PROSPECT Shared Data

This document is a demonstration of how to use the shared data from the PROSPECT experiment. All data are stored in comma-separated value text files which are readable using `numpy.loadtxt`.

The four main files are:

1. `SpectrumData.txt` - Tabulation of the measured spectrum with statistical uncertainties
2. `ResponseMatrix.txt` - Conversion matrix to change true antineutrino energy into Reconstructed Visible Energy, 200x200, 50keV-wide bins, 0-10MeV on both axes.
3. `CovarianceMatrix.txt` - 32x32 matrix (0.8-7.2MeV E_rec) containing all known correlated and uncorrelated uncertainties
4. `HFIRSpectrumPrediction.txt` - Tabulation of the predicted HFIR spectrum with it's separate components (Huber 235U, NonEqualibrium, 28Al, and 6He)

We will walk through reading and plotting each of these components below.

Setting up the environment

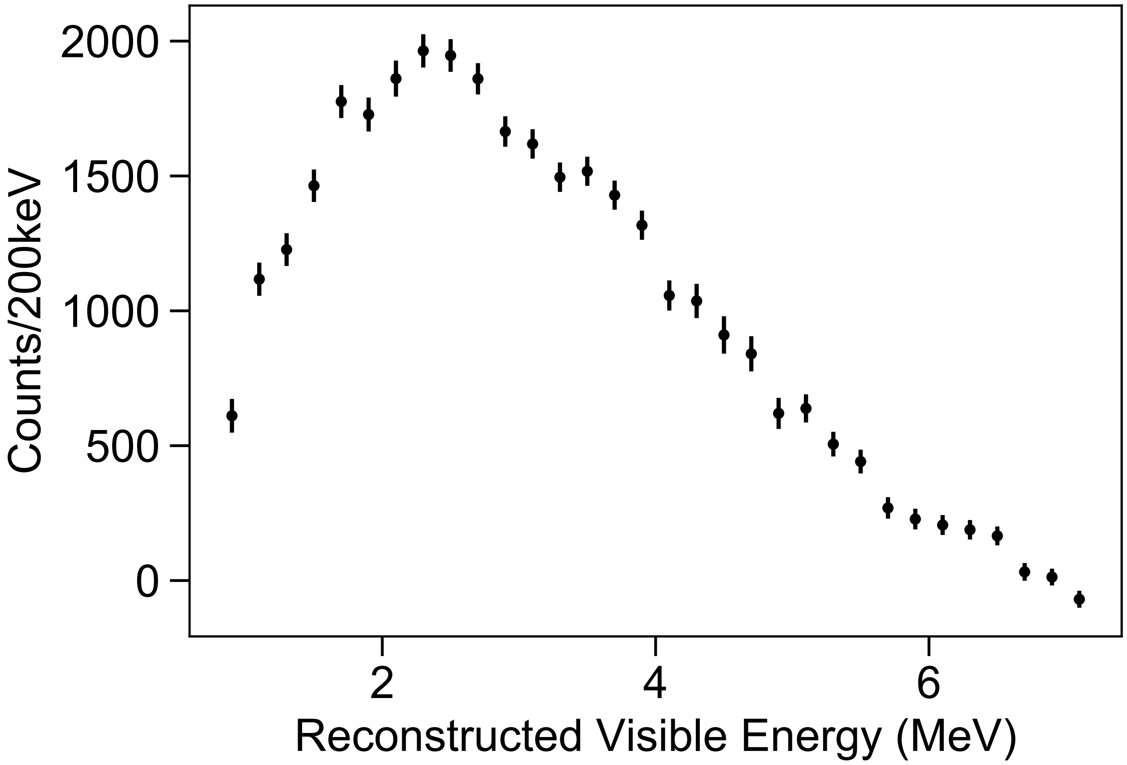
```
%pylab inline
%config InlineBackend.figure_format='svg'

import numpy as np
import matplotlib.pyplot as plt

from matplotlib.colors import LogNorm
```

```
data_dir = '/Users/langford/Yale/PROSPECT/SpectrumPaper/anc/'
```

