i, 1, 420}];
```

```
gr = Show[MapThread[ListLinePlot[Mean@#1 / #3, PlotStyle → #2, ImageSize → Large,
       DataRange → {4, 18}, PlotRange → {{4., 12.5}, {.9, 1.1}}] &,
     {Transpose@means, colors, ConstantArray[normalization, 3]}],
   AxesOrigin → {4., .9}, AspectRatio → 1 / 4, Ticks → {Automatic, {.9, 1., 1.1}}]
```




```
Export["../radii_long2.pdf", gr]
```

cell tracks and straightness index

In[13]:=
```
(*select tracklets from time interval*)
Clear[selectTrack, selectTracks];
selectTracks[tracks : {__Rule}, start_Integer, end_Integer] :=
 DeleteCases[selectTrack[#, start, end] & /@ tracks, {}]
selectTrack[track_Rule, start_Integer, end_Integer] :=
 If[First@track < start, If[Length@Last@track < (start - First@track + 1), {},
   Take[Last@track, {start - First@track + 1, UpTo[end - First@track + 1]}]],
  Take[Last@track, UpTo[Max[0, end - First@track + 1]]]]
```

In[22]:=
```
<< ColorBrewer`
```

tracks

In[53]:=
```
tracklets = DeleteCases[DeleteCases[selectTracks[#, 50, 90], x_ /; Length@x < 10],
      x_ /; AnyTrue[Norm /@ Differences@x, # > 50 &]] & /@ sample1Tracks;
```

In[54]:=
```
end = Map[Norm@First@Differences@#[[{1, -1}]] / (Total[Norm /@ Differences@#]) &,
      #] & /@ tracklets;
```

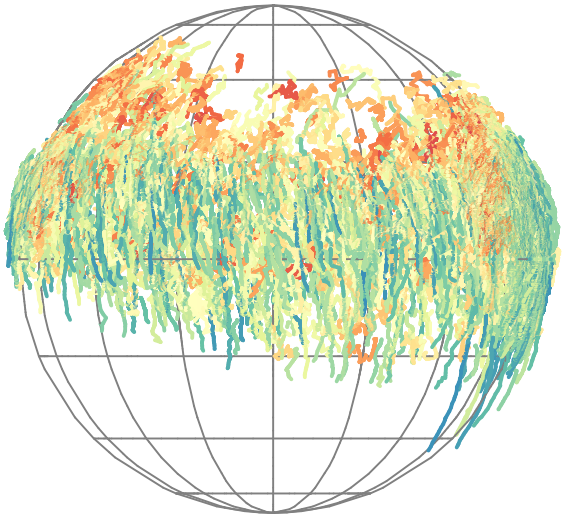
In[30]:=
```
Clear[testTrackVis]
testTrackVis[brewName_, n_: 9, inverted_: False] :=
 Module[{brewfun, mesPlot, ectPlot},
  brewfun = CreateColorFunction[brewName, n];
  mesPlot = Graphics3D[MapThread[{brewfun[If[inverted, 1 - #1, #1]], Line[#2]} &,
      {mesoLUT, mesotracks}], Boxed → False];
  ectPlot = Graphics3D[MapThread[{brewfun[If[inverted, 1 - #1, #1]], Line[#2]} &,
      {ectoLUT, ectotracks}], Boxed → False];
  GraphicsGrid[{{Show[mesPlot, ViewPoint → Front],
      Show[mesPlot, ViewPoint → Top]},
     {Show[ectPlot, ViewPoint → Front], Show[ectPlot, ViewPoint → Top]}}]
 ]
```

In[32]:=
```
sphere = SphericalPlot3D[245, {theta, 0, π}, {phi, 0, π}, PlotStyle → None,
  Mesh → 7, MeshStyle → Directive[AbsoluteThickness[1], Gray]]
```

In[33]:=
```
brewfun = CreateColorFunction[Spectral, 9];
```

In[68]:=
```
mesPlotFront =
 Show[Graphics3D[MapThread[{AbsoluteThickness[2], brewfun[#1], Line[#2]} &,
    {end[[2]], tracklets[[2]]}], Boxed → False, ViewPoint → {0, -Infinity, 0},
  ImageSize → 300], sphere, PlotRange → {All, {0, 300}, All}]
```

Out[68]=

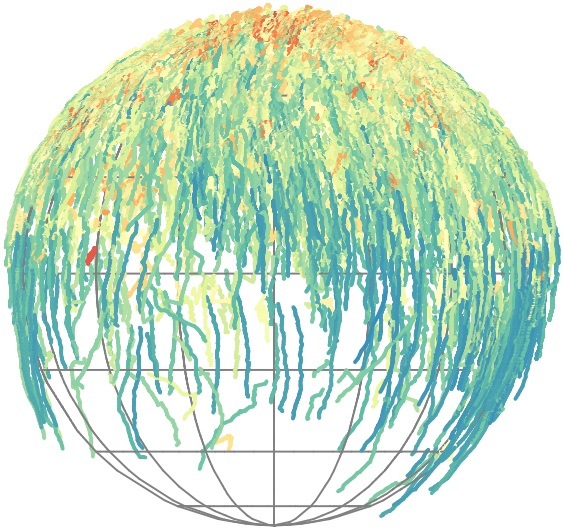


