

# Solar Radio Burst Analysis and the 2017 Solar Eclipse

THESIS

Presented to the Faculty of the Department of Physics and Astronomy  
in Partial Fulfillment of the Major Requirements  
for the Degree of Professional Physics

BACHELOR OF SCIENCE IN  
PHYSICS

Beverly Warner

May 2019

© 2019 Middle Tennessee State University  
All rights reserved.

The author hereby grants to MTSU permission to reproduce  
and to distribute publicly paper and electronic  
copies of this thesis document in whole or in part  
in any medium now known or hereafter created.

# SOLAR RADIO BURST ANALYSIS AND THE 2017 SOLAR ECLIPSE

Beverly Warner

Signature of Author:

---

Department of Physics and Astronomy  
May 2017

Certified by:

---

Dr. Charles Higgins  
Professor of Physics & Astronomy  
Thesis Supervisor

Accepted by:

---

Dr. Ronald Henderson  
Professor of Physics & Astronomy  
Chair, Physics & Astronomy

Accepted by:

---

Dr. John Vile  
Dean, University Honors College

## **ABSTRACT**

This is a study of solar bursts utilizing a record of multiple radio telescope data from differing observation sites across the country. Since identifying and comparing solar bursts manually is both cumbersome and subjective, I developed a Python program to complete this data analysis efficiently and to reduce visual bias. Utilizing this program, we are finding long-term trends in solar burst counts and intensities by running years' worth of radio data through this program. In extension to this, data taken during the solar eclipse of 2017 is being used to study the changes of the ionospheric plasma and how electromagnetic waves propagate through it. This program will be made available to all scientists using Radio Jove for their own solar data analyzing convenience.

## TABLE OF CONTENTS

Abstract .....	<i>iii</i>
List of Figures .....	<i>v</i>
I. Introduction .....	1
II. Observations and Data .....	5
A. Setup Details .....	6
B. Types of Files .....	7
C. Types of Solar Bursts .....	8
D. A Solar Burst Program .....	13
III. Methods .....	16
A. Construction of a Solar Burst Program.....	17
IV. Results .....	22
A. Accuracy of the Solar Burst Program .....	22
B. Solar Bursts Versus Sunspots .....	23
C. Solar Eclipse Results .....	27
V. Discussion and Conclusions .....	28
VI. References .....	30
VII. Acknowledgements .....	31
VIII. Appendix .....	32

## LIST OF FIGURES

<u>Figure 1:</u> Image of a solar burst taken by NASA’s Solar Dynamic .....	1
<u>Figure 2:</u> A photo of the solar eclipse at totality by Dr. Larry Burriss .....	3
<u>Figure 3:</u> Photo taken by Dr. Charles Higgins of a dipole antenna radio telescope situated at the MTSU Dairy Farm .....	5
<u>Figure 4:</u> An example of what single frequency data looks like through the Radio Skypipe software .....	6
<u>Figure 5:</u> Data recorded from Hawk’s Nest Radio Astronomy Observatory at 1650-1700UT on the day of the solar eclipse, 2017 .....	8
<u>Figure 6:</u> Examples of Type II solar bursts seen in a spectrograph .....	9
<u>Figure 7:</u> This shows examples of what singular and grouped type III burst look like in spectrograph files .....	10
<u>Figure 8:</u> Example of what a solar burst seen in a spectrograph file correlates to in single frequency, using data from HNRAO .....	11
<u>Figure 9:</u> This shows an example of what a type IV solar burst looks like in a spectrograph file .....	12
<u>Figure 10:</u> This shows an example of what a type V solar burst looks like in a spectrograph file .....	13
<u>Figure 11:</u> Four of the five sets we are using; half of these stations experienced totality .....	15
<u>Figure 12:</u> This is an example from Middle Tennessee State University of how the galactic background behaves over time .....	17

Figure 13: This is an example from Hawks Nest Radio Astronomy Observatory of how the background level can be falsely recorded to suddenly rise ..... 18

Figure 14: A simplified drawing of how the program calculates a background ..... 19

Figure 15: A simplified drawing of boundaries the program goes through to correctly count data as solar ..... 20

Figure 16: MTSU radio data example showing multiple solar bursts in one event ..... 21

Figure 17: Manual counts versus program counts for the same files in all of 2017 ..... 23

Figure 18: Graph of daily sunspot number and solar burst counts with respect to time ..... 23

Figure 19: Graph of monthly averages of both sunspot number and solar burst counts versus time ..... 24

Figure 20: Graph of solar burst counts versus sunspot number on a monthly basis ..... 25

Figure 21: Graph of the monthly averages of both solar burst intensities and sunspot number to see how they correlate ..... 26

Figure 22: Graph of monthly averages of both sunspot number and solar burst intensity versus time ..... 26

Figure 23: A graph of the differences in recorded solar burst intensity between stations ..... 27

**LIST OF TABLES**

Table 1: Names of the observatories/schools with single frequency radio data to use for our analysis ..... 4

## I. INTRODUCTION

Comparing solar data from different observatories is interesting because it presents a way for us to see what an event, such as a solar burst, may look like to a viewer set up hundreds of miles away. As a long-term goal, we are interested in finding a possible correlation of the intensity and number of recorded solar radio bursts with respect to time and compare them with sunspot activity.

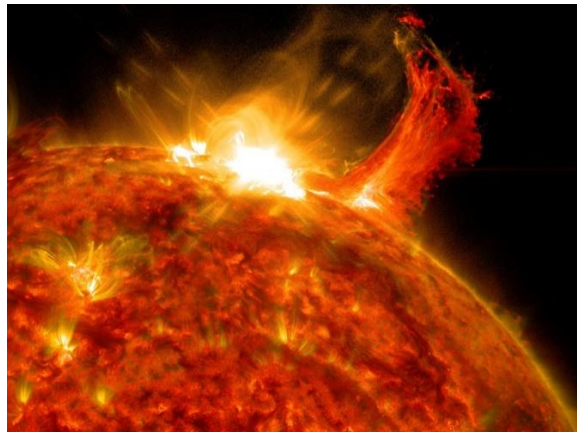


FIGURE 1: *Image of a solar burst taken on October 2<sup>nd</sup>, 2014 by NASA's Solar Dynamic Observatory (Zell, 2014).*

A solar burst, or solar flare, occurs when built up magnetic energy in the Sun's atmosphere is released suddenly as a large explosion of energy into space, as seen in Figure 1. The radiation is emitted at nearly all portions of the electromagnetic spectrum; therefore, these events can be viewed through radio waves at the longest wavelengths and studied at the optical and x-ray regions of the spectrum at the smaller wavelengths emitted (Holman, 2007). For this research, we are focusing on the far end of the spectrum: radio wave emissions. Using what we find with daily solar activity, such as sunspot number, we can

then closely analyze and compare these to recorded radio data during distinct events such as solar eclipses. Sunspots are visible “spots” on the Sun’s surface which vary in number on a day-to-day basis. Sunspots are commonly linked with solar radio bursts, but their true correlation with solar bursts is unknown.

Radio Jove is a project supported by NASA that encourages citizen scientists to observe and analyze natural radio emissions of celestial objects; we can use data from multiple observing sites submitted to the Radio Jove website to make sure what we saw in our data was truly solar activity, and not simply local radio-wave interference (Williams, 2019). Assuming another viewer sees the solar radio burst, comparing timing and amplitude can tell us more about the fluctuations of the ionosphere. Using the recent solar eclipse in the summer of 2017, we have viewed data from across the country during the few hours preceding, during, and after the solar eclipse.

An eclipse is defined to be when one celestial body moves in the shadow of another, totally or partially obscuring it from the viewer. A solar eclipse is when the Moon is seen to move in front of the Sun, blocking light from the Sun to an observer on the Earth (Wild, 2015).





FIGURE 2: *This is a photo of the solar eclipse at totality, taken on August 21<sup>st</sup>, 2017 by Dr. Larry Burriss in Murfreesboro, TN (Burriss, 2017).*

The 2017 eclipse was distinct because of the range of people that saw it; the entire United States saw at least a partial solar eclipse, where a total eclipse was seen by locations directly beneath the path of orbit. Being in Middle Tennessee, our university was in an optimal position: in the path of totality. An example of a total solar eclipse is shown in Figure 2. The radio telescopes set up in other parts of the country that did not experience totality were also important in viewing what solar activity was collected, for example, comparing their recorded solar data versus our own campus data.

In collecting solar radio data from the surface of the Earth, the electromagnetic waves radiated by the solar bursts from the Sun need to traverse through our ionosphere. The ionosphere resides in the thermosphere, about 80km above the surface of the Earth. At this height, the atmosphere is thin enough to let free electrons be on their own; the number of free electrons is enough to affect radio transmission. This ionized percentage of the atmosphere is called the ionosphere, where the amount of ionization fluctuates over a given point (Hogan, 2019).

We see a reduction in ionization of the ionosphere when the Sun is on the other side of the globe. Because we know that the ionosphere becomes less ionized at night, we are interested in seeing if the obstruction of the Sun during the eclipse caused a drop in ionization. With the Moon completely eclipsing the Sun, this might have created a “false night”, where the intensity of our collected radio data is noticeably different than a station who had not experienced a complete obstruction of the Sun. After the initial viewing of all posted data, we closely compared the most reliable data in terms of limited local interference and where the radio telescope is set up correctly.

Our radio telescope consists of a dual dipole antenna connected to a receiver, which feeds into a computer; an example dipole antenna is shown in Figure 3. We set up our telescope in many locations before finding its permanent home at the Middle Tennessee State University Dairy Farm. This enabled us to experiment with and to see what a major hurdle local interference can be to work around, as well as to understand the importance of the set-up process. For analyzing radio telescope data for the solar eclipse, we were left with about five sites to use including those of our campus, Middle Tennessee State University (MTSU). The sites used are listed in Table 1.