Measured Spectrum

```
en, spectrum, stat_error = np.loadtxt(f'{data_dir}/SpectrumData.txt',
                                      delimiter=',', unpack=True)
```

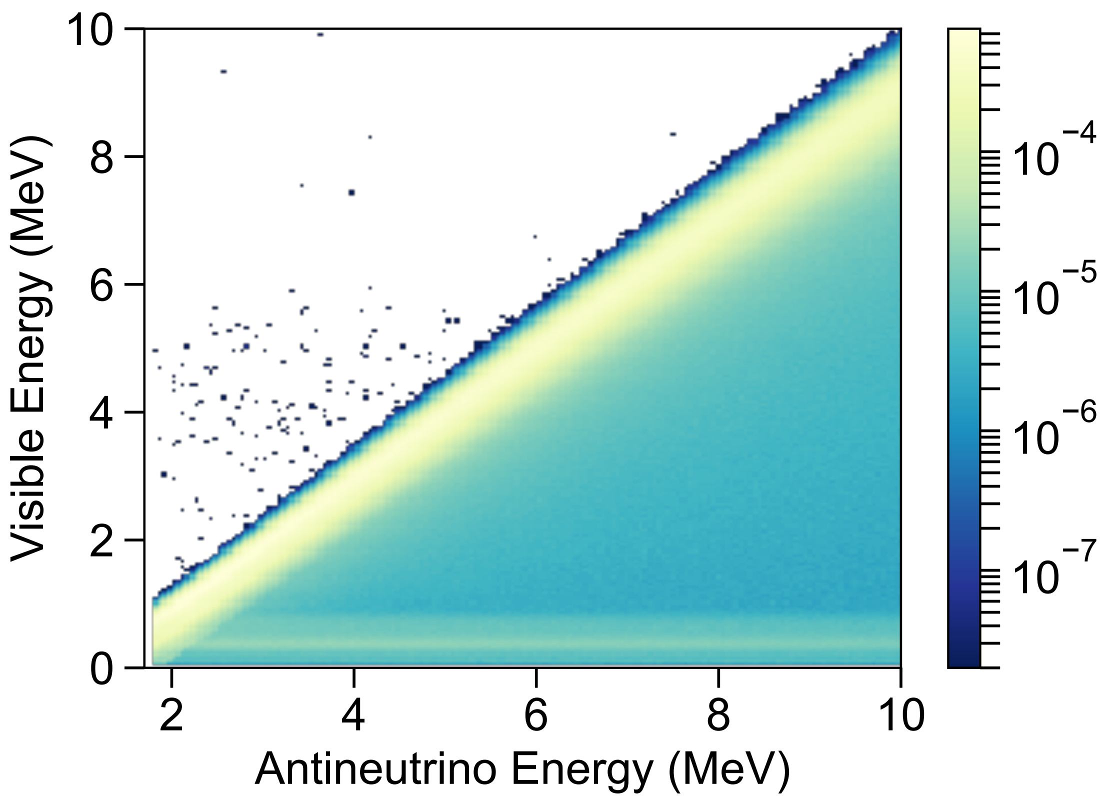
```
plt.errorbar(en, spectrum, yerr=stat_error, linestyle='None', marker='.')
plt.xlabel('Reconstructed Visible Energy (MeV)');
plt.ylabel('Counts/200keV');
```



