In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
import os
import glob
import re
```

Hyperparam

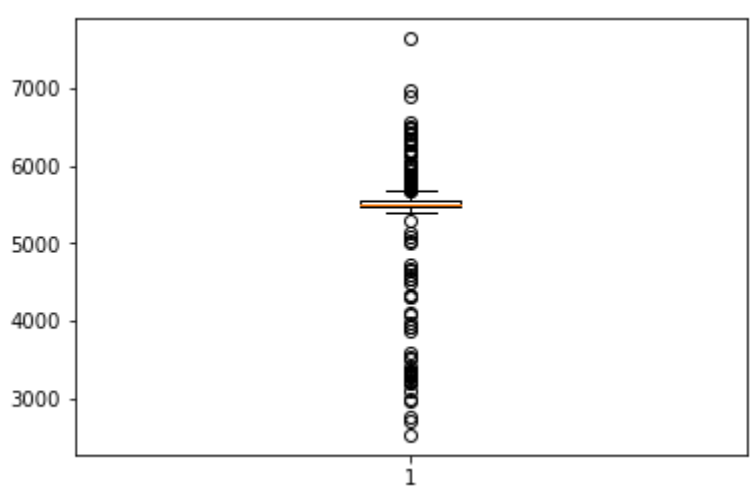
In [2]:
```
parent = 'experimentsAL/res/'
mice = ['163241', '199040', '199043', '199044', '199121']
with open(parent + 'bsoid_labelprob_10Hz_20200719_0251.csv', 'r') as f:
    line = f.readline().split(',')[1:]
labels = len(line)
```

Load file names and primary cleaning

In [3]:
```
f_xyl = glob.glob(parent + 'bsoid_labels*.csv')
f_len = glob.glob(parent + 'bsoid_runlen*.csv')
f_stats = glob.glob(parent + 'bsoid_stats*.csv')
f_trans = glob.glob(parent + 'bsoid_trans*.csv')
list(map(len, [f_xyl, f_len, f_stats, f_trans]))
```

Out[3]:
```
[599, 599, 599, 599]
```

In [4]:
```
lines = []
to_del = []
s = re.compile(r'(?<=bsoid_runlen_30Hz_).+(?=-0000)')
for fname in f_len:
    with open(fname, 'rb') as f:
        f.seek(-2, os.SEEK_END)
        while f.read(1) != b'\n':
            f.seek(-2, os.SEEK_CUR)
        last_line = f.readline().decode()
    last_line = list(map(int, last_line.split(',')))
    l = sum(last_line[-2:])
    if l < 1800:
        to_del.append(s.search(fname).group())
    else:
        lines.append(l)
plt.boxplot(lines);
```



In [5]:
```
to_del
```

Out[5]:
```
['20200719_0604199043_2017-03-24-104856',
 '20200719_0605199121_2017-03-01-113023',
 '20200719_0604199043_2017-05-22-100941',
 '20200719_0605199121_2017-03-01-113552',
 '20200719_0605199121_2017-03-10-103550',
 '20200719_0604199044_2017-03-16-092019',
 '20200719_0604199040_2017-03-23-133618',
 '20200719_0603163241_2017-03-17-092440',
 '20200719_0604199043_2017-05-23-104957',
 '20200719_0604199043_2017-05-22-101647']
```

In [6]:
```
for f in (f_xyl, f_len, f_stats, f_trans):
    for i, fname in enumerate(f):
        for d in to_del:
            if d in fname:
                del f[i]
list(map(len, [f_xyl, f_len, f_stats, f_trans]))
```

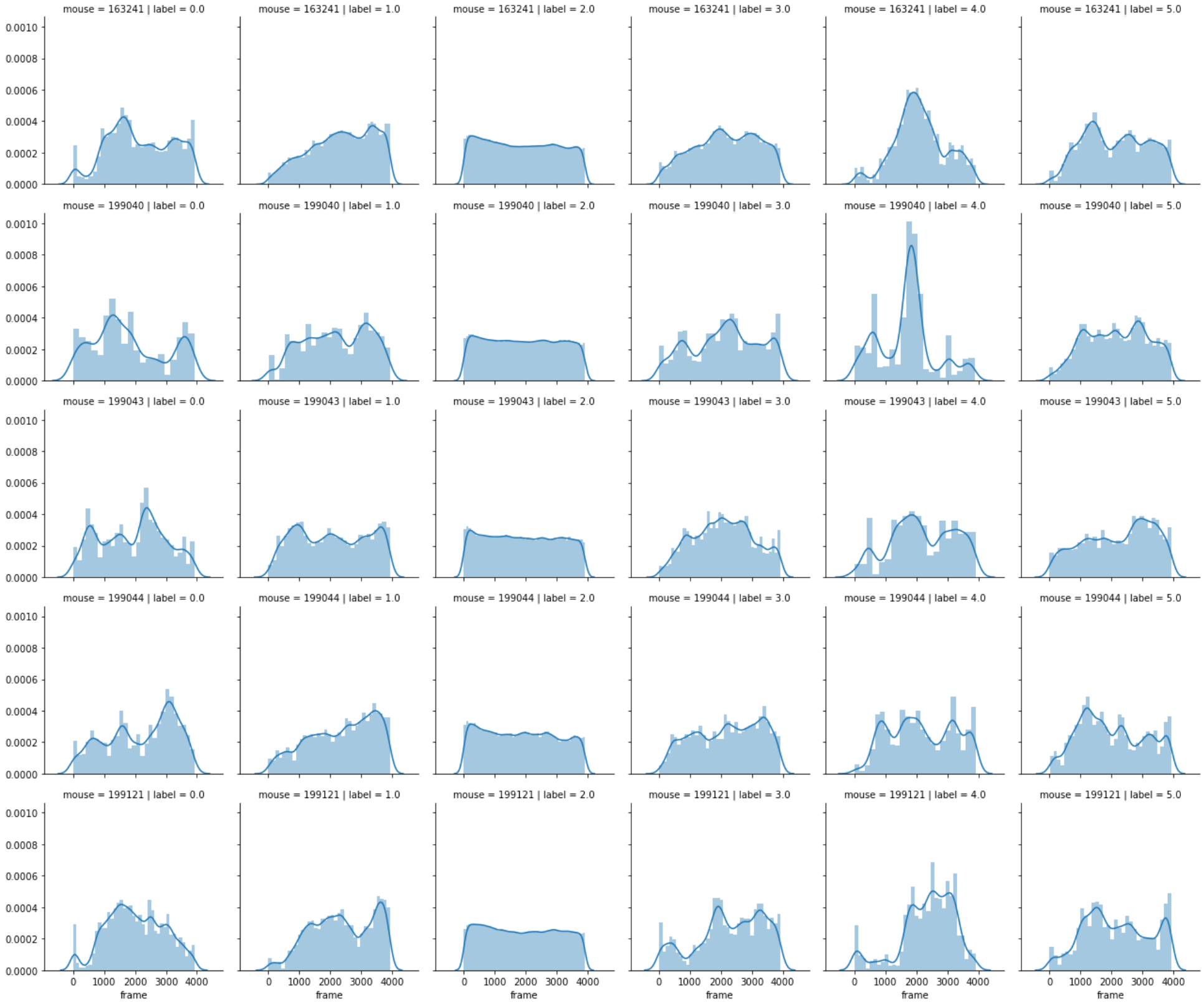
Out[6]:
```
[589, 589, 589, 590]
```

Each Label's Occurrence over Frames (Raw)

In [7]:
```
li = []