Observatory/School Name	Partial/Totality	Location
MTSU	Totality	Murfreesboro, TN
HNRAO	Partial	Industry, PA
AJ4CO	Partial	High Springs, FL
K4LED	Totality	Meeks Park Blairsville, GA
Cox	Totality	Easley, SC

TABLE 1: *Names of the observatories/schools with single frequency radio data to use for our analysis.*



FIGURE 3: *Photo taken by Dr. Charles Higgins of a dipole antenna radio telescope situated at the MTSU Dairy Farm.*

Finding possible correlations of the intensity and number of recorded solar bursts with respect to time takes years' worth of data in order to be significant. Going through data by hand is time-consuming and can be subjective, so the best way to handle data collecting is to run them through algorithms set by a computer program. Doing so eliminates the subjectivity of what is and is not solar, as well as comb through months of data in a matter of hours rather than weeks.

## **II. OBSERVATIONS AND DATA**

Seeing a radio data file of any type without any previous knowledge about radio telescopes would be confusing and not mean anything physically to the uninformed viewer. Important information includes the method of collecting data, the types of files the data is recorded into, and how to interpret said data files.

## A. Setup Details

The dual dipole antenna, half of which is shown in Figure 3, is comparable to the visual telescope in that it takes in the information available in a region of sky. In this analogy, the computer attached to the antenna via wire acts as the eyepiece to this telescope, where the viewer can see what data the dipole is intaking at 20.1 MHz. A single antenna of this type has about 60 degrees of the sky in view, being the most sensitive directly overhead, which means that it can best record about 4 hours of solar activity as it is seen passing overhead.

Using the Radio Jove receiver, single frequency data from the antenna is fed through a software called “Radio Skypipe”, which plots the observed radio data as temperature with respect to time, seen in Figure 4 (Sky, 2019).

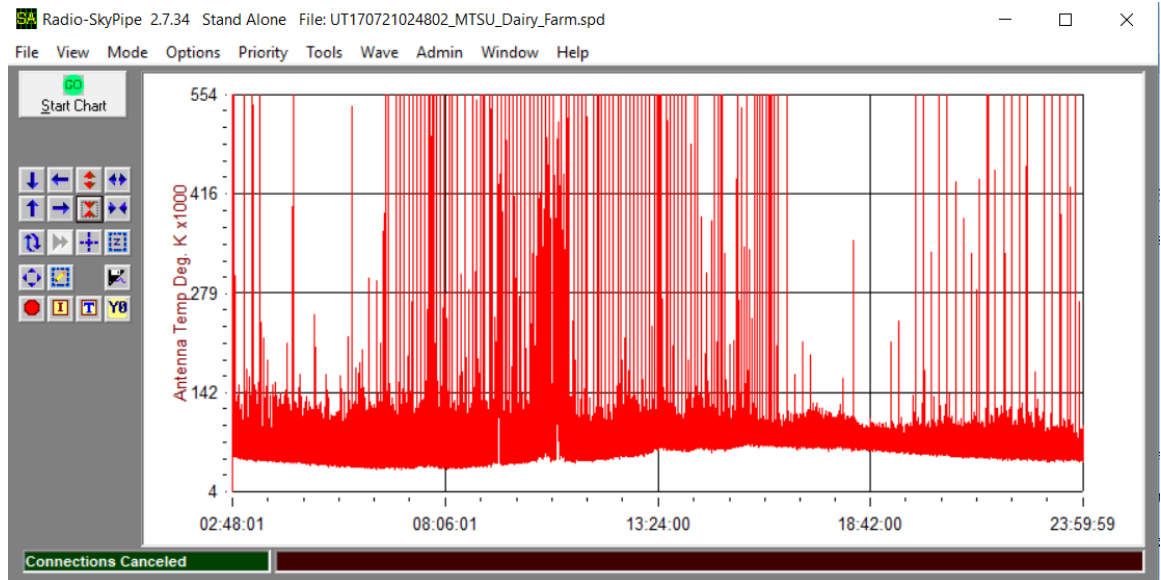


FIGURE 4: An example of what single frequency data looks like through the Radio Skypipe software, using MTSU data (Williams, 2019).

The intensity is given in temperature because this relates to the temperature of a blackbody that would give a radio signal of that strength. An object is a blackbody if it emits energy like a perfect emitter; our Sun is a good approximation to a blackbody. Data are measured in units of kilo-Kelvin, where time is recorded in the format “hour: minute: second” in Universal Time (UT). Universal Time, also known as Greenwich Mean Time or Mean Solar Time, is defined to be the standard time in a zone that includes the British Isles. This is used rather than local time in order to be able to compare easily with data taken all over the world.

Calibration is important to tune the telescope to give accurate antenna temperature values; it is used to convert the relative signals into an absolute scale of temperature values. Without calibration, the temperature readings given by the antenna would be relative to some unknown quantity, and only the change in temperature seen in the data would be accurate, not the scale.

## **B. Types of Files**

There are two types of radio files used to analyze solar activity: single frequency files, and the much larger spectrograph files. A spectrograph file shows data as frequency versus time while a single frequency file is intensity versus time. Of the two data types, it is much easier to compare stations using the single frequency files due to the intensity values being provided, whereas the intensities are given as approximate color differences in spectrograph files. However, to find which frequency a solar burst presented itself most clearly, or to classify a solar burst as one of the five possible types of bursts given its shape, a spectrograph would be the best choice. A spectrograph can be visualized as layers of

single frequency data stacked in increasing order, usually from 15 to 30 MHz, where change of color denotes change in temperature. A visual comparison of the two types of files is shown in Figure 5; Jim Brown, a regular contributor of the Radio Jove program, is behind these two different methods of data collecting using radio telescopes. I am focusing on analyzing the single frequency data and only using the spectrograph files to check solar events found in single frequency data. Not only can we compare single frequency files more easily, these files are also more readily available to us given our large database from 2005 to the present, which we can easily call upon due to their compact size.

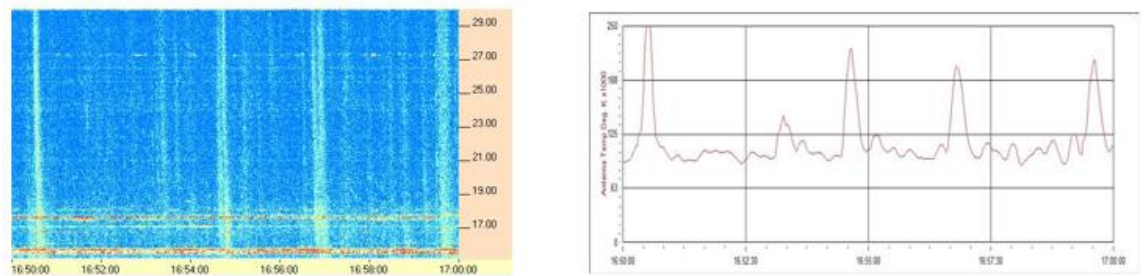


FIGURE 5: *Data recorded from Hawk's Nest Radio Astronomy Observatory at 1650-1700UT on the day of the solar eclipse, 2017. A spectrograph file is shown on the left, while the single frequency file for the same day and time is shown on the right (Williams, 2019).*

### C. Types of Solar Bursts

There are five types of known solar bursts; they do not follow the same form in radio data, so it is important to account for all possible types of solar activity.

## Type I

These types of solar bursts are always seen in groupings, called “storms”. They are short-lived, only about one second each. However, these are found at frequencies between 80 and 200 MHz. This means that we would not deal with these particular bursts in our data, so we can ignore type I solar bursts for the sake of this research.

## Type II

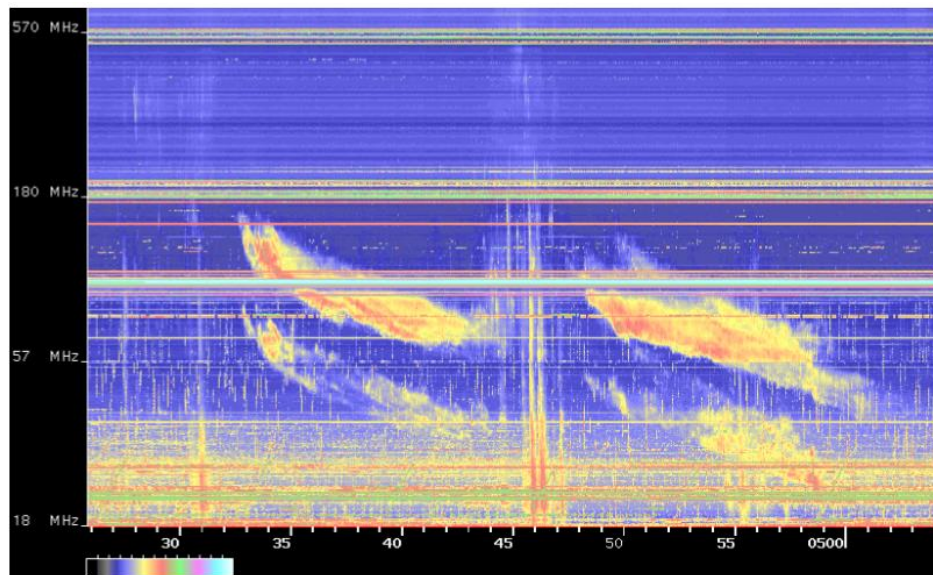


FIGURE 6: *Examples of Type II solar bursts seen in a spectrograph file (Terkildsen, 2005).*

Type II bursts are easily characterized in spectrograph files as having a “swept out” appearance that travels across a large range of frequencies. This sweeping motion, two of which are seen in Figure 6 starting at the 33-minute and 49-minute mark each, means that the burst one of the single frequencies would be drawn over a longer amount of time. Type II solar bursts are known to last for a range of 3 to 30 minutes.

