# Supernova Bootcamp

Version 0.0.0

Lauren Aldoroty

July 6, 2022

# Contents

# 1 Introduction

Compiling all the things I've learned so it's easier for future folks. This manual will attempt to cover more technical, data-analysis driven perspectives on working with SNe, particularly SNe Ia. In the Software section (Section 6), at the end of each subsection, I'll list some lesser-known functions and tricks that I've found useful. These will be indicated by section titles `in font like this`, e.g., `helio_to_cmb()` (Section 6.2.1).

# 2 Reddening and Extinction

Reddening and extinction are *nearly* the same thing, but not quite. They're both caused by light scattering off dust, but *extinction* describes the dimming that results, and *reddening* describes the preferential scattering of blue light (i.e., more shorter wavelength light gets scattered away from us than longer wavelengths, so we see objects as redder than they actually are.) This is important because it's a pretty large source of systematic uncertainty in supernova studies! Dust doesn't emit light, it just takes it away, so we can't "see" it directly. *HOW* are we supposed to correct for an effect we know so little about?

## 2.1 Supernova Intrinsic Color

## 2.2 Correcting for Dust

# 3 Hubble's Law and SN Ia Cosmology

## 3.1 What the heck is a rest frame?

Let's say you have some data, but it's from a supernova with $z = 0.1$. Because of the relativistic Doppler effect, the wavelengths you *observe* are different from the "actual" wavelengths, i.e., rest frame wavelengths. This will also affect observed flux. Ignoring ejecta velocity (see CHAPTER FOR THIS), this is why a spectral line won't appear in its "expected", or rest frame, position. Usually, you want to put your spectra in the rest frame before doing anything with them because it's hard to compare SNe at different redshifts. See Section 5.0.1 for details on correcting SN spectra to the rest frame.

## 3.2 $z_{helio}$ and $z_{CMB}$

You'll encounter two kinds of redshift—heliocentric redshift ($z_{helio}$) and CMB (Cosmic Microwave Background) redshift ($z_{CMB}$). *These are different, and* *are used for different purposes!* $z_{CMB}$ is the redshift caused by the Universe's expansion *only*, i.e., where the reference frame is the CMB frame. $z_{helio}$ is the redshift with *only* the Earth's rotation and orbit removed. There are still effects from other motion, like Galactic rotation and the Galaxy moving around with respect to other objects, as well as $z_{CMB}$. "Heliocentric" $\equiv$ "Sun at center", so "Sun at center" is the rest frame. It's how things are moving relative to the Milky Way. Note that $z_{helio}$, while often reported as an object's redshift, is *not* the observed redshfit. The observed redshift does not have the Earth's rotation and orbit removed. In other words, $z_{helio}$ contains information about a bunch of things moving relative to each other, including motion relative to the CMB, while $z_{CMB}$ contains information about *only* motion relative to the CMB.

For cosmology, you'll use $z_{CMB}$ (e.g., when making a Hubble diagram or determining other cosmological parameters). If you're dealing with data that you need to correct to the rest frame, you probably want to use $z_{helio}$.

# 4 Photometry

# 5 Spectra

### 5.0.1 Correcting to the rest frame

Correcting the observed wavelength for the relativistic Doppler effect is done by:

$$\lambda_{rest\ frame} = \frac{\lambda_{observed}}{1 + z_{helio}}. \tag{1}$$

Correcting the observed flux is similar:

$$f_{rest\ frame} = \frac{f_{observed}}{(1 + z_{helio})^2}. \tag{2}$$

Why? Because redshift affects wavelength, it also affects frequency, i.e., photon energy. You get a two-fold effect here—one factor of $(1 + z)$ is from the wavelength increasing, and the other is from the intensity decreasing.

### 5.0.2 Normalizing to a particular $z$

## 5.1 Absorption lines

## 5.2 Ejecta velocity

## 5.3 Synthetic photometry overview

If you're here, you're wondering how to make photometry from some spectra. In short, this is obtained by integrating under the spectrum in the region covered by a given filter.

The area under the spectrum, $F$, in a given filter $X$, is calculated by

$$F_X = \int_{\lambda_1}^{\lambda_2} f(\lambda) R_X(\lambda) d\lambda, \tag{3}$$

where $f(\lambda)$ is your spectrum, and $R_X(\lambda)$ is your filter. However, we can't often use the continuous version of this function with our data. So, we discretize it:

$$F_X = \sum_{i=\lambda_1}^{\lambda_2} f(\lambda_i) R_X(\lambda_i)(\lambda_i - \lambda_{i-1}), \tag{4}$$

where $f(\lambda_i)$ is the energy flux at a particular wavelength $\lambda_i$, and $R_X(\lambda_i)$ is the response function for the filter at wavelength $\lambda_i$. We're using a discretized version of an integral, so $(\lambda_i - \lambda_{i-1})$ is the same as $d\lambda$. From now on, this chapter will be written as if we are using Equation 4.