```
Export["../results/mesoPlotFront.pdf", mesPlotFront]
```

In[71]:=
```
mesPlotTop =
 Show[Graphics3D[MapThread[{AbsoluteThickness[2], brewfun[#1], Line[#2]} &,
    {end[[2]], tracklets[[2]]}], Boxed → False, ViewPoint → {0, 0, Infinity},
  ImageSize → 300], sphere, PlotRange → {All, {0, 300}, All}]
```

In[70]:=
```
ectPlotFront =
 Show[Graphics3D[MapThread[{AbsoluteThickness[2], brewfun[#1], Line[#2]} &,
    {end[[1]], tracklets[[1]]}], Boxed → False, ViewPoint → {0, -Infinity, 0},
  ImageSize → 300], sphere, PlotRange → {All, {0, 300}, All}]
```

Out[70]=



```
Export["../results/ectoPlotFront.pdf", ectPlotFront]
```

In[72]:=
```
ectPlotTop =
 Show[Graphics3D[MapThread[{AbsoluteThickness[2], brewfun[#1], Line[#2]} &,
    {end[[1]], tracklets[[1]]}], Boxed → False, ViewPoint → {0, 0, Infinity},
   ImageSize → 300], sphere, PlotRange → {All, {0, 300}, All}]
```

```
Graphics[Table[{CreateColorFunction[Spectral, 11][(i - 1) / 9],
   Rectangle[{i - 1, 0}, {i, 1}]}, {i, 1, 9}]]
```

In[39]:=
```
rotationMatrix = RotationMatrix[Pi / 2.4 + .1, {1, 0, 0}].
   RotationMatrix[-Pi / 3.2 - .1, {0, 1, 0}].RotationMatrix[-.2, {0, 0, 1}];
```

In[40]:=
```
sample1Tracks = Map[#[[1]] → (#.rotationMatrix & /@ #[[2]]) &, sample1Tracks, {2}];
```

Correlation analysis

In[43]:=
```
colors = ColorData[109, #] & /@ {14, 13, 15}
```

Out[43]= { , , }

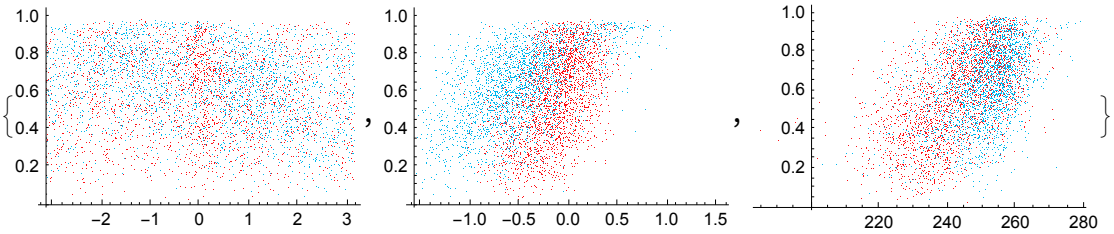
In[44]:=
```
spherical = Transpose /@ Map[ToSpherical[Last@#] &, tracklets, {2}];
```

In[45]:=
```
ecto =
 (ListPlot[Transpose[{#, end[[1]]}], PlotStyle → colors[[1]]] & /@ spherical[[1]])
```

In[46]:=
```
meso =
 (ListPlot[Transpose[{#, end[[2]]}], PlotStyle → colors[[2]]] & /@ spherical[[2]])
```

In[47]:=
```
gr = MapThread[Show[#1, #2, PlotRange → #3, AxesOrigin → #4] &,
  {ecto, meso, {{{-Pi, Pi}, Automatic}, {{-Pi / 2, Pi / 2}, Automatic},
    {Automatic, Automatic}}, {{-Pi, 0}, {-Pi / 2, 0}, Automatic}}]
```

Out[47]=



In[48]:=
```
MapThread[Export["../results/" <> #1 <> "-correlation.pdf", #2] &,
 {{"latitude", "longitude", "radius"}, gr}]
```

Out[48]= {../results/latitude-correlation.pdf,
../results/longitude-correlation.pdf, ../results/radius-correlation.pdf}

```
SpearmanRho[Sequence @@ {#, end[[1]]}] & /@ spherical[[1]]
```

{-0.270388, 0.52594, 0.406122}

```
SpearmanRho[Sequence @@ {#, end[[2]]}] & /@ spherical[[2]]
```

{-0.0456064, 0.458189, 0.577349}