## Type III

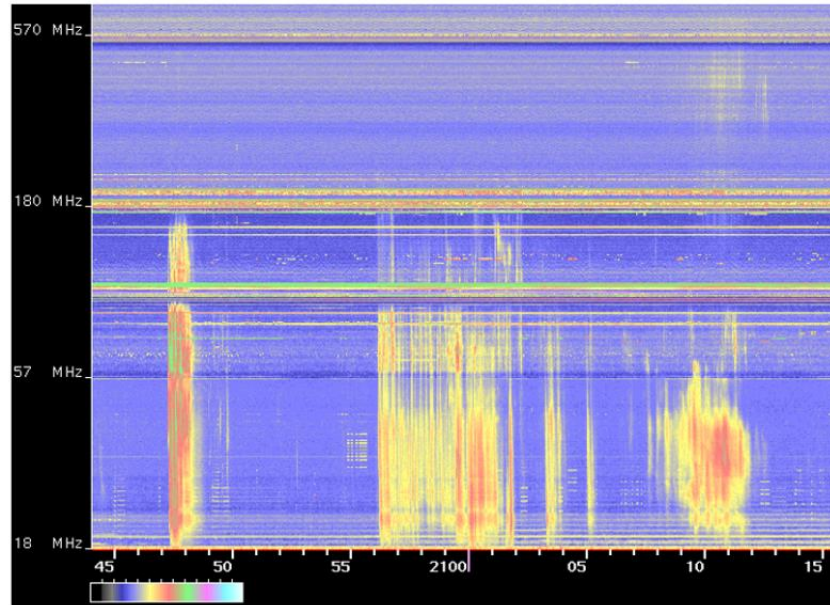


FIGURE 7: *This shows examples of what singular and grouped type III burst look like in spectrograph files (Terkildsen, 2005).*

These are the most commonly seen solar bursts in our data, where they can occur singularly or in groups. Singular type III bursts last a few seconds where groups last upwards of 5 minutes. Group and singular type III bursts are shown in Figure 7. As seen in the figure, type III bursts cover a large range of frequencies, and are characterized as a vertical slash on the spectrograph background (Antar, 2014).



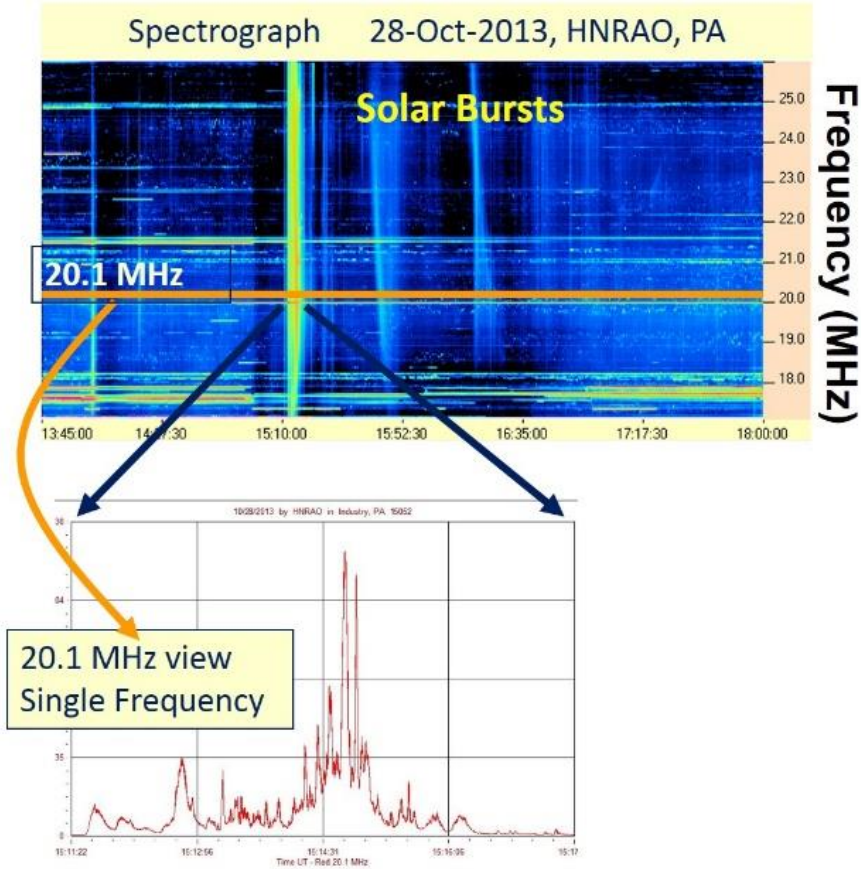


FIGURE 8: Example of what a solar burst seen in a spectrograph file correlates to in single frequency, using data from HNRAO. Image provided by Dr. Charles Higgins.

Seen in Figure 8 is how, specifically, a type III solar burst recorded in a spectrograph file presents itself in a single frequency file. As shown, this singular green stripe in the spectrograph file corresponds to multiple peaks in intensity in the single frequency file.

## Type IV

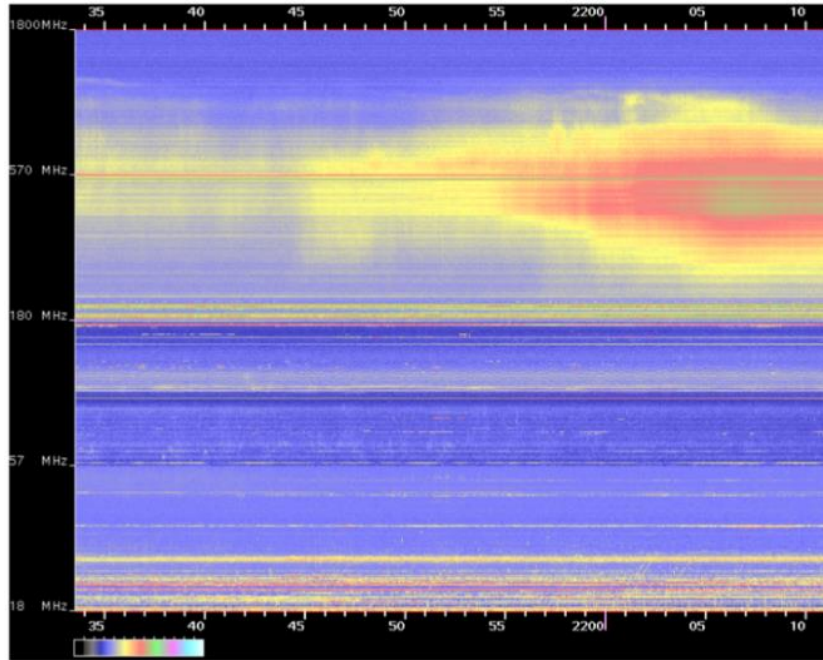


FIGURE 9: *This shows an example of what a type IV solar burst looks like in a spectrograph file (Terkildsen, 2005).*

This type of solar burst can best be summarized as a broadband, continuous burst. These are the longest lasting solar bursts and have a large range of existence from a few minutes to a few days. The type IV burst in Figure 9 is clearly seen around 570 MHz and continues in differing intensities for the length of the interval of time shown.

## Type V

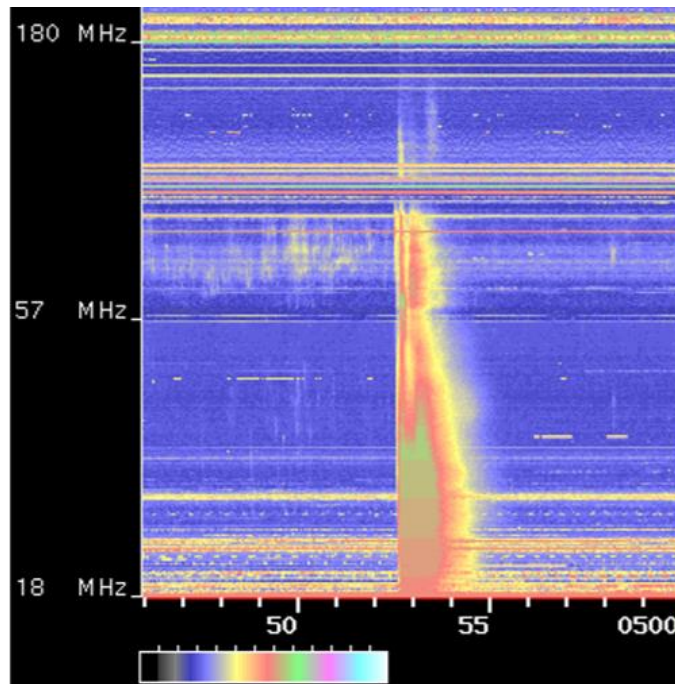


FIGURE 10: *This shows an example of what a type V solar burst looks like in a spectrograph file (Terkildsen, 2005).*

A type V solar burst is like that of a type III in its physical appearance in a spectrograph file, except it tends to linger at the low frequencies longer, thus making a slightly “swept” look similar to type II solar bursts, as seen in Figure 10. Although they linger at the low frequencies, type V bursts are not known to last longer than 3 minutes.

### **D. A Solar Burst Program**

#### **Quantifying the Recorded Differences in Number and Amplitude of Solar Bursts**

Creating a program that will go through radio data and pick out the times and intensities of individual solar bursts is useful for analyzing radio data for correlations and patterns efficiently and with limited error.

Looking at raw individual radio data first, the purpose is to record the number of solar bursts and their amplitude as a function of time for a station on the day of the solar eclipse. For each ten-minute interval of individual data, the program finds the average background temperature. Because the background changes with time, recalculating over small intervals results in a curve that tracks the galactic background. Next, it calculates the standard deviation of the found average background temperature. This is done because there is variance in the background, this interference should not be counted as solar.

Once the background and standard deviation are found, the program can identify solar bursts within that interval. Local maximas are recognized when the temperature is above the background by at least ten times the standard deviation. This value is what we call the “threshold” value for the data. Because most solar bursts rise substantially above the background, counting local maximas above the variance in the background will limit the false inclusion of interference. Once a solar burst is found, the corresponding temperature and time are recorded. We can use the temperature of a burst at a given time to directly compare data sets from different observatories.