for fname in f_xyl:
    for m in mice:
        if m in fname:
            mid = m
            break
    df = pd.read_csv(fname, index_col=0, header=[1, 2])
    df.insert(0, 'mouse', int(mid))
    li.append(df)

xyl_df = pd.concat(li, axis=0, ignore_index=True)

col = [a + b for a in ('s', 'f1', 'f2', 'h1', 'h2', 't') for b in ('x', 'y', 'l')]
col = ['mouse', 'label', 'frame'] + col

xyl_df.columns = col
xyl_df.dropna(inplace=True)
```

In [8]:
```
grid = sns.FacetGrid(xyl_df, col='label', row='mouse')

grid.map(sns.distplot, 'frame')
```

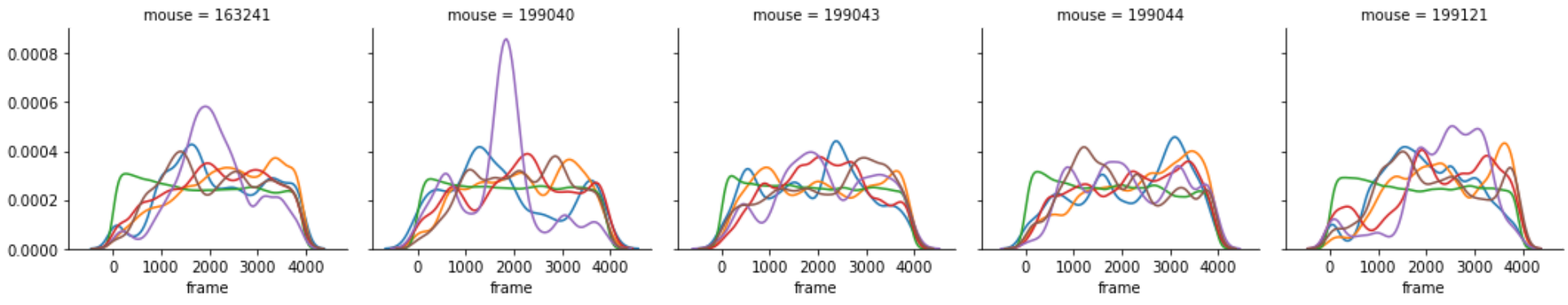
Out[8]: <seaborn.axisgrid.FacetGrid at 0x7fe7de404748>



Comparison in Terms of Mouse

In [9]:
```
grid = sns.FacetGrid(xyl_df, hue='label', col='mouse')

grid.map(sns.distplot, 'frame', hist=False)
```

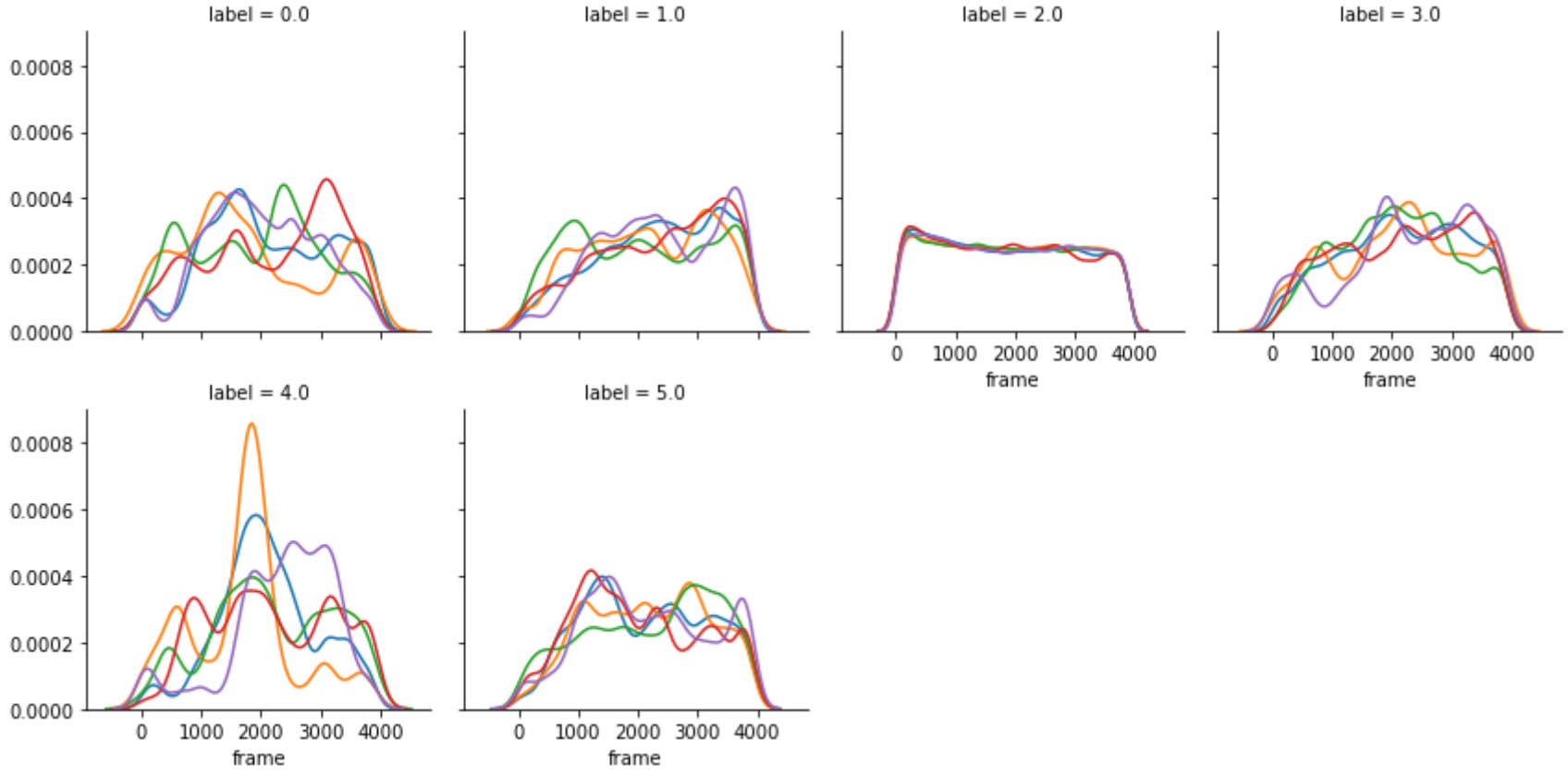
Out[9]: <seaborn.axisgrid.FacetGrid at 0x7fe7dda73ac8>



Comparison in Terms of Label

In [10]:
```
grid = sns.FacetGrid(xyl_df, col='label', hue='mouse', col_wrap=4)

