In [1]:
```
import pandas as pd
from pandas.api.types import CategoricalDtype
import numpy as np
import seaborn as sns
```

Import data

We start my importing data to a pandas dataframe (a data structure similar to a spreadsheet)

In [2]:
```
# path to csv file containing fibril data that was exported from CellProfiler
fibril_data_path = '../cellprofiler-output/data/Batch2_Fibrils.csv'

# import the fibril data into pandas
fibril_df = pd.read_csv(fibril_data_path)

# path to csv file containing fibril data that was exported from CellProfiler
fibril_skeleton_data_path = '../cellprofiler-output/data/Batch2_FibrilCenters.csv'

# import the fibril data into pandas
fibril_skeleton_df = pd.read_csv(fibril_skeleton_data_path)

# merge the fibril skeleton data into the fibril_df
fibril_df = fibril_df.merge(fibril_skeleton_df[['ImageNumber', 'ObjectNumber', 'ObjectSkeleton_NumberBranchEnds_FibrilsSkele

# show the data frame
fibril_df.head()
```

Out[2]:

	ImageNumber	ObjectNumber	Metadata_FileLocation	Metadata_Frame	Metadata_Series	Metadata_channel	Metadata_image	M
0	1	1	file:/Users/pryder/GitHub/pearlryder_projects/...	0	0	GFP	1	
1	1	2	file:/Users/pryder/GitHub/pearlryder_projects/...	0	0	GFP	1	
2	1	3	file:/Users/pryder/GitHub/pearlryder_projects/...	0	0	GFP	1	
3	1	4	file:/Users/pryder/GitHub/pearlryder_projects/...	0	0	GFP	1	
4	1	5	file:/Users/pryder/GitHub/pearlryder_projects/...	0	0	GFP	1	

5 rows × 65 columns

In [3]:
```
# print out each column name; gives a sense of what data is in the dataframe

for column in fibril_df.columns:
    print(column)
```

```
ImageNumber
ObjectNumber
Metadata_FileLocation
Metadata_Frame
Metadata_Series
Metadata_channel
Metadata_image
Metadata_substrate
Metadata_time
Metadata_timepoint
FileName_GFP
PathName_GFP
AreaShape_Area
AreaShape_BoundingBoxArea
AreaShape_BoundingBoxMaximum_X
AreaShape_BoundingBoxMaximum_Y
AreaShape_BoundingBoxMinimum_X
AreaShape_BoundingBoxMinimum_Y
AreaShape_Center_X
AreaShape_Center_Y
AreaShape_Compactness
AreaShape_Eccentricity
AreaShape_EquivalentDiameter
AreaShape_EulerNumber
AreaShape_Extent
AreaShape_FormFactor
AreaShape_MajorAxisLength
AreaShape_MaxFeretDiameter
AreaShape_MaximumRadius
AreaShape_MeanRadius
AreaShape_MedianRadius
AreaShape_MinFeretDiameter
AreaShape_MinorAxisLength
AreaShape_Orientation
AreaShape_Perimeter
AreaShape_Solidity
Intensity_IntegratedIntensityEdge_GFP
Intensity_IntegratedIntensity_GFP
```

```
Intensity_LowerQuartileIntensity_GFP
Intensity_MADIntensity_GFP
Intensity_MassDisplacement_GFP
Intensity_MaxIntensityEdge_GFP
Intensity_MaxIntensity_GFP
Intensity_MeanIntensityEdge_GFP
Intensity_MeanIntensity_GFP
Intensity_MedianIntensity_GFP
Intensity_MinIntensityEdge_GFP
Intensity_MinIntensity_GFP
Intensity_StdIntensityEdge_GFP
Intensity_StdIntensity_GFP
Intensity_UpperQuartileIntensity_GFP
Location_CenterMassIntensity_X_GFP
Location_CenterMassIntensity_Y_GFP
Location_CenterMassIntensity_Z_GFP
Location_Center_X
Location_Center_Y
Location_Center_Z
Location_MaxIntensity_X_GFP
Location_MaxIntensity_Y_GFP
Location_MaxIntensity_Z_GFP
Number_Object_Number
ObjectSkeleton_NumberBranchEnds_FibrilsSkeleton
ObjectSkeleton_NumberNonTrunkBranches_FibrilsSkeleton
ObjectSkeleton_NumberTrunks_FibrilsSkeleton
ObjectSkeleton_TotalObjectSkeletonLength_FibrilsSkeleton
```

In [4]:

```
# The data are grouped into biological classes by substrate and timepoint

# Some of the substrate names include '_A' and '_B' but not others
# Here, we'll replace these names with an empty string so that everything matches
find_replace_dict = {'HP05_A': 'HP05', 'HP05_B': 'HP05'}

fibril_df.replace(to_replace=find_replace_dict, inplace=True)

# now create one column that contains both variables

fibril_df['substrate_time']  = fibril_df['Metadata_substrate'] + '_' + fibril_df['Metadata_timepoint']


# custom sort to order variables to our desired order
# from https://towardsdatascience.com/how-to-do-a-custom-sort-on-pandas-dataframe-ac18e7ea5320

substrate_ordered = ['NP_6hr', 'NP_12hr', 'NP_24hr', 'HP05_6hr', 'HP05_12hr', 'HP05_24hr', 'HP3_6hr', 'HP3_12hr', 'HP3_24hr'
timepoint_ordered = ['NP_6hr', 'HP3_6hr', 'HP05_6hr', 'NP_12hr', 'HP3_12hr', 'HP05_12hr', 'NP_24hr', 'HP3_24hr', 'HP05_24hr'

condition_sort_order = CategoricalDtype(timepoint_ordered,
    ordered=True)


fibril_df['substrate_time'] = fibril_df['substrate_time'].astype(condition_sort_order)
fibril_df.sort_values('substrate_time', inplace=True)


# convert length metrics from pixels to microns
microns_per_pixel = 0.28112
fibril_df['ObjectSkeleton_TotalObjectSkeletonLength_FibrilsSkeleton_Microns'] = fibril_df['ObjectSkeleton_TotalObjectSkeleto

fibril_df['AreaShape_MajorAxisLength_Microns'] = fibril_df['AreaShape_MajorAxisLength'] * microns_per_pixel

fibril_df.head()
```

Out[4]:

	ImageNumber	ObjectNumber	Metadata_FileLocation	Metadata_Frame	Metadata_Series	Metadata_channel	Metadata_imag
86023	203	305	file:/Users/pryder/GitHub/pearlryder_projects/...	0	0	GFP	1
82838	191	449	file:/Users/pryder/GitHub/pearlryder_projects/...	0	0	GFP	
82839	191	450	file:/Users/pryder/GitHub/pearlryder_projects/...	0	0	GFP	
82840	191	451	file:/Users/pryder/GitHub/pearlryder_projects/...	0	0	GFP	
82841	191	452	file:/Users/pryder/GitHub/pearlryder_projects/...	0	0	GFP	

5 rows × 68 columns

Data Exploration: does morphology of fibrils change with different biological conditions?

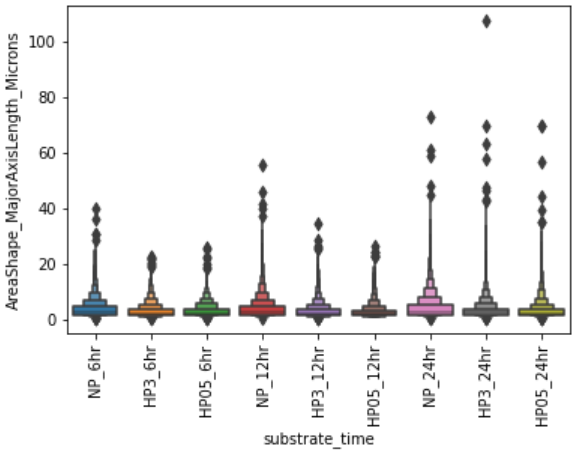
We'll start by exploring length relative to biological condition. This length measurement is approximated by fitting an ellipse to each object and getting the major axis length of that ellipse (see the CellProfiler 4.1.3 manual: MeasureObjectSizeShape)

Do fibrils change their length under different conditions?