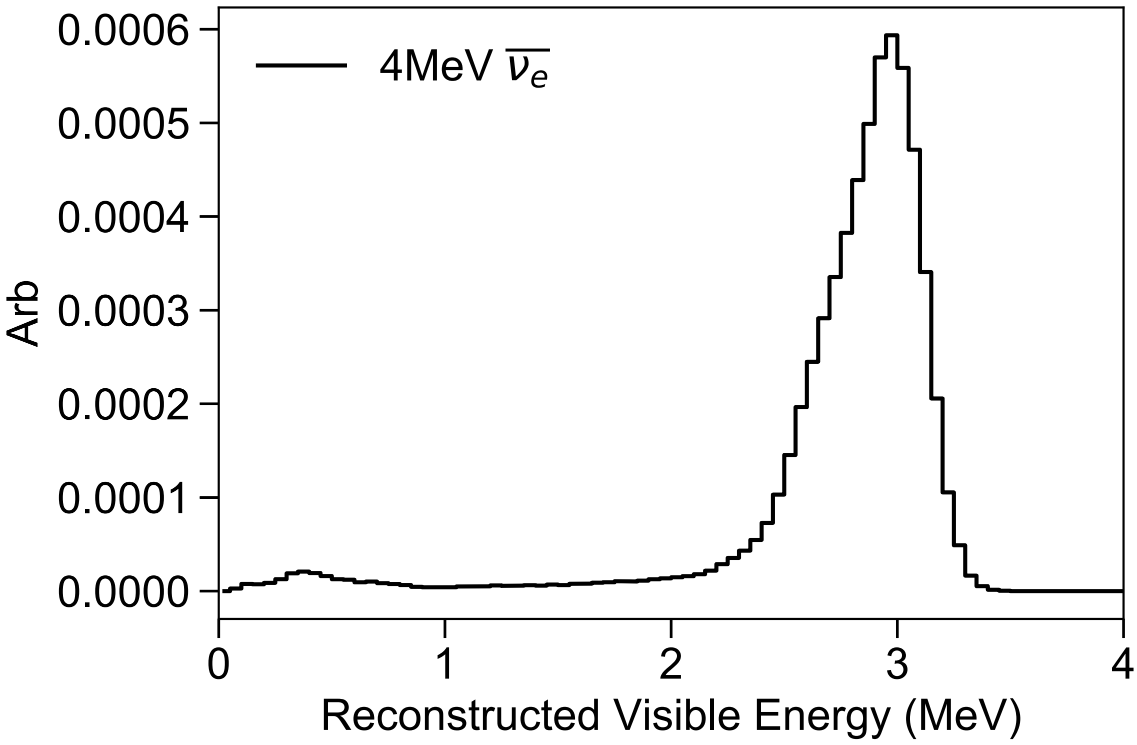
Detector Response Matrix

```
det_resp = np.loadtxt(f'{data_dir}/ResponseMatrix.txt', delimiter=',')
```

```
plt.imshow(det_resp.T, norm=LogNorm(), extent=(0,10,0,10), aspect='auto')

plt.colorbar()
plt.xlabel('Antineutrino Energy (MeV)');
plt.ylabel('Visible Energy (MeV)');
plt.xlim(1.7,);
```



```
en = np.linspace(0,10,200,endpoint=False)
en += 0.025
plt.step(en, det_resp[80], where='mid', label=r'4MeV $\overline{\nu_{e}}$')
plt.xlabel('Reconstructed Visible Energy (MeV)');
plt.ylabel('Arb');
plt.legend();
plt.xlim(0,4);
```