We then compare the found number of solar bursts and their amplitudes as a function of time at many stations; the amount of recorded solar could vary between stations as well as the average intensities. For the two-hour time intervals before, three hours during, and two hours after the solar eclipse for multiple data, we calculate the absolute value of the temperature differences seen in recorded solar events. Differences in calibration do not matter; we are only interested in relative differences between stations. Each difference of solar radio bursts between stations are plotted, where the ordinate is change in temperature, and abscissa is the time. The partial and total eclipse start and end

times are indicated on the graphs by the placement of C1, C2, C3, and C4, as seen in Figure 11. An observation station not in the path of totality would only see C1 and C4, where a station in totality would see all four stages of the eclipse.

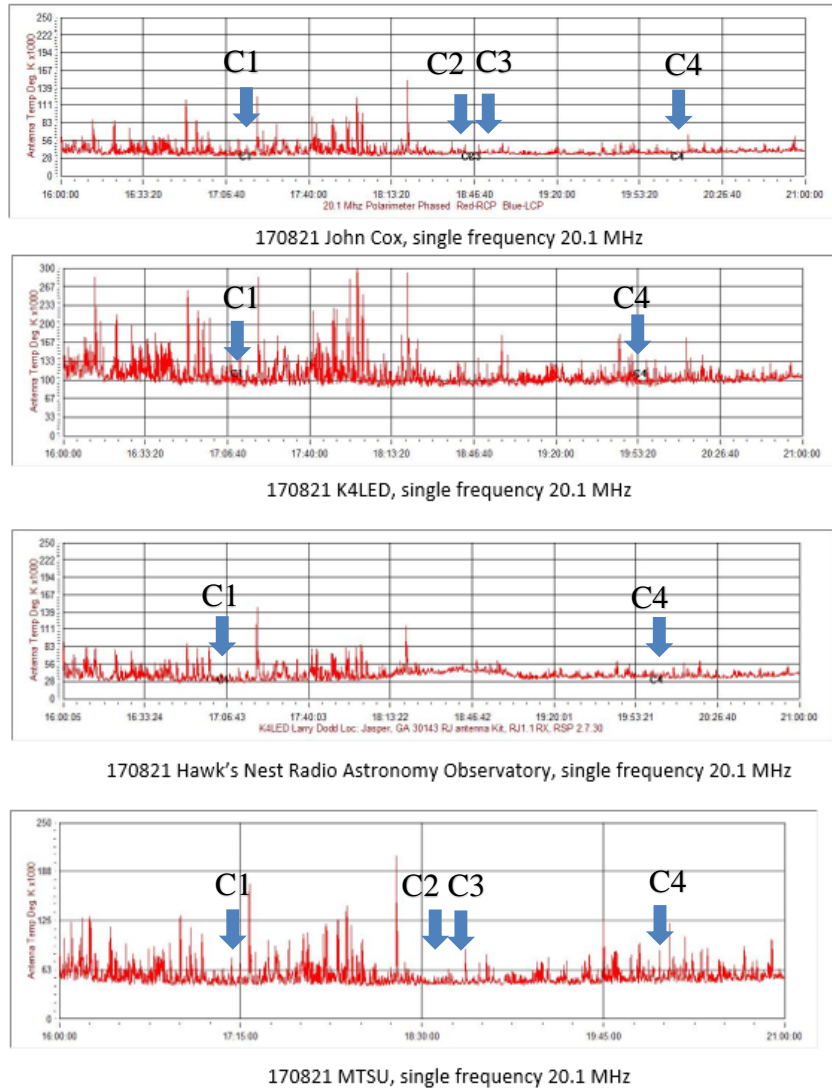


FIGURE 11: Seen are four of the five data sets we are using; half of the stations shown experienced totality (Williams, 2019).

Differences in solar intensity could be due to the presence of the lunar shadow and its effect on the ionosphere. Labeling the start and end times of the event could help show a shift in the differences due to the lunar shadow. Along with this comparison, we plot the solar burst counts as a function of time and compare overall solar burst counts; if one station sees more solar events or differences in intensity, this could be due to changes in the ionosphere.

### **III. METHODS**

The goal of this program is to identify and acquire a 2D list of intensities and corresponding times of each solar burst in a given data file. The solar bursts seen in one single frequency graph were duplicated in all other graphs at the same time intervals, the only difference being the amplitude in which they reached. This can be seen in Figure 11: the larger solar activity occur at the same time in all graphs. We marked which data sets are from those who did not see totality, and how that effected the recorded intensity of the solar events. Once the data are separated into these two groups, the intensity, amount, and placement of solar bursts in each of these sites in the days leading to, following, and the day of solar eclipse can be plotted against each other to find whether the lunar shadow presents any effect on the ionosphere and therefore the recorded radio data. My Python program will make it so more data can be analyzed in a much shorter amount of time.

## A. Construction of a Solar Burst Program

Defining a galactic background:

Separating the data file into one-hour windows, the approximate intensity of the Galaxy itself is found by slicing the one-hour window into even smaller intervals. The galactic background is a measure of how much radio energy the Milky Way Galaxy is emitting. This background value can usually be assumed to be constant over the span of an hour, however data files can experience a sudden rise in the background level due to an error in the data-intake equipment or other forms of interference. The background is the baseline of the data being recorded; seen in Figure 12, the background is approximately 50,000K with some variation each hour. In order to account for changes, the background intensity level is redefined every ten minutes. The focus here is to define small intervals of backgrounds to accurately represent the background level throughout the file, even if background levels change significantly, as it is seen to do in Figure 13.

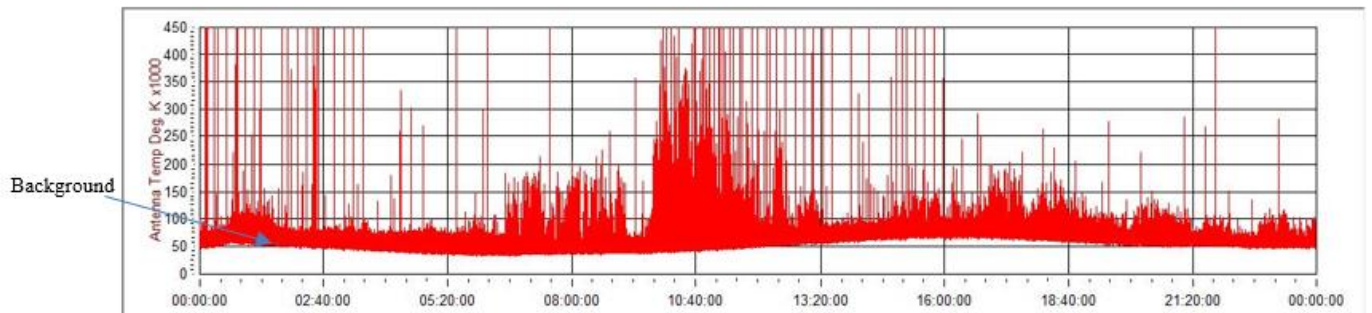


FIGURE 12: This is an example from Middle Tennessee State University of how the galactic background behaves over time. The spike-like forms protruding from the background are cases of local interference (Williams, 2019).

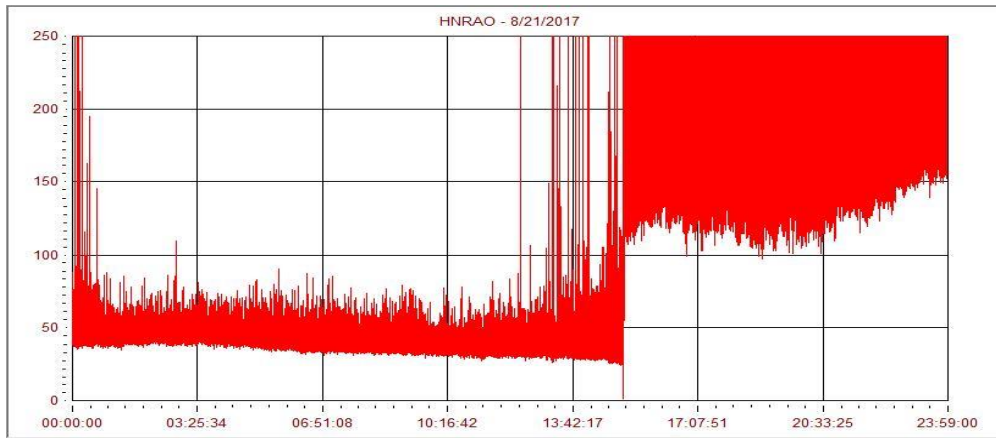


FIGURE 13: *This is an example from Hawks Nest Radio Astronomy Observatory of how the background level can be falsely recorded to suddenly rise (Williams, 2019).*

Going through the data one at a time, a minimum intensity is found every 15 seconds. To be recognized as the minimum, the slope to a point near the minimum must not be too large; this check is to ensure the minimum value is not interference from local radio waves, which are categorized by their sharp, linear appearance. Once ten minutes worth of minimums is reached, averaging the minimums yields the approximate local background intensity. The standard deviation of each background is also recorded to be used in calculating the threshold of detecting a solar event.



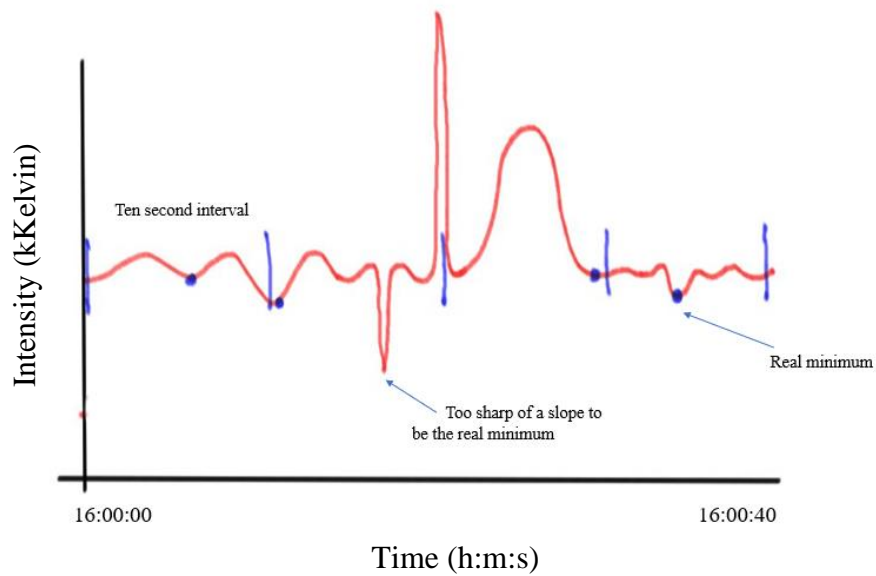


FIGURE 14: A simplified drawing of how the program calculates a background.