Length:

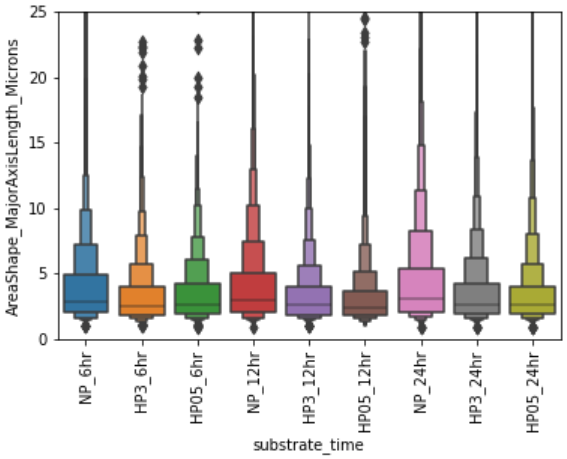
In [5]:
```
# enhanced boxplot for visualization of distribution
# see https://seaborn.pydata.org/generated/seaborn.boxenplot.html for details

ax = sns.boxenplot(x="substrate_time",
                   y="AreaShape_MajorAxisLength_Microns",
                   data=fibril_df)

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[5]:
```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
 Text(7, 0, 'HP3_24hr'),
 Text(8, 0, 'HP05_24hr')]
```



In [6]:
```
# set the y axis to better visualize the distributions (outliers make it difficult to assess)

ax = sns.boxenplot(x="substrate_time",
                   y="AreaShape_MajorAxisLength_Microns",
                   data=fibril_df)

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)

ax.set(ylim=(0,25))
```

Out[6]:
```
[(0.0, 25.0)]
```



In [7]:
```
# the underlying numbers:

fibril_df.groupby('substrate_time')['AreaShape_MajorAxisLength_Microns'].describe()
```

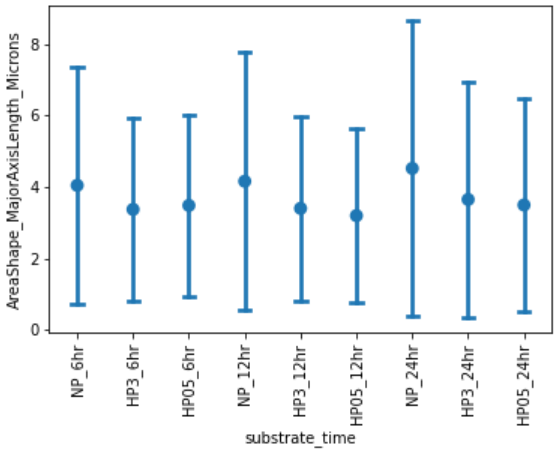
Out[7]:

substrate_time	count	mean	std	min	25%	50%	75%	max
NP_6hr	4771.0	4.043627	3.325334	0.973828	1.967866	2.854909	4.902876	40.136891
HP3_6hr	3971.0	3.370328	2.554939	0.973828	1.810284	2.471544	3.961825	22.699512
HP05_6hr	3089.0	3.480551	2.538824	0.918134	1.872351	2.592562	4.168968	26.172993
NP_12hr	9719.0	4.157844	3.624519	0.918134	1.973155	2.929228	4.962573	55.879673

substrate_time	count	mean	std	min	25%	50%	75%	max
HP3_12hr	8337.0	3.398282	2.583580	0.973828	1.825857	2.564853	4.009675	34.712998
HP05_12hr	5781.0	3.195353	2.424044	0.973828	1.789686	2.391905	3.652299	26.703772
NP_24hr	10165.0	4.519138	4.145009	0.918134	2.042717	3.083670	5.393957	73.352483
HP3_24hr	15259.0	3.647752	3.291037	0.918134	1.885550	2.639762	4.230058	107.793402
HP05_24hr	24932.0	3.493545	2.999373	0.918134	1.844873	2.538539	3.959891	69.941169

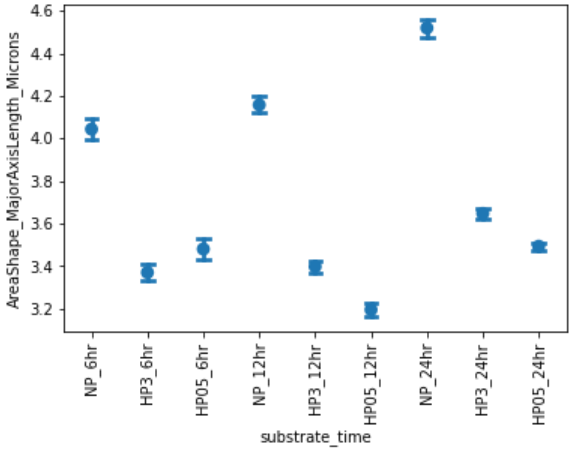
In [8]:
```
# major axis length, plot mean +/- sd

ax = sns.pointplot(x="substrate_time",
                   y="AreaShape_MajorAxisLength_Microns",
                   data=fibril_df,
                   join=False,
                   capsize=0.2,
                   ci="sd")

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[8]:
```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
 Text(7, 0, 'HP3_24hr'),
 Text(8, 0, 'HP05_24hr')]
```



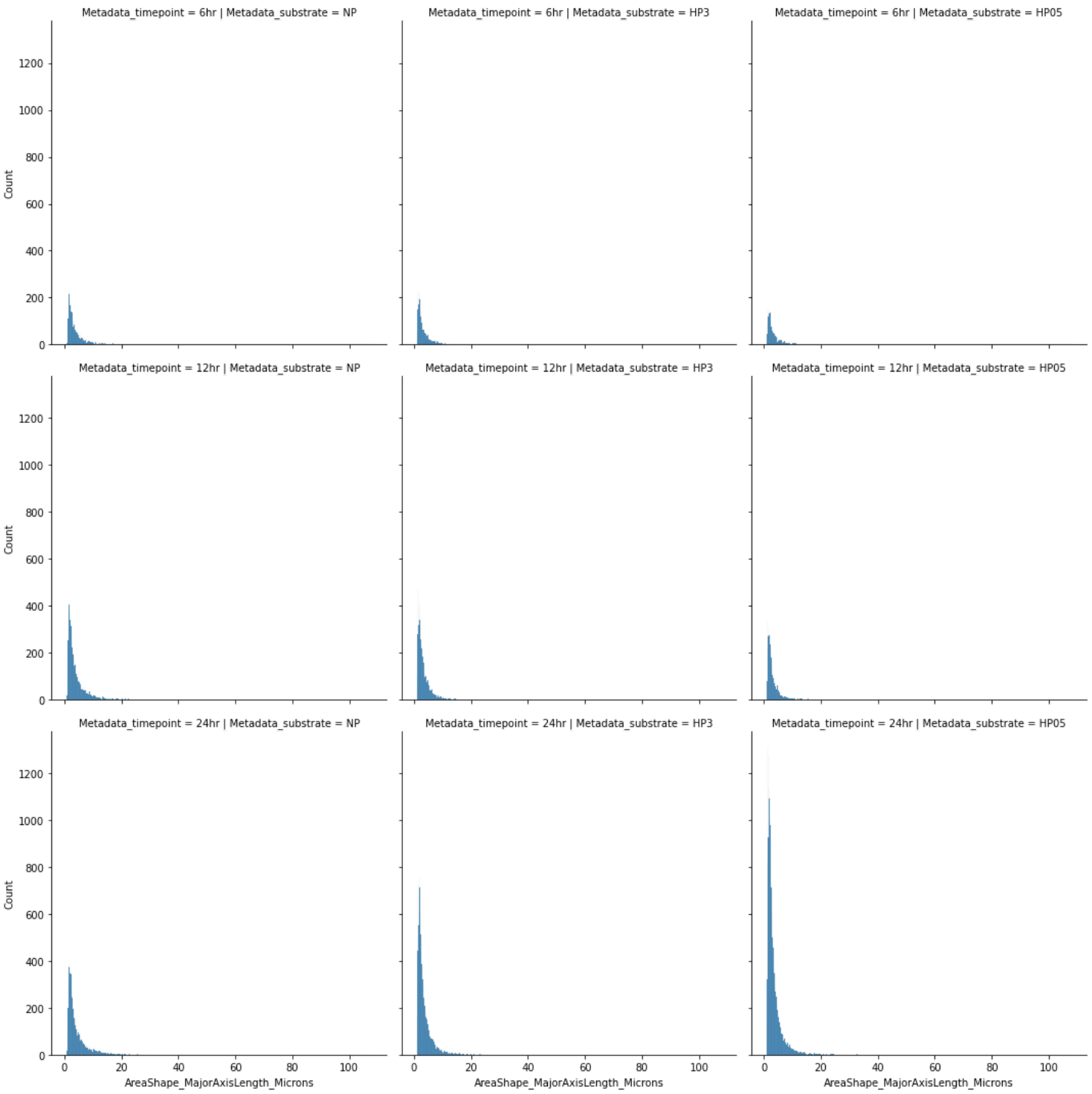
In [9]:
```
# major axis length, plot mean +/- sem

ax = sns.pointplot(x="substrate_time",
                   y="AreaShape_MajorAxisLength_Microns",
                   data=fibril_df,
                   join=False,
                   capsize=0.2,
                   ci=68)

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[9]:
```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
 Text(7, 0, 'HP3_24hr'),
 Text(8, 0, 'HP05_24hr')]
```



In [10]:

```
# Let's investigate the distribution of the length according to biological condition
# Here we can very nicely see the increase in length w/ increased time of incubation

ax = sns.displot(data=fibril_df,
                 x="AreaShape_MajorAxisLength_Microns",
                 row="Metadata_timepoint",
                 col="Metadata_substrate",
                 row_order=['6hr', '12hr', '24hr'])
```



Length conclusions:

This length metric does modestly increase with time, especially on the non porous substrate. The error bars are quite large. Note that the length is an approximate length based on modeling the object as an ellipse; curved fibrils will not be accurately measured.

Area:

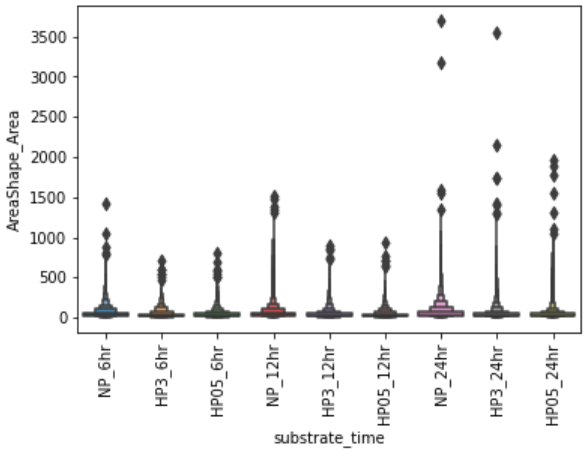
In [11]:

```
# enhanced boxplot for visualization of distribution
# see https://seaborn.pydata.org/generated/seaborn.boxenplot.html for details

ax = sns.boxenplot(x="substrate_time",
                   y="AreaShape_Area",
                   data=fibril_df)

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[11]:

```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
```

```
Text(7, 0, 'HP3_24hr'),
Text(8, 0, 'HP05_24hr')]
```

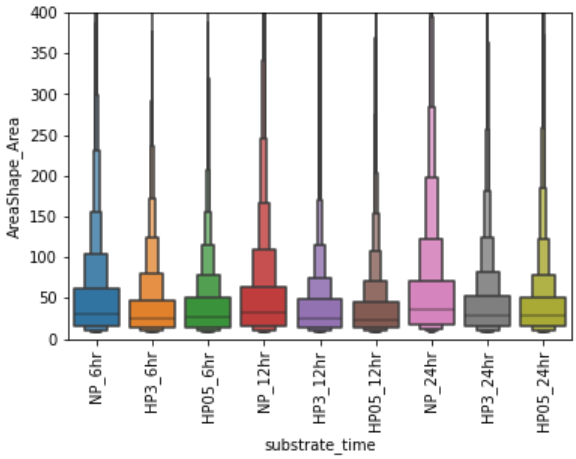


In [12]:

```
# set the y limit to better visualize distributions
# can appreciate how the non porous substrate distributions tend toward larger objects

ax = sns.boxenplot(x="substrate_time",
                   y="AreaShape_Area",
                   data=fibril_df)

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)


ax.set(ylim=(0,400))
```

Out[12]:

```
[(0.0, 400.0)]
```



In [13]:

```
fibril_df.groupby('substrate_time')['AreaShape_Area'].describe()
```

Out[13]:

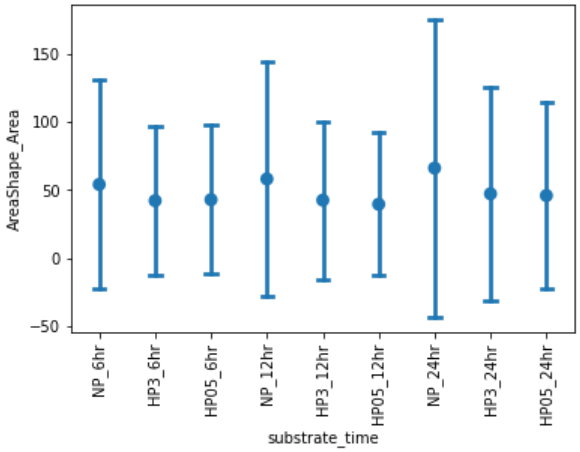
substrate_time	count	mean	std	min	25%	50%	75%	max
NP_6hr	4771.0	54.143576	76.361112	8.0	15.0	31.0	61.0	1425.0
HP3_6hr	3971.0	42.221858	54.513898	8.0	13.0	24.0	47.0	703.0
HP05_6hr	3089.0	42.973454	54.431512	8.0	14.0	26.0	50.0	812.0
NP_12hr	9719.0	58.153205	86.209724	8.0	16.5	32.0	64.0	1521.0
HP3_12hr	8337.0	42.625645	57.884949	8.0	14.0	25.0	48.0	897.0
HP05_12hr	5781.0	39.640028	52.371213	8.0	13.0	23.0	45.0	936.0
NP_24hr	10165.0	66.117167	108.822494	8.0	18.0	35.0	70.0	3701.0
HP3_24hr	15259.0	47.281604	77.837645	8.0	15.0	28.0	52.0	3547.0
HP05_24hr	24932.0	45.980868	68.466813	8.0	15.0	28.0	50.0	1969.0

In [14]:

```
# area, plot mean +/- sd

ax = sns.pointplot(x="substrate_time",
                   y="AreaShape_Area",
                   data=fibril_df,
                   join=False,
```

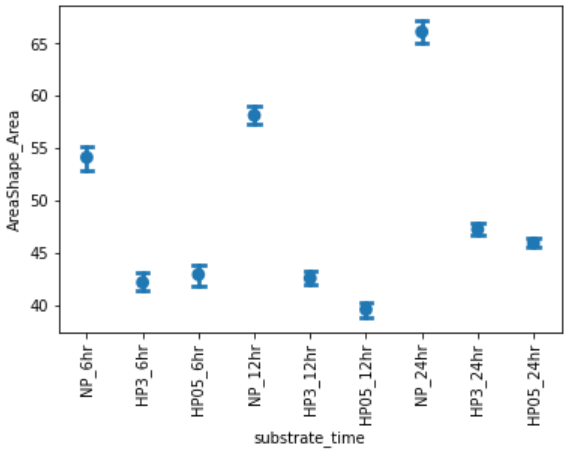
```
                capsize=0.2,
                ci="sd")

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[14]:
```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
 Text(7, 0, 'HP3_24hr'),
 Text(8, 0, 'HP05_24hr')]
```



In [15]:
```
# area, plot mean +/- sem

ax = sns.pointplot(x="substrate_time",
                y="AreaShape_Area",
                data=fibril_df,
                join=False,
                capsize=0.2,
                ci=68)

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[15]:
```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
 Text(7, 0, 'HP3_24hr'),
 Text(8, 0, 'HP05_24hr')]
```



Area conclusions:

Area seems to increase over time and to be highest in cells grown on the non porous substrates.

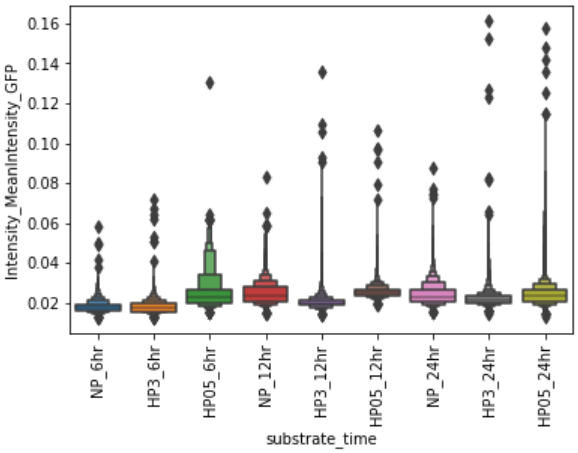
Intengrated intensity

I noted that fibrils appear to increase in intensity at later timepoints. Let's investigate if that's true across our population. I'll test the mean intensity, since that is essentially normalized for the changes in area that we've observed

In [16]:

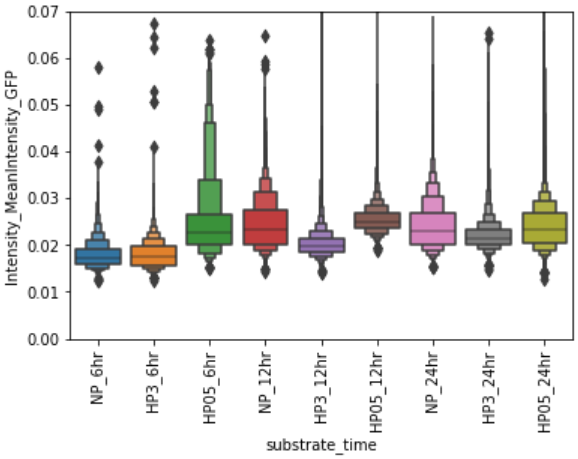
```
ax = sns.boxenplot(x="substrate_time",
                   y="Intensity_MeanIntensity_GFP",
                   data=fibril_df)

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[16]:
```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
 Text(7, 0, 'HP3_24hr'),
 Text(8, 0, 'HP05_24hr')]
```



In [17]:
```
# zooming in on the y axis:

ax = sns.boxenplot(x="substrate_time",
                   y="Intensity_MeanIntensity_GFP",
                   data=fibril_df)

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)

ax.set(ylim=(0,.07))
```

Out[17]:
```
[(0.0, 0.07)]
```



In [18]:
```
fibril_df.groupby('substrate_time')['Intensity_MeanIntensity_GFP'].describe()
```

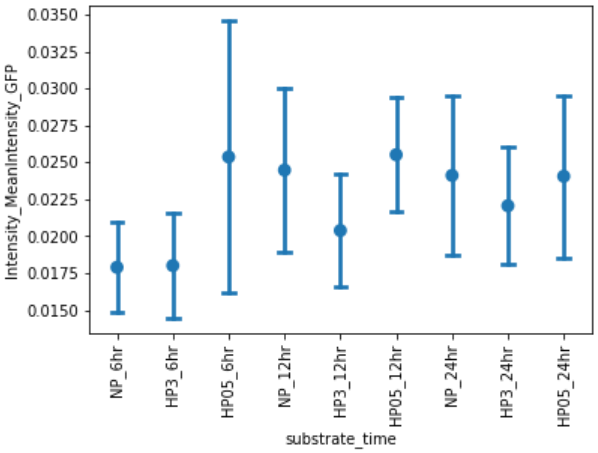
Out[18]:

substrate_time	count	mean	std	min	25%	50%	75%	max
NP_6hr	4771.0	0.017894	0.003058	0.012437	0.015816	0.017221	0.019213	0.058137
HP3_6hr	3971.0	0.018009	0.003522	0.012507	0.015526	0.017403	0.019774	0.071565
HP05_6hr	3089.0	0.025361	0.009231	0.015115	0.020043	0.022569	0.026658	0.130204
NP_12hr	9719.0	0.024470	0.005577	0.014411	0.020216	0.023189	0.027689	0.083101
HP3_12hr	8337.0	0.020383	0.003820	0.013911	0.018611	0.019898	0.021528	0.135905
HP05_12hr	5781.0	0.025509	0.003879	0.018683	0.023516	0.025065	0.026819	0.106411
NP_24hr	10165.0	0.024110	0.005428	0.015218	0.020221	0.022996	0.026769	0.087618

substrate_time	count	mean	std	min	25%	50%	75%	max
HP3_24hr	15259.0	0.022057	0.003994	0.014593	0.020008	0.021537	0.023274	0.161492
HP05_24hr	24932.0	0.024051	0.005486	0.012689	0.020386	0.023314	0.026833	0.158061

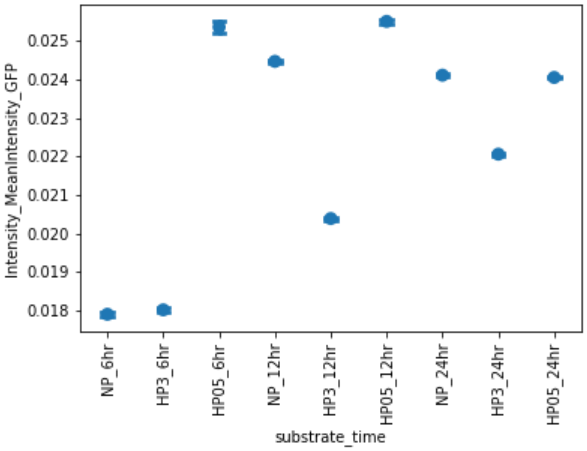
In [19]:
```
# intensity, plot mean +/- sd

