In [1]:
```
import matplotlib
import matplotlib.pyplot as plt
import numpy as np

matplotlib.rcParams['figure.autolayout'] = False
matplotlib.rcParams['font.family'] = "DejaVu Sans"
matplotlib.rcParams['font.serif'] = "Computer Modern"
matplotlib.rcParams['text.usetex'] = True

matplotlib.rcParams['lines.linewidth'] = 3
matplotlib.rcParams['xtick.direction'] = "out"
matplotlib.rcParams['ytick.direction'] = "out"
matplotlib.rcParams['axes.labelsize'] = 16
matplotlib.rcParams['xtick.labelsize'] = 14
matplotlib.rcParams['ytick.labelsize'] = 14
matplotlib.rcParams['font.size'] = 16
matplotlib.rcParams['legend.fontsize'] = 16
matplotlib.rcParams['axes.titlesize'] = 16
matplotlib.rcParams['animation.html'] = "jshtml"
matplotlib.rcParams['figure.autolayout'] = True


cm = 1 / 2.54
```

0 load the data and optimized filters

In [5]:
```
D = np.load('dat.npy', allow_pickle=True).item()

X_ = D['X_']                    # original spikes (before rearrangin
X_sorted = D['X_sorted']        # spike data with rows rearranged us
dat1 = D['dat1']                # postion on the track
FILT = (D['FILT0'], D['FILT1']) # optimized filters
N = X_.shape[0]
```

In [ ]:
```

```

1 obtain the sorting indices

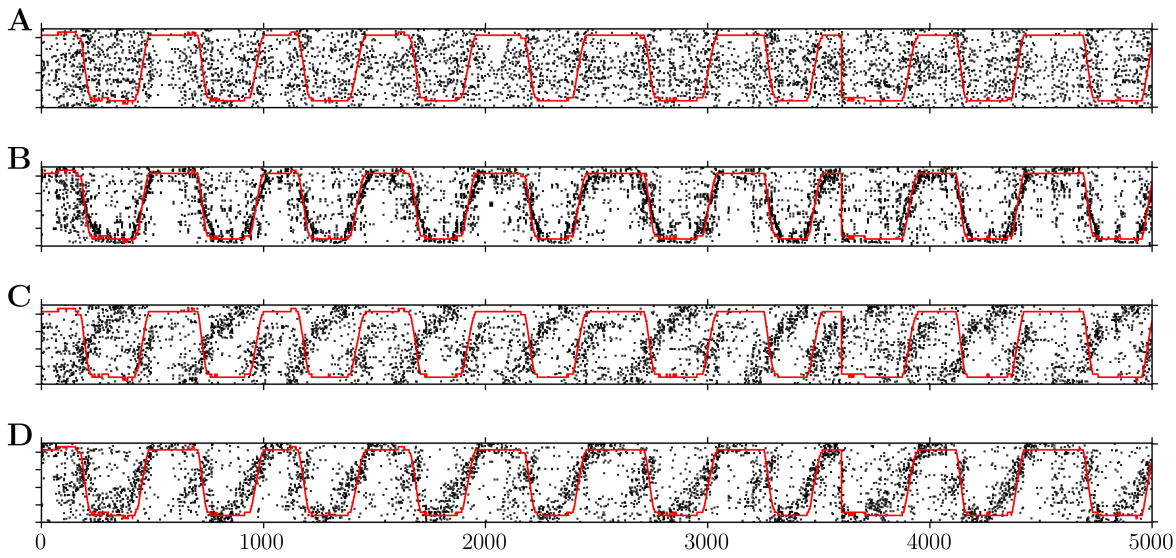
In [6]:
```
f = FILT[0].squeeze()
srt = sorted(list(zip(range(N), f.argmax(axis=1))), key=lambda tup:t
sorting_incices_0 = [i[0] for i in srt]

f = FILT[1].squeeze()
srt = sorted(list(zip(range(N), f.argmax(axis=1))), key=lambda tup:t
sorting_incices_1 = [i[0] for i in srt]
```

In [ ]:
```

```

2 use these indices to rearrange the order of neurons in the dataset

In [7]:

```
# generate fig 7

fig, (ax0, ax0a, ax1, ax2) = plt.subplots(
    4, 1,
    figsize=(26*cm, 13*cm),
    dpi=300,
    gridspec_kw=dict(height_ratios=[2,2,2,2]),
    sharex=True)


ax0.spy(X_, aspect='auto', markersize=0.2, origin='lower', color='k'
ax0a.spy(X_sorted, aspect='auto', markersize=0.2, origin='lower', co

axt = ax0.twinx()
_ = axt.plot(dat1, 'r', lw=1)
axta = ax0a.twinx()
_ = axta.plot(dat1, 'r', lw=1)
axtb = ax1.twinx()
_ = axtb.plot(dat1, 'r', lw=1)
axtc = ax2.twinx()
_ = axtc.plot(dat1, 'r', lw=1)

axt.set_yticklabels([])
axta.set_yticklabels([])
axtb.set_yticklabels([])
axtc.set_yticklabels([])


ax1.spy(X_[sorting_incices_0,:], aspect='auto', markersize=0.2, orig
ax2.spy(X_[sorting_incices_1,:], aspect='auto', markersize=0.2, orig


for a in (ax0, ax0a, ax1, ax2):
    a.set_ylabel(None)
    a.set_yticklabels([])
for a in (ax0, ax0a, ax1, ax2):
    a.set_xlim(0, 5000)
    a.set_xlabel(None)


for i, ax in enumerate((ax0, ax0a, ax1, ax2)):
    panelid = chr(65+i)
    h_adjust = -0.03
    v_adjust = 1.25
    ax.text(h_adjust, v_adjust, f'$\mathbf {panelid}$', transform=ax
    if panelid == 'E':
        ax.set_xlabel('Time steps')
```