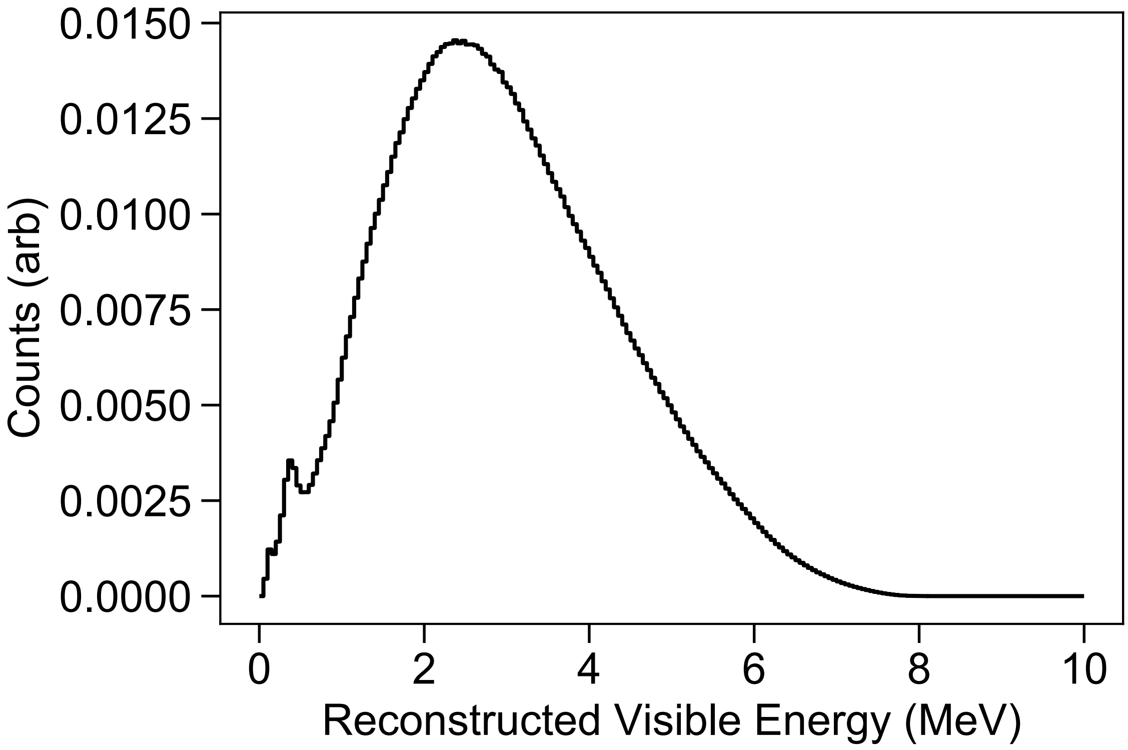


Response To Huber Model

```
huber = np.load('/Users/langford/PycharmProjects/PROST/input_models/huber_235U.npy')
```

```
huber_spec = np.matmul(huber, det_resp)
huber_spec = np.divide(huber_spec, np.sum(huber_spec))
```

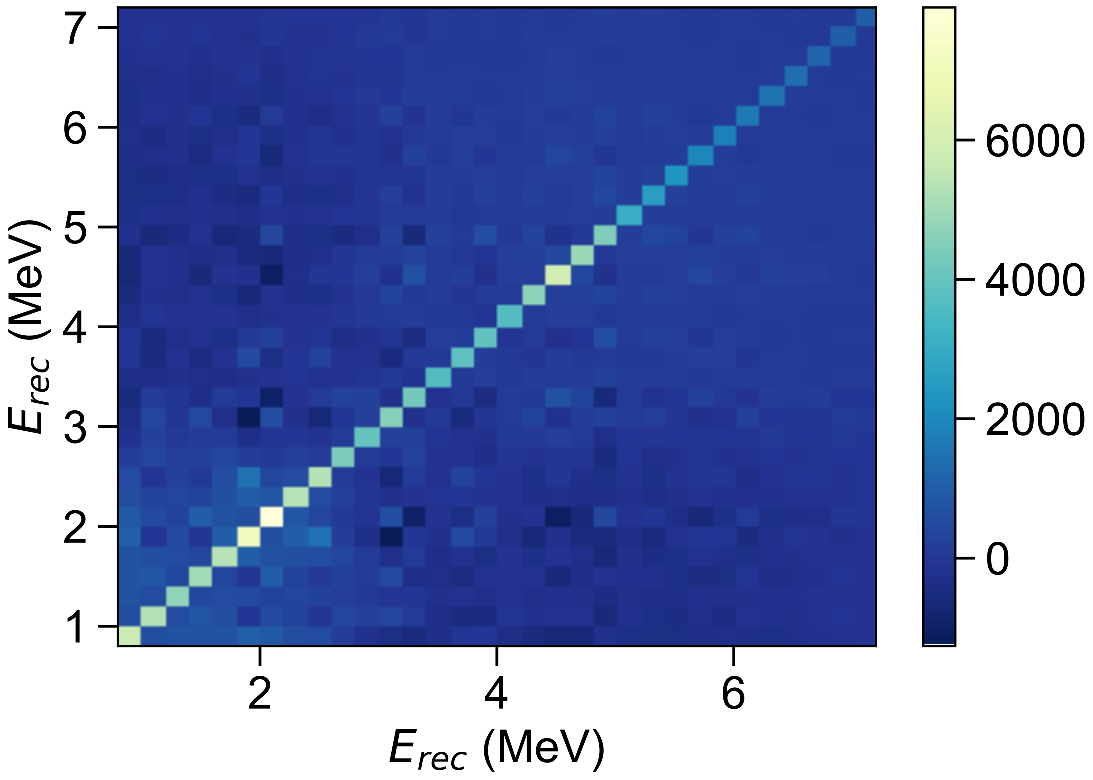
```
en = np.linspace(0,10,200,endpoint=False)
en += 0.025
plt.step(en, huber_spec, where='mid')

plt.xlabel('Reconstructed Visible Energy (MeV)');
plt.ylabel('Counts (arb)');
```



Covariance Matrix

```
cov_mat = np.loadtxt(f'{data_dir}CovarianceMatrix.txt', delimiter=',')
```

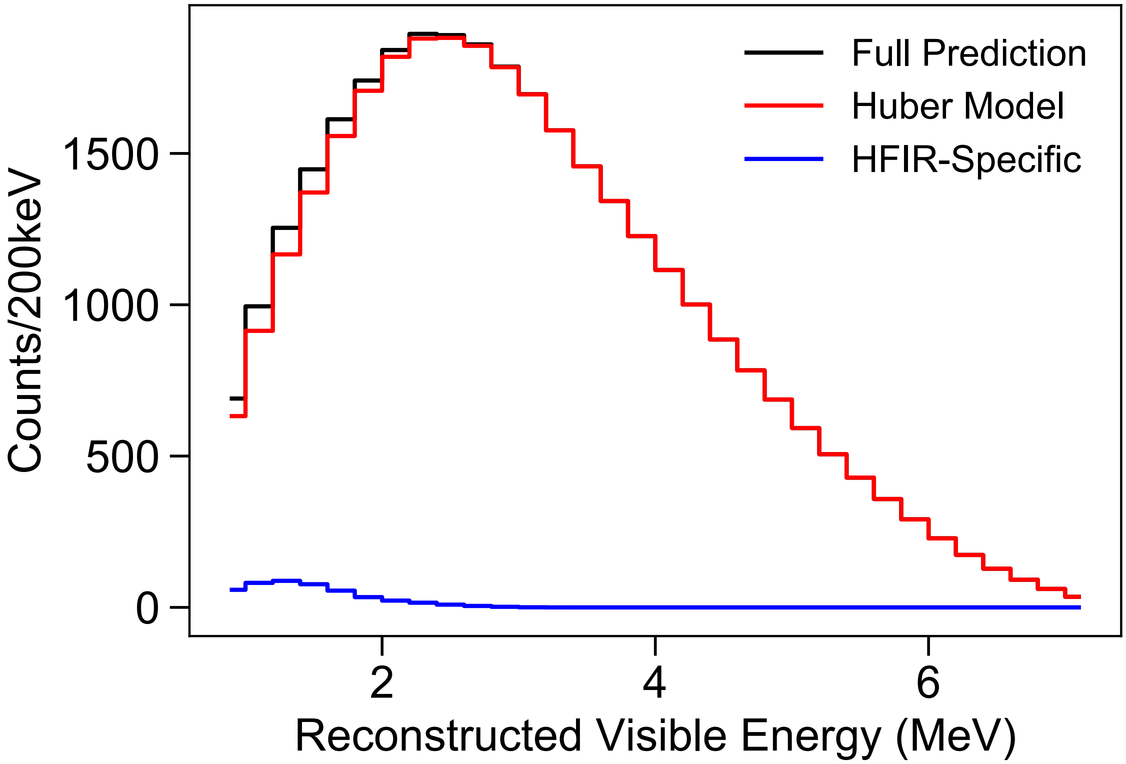
```
plt.imshow(cov_mat, extent=(0.8,7.2,0.8,7.2))
plt.colorbar()
plt.xlabel(r'$E_{rec}$ (MeV)');
plt.ylabel(r'$E_{rec}$ (MeV)');
```



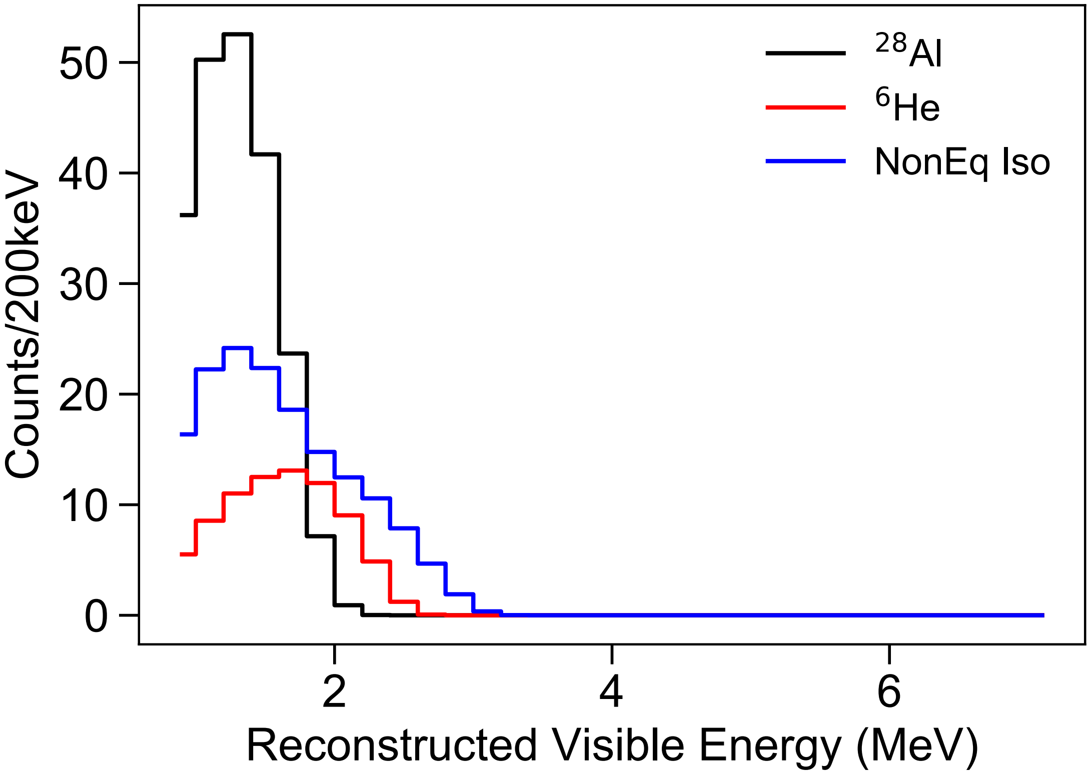
HFIR Specific Corrections

```
en, hfir, huber, al, he, noneq = np.loadtxt(f'{data_dir}/HFIRSpectrumPrediction.txt',
                                            delimiter=',', unpack=True)
```

```
plt.step(en, hfir, where='mid', label='Full Prediction')
plt.step(en, huber, where='mid', label='Huber Model')
plt.step(en, al+he+noneq, where='mid', label=r'HFIR-Specific')

plt.legend(fontsize='small');
plt.xlabel('Reconstructed Visible Energy (MeV)');
plt.ylabel('Counts/200keV');
```



```
plt.step(en, al, where='mid', label=r'$^{28}$Al')
plt.step(en, he, where='mid', label='$^{6}$He')
plt.step(en, noneq, where='mid', label='NonEq Iso')

plt.legend(fontsize='small');
plt.xlabel('Reconstructed Visible Energy (MeV)');
plt.ylabel('Counts/200keV');
```


^{28}Al
^{6}He
NonEq Iso
Counts/200keV
0
10
20
30
40
50
2
4
6
Reconstructed Visible Energy (MeV)