ax = sns.pointplot(x="substrate_time",
                   y="Intensity_MeanIntensity_GFP",
                   data=fibril_df,
                   join=False,
                   capsize=0.2,
                   ci="sd")

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[19]:
```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
 Text(7, 0, 'HP3_24hr'),
 Text(8, 0, 'HP05_24hr')]
```



In [20]:
```
# intensity, plot mean +/- sem

ax = sns.pointplot(x="substrate_time",
                   y="Intensity_MeanIntensity_GFP",
                   data=fibril_df,
                   join=False,
                   capsize=0.2,
                   ci=68)

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[20]:
```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
 Text(7, 0, 'HP3_24hr'),
 Text(8, 0, 'HP05_24hr')]
```



Intensity conclusions:

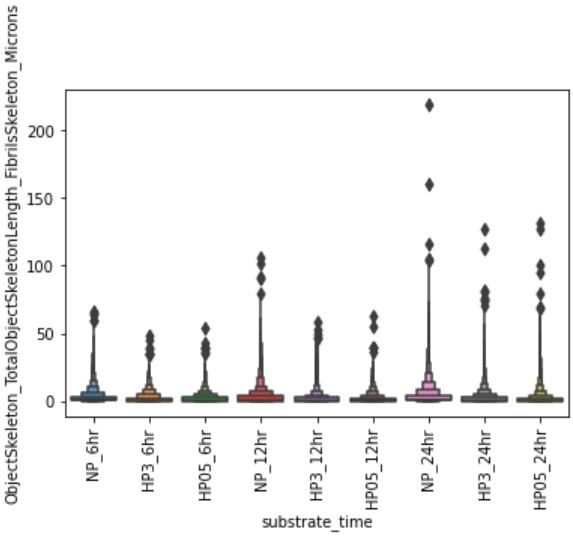
We do see an increase in mean intensity at the later timepoints, especially for the non-porous substrate.

Skeletonized length:

This metric thins the objects into a single-pixel wide skeleton. The length of this skeleton is then measured.

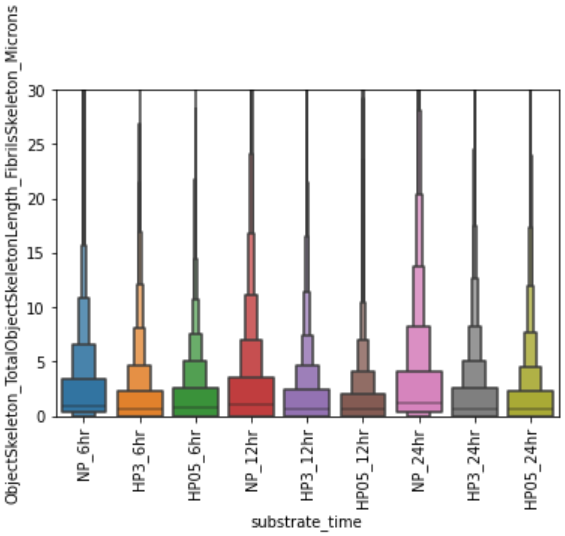
In [21]:
```
# skeletonized length enhanced box plot

ax = sns.boxenplot(x="substrate_time",
                   y="ObjectSkeleton_TotalObjectSkeletonLength_FibrilsSkeleton_Microns",
                   data=fibril_df)

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[21]:
```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
 Text(7, 0, 'HP3_24hr'),
 Text(8, 0, 'HP05_24hr')]
```



In [22]:
```
# zoom in on y to better see distributions
# the presence of outliers obscures the underlying trends:

ax = sns.boxenplot(x="substrate_time",
                   y="ObjectSkeleton_TotalObjectSkeletonLength_FibrilsSkeleton_Microns",
                   data=fibril_df)

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)

ax.set(ylim=(0,30))
```

Out[22]:
```
[(0.0, 30.0)]
```

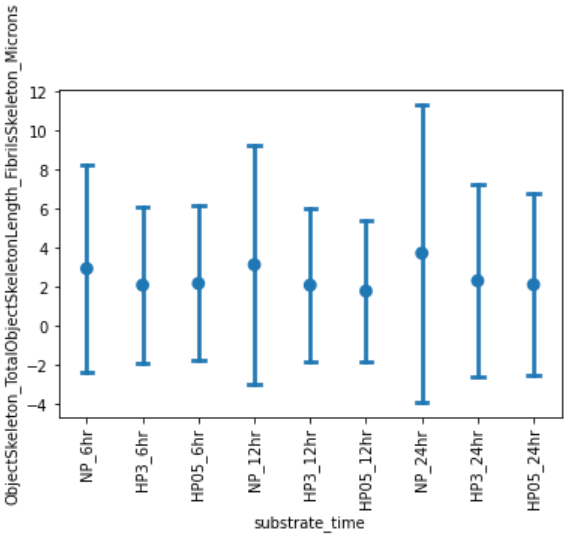


In [23]:

```
# skeletonized length, plot mean +/- sd

ax = sns.pointplot(x="substrate_time",
                   y="ObjectSkeleton_TotalObjectSkeletonLength_FibrilsSkeleton_Microns",
                   data=fibril_df,
                   join=False,
                   capsize=0.2,
                   ci="sd")

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[23]:

```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
 Text(7, 0, 'HP3_24hr'),
 Text(8, 0, 'HP05_24hr')]
```



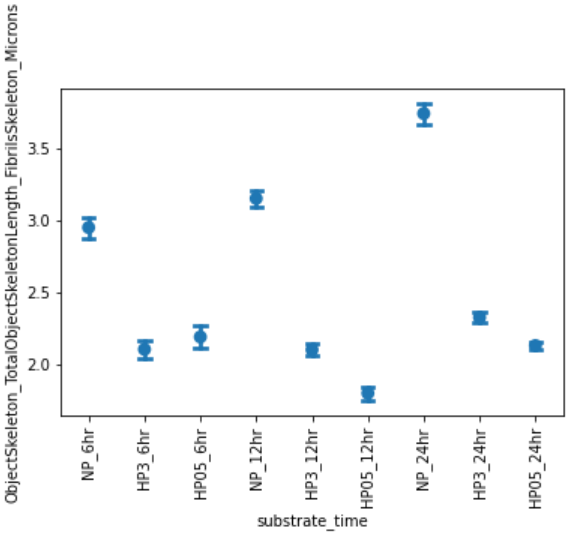
In [24]:

```
# skeletonized length, plot mean +/- sem

ax = sns.pointplot(x="substrate_time",
                   y="ObjectSkeleton_TotalObjectSkeletonLength_FibrilsSkeleton_Microns",
                   data=fibril_df,
                   join=False,
                   capsize=0.2,
                   ci=68)

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[24]:

```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
```

```
Text(7, 0, 'HP3_24hr'),
Text(8, 0, 'HP05_24hr')]
```



In [25]:

```
# underlying numbers for the fiber lengths. Note that 25% of objects are 0.0 pixels long
# An indication that this isn't the best approximation for fiber length for small round objects
# We can add a filter to eliminate these from analysis

fibril_df.groupby('substrate_time')['ObjectSkeleton_TotalObjectSkeletonLength_FibrilsSkeleton_Microns'].describe()
```

Out[25]:

substrate_time	count	mean	std	min	25%	50%	75%	max
NP_6hr	4771.0	2.946765	5.344688	0.0	0.28112	0.959804	3.441651	65.993570
HP3_6hr	3971.0	2.101716	3.993250	0.0	0.00000	0.562240	2.317171	48.475940
HP05_6hr	3089.0	2.187650	3.963057	0.0	0.00000	0.795127	2.598291	54.186530
NP_12hr	9719.0	3.148282	6.139756	0.0	0.00000	1.076247	3.558095	106.124099
HP3_12hr	8337.0	2.099038	3.937768	0.0	0.00000	0.678684	2.433615	58.014042
HP05_12hr	5781.0	1.797214	3.585985	0.0	0.00000	0.562240	2.036051	62.543643
NP_24hr	10165.0	3.737444	7.608462	0.0	0.28112	1.240924	4.072102	219.284475
HP3_24hr	15259.0	2.323006	4.904448	0.0	0.00000	0.678684	2.598291	127.560859
HP05_24hr	24932.0	2.127791	4.640348	0.0	0.00000	0.562240	2.317171	131.862421