grid.map(sns.distplot, 'frame', hist=False)
```

Out[10]: <seaborn.axisgrid.FacetGrid at 0x7fe7dcc72240>



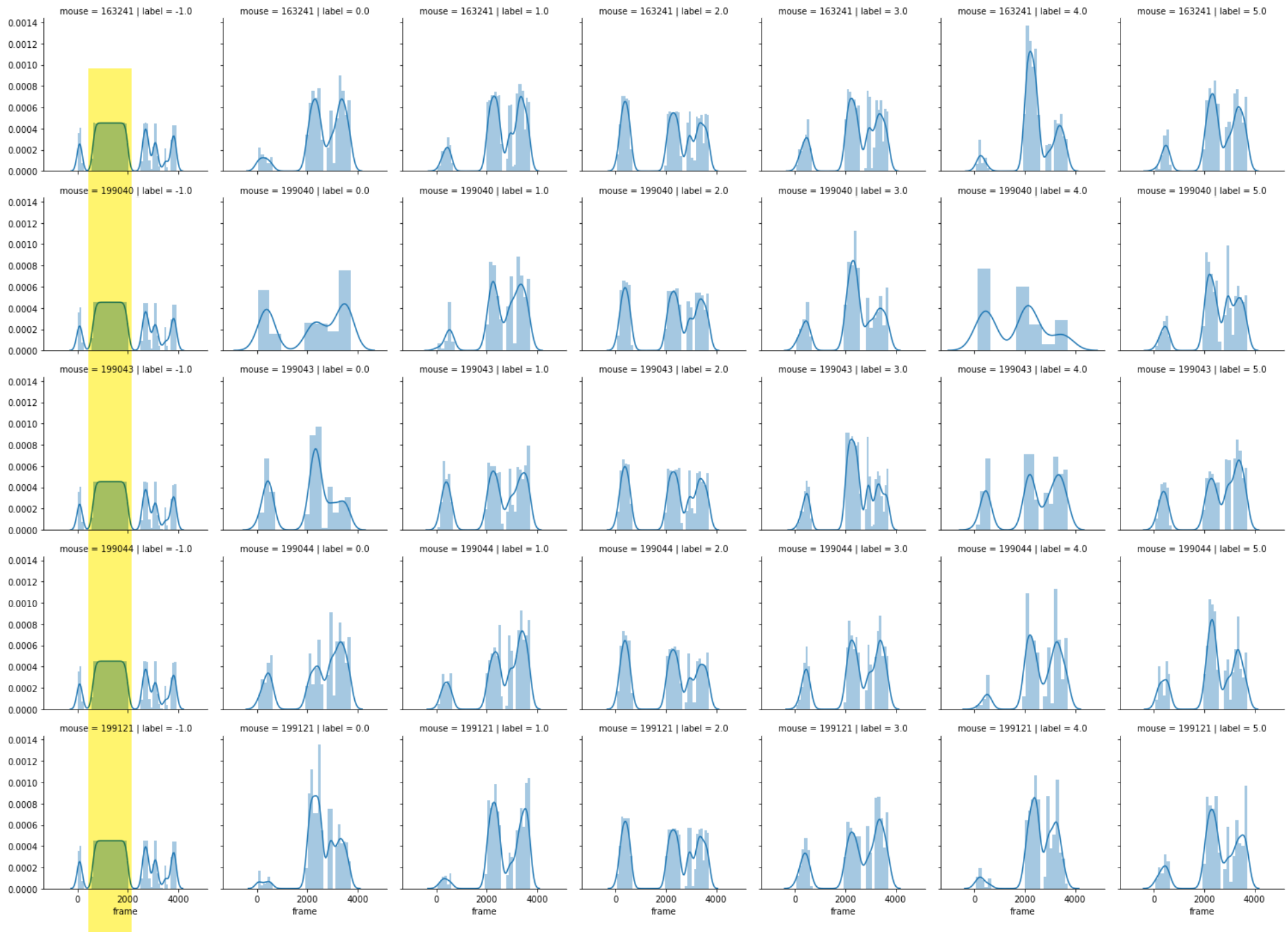
Each Label's Occurrence over Frames (Separate out Potential Detection Errors)

Relabel those with low likelihood as "-1", noise.

In [11]:
```
xyl_df_clean = xyl_df.copy()
mean_like = xyl_df_clean[['sl', 'f1l', 'f2l', 'h1l', 'h2l', 'tl']].mean(axis=1)
xyl_df_clean.loc[mean_like < 0.5, 'label'] = -1
```

In [12]:
```
grid = sns.FacetGrid(xyl_df_clean, col='label', row='mouse')

grid.map(sns.distplot, 'frame')
```

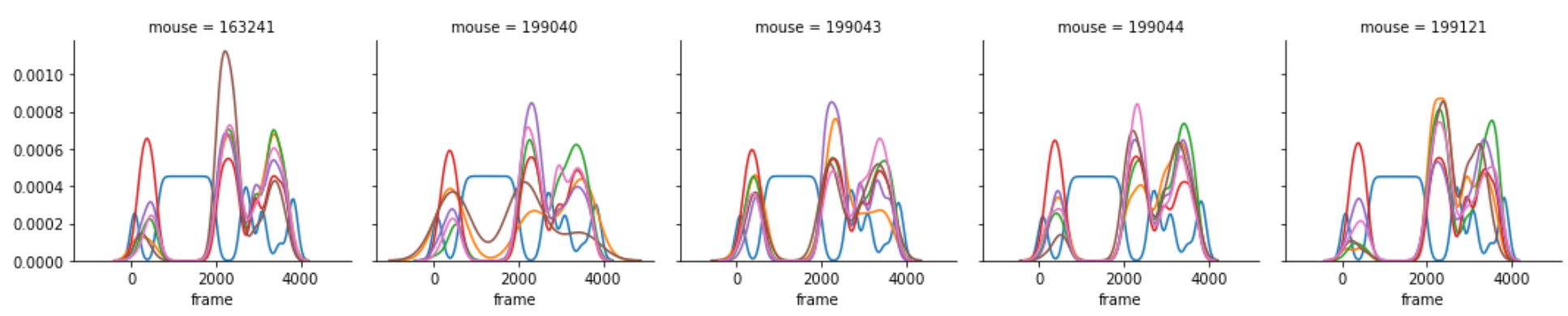
Out[12]: <seaborn.axisgrid.FacetGrid at 0x7fe7dcb02278>



Comparison in Terms of Mouse

In [13]:
```
grid = sns.FacetGrid(xyl_df_clean, hue='label', col='mouse')