#### Defining a solar burst:

The purpose is to define a solar burst so all interference values are looked over while all solar activity is found. A solar burst seen in single frequency data resembles a bell curve, spanning over seconds or minutes at a time. Viewing an hours' worth of data at a time, the data is put into ten-minute slices in order to accept a new value of the background for each iteration. Figure 14 shows a simplified version of the algorithm this program goes through to define a new background level. For solar activity to be recognized by the program, it must first exceed the threshold over the existing background. The threshold that separates solar from low intensity interference is defined as being the background value plus a standard deviation, sigma. The necessity to be an amount above the given background presents more confidence that an event counted as solar is real. The program also accounts for the amount of time the event is recorded to be above the background.

A type III solar burst, the most commonly seen, is characterized by a bell-shaped curve spanning over a time greater than or equal to a few seconds. Though type III's are the most common, the other types of solar behavior must be tested for as well. As it happens, the solar eclipse day was found to be comprised of near constant bursting: a type IV solar burst. In a single frequency file, these are characterized as many bell-shaped curves spanning over small intervals of time, one following directly after the other. Therefore, this program goes through each data point and ignores all except for the interval of data points that are above the threshold for a given time and holds a certain shape. This means that if the test fails for one type of burst, it will go down a series of tests that fit all possible solar behavior before being discarded.

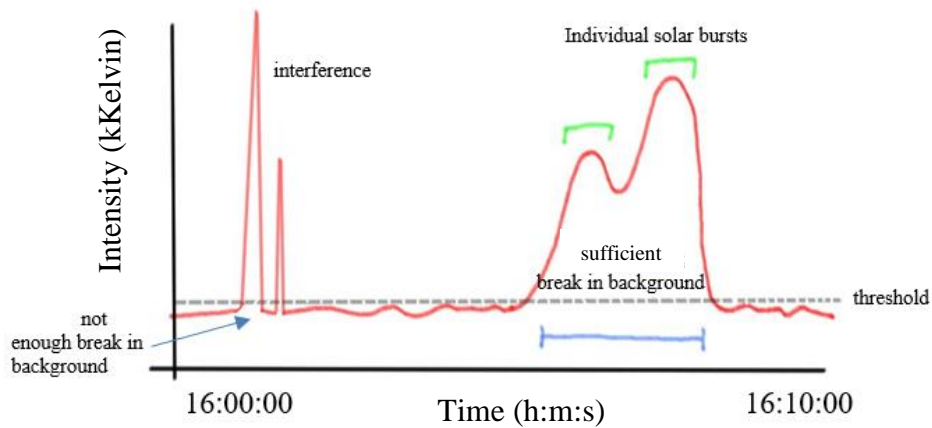


FIGURE 15: A simplified drawing of boundaries the program goes through to correctly count data as solar.

Once the conditions of time and behavior are met, the program determines the solar burst peak values. Figure 15 shows a simplified version of what the program identifies as solar and what it discards as interference. Because individual solar events do not

necessarily only have one solar burst peak comprising them, the program needs to find the local maxima for each burst separately. The intensity value and the time of the solar burst is recorded each time one is found. Real examples of solar events with multiple solar burst peaks is included in Figure 16.

Finally, for this program to be easily used, I have made custom settings available alongside the default values for the viewer. These include the desired start and end times of analysis, as well as the option to save the solar burst 2D list to a text file for further analysis.

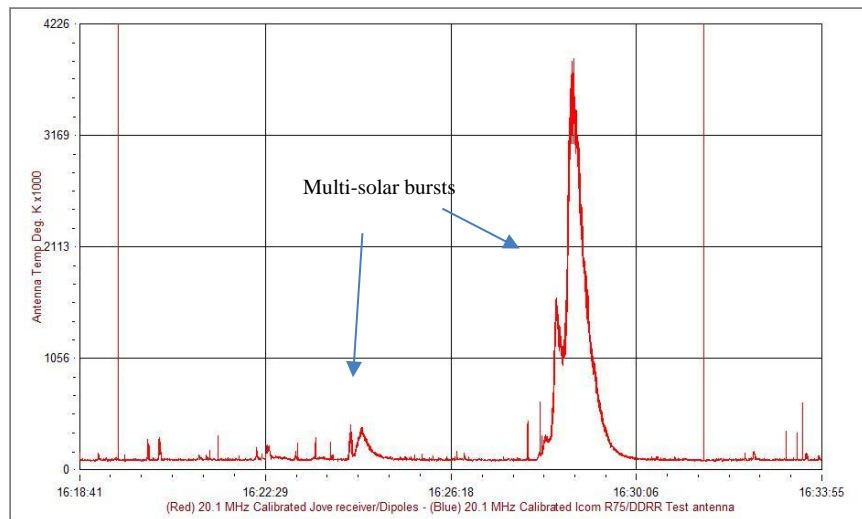


FIGURE 16: MTSU radio data example showing multiple solar bursts in one event (Williams, 2019).

## **IV. RESULTS**

The major goal of this research was to create a computer program that would accurately find solar bursts in single frequency files. The next step was to run the program through a years' worth of single frequency data and compare the found solar radio burst counts and intensities to known sunspot counts. After looking at the long-term occurrence of sunspot number, the last step was to then apply my program to analyze a specific event: the solar eclipse of 2017.

### **A. Accuracy of the Solar Burst Program**

We are interested in seeing how similar my program counts solar bursts compared to a student manually counting solar bursts through single frequency data. The correlation was found to be about 55.2 %; there are multiple reasons that could explain why this correlation is not stronger. One of the main reasons that I found when checking my program counts was that it was finding type II bursts. By eye, type III bursts are more obvious in a single frequency file due to them making a clear break in the background when they occur. However, type II bursts exist for much shorter periods of time and can be easily looked over. Because type II bursts are commonly seen in storms, there can easily be 50 solar bursts in one storm. This explains how the most variation of counts, as seen in Figure 17, was found when the manual counts were less than ten solar bursts.

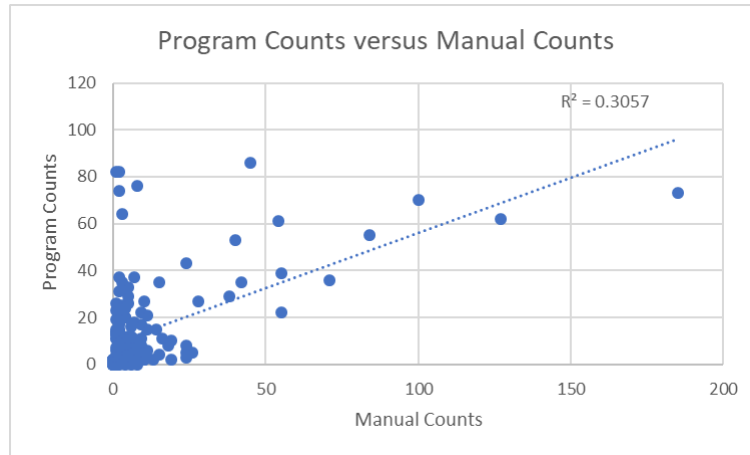


FIGURE 17: *Manual counts versus program counts for the same files from 1/1/2017 until 10/1/2017.*

## B. Solar Bursts Versus Sunspots

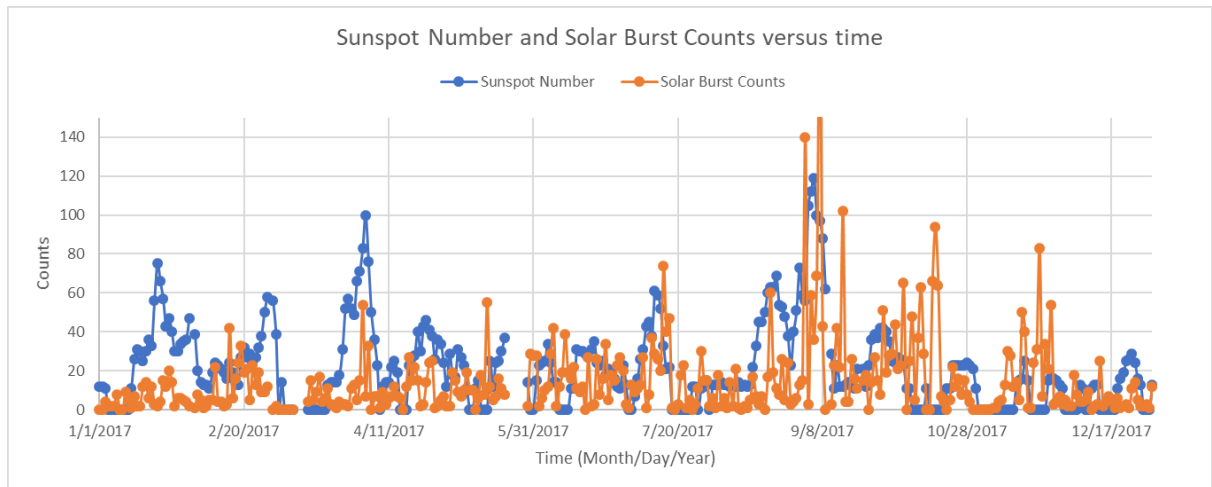


FIGURE 18: *Graph of daily sunspot number and daily solar burst counts with respect to time.*

My program ran through a year of single frequency data recorded at one observation station. We used data from Hawk’s Nest Radio Observatory (HNRAO) in Pennsylvania for this analysis because this station had the most data available during the year of 2017. Plotting the daily sunspot number and daily solar burst count with respect to time shows a

definite trend, which is encouraging because it agrees with what past studies showed us about the connection of sunspot number and solar burst counts.