Then, the variance of $F$ is:

$$\sigma_F^2 = \sum_{i=\lambda_1}^{\lambda_2} \sigma_{f(\lambda_i)}^2 R_X(\lambda_i)^2 (\lambda_i - \lambda_{i-1})^2 \tag{5}$$

Individual magnitudes are calculated by

$$m = -2.5 \log_{10} \left( \frac{F}{F_{ref}} \right) + m_{ref}. \tag{6}$$

*Wait—what are $F_{ref}$ and $m_{ref}$?* Good question. Because magnitudes are inherently relative quantities, we need to use some reference object to convert flux into magnitudes. You can get reference objects from CALSPEC. Photometric systems are, unfortunately, out of the scope of this manual at this time. Let's say we're using the Vega system, with the star Vega (Alpha Lyrae) from now on.

This means we need to use Equation 4 on our reference object, Vega to calculate $F_{ref}$. You'll use the *same* response function as you use for your supernova spectrum. This means that $m_{ref}$ is the... reference magnitude? What does THAT mean? If magnitudes are inherently relative quantities, do we then compare our reference object to something *else*? A valid concern, but no! This is called a *zero point*. For the Vega system, astronomers have defined the magnitude of the star Vega such that the $m_{Vega} = 0$ in all bands. So, if you're using the Vega system, $m_{ref}$ is probably just 0 (unless some literature tells you otherwise, which is possible, so make sure you're familiar with the photometric system and filters you're using).

So anyway, the variance of $m$ is:

$$\sigma_m^2 = \frac{\partial m}{\partial f}^2 \sigma_f^2 = \frac{\partial m}{\partial F}^2 \left( \sum_{i=\lambda_1}^{\lambda_2} \frac{\partial F}{\partial f(\lambda_i)} \sigma_{f(\lambda_i)} \right)^2 = \left( \frac{-2.5}{F \ln 10} \right)^2 \sum_{i=\lambda_1}^{\lambda_2} \sigma_{f(\lambda_i)}^2 R_X(\lambda_i)^2 (\lambda_i - \lambda_{i-1})^2 \tag{7}$$

If you're looking for *colors*, I've got your back there, too. For arbitrary color $X - Y$,

$$m_X - m_Y = -2.5 \log \left( \frac{F_X}{F_{ref,X}} \right) + m_{ref,X} + 2.5 \log \left( \frac{F_Y}{F_{ref,Y}} \right) - m_{ref,Y} \tag{8}$$

. We can propagate the error here, too. If you want, you can assume $X$ and $Y$ are independent, and do $\sigma_{X-Y} = \sqrt{\sigma_X^2 + \sigma_Y^2}$. However, we can be more rigorous and not assume independence. We're going to use $\sigma^2 = \mathbf{JCJ^T}$—the explanation of this formula is beyond the scope of this manual. $\mathbf{J}$ is the Jacobian of your spectrum (remember, your spectrum is a function!), and $\mathbf{C}$ is its covariance matrix. The following method will work *if you have an error for the flux in your data spectrum*:

For ease of calculations, we will treat $m_X - m_Y$ as a function of $f(\lambda_i)$ (our supernova spectrum, in discrete wavelength chunks). Then, for arbitrary color $X - Y$, the Jacobian is:

$$\mathbf{J}_{X-Y} = \begin{bmatrix} \frac{\partial m_{X-Y}}{\partial f(\lambda_0)} & \frac{\partial m_{X-Y}}{\partial f(\lambda_1)} & \cdots & \frac{\partial m_{X-Y}}{\partial f(\lambda_N)} \end{bmatrix} \tag{9}$$

where $N$ is the last measured wavelength in the spectrum. The $i$th entry is:

$$\frac{\partial m_{X-Y}}{\partial f(\lambda_i)} = -\frac{1.09}{F_X} R_X(\lambda_i)(\lambda_i - \lambda_{i-1}) + \frac{1.09}{F_Y} R_Y(\lambda_i)(\lambda_i - \lambda_{i-1}) \tag{10}$$

Then, the covariance matrix is diagonal, and each entry is the spectrum error provided in the data:

$$\mathbf{C_{X-Y}} = \begin{bmatrix} \sigma_{f(\lambda_0)}^2 & & & \\ & \sigma_{f(\lambda_1)}^2 & & \\ & & \ddots & \\ & & & \sigma_{f(\lambda_N)}^2 \end{bmatrix}. \tag{11}$$

Now, we use $\sigma^2 = \mathbf{JCJ^T}$, and boom, we have color error without assuming independence for the filters.

## 5.4 Things to watch out for

### 5.4.1 Response function units

Your response function will likely be normalized, but may be given in either normalized flux units or normalized photon counts. You need to know which units you have. If you need the other, don't fret—you can convert it to the other unit system. Let's say you have a response function in normalized photon units, but want it in normalized ergs.

$$R_{X,ergs}(\lambda) = hcR_{X,\gamma}(\lambda), \tag{12}$$

where $h \approx 6.626 \times 10^{-27}$ is Planck's constant in ergs $\cdot$ s and $c = 3 \times 10^{10}$ is the speed of light in cm/s. Then, normalize:

$$R_{X,ergs,normalized}(\lambda) = \frac{R_{X,ergs}(\lambda)}{\max\left(R_{X,ergs}(\lambda)\right)} \tag{13}$$

Now, $R_{X,ergs,normalized}(\lambda)$ is a function with range 0 to 1, and you can plug it into Equation 4.