In [26]:

```
# we add a size filtering to our graphing; here we filter for > 0 microns

fibril_df['ObjectSkeleton_TotalObjectSkeletonLength_FibrilsSkeleton_Microns_Filtered'] = fibril_df['ObjectSkeleton_TotalObje

fibril_df.groupby('substrate_time')['ObjectSkeleton_TotalObjectSkeletonLength_FibrilsSkeleton_Microns_Filtered'].describe()
```

Out[26]:

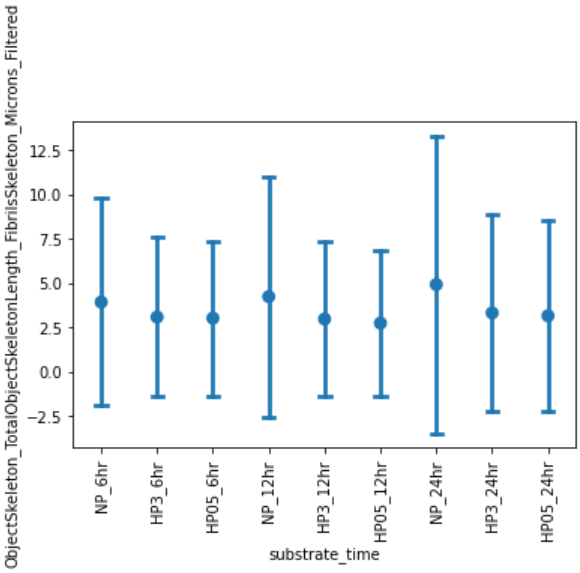
substrate_time	count	mean	std	min	25%	50%	75%	max
NP_6hr	3585.0	3.921623	5.847600	0.28112	0.678684	1.919607	4.634342	65.993570
HP3_6hr	2714.0	3.075134	4.509960	0.28112	0.562240	1.522044	3.558095	48.475940
HP05_6hr	2244.0	3.011430	4.375017	0.28112	0.678684	1.522044	3.626306	54.186530
NP_12hr	7236.0	4.228600	6.787116	0.28112	0.795127	2.036051	4.867230	106.124099
HP3_12hr	5905.0	2.963535	4.396694	0.28112	0.562240	1.522044	3.441651	58.014042
HP05_12hr	3803.0	2.731973	4.122494	0.28112	0.562240	1.357367	3.136415	62.543643
NP_24hr	7734.0	4.912222	8.385432	0.28112	0.843360	2.200727	5.429470	219.284475
HP3_24hr	10711.0	3.309378	5.568069	0.28112	0.678684	1.638487	3.722771	127.560859
HP05_24hr	16862.0	3.146132	5.351153	0.28112	0.562240	1.473811	3.509862	131.862421

In [27]:

```
# a very modest increase in fiber length on the non porous substrate but very big error bars:

ax = sns.pointplot(x="substrate_time",
                   y="ObjectSkeleton_TotalObjectSkeletonLength_FibrilsSkeleton_Microns_Filtered",
                   data=fibril_df,
                   join=False,
                   capsize=0.2,
```

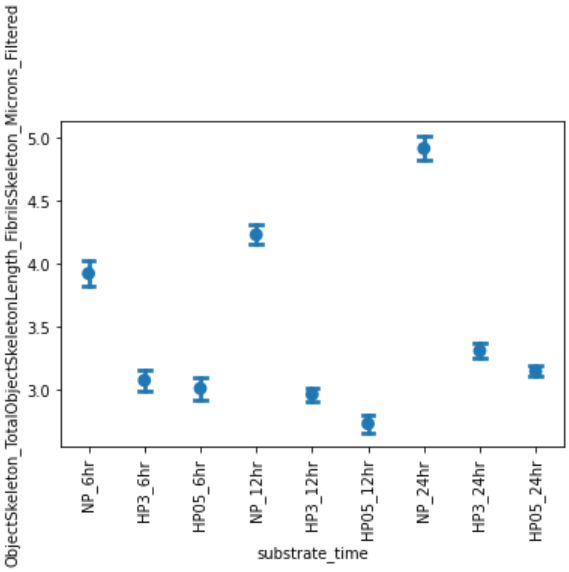
```
                ci="sd")

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[27]:
```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
 Text(7, 0, 'HP3_24hr'),
 Text(8, 0, 'HP05_24hr')]
```



In [28]:
```
# a very modest increase in fiber length on the non porous substrate
# mean +/- sem:

ax = sns.pointplot(x="substrate_time",
                   y="ObjectSkeleton_TotalObjectSkeletonLength_FibrilsSkeleton_Microns_Filtered",
                   data=fibril_df,
                   join=False,
                   capsize=0.2,
                   ci=68)

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[28]:
```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
 Text(7, 0, 'HP3_24hr'),
 Text(8, 0, 'HP05_24hr')]
```



Skeleton length conclusions:

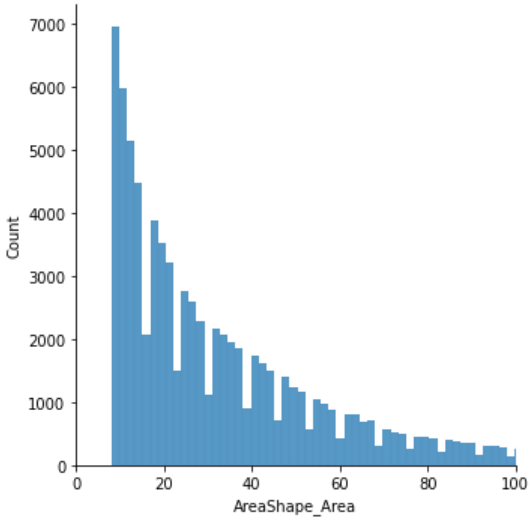
We see the same trend for fibril length as for the MajorAxisLength. At the 24 hr time point for the non porous substrate, the skeleton length is a bit longer than the MajorAxisLength, which would suggest that these long fibrils have some curvature. Keep in mind that MeasureObjectSkeleton makes the objects smaller in all dimensions (it doesn't shrink them along the long axis only; it also takes length off the end of the objects). Round objects will be shrunk proportionally more along their long axis (pixels will be taken off the long axis until the short axis reaches 1 pixel in width).

Data exploration: filter out small and round objects to identify fibrils

As discussed above, skeletonizing objects inaccurately skeletonizes very small and round objects. For that reason, we want to apply filtering to only analyze objects that are long and narrow (require a fibril to be long and narrow).

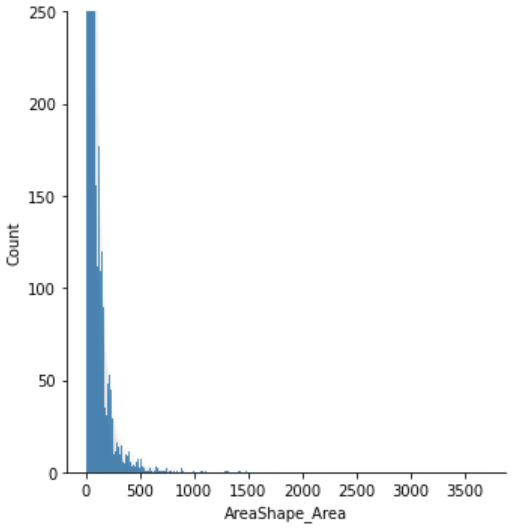
In [29]:
```
# Let's investigate the distribution of the small areas in order to set a cutoff
# Here we see a lot of objects that are very small and then levels off:
# A threshold of 30 pixels would eliminate many small objects
# Looking at the images confirms that this is a good first approximation to eliminate small round blobs

ax = sns.displot(data=fibril_df,
                 x="AreaShape_Area")

ax.set(xlim=(0,100))
```

Out[29]:
```
<seaborn.axisgrid.FacetGrid at 0x137773e80>
```



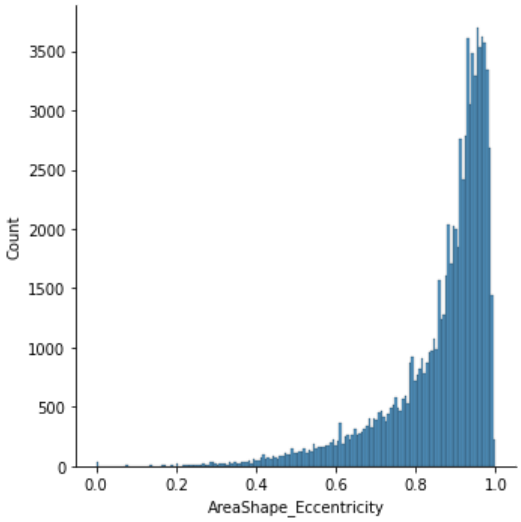
In [30]:
```
# zooming in on the y axis:

ax = sns.displot(data=fibril_df,
                 x="AreaShape_Area")

ax.set(ylim=(0,250))
```