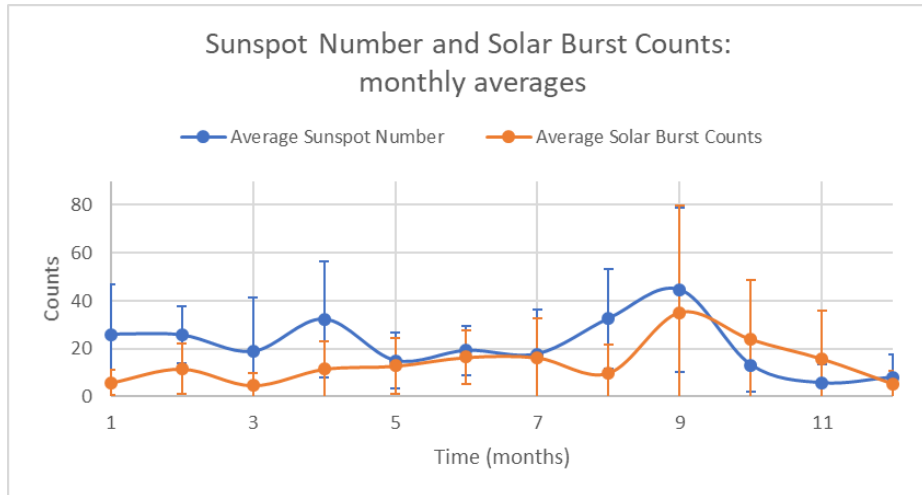


FIGURE 19: Graph of monthly averages of both sunspot number and solar burst counts versus time.

Figure 19 shows what this trend looks like when we average all the daily counts of sunspots and solar bursts in a monthly format. You can clearly see the similarities of the rises and falls of both sunspots and solar burst counts; this leads us to believe that these are in fact related occurrences. In order to draw more firm conclusions on this, we will need to run my program through many years of data and analyze the long-term trends throughout the solar cycle.

Although sunspot number and solar burst counts follow a similar pattern with respect to time, we noticed that the actual number of sunspots and solar bursts could vary from each other quite extensively. This is partially due to the high variability of the Sun on a daily basis. This causes the correlation value for these two to be rather low at a 37.1% correlation, as seen in Figure 20. This means that while we are getting large amounts of

sunspots, we are also getting many solar bursts, but not necessarily the same number of bursts as we have sunspots.

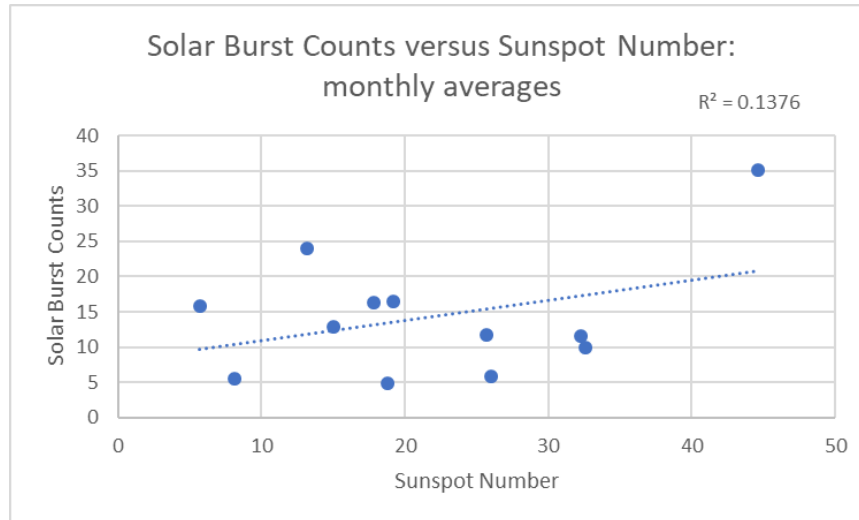


FIGURE 20: Graph of solar burst counts versus sunspot number on a monthly basis.

We also examined the average intensities of the found solar bursts and compare these with the sunspot data. This has never been done before, as recording the intensities of solar bursts found by hand would be an even longer process than finding solar burst counts. Figure 21 shows a years' worth of average intensities per month compared to sunspot number. We found there seemed to be no connection between the two. This means that on days when there are many sunspots, the average intensity of solar bursts we are receiving remains nearly constant. From Figure 22, the average intensity per month does not respond to the change in sunspot number and is even seen to remain rather constant throughout the year. The new availability to study the trend of solar radio burst intensities means that we can now plot the average intensities over many years of data to see if the intensities have no relation to sunspot number and the overall solar cycle.

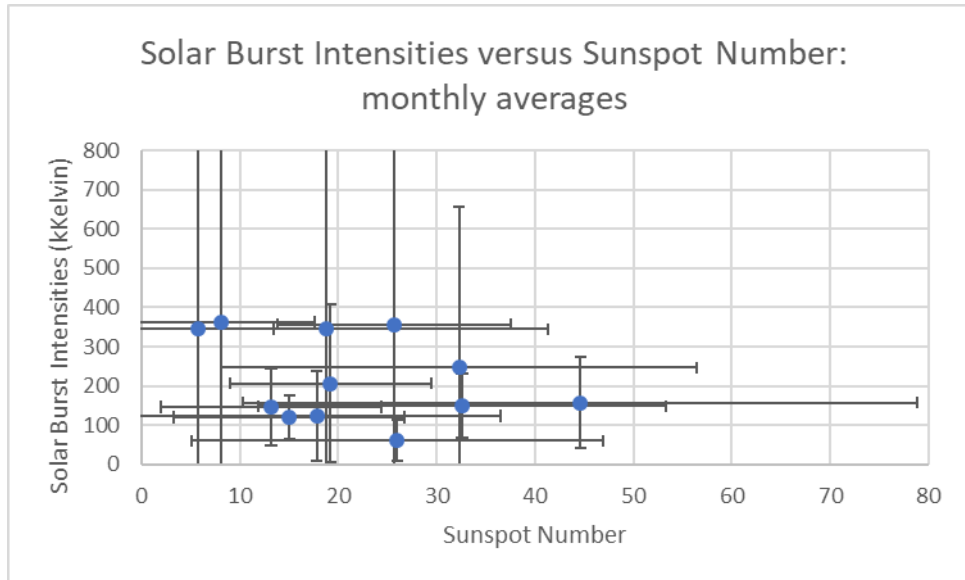


FIGURE 21: Graph of the monthly averages of both solar burst intensities and sunspot number to see how they correlate.

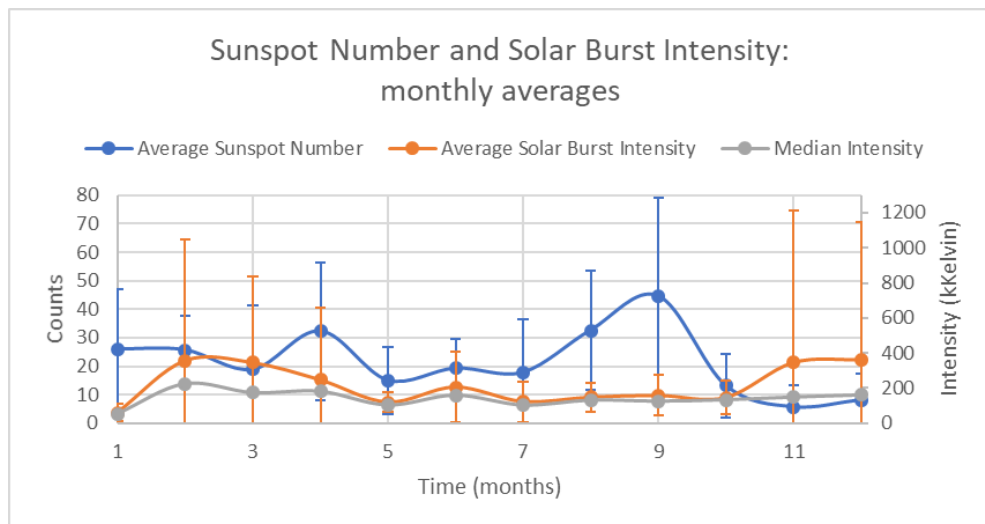


FIGURE 22: Graph of monthly averages of both sunspot number and solar burst intensity versus time.



### C. Solar Eclipse Results

In looking closely at the solar eclipse of 2017, we have some preliminary results that could be interesting to research further. In three of the five observation stations we were utilizing for this analysis, we found an average of 30 type II solar bursts per hour during our defined interval of 16-22 UT on the day of the solar eclipse. To identify if we were seeing a difference in average intensity as the eclipse progressed, we took the difference of found solar burst intensities between MTSU and two other stations located in Pennsylvania and South Carolina. The Pennsylvania observation station, titled HNRAO in Figure 23, experienced only a partial eclipse where the station in South Carolina, titled Cox, was in the line of totality like MTSU. Noting that the eclipse reached totality at 18.5 UT, the plot of these data implies that there is a significant change in the difference in intensity inside totality versus out. At this point in our research, there are no conclusions that we can draw from this, however knowing that there is something significant happening during this time allows for more tests to be done on this event.

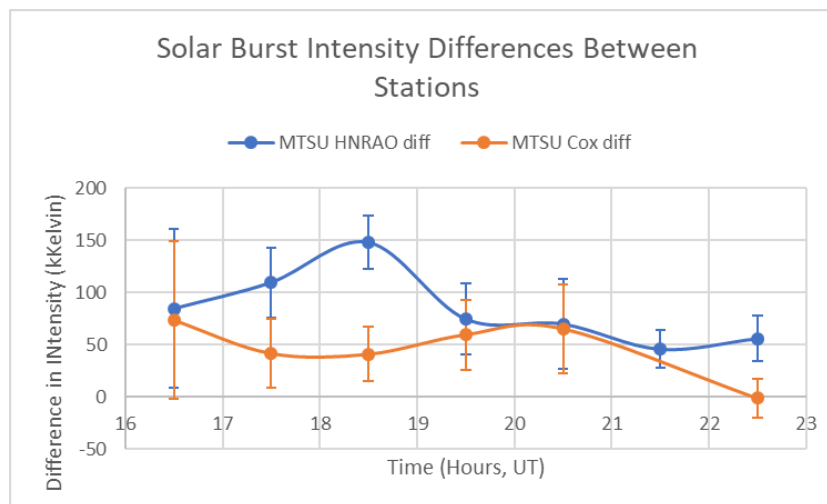


FIGURE 23: A graph of the differences in recorded solar burst intensity between stations.

## V. DISCUSSIONS AND CONCLUSIONS

It is possible that some interference that looks like solar is finding its way past my checks for interference, however I have eliminated the lion-share. The program originally only accounted for static, but through multiple tests of running through data, I have found two more types of interference that needed their own functions for the program to recognize and ignore. These include the radio interference we get from thunderstorms as well as what is commonly called “box interference” because of its square appearance on the single frequency file.

Box interference comes from local appliances that switch on and jump to a certain frequency level and then after a period, switch back off. This is seen in the single frequency file as jumps from the background level to an intensity, where it wavers slightly in intensity throughout an amount of time, and then disappears back into the background as quickly as it appeared. The problem with this type of interference is that the jump in intensity happens slowly enough that it passes the static test, and therefore the program would continue to count the slight wavers in intensity of the box interference as solar events.

Thunderstorms affect large portions of radio data, making it near impossible to unearth solar events within them. Someone manually counting solar bursts would likely skip over sections that experienced a thunderstorm and so I have implemented a function that will do just that: check for thunderstorms and skip over those periods of data. Without the inclusion of this function, I was finding that the peak intensities that the thunderstorm took were confusing my program into thinking they were solar events.

With the completion of this program, we have found that it works well and can be perfected more with small changes as we see them necessary. By running through a years' worth of data, the program agrees with previous speculations that solar radio burst counts are directly related to sunspot number. We were also able to identify if we see a relationship of the solar bursts' average intensity with sunspot number. From the data we have collected and analyzed thus far, we are finding that there is no connection between intensity of bursts and sunspot number. My program will continue to be used to record the counts and intensities of solar bursts through many more years of data to solidify our conclusions. Lastly, utilizing my program, we were able to make some preliminary results on the solar eclipse of 2017. We are finding an interesting change in recorded intensity depending on if the observation station experienced totality or just a partial eclipse. Future research on this topic could help us identify the reasons for this perceived change.

## VI. REFERENCES

Antar, Y.M.M. & Zavvari, Azam & Islam, Mohammad & Anwar, Radial & Marie Hasbi, Alina & Asillam, Mhd Fairos & Monstein, C. Antennas and propagation around the world: CALLISTO radio spectrometer construction at Universiti Kebangsaan Malaysia. *Antennas and Propagation Magazine, IEEE*. 56. 278-288. 10.1109/MAP.2014.6837099, 2014.

Burriss, Larry. Murfreesboro, TN, 21 Aug. 2017.

Hogan, Dan. "Ionosphere." *ScienceDaily*, [www.sciencedaily.com/terms/ionosphere.htm](http://www.sciencedaily.com/terms/ionosphere.htm), 2019.

Holman, Gordon. "What Is a Solar Flare?" NASA, [hesperia.gsfc.nasa.gov/sftheory/flare.htm](http://hesperia.gsfc.nasa.gov/sftheory/flare.htm), 1 Aug. 2007.

Sky, Jim. Radio-Sky Publishing, [radiosky.com](http://radiosky.com), 2019.

Terkildsen, Mike. "Spectrograph: A Brief Introduction of Radiospectrogram Analysis." *Australian Government Bureau of Meteorology: Space Weather Services*, [www.sws.bom.gov.au/World\\_Data\\_Centre/1/9/5](http://www.sws.bom.gov.au/World_Data_Centre/1/9/5), 9 Jan. 2005.

Wild, Flint. "What Is an Eclipse?" NASA, [www.nasa.gov/audience/forstudents/5-8/features/nasa-knows/what-is-an-eclipse-58](http://www.nasa.gov/audience/forstudents/5-8/features/nasa-knows/what-is-an-eclipse-58), 1 June 2015.

Williams, David. "The Radio JOVE Project." NASA, [radiojove.gsfc.nasa.gov/](http://radiojove.gsfc.nasa.gov/), 2019.

Zell, Holly. "NASA Releases Images of a Mid-Level Solar Flare." NASA, [www.nasa.gov/content/goddard/nasa-releases-images-of-mid-level-solar-flare/](http://www.nasa.gov/content/goddard/nasa-releases-images-of-mid-level-solar-flare/), 2 Oct. 2014.

## VII. ACKNOWLEDGEMENTS

- 1) Dr. Baptiste Cecconi  
Wrote software to change the format of a single frequency file from spd (standard Radio Skypipe file) to CDF (common data format), a standard scientific data format where my program can read in the desired data.
- 2) Dr. Charles Higgins  
Mentor, advisor, professor that guided and advised my direction in this research.
- 3) Benjamin Kulas  
Adjusted Dr. Cecconi's program to read date/time data in a better format for my program to analyze.
- 4) Travis Marlow  
Found manual counts for the full year of 2017 that I compared my program counts with.

## VIII. APPENDIX

```
#!/usr/bin/python
#Beverly Warner
#This program only reads raw data with an intake rate of 10 data points per second
#Goals:
#1) reading in a folder worth of data >
#2) changing raw data into subsample10-smooth10 data >
#3) exporting new files as txt files >

import radiojove_spd3
import numpy as np
from pprint import pprint
import json
import os
from pathlib import Path

#constants:
MINUTE=1.0/60.0
TEN_MINUTES=MINUTE*10
THIRTY_MINUTES=MINUTE*30
ONE_SECOND=((1.0/60.0)/60.0)
TEN_SECONDS=ONE_SECOND*10
TWO_SECONDS=ONE_SECOND*2

#read in the Radio Jove Single Frequency Files in their default format
#(daily files in day folder, all in month folder)
file=[] #individual files
folder=[] #the month folder
#Allow custom calling of folders in directory:
name=input("Name of month folder: ")
base = Path("/Users/BWarn/Documents/FullYear") / name
files=os.listdir()
for dir in (x for x in base.iterdir() if x.is_dir()):
    subs = [str(sub.resolve()) for sub in dir.iterdir() if sub.is_file() and sub.suffix ==
'.spd']
    folder.extend(subs)
    file.extend([Path(sub).name for sub in subs])

def main():
#present all data in one hour format:
#create a textfile for this observatory's 2D list of solar burst data:
    n=0
    #Allow custom intervals of solar burst analysis:
    begin=input("What is the starting hour (UT)? [If using default interval, press
enter]: ")
    #default values go from 12 UT to 22 UT
    #ask to save each day in a separate text file to analyze the time and intensities of
bursts later
    save=input("Do you want to put the solar burst lists into text files? [Yes=1,
No=press enter]: ")
    #default interval of time:
    if begin=="":
        begin=12
```

```

    hours=11
else:
    begin=int(begin)
    hours=int(input("How many hours to analyze?: "))
for x in range(len(folder)):
#go through all files in the month folder
    q=begin #beginning time
    daily_solar=[] #2D list of all solar bursts found daily
    daily_counts=0 #daily count of solar bursts
    sum_int=0 #summation of intensity values of bursts
    average_solar=0
    #2D list of all data during each day: (time, value):
    pairs=file_grab(n)
#    print("starting hour: ",pairs[0][0], "UT")
#    print("ending hour: ",pairs[len(pairs)-1][0],"UT")
    for x in range(hours):
        start=pairs[0][0]
        #check to make sure the time given is not less than when the file starts
        if not float(q)<start:
            while float(q)>start:
                start+=MINUTE
            j=0
            while j<len(pairs) and pairs[j][0]<start:
                j+=1
            #check to make sure the time given is not more than when the file ends
            if j<len(pairs):
#                print("solar bursts found in",file[n],"for the hour", q, "UT:")
#                back_std_list=background_std(q,n)
                solar_list=solar_bursts(q,n)
#                print("hourly solar burst count: ",len(solar_list))
#                print("background:")
#                print(back_std_list)
#                print("solar bursts:")
#                pretty_print_bursts(solar_list)
                daily_solar.append(solar_list)
            #show the average solar burst intensity per day
            if len(solar_list)>0:
                for d in range(len(solar_list)):
                    sum_int+=solar_list[d][1]
                daily_counts+=len(solar_list)
            q+=1
        #avoid dividing by zero
        if daily_counts>0:
            average_solar=sum_int/daily_counts
        #allow to save the time, intensities in a text file each day (into the same folder
as Find_Raw_Solar.py code)
        if save==1:
            not_split=file[n]
            split_file=not_split.split('.')
            data_file=split_file[0]
            data_file+="txt"
            print("These data will be saved as",data_file)
            print()
            with open(data_file,"w") as new_file:
                json.dump(daily_solar,new_file)

