

# Temperature Map Pipeline

Carter Rhea

---

## 1. Temperature Maps

Before we can get to the temperature map (I promise I'm getting there...), we first have to actually use the data we got from the Weighted Voronoi Tessellation Algorithm; we need the pixels binning information.

### 5 1.1. Intermediate Step: Binning the Data

As this subsection's title suggests (look up), we must first go through the process of binning the data (our fits file) based of the recommendation of our WVT algorithm.

Instead of describing this in great detail, I will simply include an algorithm  
10 outlining the process and point the reader toward the python file (*Bin\_data.py*)  
for more in depth analysis.

**Data:** WVT Bins and Fits File

**Result:** Binned PHA files

initialization -- > read in WVT bin information;

```
for bin do
    for pixel in bin do
        Create fits file for pixel;
        Generate PI/PHA file;
    end
    Combine Pixel's PI/PHA files;
end
```

**Algorithm 1:** Binning Algorithm

Clearly the implementation is a little more complicated and uses several *CIAO* tools such as *specextract* and *dmextract*. For more information on these  
15 wonderful tools, check out the *CIAO* website:

*http://cxc.harvard.edu/ciao/*

## 1.2. Generating Temperature Maps

We can now (FINALLY) use our combined pixels to generate a temperature map assuming you have already created a binning of some sort – I would suggest my Weighted Voronoi Tessellations algorithm<sup>1</sup>. We will be employing *XSPEC*<sup>2</sup>  
20 for our modeling needs.

There are two main types of models:

- Additive: Source of emission i.e. producer of photons
- Multiplicative: Modifies spectrum i.e. acts on photons

Let's examine the major componenets of our model...

### 25 1.2.1. *zpow*

**zpow** is a simple photon power law taking into account redshift. This is just classic blackbody goodness...

$$A(E) = K[E(1+z)]^{-\alpha} \quad (1)$$

where  $\alpha$  is the dimensionless photon index of the power law,  $z$  is our objects redshift, and  $K$  is a normalization factor at  $1\text{keV}$  in units of *photons/keV/cm<sup>2</sup>*.  
30 Photon indices of AGN are typically between  $1.5 - 2.5$ ; this value needs to be determined from the literature.

Its worth noting that **zpow** is an additive model.

---

<sup>1</sup><https://github.com/crhea93/WVT>

<sup>2</sup><https://heasarc.gsfc.nasa.gov/xanadu/xspec/>

### 1.2.2. *zphabs*

**zphabs** is a model for hydrogen column density. Hydrogen column density  
35 is the "number of units of matter observed along a line of sight that has an area  
of observation"<sup>3</sup>. Basically, it takes into account all the stuff in between us and  
the object we are looking at (though it could be missing things such as a warm  
absorber like a galactic cluster which would require another model component  
– wabs in this instance). **zphabs** has a very simple equation:

$$M(E) = e^{-n_H * \sigma(E(1+z))} \quad (2)$$

40 where  $n_H$  is the equivalent column density in  $10^{22} \text{ atoms cm}^{-2}$ ,  $\sigma(E)$  is the  
electron cross section – not Thompson – where  $z$  is the redshift.

All we specify are  $n_H$  and  $z$ .

**zphabs** is a multiplicative model

### 1.2.3. *apec*

45 **apec** is a wonderful model to handle the emission spectrum of a collisionally-  
drive optically-thin plasma (think AGN). Hence we use it to constrain the tem-  
perature of the object. For real details please check out the following website:

*[http : //www.atomdb.org/physics.php](http://www.atomdb.org/physics.php)*

The most important thing to know is that it does in fact model thermal  
bremsstrahlung.

50 We need only specify  $T$  (KeV), as a guess, and  $z$  which is our redshift.

**Apec** is a multiplicative model

### 1.2.4. $\chi^2$ statistic

The  $\chi^2$  is defined as the following:

$$\chi^2 = \sum_{i=1}^{N_{el}} \left( \frac{x_i - \mu_i}{\sigma_i} \right)^2 \quad (3)$$

---

<sup>3</sup>[https://en.wikiversity.org/wiki/Column\\_densities](https://en.wikiversity.org/wiki/Column_densities)

Where  $x_i$  is the data value at point  $i$ ,  $\sigma_i$  is the corresponding error, and  $\mu_i$  is  
55 the model value.

By calculating the  $\chi^2$  we can get a decent guess on how well our fit is. Optimally, our  $\chi^2$  will equal the number of **degrees of freedom**. We call this ratio the **Reduced Chi-Squared Statistic**. Hence we can very easily calculate the reduced chi-square if we know the number of degrees of freedom  
60 for a model:

$$\chi_{red}^2 = \frac{\chi^2}{N_{dof}} \quad (4)$$

So an optimal fit using a reduced  $\chi^2$ -statistic means  $\chi_{red}^2 = 1$ . Hence for our algorithm, we will try to hit a  $\chi_{red}^2 = 1$  ... or as close as we can get ...

#### 1.2.5. *C-statistic*

For low counts (in my opinion most X-ray observations are relatively starved  
65 in the photon department), we should instead use the *Cash-statistic*, *c-stat*, beacuse at low counts the variance approaches 0 and thus the  $\chi^2$ -*statistic* fails. We define the *c-stat* in the following manner:

$$c = 2 \sum_{i=1}^n s_i - N_i + N_i \ln(N_i/s_i) \quad (5)$$

where  $s$  is the number of counts according to the model,  $N_i$  is the number of observed counts, and  $n$  is the number of bins. In addition, we are able to  
70 calculate a “goodness of fit” for this statistic,  $C_R$ :

$$C_R = \frac{C}{n - p} \quad (6)$$

where  $p$  represents the number of free parameters.

In XSPEC we simply need to set `fit.statistic = ‘cstat’`.

## 2. The Pipeline

**Data:** Binning Map

**Result:** Temperature Map and Graphic

Step 1: Create binned spectra using *binned\_spectra.py* ;

Step 2: Run XSPEC Model on binned data by "simply"<sup>a</sup> running

*XSPEC.py* ;

**Algorithm 2:** Temperature Map Pipeline

---

<sup>a</sup>Like fitting a model is simple..... But seriously this is where a mistake could be made resulting in incorrect Temperature Profiles so play around with the model on a *single* observation to ensure it is a decent fit...

75     The algorithm itself is fairly succinct because all of the mechanics are intentionally under the hood. For instance, we actually create what I deem as super-pixels in the creation of the spectra for each bin since **specextract** runs faster with less regions. The super-pixel algorithm groups as many pixels together to create a box-region to reduce computational time<sup>4</sup>

---

<sup>4</sup>The mechanics for doing this are well commented in the *binned\_spectra.py* file.

### 80 3. Dealing with Merged Data Set

How do we create temperature maps for merged data sets? As documented on the CIAO site <sup>5</sup>, we are not able to create spectra from merged observations. Thus we have to be very careful in the creation of temperature maps for merged data sets.

85 The basic steps are as follows:

1. Merge Observations (Don't worry! We are just going to use the event file and exposure map).
2. Create a bin map using your favorite binning algorithm; I suggest using Weighted Voronoi Tessellations.
- 90 3. For each individual observation do the following:
  - (a) Create individual event files for each bin
  - (b) Extract spectrum for each bin
4. Run fitting software on each bin by fitting all datasets simultaneously

In following these steps, we are able to reconstruct a temperature map for  
95 a merged observation while avoiding the inherent drawbacks of merging data.

---

<sup>5</sup>[http://cxc.harvard.edu/ciao/caveats/merged\\_events.html](http://cxc.harvard.edu/ciao/caveats/merged_events.html)