grid.map(sns.distplot, 'frame', hist=False)
```

Out[13]: <seaborn.axisgrid.FacetGrid at 0x7fe7da9e7be0>



Comparison in Terms of Label

Note on block OUT[12]:
Why in the middle of the trails there are many low-confidence detections?

A) Mouse appears besides the edges, possibly out-of-canvas.

B) Mouse stands up, snout and front paws become invisible.

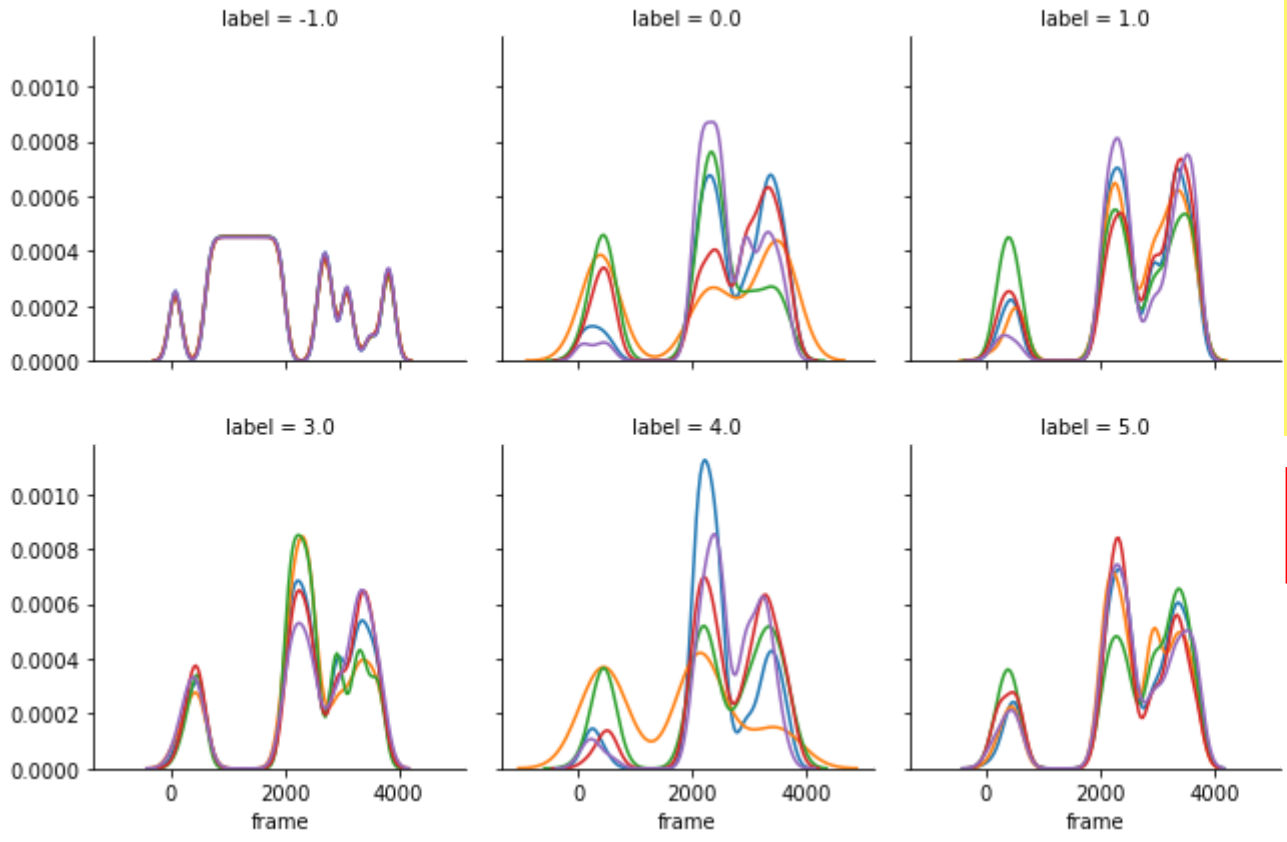
C) Training problem, which is less likely: 1) had manually labeled enough frames (200 as recommended); 2) checked the sample video, detections are fine most of the time, except A) or B) happens.

D) Moving faster-- blurry frames

Note on block OUT[13]:
If at least one of A), B) and D) holds, what does the alternation of noise and other labels manifest?

In [14]:
```
grid = sns.FacetGrid(xyl_df_clean, col='label', hue='mouse', col_wrap=4)

grid.map(sns.distplot, 'frame', hist=False)
```

Out[14]: <seaborn.axisgrid.FacetGrid at 0x7fe7d99c9748>



Note on block OUT[14]:
The most likely label has the lowest variation among mice.

Each Label's Occurrence over Frames (Remove Potential Detection Errors)

In [15]:
```
mean_like = xyl_df[['sl', 'f1l', 'f2l', 'h1l', 'h2l', 'tl']].mean(axis=1)
xyl_df_clean2 = xyl_df.drop(xyl_df[mean_like < 0.5].index)
```

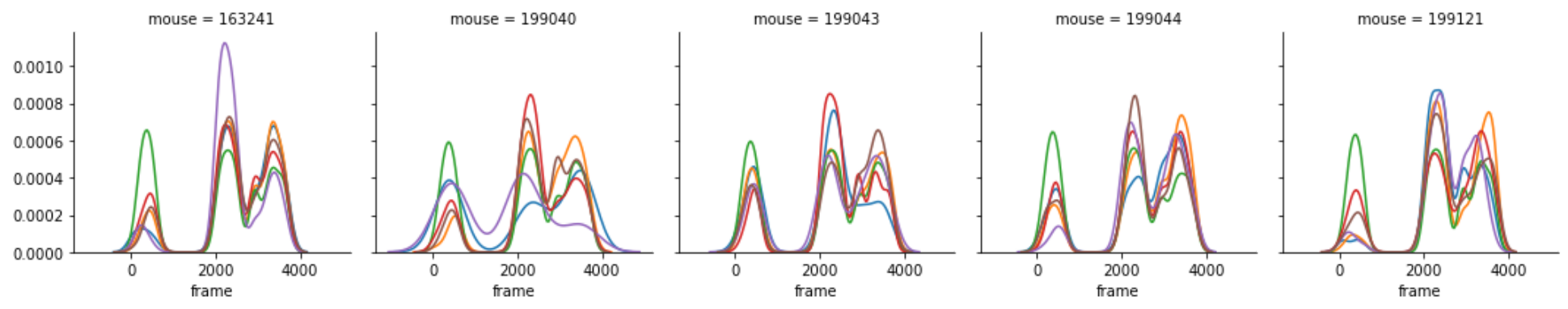
In [16]:
```
grid = sns.FacetGrid(xyl_df_clean2, col='label', row='mouse')

grid.map(sns.distplot, 'frame')
```

Out[16]: <seaborn.axisgrid.FacetGrid at 0x7fe7d99c9da0>

Comparison in Terms of Mouse

In [17]:
```
grid = sns.FacetGrid(xyl_df_clean2, hue='label', col='mouse')

grid.map(sns.distplot, 'frame', hist=False)
```

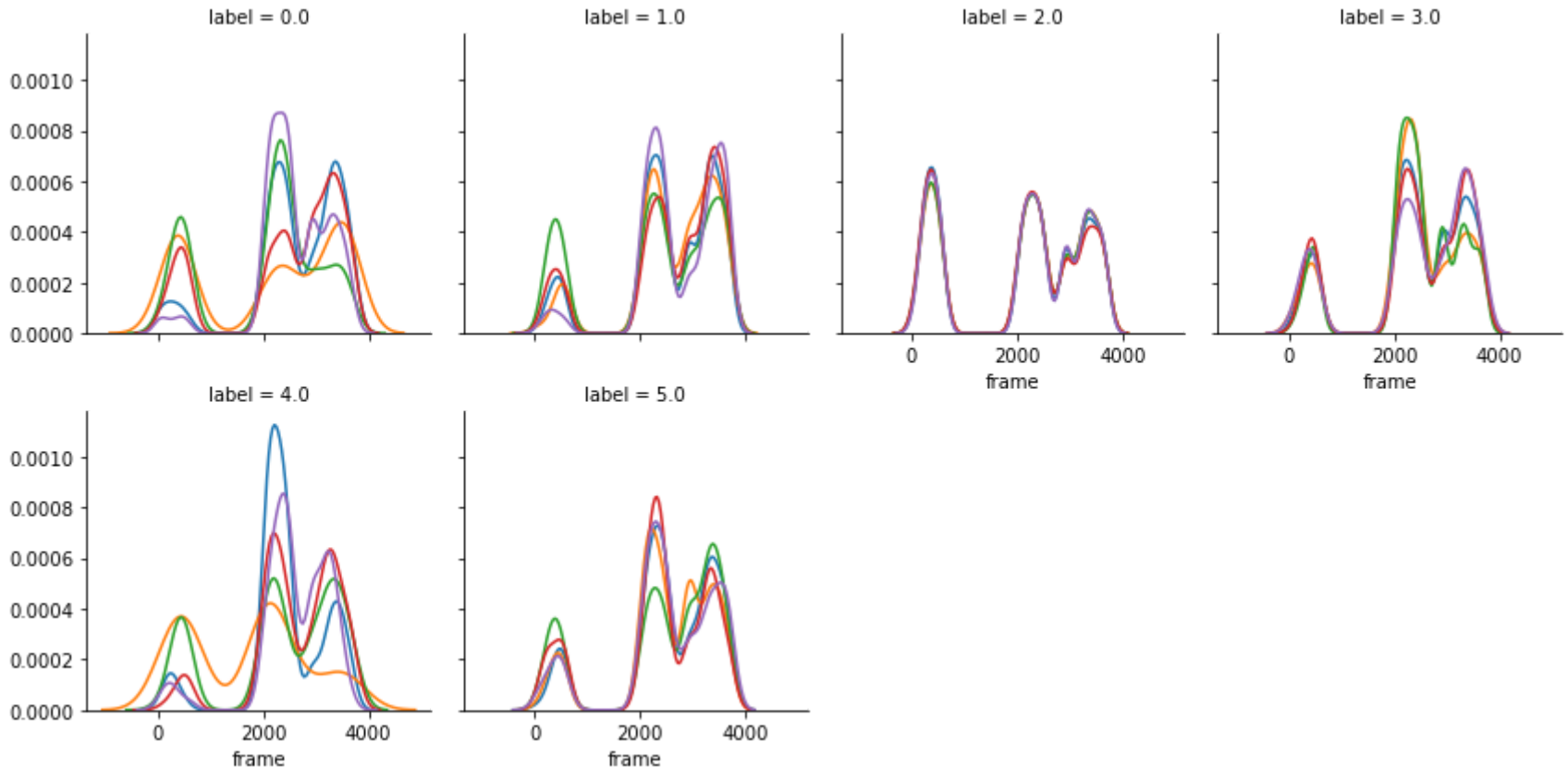
Out[17]: <seaborn.axisgrid.FacetGrid at 0x7fe7c113a940>



Comparison in Terms of Label

In [18]:
```
grid = sns.FacetGrid(xyl_df_clean2, col='label', hue='mouse', col_wrap=4)

grid.map(sns.distplot, 'frame', hist=False)
```

Out[18]: <seaborn.axisgrid.FacetGrid at 0x7fe7c051bd68>



Random Gesture Samples for Each Label (Normalized Relative to the Head; Head-TailRoot Axis Rotated to 0 deg, i.e. Along x Axis)

The sides of the quadrilateral whose vertices are the four paws should not cross.

In [19]:
```
xyl_df_rel = xyl_df_clean2.copy()
for b in ('f1', 'f2', 'h1', 'h2', 't'):
    xyl_df_rel[b + 'x'] -= xyl_df_rel.sx
    xyl_df_rel[b + 'y'] -= xyl_df_rel.sy
xyl_df_rel['sx'] -= xyl_df_rel.sx
xyl_df_rel['sy'] -= xyl_df_rel.sy

angles = np.arctan2(xyl_df_rel.ty, xyl_df_rel.tx)
for b in ('f1', 'f2', 'h1', 'h2', 't'):
    x = xyl_df_rel[b + 'x']#.copy()
    y = xyl_df_rel[b + 'y']#.copy()
    newx = x * np.cos(angles) + y * np.sin(angles)
    newy = y * np.cos(angles) - x * np.sin(angles)
    newx[np.abs(newx) < 1e-5] = 0
    newy[np.abs(newy) < 1e-5] = 0
    xyl_df_rel[b + 'x'] = newx
    xyl_df_rel[b + 'y'] = newy
```

In [20]:
```
N = 10
c = np.linspace(0, 1, num=6)

xs = [s + 'x' for s in ('s', 'f1', 'f2', 'h1', 'h2', 't')]
ys = [s + 'y' for s in ('s', 'f1', 'f2', 'h1', 'h2', 't')]

fig, ax = plt.subplots(N, labels, figsize=(2 * N, 14), sharex='all', sharey='all')

for i in range(labels):
    x = xyl_df_rel.loc[xyl_df_rel.label == i, xs]
    y = xyl_df_rel.loc[xyl_df_rel.label == i, ys]
    n = np.random.choice(len(x), N, replace=False)

    for j, k in enumerate(n):
        for l in ([0, 5], [1, 2], [2, 4], [4, 3], [3, 1]):
            ax[j, i].plot(x.iloc[k, l], y.iloc[k, l], c='black', alpha=0.3)

        ax[j, i].scatter(x.iloc[k], y.iloc[k], c=c, cmap='Set1')


for i in range(labels):
    ax[0, i].set_title('label = %d' % i)
    ax[-1, i].set_xlabel('x')