```

```

        print()
        print("Total solar bursts during ",file[n], ":",daily_counts)
        print("Average intensity: ",average_solar)
        n+=1
    print()
    print("***END OF FOLDER**:")
    print("Read in ", n,"file(s) to analyze")

def file_grab(n):
#read in each file as a (time,value) 2D list:
    data=radiojove_spd3.read_radiojove_spx(folder[n])
    table=radiojove_spd3.make_time_data_dict(data)
    pairs=table.items()
    pairs = sorted(pairs, key=lambda pair:pair[0])
    #definitions: time_list=pairs[i][0], values_list=pairs[i][1]

    #smooth the data: rolling average of 10 data points:
    smo_values=[]
    smo_times=[]
    END_TIME=pairs[len(pairs)-10][0]
    x=0
    while pairs[x][0]<END_TIME:
        summation=0
        for n in range(0,10,1):
            if pairs[x][0]<END_TIME:
                summation+=pairs[x+n][1]
        avg=summation/10
        smo_values.append(avg)
        smo_times.append(pairs[x][0])
        x+=1
    smo_pairs=list(zip(smo_times,smo_values))

    #subsample data: change from 10 points per second to 1 point per second
    #choose the 10th value per second:
    smo_sub_values=[]
    smo_sub_times=[]
    smo_sub_values.append(smo_pairs[10][1])
    smo_sub_times.append(smo_pairs[10][0])
    x=20
    while pairs[x][0]<END_TIME:
        smo_sub_values.append(smo_pairs[x][1])
        smo_sub_times.append(smo_pairs[x][0])
        x+=10
    smo_sub=list(zip(smo_sub_times,smo_sub_values))
    #data will be combed through for solar bursts as a smoothed10, subsampled10 file
    return smo_sub

#find background values for every ten minutes of data:
def background_std(time,n):
    pairs=file_grab(n) #smoothed, subsampled data file
    END_TIME=pairs[len(pairs)-1][0] #the exact end time of the file
    start=pairs[0][0] #exact start time of the file
    background_list=[] #create a list for the hour of background values
    std_list=[] #create a list for the standard deviation for each background value
    #increase the start time until it is at the beginning of the desired hour:

```



```

while float(time)>start:
    start+=MINUTE
min_list=[] #background is comprised of minimum values
#initialize minimum value with absurd value:
minimum=10000
ceiling=start+1 #ceiling=end of current hour
#stepping through data point by point to get to beginning of hour
i=0
while pairs[i][0]<start:
    i+=1
for x in range(6): #6 ten-minute intervals= 1 hour
    for y in range(40): #interate 4 times a minute: interate 40 times per 10 minutes
        #check for end of file, end of 1/4*minute interval, end of hour:
        while i<len(pairs) and pairs[i][0]!=END_TIME and pairs[i][0]-
start<=(1/4)*MINUTE and pairs[i][0]<=ceiling:
            #each 1/4 of minute, find the local minimum
            if pairs[i][1]<minimum:
                #make sure the minimum is not an interference spike downward:
                if (pairs[i+1][1]/pairs[i][1])<2:
                    minimum=pairs[i][1]
            i+=1
        if minimum!=10000: #make sure the initial absurd value did not carry through
            min_list.append(minimum) #list of all minimums found
        start+=(1/4)*MINUTE
        minimum=10000 #re-initialize minimum to absurd value

#take out values of minimums that are too far from the others:
if not len(min_list)==0: #make sure minimums were found during the ten minutes
    avg=float(sum(min_list)/len(min_list)) #find the average value of minimums
    new_min=[]
    num=3

    if not len(min_list)<=10: #need a list of at least ten local minimums
        #make a new list of minimums that disclude a value that is too different
from the other minimums
        while len(new_min)<10:
            new_min=[]
            for m in range(len(min_list)):
                if abs(avg-min_list[m])<num:
                    new_min.append(min_list[m])
                num+=1
            if len(min_list)<=10: #cannot take away higher values if the list of minimums
is too small
                new_min=min_list
            background=float((sum(new_min))/len(new_min)) #each background= average of
ten minutes of minimums
            std=np.std(new_min) #standard deviation of the minimums
            background_list.append(background) #6 backgrounds per hour
            std_list.append(std) #6 corresponding standard deviations
            min_list=[] #start over a minimum list for the next iteration
            back_std_list=list(zip(background_list,std_list)) #2D list of all background,
standard deviation per hour
            return back_std_list

#find and return solar burst time, intensity for each hour:

```

```

def solar_bursts(time,n):
    pairs=file_grab(n)
    END_TIME=pairs[len(pairs)-1][0]
    start=pairs[0][0]
    while float(time)>start:
        start+=MINUTE
    solar_value=[] #intensity of each solar burst
    time_solar=[] #corresponding time of each solar burst
    back_std_list=background_std(time,n)
    #definition: background=back_std_list[b][0]
    #definition: sigma=standard deviation=back_std_list[b][1]
    b=0 #iteration of background list
    ceiling=start+1
    i=0
    while pairs[i][0]<start:
        i+=1
#account for possible thunderstorms every thirty minutes:
    first_run=1
    aver_back=0
    for m in range(2):
        continue_analyse=0
        h=i
        #find the average background level for each thirty minutes
        #use the first three background values in an hour for first 30 minute check
        if first_run==1 and h<len(pairs) and pairs[h][0]<=ceiling:
            sum_back=0
            if len(back_std_list)>=3:
                for q in range(3):
                    sum_back+=back_std_list[q][0]
                aver_back=sum_back/3
            #if not a full thirty minutes, use what backgrounds you have:
            else:
                amount=0
                for q in range(len(back_std_list)):
                    sum_back+=back_std_list[q][0]
                    amount+=1
                aver_back=sum_back/amount
        if first_run==0 and h<len(pairs) and pairs[h][0]<=ceiling:
            sum_back=0
            if len(back_std_list)==6:
                for r in range(3):
                    sum_back+=back_std_list[r+3][0]
                aver_back=sum_back/3
            #if not a full hour, use what backgrounds you have:
            else:
                amount=0
                for r in range(len(back_std_list)):
                    sum_back+=back_std_list[r][0]
                    amount+=1
                aver_back=sum_back/amount

    #find the average intensity level each thirty minutes
    length=0 #how many data points
    sum_intensity=0 #summation of all data in a 30 minute window
    #make sure you're within thirty minutes of data

```

```

    while h<len(pairs) and pairs[h][0]!=END_TIME and pairs[h][0]-
start<=THIRTY_MINUTES and pairs[h][0]<=ceiling:
    sum_intensity+=pairs[h][1]
    h+=1
    length+=1
#check to see if the time is still within the hour
if not length==0:
    average_intensity=sum_intensity/length
if length==0:
    average_intensity=1
#if the average intensity value is too much over the background, do not search
for solar
if average_intensity>=1.7*aver_back:
    i=h
    #skip data that is too noisy
    start+=THIRTY_MINUTES
    first_run=0 #second half of the hour
if not average_intensity>=1.7*aver_back:
    continue_analyse=1
    #data is not too noisy: go back and search for solar
#solar burst needs to only be counted if above threshold for >=6 sec:
if continue_analyse==1:
    for y in range(3): #search 30 minutes of data, in 10 minute intervals
        while i<len(pairs) and pairs[i][0]!=END_TIME and pairs[i][0]-
start<=TEN_MINUTES and pairs[i][0]<=ceiling:
            #threshold=background+10*standard deviation
            threshold=(back_std_list[b][0])+(10*back_std_list[b][1])
            count=0
            #see how long the data is above the threshold
            while i<len(pairs) and pairs[i][1]>threshold and pairs[i][0]!=END_TIME:
                count+=1
                i+=1
            #only could be solar if over threshold for 6 seconds
            if count>=6:
                c=i-(count) #c=first value found above background in event
                solar_count=0 #how many solar events are counted
                tested=0 #how many peaks are tested as solar
                possible=[] #possible solar bursts
                interference=0 #test for interference
                #make sure it is increasing
                if pairs[c+1][1]>pairs[c][1]:
                    #check for spike interference at the beginning of event
                    if pairs[c][1]/pairs[c-1][1]<3 and pairs[c+1][1]/pairs[c][1]<3:
                        while pairs[c][0]<pairs[i][0]:
                            p=0
                            if pairs[c][0]<END_TIME and (pairs[c+1][1]-pairs[c][1])>0:
                                #trend increasing:
                                while pairs[c][0]<END_TIME and (pairs[c+1][1]-pairs[c][1])>0:
                                    c+=1
                                    p=c #p=local max
                                    if pairs[c][0]<END_TIME and (pairs[c+1][1]-pairs[c][1])<0:
                                        #trend decreasing:
                                        while pairs[c][0]<END_TIME and (pairs[c+1][1]-pairs[c][1])<0:
                                            c+=1
                                            d=c #d=local min

```

```

        if p==0:
            #trend increasing:
            while pairs[c][0]<END_TIME and (pairs[c+1][1]-pairs[c][1])>0:
                c+=1
                p=c #p=local max
            #record solar activity
            #check if it's the end of the file by checking the presence of local
minimum
            if p!=0 and pairs[p][1]>(pairs[d][1]+10*back_std_list[b][1]):
                #checking for spike interference
                if pairs[p][0]<END_TIME and float(pairs[p][1]/pairs[p-
1][1])<3 and float(pairs[p][1]/pairs[p+1][1])<3:
                    solar_value.append(pairs[p][1])
                    time_solar.append(pairs[p][0])
                    solar_count+=1
                    possible.append(pairs[p][1])
            #get rid of "boxy" interference: no more than 3 bursts per event:
            if p!=0 and pairs[p][1]>(pairs[d][1]+2*back_std_list[b][1]):
                tested+=1
            if len(possible)<=3 and len(possible)>1:
                if possible[0]>possible[(len(possible)-1)]:
                    if possible[0]/possible[(len(possible)-1)]<1.2:
                        interference=1
                if possible[0]<possible[(len(possible)-1)]:
                    if possible[(len(possible)-1)]/possible[0]<1.2:
                        interference=1
            #delete the wrongly appended interference values
            if tested>3 or interference==1:
                for z in range(solar_count):
                    val=(len(solar_value)-1)
                    del solar_value[val]
                    del time_solar[val]

            i+=1
            start+=TEN_MINUTES
            b+=1
            solar_list=list(zip(time_solar,solar_value))
            return solar_list

def pretty_print_bursts(solar_bursts):
    print('[')
    for time, value in solar_bursts:
        print("(%8.5f, %8.3f)," % (time, value))
    print(']')

main()

```