### 5.5 Step-by-step Explain Like I'm 5

## 6 Software

### 6.1 SALT

ADD SALT CITATIONS. ALL PAPERS. ALL OF THEM. AND WRITE WHAT THEY ARE

### 6.2 sncosmo

sncosmo has a whole bunch of contributors. It's a Python package, useful for middle steps of supernova cosmology (e.g., fitting models).

#### 6.2.1 `helio_to_cmb()`

The below code is handy for converting your heliocentric redshift to CMB redshift. Honestly, I can't find its location in the sncosmo github right now, so I've copied and pasted it below: RUN THIS TO CHECK IT

```
    import math
    from astropy.coordinates import SkyCoord

    # From sncosmo:
def radec_to_xyz(ra, dec):
    # SUPERNOVA BOOTCAMP MANUAL AUTHOR ADDITION:
        Modified to add the try/except statement
    try:
```

```
        x = math.cos(np.deg2rad(dec))
            * math.cos(np.deg2rad(ra))
        y = math.cos(np.deg2rad(dec))
            * math.sin(np.deg2rad(ra))
        z = math.sin(np.deg2rad(dec))
    except:
        coord = SkyCoord('%s %s' % (ra, dec), unit=(u.hourangle,u.deg))
        x = math.cos(np.deg2rad(coord.dec.degree))
            * math.cos(np.deg2rad(coord.ra.degree))
        y = math.cos(np.deg2rad(coord.dec.degree))
            * math.sin(np.deg2rad(coord.ra.degree))
        z = math.sin(np.deg2rad(coord.dec.degree))

    return np.array([x, y, z], dtype=np.float64)

def cmb_dz(ra, dec):
    # See http://arxiv.org/pdf/astro-ph/9609034
    CMBcoordsRA = 167.98750000 # J2000
    CMBcoordsDEC = -7.22000000

    # J2000 coords from NED\n",
    CMB_DZ = 371000. / 299792458.
    CMB_RA = 168.01190437
    CMB_DEC = -6.98296811
    CMB_XYZ = radec_to_xyz(CMB_RA, CMB_DEC)
    coords_xyz = radec_to_xyz(ra, dec)
    dz = CMB_DZ * np.dot(CMB_XYZ, coords_xyz)

    return dz

def helio_to_cmb(z, ra, dec):
    # Convert from heliocentric redshift to CMB-frame redshift.
    "    Parameters\n",
    "    ----------\n",
    "    z : float\n",
    "        Heliocentric redshift.\n",
    "    ra, dec: float\n",
    "        RA and Declination in degrees (J2000).\n",
    "    \"\"\"\n",

    dz = -cmb_dz(ra, dec)
```

```
one_plus_z_pec = math.sqrt((1. + dz) / (1. - dz))
one_plus_z_CMB = (1. + z) / one_plus_z_pec

return one_plus_z_CMB - 1.
```

## 6.3 SNooPy

### 6.3.1 What does it do?

SNooPy, written in Python by Chris Burns for the Carnegie Supernova Project, has a lot of handy functions. CITE.
Below, I'll list some of my favorite functions that are not-so-discussed in the documentation.

### 6.3.2 `get_dust_RADEC()` and `get_dust_sigma_RADEC()`

`get_dust_RADEC()` and `get_dust_sigma_RADEC()` query IRSA using a given RA and dec to get the Milky Way $E(B-V)$ and error in $E(B-V)$, respectively. These are located in `snpy.utils.IRSA_dust_getval`. They accept arguments for RA and dec, with the default dust map set as SF11 citation. They return two things: the result and a flag. The flag indicates that the function worked. You can throw this out.
**Usage example:** CHECK THIS IS CORRECT, RUN IT:

```
from snpy.utils.IRSA_dust_getval import get_dust_RADEC,
    get_dust_sigma_RADEC

'''
SN 1987A coordinates from
http://simbad.u-strasbg.fr/simbad/sim-id?Ident=SN+1987A.
'''

ra, dec = 279.703427, -31.937066
mwreddening,_ = get_dust_RADEC(ra, dec, calibration='SF11')
e_mwreddening,_ = get_dust_sigma_RADEC(ra, dec, calibration='SF11')
mwreddening = mwreddening[0]
e_mwreddening = e_mwreddening[0]

print(f'The Milky Way reddening for SN 1987A is {mwreddening}
    +/- {e_mwreddening}.')
```

# 7 Statistics Stuff

I guess you could argue this is outside the scope of a supernova manual, but these are things I use all the time. Plus, it's my darn manual, so I get to put what I want in here.

## 7.1 Error Propagation

## 7.2 Least Squares and Minimizing $\chi^2$

### 7.2.1 Example: Fitting a line

### 7.2.2 Example: Fitting a Hubble diagram

# Acknowledgements

# References

blah blah blah