fig.text(0, 0.5, 'Random Samples', rotation=90)
fig.tight_layout()
```

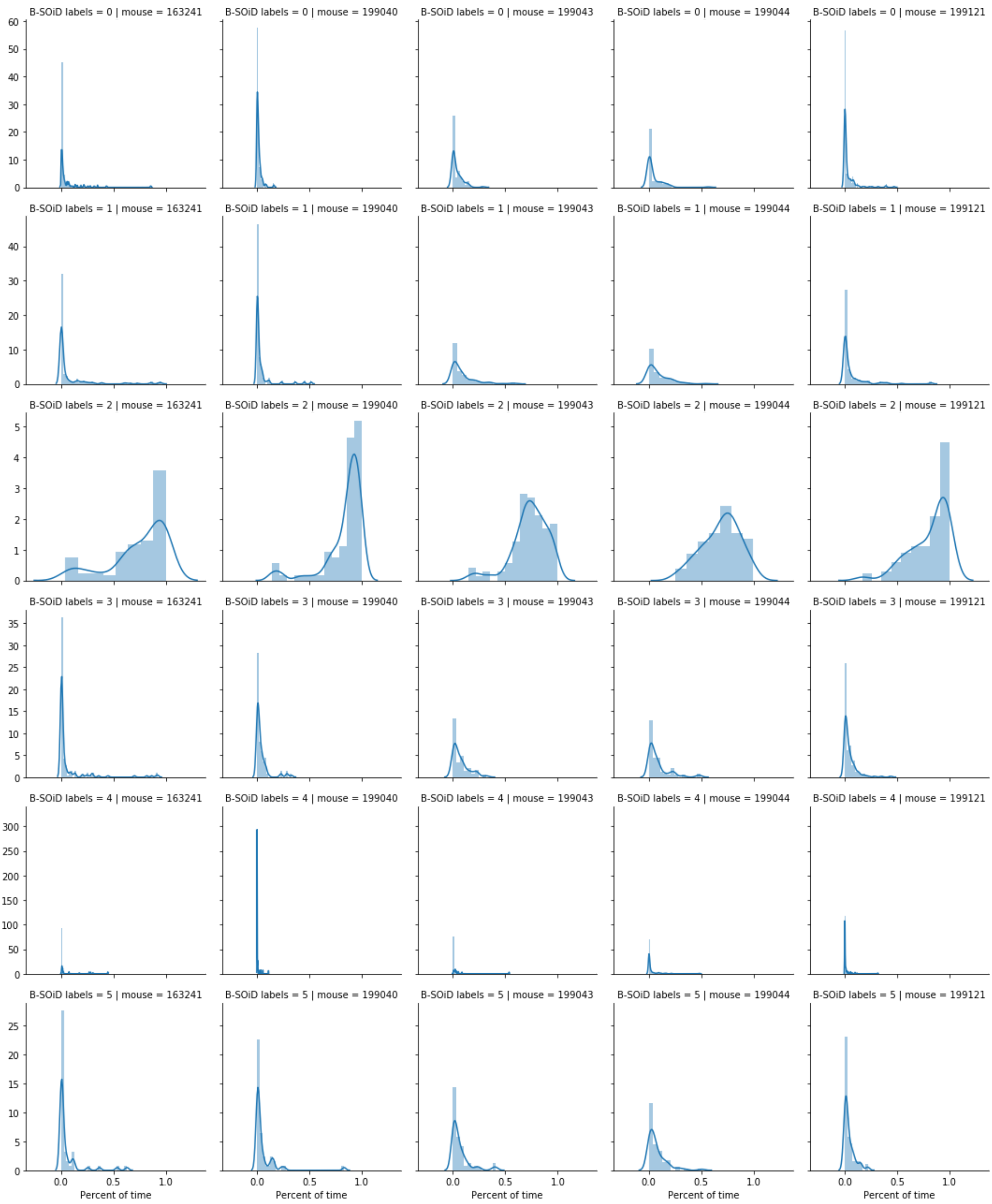
Percent of Time Occurrence

In [21]:
```
li = []

for fname in f_stats:
    for m in mice:
        if m in fname:
            mid = m
            break
    df = pd.read_csv(fname, header=1)
    df.insert(0, 'mouse', int(mid))
    li.append(df)

time_df = pd.concat(li, axis=0, ignore_index=True)
time_df.columns
```

Out[21]:
```
Index(['mouse', 'B-SOiD labels', 'Percent of time', 'Mean duration (frames)',
       '10th %tile (frames)', '25th %tile (frames)', '50th %tile (frames)',
       '75th %tile (frames)', '90th %tile (frames)'],
      dtype='object')
```

In [22]:
```
grid = sns.FacetGrid(time_df, row='B-SOiD labels', col='mouse', sharey='row')
grid.map(sns.distplot, 'Percent of time')
```

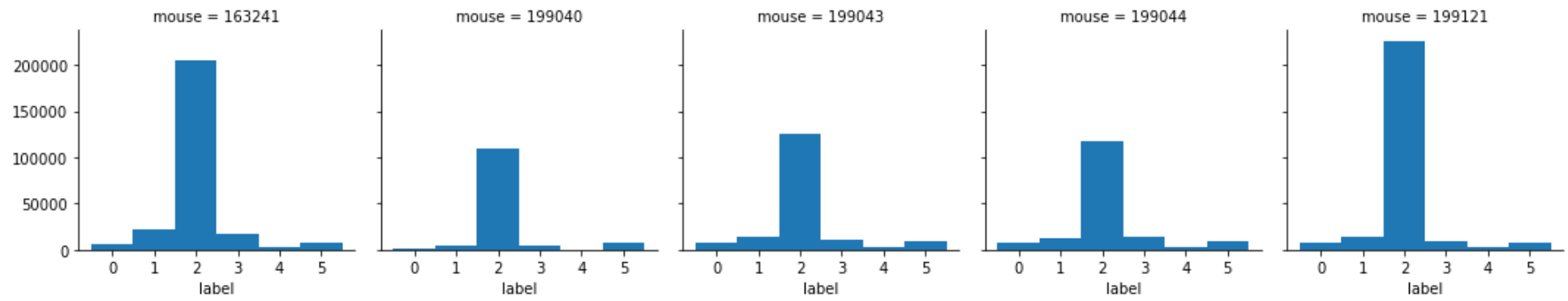
Out[22]: <seaborn.axisgrid.FacetGrid at 0x7fe784cb9c18>



Percent of Time Occurrence for Each Mouse

In [23]:
```
def hist(x, color, **kwargs):
    bins = []
    for i in range(6):
        bins += [i - 0.5, i + 0.5]
    plt.hist(x, align='mid', bins=bins)
    plt.xticks(np.arange(6))

grid = sns.FacetGrid(xyl_df_clean2, col='mouse')
grid.map(hist, 'label')
```

Out[23]: <seaborn.axisgrid.FacetGrid at 0x7fe78456b5f8>

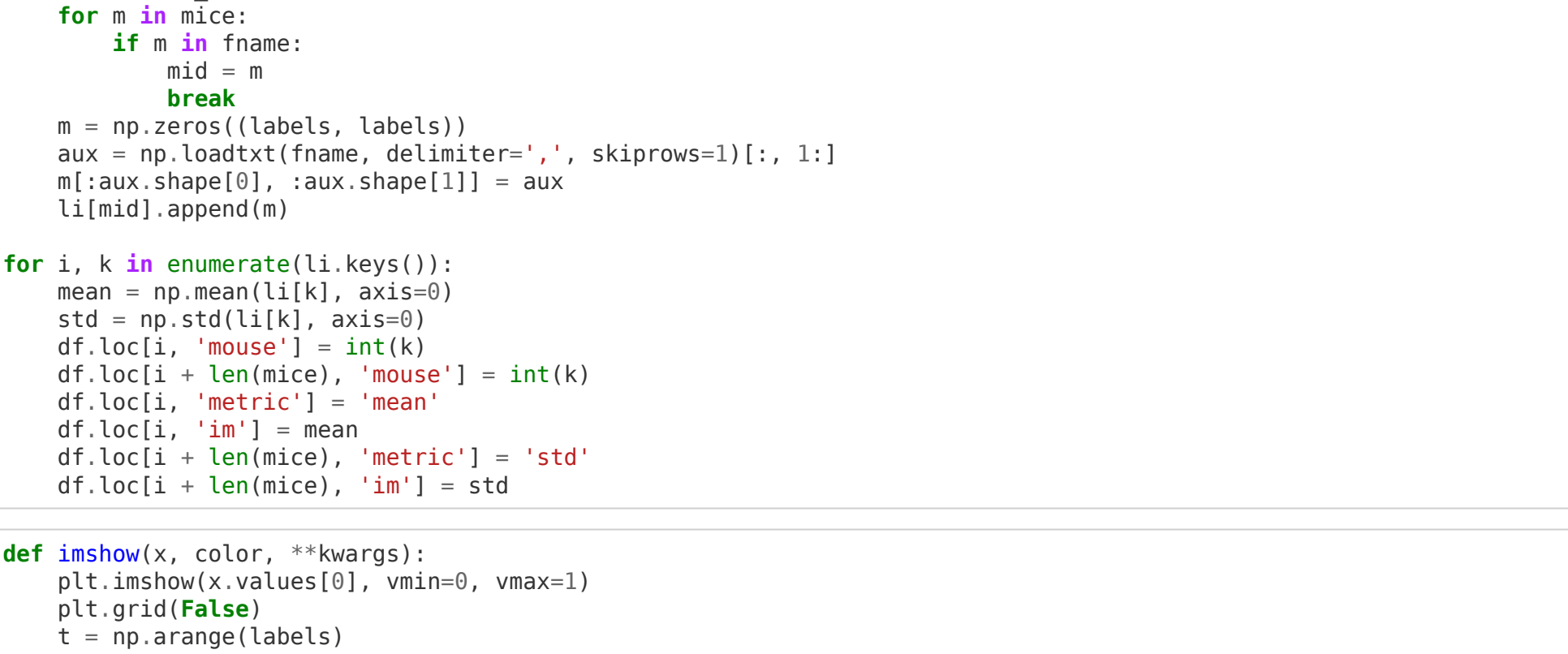


Transition Matrix (Mean and STD over All KP Trials)

In [24]:
```
li = {m: [] for m in mice}
df = pd.DataFrame(columns=['mouse', 'metric', 'im'], index=np.arange(len(mice)*2))

for fname in f_trans:
    for m in mice:
        if m in fname:
            mid = m
            break
    m = np.zeros((labels, labels))
    aux = np.loadtxt(fname, delimiter=',', skiprows=1)[:, 1:]
    m[:aux.shape[0], :aux.shape[1]] = aux
    li[mid].append(m)

for i, k in enumerate(li.keys()):
    mean = np.mean(li[k], axis=0)
    std = np.std(li[k], axis=0)
    df.loc[i, 'mouse'] = int(k)
    df.loc[i + len(mice), 'mouse'] = int(k)
    df.loc[i, 'metric'] = 'mean'
    df.loc[i, 'im'] = mean
    df.loc[i + len(mice), 'metric'] = 'std'
    df.loc[i + len(mice), 'im'] = std
```

In [25]:
```
def imshow(x, color, **kwargs):
    plt.imshow(x.values[0], vmin=0, vmax=1)
    plt.grid(False)
    t = np.arange(labels)
    plt.xticks(t)
    plt.yticks(t)
grid = sns.FacetGrid(df, col='mouse', row='metric')
grid.map(imshow, 'im')
ax = plt.axes((1, 0.25, 0.01, 0.5))
plt.colorbar(cax=ax)
```

Out[25]: <matplotlib.colorbar.Colorbar at 0x7fe767951550>



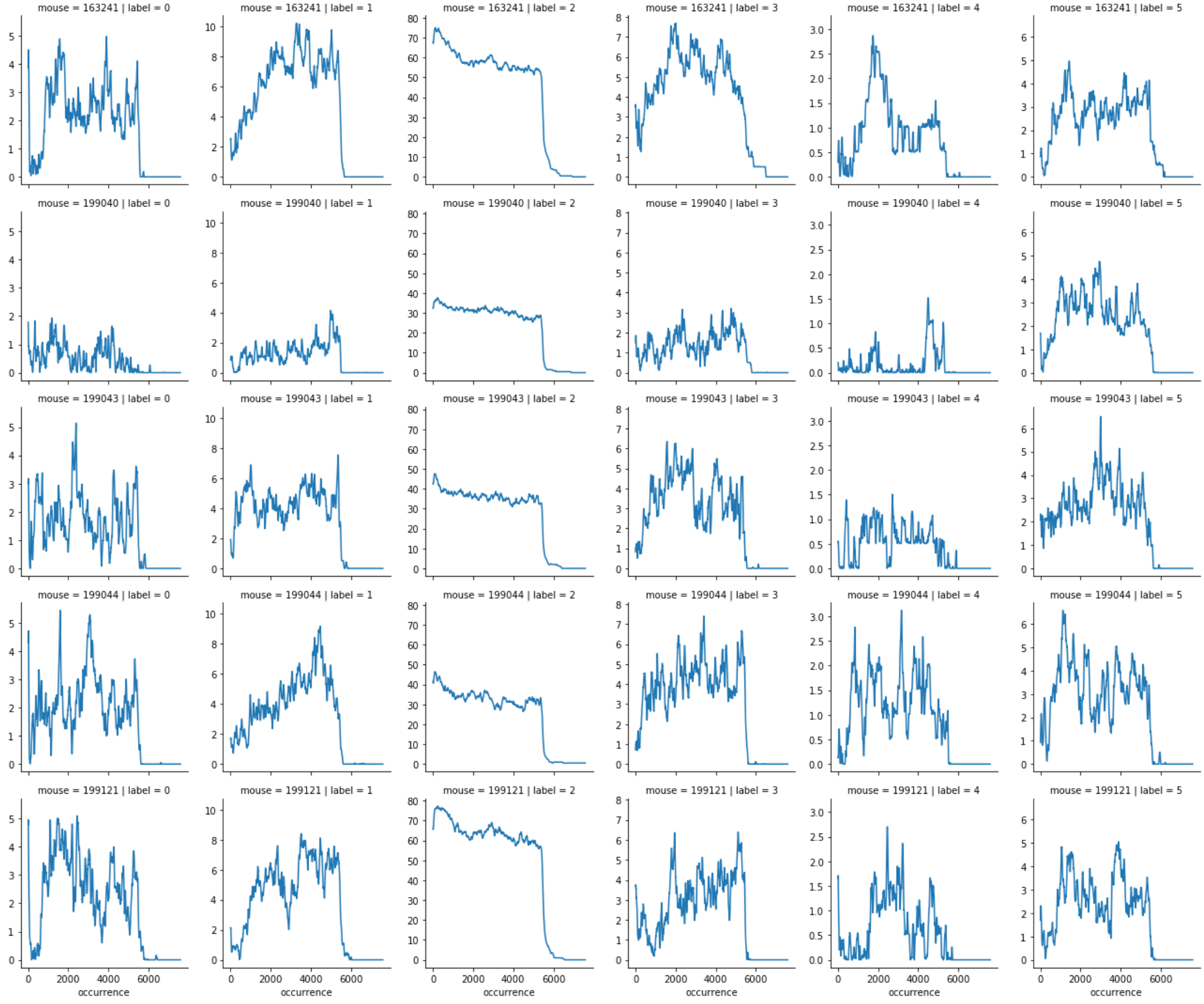
Running Length Distribution for Each Label

In [26]:
```
runlen = np.zeros((len(mice), labels, max(lines)), dtype=np.int32)
for fname in f_len:
    data = np.loadtxt(fname, delimiter=',', skiprows=1, dtype=np.int32, ndmin=2)
    for i, mouse in enumerate(mice):
        if mouse in fname:
            for r in data:
                runlen[i, r[1], r[2]:r[2]+r[3]] += 1
            break
```

In [27]:
```
Gaussian = norm.pdf(np.arange(-20, 21), scale=30)
smooth = lambda x: np.convolve(x, Gaussian, 'valid')

runlen_df = {'mouse': [], 'label': [], 'occurrence': []}
for i in range(runlen.shape[0]):
    for j in range(runlen.shape[1]):
        runlen_df['mouse'].append(mice[i])
        runlen_df['label'].append(j)
        runlen_df['occurrence'].append(smooth(runlen[i, j]))
runlen_df = pd.DataFrame(runlen_df)
```

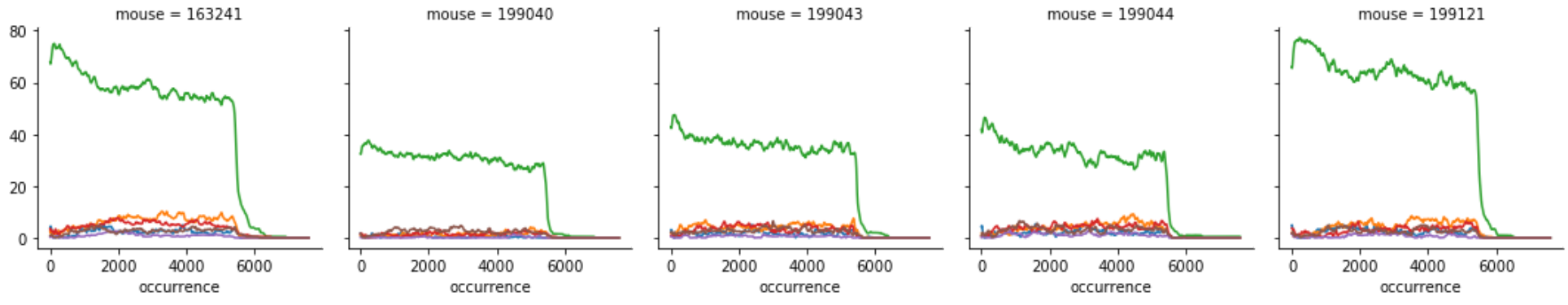
In [28]:
```
grid = sns.FacetGrid(runlen_df, col='label', row='mouse', sharey='col')

def plot(x, **kwargs):
    plt.plot(x.iloc[0])
grid.map(plot, 'occurrence')
```

Out[28]: <seaborn.axisgrid.FacetGrid at 0x7fe7678d4e48>



In [29]:
```
grid = sns.FacetGrid(runlen_df, hue='label', col='mouse')

def plot(x, **kwargs):
    plt.plot(x.iloc[0])
grid.map(plot, 'occurrence')
```

Out[29]: <seaborn.axisgrid.FacetGrid at 0x7fe767183eb8>



Summary of the Training Dataset (AL)

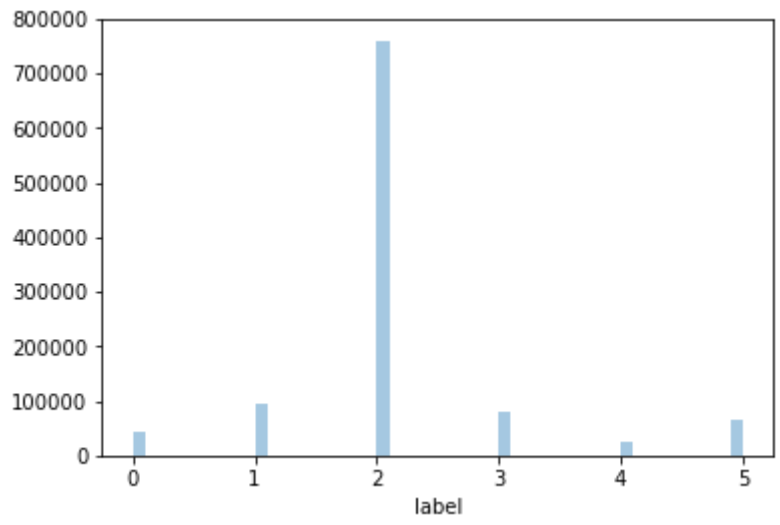
In [31]:
```
df = pd.read_csv(parent + 'bsoid_labelprob_10Hz_20200719_0251.csv', index_col=0)
```

```
/ihome/crc/install/python/anaconda3.7-5.3.1_genomics/lib/python3.7/site-packages/numpy/lib/arraysetops.py:569: FutureWarning: elementwise comparison failed; returning scalar instead, but in the future will perform elementwise comparison
  mask |= (ar1 == a)
```

In [32]:
```
df['maxid'] = df.idxmax(axis=1)
df['maxval'] = df.max(axis=1)
```

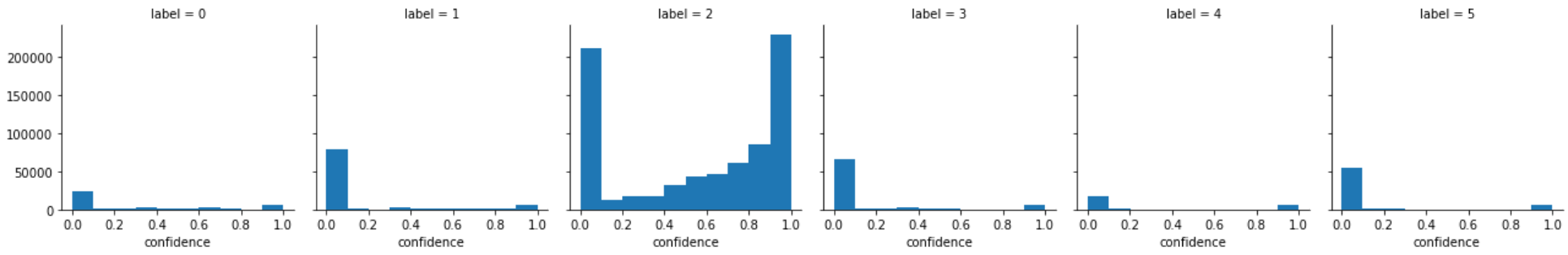
In [35]:
```
sns.distplot(pd.to_numeric(df['maxid']), kde=False)
plt.xlabel('label')
```

Out[35]: Text(0.5,0,'label')



In [36]:
```
grid = sns.FacetGrid(df, col='maxid')
grid.map(plt.hist, 'maxval')
grid.set_axis_labels(x_var='confidence')
grid.set_titles('label = {col_name}')
```

Out[36]: <seaborn.axisgrid.FacetGrid at 0x7fe7665c94a8>



Note on block OUT[36]:
Labels other than 2 have low confidences, as well as low occurrences. Are they the ramifications of low-confidence detection in which some of the body points are far from true positions?