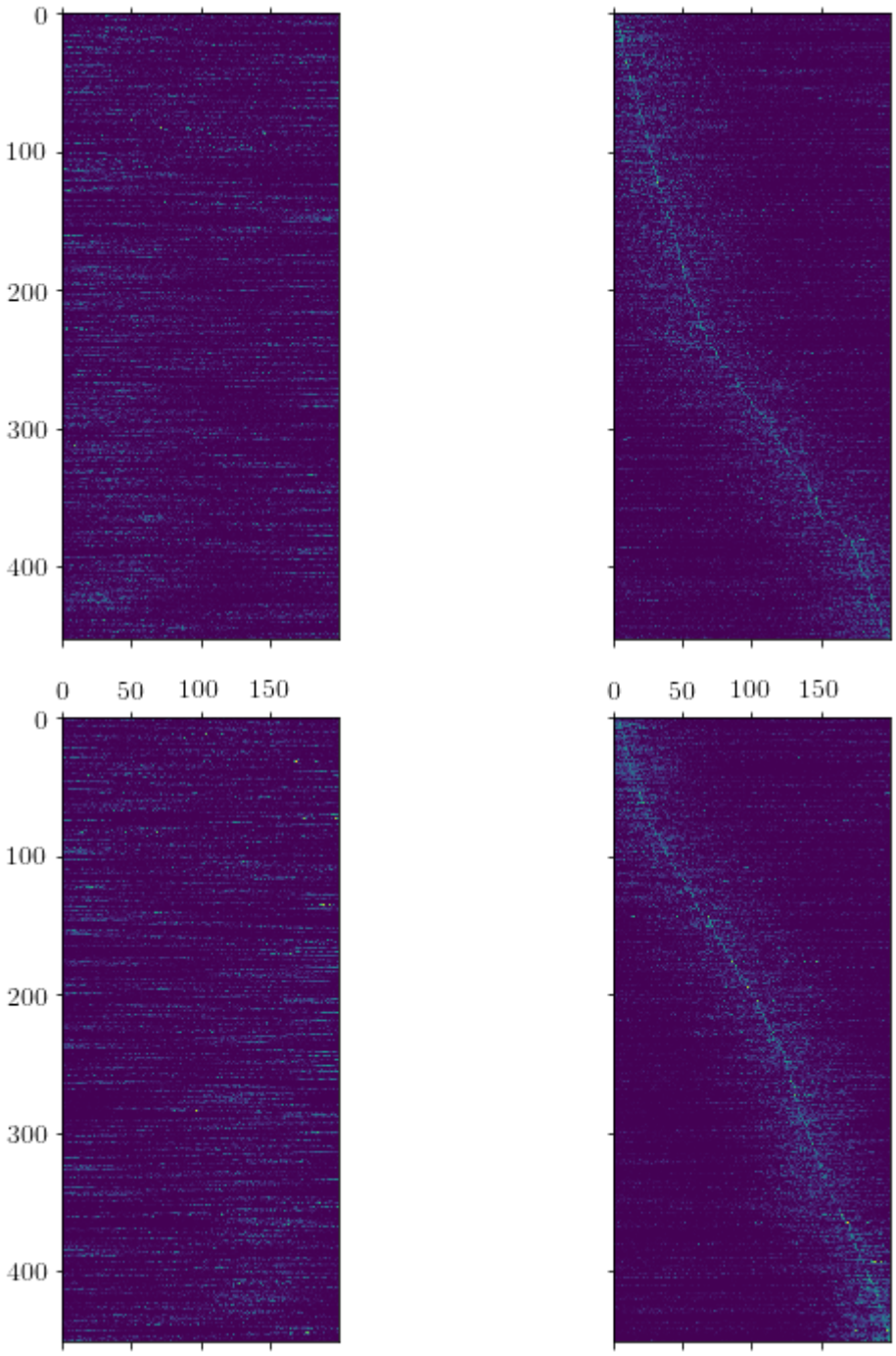
In [ ]:
```
1
```

Unsorted (left) and sorted filters (right).

Top row - filter 0, bottom row - filter 1.

In [18]:
```
fig, ax = plt.subplots(2,2, figsize=(10,10), sharex=True, sharey=Tru
ax[0, 0].matshow(FILT[0])
ax[0, 1].matshow(FILT[0][sorting_incices_0,:])
ax[1, 0].matshow(FILT[1])
ax[1, 1].matshow(FILT[1][sorting_incices_1,:])
```

Out[18]: <matplotlib.figure.AxesImage at 0x7f90165fd590>



In [17]:
```
FILT[0].shape
```

Out[17]: (452, 200)

In [ ]:
```

```