Out[30]:
```
<seaborn.axisgrid.FacetGrid at 0x1377961f0>
```



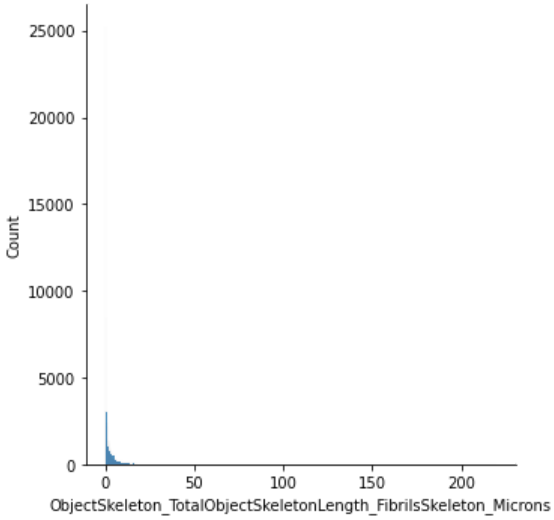
In [31]:
```
# Let's investigate the distribution of eccentricity (roundness) in order to investigate
# A sphere has Eccentricity of 0 and a line has Eccentricity of 1
# The good news here is that most of the objects in this dataset are very eccentric (> 0.8)
# Investigating the images themselves suggests that we should use a threshold of ~ 0.8 for eccentricity
```

```
ax = sns.displot(data=fibril_df,
                 x="AreaShape_Eccentricity")
```



In [32]:

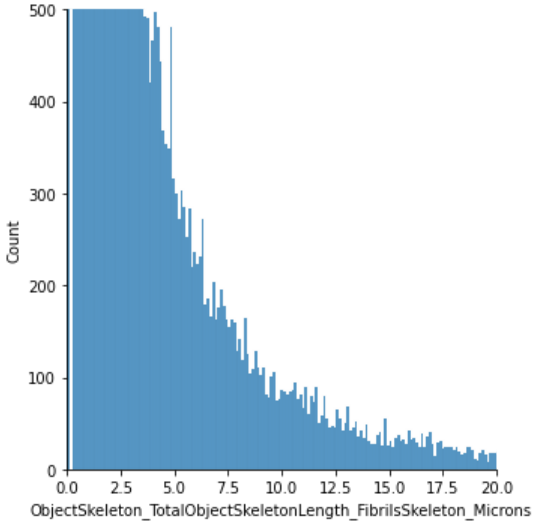
```
# Exploring skeletonized length
# There's a huge # of objects w/ length 0 or thereabouts

ax = sns.displot(data=fibril_df,
                 x="ObjectSkeleton_TotalObjectSkeletonLength_FibrilsSkeleton_Microns")
```



In [33]:

```
# seems to be an inflection point around 5

ax = sns.displot(data=fibril_df,
                 x="ObjectSkeleton_TotalObjectSkeletonLength_FibrilsSkeleton_Microns")

ax.set(ylim=(0,500), xlim=(0,20))
```

Out[33]: <seaborn.axisgrid.FacetGrid at 0x133046bb0>



In [34]:

```
# Let's apply these filters - area > 30 px & eccentricity > 0.8 and then repeat the analysis.

filtered_fibril_df = pd.DataFrame()

filtered_fibril_df = fibril_df.loc[((fibril_df['AreaShape_Area'] >= 30) & (fibril_df['AreaShape_Eccentricity'] >= 0.8))]

# confirm that the Area min is 30:
print("Area:\n", filtered_fibril_df['AreaShape_Area'].describe(), "\n")

# confirm that the Eccentricity min is 0.9:
print("Eccentricity:\n", filtered_fibril_df['AreaShape_Eccentricity'].describe())
```

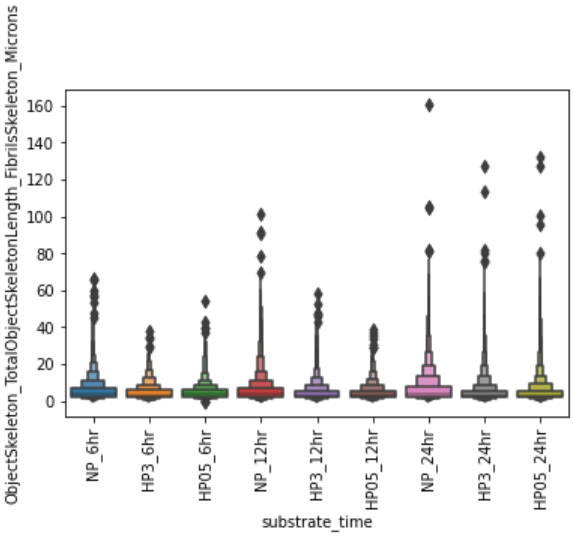
```
Area:
 count    31597.000000
mean        88.959427
std         99.435041
min         30.000000
25%         42.000000
50%         60.000000
75%         97.000000
max       3547.000000
Name: AreaShape_Area, dtype: float64

Eccentricity:
 count    31597.000000
mean         0.925495
std          0.050769
min          0.800075
25%          0.891669
50%          0.938378
75%          0.967379
max          0.998642
Name: AreaShape_Eccentricity, dtype: float64
```

In [35]:

```
# now analyze skeletonized length w/in these more fibril like objects:

ax = sns.boxenplot(x="substrate_time",
                   y="ObjectSkeleton_TotalObjectSkeletonLength_FibrilsSkeleton_Microns",
                   data=filtered_fibril_df)

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[35]:

```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
 Text(7, 0, 'HP3_24hr'),
 Text(8, 0, 'HP05_24hr')]
```

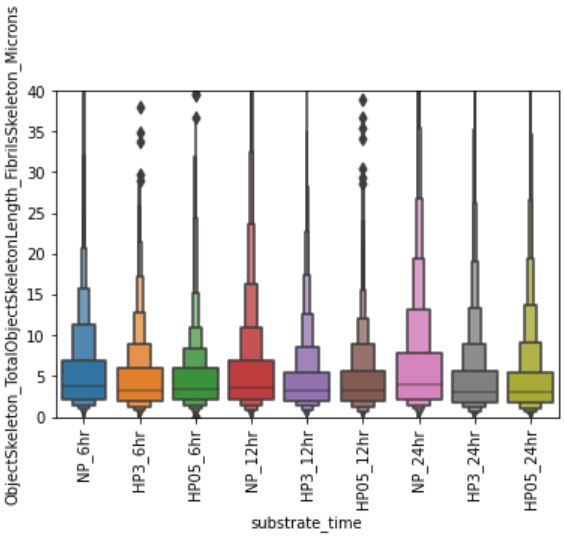


In [36]:

```
# zoom in on y:
# easier to see the increase in length for the NP substrate at 6 vs 12 vs 24 hrs

ax = sns.boxenplot(x="substrate_time",
                   y="ObjectSkeleton_TotalObjectSkeletonLength_FibrilsSkeleton_Microns",
                   data=filtered_fibril_df)

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)

ax.set(ylim=(0,40))
```

Out[36]:

```
[(0.0, 40.0)]
```



In [37]:

```
filtered_fibril_df.groupby("substrate_time")['ObjectSkeleton_TotalObjectSkeletonLength_FibrilsSkeleton_Microns'].describe()
```

Out[37]:

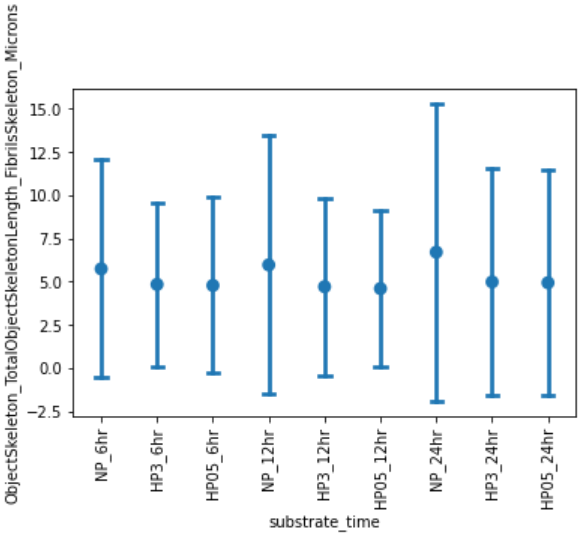
substrate_time	count	mean	std	min	25%	50%	75%	max
NP_6hr	1959.0	5.735259	6.314215	0.0	2.152495	3.790982	6.869175	65.993570
HP3_6hr	1253.0	4.835363	4.749716	0.0	1.919607	3.276975	6.059921	38.014564
HP05_6hr	1109.0	4.776092	5.092752	0.0	2.036051	3.393419	5.923499	54.186530
NP_12hr	4132.0	5.963787	7.494625	0.0	2.036051	3.674539	6.971492	101.602773
HP3_12hr	2783.0	4.704049	5.149306	0.0	1.919607	3.160531	5.347132	58.014042
HP05_12hr	1621.0	4.594106	4.545904	0.0	1.919607	3.160531	5.594146	38.994346
NP_24hr	4585.0	6.701394	8.596354	0.0	2.200727	3.955659	7.863085	160.560068
HP3_24hr	5662.0	4.979502	6.599976	0.0	1.803164	3.112299	5.690611	127.560859
HP05_24hr	8493.0	4.932108	6.539241	0.0	1.754931	3.044087	5.477702	131.862421

In [38]:

```
# a very modest increase in fiber length on the non porous substrate but very big error bars:
```

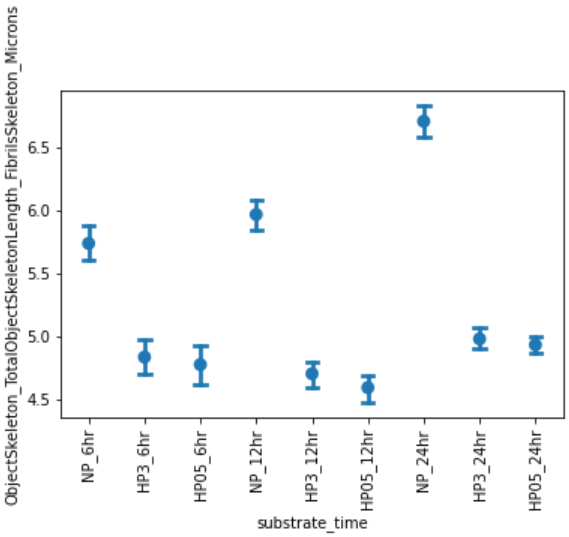
```
ax = sns.pointplot(x="substrate_time",
                   y="ObjectSkeleton_TotalObjectSkeletonLength_FibrilsSkeleton_Microns",
                   data=filtered_fibril_df,
                   join=False,
                   capsize=0.2,
                   ci="sd")

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[38]:
```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
 Text(7, 0, 'HP3_24hr'),
 Text(8, 0, 'HP05_24hr')]
```



In [39]:
```
# we have indeed filtered out the majority of small objects

ax = sns.pointplot(x="substrate_time",
                   y="ObjectSkeleton_TotalObjectSkeletonLength_FibrilsSkeleton_Microns",
                   data=filtered_fibril_df,
                   join=False,
                   capsize=0.2,
                   ci=68)

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

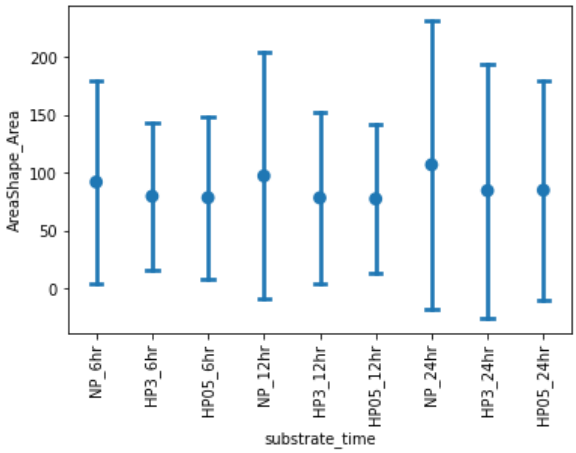
Out[39]:
```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
 Text(7, 0, 'HP3_24hr'),
 Text(8, 0, 'HP05_24hr')]
```



In [40]:

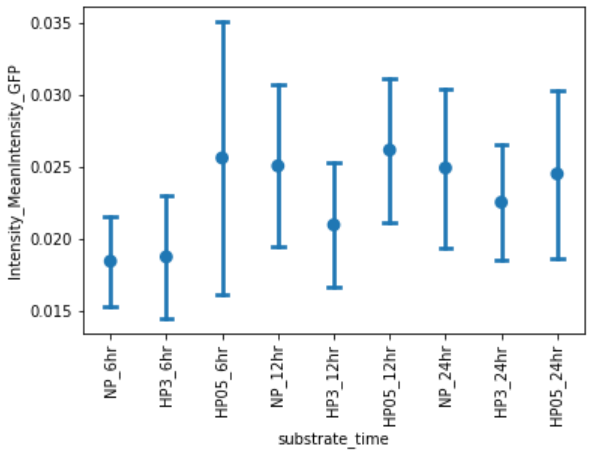
```
# a very modest increase in area on the non porous substrate but very big error bars:

ax = sns.pointplot(x="substrate_time",
                   y="AreaShape_Area",
                   data=filtered_fibril_df,
                   join=False,
                   capsize=0.2,
                   ci="sd")

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[40]:
```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
 Text(7, 0, 'HP3_24hr'),
 Text(8, 0, 'HP05_24hr')]
```



In [41]:
```
# a very modest increase in mean intensity on the non porous substrate but very big error bars:

ax = sns.pointplot(x="substrate_time",
                   y="Intensity_MeanIntensity_GFP",
                   data=filtered_fibril_df,
                   join=False,
                   capsize=0.2,
                   ci="sd")

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[41]:
```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
 Text(7, 0, 'HP3_24hr'),
 Text(8, 0, 'HP05_24hr')]
```



Does image area occupied change with biological condition?

In [42]:
```
# path to csv file containing fibril data that was exported from CellProfiler
fibril_image_path = '../cellprofiler-output/data/Batch2_Image.csv'
```

```
# import the fibril data into pandas
fibril_image_df = pd.read_csv(fibril_image_path)

fibril_image_df.head()
```

Out[42]:

	AreaOccupied_AreaOccupied_Fibrils	AreaOccupied_Perimeter_Fibrils	AreaOccupied_TotalArea_Fibrils	Channel_GFP	Count_BrightNoise	Count_FibrilCen
0	16232.0	10897.0	966288.0	-1	0.0	5
1	15707.0	10151.0	966288.0	-1	0.0	3
2	11282.0	7131.0	966288.0	-1	0.0	2
3	11312.0	7546.0	926822.0	-1	1.0	3
4	14267.0	9151.0	966288.0	-1	0.0	3

5 rows × 106 columns

In [43]:

```
# all the image data available

for column in fibril_image_df.columns:
    print(column)
```

```
AreaOccupied_AreaOccupied_Fibrils
AreaOccupied_Perimeter_Fibrils
AreaOccupied_TotalArea_Fibrils
Channel_GFP
Count_BrightNoise
Count_FibrilCenters
Count_Fibrils
Count_NoiseExpanded
Crop_AreaRetainedAfterCropping_EnhancedMaskedCroppedTubes
Crop_AreaRetainedAfterCropping_GFPRescaledCropped
Crop_OriginalImageArea_EnhancedMaskedCroppedTubes
Crop_OriginalImageArea_GFPRescaledCropped
ExecutionTime_01Images
ExecutionTime_02Metadata
ExecutionTime_03NamesAndTypes
ExecutionTime_04Groups
ExecutionTime_05CorrectIlluminationCalculate
ExecutionTime_06CorrectIlluminationApply
ExecutionTime_07EnhanceOrSuppressFeatures
ExecutionTime_08GaussianFilter
ExecutionTime_09IdentifyPrimaryObjects
ExecutionTime_10ExpandOrShrinkObjects
ExecutionTime_11MaskImage
ExecutionTime_12Crop
ExecutionTime_13IdentifyPrimaryObjects
ExecutionTime_14MeasureObjectSizeShape
ExecutionTime_15MeasureImageAreaOccupied
ExecutionTime_16MeasureObjectIntensity
ExecutionTime_17ConvertObjectsToImage
ExecutionTime_18MorphologicalSkeleton
ExecutionTime_19ExpandOrShrinkObjects
ExecutionTime_20MeasureObjectSkeleton
ExecutionTime_21RescaleIntensity
ExecutionTime_22ImageMath
ExecutionTime_23Crop
ExecutionTime_24OverlayObjects
ExecutionTime_25SaveImages
ExecutionTime_26SaveImages
ExecutionTime_27DisplayDataOnImage
ExecutionTime_28SaveImages
ExecutionTime_29DisplayDataOnImage
ExecutionTime_30SaveImages
ExecutionTime_31DisplayDataOnImage
ExecutionTime_32SaveImages
FileName_GFP
Frame_GFP
Group_Index
Group_Number
Height_GFP
ImageNumber
ImageSet_ImageSet
MD5Digest_GFP
Metadata_FileLocation
Metadata_Frame
Metadata_Series
Metadata_channel
Metadata_image
Metadata_substrate
Metadata_time
Metadata_timepoint
ModuleError_01Images
ModuleError_02Metadata
ModuleError_03NamesAndTypes
ModuleError_04Groups
ModuleError_05CorrectIlluminationCalculate
ModuleError_06CorrectIlluminationApply
ModuleError_07EnhanceOrSuppressFeatures
```

```
ModuleError_08GaussianFilter
ModuleError_09IdentifyPrimaryObjects
ModuleError_10ExpandOrShrinkObjects
ModuleError_11MaskImage
ModuleError_12Crop
ModuleError_13IdentifyPrimaryObjects
ModuleError_14MeasureObjectSizeShape
ModuleError_15MeasureImageAreaOccupied
ModuleError_16MeasureObjectIntensity
ModuleError_17ConvertObjectsToImage
ModuleError_18MorphologicalSkeleton
ModuleError_19ExpandOrShrinkObjects
ModuleError_20MeasureObjectSkeleton
ModuleError_21RescaleIntensity
ModuleError_22ImageMath
ModuleError_23Crop
ModuleError_24OverlayObjects
ModuleError_25SaveImages
ModuleError_26SaveImages
ModuleError_27DisplayDataOnImage
ModuleError_28SaveImages
ModuleError_29DisplayDataOnImage
ModuleError_30SaveImages
ModuleError_31DisplayDataOnImage
ModuleError_32SaveImages
PathName_GFP
ProcessingStatus
Scaling_GFP
Series_GFP
Threshold_FinalThreshold_BrightNoise
Threshold_FinalThreshold_Fibrils
Threshold_OrigThreshold_BrightNoise
Threshold_OrigThreshold_Fibrils
Threshold_SumOfEntropies_BrightNoise
Threshold_SumOfEntropies_Fibrils
Threshold_WeightedVariance_BrightNoise
Threshold_WeightedVariance_Fibrils
URL_GFP
Width_GFP
```

In [44]:

```
# The data are grouped into biological classes by substrate and timepoint

# Some of the substrate names include '_A' and '_B' but not others
# Here, we'll replace these names with an empty string so that everything matches
find_replace_dict = {'HP05_A': 'HP05', 'HP05_B': 'HP05'}

fibril_image_df.replace(to_replace=find_replace_dict, inplace=True)

# now create one column that contains both variables

fibril_image_df['substrate_time']  = fibril_image_df['Metadata_substrate'] + '_' + fibril_image_df['Metadata_timepoint']


# custom sort to order variables to our desired order
# from https://towardsdatascience.com/how-to-do-a-custom-sort-on-pandas-dataframe-ac18e7ea5320
substrate_ordered = ['NP_6hr', 'NP_12hr', 'NP_24hr', 'HP05_6hr', 'HP05_12hr', 'HP05_24hr', 'HP3_6hr', 'HP3_12hr', 'HP3_24hr'
timepoint_ordered = ['NP_6hr', 'HP3_6hr', 'HP05_6hr', 'NP_12hr', 'HP3_12hr', 'HP05_12hr', 'NP_24hr', 'HP3_24hr', 'HP05_24hr'

condition_sort_order = CategoricalDtype(timepoint_ordered,
    ordered=True)


fibril_image_df['substrate_time'] = fibril_image_df['substrate_time'].astype(condition_sort_order)
fibril_image_df.sort_values('substrate_time', inplace=True)

fibril_image_df.head()
```

Out[44]:

	AreaOccupied_AreaOccupied_Fibrils	AreaOccupied_Perimeter_Fibrils	AreaOccupied_TotalArea_Fibrils	Channel_GFP	Count_BrightNoise	Count_FibrilC
202	16268.0	10202.0	966288.0	-1	0.0	
185	23166.0	14429.0	966288.0	-1	0.0	
186	4886.0	2924.0	966288.0	-1	0.0	
187	16937.0	9226.0	966288.0	-1	0.0	
188	4106.0	2524.0	966288.0	-1	0.0	

5 rows × 107 columns

In [45]:

```
# compute the percent of the fibril occupied area relative to the total area
# the AreaOccupied_TotalArea_Fibrils column does not include masked areas
# we'll use that to calculate the % area occupied

fibril_image_df['percent_fibril_occupied'] = fibril_image_df['AreaOccupied_AreaOccupied_Fibrils'] / fibril_image_df['AreaOccu

fibril_image_df
```

Out[45]:

	AreaOccupied_AreaOccupied_Fibrils	AreaOccupied_Perimeter_Fibrils	AreaOccupied_TotalArea_Fibrils	Channel_GFP	Count_BrightNoise	Count_FibrilC
202	16268.0	10202.0	966288.0	-1	0.0	
185	23166.0	14429.0	966288.0	-1	0.0	
186	4886.0	2924.0	966288.0	-1	0.0	
187	16937.0	9226.0	966288.0	-1	0.0	
188	4106.0	2524.0	966288.0	-1	0.0	
...	...	...	...	...	...	
79	29888.0	18516.0	907772.0	-1	1.0	
78	39050.0	22585.0	943980.0	-1	1.0	
77	28719.0	15691.0	966288.0	-1	0.0	
90	22636.0	12302.0	966288.0	-1	0.0	
101	20792.0	13618.0	966288.0	-1	0.0	

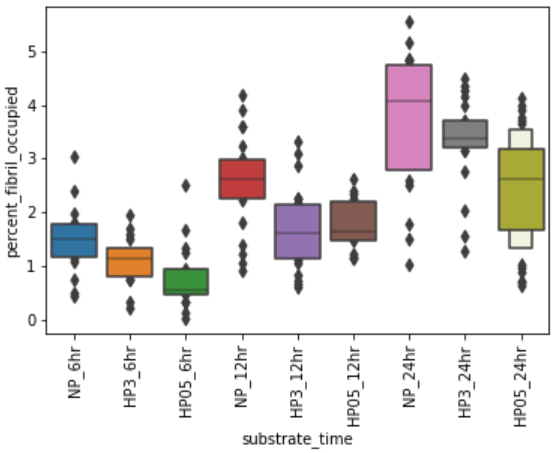
203 rows × 108 columns

In [46]:

```
ax = sns.boxenplot(x="substrate_time",
                   y="percent_fibril_occupied",
                   data=fibril_image_df)

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[46]:

```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
 Text(7, 0, 'HP3_24hr'),
 Text(8, 0, 'HP05_24hr')]
```



In [47]:

```
# area occupied numbers

fibril_image_df.groupby('substrate_time')['percent_fibril_occupied'].describe()
```

Out[47]:

substrate_time	count	mean	std	min	25%	50%	75%	max
NP_6hr	18.0	1.487571	0.636037	0.424925	1.150382	1.504670	1.778947	3.041019
HP3_6hr	17.0	1.097076	0.456101	0.201803	0.813880	1.146743	1.330042	1.951660
HP05_6hr	18.0	0.768585	0.603785	0.027425	0.456851	0.552061	0.935565	2.496668
NP_12hr	23.0	2.552651	0.875704	0.900249	2.255694	2.631410	2.979702	4.192953
HP3_12hr	22.0	1.699570	0.762412	0.595856	1.140125	1.620221	2.156105	3.313401
HP05_12hr	14.0	1.776517	0.481569	1.131236	1.482607	1.652665	2.197715	2.614625
NP_24hr	19.0	3.670957	1.308610	1.026299	2.783953	4.070526	4.749321	5.556728
HP3_24hr	23.0	3.334387	0.808904	1.273865	3.196861	3.383877	3.707049	4.490857
HP05_24hr	49.0	2.469461	0.954372	0.622278	1.660892	2.605538	3.169256	4.136740

% Area occupied / image conclusions

Here we see a very nice trend w/ increasing % occupied with time and more area occupied on the nonporous substrates compared to the porous substrates. Since the area and length don't seem to change significantly for the 0.5 micron and 3 micron surfaces, this suggests to me that an increase in the # of fibrils accounts for the increase in area rather than a change in the size / shape.

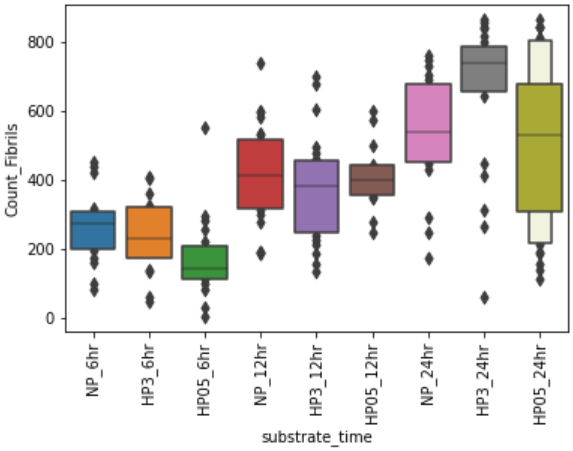
Unfortunately we don't have cell-level segmentation, which would really help to test this out.

Does number of fibrils per image change with biological condition?

See above -- since the area occupied changes without the morphology significantly changing (especially in the porous substrate examples), I wanted to test if the number of fibrils increases over time. We do see that and the number doesn't seem to be affected very much by biological condition

In [48]:
```
ax = sns.boxenplot(x="substrate_time",
                   y="Count_Fibrils",
                   data=fibril_image_df)

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

Out[48]:
```
[Text(0, 0, 'NP_6hr'),
 Text(1, 0, 'HP3_6hr'),
 Text(2, 0, 'HP05_6hr'),
 Text(3, 0, 'NP_12hr'),
 Text(4, 0, 'HP3_12hr'),
 Text(5, 0, 'HP05_12hr'),
 Text(6, 0, 'NP_24hr'),
 Text(7, 0, 'HP3_24hr'),
 Text(8, 0, 'HP05_24hr')]
```



In [49]:
```
# fibril count numbers

fibril_image_df.groupby('substrate_time')['Count_Fibrils'].describe()
```

Out[49]:

substrate_time	count	mean	std	min	25%	50%	75%	max
NP_6hr	18.0	265.055556	106.138906	82.0	200.50	277.0	310.25	455.0
HP3_6hr	17.0	233.588235	108.743999	47.0	174.00	230.0	322.00	409.0
HP05_6hr	18.0	171.611111	122.293350	7.0	114.25	144.5	207.75	553.0
NP_12hr	23.0	422.565217	143.818960	187.0	317.00	414.0	518.50	741.0
HP3_12hr	22.0	378.954545	156.774171	134.0	247.25	386.0	456.25	701.0
HP05_12hr	14.0	412.928571	99.118782	247.0	359.25	400.5	445.75	601.0
NP_24hr	19.0	535.000000	168.818772	176.0	455.00	539.0	679.50	764.0
HP3_24hr	23.0	663.434783	213.557705	63.0	657.50	740.0	788.00	867.0
HP05_24hr	49.0	508.816327	222.525998	114.0	308.00	533.0	681.00	866.0

Number of fibril numbers

We indeed see a nice trend for all conditions that more objects are detected with an increase in time. This change in distribution is more pronounced than the changes for area or fiber length.

Overall conclusions

We find similar results to those reported in https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5921834/. Area and length appear to increase with time on the non porous substrates. These changes are minimal for the porous substrates with time. The percent area occupied and number of detected fibrils

both increase over time for all conditions.

In [ ]: