THESIS

DETECTION OF A WEAK RADIOLOGICAL SOURCE IN AMBIENT BACKGROUND USING SPECTRAL ANALYSIS.

Submitted by

Matthew Richard Meengs

Department of Environmental and Radiological Health Sciences

In partial fulfillment of the requirements

For the Degree of Masters of Science

Colorado State University

Fort Collins, Colorado

Summer 2018

Master's Committee:

Advisor: Alex Brandl
Co-advisor: Thomas Johnson

Piotr Kokoszka

ABSTRACT

DETECTION OF A WEAK RADIOLOGICAL SOURCE IN AMBIENT BACKGROUND USING SPECTRAL ANALYSIS

The detection of radiation requires the use of statistical tools due to the probabilistic nature of the emission and interaction properties of radiation, an analysis that includes the testing of a hypothesis regarding the presence or absence of a source against background. Traditionally, a false positive rate of 5% is used to calculate a y*, the decision threshold, above which a source is determined to be present. However, in radiological conditions where a source is both improbable and weak, and where counting time is limited, detection of a source becomes increasingly challenging using this traditional method. The detection of clandestine fissile materials presents such a challenge, and with the increasing risk of nuclear proliferation, there exists a growing desire to research more optimal methods in detecting these sources, especially where a missed detection is of such high consequence.

Previous research has shown that using a string of measurements, and calculating a detection limit based on a certain number of false positives within that string, consistently outperforms the traditional method of basing the detection limit on just one measurement. Such research to date has only been applied to counts of all energies (gross counts). The purpose of this research is to apply the success of this new detection algorithm to certain energies within the spectrum, and to discover whether further optimization is possible by this process. Optimization was evaluated using receiver operator characteristic (ROC) curves, where special emphasis was placed at the lower false positive values.

Over the course of this research, two hypothesis were tested. The first hypothesis conjectures that it is indeed possible to further optimize source detection when using an energy bin other than gross

counts. The second hypothesis postulates that if the first hypothesis is true, than there exists a mathematical criterion that predicts this behavior. Both hypothesis were verified to be correct.

ACKNOWLEDGEMENTS

This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number, 2014-DN-077-ARI091-01. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

TABLE OF CONTENTS

INTRODUCTION

Radioactive decay is the release of energy from an atomic nucleus as it transitions from one state to another. This process occurs through different pathways, each pathway representing a particular type of radiation. Radiations are broadly classified by two characteristics: particle type and energy. In this research, gamma radiations of energies of the order of kilo-electron volts (keV) up to 1711 keV are considered. Radioactive decay is not a deterministic process, as classical physics would suggest. Concurrently, quantum theory has been the most successful at modeling these processes. According to quantum theory, the process of radiation decay is not deterministic; it is impossible to determine when a particular nucleus will undergo radioactive decay. Instead, quantum theory can predict quite accurately the probability that a radionuclide will decay in a given period of time. Thus, the process that governs radioactive decay is fundamentally stochastic.

It is often the case where one wishes to find an unknown radioactive source. This is inherently difficult for two reasons: the presence of ubiquitous background radiation and the stochastic nature of the radioactive decay. Consequently, a signal from a source may not be distinguishable from the background signals since the measurable qualities of these radiations are not necessarily distinctive. In any interval of time, only the probability that a source is present can be determined. Yet in practical applications it is usually necessary to make an unequivocal decision of whether or not a source is present, notwithstanding the probabilistic nature of these processes. When an unequivocal decision is made that a source has been detected, it is called a detection event. Because the successful detection of the source is never certain and always probabilistic, statistical tools must be employed in order to determine whether or not a detection event has taken place.

The statistical tools are traditionally applied to single measurements; each measurement is taken as an isolated event irrespective to prior or future measurements. Symbolically, these single measurements can be represented by the random variable *Y,* where *Y* is always a non-negative integer. There are, however, some scenarios where consecutive or contiguous measurements are by no means irrelevant. For instance, a technician surveying a plot of land for decontamination will undoubtedly gather multiple contiguous measurements. Rather than evaluating each measurement as an isolated event, it is advantageous to define a single event Y_N as the set of *N* total individual measurements. Under this characterization, the event Y_N can be expressed as a set such that $Y_N = \{Y_1, Y_2, \dots, Y_N\}$; (the idea of vectors is intentionally disregarded since the mathematical structure of statistical random variables does not conform to that of a vector space). This model can also apply to measurements from different detectors. For example, the Department of Homeland Security might establish multiple portal monitoring devices at a single checkpoint for the detection of clandestine fissile materials imported in containers. Additionally, this model can also be applied to consecutive measurements. For example, if a technician is daily measuring for the possible leakage of a source, it may be advantageous to consider a set of measurements over multiple days. This may provide a more sensitive test for leakage. In every aforementioned scenario, a sequence of measurements may provide a better model for detection than a cluster of isolated measurements.

The traditional statistical criterion for source detection – source detection via single measurements

When detecting ionizing radiation from a possible source, statistical tools must be employed in order to discern the signal of interest from background. A detection event occurs when a signal other than background is detected. Background refers to any contribution to the radiation signal not originating from the source of interest. The statistical tool most traditionally employed is the Neyman-Pearson lemma [1], a form of hypothesis testing where the error of the first kind is tested such that

$$P(\text{reject } H_0 \mid H_0 \text{ is true}) = \alpha, \tag{1}$$

where H_0 is the null hypothesis and α is the probability of a type I error [2]. For the purposes of radiation detection, the H_0 represents the scenario where no source is present (i.e., only counts from background are assumed to contribute to the signal), and α denotes the probability of a false positive detection. The value for α is a predetermined and fixed parameter, and is traditionally set to 0.05 [3,4], which corresponds to a false positive rate of 5%. Additionally, embedded within the H_0 is the statistical distribution that the background counts are assumed to follow. To reject the H_0 is to declare that a signal originating from a source other than background exists. According to this criterion, the H_0 is rejected when a measurement, represented by the random variable *Y,* exceeds a preset value *y*.* The random variable *Y* is used as an estimator for the true value of background counts. The preset value *y** is defined in ISO 11292 as the decision threshold; it is "the value of the estimator, *Y*, which allows the conclusion that the physical effect is present, if the primary measurement result, *y*, exceeds the decision threshold" [5]. Using these definitions, equation 1 can be more concretely expressed as

$$P(y > y^{*} | H_0 \text{ is true}) = \alpha \tag{2}$$

for the measurement *Y,* where *y* denotes a particular outcome. Figure 1 provides a graphical representation of this concept.

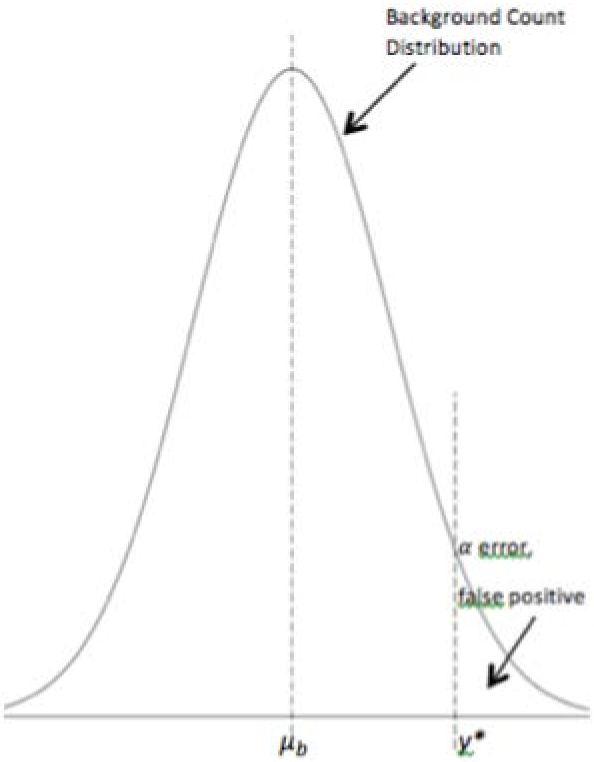


Figure 1: Graphic Illustration of Type I Error with a Gaussian distribution

The value for *y** is both functionally dependent on α and contingent upon the assumed background distribution. For the purpose of this research, it is assumed that background radiation takes a Poisson distribution. At sufficiently high counts, the Poisson distribution is assumed to approximate a Gaussian distribution. Figure 1 shows a representation of the Gaussian distribution. The false negative probability, traditionally denoted as β, is beyond the scope of this research and will not be investigated. Since two different distributions are used for this research, two different methods of calculating *y** are required.

Calculating y* using a Gaussian distribution

Using the criterion given by equation (2), the decision threshold value *y** can be calculated such that

$$P(y > y^*) = \alpha = \int_{y^*}^{\infty} \phi_{bkgd}(y')dy' = 1 - \int_{-\infty}^{y^*} \phi_{bkgd}(y')dy' \tag{3}$$

where ϕ_{bkgd} is the background probability density function of the continuous random variable *Y*. The Gaussian distribution is the only continuous distribution that is assumed in this research; the equation is given by equation 4:

$$\phi_{bkgd}(y) = \frac{1}{\sigma_b\sqrt{2\pi}} \exp\left[-\left(\frac{1}{2\sigma_b^2}\right)(y - \mu_b)^2\right] \tag{4}$$

where μ_b and σ_b are the parameters representing the average background counts and the standard deviation of the background counts, respectively [6]. When ϕ_{bkgd} assumes a Gaussian distribution, the solution to equation 3 has no closed form, and requires the use of Z tables. Equation 3 then simplifies to

$$y^* = \mu_b + \sigma_b Z(\alpha) \tag{5}$$

where $Z(\alpha)$ represents the Z value for a normal distribution in terms of the single-tail probability α. Equation 5 can be further simplified to the familiar equation

$$y^* = k_\alpha \sigma_b \tag{6}$$

where k_α=1.645, which is the single tailed *Z* value for α=0.05 and μ_b=0, indicating that the *y** in equation 6 denotes net counts [2]. In this research, the more generalized form given by equation 5 is used since multiple values of α will be needed. The Excel software easily performs the computation required by equation 5.

Calculating y* using a Poisson distribution

The discrete analog for equation 3 is

$$P(y > y^*) = \alpha = \sum_{y_i = y^*+1}^{\infty} p_{bkgd}(y_i) = 1 - \sum_{y_i=0}^{y^*} p_{bkgd}(y_i) \tag{7}$$

where p_{bkgd} is the probability mass function for the discrete random variable *Y.* The Poisson distribution is the only discrete distribution that is assumed in this research, and is given by equation 8:

$$p_{bkgd}(y) = \frac{\mu_b^y e^{-\mu_b}}{y!}\,, \tag{8}$$

where $p_{bkgd}(y)$ is the probability of exactly *y* counts being detected, and $\mu_b$is the parameter representing the average background counts [6]. When p_{bkgd} assumes a Poisson distribution, equation 7 becomes

$$P(y > y^*) = \alpha = 1 - \sum_{i=0}^{y^*} \frac{\mu_b^i e^{-\mu_b}}{i!}. \tag{9}$$

Recall that $P(y > y^*) \; + \; P(y \leq y^*) \; = \; 1$, thus, equation 9 is equivalently expressed as

$$P(y \leq y^*) \; = \; 1 - \alpha \; = \sum_{i=0}^{y^*} \frac{\mu_b^i e^{-\mu_b}}{i!}. \tag{10}$$

In order for the equality given by equation 10 to hold for all values of *y**, *y** must be a real number. The allowance of non-integer values for *y** makes equation 10 very unwieldy, especially when the calculation of *y** is necessary. Since counts take only discrete values, it is expedient to restrict the domain of *y** to the non-negative integers. A complication arises, however, if *y** assumes only discrete values when α is erstwhile a real number because the equality no longer holds in general. Given this new restraint, a more precise treatment of *y** is given by equation 11:

$$\sum_{i=0}^{y^*} \frac{\mu_b^i e^{-\mu_b}}{i!} \; \leq \; 1 - \alpha \; < \; \sum_{i=0}^{y^*+1} \frac{\mu_b^i e^{-\mu_b}}{i!}. \tag{11}$$

Since $e^{-\mu_b}$ is constant, and its inverse e^{μ_b} is always greater than zero, equation 11 can be alternatively expressed as

$$\sum_{i=0}^{y^*} \frac{\mu_b^i}{i!} \leq e^{\mu_b}(1-\alpha) < \sum_{i=0}^{y^*+1} \frac{\mu_b^i}{i!}. \tag{12}$$

For the purpose of computation, it is more intuitive to define *y** within the one sided inequality

$$e^{\mu}(1-\alpha) < \sum_{i=0}^{n+1} \frac{\mu_b^i}{i!}, \tag{13}$$

such that $n \in \{y^*, y^* + 1, y^* + 2, \ldots\}$ where *y** is defined as the lowest bound of the solution set (i.e., it is the first integer to satisfy the inequality). Solving for the Poisson *y** in this manner is more complicated than solving for the Gaussian *y** using equation 5. Consequently, the computation of the Poisson *y** required the fabrication of a computational sheet in Excel. A description of this process is discussed in detail in Appendix F.

The extended statistical criterion for source detection – source detection via a string of measurements

For most applications, the criterion for source detection given by equation 2 is adequate for individual measurements of a sufficiently long counting time. In radiological conditions, however, where a source is both improbable and weak, and where counting time is limited, detection of a source becomes increasingly challenging using the traditional criterion. This challenge is especially pronounced with the detection of clandestine fissile materials, a scenario that is becoming increasingly relevant with the increasing risk of nuclear proliferation. Hence, there exists a desire to research more optimal methods. One such method involves comparing the ratios of different energies in a two window system (i.e., two energy groups, high energy and low energy) [7]. The method studied in this research involves using the above criterion for not just one measurement, but multiple measurements.

The traditional criterion given by equation 2, expressed as event *Y*, is used for only one measurement at a time. When taking into account a string of measurements, however, the false positive probability in terms of *y** is expressed as

$$P_{n/N} \ = P(y_1, y_2, \ldots, y_n > y^* | y_{n+1}, \ldots, y_N \leq y^* \mid H_0) = \ \alpha. \tag{14}$$

In the criterion given by equation 14, when *n* out of *N* measurements exceed *y**, the string of measurements given by Y_N is considered to originate from a distribution other than background, and hence Y_N is considered a detection event. The probability of *n* out of *N* measurements exceeding *y** is expressed as a binomial distribution:

$$P_{n/N}(y_1, y_2, \ldots, y_n > y^* | y_{n+1}, \ldots, y_N \leq y^*) = \binom{N}{n} \alpha_{ind}^{n} (1 - \alpha_{ind})^{N-n} \tag{15}$$

from whence the following is obtained:

$$\alpha = \ \binom{N}{n} \alpha_{ind}^{n} (1 - \alpha_{ind})^{N-n}, \tag{16}$$

where α_{ind} is the calculated false positive probability for an individual measurement and $\binom{N}{n} \ = \frac{N!}{n!(N-n)!}$. It is essential to remember that α_{ind} is *not* the type I error probability for the event Y_N, instead, it is the type I error probability required of each individual measurement Y_i such that equation 14 is satisfied. The combinatorial factor $\binom{N}{n}$ in equations 15 and 16 exists because the order of which the *n* out of *N* measurements exceed *y** is not considered relevant. In evaluating the decision threshold from a series of *N* measurements, it is impractical to require that exactly *n* out of *N* measurements exceed *y*.* Indeed, if *n* out of *N* measurements exceeding *y** is considered a detection event, than certainly an excess of *n* measurements exceeding *y** ought also be considered a detection event. Instead, the probability that at least *n* out of *N* measurements exceed the decision threshold is computed. Since the individual events of exactly *n* of *N* measurements exceeding the threshold are mutually exclusive for different values of *n*, the union of these events is expressed as the sum of the individual probabilities such that

$$P_{n+/N} = \sum_{i=n}^{N} P_{i/N}. \tag{17}$$

Following substitution, the final equation of

$$P_{n+/N} = \sum_{i=n}^{N} \binom{N}{i} \alpha_{ind}^{i} (1 - \alpha_{ind})^{N-i} = \alpha \tag{18}$$

is obtained, where *N* and *n* are both positive integers and α is a real number such that $n \in [0, N]$ and $\alpha \in [0,1]$. Henceforth, a "data string metric" will refer to any combination of *n* and *N* (e.g., 1/5, 2/5, 3/5, 4/5, or 5/5). Equation 18 expresses the necessary relationship between α and α_{ind} for a given data string metric *n*/*N* such that when

$$P(y > y^*) = \alpha_{ind}$$

for an individual measurement Y_i, then

$$P_{n+/N} = \alpha$$

for the measurement Y_N with the *n*/*N* data string metric. Observe that for the data string metric given by 1/1, equation 18 simplifies to α_{ind} = α. This is the trivial solution and represents the traditional criterion given by equation 2. Hence, the alternative criterion heretofore mentioned is simply a binomial extension of the traditional criterion. This extended criterion is given by equation 19:

$$P_{n+/N} = \sum_{i=n}^{N} P(y_1, y_2, \dots, y_i > y^* \mid y_{i+1}, \dots, y_N \leq y^* \mid H_0) = \alpha \tag{19}$$

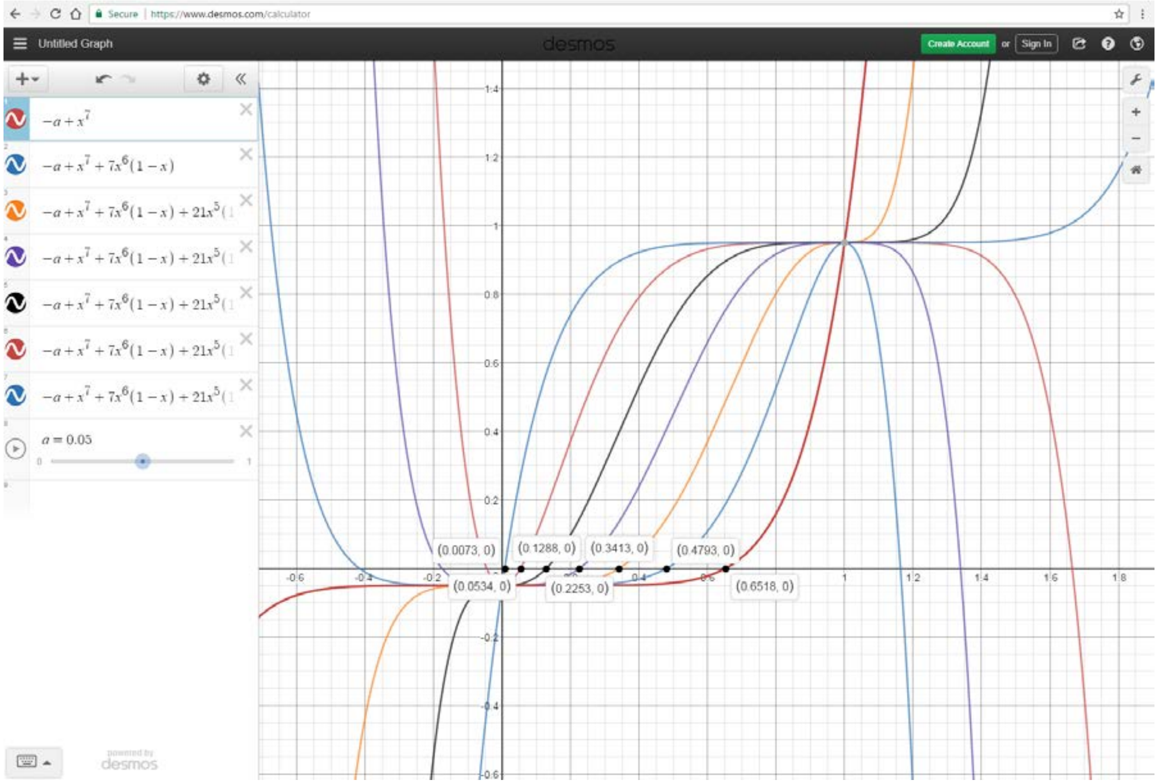
where *y** is functionally dependent on α_{ind}. Therefore, the computation of *y** in equation 19 requires the substitution of α_{ind} for α in equations 5 and 13 for the Gaussian and Poisson distributions, respectively.

Obtaining a function for $\boldsymbol{\alpha_{ind}}$

In order to use the criterion given by equation 19, it is necessary to be able to solve for α_{ind}. Equation 18 is always a polynomial of degree *N*, and as a general rule, polynomials of degree two or greater are not invertible. Within the restricted domain of α, however, equation 18 is invertible, and thus well defined. As such, α_{ind} can be expressed as an explicit function of α for a given data string metric. Solving for α_{ind} in equation 18 cannot be accomplished analytically. For this research, DESMOS online graphing calculator [8] was used to find the roots of the equation

$$0 = -a + \sum_{i=n}^{N} \binom{N}{i} x_{ind}^{i} (1 - x_{ind})^{N-i}. \tag{20}$$

The constant *a* is used in place of α, and is set to a slider bar in the calculator; its value can be set to vary between 0 and 1. Additionally, the variable *x* is used in place of α_{ind}. This is shown in Figure 2 where seven curves are given for *N*=7.



https://www.desmos.com/calculator

$-a+x^7$

$-a+x^7+7x^6(1-x)$

$-a+x^7+7x^6(1-x)+21x^5(1$

$-a+x^7+7x^6(1-x)+21x^5(1$

$-a+x^7+7x^6(1-x)+21x^5(1$

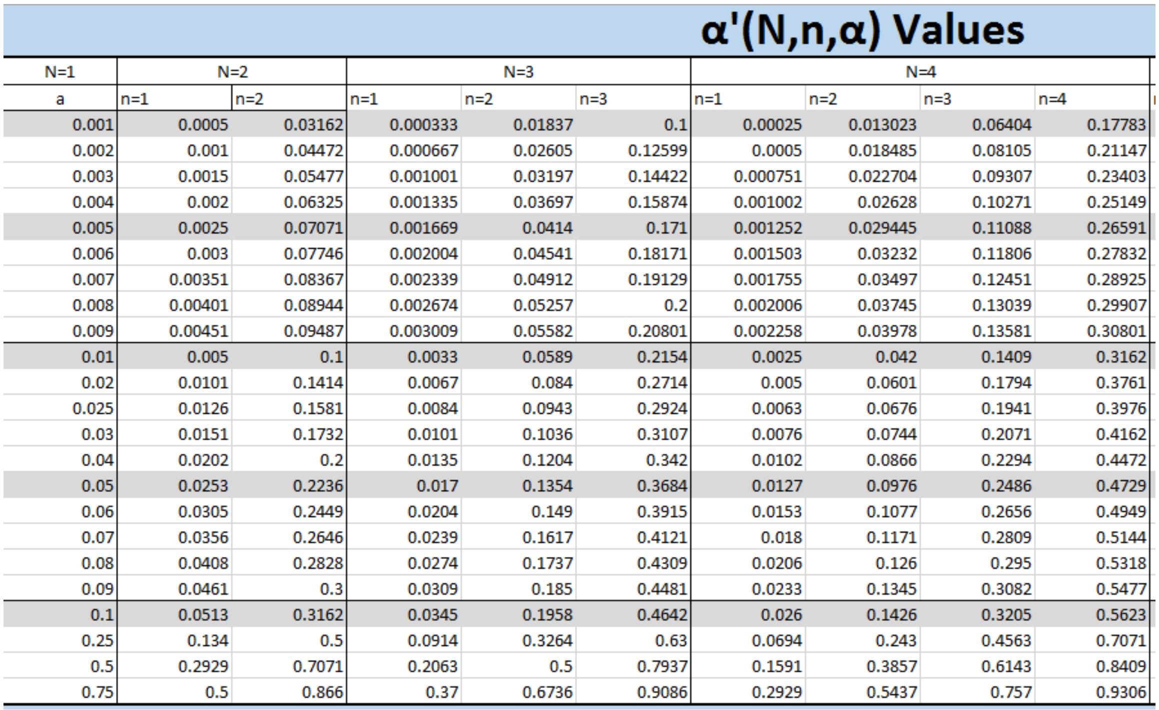
$-a+x^7+7x^6(1-x)+21x^5(1$

$-a+x^7+7x^6(1-x)+21x^5(1$

$a = 0.05$

Figure 2: DESMOS online graphing calculator [8]

The curves from left to right represent the data string metrics of 1/7, 2/7, 3/7, 4/7, 5/7, 6/7, and 7/7 with the a slide bar set to 0.05. Each root displayed in Figure 2 is the α_{ind} for each respective data string metric for *N*=7 and α = 0.05. Observe that all curves pass the horizontal line test within the unit interval, thus providing visual proof of the invertability of equation 18 within its proper domain. Since the function $\alpha_{ind}(\alpha; n, N)$ cannot be expressed analytically, its expression is given in table format, thus giving rise to the alpha table shown in Figure 3. The actual table is stored in an Excel spreadsheet. The table is truncated between *N*=4 and *N*=5, but its values extend all the way to *N*=10 for all values of *n.* The solutions to α_{ind} are computed for 23 different values of α. These preselected values of α are given in the far left column in Figure 3.



α'(N,n,α) Values

N=1	N=2		N=3			N=4			
a	n=1	n=2	n=1	n=2	n=3	n=1	n=2	n=3	n=4
0.001	0.0005	0.03162	0.000333	0.01837	0.1	0.00025	0.013023	0.06404	0.17783
0.002	0.001	0.04472	0.000667	0.02605	0.12599	0.0005	0.018485	0.08105	0.21147
0.003	0.0015	0.05477	0.001001	0.03197	0.14422	0.000751	0.022704	0.09307	0.23403
0.004	0.002	0.06325	0.001335	0.03697	0.15874	0.001002	0.02628	0.10271	0.25149
0.005	0.0025	0.07071	0.001669	0.0414	0.171	0.001252	0.029445	0.11088	0.26591
0.006	0.003	0.07746	0.002004	0.04541	0.18171	0.001503	0.03232	0.11806	0.27832
0.007	0.00351	0.08367	0.002339	0.04912	0.19129	0.001755	0.03497	0.12451	0.28925
0.008	0.00401	0.08944	0.002674	0.05257	0.2	0.002006	0.03745	0.13039	0.29907
0.009	0.00451	0.09487	0.003009	0.05582	0.20801	0.002258	0.03978	0.13581	0.30801
0.01	0.005	0.1	0.0033	0.0589	0.2154	0.0025	0.042	0.1409	0.3162
0.02	0.0101	0.1414	0.0067	0.084	0.2714	0.005	0.0601	0.1794	0.3761
0.025	0.0126	0.1581	0.0084	0.0943	0.2924	0.0063	0.0676	0.1941	0.3976
0.03	0.0151	0.1732	0.0101	0.1036	0.3107	0.0076	0.0744	0.2071	0.4162
0.04	0.0202	0.2	0.0135	0.1204	0.342	0.0102	0.0866	0.2294	0.4472
0.05	0.0253	0.2236	0.017	0.1354	0.3684	0.0127	0.0976	0.2486	0.4729
0.06	0.0305	0.2449	0.0204	0.149	0.3915	0.0153	0.1077	0.2656	0.4949
0.07	0.0356	0.2646	0.0239	0.1617	0.4121	0.018	0.1171	0.2809	0.5144
0.08	0.0408	0.2828	0.0274	0.1737	0.4309	0.0206	0.126	0.295	0.5318
0.09	0.0461	0.3	0.0309	0.185	0.4481	0.0233	0.1345	0.3082	0.5477
0.1	0.0513	0.3162	0.0345	0.1958	0.4642	0.026	0.1426	0.3205	0.5623
0.25	0.134	0.5	0.0914	0.3264	0.63	0.0694	0.243	0.4563	0.7071
0.5	0.2929	0.7071	0.2063	0.5	0.7937	0.1591	0.3857	0.6143	0.8409
0.75	0.5	0.866	0.37	0.6736	0.9086	0.2929	0.5437	0.757	0.9306

Figure 3: The Alpha Table – used to define the $\boldsymbol{\alpha_{ind}(\alpha; n, N)}$ function

The 23 values for α define the domain of the independent variable α for the $\alpha_{ind}(\alpha; n, N)$ function which is defined as the set *A* = {0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01, 0.02, 0.025, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.25, 0.50, 0.75}. The function is defined as follows:

$$\alpha_{ind}(\alpha; n, N) \coloneqq \text{Alpha Table (Figure 3) such that}$$

$$N \in [1,10]$$

$$n \in [1, N]$$

$$\alpha \in A.$$

The set A consists of three decades of values: 0.001-0.01, 0.01-0.1, and 0.1-0.75. Emphasis is placed on the lower values of α. The values 0.25, 0.50, and 0.75 are calculated for the sake of completeness. Multiple values for α are necessary for the computation of Receiver Operator Characteristics (ROC) curves.

ROC curves

ROC curves are employed to provide a visual representation of the relationship between the false positive rate r_b and the true positive rate r_s for various false positive (α) values [4]. Note that the expected value of r_b is α (E $[r_b]$ = α). An example of ROC curves are shown in Figure 4 for all data string metrics where $N \leq 5$.

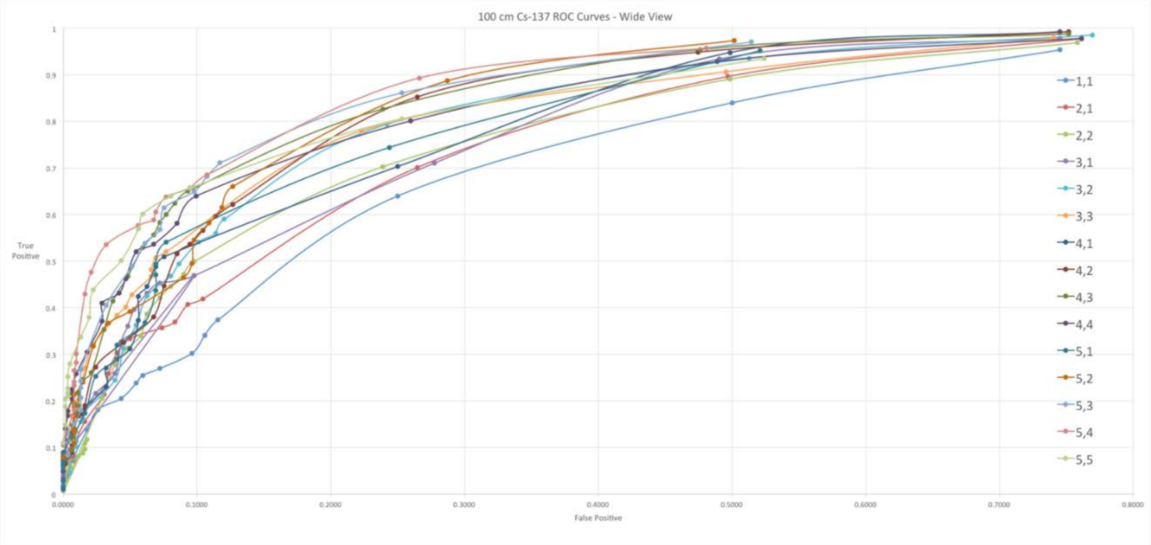


Figure 4: 100 cm ^{137}Cs ROC curves. A concave up shape indicates superior performance (i.e., a higher true positive rate for a given false positive rate).

The region of an ROC curve is the square of the unit interval, with the false positive rate along the horizontal axis and the true positive rate along the vertical axis. The points defining the ROC curves are composed of the ordered pairs (r_b, r_s). These rates are calculated from the total number of false or true positives from a total of 2200 1-s spectra divided by 2200. Recall that as a control, the total number of

1-s spectra is consistently 2200 throughout the entirety of this research. Each curve is a function of $\alpha \in A$, with a particular data string metric as a parameter. An ROC curve resembling the *y=x* line would be indicative of a complete failure to discriminate a source term from background. An ROC curve that is concave up relative to the *y=x* line indicates the ability to discriminate a source from background where an increase in concavity is indicative of superior performance. Throughout the course of this research, performance will be measured by the concavity of its ROC curves. In more quantitative terms, performance is measured by the true positive rate for a given false positive rate. A more thorough treatment of ROC curves is given in reference [4].

Previous results

Previous research has shown, and Figure 4 corroborates, that this new criterion consistently outperforms the traditional method [9], (recall that the traditional method is simply the 1/1 data string metric). Research has also shown that the least optimal data string metrics tend to be both 1/*N* and *N*/*N*, with the most optimal data string metrics given by *n*/*N* with $n \gtrsim \frac{N}{2}$. All previous results obtained from this method, including the aforementioned results in Figure 4, were obtained from gross counts only. The objective of this research is to apply this method to spectral data, and to test this criterion to specific groups of energies within the spectrum. The hypothesis is that further optimization for detecting a weak source is possible when honing in on specific groups of energies. Thus, ROC curves will be formulated not just for a particular data string metric, but also for specific energy groups.

Uncertainty calculation

Each point in the ROC curve has two associated uncertainties: the false positive uncertainty and the true positive uncertainty. The calculation of uncertainty requires knowledge of the statistical distribution. Counting statistics always assumes a Poisson distribution. The assumption in this research is that both the false positive rates and true positive rates assume a Poisson distribution. In other words, the Poisson distribution is assumed to be maintained through the algorithm defined by the extended criteria given by equation 19. In Poisson statistics, the variance is equal to the mean. In this case, the mean is the number of false or true positives in a set of spectra. The number of false or true positives is given by the product $r \cdot T$, where *r* is either the false positive rate or the true positive rate and likewise *T* is either the number of background spectra or the number of source spectra, respectively. As previously mentioned, the values for *r* are the values on each ordered pair of the ROC curve (r_b, r_s). The standard deviation of the number of false or true positives in a total of *T* spectra is thus

$$\sigma_{pos/T} \; = \sqrt{r \cdot T}. \tag{21}$$

Following normalization, the equation for the absolute uncertainty of the positive rate is given below:

$$\sigma_{abs} = \frac{\sqrt{r \cdot T}}{T} = \sqrt{\frac{r}{T}}. \tag{22}$$

Each point of the ROC curve will include both the false positive rate and the true positive rate uncertainty. Thus, the points in the ROC curve will be expressed as the following ordered pair with uncertainty: $\left(r_b \pm \sqrt{\frac{r_b}{T_b}} \;, r_s \pm \sqrt{\frac{r_s}{T_s}}\right)$. Observe that for very low α values, r_b may equal zero, and thus the uncertainty will also be equal to zero. This may initially seem to indicate perfect results, i.e., 0% false positive rate with 100% certainty. This is, however, an incorrect interpretation. It becomes imperative to consider the relative uncertainty. The equation for the relative uncertainty of the positive rate is given by equation 23:

$$\sigma_{rel} = \frac{\sigma_{abs}}{r} = \frac{1}{r}\sqrt{\frac{r}{T}} = \sqrt{\frac{1}{r \cdot T}}. \tag{23}$$

Note that when the absolute uncertainly is equal to zero, the relative uncertainly becomes infinite. This conforms with intuition; a zero false positive rate with a non-zero true positive rate at zero uncertainty is not possible. For this reason, the zero error associated with the data points that fall along the vertical axis is to be disregarded. In this research, T=2200.

Statement of Hypothesis

Two hypothesis will be tested. The first hypothesis conjectures that it is possible to further optimize source detection when using an energy bin other than gross counts. An additional parameter, E, will be introduced. The parameter E will represent a subset of energies from the total spectrum. The second hypothesis postulates that if the first hypothesis is true, than there exists a mathematical criterion that predicts this behavior.

MATERIALS AND METHODS

The following discussion will proceed in three parts. Part I will discuss data acquisition. Part II will deal with the development of the computational tool used to analyze these data for optimization. This tool essentially bins energies and generates ROC curves. Part III will discuss the systematic method by which the tool is used to achieve optimal results.

Part I: Data acquisition

The data acquired for this analysis were taken with an open window 2 × 2 NaI scintillation detector (Model 902, Canberra Industries Inc., Meriden, CT). The resolution of the 662 keV ^{137}Cs photopeak was 8.5%. The detector was connected to a photomultiplier tube (Model 2007P, Canberra Industries Inc. Meriden, CT) with a preamplifier. The preamplification signal was digitally modified by a Lynx Digital Signal Processor (Canberra Industries Inc., Meriden, CT). The energies were binned in 1024 channels using the Prospect software as the multi-channel analyzer (MCA). Prospect then exported these data to an Excel spreadsheet. Each spectrum contains data for 1 second in measurement duration, and thus many spectra were taken to span about an hour of time. Previous research has shown that background varies significantly throughout the day; therefore, background spectra are taken continuously for several hours, from late morning to late afternoon, when the background is relatively low. Once this is accomplished, any set of source spectra is matched with a set of background spectra taken at roughly the same time of day. For the sake of consistency, the geometries are kept constant for all measurements, with the exception of distance. All measurements were taken in room 312 in the MRB building at Colorado State University. The laboratory set up is shown in Figure 5.

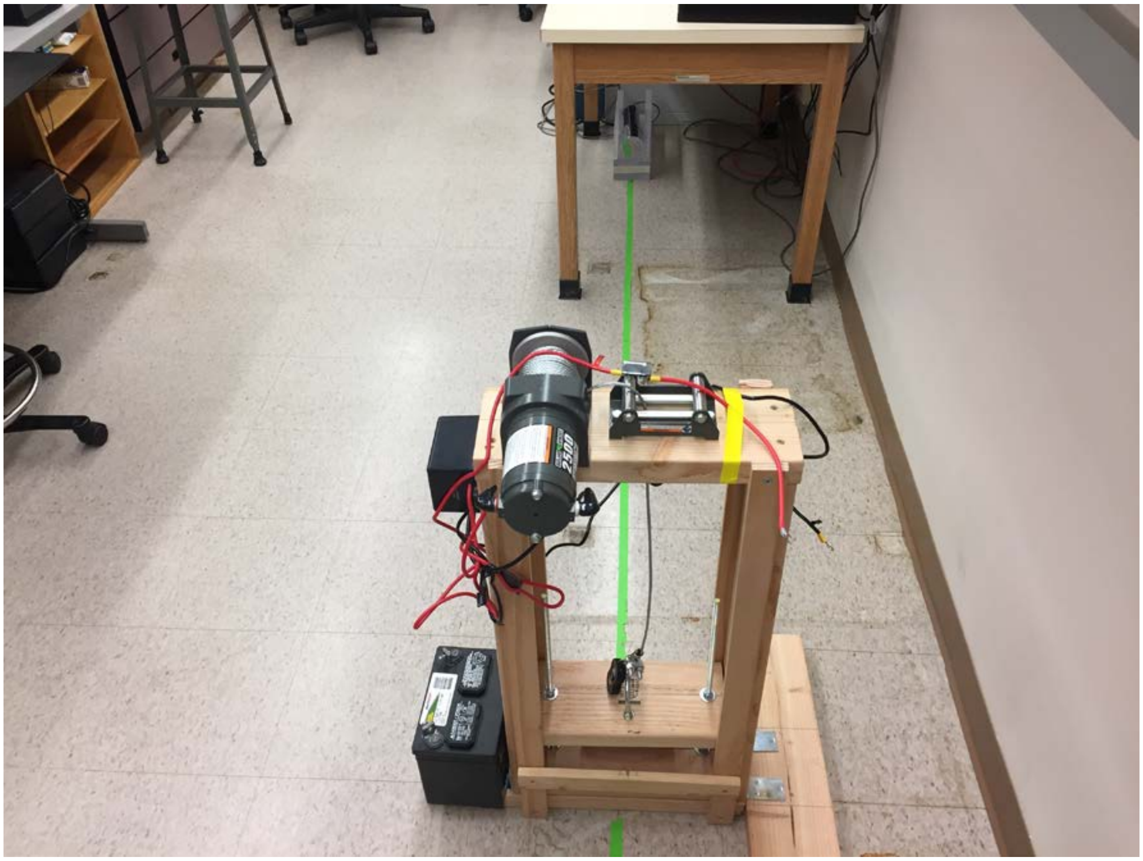


Figure 5: Data Collection Setup, Room 312 MRB Building, Colorado State University

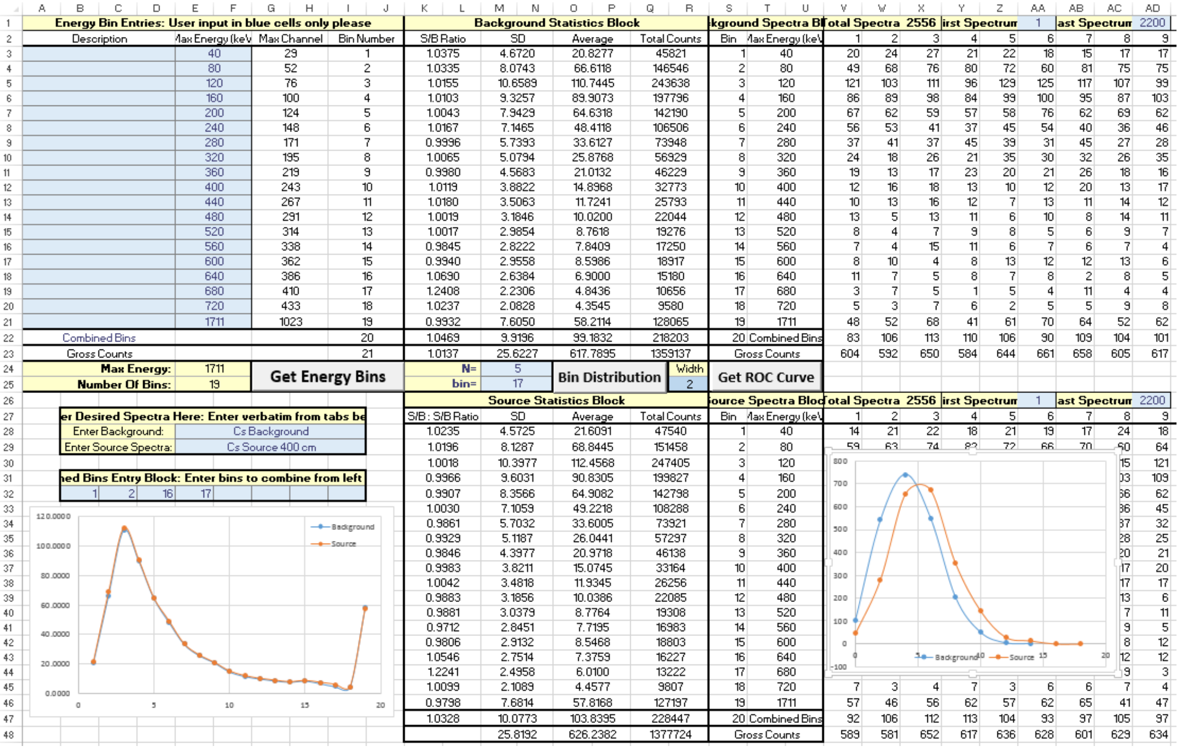
A set of 2200 1-s duration spectra is used in this research for all source strengths. These spectra are intended to be analyzed in a sequence of length *N.* It is important to realize that the value *N* is not the total number of 2200 spectra (i.e., $N \neq 2200$). Instead, *N* is a string of consecutive 1-s measurements and is a subset of the total number of spectra. An arbitrary 1-s measurement within the set of 2200 measurements is denoted by Y_i. For instance, the first string of length *N* is given by the sequence $\{Y_1, Y_2, \dots, Y_N\}$. An arbitrary string of length *N* is given by the set $\{Y_i, Y_{i+1}, \dots, Y_{i+N-1}\}$ provided that $1 \leq i \leq (2200 - N + 1)$. Near the end of the sequence when $(2200 - N + 1) < i \leq 2200$, the spectra loop to include the first values in order to obtain a string of length *N*, (e.g., $\{Y_{2199}, Y_{2200}, Y_1, Y_2, \dots, Y_{N-2}\}$). Using this pattern, there are a total of 2200 strings of length *N*. Each of these strings represents an event Y_N, with a total of 2200 events to be tested.

Part II: Computational tool for analysis

Due to the computational rigors of this research, it was necessary to develop a computational tool designed to bin energies and graph ROC curves with ease. The tool uses Excel 2013 as its software in addition to the incumbent VBA programing language. The Excel file is entitled "ROC_curve_generator." The tool utilizes many different types of sheets; these sheets are: User interface sheet, spectra sheets, the Poisson *y** calculator sheet, and the alpha prime and Z value sheet. Each of these sheet types will be discussed in turn.

User Interface Sheet:

As the name suggests, this is the main sheet and is called "Energy Bins 2," (see Figure 6).



	A	B	C	D	E	F	G	H	I	J	K L	M N	O P	Q R	S	T U	V	W	X	Y	Z	AA	AB	AC	AD
1		Energy Bin Entries: User input in blue cells only please									Background Statistics Block				kground Spectra Bl		otal Spectra		2556	irst Spectrum		1	ast Spectrum		2200
2		Description			Max Energy (keV		Max Channel		Bin Number		S/B Ratio	SD	Average	Total Counts	Bin	Max Energy (keV	1	2	3	4	5	6	7	8	9
3					40		29		1		1.0375	4.6720	20.8277	45821	1	40	20	24	27	21	22	18	15	17	17
4					80		52		2		1.0335	8.0743	66.6118	146546	2	80	49	68	76	80	72	60	81	75	75
5					120		76		3		1.0155	10.6589	110.7445	243638	3	120	121	103	111	96	129	125	117	107	99
6					160		100		4		1.0103	9.3257	89.9073	197796	4	160	86	89	98	84	99	100	95	87	103
7					200		124		5		1.0043	7.9429	64.6318	142190	5	200	67	62	59	57	58	76	62	69	62
8					240		148		6		1.0167	7.1465	48.4118	106506	6	240	56	53	41	37	45	54	40	36	46
9					280		171		7		0.9996	5.7393	33.6127	73948	7	280	37	41	37	45	39	31	45	27	28
10					320		195		8		1.0065	5.0794	25.8768	56929	8	320	24	18	26	21	35	30	32	26	35
11					360		219		9		0.9980	4.5683	21.0132	46229	9	360	19	13	17	23	20	21	26	18	16
12					400		243		10		1.0119	3.8822	14.8968	32773	10	400	12	16	18	13	10	12	20	13	17
13					440		267		11		1.0180	3.5063	11.7241	25793	11	440	10	13	16	12	7	13	11	14	12
14					480		291		12		1.0019	3.1846	10.0200	22044	12	480	13	5	13	11	6	10	8	14	11
15					520		314		13		1.0017	2.9854	8.7618	19276	13	520	8	4	7	9	8	5	6	9	7
16					560		338		14		0.9845	2.8222	7.8409	17250	14	560	7	4	15	11	6	7	6	7	4
17					600		362		15		0.9940	2.9558	8.5986	18917	15	600	8	10	4	8	13	12	12	13	6
18					640		386		16		1.0690	2.6384	6.9000	15180	16	640	11	7	5	8	7	8	2	8	5
19					680		410		17		1.2408	2.2306	4.8436	10656	17	680	3	7	5	1	5	4	11	4	4
20					720		433		18		1.0237	2.0828	4.3545	9580	18	720	5	3	7	6	2	5	5	9	8
21					1711		1023		19		0.9932	7.6050	58.2114	128065	19	1711	48	52	68	41	61	70	64	52	62
22		Combined Bins							20		1.0469	9.9196	99.1832	218203	20	Combined Bins	83	106	113	110	106	90	109	104	101
23		Gross Counts							21		1.0137	25.6227	617.7895	1359137		Gross Counts	604	592	650	584	644	661	658	605	617
24		Max Energy:			1711		Get Energy Bins				N=	5	Bin Distribution	Width	Get ROC Curve										
25		Number Of Bins:			19						bin=	17		2											
26											Source Statistics Block				ource Spectra Bloc		otal Spectra		2556	irst Spectrum		1	ast Spectrum		2200
27		er Desired Spectra Here: Enter verbatim from tabs be									S/B : S/B Ratio	SD	Average	Total Counts	Bin	Max Energy (keV	1	2	3	4	5	6	7	8	9
28		Enter Background:			Cs Background						1.0235	4.5725	21.6091	47540	1	40	14	21	22	18	21	19	17	24	18
29		Enter Source Spectra:			Cs Source 400 cm						1.0196	8.1287	68.8445	151458	2	80	59	63	74	82	72	66	70	60	64
30											1.0018	10.3977	112.4568	247405	3	120								15	121
31		hed Bins Entry Block: Enter bins to combine from left									0.9966	9.6031	90.8305	199827	4	160								03	109
32		1	2	16	17						0.9907	8.3566	64.9082	142798	5	200								66	62
33											1.0030	7.1059	49.2218	108288	6	240								36	45
34											0.9861	5.7032	33.6005	73921	7	280								37	32
35											0.9929	5.1187	26.0441	57297	8	320								28	25
36											0.9846	4.3977	20.9718	46138	9	360								20	21
37											0.9983	3.8211	15.0745	33164	10	400								17	20
38											1.0042	3.4818	11.9345	26256	11	440								17	17
39											0.9883	3.1856	10.0386	22085	12	480								13	6
40											0.9881	3.0379	8.7764	19308	13	520								7	11
41											0.9712	2.8451	7.7195	16983	14	560								9	5
42											0.9806	2.9132	8.5468	18803	15	600								8	12
43											1.0546	2.7514	7.3759	16227	16	640								12	12
44											1.2241	2.4958	6.0100	13222	17	680								9	3
45											1.0099	2.1089	4.4577	9807	18	720	7	3	4	7	3	6	6	7	4
46											0.9798	7.6814	57.8168	127197	19	1711	57	46	56	62	57	62	65	41	47
47											1.0328	10.0773	103.8395	228447	20	Combined Bins	92	106	112	113	104	93	97	105	97
48												25.8192	626.2382	1377724		Gross Counts	589	581	652	617	636	628	601	629	634

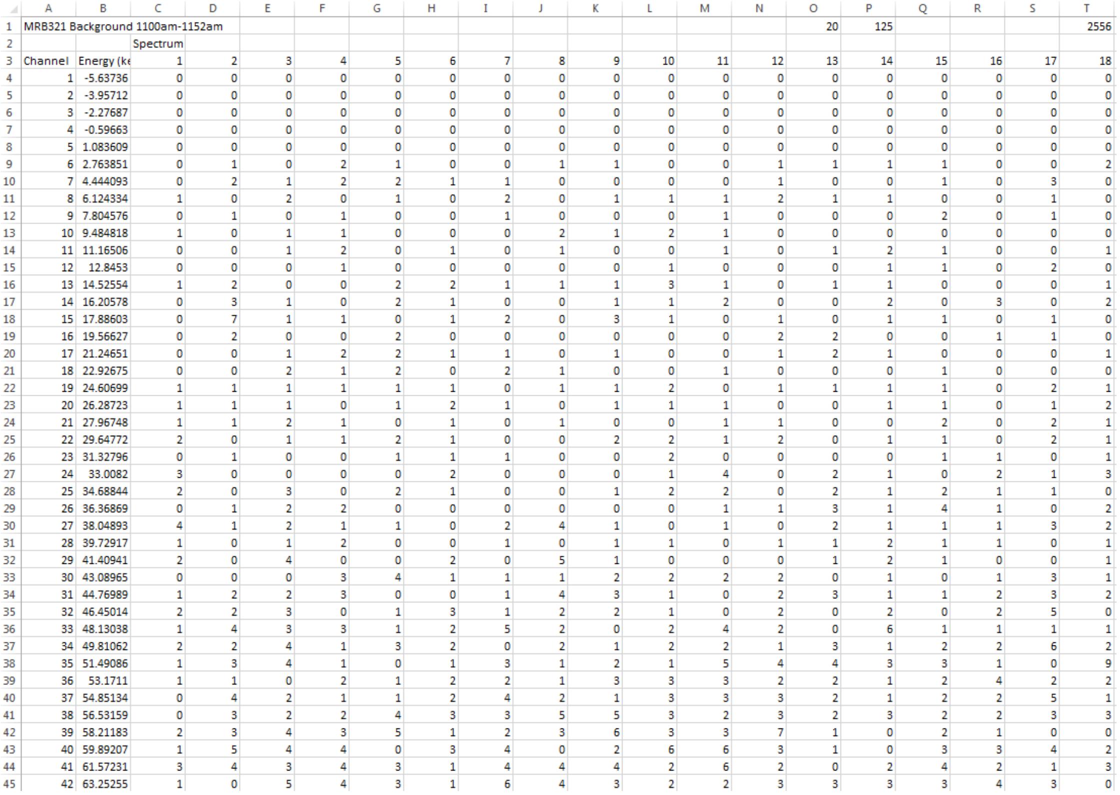
Figure 6: User Interface Sheet

The cells are color coded to provide clarity to the user. Yellow cells are label cells that provide ease of navigation. The blue cells are user entry cells. Under no circumstances should the user enter anything in a non-blue cell, as this action may crash the program. The white cells are data and data heading cells.

Gray cells with orange text are calculation output cells; these include the individual ordered pairs for the ROC curves and the error calculations for each point in the ROC curve. The sheet also includes three buttons: The “Get Energy Bins” button, the “Bin Distribution” button, and the “Get ROC Curve” button. Each button is linked to an Excel macro code that performs the necessary computations for that task. These buttons execute the program, and the user is strongly advised to consult the procedures in Appendix A prior to execution. The VBA codes for these buttons are provided in Appendix C, Appendix D, and Appendix E, respectively.

Spectra Sheets:

The spectra sheets are the sheets where the actual data reside. There can be any number of spectra sheets. The sheets should be intuitively labeled, such as “Cs background,” or “Cs source 400 cm.” The sheets must be properly formatted in accordance with Procedure 1 found in Appendix A. Figure 7 provides an example of a spectra sheet.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	MRB321 Background 1100am-1152am														20	125				2556
2			Spectrum																	
3	Channel	Energy (ke	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
4	1	-5.63736	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	2	-3.95712	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	3	-2.27687	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	4	-0.59663	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	5	1.083609	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	6	2.763851	0	1	0	2	1	0	0	1	1	0	0	1	1	1	1	0	0	2
10	7	4.444093	0	2	1	2	2	1	1	0	0	0	0	1	0	0	1	0	3	0
11	8	6.124334	1	0	2	0	1	0	2	0	1	1	1	2	1	1	0	0	1	0
12	9	7.804576	0	1	0	1	0	0	1	0	0	0	1	0	0	0	2	0	1	0
13	10	9.484818	1	0	1	1	0	0	0	2	1	2	1	0	0	0	0	0	0	0
14	11	11.16506	0	0	1	2	0	1	0	1	0	0	1	0	1	2	1	0	0	1
15	12	12.8453	0	0	0	1	0	0	0	0	0	1	0	0	0	1	1	0	2	0
16	13	14.52554	1	2	0	0	2	2	1	1	1	3	1	0	1	1	0	0	0	1
17	14	16.20578	0	3	1	0	2	1	0	0	1	1	2	0	0	2	0	3	0	2
18	15	17.88603	0	7	1	1	0	1	2	0	3	1	0	1	0	1	1	0	1	0
19	16	19.56627	0	2	0	0	2	0	0	0	0	0	0	2	2	0	0	1	1	0
20	17	21.24651	0	0	1	2	2	1	1	0	1	0	0	1	2	1	0	0	0	1
21	18	22.92675	0	0	2	1	2	0	2	1	0	0	1	0	0	0	1	0	0	0
22	19	24.60699	1	1	1	1	1	1	0	1	1	2	0	1	1	1	1	0	2	1
23	20	26.28723	1	1	1	0	1	2	1	0	1	1	1	0	0	1	1	0	1	2
24	21	27.96748	1	1	2	1	0	1	0	1	0	0	1	1	0	0	2	0	2	1
25	22	29.64772	2	0	1	1	2	1	0	0	2	2	1	2	0	1	1	0	2	1
26	23	31.32796	0	1	0	0	1	1	1	0	0	2	0	0	0	0	1	1	0	1
27	24	33.0082	3	0	0	0	0	2	0	0	0	1	4	0	2	1	0	2	1	3
28	25	34.68844	2	0	3	0	2	1	0	0	1	2	2	0	2	1	2	1	1	0
29	26	36.36869	0	1	2	2	0	0	0	0	0	0	1	1	3	1	4	1	0	2
30	27	38.04893	4	1	2	1	1	0	2	4	1	0	1	0	2	1	1	1	3	2
31	28	39.72917	1	0	1	2	0	0	1	0	1	1	0	1	1	2	1	1	0	1
32	29	41.40941	2	0	4	0	0	2	0	5	1	0	0	0	1	2	1	0	0	1
33	30	43.08965	0	0	0	3	4	1	1	1	2	2	2	2	0	1	0	1	3	1
34	31	44.76989	1	2	2	3	0	0	1	4	3	1	0	2	3	1	1	2	3	2
35	32	46.45014	2	2	3	0	1	3	1	2	2	1	0	2	0	2	0	2	5	0
36	33	48.13038	1	4	3	3	1	2	5	2	0	2	4	2	0	6	1	1	1	1
37	34	49.81062	2	2	4	1	3	2	0	2	1	2	2	1	3	1	2	2	6	2
38	35	51.49086	1	3	4	1	0	1	3	1	2	1	5	4	4	3	3	1	0	9
39	36	53.1711	1	1	0	2	1	2	2	1	3	3	3	2	2	1	2	4	2	2
40	37	54.85134	0	4	2	1	1	2	4	2	1	3	3	3	2	1	2	2	5	1
41	38	56.53159	0	3	2	2	4	3	3	5	5	3	2	3	2	3	2	2	3	3
42	39	58.21183	2	3	4	3	5	1	2	3	6	3	3	7	1	0	2	1	0	0
43	40	59.89207	1	5	4	4	0	3	4	0	2	6	6	3	1	0	3	3	4	2
44	41	61.57231	3	4	3	4	3	1	4	4	4	2	6	2	0	2	4	2	1	3
45	42	63.25255	1	0	5	4	3	1	6	4	3	2	2	3	3	3	3	4	3	0

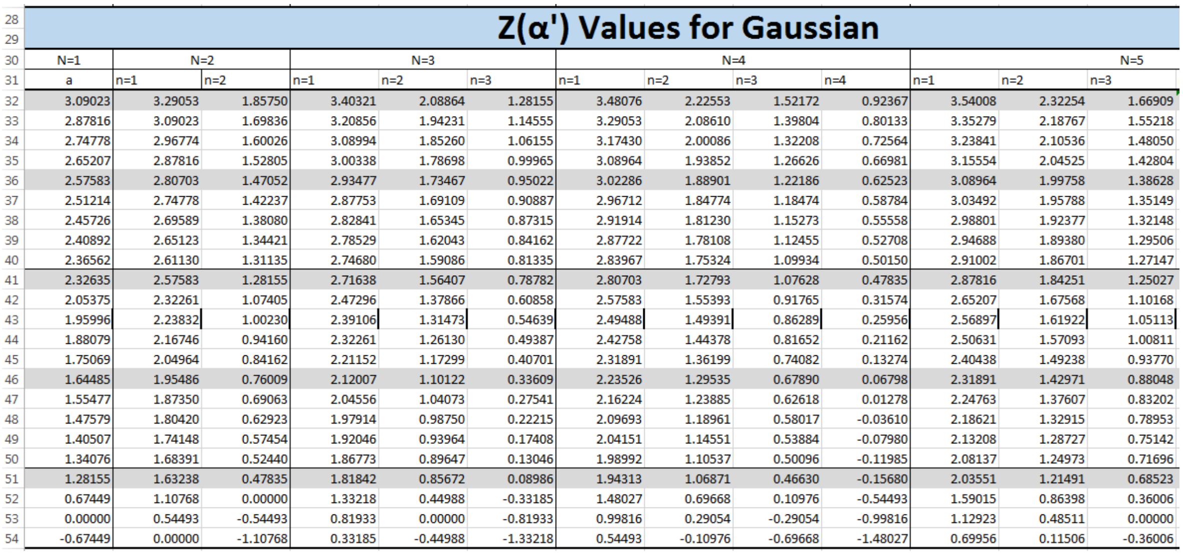
Figure 7: Example of Spectra Sheet

Poisson y Calculator Sheet:*

The calculation of *y** requires the use of a known statistical distribution. For sufficiently large counts, a Gaussian distribution adequately models detector response to radiation decay. The calculation of *y** using a Gaussian distribution is an easy task for Excel. Lower counts, however, require the use of a Poisson distribution. The calculation of *y** using a Poisson distribution is not an operation incumbent to Excel, thus it was necessary to fabricate a Poisson *y** calculator. This calculator sheet is invoked for every *y** calculation for a bin whose average counts are less than or equal to 30. Equation 13 is invoked for the calculation of *y**. Appendix F provides a more detailed summary of how the calculator functions.

Alpha prime and Z value sheets:

The algorithm requires the α_{ind} value for every *y** for any given set of (*N,n,*α). This sheet contains the table that defines the α_{ind} values given by Figure 3 in the introduction. Directly below the table is the *Z(α')* table. The table in Figure 8 uses a –NORM.S.INV(*α'*) Excel function for each cell to calculate the one tailed *Z* value for its corresponding α_{ind} cell. These calculations are used only when the Gaussian *y** calculation is invoked.



Z(α') Values for Gaussian

N=1	N=2		N=3			N=4				N=5		
a	n=1	n=2	n=1	n=2	n=3	n=1	n=2	n=3	n=4	n=1	n=2	n=3
3.09023	3.29053	1.85750	3.40321	2.08864	1.28155	3.48076	2.22553	1.52172	0.92367	3.54008	2.32254	1.66909
2.87816	3.09023	1.69836	3.20856	1.94231	1.14555	3.29053	2.08610	1.39804	0.80133	3.35279	2.18767	1.55218
2.74778	2.96774	1.60026	3.08994	1.85260	1.06155	3.17430	2.00086	1.32208	0.72564	3.23841	2.10536	1.48050
2.65207	2.87816	1.52805	3.00338	1.78698	0.99965	3.08964	1.93852	1.26626	0.66981	3.15554	2.04525	1.42804
2.57583	2.80703	1.47052	2.93477	1.73467	0.95022	3.02286	1.88901	1.22186	0.62523	3.08964	1.99758	1.38628
2.51214	2.74778	1.42237	2.87753	1.69109	0.90887	2.96712	1.84774	1.18474	0.58784	3.03492	1.95788	1.35149
2.45726	2.69589	1.38080	2.82841	1.65345	0.87315	2.91914	1.81230	1.15273	0.55558	2.98801	1.92377	1.32148
2.40892	2.65123	1.34421	2.78529	1.62043	0.84162	2.87722	1.78108	1.12455	0.52708	2.94688	1.89380	1.29506
2.36562	2.61130	1.31135	2.74680	1.59086	0.81335	2.83967	1.75324	1.09934	0.50150	2.91002	1.86701	1.27147
2.32635	2.57583	1.28155	2.71638	1.56407	0.78782	2.80703	1.72793	1.07628	0.47835	2.87816	1.84251	1.25027
2.05375	2.32261	1.07405	2.47296	1.37866	0.60858	2.57583	1.55393	0.91765	0.31574	2.65207	1.67568	1.10168
1.95996	2.23832	1.00230	2.39106	1.31473	0.54639	2.49488	1.49391	0.86289	0.25956	2.56897	1.61922	1.05113
1.88079	2.16746	0.94160	2.32261	1.26130	0.49387	2.42758	1.44378	0.81652	0.21162	2.50631	1.57093	1.00811
1.75069	2.04964	0.84162	2.21152	1.17299	0.40701	2.31891	1.36199	0.74082	0.13274	2.40438	1.49238	0.93770
1.64485	1.95486	0.76009	2.12007	1.10122	0.33609	2.23526	1.29535	0.67890	0.06798	2.31891	1.42971	0.88048
1.55477	1.87350	0.69063	2.04556	1.04073	0.27541	2.16224	1.23885	0.62618	0.01278	2.24763	1.37607	0.83202
1.47579	1.80420	0.62923	1.97914	0.98750	0.22215	2.09693	1.18961	0.58017	-0.03610	2.18621	1.32915	0.78953
1.40507	1.74148	0.57454	1.92046	0.93964	0.17408	2.04151	1.14551	0.53884	-0.07980	2.13208	1.28727	0.75142
1.34076	1.68391	0.52440	1.86773	0.89647	0.13046	1.98992	1.10537	0.50096	-0.11985	2.08137	1.24973	0.71696
1.28155	1.63238	0.47835	1.81842	0.85672	0.08986	1.94313	1.06871	0.46630	-0.15680	2.03551	1.21491	0.68523
0.67449	1.10768	0.00000	1.33218	0.44988	-0.33185	1.48027	0.69668	0.10976	-0.54493	1.59015	0.86398	0.36006
0.00000	0.54493	-0.54493	0.81933	0.00000	-0.81933	0.99816	0.29054	-0.29054	-0.99816	1.12923	0.48511	0.00000
-0.67449	0.00000	-1.10768	0.33185	-0.44988	-1.33218	0.54493	-0.10976	-0.69668	-1.48027	0.69956	0.11506	-0.36006

Figure 8: Alpha Prime and Z Value Sheet

There needs to be a method of benchmarking the α_{ind} values, since these calculated values are at the root of all calculations. The calculated false positives for each point on an ROC curve should approximate the input alpha values (i.e., the left column of each *n* block in the Calculation Table should approximately match the left most column of the table). It is expected that the lower α_{ind} values will yield false positive values that more poorly match their corresponding *α* value due to the small number of false positives at these low values. The most important benchmarking, however, is independent verification from another researcher. The *y** values calculated from the points 0.001, 0.005, 0.01, 0.025, 0.05, 0.01, 0.05, and 0.01 have been shown to agree with the *y** values calculated by other researchers for the same set of data.

Program execution

The use of this tool proceeds in four basic steps (step three is optional). The principle behind each step is explained below. Appendix A contains a detailed procedure for the completion of each step; each step is aptly named Procedure 1, 2, 3, and 4, respectively.

Step 1: Import Background and Source Spectra.

This program flexibly allows for the importation of any number of background and source spectra from any source. There must be at a minimum one sheet with a set of background spectra and one sheet with a set of source spectra. The spectra must be properly formatted on an Excel sheet for the code to function. Procedure 1 in Appendix A explains how this is to be accomplished. The method by which the user uploads the properly formatted spectra onto Excel is a matter of taste; for this research, however, an Excel macro was utilized. See Appendix B for a detailed description of how this was accomplished. As previously discussed, each set of spectra is imported to a new sheet in the "ROC_curve_generator" Excel file.

Step 2: Bin Energies.

Since the purpose of this research is to optimize the detection of a weak source using information from the entire spectrum, it becomes necessary for the user to bin (group) energies. This code is designed to maximize the flexibility of this grouping. On the main spreadsheet entitled "Energy Bins 2," the user enters the maximum energy for each bin in the "Max Energy" column; a description column is also provided to the left of each max energy cell. Up to 19 bins are allowed. In effect, the spectra are partitioned in accordance with the desire of the user. The user also has the option to determine how much of the total available spectra to use for both the background and source. An additional feature is provided that allows the user to combine up to eight different bins into one "combined bin" (defaulted as bin 20), thereby furnishing the user with nearly unlimited possibilities for grouping energies for analysis. Additionally, the gross counts are automatically computed and defaulted as bin 21. Several statistics are automatically computed in the statistics blocks. The common statistics computed are the total counts, the averages, and the standard deviations across each bin. The source to background ratios (S/B) are also computed in the background statistics block for each bin, including the gross count S/B Ratio. Directly below in the source statistics block are calculated the ratios of the source to background ratio on each bin to the gross count source to background ratio (S/B : S/B Ratio). These ratios, however, were not utilized through the course of this research.

Step 3: Obtain Primary Mass Function (PMF) - Optional

This optional step is performed after step 2; (steps 3 and 4 may be done in any order). This step organizes the data in such a format that the PMF for each set of spectra can be graphed. These data are organized on the "Energy Bins 2" sheet on rows 103-104 and rows 107-108 for the background PMF and the source PMF, respectively.

The motivation for this feature is to provide assurance that the count distributions do indeed follow a Poisson distribution in general, and at greater counts follow a Gaussian distribution, as postulated. This assurance is essential, as all subsequent *y** calculations, and hence, all the data used in this research, are utterly contingent on the proper insight concerning the count distributions. Figures 9 and 10 prove salutary in reinforcing these assumptions. Figure 9 graphs the PMF for both the actual background and source counts for a relatively narrow energy bin. Included within the graph are the actual Poisson distributions calculated from the same number of spectra and the same mean for the background and source counts each.

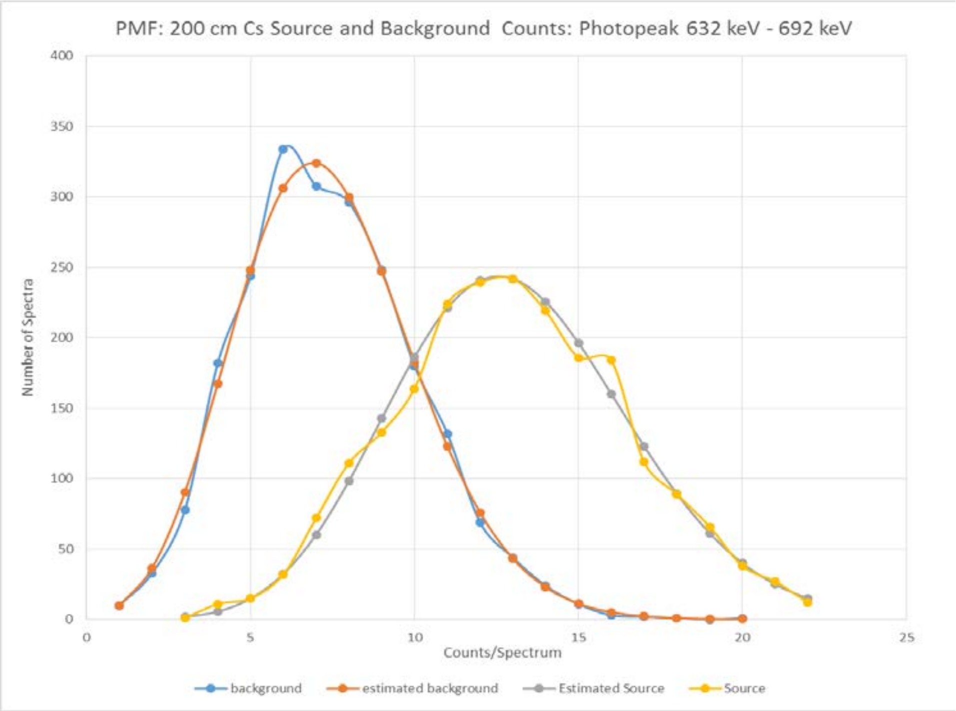


Figure 9: PMF for 200 cm ^{137}Cs and Background – Photopeak Region 632 keV – 692 keV

Figure 10 provides the same type of graph for gross counts. As expected, there is much greater statistical fluctuation from the expected Gaussian distribution. Naturally, this is due to the larger range of the PMF.

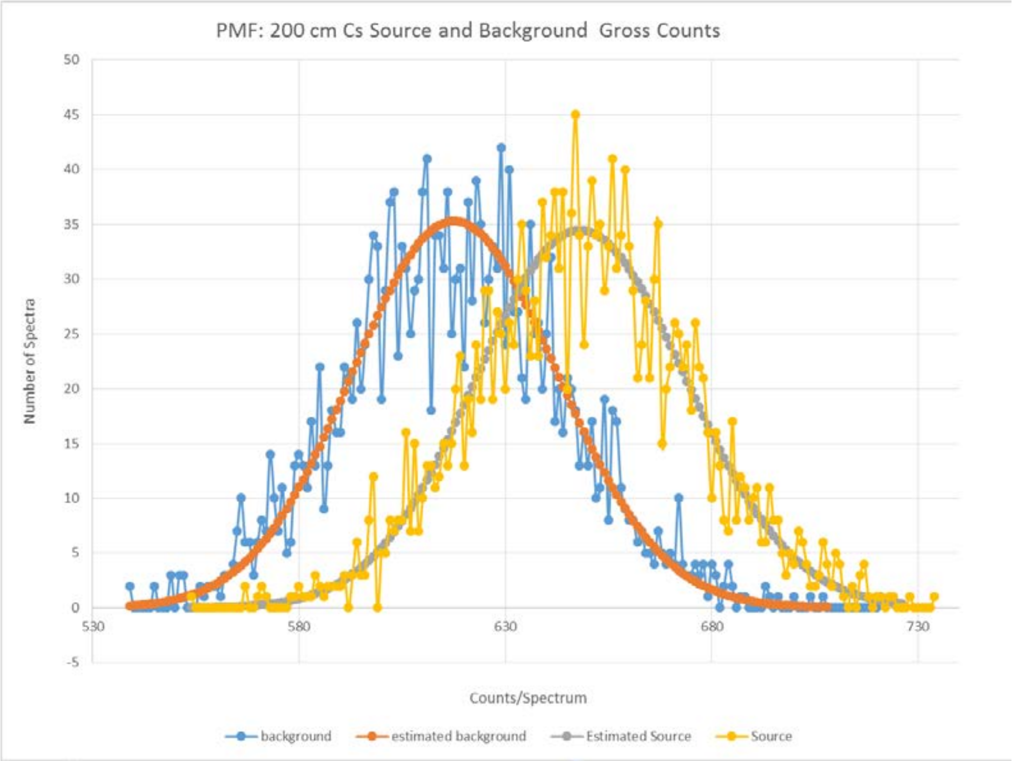


Figure 10: PMF for200 cm ^{137}Cs Source and Background – Gross Counts

From Figures 9 and 10 it is readily apparent that the Poisson and Gaussian distributions provide a reliable model from which to base the *y** calculations. This pattern was found to be consistent for all PMFs; for the sake of brevity, only these two figures are used as a representation.

Step 4: Compute ROC Curves for a given energy bin and data string length.

This step must be performed after step 2. Once the energies are binned, the user is able to choose the data string metric with user entry into cells M24 and M25. An *N* + 1 number of ROC curves are automatically computed: one curve for each *n* value and one curve for the traditional method. The data points for each curve are provided in the ROC curve points table on the "Energy Bins 2" sheet. As stated in the introduction, the data string metrics programed thus far are for $N \in [1,10]$ and $n \in [1, N]$. Directly beneath the ROC curve points table is the uncertainty table.

Part III: Method of Analysis

The greatest difficulty with this research lies within the multidimensional nature of the analysis. The objective is to optimize the ROC curves amidst multiple variables. There are at least five variables involved with this analysis:

1. Data string metric (*n*/*N*)
2. Energy bin
3. Source Strength (simulated by distance)
4. Interference (elevated background and shielding).
5. Source Type

The feasibility of any multivariable analysis relies on the systematic constraint of multiple variables while only one variable is allowed to vary. In the following analysis, the string length will be set to *N*=5 while all the ROC curves with values of *n* such that $n \in [1,5]$ will be examined. The ROC curve representing the traditional method will also be examined along with the *n*/5 data string metrics. Thus, six ROC curves will be graphed simultaneously.

Recall that hypothesis 1 of this research it to discover if focusing on an energy subset *E* within the spectrum can optimize each of these curves beyond those found from the gross counts. Additionally, whether or not there is any variance between the data string metrics of *N*=5 for different energy bins will be examined. Major emphasis will also be placed on verifying hypothesis 2, which entails finding a systematic criterion by which the optimal grouping of energies can be known, thereby eliminating blind trial and error. Hence, it is necessary to discover a suitable way of measuring the effect size of each bin compared to the effect size of the bin representing the gross counts. The effect size is defined to be the degree to which the null hypothesis is false [10]. An effect size is a statistical measure of performance; for instance, the source to background ratio or the number of standard deviations by which the means differ are examples of effect size measurements.

The effect that the source strength will have on the ROC curves is obvious, and its inclusion as a variable may hardly seem necessary. The sole reason for its inclusion in the analysis is to check for consistency of results across various source strengths. It cannot be assumed that what is optimal for a strong source is optimal for a weak source.

The study of interference effects, such as elevated background and shielding, is beyond the scope of this paper. It is anticipated that the results from this analysis will impart understanding of the effects of elevated background and shielding. This will show a suitable way to optimize the detection of a source with such encumbrances. The study of different sources such as ^{60}Co or natural uranium is likewise beyond the scope of this paper.

RESULTS AND DISCUSSION

In satisfaction of hypothesis 1, one of the goals of this research is to optimize the true positive detection rate for a given false positive detection rate. The ordered pair of the false positive detection rate and the true positive detection rate represents a point of the ROC curve for a particular data sting metric. Thus, the ROC curves are examined to determine detection optimization. The full spectrum is utilized in this optimization by partitioning the spectrum into energy groups (bins) and determining which energies ought to be counted and which energies ought to be ignored. It is hypothesized that further improvements can be made when only counting certain groups of energies than when considering only gross counts (i.e., the entire spectrum).

Several source strengths of ^{137}Cs are considered for all *N*=5 data string metrics. Source strengths are simulated by varying the distance between the ^{137}Cs source and the detector. The distances utilized are 100 cm, 200 cm, and 400 cm. From the $\frac{1}{r^2}$ law, this simulates relative source strengths of 1, 1/4, and 1/16, respectively. To eliminate blind trial and error, it is essential to employ a criterion by which to measure the relative performance of a bin of energies against gross counts. Two different criteria are tested; they are the $\frac{\Delta\mu}{\sigma}$ and the source to background ratio, $\left(\frac{S}{B}\right)$. Both of these measures of effect size are commonly used in counting statistics. The $\frac{\Delta\mu}{\sigma}$ is the measure of the number of standard deviations of the background spectra that the average source counts exceed the background source counts. The $\frac{S}{B}$ ratio is the ratio of the average source counts to the average background counts. The $\frac{\Delta\mu}{\sigma}$ is more generally known as the Cohen's d measure of effect size [10] and is expressed as

$$d = \frac{m_A - m_B}{\sigma}. \tag{24}$$

Equation 24 is for the unidirectional, one-tailed case only. In the Cohen’s d equation, the standard deviation can be the standard deviation of either of the means, or it can be the pooled standard deviation. In the application of radiation detection, the standard deviation used is the background standard deviation. Both of these measures of effect size have the desired characteristic of being a pure number. In both cases, the raw effects are standardized to create a unitless and dimensionless quantity.

The effectiveness of these measures as criteria of optimization are tested against the actual performance of the ROC curves for those energy groupings compared to the ROC curves from the gross counts. These two measures are first tested at the 100 cm distance. Initially, the entire set of spectra is partitioned into equal bins of 25 keV width, as this is commensurate with the resolution of the NaI detector. As stated in the introduction, performance is evaluated by the use of ROC curves.

100 cm ^{137}Cs Source

Figures 11 and 12 compare the $\frac{\Delta\mu}{\sigma}$ and the $\frac{S}{B}$ ratio at 100 cm, respectively. The $\frac{\Delta\mu}{\sigma}$ graph suggests that bins of energies within the vicinity of the photopeak outperform the gross counts; specifically, that only the 625 keV - 650 keV and 650 keV - 675 keV bins outperform the gross counts. The $\frac{S}{B}$ ratio graph, however, suggests that the photopeak, ranging from 600 keV to 700 keV in this case, as well as the energies around the Compton edge, outperform the gross counts.

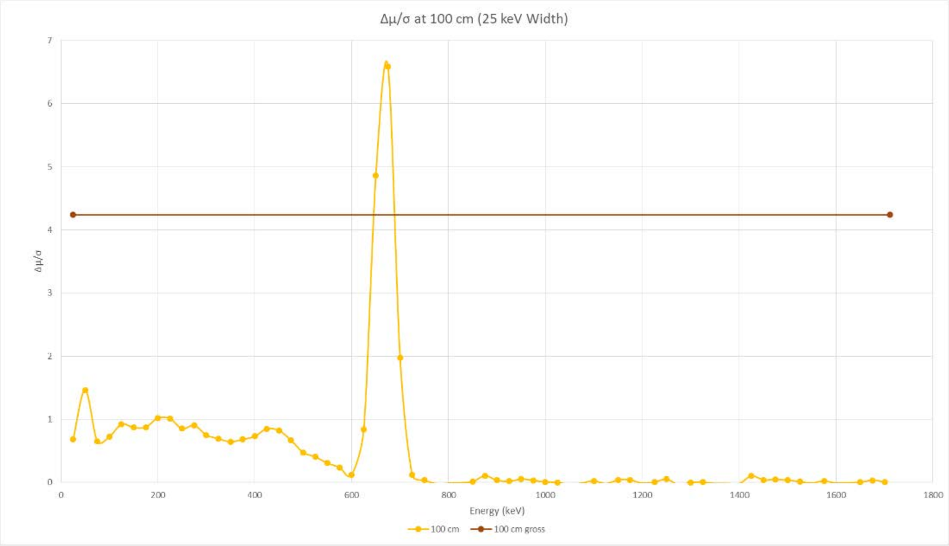


Figure 11: $\frac{\Delta\mu}{\sigma}$ at 100 cm (25 keV Width)

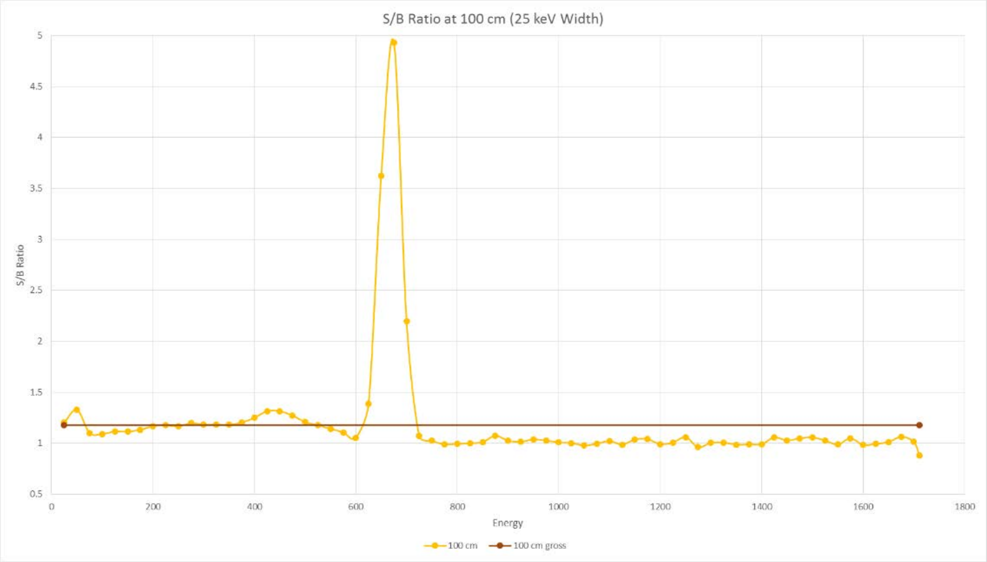


Figure 12: $\frac{S}{B}$ Ratio at 100 cm (25 keV Width)

To observe which of these comparisons most accurately predicts optimal performance for binned data, it is necessary to see the ROC curves for these particular bins and compare them to the ROC curves from the gross counts. Figure 13 shows the ROC curves for the gross counts at 100 cm.

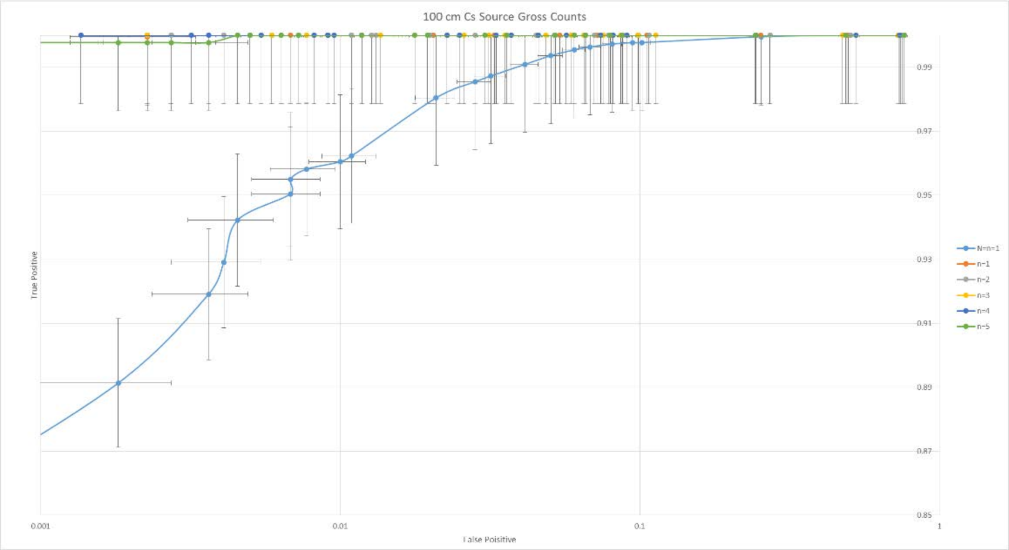


Figure 13: 100 cm ^{137}Cs Source Gross Counts

The *N*=5 data string parameters clearly outperform the traditional method (denoted as N=n=1 in the legend), as expected. The $\frac{\Delta\mu}{\sigma}$ criterion in Figure 11 predicts that the energy bins ranging from 625 keV – 675 keV outperform the gross counts while the $\frac{S}{B}$ ratio in Figure 12 predicts that the energy bins ranging from 600 keV – 700 keV outperform the gross counts. Figures 14 and 15 display the ROC curves for the 625 keV – 675 keV bins and the 600 keV – 700 keV respectively.

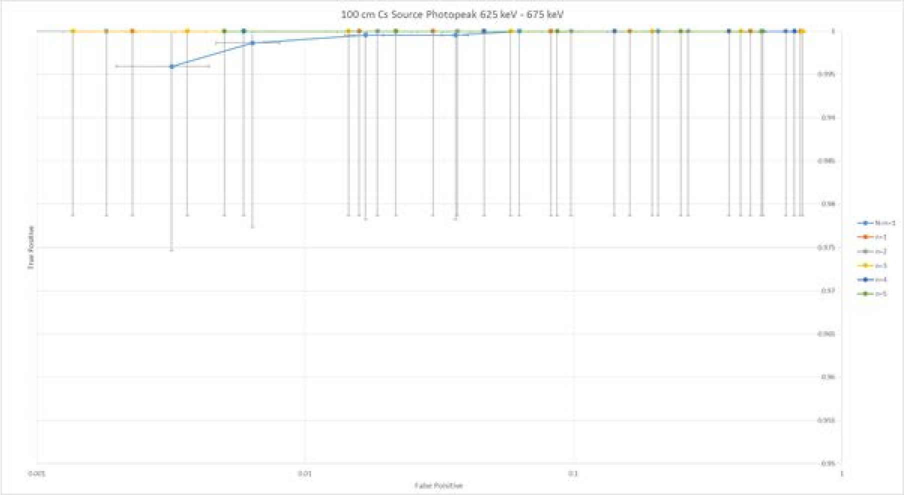


Figure 14: 100 cm ^{137}Cs Source Photopeak Region 625 keV – 675 keV

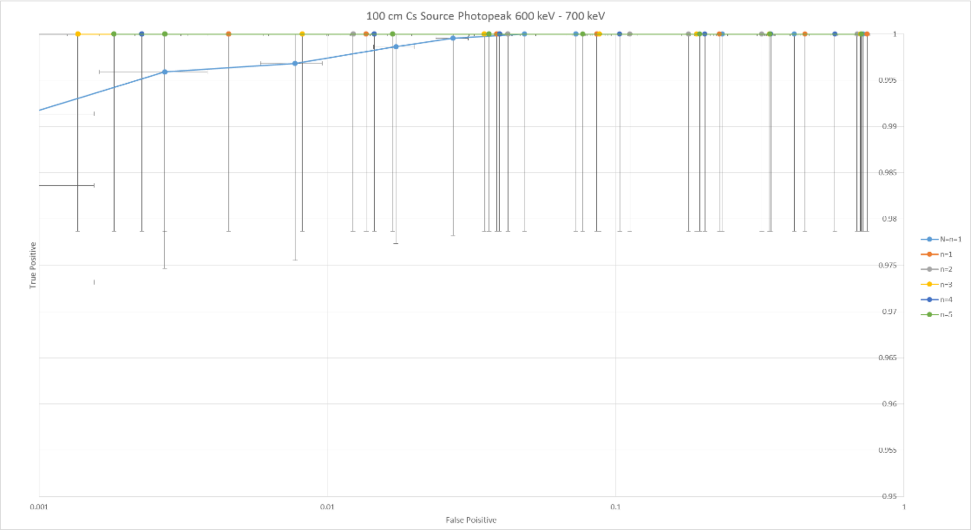


Figure 15: 100 cm ^{137}Cs Source Photopeak Region 600 keV – 700 keV

Both of these sets of energy bins clearly outperform the gross counts. It is difficult to discern from these graphs alone which energy groupings are superior. Figure 16 isolates the bins with energies 600 keV – 625 keV and 675 keV – 700 keV and aggregates them into one combined bin.

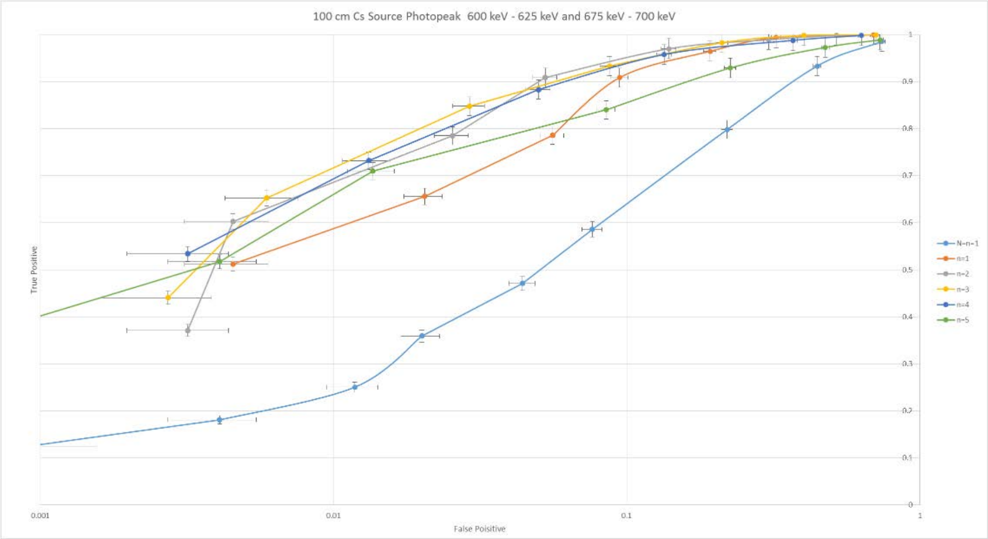


Figure 16: 100 cm ^{137}Cs Source: 600 keV – 625 keV and 675 keV – 700 keV. The $\frac{S}{B}$ ratio predicts that these curves will outperform the curves from the gross counts. The $\frac{\Delta\mu}{\sigma}$, however, predicts that these curves will underperform compared to the curves from the gross counts.

The ROC curves in Figure 16 indicate that these energy bins underperform compared to the gross counts. The $\frac{S}{B}$ ratio criterion given by Figure 12 also predicts that several energy bins just below the 478 keV Compton edge outperform the gross counts. The ROC curves for these bins are graphed in Figure 17.

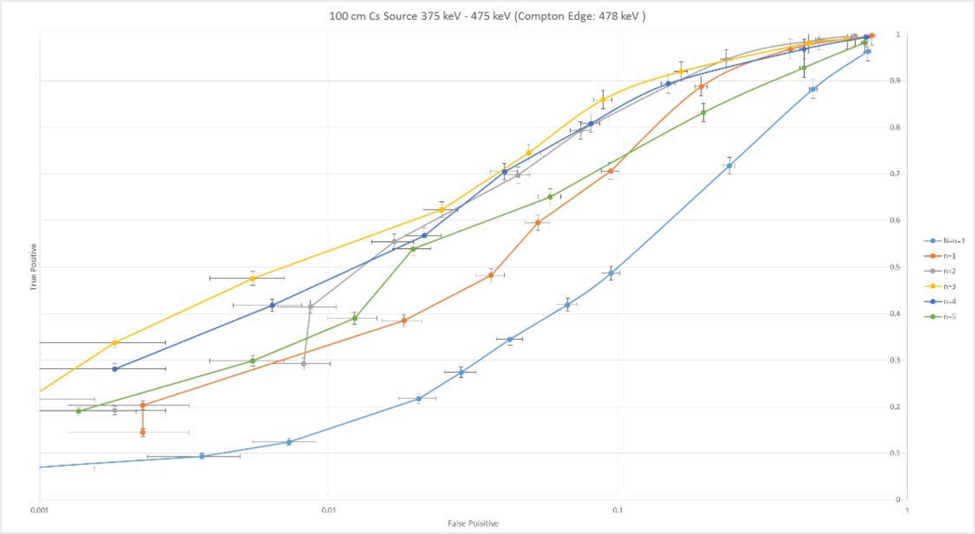


Figure 17: 100 cm ^{137}Cs Source Compton Edge Region: 375 keV – 475 keV

Observe that the ROC curves in Figure 17 are suboptimal compared to the gross counts in Figure 13. For further verification, all the energies that are below the $\frac{\Delta\mu}{\sigma}$ for the gross counts are combined into one bin, and the ROC curves for these energies are graphed in Figure 18.

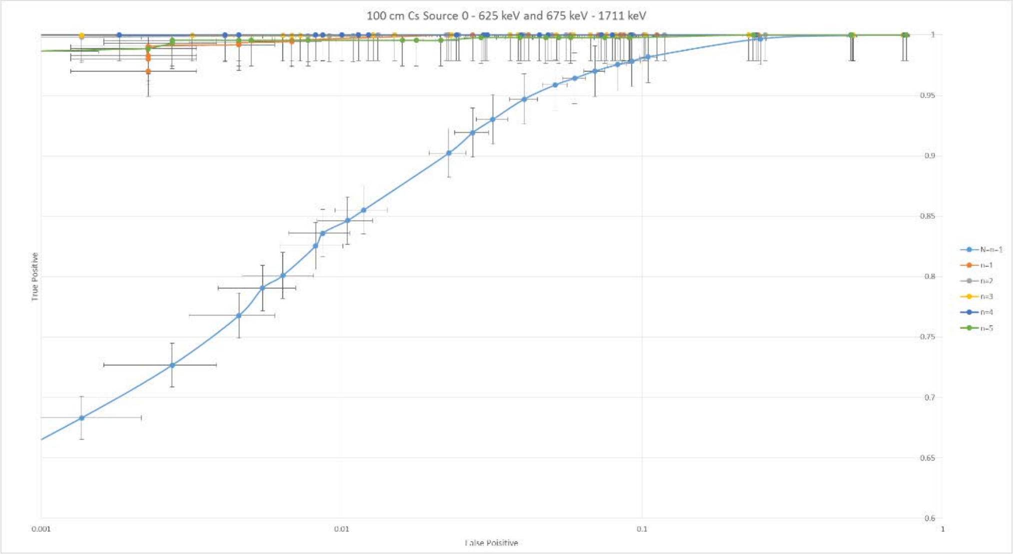


Figure 18: 100 cm ^{137}Cs Source: 0 keV – 625 keV and 675 keV – 1711 keV

From the above figures, including Figures 16, 17, and 18, it is clear that the $\frac{S}{B}$ ratio criterion shown by Figure 12 fails at predicting ROC curve behavior for groups of energies compared to the ROC curve for gross counts. Henceforth, the $\frac{\Delta\mu}{\sigma}$ criterion is used to determine which groups of energies to focus on in subsequent analysis, since this method has thus far succeeded. Specifically, if $\left(\frac{\Delta\mu}{\sigma}\right)_{E_i} > \left(\frac{\Delta\mu}{\sigma}\right)_{gross}$, then the energy grouping E_i likely outperforms the gross counts. Because these results are so obvious at the 100 cm distance, the $\frac{S}{B}$ ratio criterion is not tested for the weaker source. Instead, further testing of the $\left(\frac{\Delta\mu}{\sigma}\right)_{E_i} > \left(\frac{\Delta\mu}{\sigma}\right)_{gross}$ criterion for superior ROC curve performance is conducted in order to verify that this criterion holds for all source strengths.

The extended criterion using the data string metrics, (in this case the *N*=5 data string metrics), continues to outperform the traditional method. This is seen in both the energy binning that outperforms and underperforms the gross counts. Also, Figures 14 and 15 both clearly indicate that specific energy bins can indeed outperform the gross counts, as hypothesized. Moreover, the specific data string metrics that perform most optimally are those of the middle *n* values: 2/5, 3/5, and 4/5, with the extreme data string metrics of 1/5 and 5/5 underperforming.

With the 100 cm count, the ^{137}Cs source is too strong to draw any conclusion about individual data string metric performances, as both Figures 14 and 15 indicate, since all data string parameters of *N*=5 are nearly perfect. The ^{137}Cs must be weaker; for that an analysis of the ^{137}Cs source at 200 cm is needed.

200 cm ^{137}Cs Source

The $\left(\frac{\Delta\mu}{\sigma}\right)_{E_i} > \left(\frac{\Delta\mu}{\sigma}\right)_{gross}$ criterion for energy bin performance is tested for consistency at this source strength. The $\frac{\Delta\mu}{\sigma}$ graph for the 200 cm source strength at 25 keV bin widths is given in Figure 19.

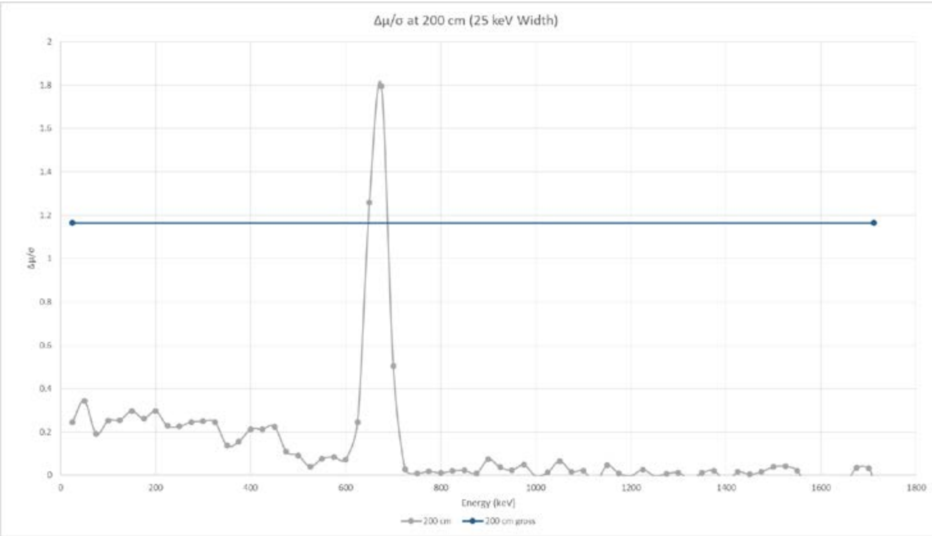


Figure 19: $\frac{\Delta\mu}{\sigma}$ at 200 cm (25 keV Width)

Notice that this resembles the $\frac{\Delta\mu}{\sigma}$ at 100 cm; the only bins where the $\frac{\Delta\mu}{\sigma}$ exceeds the gross counts are the bins containing the 625 keV – 675 keV energies. Therefore, it is expected that those bins outperform the gross counts ROC curve. The ROC curve for the gross counts at 200 cm is given in Figure 20.

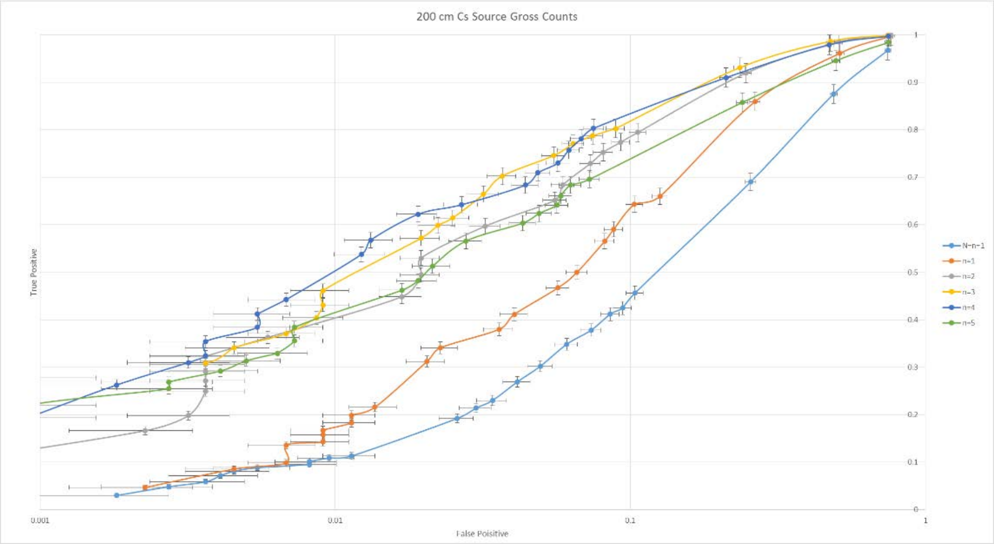


Figure 20: 200 cm ^{137}Cs Source Gross Counts

The ROC curves for the 625 keV – 675 keV photopeak are shown in Figure 21.

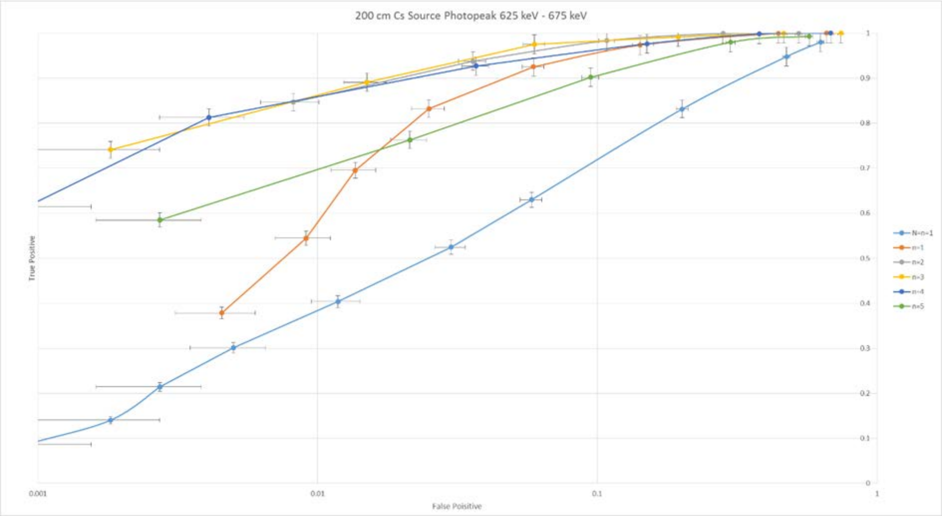


Figure 21: 200 cm ^{137}Cs Source Photopeak Region: 625 keV – 675 keV

The ROC curves representing the photopeak outperform the gross counts, as expected. In Figure 22, the ROC curves representing all the energies not including the range 625 keV – 675 keV are given.

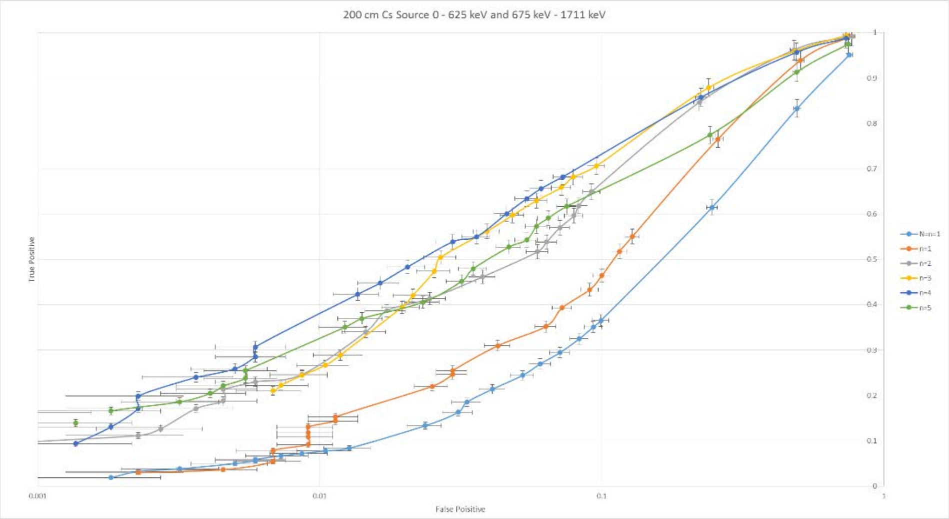


Figure 22: 200 cm ^{137}Cs Source: 0 keV – 625 keV and 675 keV – 1711 keV

As predicted, these ROC curves underperform the ROC curves from the gross counts at 200 cm. The $\left(\frac{\Delta\mu}{\sigma}\right)_{E_i} > \left(\frac{\Delta\mu}{\sigma}\right)_{gross}$ criterion for energy bin performance relative to gross counts holds for the weaker source strength.

Unlike the 100 cm ^{137}Cs source distance, the ^{137}Cs source at 200 cm is significantly less than perfect; the true positive rates are not consistently (or nearly consistently) equal to one for all false positive values. As such, this source strength is optimal for a higher fidelity analysis. It has been demonstrated that the photopeak is the only region of interest that shows any ROC curve improvement from the gross counts. Figure 23 shows the $\frac{\Delta\mu}{\sigma}$ graph for energy bins of a width of 5 keV centered around the 662 keV photopeak.

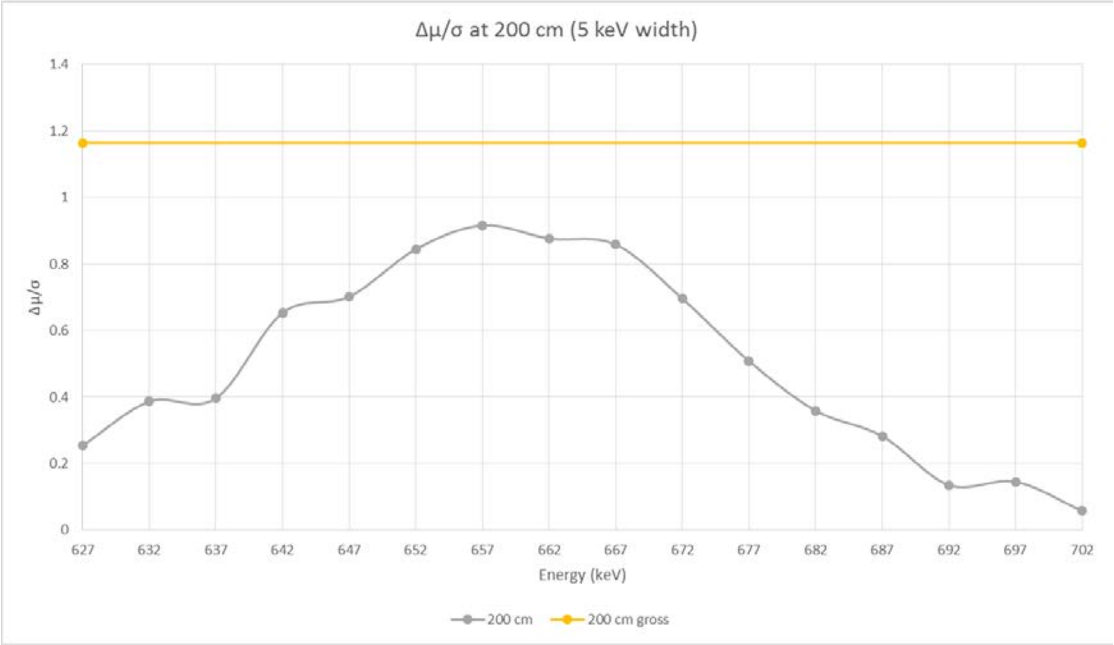


Figure 23: $\frac{\Delta\mu}{\sigma}$ at 200 cm (5 keV Width)

The result shown in Figure 23 indicates that none of these bins in the photopeak region outperform the gross counts. At first glance, this appears to contradict the result given by Figure 19. It is important to recall, however, that the $\frac{\Delta\mu}{\sigma}$ graph given by Figure 19 consists of energy bins of a 25 keV width along the abscissa. On the contrary, Figure 23 contains energy bins of width of only 5 keV along the abscissa. These results seem to indicate that bin performance is dependent on not only the energy, but also the bin width. In Figure 23, the bin with the best $\frac{\Delta\mu}{\sigma}$ performance is the 652 keV – 657 keV bin. Nevertheless, the ROC curves for this bin are predicted to underperform the ROC curves for the gross counts in Figure 20. Figure 24 contains the ROC curves for this energy bin.

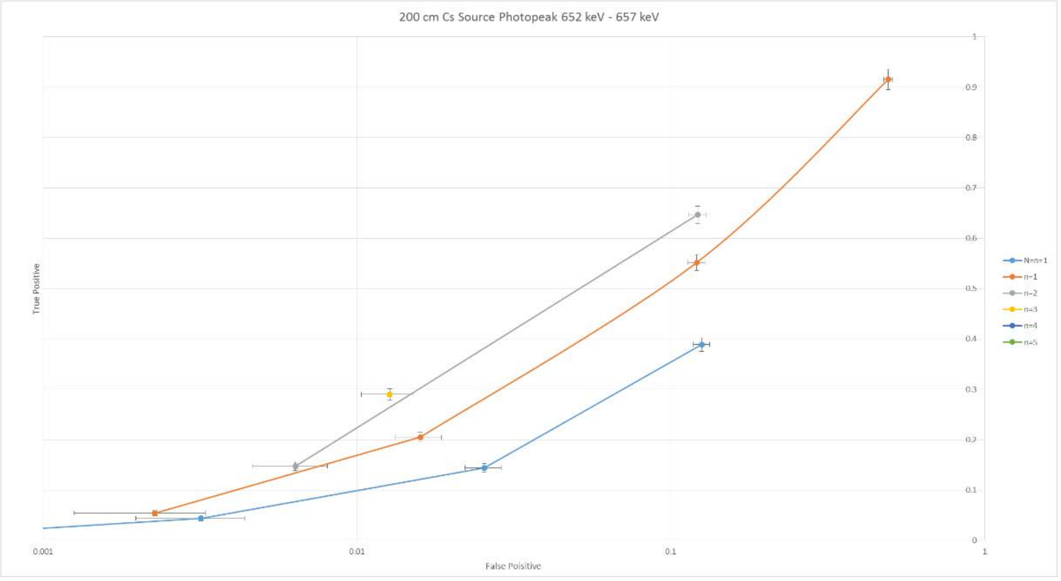


Figure 24: 200 cm ^{137}Cs Source Photopeak Region 652 keV – 657 keV

This result verifies the prediction given by Figure 23, as these ROC curves underperform compared to the ROC curves of the gross counts. Notice that for low counts, the higher data string metrics, (in this case the 5/5, 4/5, and 3/5), fail completely; this result is revisited later.

There now exists a motivation to further verify this result within the photopeak region and to discover the minimum bin size needed to outperform the gross counts. Figure 25 shows the $\frac{\Delta\mu}{\sigma}$ graph around the photopeak for various bin widths increasing in increments of 5 keV from 5 keV to 25 keV.

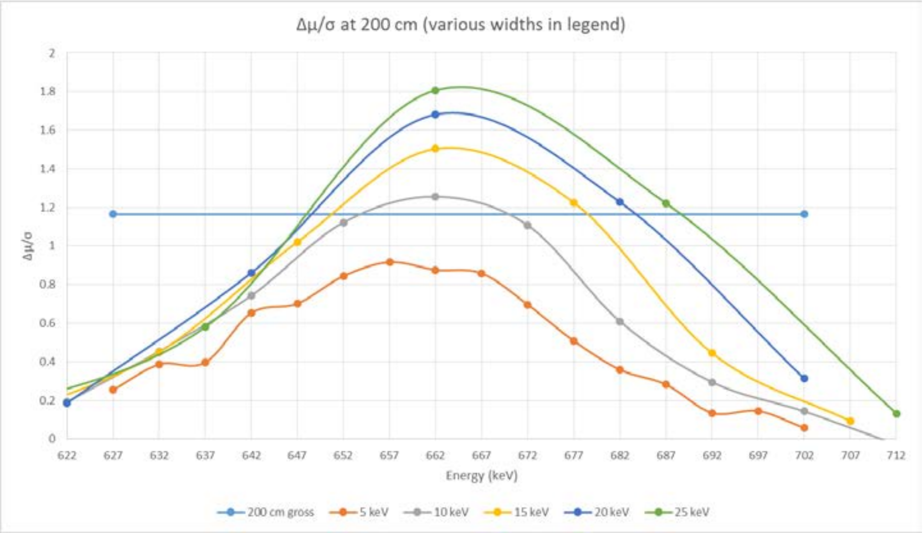


Figure 25: $\frac{\Delta\mu}{\sigma}$ at 200 cm at Various Widths

Figure 25 suggests that the $\frac{\Delta\mu}{\sigma}$ increases by a factor of about $\sqrt{n}$ for a bin width increase of *n* for unchanging source and background count rates, (this is in fact the case, as is demonstrated later). Note that this relationship only applies for the middle of the photopeak. Figure 25 shows that the bin containing energies 652 keV – 662 keV should scarcely outperform the ROC curves for the gross counts. Figure 26 shows the graph of the ROC curves representing the 652 keV – 662 keV bin.

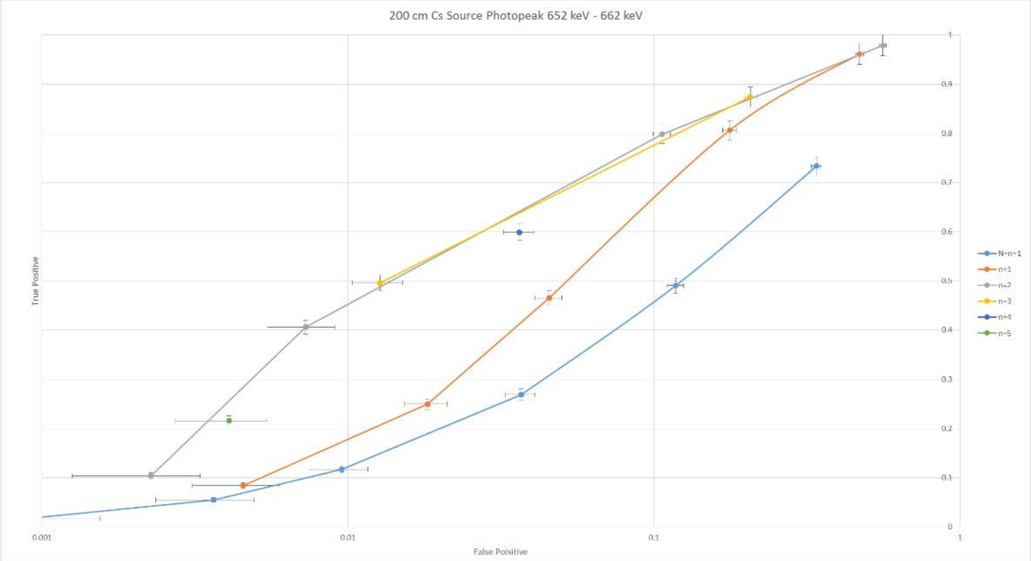


Figure 26: 200 cm ^{137}Cs Source Photopeak 652 keV – 662 keV

The performance of the energy bin in Figure 26 is commensurate with the performance of the gross counts in Figure 20. Figures 27, 28, and 29 show the ROC curves for the 15, 20, and 25 keV width bins centered around the photopeak, respectively.

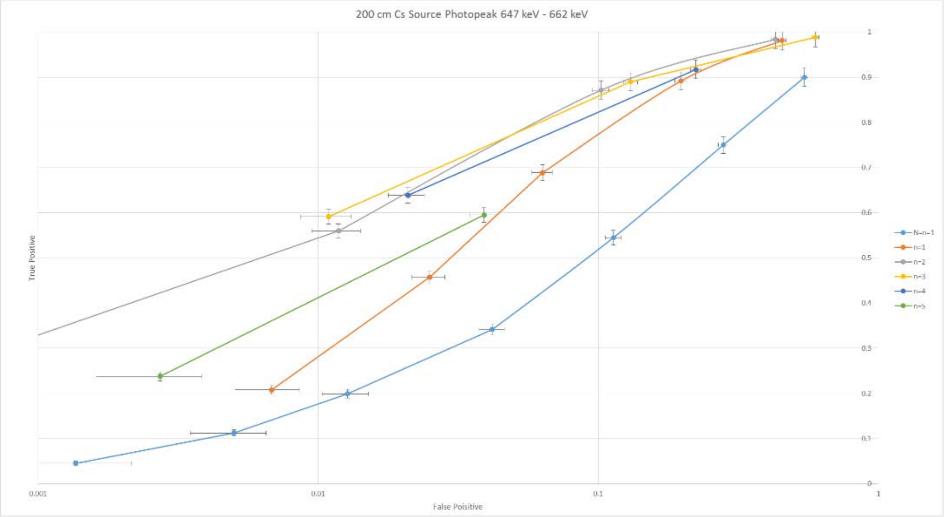


Figure 27: 200 cm ^{137}Cs Source Photopeak Region 647 keV – 662 keV

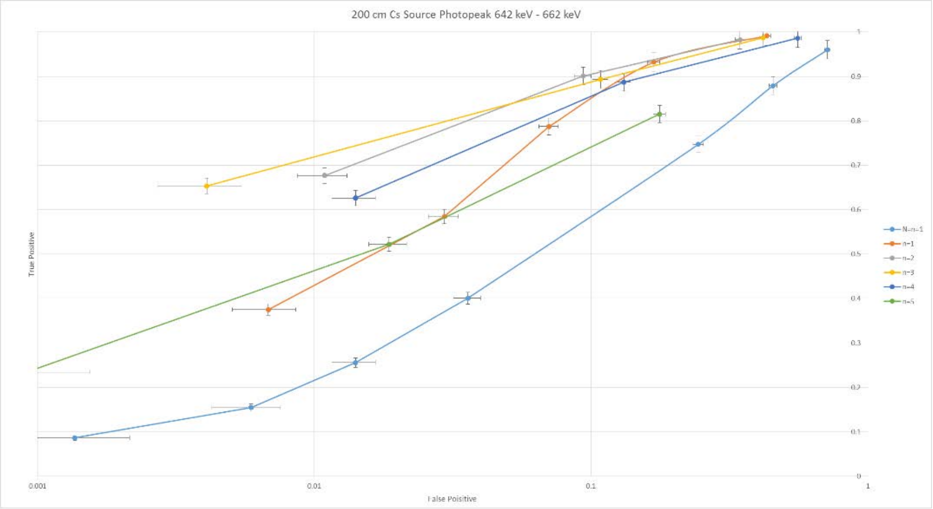


Figure 28: 200 cm ^{137}Cs Source Photopeak Region 642 keV – 662 keV

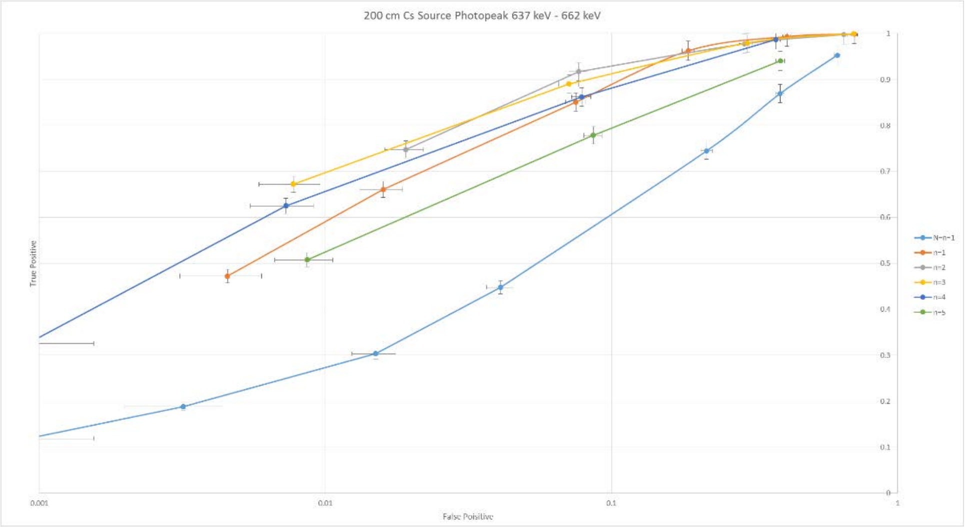


Figure 29: 200 cm ^{137}Cs Source Photopeak Region 637 keV – 662 keV

As expected, the previous three figures show a subsequent improvement in ROC curves as $\frac{\Delta\mu}{\sigma}$ increases. Observe also that the ROC curves in the previous three figures outperform the ROC curves for the gross counts.

It yet remains to find the most optimal energy bin for the 200 cm ^{137}Cs source. The results thus far indicate that the most optimal bin is centered on the 662 keV photopeak. It remains only to determine the precise width of this energy bin. Figure 30 displays the $\frac{\Delta\mu}{\sigma}$ for bins centered on 662 keV for various widths.

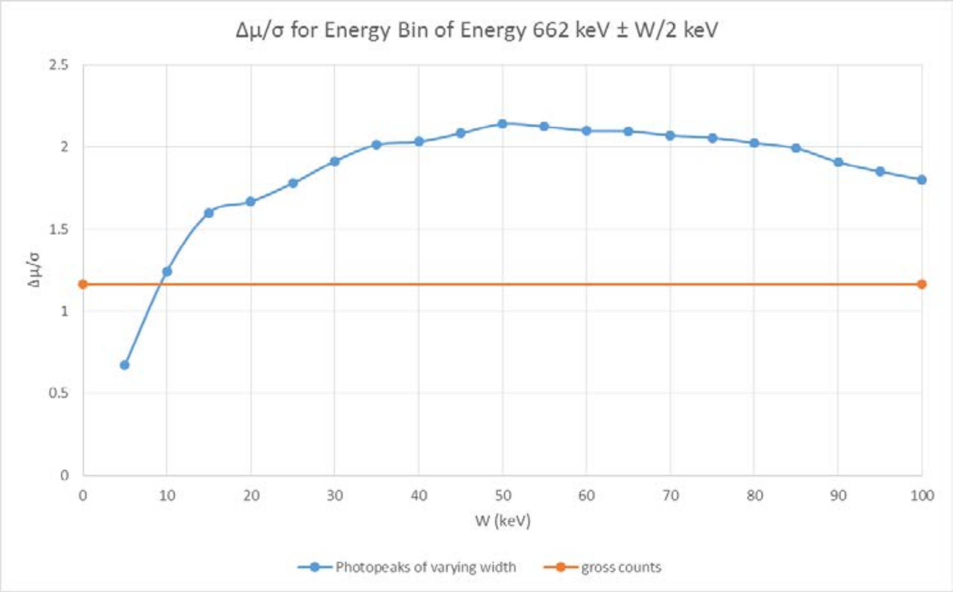


Figure 30: $\frac{\Delta\mu}{\sigma}$ for Energy Bins Centered on the Photopeak Region with various widths at 200 cm Source Strength

Figure 30 predicts that the bin consisting of energies 662 keV ± 25 keV will produce the best ROC curves.

These ROC curves are graphed in Figure 31.

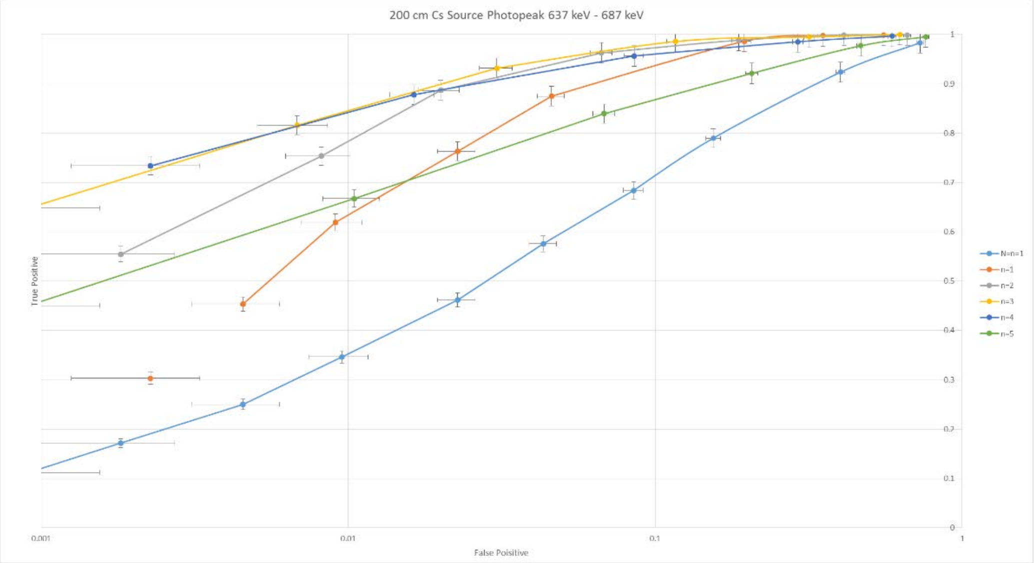


Figure 31: 200 cm ^{137}Cs Source Photopeak Region 637 keV – 687 keV

This set of ROC curves is a strong improvement from the set of ROC curves from the gross counts in Figure 20. Figure 31 is comparable to Figure 21; in fact, they are both in the photopeak region and have the same width of 50 keV. Figure 21 actually performs slightly better. In both Figure 21 and Figure 31, the best detection rates are from both the *n*=3 and *n*=4 curves with a true positive detection rate of about 0.85 at a false positive detection rate of 0.01. The best performing curve in Figure 20 is the *n*=4 curve with a true positive detection rate of 0.50 at a false positive detection rate of 0.01. Every point in Figure 30 represents an energy bin of width *W* centered on the photopeak, each with its own set of ROC curves. These additional curves are given in Appendix G.

The observations of how the particular data string metrics perform for *N*=5 are mostly consistent with what is seen in the gross counts. All of the data string metrics continue to outperform the traditional method. Additionally, the best performing data string metrics are typically both *n*=3 and *n*=4; *n*=2 is usually second or third best, and *n*=1 and *n*=5 perform least optimally, with *n*=1 typically being the worst. Moreover, these sets of ROC curves confirm the hypothesis that the performance of the data string metrics are independent of how the energies are binned as long as the bin is of sufficient width, (i.e., count rate must not be too low). When count rates are too low, the data becomes too discrete and the ROC curves begin to collapse in to single points, (Figure 24 is a prime example of this phenomenon). This occurs for the *n*=5 data string metric first and the *n*=1 data string metric last, with the other data string metrics in sequential order. The curve representing the traditional method typically collapses between the *n*=2 and *n*=3 data string metrics. Thus, for very low counts, the *n*=1 metric performs best.

400 cm ^{137}Cs Source

The purpose of the 400 cm source is to test the effectiveness of the detection method on a very weak source. The $\frac{\Delta\mu}{\sigma}$ graph for the 400 cm source strength at 25 keV bin widths is given in Figure 32.

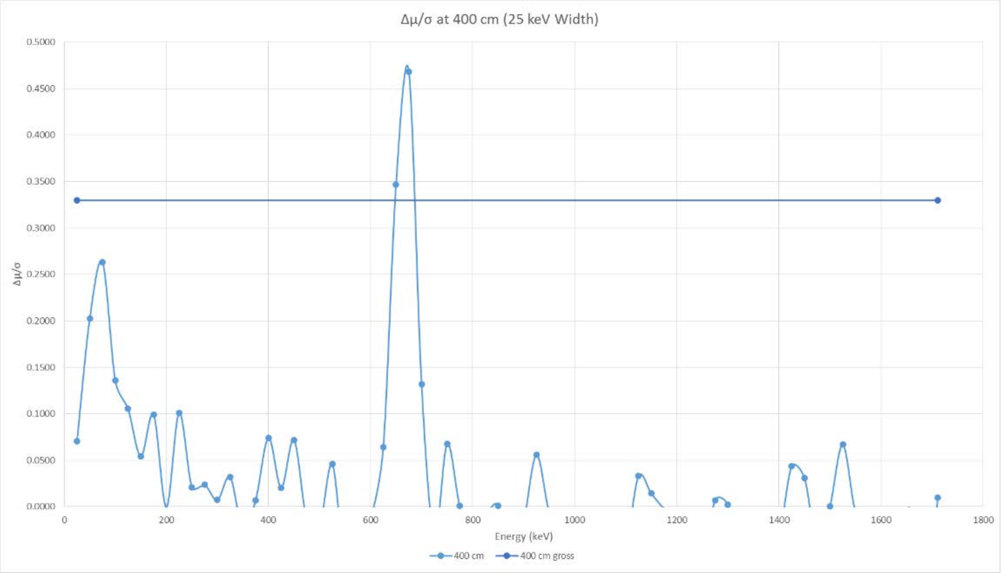


Figure 32: $\frac{\Delta\mu}{\sigma}$ at 400 cm (25 keV Width)

Observe that the same two bins containing photopeak energies 625 keV – 675 keV are predicted to have ROC curves that outperform the gross counts ROC curves. The reader will undoubtedly observe that the subsequent ROC curves perform considerably worse than the previous ROC curves. As a graphical aid, Appendix H provides scaled linear versions of each set of curves. The ROC curve for the gross counts is given in Figure 33.

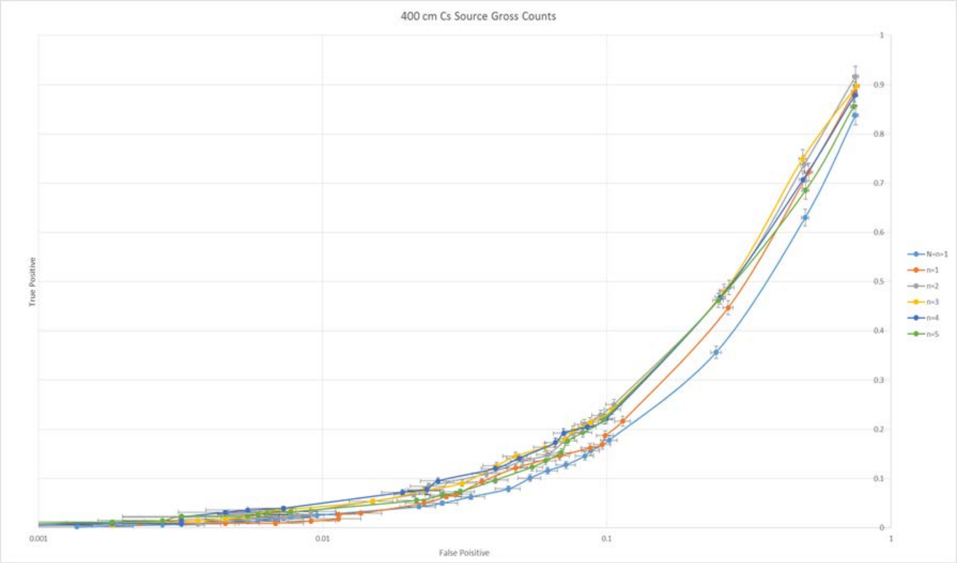


Figure 33: 400 cm ^{137}Cs Source Gross Counts

The $\left(\frac{\Delta\mu}{\sigma}\right)_{E_i} > \left(\frac{\Delta\mu}{\sigma}\right)_{gross}$ criterion for bin performance needs to be tested for consistency for the weak source. The ROC curves for the photopeak energies 625 keV – 675 keV are given in Figure 34 and the remainder of the energies, the energy bin consisting of 0 – 625 keV and 675 keV – 1711 keV, is given in Figure 35.

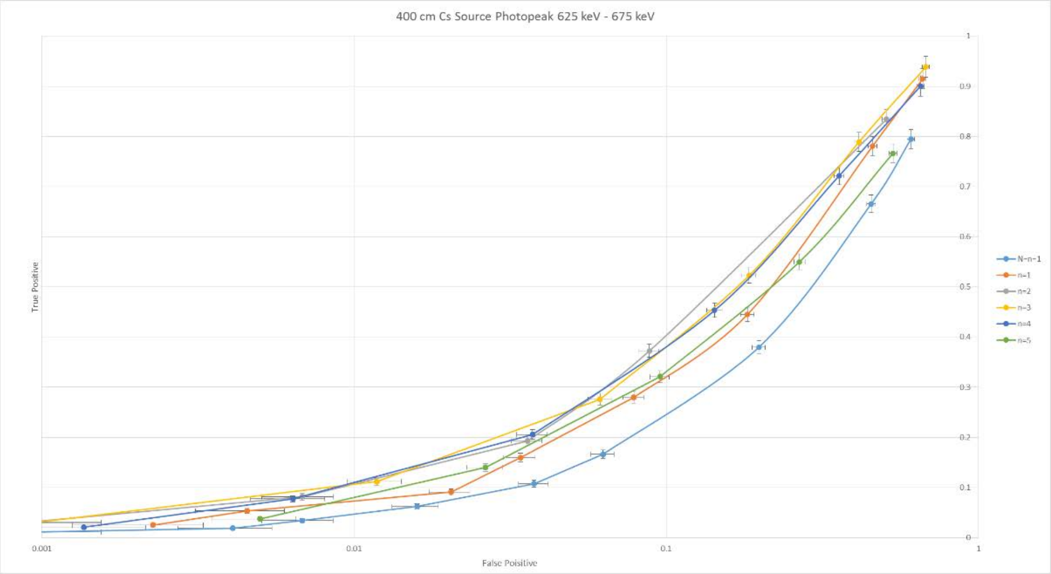


Figure 34: 400 cm ^{137}Cs Source Photopeak Region 625 keV – 675 keV

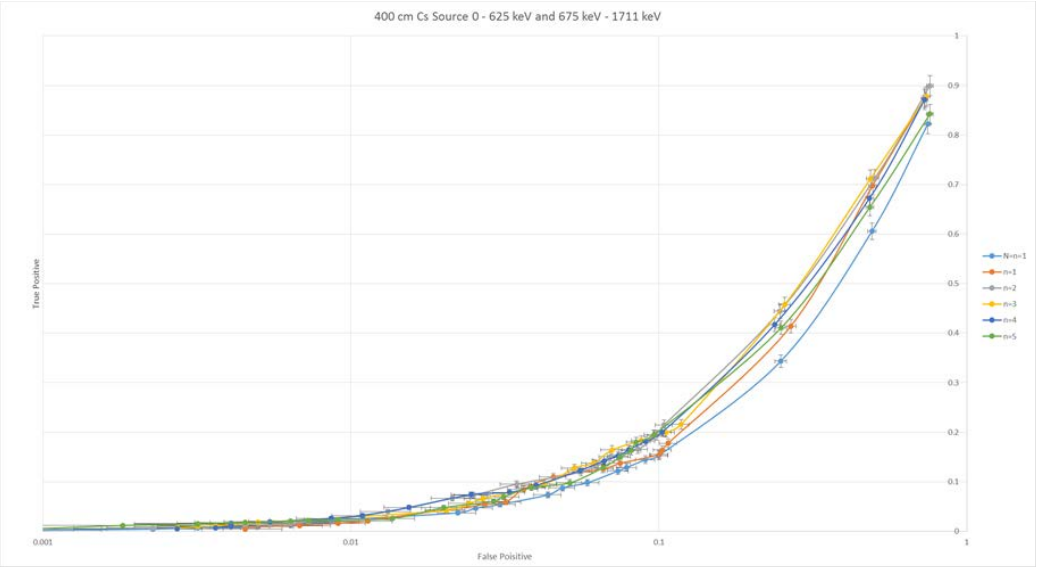


Figure 35: 400 cm ^{137}Cs Source 0 keV – 625 keV and 675 keV – 1711 keV

These sets of curves behave as predicted, with the curves in Figure 34 outperforming the gross counts curves and the curves in Figure 35 underperforming.

Figure 36 graphs the $\frac{\Delta\mu}{\sigma}$ for energy bins of various width centered on the 662 keV photopeak.

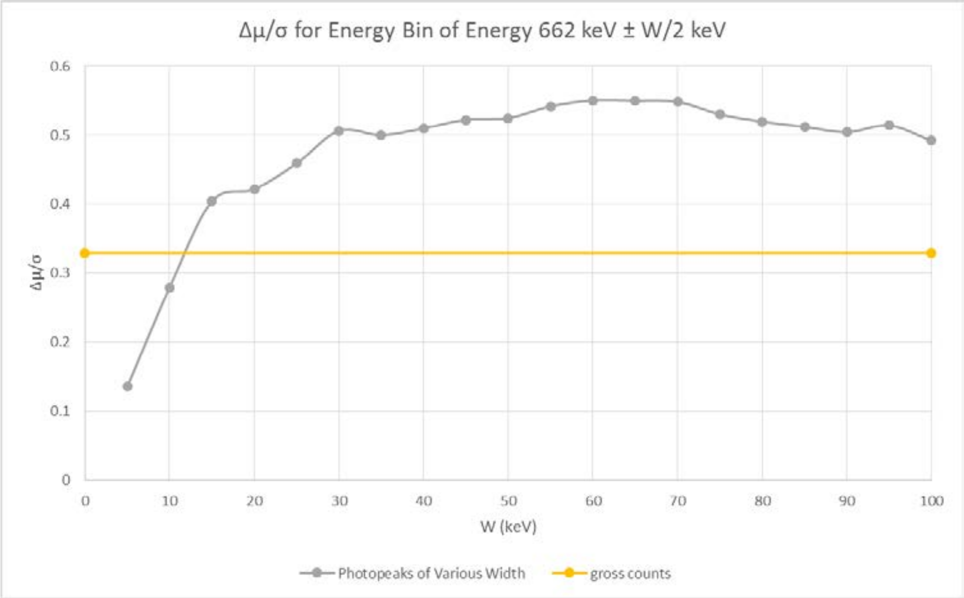


Figure 36: $\frac{\Delta\mu}{\sigma}$ for Energy Bins centered on photopeak region of various widths at 400 cm source strength

This graph shows that the most optimal width is within the 60 keV – 70 keV range. Unsurprisingly, the optimal bin width is larger for the weaker 400 cm ^{137}Cs source. Figure 37 displays the ROC curves for the bin representing the energies 662 keV ± 60 keV.

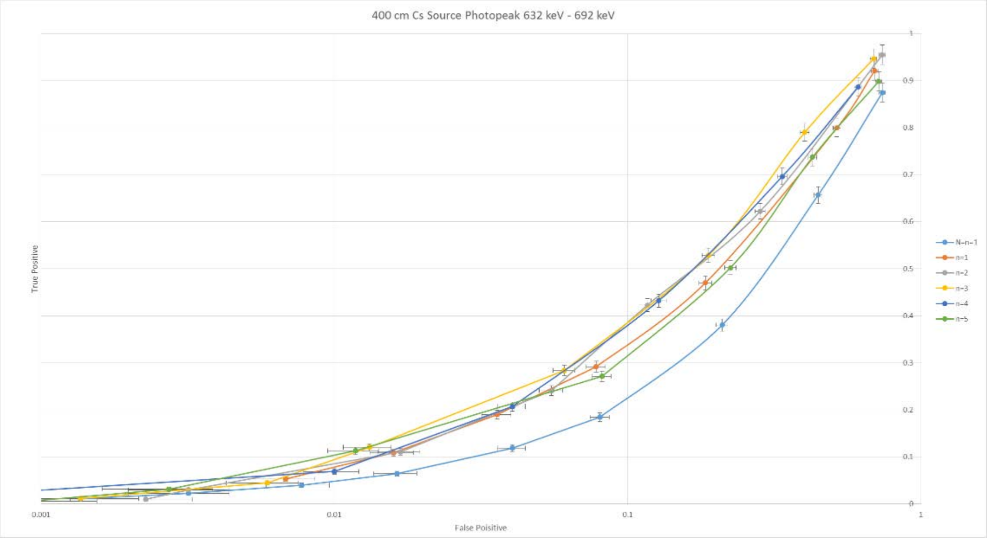


Figure 37: 400 cm ^{137}Cs Source Photopeak Region 632 keV – 692 keV

The ROC curves in Figure 37 are an improvement from the ROC curves from the gross counts in Figure 33; albeit, the improvement is not as strong as that seen in the 200 cm spectra. The remainder of the ROC graphs from Figure 36 is found in Appendix I.

The patterns observed in the 400 cm ^{137}Cs source are very similar to those observed previously. In all cases, the $\frac{\Delta\mu}{\sigma}$ is a good indicator of how well a particular energy bin will perform. Additionally, the relation $\left(\frac{\Delta\mu}{\sigma}\right)_{E_i} > \left(\frac{\Delta\mu}{\sigma}\right)_{gross}$ has thus far been an accurate predictor of whether an energy bin will outperform the gross counts. The same patterns with the data string metrics are also observed: that the rank of performance between each data string metric is unaltered by the energy binning (i.e., there continues to be an independent relationship between energy binning and data string metrics). There does appear to be an exception, however, in the 632 keV – 692 keV ROC curves in Figure 34. In this case, the *n*=5 data string metric performs optimally, albeit barely so.

Mathematics for bin optimization

The $\frac{\Delta\mu}{\sigma}$ for an energy bin is found to be an accurate measure of the overall performance of the ROC curves. In every case, whenever $\left(\frac{\Delta\mu}{\sigma}\right)_{E_i} > \left(\frac{\Delta\mu}{\sigma}\right)_{gross}$, the ROC curves for the bin E_i outperform the ROC curves for the gross counts. The improvement is consistently seen in not just the ROC curves for the *N*=5 data string metrics, but in the ROC curves representing the traditional method as well. Additionally, the $\frac{\Delta\mu}{\sigma}$ is functionally dependent on not just the region of interest on the spectrum, but also the bin width.

It is easy to show that the $\frac{\Delta\mu}{\sigma}$ for a particular bin is dependent on width. Recall in counting statistics that $\sigma_b \cong \sqrt{\mu_b}$, thus, an energy bin of width *W* yields the following: $\left(\frac{\Delta\mu}{\sigma}\right)_W = \frac{\mu_s-\mu_b}{\sqrt{\mu_b}}$, where μ_s and μ_b represent the average source counts and average background counts for a bin of width W, respectively. Therefore, the $\frac{\Delta\mu}{\sigma}$ is functionally dependent on both μ_s and μ_b, hence clarifying the correlation to the region of interest and bin width. If the bin width increases by a factor of *n, and the count rates for both source and background remain constant* (i.e., $\Delta\mu$ increases by a factor of *n*), then the resulting bin of width *nW* yields the following:

$$\left(\frac{\Delta\mu}{\sigma}\right)_{nW} = \frac{n\Delta\mu}{\sqrt{n\mu_b}} = \frac{n}{\sqrt{n}} \cdot \frac{\Delta\mu}{\sqrt{\mu_b}} = \sqrt{n} \cdot \left(\frac{\Delta\mu}{\sigma}\right)_W. \tag{25}$$

Equation 25 proves what is already demonstrated by the data in Figure 25. For the sake of completeness, it can also be shown that by augmenting a bin of width *W* by a factor of *n* to include energies that do not contribute to a signal (i.e., only the background increases by a factor of *n* so $\Delta\mu$ remains constant), the following result is obtained:

$$\left(\frac{\Delta\mu}{\sigma}\right)_{nW} = \frac{\Delta\mu}{\sqrt{n\mu_b}} = \frac{1}{\sqrt{n}} \cdot \left(\frac{\Delta\mu}{\sigma}\right)_W. \tag{26}$$

Equivalently, equation 26 also shows that eliminating energies that contribute only to noise also increases the $\frac{\Delta\mu}{\sigma}$ by a factor of $\sqrt{n}$.

More generally, suppose energy bin *E* is combined with energy bin *E'*, $(E \cup E')$, such that $E \cap E' = \emptyset$, (i.e., the bins do not overlap). There exists a desire to determine when $\left(\frac{\Delta\mu}{\sigma}\right)_E = \left(\frac{\Delta\mu}{\sigma}\right)_{E\cup E'}$. When these two bins are combined, equation 27 satisfies the equality:

$$\frac{\Delta\mu + \Delta\mu'}{\sqrt{\mu_b + {\mu_b}'}} = \frac{\Delta\mu}{\sqrt{\mu_b}}. \tag{27}$$

Following rearrangement, the following relationship is obtained:

$$\frac{\Delta\mu + \Delta\mu'}{\Delta\mu} = \sqrt{\frac{\mu_b + \mu_b'}{\mu_b}}. \tag{28}$$

Equation 28 indicates that the relative increase in $\Delta\mu$ is equal to the square root of the relative increase in μ_b if the $\frac{\Delta\mu}{\sigma}$ is to remain unchanged. Thus, the following criterion is reached, if

$$\frac{\Delta\mu + \Delta\mu'}{\Delta\mu} > \sqrt{\frac{\mu_b + \mu_b'}{\mu_b}}, \tag{29}$$

then the energy bin $E \cup E'$ has improved ROC curves compared to the energy bin *E.* The criterion for subtraction of energies in bin *E'* from bin *E,* $(E - E')$, assuming of course $E' \subset E$, is as follows:

$$\frac{\Delta\mu - \Delta\mu'}{\Delta\mu} > \sqrt{\frac{\mu_b - \mu_b'}{\mu_b}}. \tag{30}$$

If equation 30 is satisfied, then the energy bin $E - E'$ has improved ROC curves compared to the energy bin *E*. Using the criteria represented by equations 29 and 30, the optimization of energy binning from the whole spectrum can be done systematically.

CONCLUSION

The statistical tools generally employed for distinguishing a radioactive source from ambient background take into account individual measurements only. In practice, however, treating each individual measurement as an isolated event isn't always the best representation of the phenomenon being measured. For scenarios where consecutive or contiguous measurements are not irrelevant, the method of source detection regarding a string of measurements as a single event has been found to enhance source detection. Examples of where this new technique may be useful include: the decontamination of a plot of land using multiple measurements, searching for orphan sources, the detection of a weak source with multiple detectors in a short period of time, or discerning long term trends by considering multiple measurements over longer periods of time. Furthermore, this research in particular has shown that this technique of source detection can be further improved when analyzing only certain energies within the spectrum.

In accordance with the traditional criterion for source detection, a radioactive source other than background is said to be present if a measurement result, denoted by *y*, exceeds a decision threshold, *y**. If the measurement *y* is below the decision threshold *y*,* no physical effect other than background is said to be present. The decision threshold *y** is defined such that the probability of a false source identification for any measurement *y* exceeding *y** is less than α. This is symbolically expressed as $P(y > y^* \mid H_0 \text{ is true}) = \alpha$, where the null hypothesis, H_0, represents the scenario where only background radiations are present and α is the preselected false positive probability.

It has been demonstrated that this criterion of source detection can be improved upon when observing a string of measurements. This new "extended" criterion is symbolically expressed as

$$P_{n+/N} = \sum_{i=n}^{N} P(y_1, y_2, \dots, y_i > y^* \mid y_{i+1}, \dots, y_N \le y^* \mid H_0 \text{ is true}) = \alpha$$

which is the probability that at least *n* out of *N* measurements exceed *y** is equal to a preselected false positive probability. In this paper, improvement indicates a higher true positive detection for a given false positive detection. Research has shown that the least optimal *n*/*N* combinations tend to be both 1/*N* and *N*/*N*, with the most optimal combination given by *n*/*N* with $n \gtrsim \frac{N}{2}$ for some value *N*.

This research in particular has demonstrated that further improvements of source detections can be made when considering only certain energies. The Cohen's d $\frac{\Delta\mu}{\sigma}$ measure of effect size is a suitable standard for energy bin performance regarding source detection. Using the $\frac{\Delta\mu}{\sigma}$ for the gross counts as a benchmark, the criterion that $\left(\frac{\Delta\mu}{\sigma}\right)_{E_i} > \left(\frac{\Delta\mu}{\sigma}\right)_{gross}$ accurately indicates whether or not an energy bin E_i will detect a source better than just the gross counts. This comparison holds not only for an energy bin compared to gross counts, but also the comparison of any pair of bins. More generally, two bins can be combined with increasing detection results if the following criterion is met: given energy bins *E* and *E'* such that $E \cap E' = \emptyset$, if

$$\frac{\Delta\mu + \Delta\mu'}{\Delta\mu} > \sqrt{\frac{\mu_b + \mu_b'}{\mu_b}},$$

then the energy bin $E \cup E'$ will have improved source detection compared to the energy bin *E.* The criterion for subtraction of energies in bin *E'* from bin *E,* $(E - E')$, assuming $E' \subset E$, is as follows:

$$\frac{\Delta\mu - \Delta\mu'}{\Delta\mu} > \sqrt{\frac{\mu_b - \mu_b'}{\mu_b}}.$$

The improvement of source detection brought about by energy binning has been shown to be independent of the improvements brought about by the *n*/*N* combinations. Consequently, these can be optimized separately. For very low counts, however, the lower *n* values perform better.

The next step would be to apply these findings to shielded sources and elevated backgrounds. The $\frac{\Delta\mu}{\sigma}$ criterion for energy bin effect size is expected to hold for these different scenarios. It would be likewise useful to test this criterion on different sources altogether.

REFERENCES

[1] J. Neyman, E. S. Pearson. *On the problem of the most efficient tests of statistical hypothesis.* Philosophical Transaction of the Royal Society of London, Series A 238: 289-337; 1933.

[2] L. Currie. *Limits for Qualitative Detection and Quantitative Determination.* Anal Chem 40:586-593; 1968

[3] H.Cember, T. Johnson, *Introduction To Health Physics 4th ed.* The McGraw-Hill Companies, Inc. 2009

[4] G. F. Knoll. *Radiation Detection and Measurement, 4th ed.* John Wiley and Sons, Inc.; 2010.

[5] *International Organization for Standards (ISO) 11929 1st ed.* ISO copyright office; 2010

[6] D. Wackerly, W. Mendenhal, R. Schaeffer, *Mathematical Statistics with Applications, 7th ed.* Belmont, CA: Brooks / Cole CENGAGE Learning; 2008

[7] T. F. Sanquist, P. Doctor, R. Parasuraman, *Designing Effective Alarms for Radiation Detection in Homeland Security Screening,* IEEE Transactions on Systems, Man, and Cybernetics, Part C Nov. 2008, Vol.38(6), pp.856-860

[8] www.desmos.com. Desmos Inc. 2018

[9] J. Brogan, A. Brandl. *Sequential Measurement Strings in Radiation Detection.* Health Physics, in Print

[10] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences 2nd Ed.* Lawrence Erlbaum Associates; 1988

APPENDIX A

Procedure 1: Importing a set of spectra to Excel file "ROC_curve_generator"

In order for this tool to work, there needs to be at least two sets of spectra: the background spectra and the source spectra. This procedure instructs the user on how to import a spectra with the proper format so that the code can read the spectra. The following steps are only the necessary steps that must be followed in order for the code to work.

1. Open the excel file "ROC_curve_generator
2. Create a new sheet and give it an appropriate title. This title will be used in Procedure 2 step 3.
3. Import (copy and paste) the spectra with the following matrix format:
    a. The general format is matrix form, with the channels occupying the rows and the spectra occupying the columns.
    b. Column A is the channel column. The entirety of row 4 is channel 1 for every spectrum, and every subsequent row in column A shall be the subsequent channel up until the last channel (typically a power of 2 such as 512, 1024, or 2048). Cell A3 should be the heading reading "Channel."
    c. Column B is the energy column. The entirety of row 4 must be the highest energy associated with channel 1 for all spectra. This pattern continues for each subsequent row in column B for each channel until the last channel. Cell B3 should be the heading reading "Energy (keV)".
    d. Column C and beyond are the spectrum columns. Starting with C3, the cells should count up from 1 in row 3. This continues until the last spectrum. Every cell within the matrix, (with cell C4 in the upper left corner), has the counts associated with the energy bin (in its corresponding row) and the spectrum (in its corresponding column).
    e. In cell T1, the user must manually enter in the total number of spectra in this sheet.

f. OPTIONAL: in any other cells, the user may wish to enter in any other information, such as titles, headings, and information pertaining to the spectra such as date, place, and time.

Note: Appendix B provides the method utilized by the current researcher along with the macro code.

Procedure 2: Obtaining energy bins

1. Open the excel file “ROC_curve_generator”
2. If not already selected, select the “Energy Bins 2” sheet. This is the main user interface. To facilitate use, the cells are color coded in the following manner:
    a. Blue cells with blue text are user entry cells
    b. Yellow cells are data label cells
    c. White cells are data cells (for both the data headings and the data itself)
    d. Gray cells with orange text are ROC curve data point cells and their associated uncertainties.
3. In the Spectra Entry Block (B27), enter the desired background and source spectra. THE ENTRIES MUST BE VERBATIM FROM THE TABS BELOW. In this version of Excel, the tabs below refer to the spectra sheets. See Procedure 1 for the proper way to import a set of spectra. This step may be omitted if the desired spectra have already been selected.
4. Input energy bins and descriptions.
    a. In the “Energy Bins Entries” block, beginning with cell E3, input the maximum energy in the “Max Energy Column.”

       Example: If cell E4 is set to 100 and cell E5 is set to 200, the entire 5th row (bin 3 in this case), will be the group of energies between 100 keV and 200 keV. Previous entries may need to be deleted.

b. The entries in the "Max Energy" column must be in ascending order.

c. Do not skip any cells.

d. The last entry must be the max energy, given by cell E24, if the actual gross counts of the spectra are to be calculated in bin 21 (row 23). If this is not done, the gross counts will be calculated based on the max energy given.

e. If desired, enter a description to the left of each energy bin.

Example: Cs-137 photopeak

5. Optional: Input a combined bin. This feature allows for any combination of energies in the spectrum, essentially taking the union of the bins entered into the "Combined Bins Entry Block" starting with cell B32.

    a. Enter the bin numbers (column I) in order from left to right without skipping any cells.

    b. The second cell in the block must have an input for this feature to be activated.

The user may take this step following step 7, in which case step 7 must be repeated.

6. Optional: Enter in first spectra and last spectra to be analyzed for both the background and the source spectra in cells AA1 and AD1 respectively for the background, and in cells AA26 and AD26 respectively for the source. The first spectrum entry should be between 1 and the last spectrum entry and the last spectrum entry should be less than or equal to the total number of spectra, calculated in cells X1 and X26 for the background and source respectively. Note that if step 3 was not skipped, the total number of spectra in cells X1 and X26 may not be correct. If any of the entries are out of range, the following algorithm is applied:

- If First Spectrum < 1, then First Spectrum = 1
- If First Spectrum > Total Spectra, then First Spectrum = Total Spectra
- If Last Spectrum < First Spectrum, then Last Spectrum = First Spectrum
- If Last Spectrum > Total Spectra, then Last Spectrum = Total Spectra

The user may take this step following step 7, in which case step 7 must be repeated.

7. Click on the “Get Energy Bin” button to obtain energy bins and statistics. The gross counts are also calculated. The combined bin, if used, is automatically bin 20 (row 22). The gross counts are defaulted as bin 21 (row 23). In the lower left of the sheet, a graph of the modified spectra is provided for both the background and source. Along the abscissa is each bin, and along the ordinate is the average counts for the corresponding bin. The spectra may look abnormal if the bins were not uniformly sized. The macro code executed in this step is given in Appendix C.

Precautionary Note: anytime steps 3, 4, 5, or 6 are executed, step 7 must also be executed, (i.e., the bins must be recalculated).

Procedure 3: Obtain the bin distribution (primary mass function) of the background and source spectra

This procedure generates the primary mass function (PMF) of both the background and source spectra. For background, row 103 gives the number of counts, and row 104 gives the frequency of that number of counts. Likewise, rows 107 and 108 record the same for the source spectra respectively. This function is useful for determining whether or not the spectra resemble a Poisson or Gaussian distribution.

1. Enter the bin number in cell M25.
2. Enter the Width in cell R25. This is useful for bins with a larger number of counts as it will smooth out the PMF.
3. Click on the “Bin Distribution” button. The Macro code executed in this step is given in Appendix D.

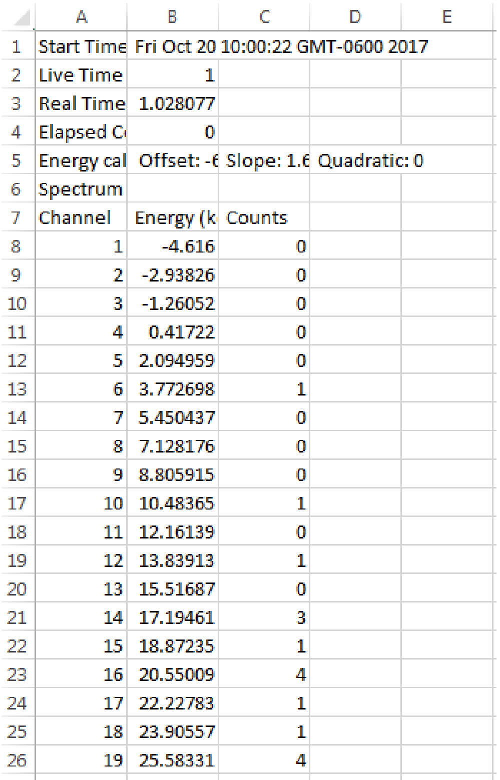
Procedure 4: Obtain ROC Curve

1. Ensure step 7 of procedure 2 has been freshly executed. If the bins have been tampered with in any way, the results may be erroneous.

2. In cell M24, enter in the desired value for N
3. In cell M25, enter in the desired bin to evaluate
4. Compute ROC curves by clicking the “Get ROC Curve” button. The Macro code executed in this step is given in Appendix E.
5. During the computation, the user will observe the data points for each curve being calculated in real time.
6. If the user wishes to save a set of ROC curves, the graph must be copied and pasted using the picture option under the paste options (not the source formatting/default option). If this is not done, the copied ROC curves will be recalculated whenever the program is executed and thus will not be saved.

APPENDIX B

When gathering spectra with the NaI detector, the Canberra software failed to input the data into the Excel software in the desirable matrix format required for data analysis, despite much effort and troubleshooting on part of the researchers. Procedure 1 describes the necessary format only; it does not describe the means by which the format was obtained from these thousands of lines of data. The raw output from Canberra into Excel has the spectra stacked vertically down hundreds of thousands of rows until the 128th spectra, when Canberra terminates. The subsequent spectra are then recorded in similar fashion on different sheets. Figure B shows the initial undesirable format that needs correction, the first 20 channels of the first spectrum are shown.



	A	B	C	D	E
1	Start Time	Fri Oct 20 10:00:22 GMT-0600 2017			
2	Live Time	1			
3	Real Time	1.028077			
4	Elapsed C	0			
5	Energy cal	Offset: -6	Slope: 1.6	Quadratic: 0	
6	Spectrum				
7	Channel	Energy (k	Counts		
8	1	-4.616	0		
9	2	-2.93826	0		
10	3	-1.26052	0		
11	4	0.41722	0		
12	5	2.094959	0		
13	6	3.772698	1		
14	7	5.450437	0		
15	8	7.128176	0		
16	9	8.805915	0		
17	10	10.48365	1		
18	11	12.16139	0		
19	12	13.83913	1		
20	13	15.51687	0		
21	14	17.19461	3		
22	15	18.87235	1		
23	16	20.55009	4		
24	17	22.22783	1		
25	18	23.90557	1		
26	19	25.58331	4		
27					

Figure B: Initial spectra format. Spectrum stacked vertically

The first seven rows contain spectrum information. This gap of seven repeats prior to every spectrum and is represented by the "GAP" variable in the macro; it is set equal to seven. Rows A, B, and C contain the channel numbers, corresponding energies in keV, and the counts respectively. These data were properly organized as follows:

Step 1. Ensure that all the sheets are on one Excel workbook and that each sheet is labeled "sheet1", "sheet2", up until "sheet*x*" where *x* is the maximum number of sheets.

Step 2. Create a new sheet and put as a title, "Combined".

Step 3. Click on the Developer tab in the menu above, (enabling macros might first be necessary), and then click on the Macro button and create a new macro. The title is unimportant.

Step 4. Copy and paste the Macro given at the end of the Appendix. Note that the macro begins with the line "Sub OrganizeData()" and ends with the line "End Sub".

Step 5. Enter in the necessary parameters under the ‘USER ENTRIES block of code. Enter the title of the set of spectra where the line of code begins with “Title =”, ensure that the title is flanked by quotes; the line of code should read: Title = “<*Insert Title Here*>”. Typically, this title contains the spectra type, the person taking the spectra, and the data and time. Additionally, the user must enter in the number of sheets following the

NumberOfSheets= line of code, and the user must enter in the number of channels following the NumberOfChannels= line of code. The variable “Gap” has always been set to 7 in this research, as stated above.

Step 6. Run the macro by clicking on the play button in the upper menu. The program will automatically arrange the data in a logical matrix format on the newly created Combined sheet. The user may simply copy and paste the Combined sheet onto a spectra sheet on the “ROC_curve_generator” workbook in Excel.

The steps above automatically satisfy the requirement in step 3 of procedure 1.

NOTE: All individual lines of code are double spaced. Any characters following an apostrophe (‘) are comments in VBA. Two tabs are used for each subsequent line whenever a single line of code exceeds the page width.

```
Sub OrganizeData()
‘ Variable declarations
Dim StartingRow         As Integer
Dim ChannelColumn       As Integer
Dim i                   As Integer
Dim s                   As Integer
Dim SpecForSheet        As Long
Dim Spectrum            As Long
Dim Gap                 As Integer
Dim sheet               As String
Dim sheetNumber         As String
Dim Title               As String
Dim SheetFinished       As Boolean

' USER ENTRIES ----------------------------------------------------------------------------------------------------
' User enter title here - make sure there are quotes
Title = "1 s Cs-137 Background Spectra taken by J.C. Fischer October 20 2017,  10:26 - 10:52"
NumberOfSheets = 10
NumberOfChannels = 1024
StartingRow = 4                 ' Must be >= 4
ChannelColumn = 1               ' Must be > 0
Gap = 7
Sheets("Combined").Select
' This block of code clears the sheet and reprints title
Cells.Select
Selection.ClearContents
Range("A1:M1").Select
ActiveCell.FormulaR1C1 = Title
' This block of code puts down the column headings
Cells(StartingRow - 1, ChannelColumn).Value = "Channel"
Cells(StartingRow - 1, ChannelColumn + 1).Value = "Energy (keV)"
Cells(StartingRow - 2, ChannelColumn + 2).Value = "Spectrum"
' This block of code puts down the channel numbers
i = 1
Do Until i > NumberOfChannels
   Cells(i + StartingRow - 1, ChannelColumn).Value = i
   i = i + 1
Loop
' This block puts down the energies
i = 1
Do Until i > NumberOfChannels
   Cells(StartingRow + i - 1, ChannelColumn + 1).Value = Sheets("Sheet1").Cells(i + 7, 2).Value
   i = i + 1
Loop
' This block organizes all the spectra in matrix form
```

```
i = 1
s = 1
sheetNumber = s
sheet = "sheet" + sheetNumber
Spectrum = 1
Do Until s > NumberOfSheets
  SpecForSheet = 1
  SheetFinished = False
  Do Until SheetFinished
    Cells(StartingRow - 1, ChannelColumn + Spectrum + 1).Value = Spectrum
    Do Until i > NumberOfChannels
      Cells(StartingRow + i - 1, ChannelColumn + Spectrum + 1).Value =
Sheets(sheet).Cells((SpecForSheet - 1) * NumberOfChannels + (SpecForSheet * Gap) + i, 3).Value
      i = i + 1
    Loop
    Range("O1").Value = s
    Range("P1").Value = SpecForSheet
    SpecForSheet = SpecForSheet + 1
    Spectrum = Spectrum + 1
    i = 1
    If Sheets(sheet).Cells((SpecForSheet - 1) * NumberOfChannels + (SpecForSheet * Gap) + i, 1).Value =
          0 Then
      SheetFinished = True
      s = s + 1
      sheetNumber = s
      sheet = "sheet" + sheetNumber
    End If
  Loop
Loop
‘ This code is necessary to record the total number of spectra
Range("T1").Value = Spectrum - 1
End Sub
```

APPENDIX C

NOTE: All consecutive lines of code are double spaced. All characters following an apostrophe (‘) are a comments, and thus ignored, by VBA. Two tabs are used for each subsequent line whenever a single line of code exceeds the page width. The clicking of the “Get Energy Bins” button executes this code.

```
Sub Get_EBin2()
' This code retrieves the energy bins for the "Energy Bins 2" sheet
Sheets("Energy Bins 2").Select

' Block Variables -----------------------------------------------------------------------------------------------------
Dim InputBlockRow                  As Integer
Dim InputBlockColumn               As Integer
Dim StatsBlockRow                  As Integer
Dim StatsBlockColumn               As Integer
Dim SpectraBlockRow                As Integer
Dim SpectraBlockColumn             As Integer
Dim SpectraInfoBlockRow            As Integer
Dim SpectraInfoBlockColumn         As Integer
Dim CombinedBinBlockRow            As Integer
Dim CombinedBinBlockColumn         As Integer
Dim CurvePointsBlockRow            As Integer
Dim CurvePointsBlockColumn         As Integer
'Values for Block Variables
InputBlockRow = 1                  ' Row 1
InputBlockColumn = 1               ' Column A
StatsBlockRow = 1                   ' Row 1 - Should be parallel with spectra
StatsBlockColumn = 11               ' Column K
SpectraBlockRow = 1                 ' Row 1
SpectraBlockColumn = 19             ' Column S
SpectraInfoBlockRow = 27            ' Row 27
SpectraInfoBlockColumn = 2          ' Column B
CombinedBinBlockRow = 31            ' Row 31
CombinedBinBlockColumn = 2          ' Column B
CurvePointsBlockRow = 51            ' Row 51
CurvePointsBlockColumn = 2          ' Column B
```

```
' Other variables ----------------------------------------------------------------------------------------------------
Dim NumberOfBins                          As Integer
Dim TotalNumberOfBackgroundSpectra        As Integer
Dim TotalNumberOfSourceSpectra            As Integer
Dim FirstBackgroundSpectrum               As Integer
Dim FirstSourceSpectrum                   As Integer
Dim LastBackgroundSpectrum                As Integer
Dim LastSourceSpectrum                    As Integer
Dim NumberOfBackgroundSpectra             As Integer
Dim NumberOfSourceSpectra                 As Integer
Dim BGap                                  As Integer
Dim SGap                                  As Integer
Dim Background                            As String
Dim Source                                As String
Dim bin                                   As Integer
Dim Spectrum                              As Integer
Dim channel                               As Integer
Dim Max                                   As Integer
Dim Min                                   As Integer
Dim sum                                   As Long
Dim TotalSum                              As Long
Dim average                               As Double
Dim Qsum                                  As Double
Dim i                                     As Integer
Dim NoMoreBins                            As Boolean

' Get spectra info
NumberOfBins = Cells(InputBlockRow + 24, InputBlockColumn + 4).Value
Background = Cells(SpectraInfoBlockRow + 1, SpectraInfoBlockColumn + 3).Value
Source = Cells(SpectraInfoBlockRow + 2, SpectraInfoBlockColumn + 3).Value
TotalNumberOfBackgroundSpectra = Sheets(Background).Range("T1").Value
TotalNumberOfSourceSpectra = Sheets(Source).Range("T1").Value
Cells(SpectraBlockRow, SpectraBlockColumn + 5).Value = TotalNumberOfBackgroundSpectra
Cells(SpectraBlockRow + 25, SpectraBlockColumn + 5).Value = TotalNumberOfSourceSpectra

' Clear previous contents ------------------------------------------------------------------------------------------
```

```
Range(Cells(InputBlockRow + 2, InputBlockColumn + 6), Cells(InputBlockRow + 21, InputBlockColumn +
            9)).Select
Selection.ClearContents
Cells(InputBlockRow + 21, InputBlockColumn).Select
Selection.ClearContents
Range(Cells(InputBlockRow + 22, InputBlockColumn + 4), Cells(InputBlockRow + 22, InputBlockColumn +
            9)).Select
Selection.ClearContents
Range(Cells(StatsBlockRow + 2, StatsBlockColumn), Cells(StatsBlockRow + 22, StatsBlockColumn +
            7)).Select
Selection.ClearContents
Range(Cells(StatsBlockRow + 27, StatsBlockColumn), Cells(StatsBlockRow + 48, StatsBlockColumn +
            7)).Select
Selection.ClearContents
Range(Cells(SpectraBlockRow + 2, SpectraBlockColumn), Cells(SpectraBlockRow + 21,
            SpectraBlockColumn + 2)).Select
Selection.ClearContents
Range(Cells(SpectraBlockRow + 27, SpectraBlockColumn), Cells(SpectraBlockRow + 46,
            SpectraBlockColumn + 2)).Select
Selection.ClearContents
Range(Cells(SpectraBlockRow + 1, SpectraBlockColumn + 3), Cells(SpectraBlockRow + 24,
            SpectraBlockColumn + 2500)).Select
Selection.ClearContents
Range(Cells(SpectraBlockRow + 26, SpectraBlockColumn + 3), Cells(SpectraBlockRow + 49,
            SpectraBlockColumn + 2500)).Select
Selection.ClearContents
Range(Cells(CurvePointsBlockRow + 1, CurvePointsBlockColumn), Cells(CurvePointsBlockRow + 23,
            CurvePointsBlockColumn + 33)).Select
Selection.ClearContents
Range(Cells(CurvePointsBlockRow + 27, CurvePointsBlockColumn), Cells(CurvePointsBlockRow + 49,
            CurvePointsBlockColumn + 33)).Select
Selection.ClearContents
Range("A1").Select
Cells(InputBlockRow + 22, InputBlockColumn + 8).Value = 21

' Method to calculate max channels, bin number, total number of spectra
Dim MaxEnergy As Boolean
bin = 1
Do Until bin > NumberOfBins
```

```
   channel = 1
   Max = Cells(InputBlockRow + bin + 1, InputBlockColumn + 4).Value
   MaxEnergy = False
   Do Until MaxEnergy
      If Sheets(Background).Cells(channel + 3, 2).Value > Max Then
         MaxEnergy = True
      Else
         channel = channel + 1
      End If
      If channel > 2050 Then  'This is to avoid an infinite loop
         MaxEnergy = True
      End If
   Loop
   Cells(InputBlockRow + bin + 1, InputBlockColumn + 6).Value = channel
   Cells(SpectraBlockRow + bin + 1, SpectraBlockColumn + 1).Value = Cells(InputBlockRow + bin + 1,
                  InputBlockColumn + 4).Value
   Cells(SpectraBlockRow + bin + 26, SpectraBlockColumn + 1).Value = Cells(InputBlockRow + bin + 1,
                  InputBlockColumn + 4).Value
   Cells(InputBlockRow + bin + 1, InputBlockColumn + 8).Value = bin
   Cells(SpectraBlockRow + bin + 1, SpectraBlockColumn).Value = bin
   Cells(SpectraBlockRow + bin + 26, SpectraBlockColumn).Value = bin
   bin = bin + 1
Loop

'Get spectra sizes and data-----------------------------------------------------------------------------
FirstBackgroundSpectrum = Cells(SpectraBlockRow, SpectraBlockColumn + 8).Value
FirstSourceSpectrum = Cells(SpectraBlockRow + 25, SpectraBlockColumn + 8).Value
LastBackgroundSpectrum = Cells(SpectraBlockRow, SpectraBlockColumn + 11).Value
LastSourceSpectrum = Cells(SpectraBlockRow + 25, SpectraBlockColumn + 11).Value
'Set Lower Bound for FirstBackgroundSpectrum
If FirstBackgroundSpectrum < 1 Then
   FirstBackgroundSpectrum = 1
   Cells(SpectraBlockRow, SpectraBlockColumn + 8).Value = FirstBackgroundSpectrum
End If
'Set Upper bound for FirstBackgroundSpectrum
If FirstBackgroundSpectrum > TotalNumberOfBackgroundSpectra Then
   FirstBackgroundSpectrum = TotalNumberOfBackgroundSpectra
```

```
    Cells(SpectraBlockRow, SpectraBlockColumn + 8).Value = FirstBackgroundSpectrum
End If
'set lower bound for LastBackgroundSpectrum
If LastBackgroundSpectrum < FirstBackgroundSpectrum Then
    LastBackgroundSpectrum = FirstBackgroundSpectrum
    Cells(SpectraBlockRow, SpectraBlockColumn + 11).Value = LastBackgroundSpectrum
End If
'set upper bound for LastBackgroundSpectrum
If LastBackgroundSpectrum > TotalNumberOfBackgroundSpectra Then
    LastBackgroundSpectrum = TotalNumberOfBackgroundSpectra
    Cells(SpectraBlockRow, SpectraBlockColumn + 11).Value = LastBackgroundSpectrum
End If
'Set lower bound for FirstSourceSpectrum
If FirstSourceSpectrum < 1 Then
    FirstSourceSpectrum = 1
    Cells(SpectraBlockRow + 25, SpectraBlockColumn + 8).Value = FirstSourceSpectrum
End If
'set upper bound for FirstSourceSpectrum
If FirstSourceSpectrum > TotalNumberOfSourceSpectra Then
    FirstSourceSpectrum = TotalNumberOfSourceSpectra
    Cells(SpectraBlockRow + 25, SpectraBlockColumn + 8).Value = FirstSourceSpectrum
End If
'set lower bound for LastSourceSpectrum
If LastSourceSpectrum < FirstSourceSpectrum Then
    LastSourceSpectrum = FirstSourceSpectrum
    Cells(SpectraBlockRow + 25, SpectraBlockColumn + 11).Value = LastSourceSpectrum
End If
'set upper bound for LastSourceSpectrum
If LastSourceSpectrum > TotalNumberOfSourceSpectra Then
    LastSourceSpectrum = TotalNumberOfSourceSpectra
    Cells(SpectraBlockRow + 25, SpectraBlockColumn + 11).Value = LastSourceSpectrum
End If
NumberOfBackgroundSpectra = LastBackgroundSpectrum - FirstBackgroundSpectrum + 1
NumberOfSourceSpectra = LastSourceSpectrum - FirstSourceSpectrum + 1
BGap = FirstBackgroundSpectrum - 1
SGap = FirstSourceSpectrum - 1
```

```
' Get spectrum numbers for background ----------------------------------------------------------------------------------
Spectrum = FirstBackgroundSpectrum
Do Until Spectrum > LastBackgroundSpectrum
   Cells(SpectraBlockRow + 1, SpectraBlockColumn + Spectrum + 2 - BGap).Value = Spectrum
   Spectrum = Spectrum + 1
Loop

' These nested Do loops get the background E bins, total, and averages --------------------------------------------
bin = 1
Max = 0
Do Until bin > NumberOfBins
   TotalSum = 0
   Spectrum = FirstBackgroundSpectrum
   Min = Max
   Max = Cells(InputBlockRow + bin + 1, InputBlockColumn + 6).Value
   Do Until Spectrum > LastBackgroundSpectrum
      sum = 0
      channel = Min + 1
      Do Until channel > Max
         sum = sum + Sheets(Background).Cells(channel + 3, Spectrum + 2).Value
         channel = channel + 1
      Loop
      Cells(SpectraBlockRow + bin + 1, SpectraBlockColumn + Spectrum + 2 - BGap).Value = sum
      TotalSum = TotalSum + sum
      Spectrum = Spectrum + 1
   Loop
   Cells(StatsBlockRow + bin + 1, StatsBlockColumn + 6).Value = TotalSum            'Total Counts
   Cells(StatsBlockRow + bin + 1, StatsBlockColumn + 4).Value = TotalSum /
                    NumberOfBackgroundSpectra                                        'Average cell entry
   bin = bin + 1
Loop

' Background Gross Counts and average ---------------------------------------------------------------------------------
TotalSum = 0
Spectrum = FirstBackgroundSpectrum
Do Until Spectrum > LastBackgroundSpectrum
   bin = 1
   sum = 0
   Do Until bin > NumberOfBins
      sum = sum + Cells(SpectraBlockRow + bin + 1, SpectraBlockColumn + Spectrum + 2 - BGap).Value
      bin = bin + 1
   Loop
   Cells(SpectraBlockRow + 22, SpectraBlockColumn + Spectrum + 2 - BGap).Value = sum
   TotalSum = TotalSum + sum
   Spectrum = Spectrum + 1
Loop
Cells(StatsBlockRow + 22, StatsBlockColumn + 6).Value = TotalSum                    'Total Counts
Cells(StatsBlockRow + 22, StatsBlockColumn + 4).Value = TotalSum / NumberOfBackgroundSpectra
                                                                                    'Average cell entry

' Combined Bin Counts, number, description, Average, and SD for Background -----------------------------------
If Cells(CombinedBinBlockRow + 1, CombinedBinBlockColumn + 1).Value > 0 Then
   TotalSum = 0
   Qsum = 0
   average = 0
   Spectrum = FirstBackgroundSpectrum
   Do Until Spectrum > LastBackgroundSpectrum
      NoMoreBins = False
      sum = 0
      i = 0
      Do Until NoMoreBins = True
         If Cells(CombinedBinBlockRow + 1, CombinedBinBlockColumn + i).Value = 0 Then
            NoMoreBins = True
         Else
            bin = Cells(CombinedBinBlockRow + 1, CombinedBinBlockColumn + i).Value
            sum = sum + Cells(SpectraBlockRow + bin + 1, SpectraBlockColumn + Spectrum + 2 -
BGap).Value
            i = i + 1
         End If
      Loop
      Cells(SpectraBlockRow + 21, SpectraBlockColumn + Spectrum + 2 - BGap).Value = sum
      TotalSum = TotalSum + sum
      Spectrum = Spectrum + 1
   Loop
```

```
    Cells(SpectraBlockRow + 21, SpectraBlockColumn + 1).Value = "Combined Bins"
    average = TotalSum / NumberOfBackgroundSpectra
    Spectrum = FirstBackgroundSpectrum
    Do Until Spectrum > LastBackgroundSpectrum
        Qsum = Qsum + (Cells(SpectraBlockRow + 21, SpectraBlockColumn + Spectrum + 2 - BGap).Value -
                    average) ^ 2
        Spectrum = Spectrum + 1
    Loop
    Cells(StatsBlockRow + 21, StatsBlockColumn + 6).Value = TotalSum                  'Total counts
    Cells(StatsBlockRow + 21, StatsBlockColumn + 4).Value = average                   'average
    Cells(StatsBlockRow + 21, StatsBlockColumn + 2).Value = (Qsum / NumberOfBackgroundSpectra) ^ 0.5
                                                                                      'SD
    Cells(InputBlockRow + 21, InputBlockColumn + 8).Value = 20
    Cells(SpectraBlockRow + 21, SpectraBlockColumn).Value = 20
    Cells(InputBlockRow + 21, InputBlockColumn).Value = "Combined Bins"
End If

' Get SD for Background -----------------------------------------------------------------------------
bin = 1
Do Until bin > NumberOfBins
    average = Cells(StatsBlockRow + bin + 1, StatsBlockColumn + 4).Value                    'Average
                    input from cell
    Spectrum = FirstBackgroundSpectrum
    Qsum = 0
    Do Until Spectrum > LastBackgroundSpectrum
        Qsum = Qsum + (Cells(SpectraBlockRow + bin + 1, SpectraBlockColumn + Spectrum + 2 -
                    BGap).Value - average) ^ 2
        Spectrum = Spectrum + 1
    Loop
    Cells(StatsBlockRow + bin + 1, StatsBlockColumn + 2).Value = (Qsum / NumberOfBackgroundSpectra)
                    ^ 0.5    'SD cell entry
    bin = bin + 1
Loop

' SD for gross background counts --------------------------------------------------------------------
average = Cells(StatsBlockRow + 22, StatsBlockColumn + 4).Value                          'Average input
Spectrum = FirstBackgroundSpectrum
Qsum = 0
```

```
Do Until Spectrum > LastBackgroundSpectrum
   Qsum = Qsum + (Cells(SpectraBlockRow + 22, SpectraBlockColumn + Spectrum + 2 - BGap).Value -
                      average) ^ 2
   Spectrum = Spectrum + 1
Loop
Cells(StatsBlockRow + 22, StatsBlockColumn + 2).Value = (Qsum / NumberOfBackgroundSpectra) ^ 0.5
                      'SD cell entry

' Get spectrum numbers for Source ------------------------------------------------------------------------------------
Spectrum = FirstSourceSpectrum
Do Until Spectrum > LastSourceSpectrum
   Cells(SpectraBlockRow + 26, SpectraBlockColumn + Spectrum + 2 - SGap).Value = Spectrum
   Spectrum = Spectrum + 1
Loop

' These nested Do loops get the source E bins and averages -----------------------------------------------------------
bin = 1
Max = 0
Do Until bin > NumberOfBins
   TotalSum = 0
   Spectrum = FirstSourceSpectrum
   Min = Max
   Max = Cells(InputBlockRow + bin + 1, InputBlockColumn + 6).Value
   Do Until Spectrum > LastSourceSpectrum
      sum = 0
      channel = Min + 1
      Do Until channel > Max
         sum = sum + Sheets(Source).Cells(channel + 3, Spectrum + 2).Value
         channel = channel + 1
      Loop
      Cells(SpectraBlockRow + bin + 26, SpectraBlockColumn + Spectrum + 2 - SGap).Value = sum
      TotalSum = TotalSum + sum
      Spectrum = Spectrum + 1
   Loop
   Cells(StatsBlockRow + bin + 26, StatsBlockColumn + 6).Value = TotalSum                    'Total Counts
   Cells(StatsBlockRow + bin + 26, StatsBlockColumn + 4).Value = TotalSum / NumberOfSourceSpectra
                                 'Average cell entry
   bin = bin + 1
```

```
Loop

' Source Gross Counts ----------------------------------------------------------------------------------------------
TotalSum = 0
Spectrum = FirstSourceSpectrum
Do Until Spectrum > LastSourceSpectrum
   bin = 1
   sum = 0
   Do Until bin > NumberOfBins
      sum = sum + Cells(SpectraBlockRow + bin + 26, SpectraBlockColumn + Spectrum + 2 - SGap).Value
      bin = bin + 1
   Loop
   Cells(SpectraBlockRow + 47, SpectraBlockColumn + Spectrum + 2 - SGap).Value = sum
   TotalSum = TotalSum + sum
   Spectrum = Spectrum + 1
Loop
Cells(StatsBlockRow + 47, StatsBlockColumn + 6).Value = TotalSum                              'Total Counts
Cells(StatsBlockRow + 47, StatsBlockColumn + 4).Value = TotalSum / NumberOfSourceSpectra
                                'Average cell entry

' Combined Bin Counts, Average, and SD for Source ----------------------------------------------------------------------
If Cells(CombinedBinBlockRow + 1, CombinedBinBlockColumn + 1).Value > 0 Then
   TotalSum = 0
   Qsum = 0
   average = 0
   Spectrum = FirstSourceSpectrum
   Do Until Spectrum > LastSourceSpectrum
      NoMoreBins = False
      sum = 0
      i = 0
      Do Until NoMoreBins = True
         If Cells(CombinedBinBlockRow + 1, CombinedBinBlockColumn + i).Value = 0 Then
            NoMoreBins = True
         Else
            bin = Cells(CombinedBinBlockRow + 1, CombinedBinBlockColumn + i).Value
            sum = sum + Cells(SpectraBlockRow + bin + 26, SpectraBlockColumn + Spectrum + 2 -
                     SGap).Value
            i = i + 1
         End If
      Loop
      TotalSum = TotalSum + sum
      Cells(SpectraBlockRow + 46, SpectraBlockColumn + Spectrum + 2 - SGap).Value = sum
      Spectrum = Spectrum + 1
   Loop
   Cells(SpectraBlockRow + 46, SpectraBlockColumn + 1).Value = "Combined Bins"
   Cells(SpectraBlockRow + 46, SpectraBlockColumn).Value = 20
   average = TotalSum / NumberOfSourceSpectra
   Spectrum = FirstSourceSpectrum
   Do Until Spectrum > LastSourceSpectrum
      Qsum = Qsum + (Cells(SpectraBlockRow + 46, SpectraBlockColumn + Spectrum + 2 - SGap).Value -
                  average) ^ 2
      Spectrum = Spectrum + 1
   Loop
   Cells(StatsBlockRow + 46, StatsBlockColumn + 6).Value = TotalSum                  'Total Counts
   Cells(StatsBlockRow + 46, StatsBlockColumn + 4).Value = average                   'Average
   Cells(StatsBlockRow + 46, StatsBlockColumn + 2).Value = (Qsum / NumberOfSourceSpectra) ^ 0.5
                                                                                    'SD
End If

' SD for Source ----------------------------------------------------------------------------------
bin = 1
Do Until bin > NumberOfBins
   average = Cells(StatsBlockRow + bin + 26, StatsBlockColumn + 4).Value                     'Average
                  input from cell
   Spectrum = FirstSourceSpectrum
   Qsum = 0
   Do Until Spectrum > LastSourceSpectrum
      Qsum = Qsum + (Cells(SpectraBlockRow + bin + 26, SpectraBlockColumn + Spectrum + 2 -
                  SGap).Value - average) ^ 2
      Spectrum = Spectrum + 1
   Loop
   Cells(StatsBlockRow + bin + 26, StatsBlockColumn + 2).Value = (Qsum / NumberOfSourceSpectra) ^ 0.5
                           'SD cell entry
   bin = bin + 1
Loop
```

```
' SD for source gross counts ------------------------------------------------------------------------
average = Cells(StatsBlockRow + 47, StatsBlockColumn + 4).Value                    'Average input from cell
Spectrum = FirstSourceSpectrum
Qsum = 0
Do Until Spectrum > LastSourceSpectrum
   Qsum = Qsum + (Cells(SpectraBlockRow + 47, SpectraBlockColumn + Spectrum + 2 - SGap).Value -
                  average) ^ 2
   Spectrum = Spectrum + 1
Loop
Cells(StatsBlockRow + 47, StatsBlockColumn + 2).Value = (Qsum / NumberOfSourceSpectra) ^ 0.5
                              'SD cell entry

' Get Delta Mu / Sigma
bin = 1
Do Until bin > NumberOfBins
   Cells(StatsBlockRow + bin + 26, StatsBlockColumn + 3).Value = (Cells(StatsBlockRow + bin + 26,
                  StatsBlockColumn + 4).Value - Cells(StatsBlockRow + bin + 1, StatsBlockColumn +
                  4).Value) / Cells(StatsBlockRow + bin + 1, StatsBlockColumn + 2).Value
   bin = bin + 1
Loop
Cells(StatsBlockRow + 47, StatsBlockColumn + 3).Value = (Cells(StatsBlockRow + 47, StatsBlockColumn +
                  4).Value - Cells(StatsBlockRow + 22, StatsBlockColumn + 4).Value) / Cells(StatsBlockRow
                  + 22, StatsBlockColumn + 2).Value
If Cells(CombinedBinBlockRow + 1, CombinedBinBlockColumn + 1).Value > 0 Then
  Cells(StatsBlockRow + 46, StatsBlockColumn + 3).Value = (Cells(StatsBlockRow + 46, StatsBlockColumn
                  + 4).Value - Cells(StatsBlockRow + 21, StatsBlockColumn + 4).Value) /
                  Cells(StatsBlockRow + 21, StatsBlockColumn + 2).Value
End If

' Get Source/Background Ratio
bin = 1
Do Until bin > NumberOfBins
   If Cells(StatsBlockRow + bin + 1, StatsBlockColumn + 4).Value = 0 Then
      Cells(StatsBlockRow + bin + 1, StatsBlockColumn).Value = "-"
   Else
      Cells(StatsBlockRow + bin + 1, StatsBlockColumn).Value = Cells(StatsBlockRow + bin + 26,
                  StatsBlockColumn + 4).Value / Cells(StatsBlockRow + bin + 1, StatsBlockColumn +
                  4).Value - 1
```

```
    End If
    bin = bin + 1
Loop
If Cells(CombinedBinBlockRow + 1, CombinedBinBlockColumn + 1).Value > 0 Then
    Cells(StatsBlockRow + 21, StatsBlockColumn).Value = Cells(StatsBlockRow + 46, StatsBlockColumn +
                        4).Value / Cells(StatsBlockRow + 21, StatsBlockColumn + 4).Value - 1
End If
Cells(StatsBlockRow + 22, StatsBlockColumn).Value = Cells(StatsBlockRow + 47, StatsBlockColumn +
                        4).Value / Cells(StatsBlockRow + 22, StatsBlockColumn + 4).Value - 1

' Get Source/Background (bin) to Source/Background (Gross) Ratio
bin = 1
Do Until bin > NumberOfBins
    If Cells(StatsBlockRow + bin + 26, StatsBlockColumn + 4).Value = 0 Then
        Cells(StatsBlockRow + bin + 26, StatsBlockColumn).Value = "-"
    Else
        Cells(StatsBlockRow + bin + 26, StatsBlockColumn).Value = Cells(StatsBlockRow + bin + 1,
                        StatsBlockColumn).Value / Cells(StatsBlockRow + 22, StatsBlockColumn).Value - 1
    End If
    bin = bin + 1
Loop
If Cells(CombinedBinBlockRow + 1, CombinedBinBlockColumn + 1).Value > 0 Then
   Cells(StatsBlockRow + 46, StatsBlockColumn).Value = Cells(StatsBlockRow + 21,
                        StatsBlockColumn).Value / Cells(StatsBlockRow + 22, StatsBlockColumn).Value - 1
End If

' Testing block -----------------------------------------------------------------------------------------------------
Range("G24").Select
End Sub
```

APPENDIX D

NOTE: All consecutive lines of code are double spaced. All characters following an apostrophe (‘) are a comments, and thus ignored, by VBA. Two tabs are used for each subsequent line whenever a single line of code exceeds the page width. The clicking of the “Bin Distribution” button executes this code.

```
Sub BinDistribution()

Dim StatsBlockRow               As Integer
Dim StatsBlockColumn            As Integer
Dim BackgroundEnergyRow         As Integer
Dim SourceEnergyRow             As Integer
Dim SpectraBlockRow             As Integer
Dim SpectraBlockColumn          As Integer

Dim bin                         As Integer
Dim Max                         As Integer
Dim Min                         As Integer
Dim TotalBackgroundSpectra      As Integer
Dim TotalSourceSpectra          As Integer
Dim Spectrum                    As Integer
Dim N                           As Integer
Dim Counts                      As Integer
Dim Width                       As Integer
Dim i                           As Integer
Dim BackgroundMean              As Double
Dim SourceMean                  As Double

StatsBlockRow = 1
StatsBlockColumn = 11
BackgroundEnergyRow = 103
SourceEnergyRow = 107
SpectraBlockRow = 1
SpectraBlockColumn = 19

Width = Range("R25").Value
bin = Range("M25").Value
```

```
BackgroundMean = Cells(StatsBlockRow + bin + 1, StatsBlockColumn + 4).Value
SourceMean = Cells(StatsBlockRow + bin + 26, StatsBlockColumn + 4).Value

Range(Cells(BackgroundEnergyRow, 1), Cells(BackgroundEnergyRow + 1, 500)).Select
Selection.ClearContents
Range(Cells(SourceEnergyRow, 1), Cells(SourceEnergyRow + 1, 500)).Select
Selection.ClearContents
'Range("J102").ClearContents
'Range("J106").ClearContents
Range("J102").Value = BackgroundMean
Range("J106").Value = SourceMean

' Background
TotalBackgroundSpectra = Cells(SpectraBlockRow, SpectraBlockColumn + 11).Value -
                    Cells(SpectraBlockRow, SpectraBlockColumn + 8).Value + 1
' Find Max
Spectrum = 1
Max = 0
Do Until Spectrum = TotalBackgroundSpectra + 1
   If Cells(SpectraBlockRow + bin + 1, SpectraBlockColumn + Spectrum + 2).Value > Max Then
      Max = Cells(SpectraBlockRow + bin + 1, SpectraBlockColumn + Spectrum + 2).Value
   End If
   Spectrum = Spectrum + 1
Loop
' Find Min
Spectrum = 1
Min = Max
Do Until Spectrum = TotalBackgroundSpectra + 1
   If Cells(SpectraBlockRow + bin + 1, SpectraBlockColumn + Spectrum + 2).Value < Min Then
      Min = Cells(SpectraBlockRow + bin + 1, SpectraBlockColumn + Spectrum + 2).Value
   End If
   Spectrum = Spectrum + 1
Loop
' Lay Abscissa
Spectrum = Min
i = 0
Do Until i > (Max - Min) / Width
```

```
   Cells(BackgroundEnergyRow, i + 1).Value = Spectrum
   Spectrum = Spectrum + Width
   i = i + 1
Loop

' Get PMF
Counts = 1
Do Until Counts > (Max - Min + 2) / Width + 1
   N = 0
   i = 0
   Do Until i = Width
      Spectrum = 1
      Do Until Spectrum = TotalBackgroundSpectra + 1
         If Cells(SpectraBlockRow + bin + 1, SpectraBlockColumn + Spectrum + 2).Value =
                      Cells(BackgroundEnergyRow, Counts).Value + i Then
            N = N + 1
         End If
         Spectrum = Spectrum + 1
      Loop
      i = i + 1
   Loop
   Cells(BackgroundEnergyRow + 1, Counts).Value = N
   Counts = Counts + 1
Loop

' Source
TotalSourceSpectra = Cells(SpectraBlockRow + 25, SpectraBlockColumn + 11).Value -
                      Cells(SpectraBlockRow + 25, SpectraBlockColumn + 8).Value + 1
' Find Max
Spectrum = 1
Max = 0
Do Until Spectrum = TotalSourceSpectra + 1
   If Cells(SpectraBlockRow + bin + 26, SpectraBlockColumn + Spectrum + 2).Value > Max Then
      Max = Cells(SpectraBlockRow + bin + 26, SpectraBlockColumn + Spectrum + 2).Value
   End If
   Spectrum = Spectrum + 1
Loop
```

```
' Find Min
Spectrum = 1
Min = Max
Do Until Spectrum = TotalSourceSpectra + 1
  If Cells(SpectraBlockRow + bin + 26, SpectraBlockColumn + Spectrum + 2).Value < Min Then
    Min = Cells(SpectraBlockRow + bin + 26, SpectraBlockColumn + Spectrum + 2).Value
  End If
  Spectrum = Spectrum + 1
Loop
' Lay Abscissa
Spectrum = Min
i = 0
Do Until i > (Max - Min) / Width
  Cells(SourceEnergyRow, i + 1).Value = Spectrum
  Spectrum = Spectrum + Width
  i = i + 1
Loop
' Get PMF
Counts = 1
Do Until Counts > (Max - Min + 2) / Width + 1
  N = 0
  i = 0
  Do Until i = Width
    Spectrum = 1
    Do Until Spectrum = TotalSourceSpectra + 1
      If Cells(SpectraBlockRow + bin + 26, SpectraBlockColumn + Spectrum + 2).Value =
                  Cells(SourceEnergyRow, Counts).Value + i Then
        N = N + 1
      End If
      Spectrum = Spectrum + 1
    Loop
    i = i + 1
  Loop
  Cells(SourceEnergyRow + 1, Counts).Value = N
  Counts = Counts + 1
Loop
```

```
Range("A1").Select

End Sub
```

APPENDIX E

NOTE: All consecutive lines of code are double spaced. All characters following an apostrophe (‘) are comments, and thus ignored, by VBA. Two tabs are used for each subsequent line whenever a single line of code exceeds the page width. The clicking of the “Get ROC Curve” button executes this code.

```
Sub Get_ROC_Curve2()
' This code computes the ROC curves for the "Energy Bins 2" sheet
Sheets("Energy Bins 2").Select
Range("A52").Select

' Block Variables -----------------------------------------------------------------------------------------------
Dim InputBlockRow                    As Integer
Dim InputBlockColumn                 As Integer
Dim StatsBlockRow                    As Integer
Dim StatsBlockColumn                 As Integer
Dim SpectraBlockRow                  As Integer
Dim SpectraBlockColumn               As Integer
Dim SpectraInfoBlockRow              As Integer
Dim SpectraInfoBlockColumn           As Integer
Dim CombinedBinBlockRow              As Integer
Dim CombinedBinBlockColumn           As Integer
Dim CurvePointsBlockRow              As Integer
Dim CurvePointsBlockColumn           As Integer
'Values for Block Variables
InputBlockRow = 1                ' Row 1
InputBlockColumn = 1             ' Column A
StatsBlockRow = 1             ' Row 1 - Should be parallel with spectra
StatsBlockColumn = 11         ' Column K
SpectraBlockRow = 1           ' Row 1
SpectraBlockColumn = 19       ' Column S
SpectraInfoBlockRow = 27      ' Row 27
SpectraInfoBlockColumn = 2    ' Column B
CombinedBinBlockRow = 31      ' Row 31
CombinedBinBlockColumn = 2    ' Column B
CurvePointsBlockRow = 51      ' Row 51
CurvePointsBlockColumn = 2    ' Column B
```

```
Dim N                        As Integer
Dim bin                      As Integer
Dim Spectrum                 As Integer
Dim PoissonCutoff            As Integer  ' This variable determines when the Gausian or Poisson is used in
                                               calculating Ystar
Dim alpha                     As Integer  ' 1<=alpha<=22
Dim Ystar                    As Double
Dim k                        As Integer
Dim i                        As Integer
Dim sum                      As Integer
Dim Bsum                     As Integer
Dim Ssum                     As Integer
Dim average                  As Double
Dim NumberOfBackgroundSpectra          As Integer
Dim NumberOfSoucespectra              As Integer
Dim LastCurve                         As Boolean  ' This variable is used for last N=n=1 curve - avoids
                                                        infinite loop

' Get Parameters
PoissonCutoff = 30
N = Range("M24").Value
bin = Range("M25").Value
average = Cells(StatsBlockRow + bin + 1, StatsBlockColumn + 4).Value
NumberOfBackgroundSpectra = Cells(SpectraBlockRow, SpectraBlockColumn + 11).Value -
                Cells(SpectraBlockRow, SpectraBlockColumn + 8).Value + 1
NumberOfSourceSpectra = Cells(SpectraBlockRow + 25, SpectraBlockColumn + 11).Value -
                Cells(SpectraBlockRow + 25, SpectraBlockColumn + 8).Value + 1

' Clear contents for computation
Range(Cells(SpectraBlockRow + 2, SpectraBlockColumn + NumberOfBackgroundSpectra + 4),
                Cells(SpectraBlockRow + 24, SpectraBlockColumn + (2 * NumberOfBackgroundSpectra) +
                12)).Clear
Range(Cells(SpectraBlockRow + 27, SpectraBlockColumn + NumberOfSourceSpectra + 4),
                Cells(SpectraBlockRow + 49, SpectraBlockColumn + (2 * NumberOfSourceSpectra) +
                12)).Clear
Range(Cells(CurvePointsBlockRow + 1, CurvePointsBlockColumn), Cells(CurvePointsBlockRow + 23,
                CurvePointsBlockColumn + 21)).ClearContents

' Calculate k ROC curves
```

```
LastCurve = False
k = 1
Do Until k > N + 1
   If k = N + 1 Then       ' This calculates last N=n=1 (traditional) curve
      LastCurve = True
      N = 1
      k = 1
   End If
   'Calculate One Curve
   alpha = 1
   Do Until alpha > 23
      'The If statements calculate Y* using either a poisson or gaussian
      If average > PoissonCutoff Then
         Ystar = average + Sheets("Alpha Values").Cells(alpha + 31, (N * (N - 1) + 2) / 2 + k - 1).Value *
                 Cells(StatsBlockRow + bin + 1, StatsBlockColumn + 2).Value
      Else
         Sheets("Poisson Ystar Calculator").Range("B2").Value = Sheets("Alpha Values").Cells(alpha + 4, (N
                 * (N - 1) + 2) / 2 + k - 1).Value
         Sheets("Poisson Ystar Calculator").Range("B1").Value = Cells(StatsBlockRow + bin + 1,
                 StatsBlockColumn + 4).Value
         Ystar = Sheets("Poisson Ystar Calculator").Range("H1").Value
      End If

      ' This computes number of positives in a bin - 1st Computation - independent of N or k
      ' Background
      Spectrum = 1
      Do Until Spectrum > NumberOfBackgroundSpectra
         If Cells(SpectraBlockRow + bin + 1, SpectraBlockColumn + Spectrum + 2).Value > Ystar Then
            Cells(SpectraBlockRow + bin + 1, SpectraBlockColumn + NumberOfBackgroundSpectra +
                 Spectrum + 3).Value = 1
         Else
            Cells(SpectraBlockRow + bin + 1, SpectraBlockColumn + NumberOfBackgroundSpectra +
                 Spectrum + 3).Value = 0
         End If
         Spectrum = Spectrum + 1
      Loop
      i = 1
      Do Until i > N - 1  'This loop repeats the first N-1 entries in the first computation
         Cells(SpectraBlockRow + bin + 1, SpectraBlockColumn + NumberOfBackgroundSpectra +
                 NumberOfBackgroundSpectra + i + 3).Value = Cells(SpectraBlockRow + bin + 1,
                 SpectraBlockColumn + NumberOfBackgroundSpectra + i + 3).Value
         i = i + 1
      Loop

      ' This records all of the y*s
      If LastCurve Then
         N = Range("M24").Value
         k = N + 1
         Range("W76").Value = k
      End If
      Cells(51 + alpha, 24 + k).Value = Ystar
      If LastCurve Then
         N = 1
         k = 1
      End If

      ' Source
      Spectrum = 1
      Do Until Spectrum > NumberOfSourceSpectra
         If Cells(SpectraBlockRow + bin + 26, SpectraBlockColumn + Spectrum + 2).Value > Ystar Then
            Cells(SpectraBlockRow + bin + 26, SpectraBlockColumn + NumberOfSourceSpectra + Spectrum
                 + 3).Value = 1
         Else
            Cells(SpectraBlockRow + bin + 26, SpectraBlockColumn + NumberOfSourceSpectra + Spectrum
                 + 3).Value = 0
         End If
         Spectrum = Spectrum + 1
      Loop
      i = 1
      Do Until i > N - 1  'This loop repeats the first N-1 entries in the first computation
         Cells(SpectraBlockRow + bin + 26, SpectraBlockColumn + (2 * NumberOfSourceSpectra) + i +
                 3).Value = Cells(SpectraBlockRow + bin + 26, SpectraBlockColumn +
                 NumberOfSourceSpectra + i + 3).Value
         i = i + 1
      Loop
```

```
' The sums the various strings of length N - 2nd Computation
' Background
Spectrum = 1
Do Until Spectrum > NumberOfBackgroundSpectra
  i = 0
  sum = 0
  Do Until i > N - 1
    sum = sum + Cells(SpectraBlockRow + bin + 1, SpectraBlockColumn +
          NumberOfBackgroundSpectra + Spectrum + i + 3).Value
    i = i + 1
  Loop
  Cells(SpectraBlockRow + bin + 2, SpectraBlockColumn + NumberOfBackgroundSpectra + Spectrum
          + 3).Value = sum
  Spectrum = Spectrum + 1
Loop
' Source
Spectrum = 1
Do Until Spectrum > NumberOfSourceSpectra
  i = 0
  sum = 0
  Do Until i > N - 1
    sum = sum + Cells(SpectraBlockRow + bin + 26, SpectraBlockColumn + NumberOfSourceSpectra
          + Spectrum + i + 3).Value
    i = i + 1
  Loop
  Cells(SpectraBlockRow + bin + 27, SpectraBlockColumn + NumberOfSourceSpectra + Spectrum +
          3).Value = sum
  Spectrum = Spectrum + 1
Loop

' This computes the number of N strings of k or more positives - 3rd Computation
' Background
Spectrum = 1
Bsum = 0
Do Until Spectrum > NumberOfBackgroundSpectra
  If Cells(SpectraBlockRow + bin + 2, SpectraBlockColumn + NumberOfBackgroundSpectra +
          Spectrum + 3).Value > k - 1 Then
    Cells(SpectraBlockRow + bin + 3, SpectraBlockColumn + NumberOfBackgroundSpectra +
             Spectrum + 3).Value = 1
  Else
    Cells(SpectraBlockRow + bin + 3, SpectraBlockColumn + NumberOfBackgroundSpectra +
             Spectrum + 3).Value = 0
  End If
  Bsum = Bsum + Cells(SpectraBlockRow + bin + 3, SpectraBlockColumn +
             NumberOfBackgroundSpectra + Spectrum + 3).Value
  Spectrum = Spectrum + 1
Loop
'Source
Spectrum = 1
Ssum = 0
Do Until Spectrum > NumberOfSourceSpectra
  If Cells(SpectraBlockRow + bin + 27, SpectraBlockColumn + NumberOfSourceSpectra + Spectrum +
             3).Value > k - 1 Then
    Cells(SpectraBlockRow + bin + 28, SpectraBlockColumn + NumberOfSourceSpectra + Spectrum
             + 3).Value = 1
  Else
    Cells(SpectraBlockRow + bin + 28, SpectraBlockColumn + NumberOfSourceSpectra + Spectrum
             + 3).Value = 0
  End If
  Ssum = Ssum + Cells(SpectraBlockRow + bin + 28, SpectraBlockColumn + NumberOfSourceSpectra
             + Spectrum + 3).Value
  Spectrum = Spectrum + 1
Loop

' Get Point for ROC Curve
If LastCurve Then           'This If/Then is redundant for N=10, necessary for other N values
  Cells(CurvePointsBlockRow + alpha, CurvePointsBlockColumn + 20).Value = Bsum /
             NumberOfBackgroundSpectra
  Cells(CurvePointsBlockRow + alpha, CurvePointsBlockColumn + 21).Value = Ssum /
             NumberOfSourceSpectra
Else
  Cells(CurvePointsBlockRow + alpha, CurvePointsBlockColumn + 2 * (k - 1)).Value = Bsum /
             NumberOfBackgroundSpectra
  Cells(CurvePointsBlockRow + alpha, CurvePointsBlockColumn + 2 * (k - 1) + 1).Value = Ssum /
             NumberOfSourceSpectra
End If
```

```
    ' Get Error Points
    If LastCurve Then             'This If/Then is redundant for N=10, necessary for other N values
      Cells(CurvePointsBlockRow + alpha + 26, CurvePointsBlockColumn + 20).Value = ((Bsum) ^ 0.5) /
              NumberOfBackgroundSpectra
      Cells(CurvePointsBlockRow + alpha + 26, CurvePointsBlockColumn + 21).Value = ((Ssum) ^ 0.5) /
              NumberOfSourceSpectra
    Else
      Cells(CurvePointsBlockRow + alpha + 26, CurvePointsBlockColumn + 2 * (k - 1)).Value = ((Bsum) ^
              0.5) / NumberOfBackgroundSpectra
      Cells(CurvePointsBlockRow + alpha + 26, CurvePointsBlockColumn + 2 * (k - 1) + 1).Value =
              ((Ssum) ^ 0.5) / NumberOfSourceSpectra
    End If
    alpha = alpha + 1
  Loop

  If LastCurve Then          ' this restores previous N and k values to avoid infinite loop
    N = Range("M24").Value
    k = N + 1
  End If
  k = k + 1
Loop

' Clear computational content
'Range(Cells(SpectraBlockRow + 2, SpectraBlockColumn + NumberOfBackgroundSpectra + 4),
              Cells(SpectraBlockRow + 24, SpectraBlockColumn + (2 * NumberOfBackgroundSpectra) +
              12)).Clear
'Range(Cells(SpectraBlockRow + 27, SpectraBlockColumn + NumberOfSourceSpectra + 4),
              Cells(SpectraBlockRow + 49, SpectraBlockColumn + (2 * NumberOfSourceSpectra) +
              12)).Clear

Range("A143").Select
End Sub
```

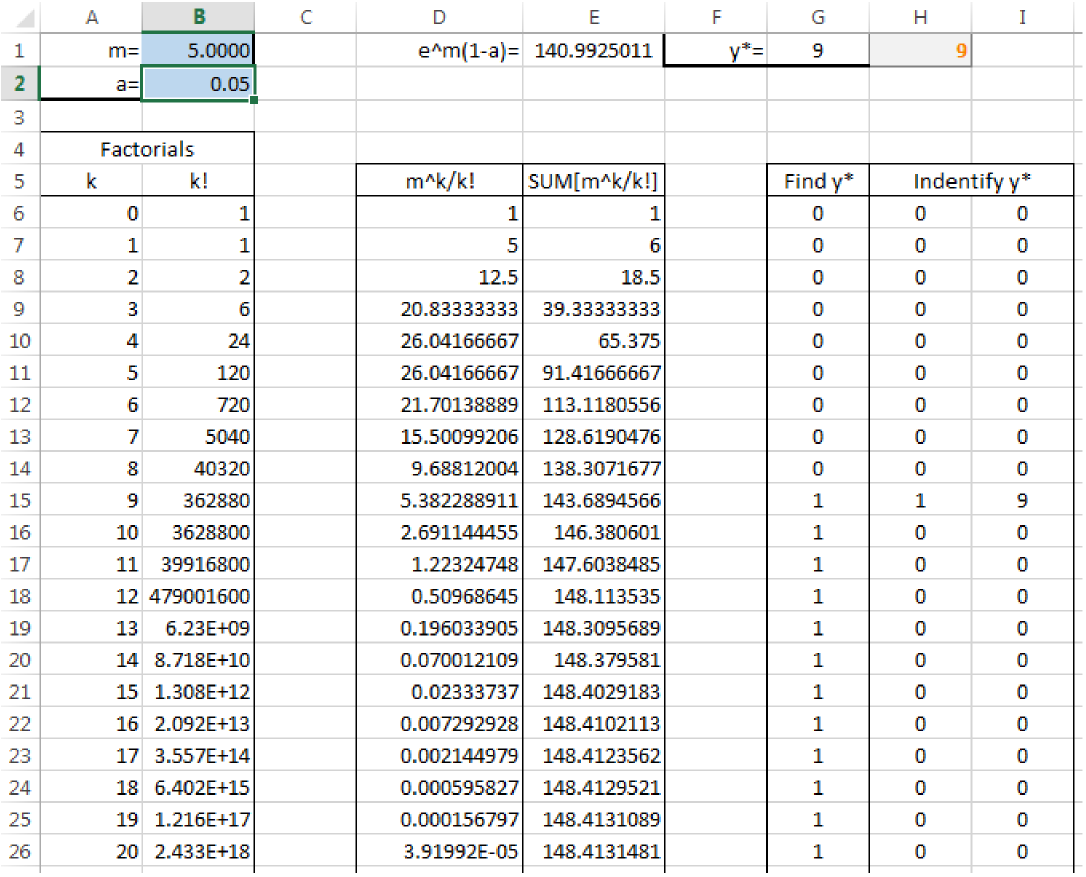
APPENDIX F

Recall from the introduction that the value of *y** is given by the relation

$$e^{\mu}(1-\alpha) < \sum_{i=0}^{n+1} \frac{\mu_b^i}{i!} \tag{F}$$

such that $n \in \{y^*, y^* + 1, y^* + 2, \ldots\}$ where *y** is defined as the lowest bound of the solution set. This is the iterative method that is used in calculating y* using the Excel tool. Each successive sum is calculated beginning with *i* = 0 until equation (F) is satisfied for the first time.

Figure F shows a portion of the y* calculator page. The mean and alpha are input into cells B1 and B2 respectively. The number $e^{\mu}(1-\alpha)$ is easily calculated and given in Cell E1. Columns A and B, beginning with row 4, calculate *k!* (the variable *k* is used in the place of *i* in this spreadsheet). Beginning with cell A6, the natural numbers are given down to 50. The cells in the B column calculate the product of the values in the cells above and directly to the left, thus resulting in *k!*. Column D calculates the $\frac{\mu^k}{k!}$ term for each *k* by raising the mean in cell B1 to the *k* power given in the cell exactly three to the left, then dividing by the value of the cells exactly two to the left (*k!*). Column E then computes the sum of the terms $\frac{\mu^k}{k!}$ by adding the values of the cell directly above to the cell directly to the left. The cells in column G contain an If-then statement that gives a value of 1 if the cell exactly two to the left is greater than cell E1. This first 1 in column G corresponds to *y**. Column H simply takes the difference between the cell to the left and the cell above to the left, thus translating the first 1 in column G to column H and calculating a 0 otherwise. Column I is an If-then statement such that a value of 1 gives the *k* value seven cells to the right in column A (*k=y**), and leaves a zero otherwise. Cell G1 contains an If-then statement that sets G1 equal to zero if B1 equals zero (to avoid computation error), else G1 equals the sum of all the values in column I, thus setting G1 equal to *y*.* Finally, H1 contains an If-then statement that sets H1 equal to 1 if G1 is zero (since it doesn't make sense to have a *y** equal to zero), or sets H1 equal to G1 otherwise. H1, then, is the final output for the *y** calculator.



	A	B	C	D	E	F	G	H	I
1	m=	5.0000		e^m(1-a)=	140.9925011	y*=	9	9	
2	a=	0.05							
3									
4	Factorials								
5	k	k!		m^k/k!	SUM[m^k/k!]		Find y*	Indentify y*	
6	0	1		1	1		0	0	0
7	1	1		5	6		0	0	0
8	2	2		12.5	18.5		0	0	0
9	3	6		20.83333333	39.33333333		0	0	0
10	4	24		26.04166667	65.375		0	0	0
11	5	120		26.04166667	91.41666667		0	0	0
12	6	720		21.70138889	113.1180556		0	0	0
13	7	5040		15.50099206	128.6190476		0	0	0
14	8	40320		9.68812004	138.3071677		0	0	0
15	9	362880		5.382288911	143.6894566		1	1	9
16	10	3628800		2.691144455	146.380601		1	0	0
17	11	39916800		1.22324748	147.6038485		1	0	0
18	12	479001600		0.50968645	148.113535		1	0	0
19	13	6.23E+09		0.196033905	148.3095689		1	0	0
20	14	8.718E+10		0.070012109	148.379581		1	0	0
21	15	1.308E+12		0.02333737	148.4029183		1	0	0
22	16	2.092E+13		0.007292928	148.4102113		1	0	0
23	17	3.557E+14		0.002144979	148.4123562		1	0	0
24	18	6.402E+15		0.000595827	148.4129521		1	0	0
25	19	1.216E+17		0.000156797	148.4131089		1	0	0
26	20	2.433E+18		3.91992E-05	148.4131481		1	0	0

Figure F: y* calculator sheet

APPENDIX G

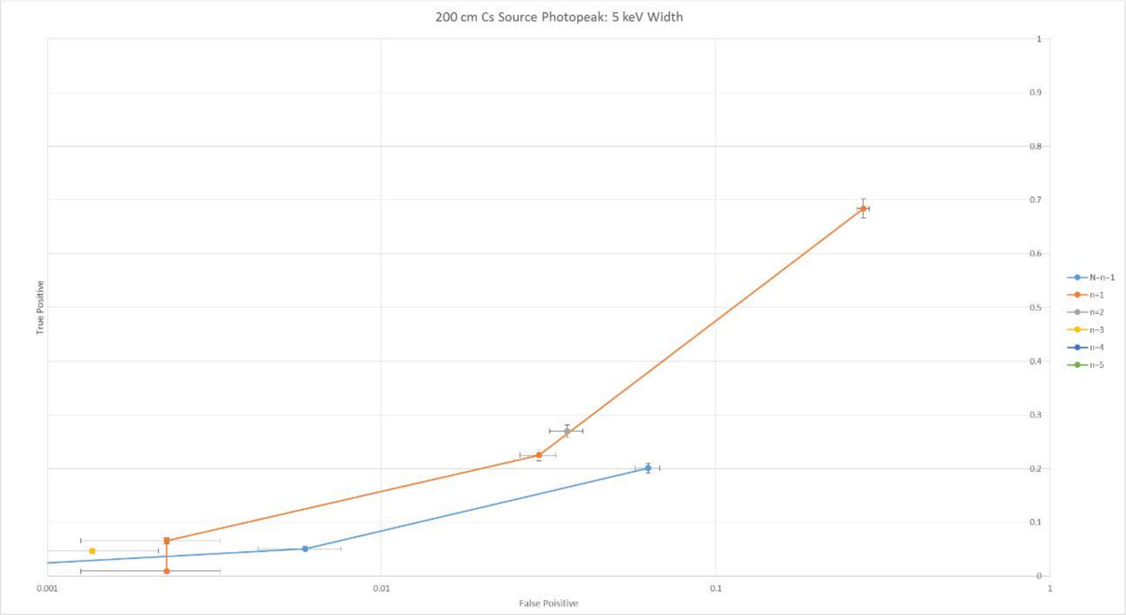


Figure G.1: 200 cm ^{137}Cs Source Photopeak Region – 5 keV Width

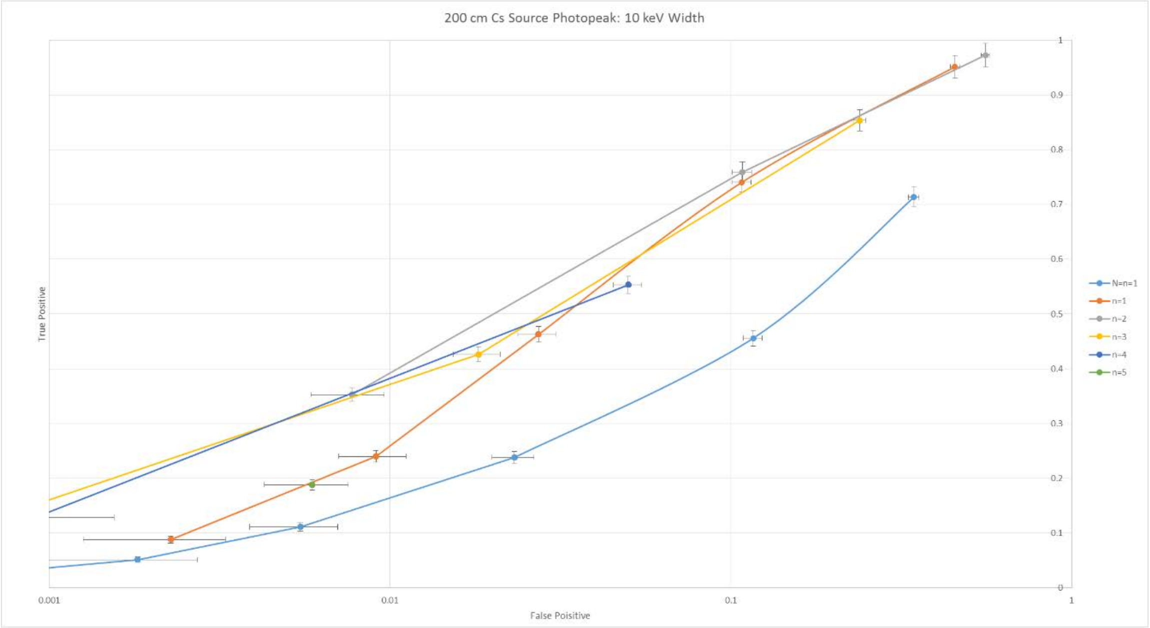


Figure G.2: 200 cm ^{137}Cs Source Photopeak Region – 10 keV Width

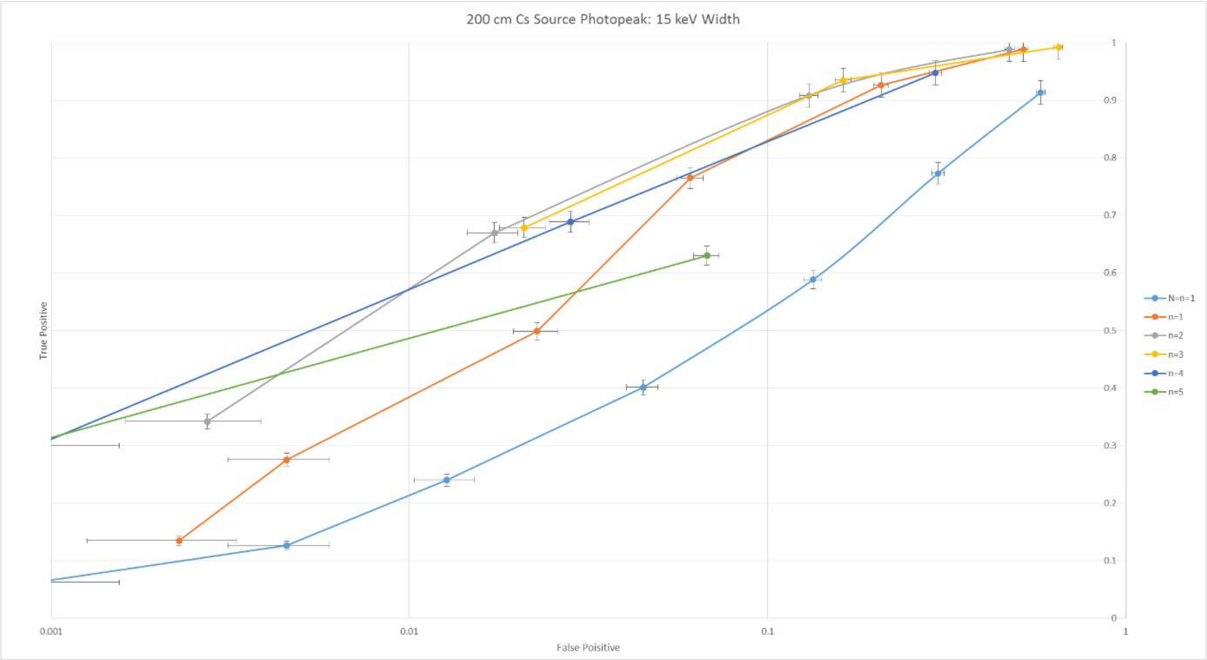


Figure G.3: 200 cm ^{137}Cs Source Photopeak Region – 15 keV Width

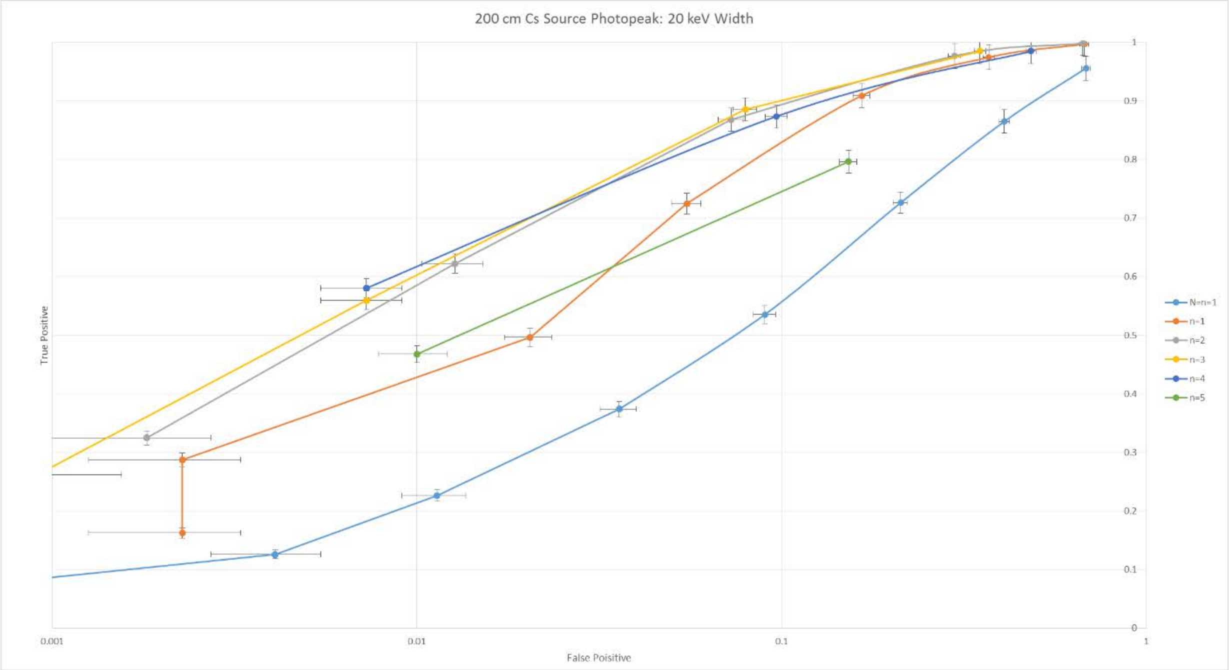


Figure G.4: 200 cm ^{137}Cs Source Photopeak Region – 20 keV Width

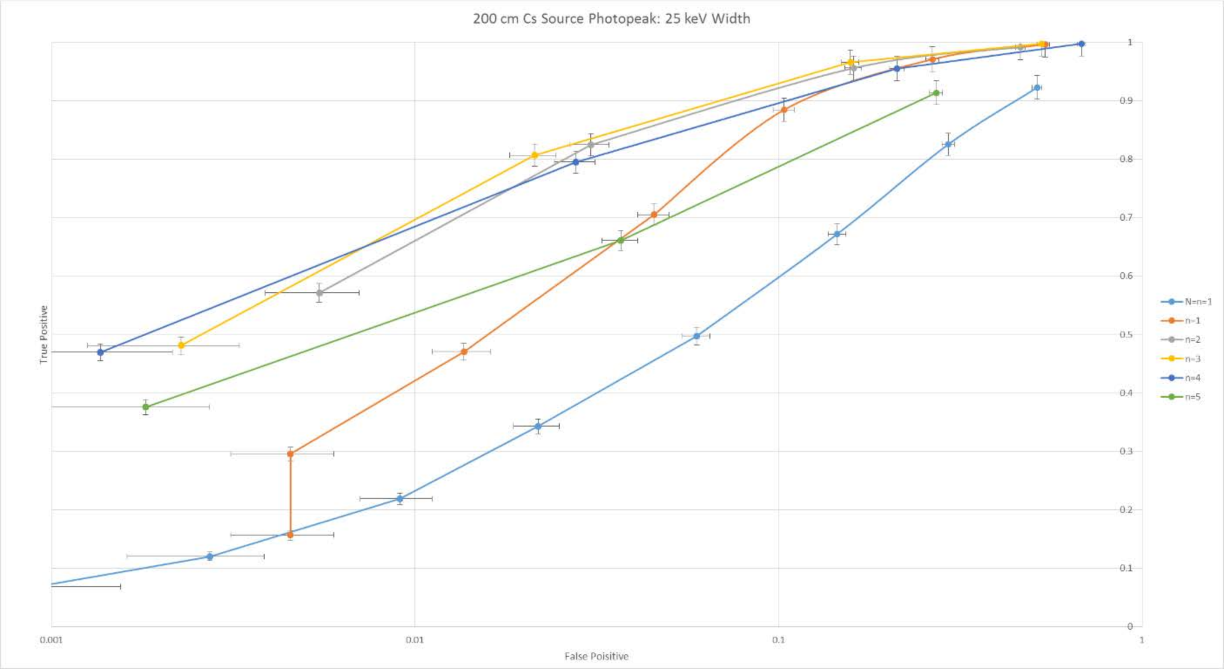


Figure G.5: 200 cm ^{137}Cs Source Photopeak Region – 25 keV Width

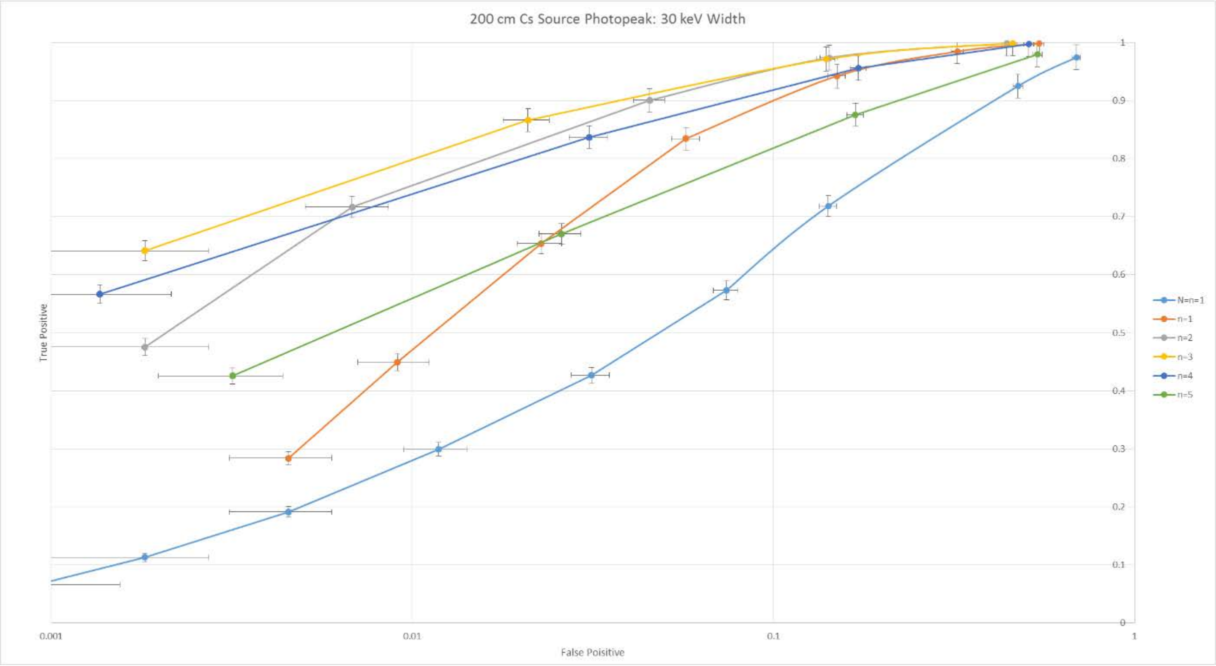


Figure G.6: 200 cm ^{137}Cs Source Photopeak Region – 30 keV Width

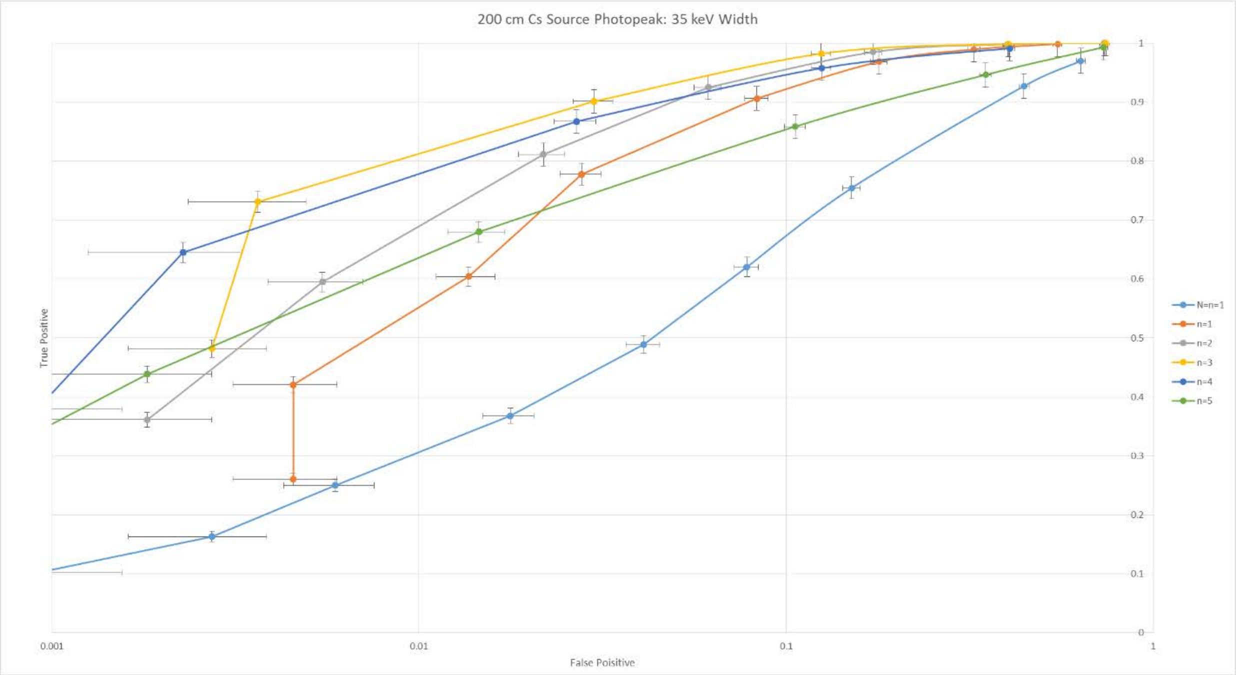


Figure G.7: 200 cm ^{137}Cs Source Photopeak Region – 35 keV Width

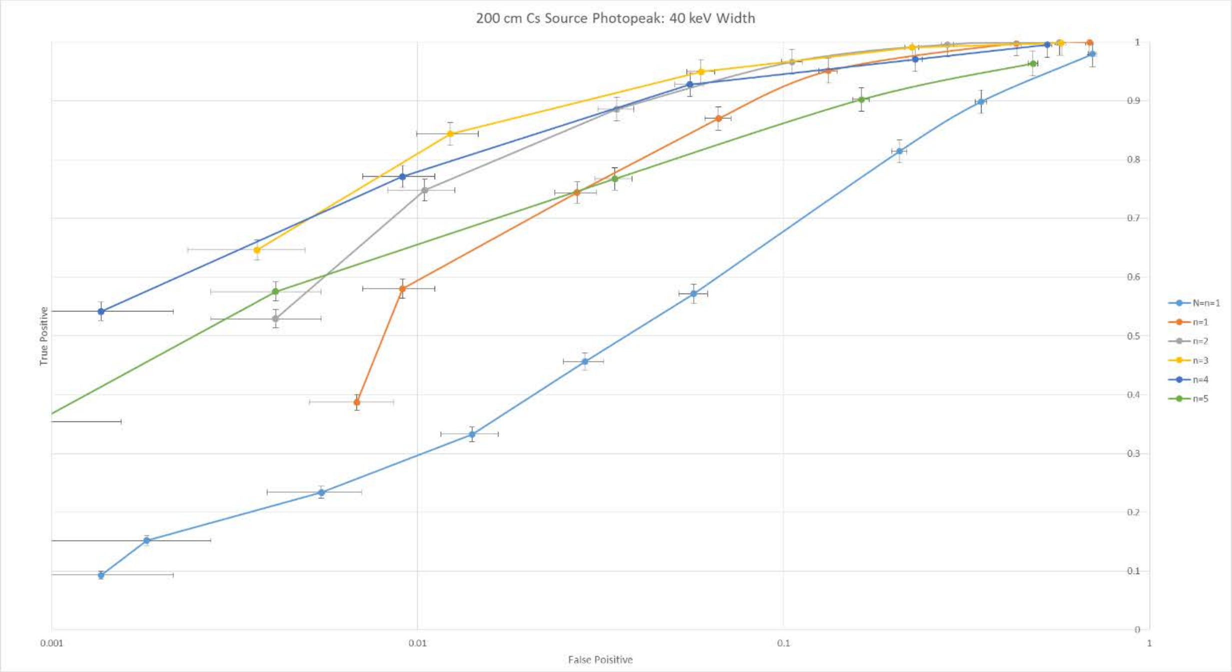


Figure G.8: 200 cm ^{137}Cs Source Photopeak Region – 40 keV Width

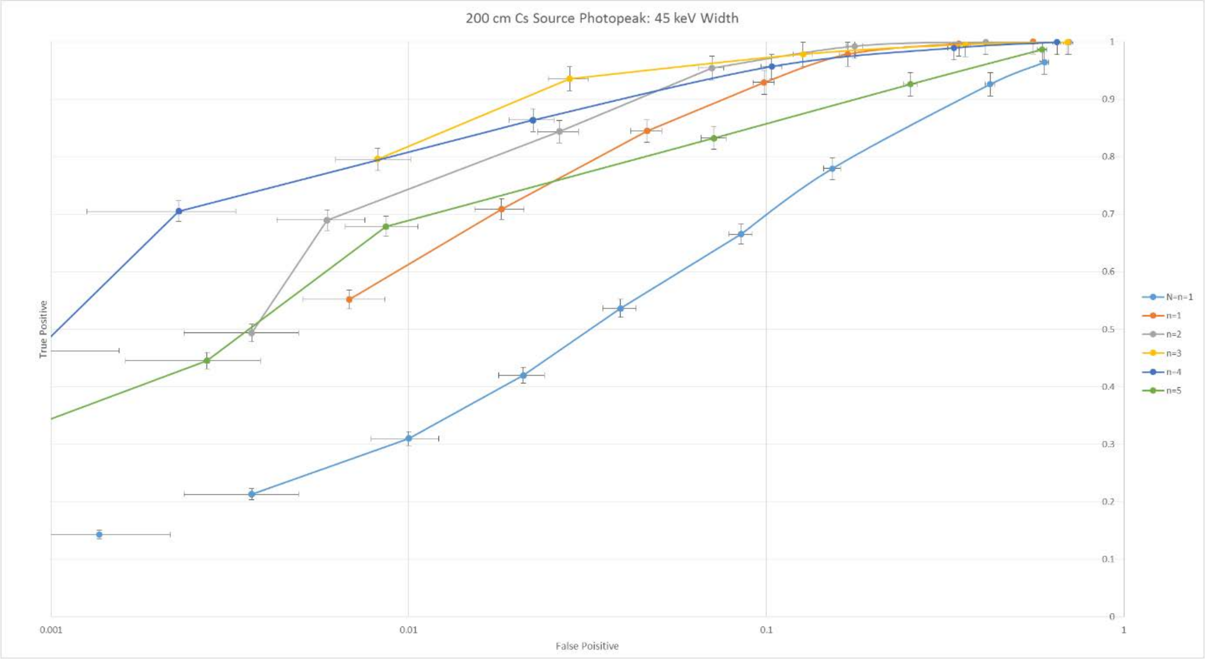


Figure G.9: 200 cm ^{137}Cs Source Photopeak Region – 45 keV Width

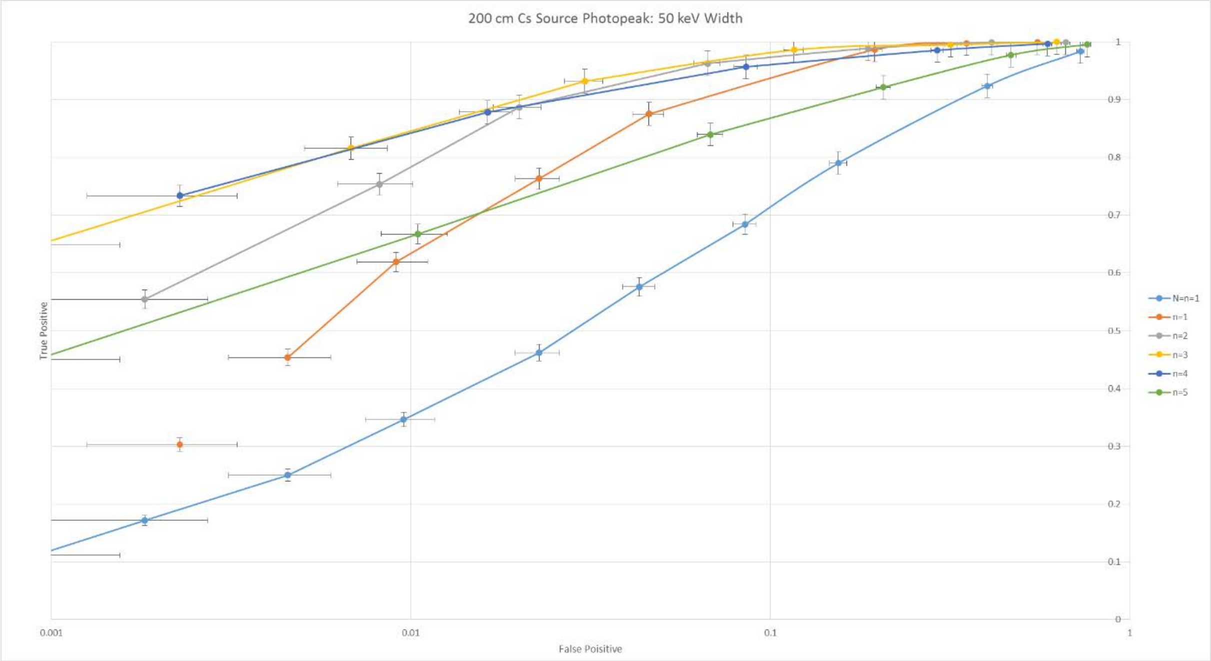


Figure G.10: 200 cm ^{137}Cs Source Photopeak Region – 50 keV Width

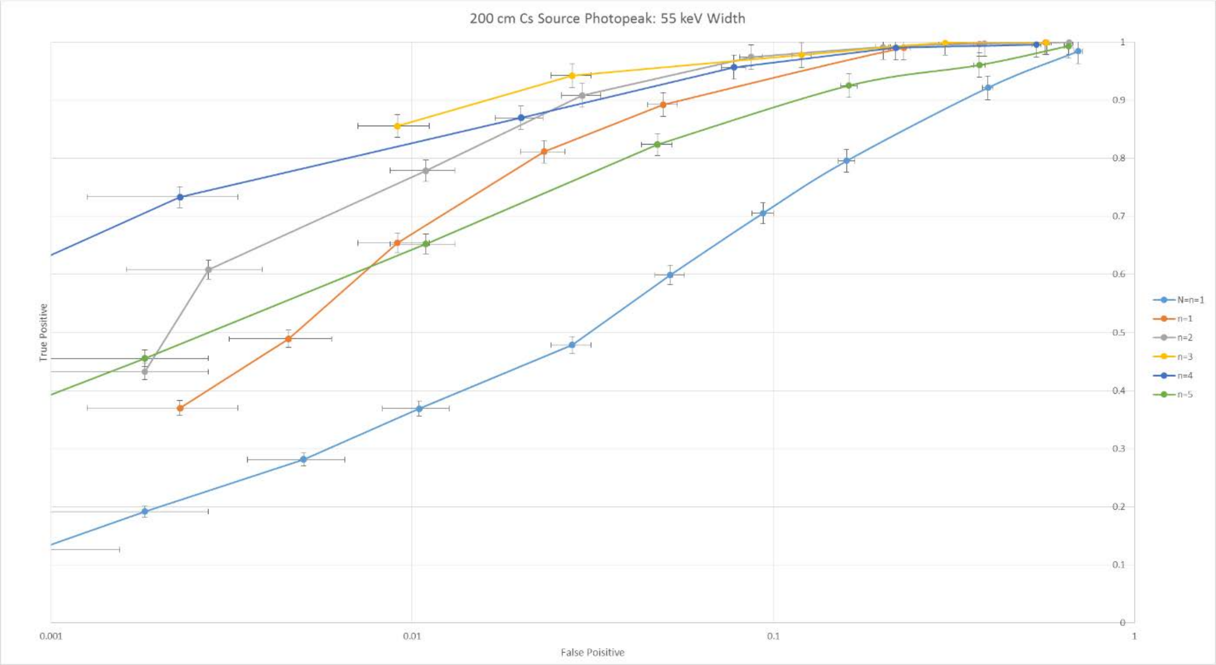


Figure G.11: 200 cm ^{137}Cs Source Photopeak Region – 55 keV Width

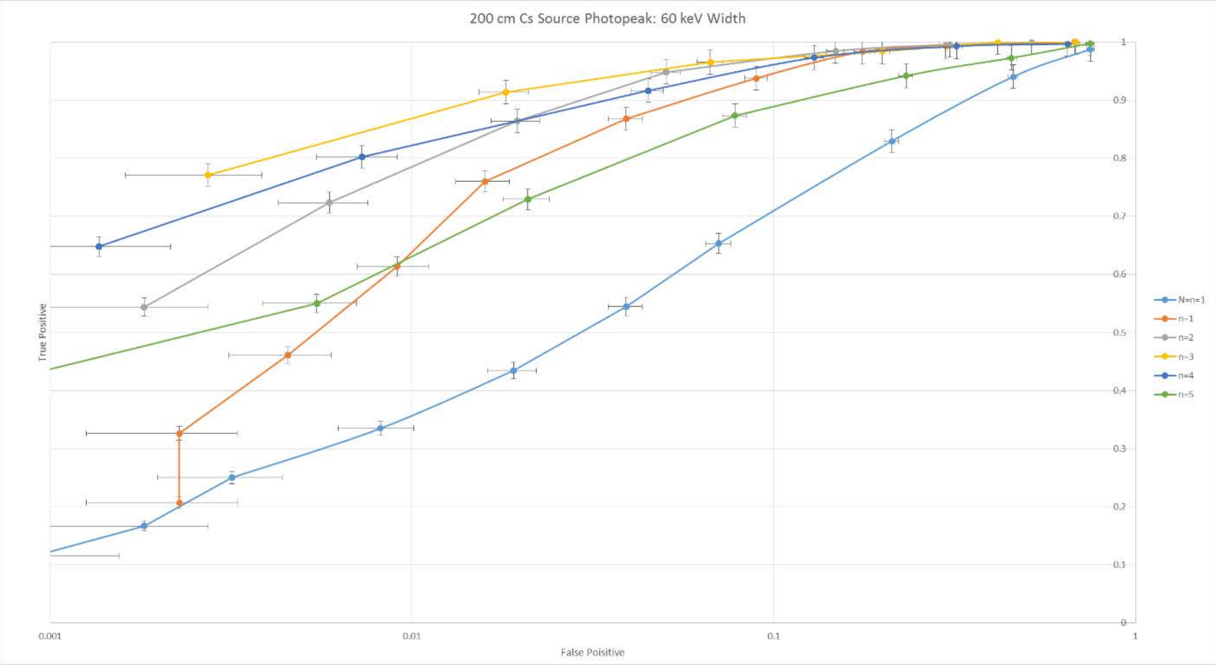


Figure G.12: 200 cm ^{137}Cs Source Photopeak Region – 60 keV Width

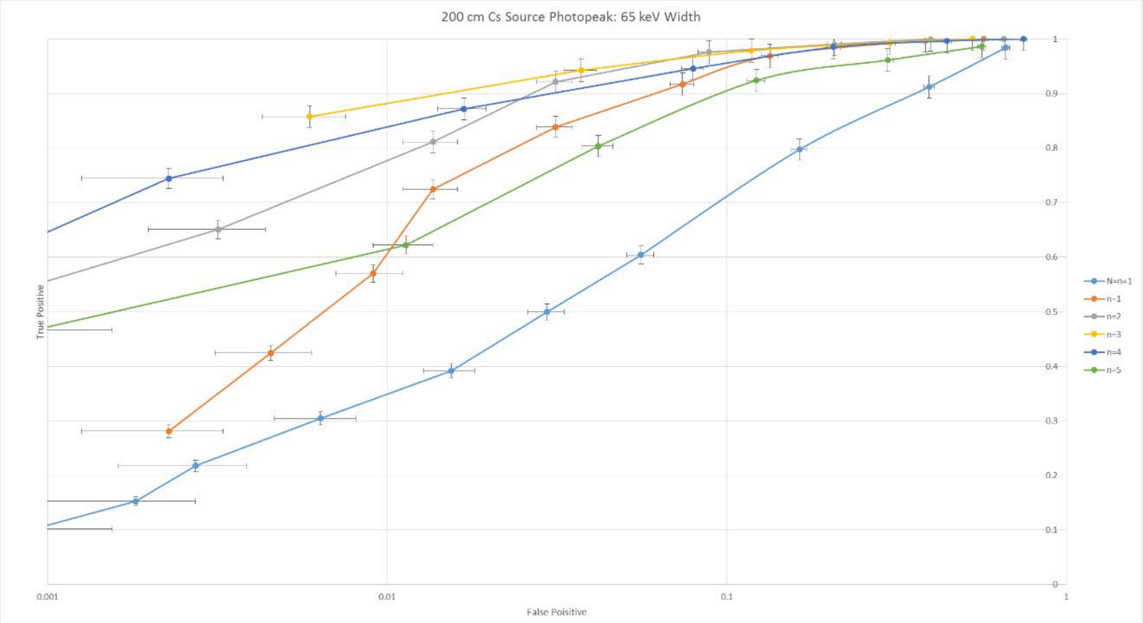


Figure G.13: 200 cm ^{137}Cs Source Photopeak Region – 65 keV Width

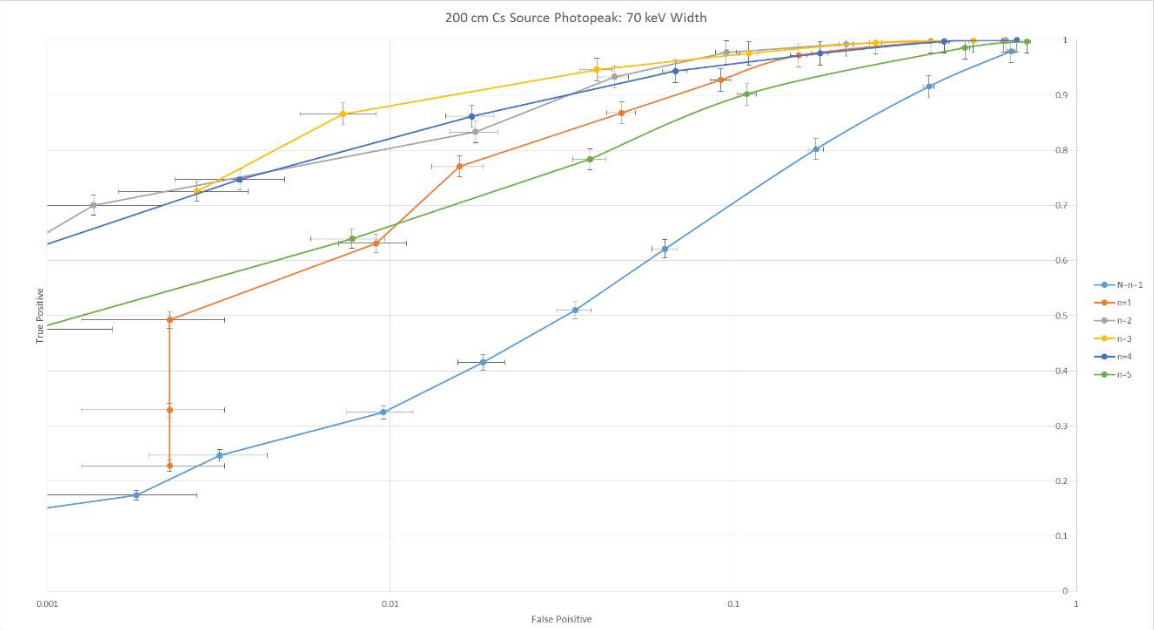


Figure G.14: 200 cm ^{137}Cs Source Photopeak Region – 70 keV Width

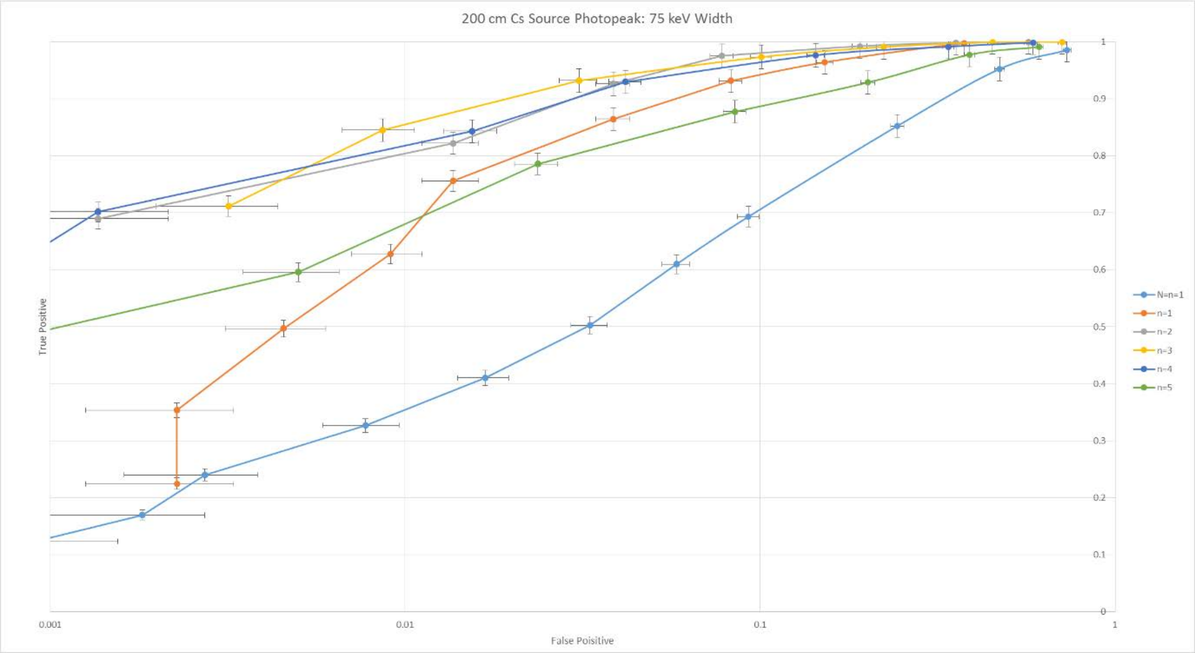


Figure G.15: 200 cm ^{137}Cs Source Photopeak Region – 75 keV Width

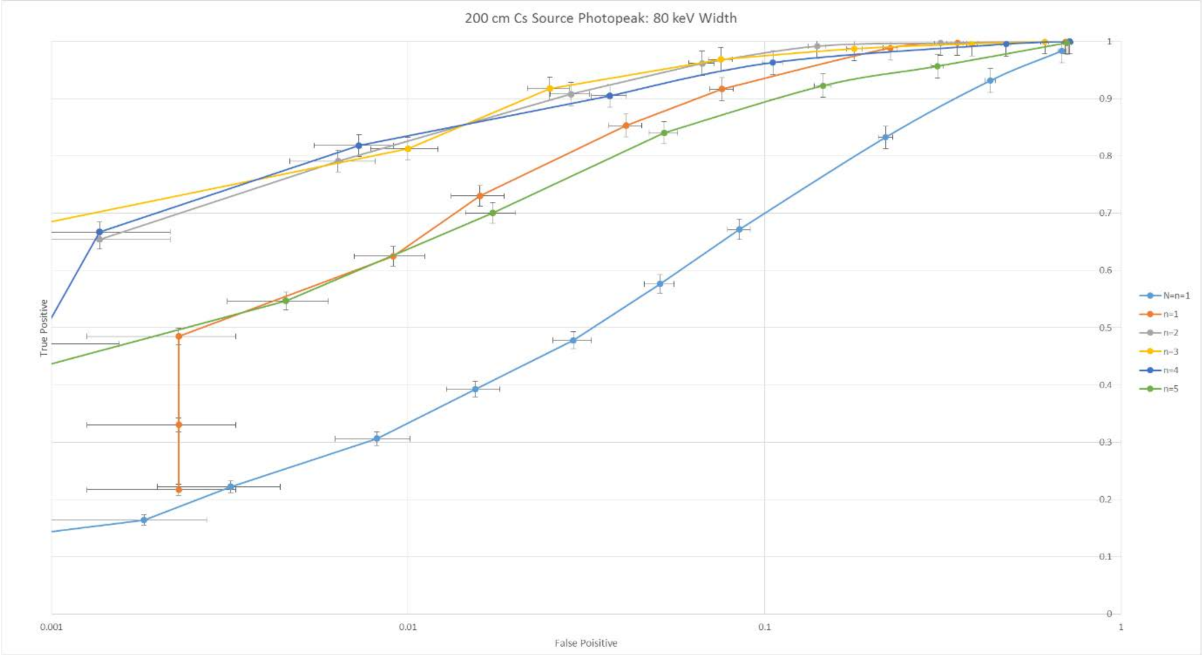


Figure G.16: 200 cm ^{137}Cs Source Photopeak Region – 80 keV Width

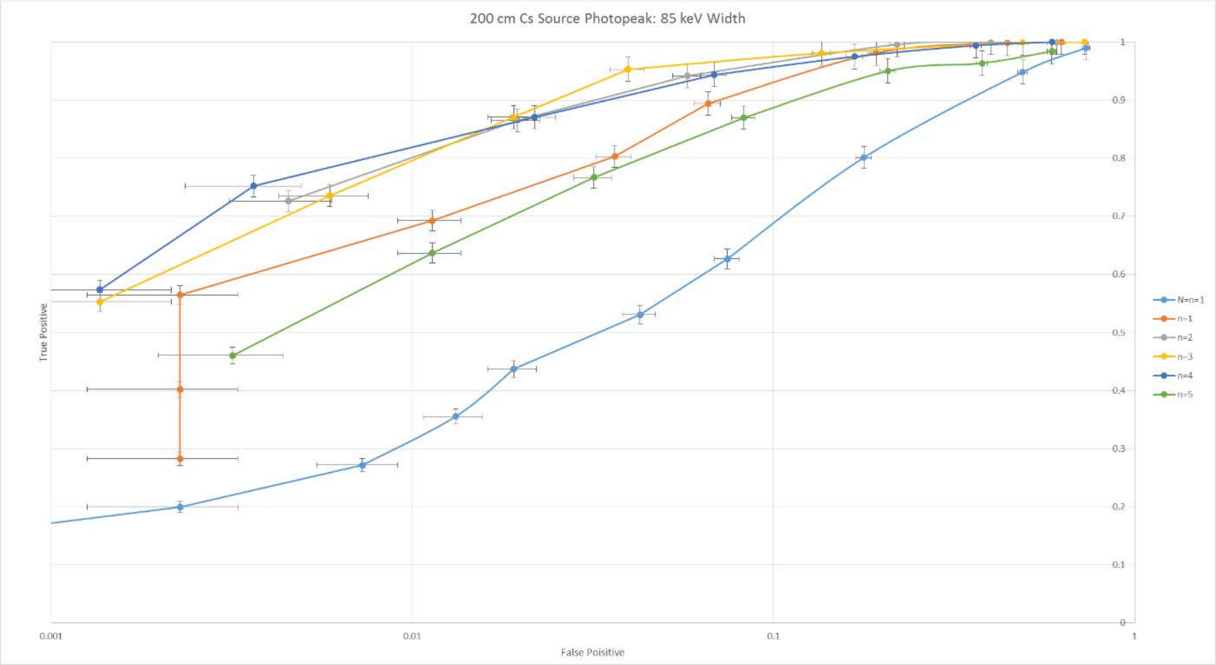


Figure G.17: 200 cm ^{137}Cs Source Photopeak Region – 85 keV Width

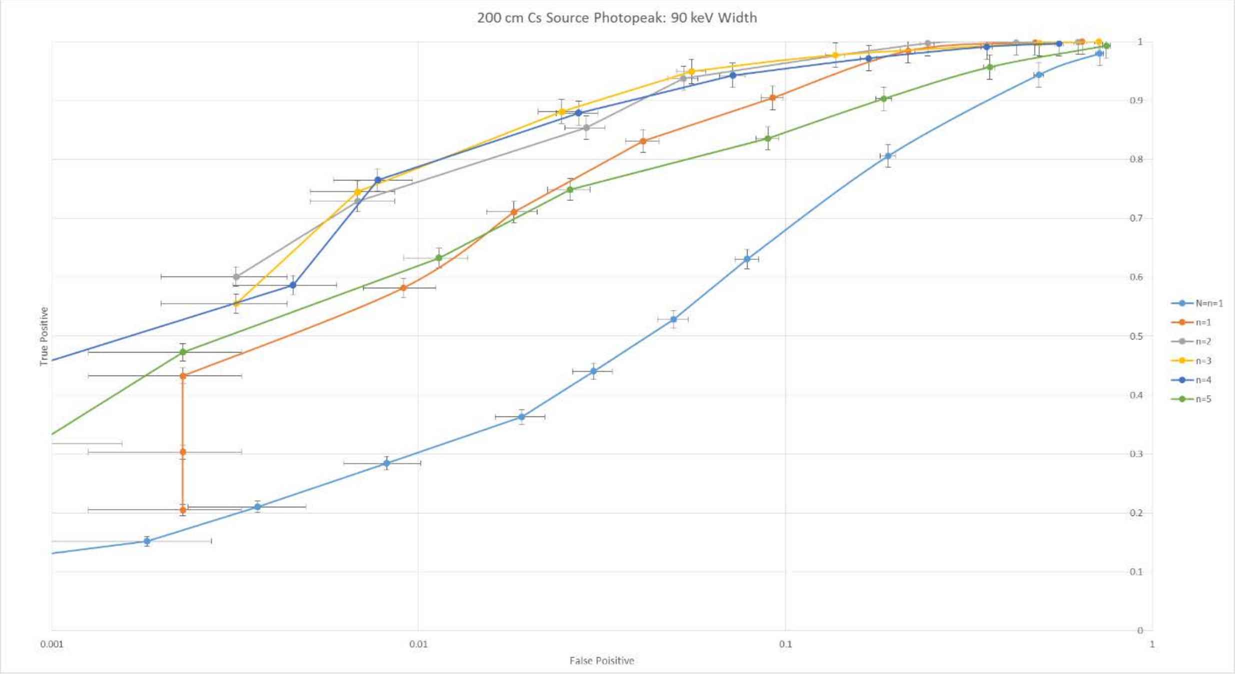


Figure G.18: 200 cm ^{137}Cs Source Photopeak Region – 90 keV Width

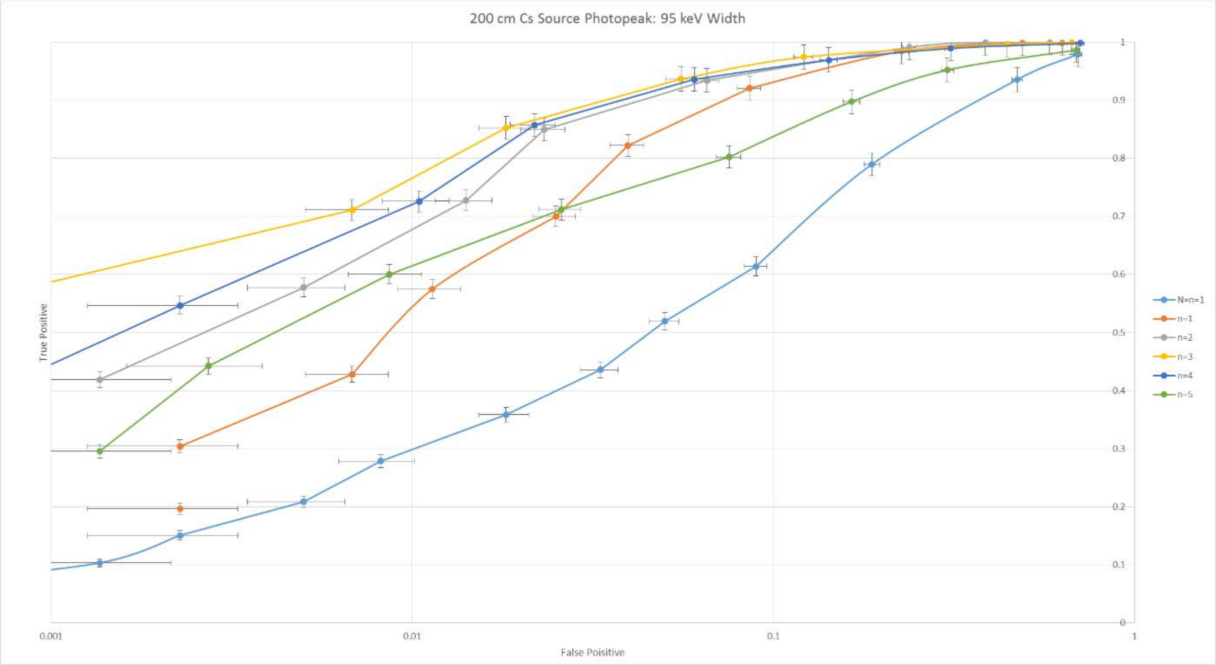


Figure G.19: 200 cm ^{137}Cs Source Photopeak Region – 95 keV Width

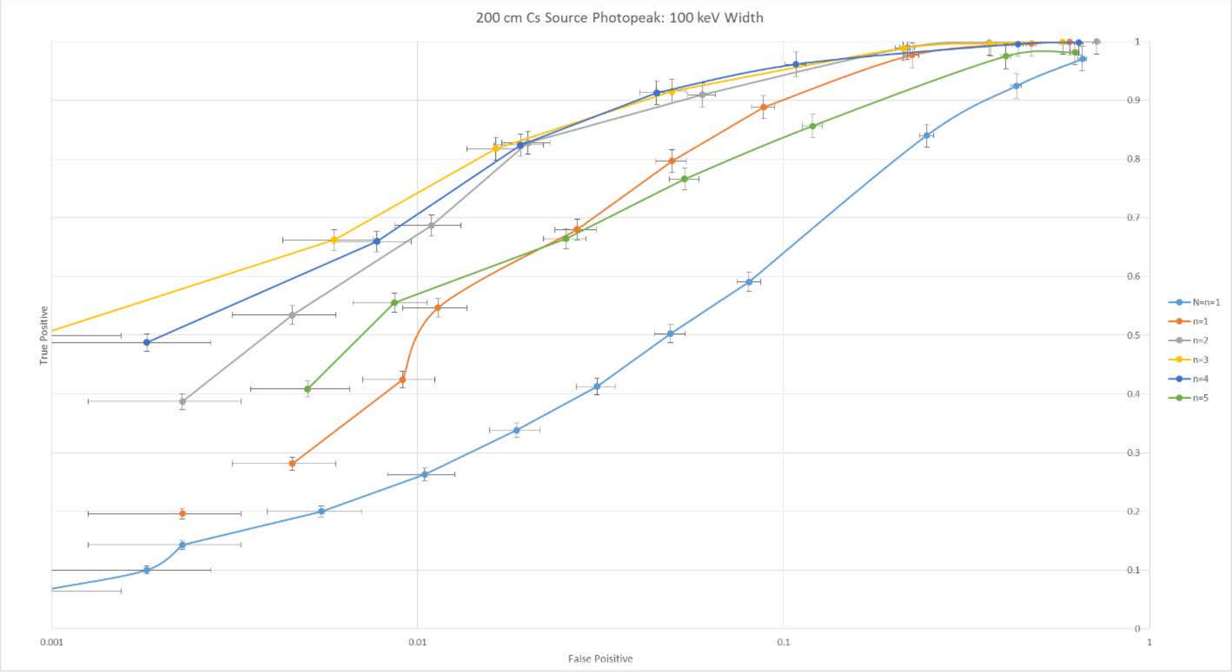


Figure G.20: 200 cm ^{137}Cs Source Photopeak Region – 100 keV Width

Appendix H

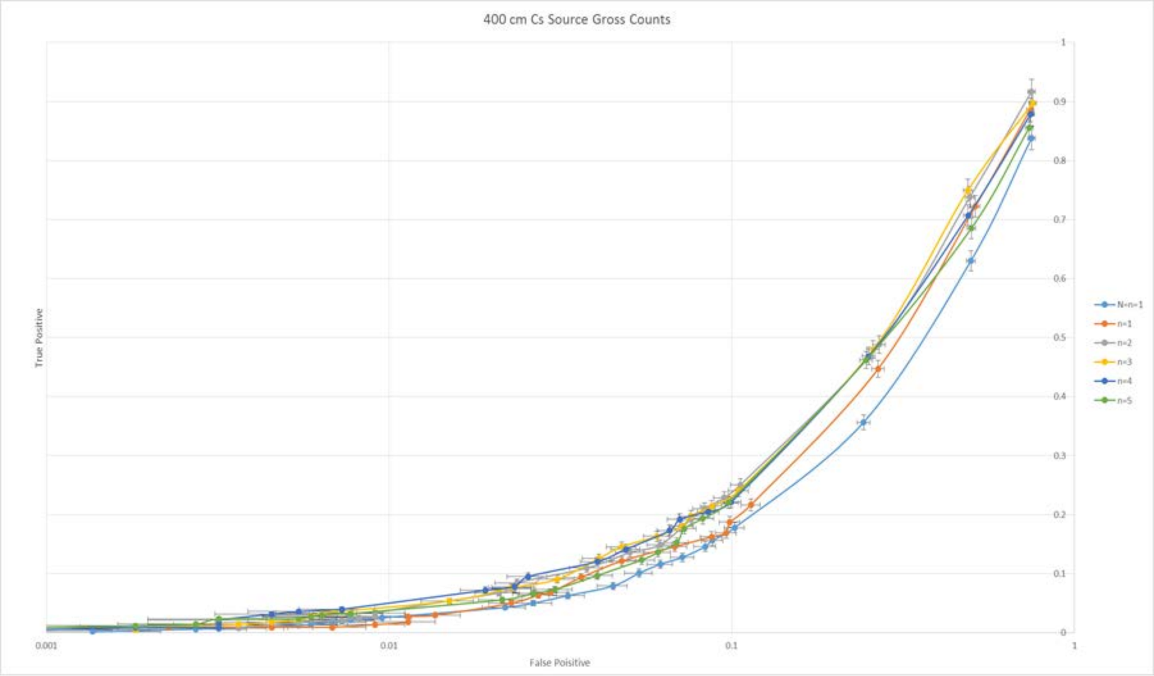


Figure H.1: 400 cm ^{137}Cs Source Gross Counts: Logarithmic

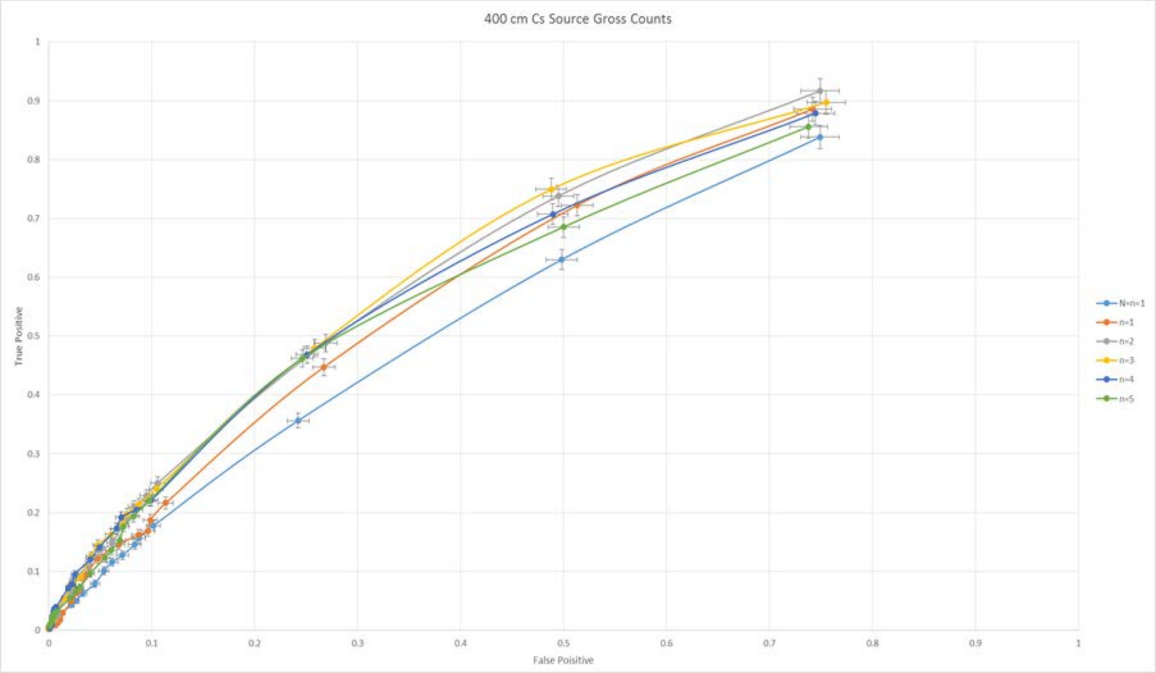


Figure H.2: 400 cm ^{137}Cs Source Gross Counts: Linear 0 - 1

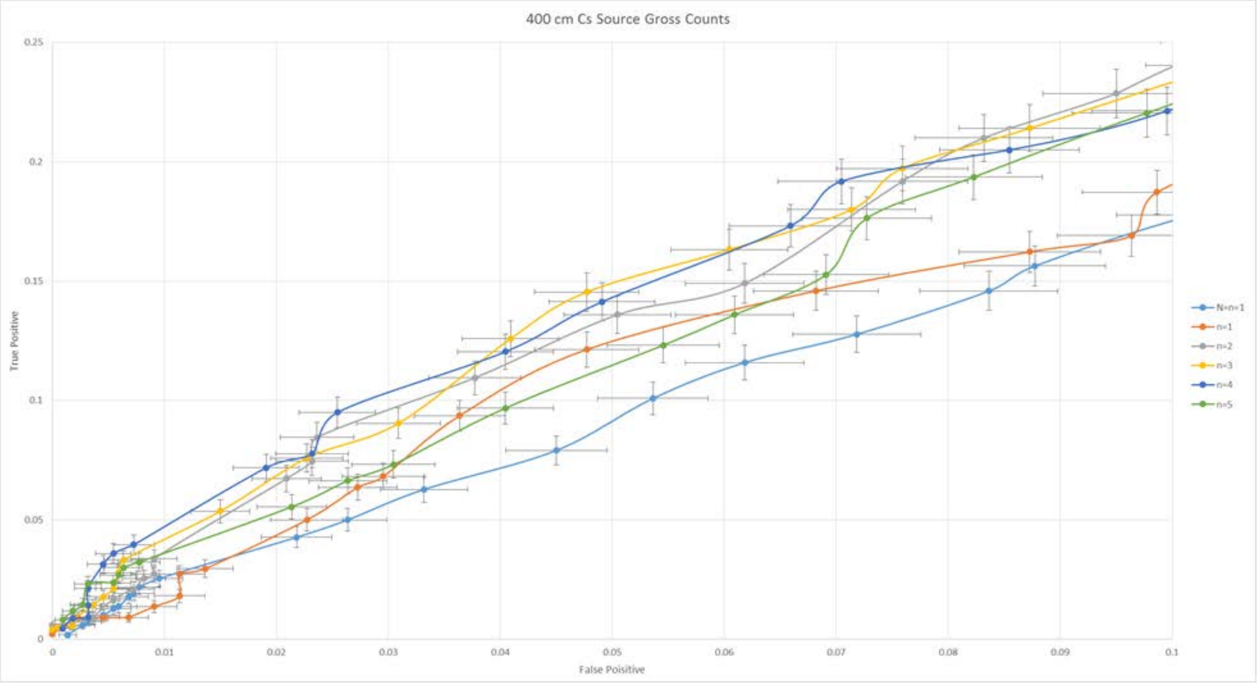


Figure H.3: 400 cm ^{137}Cs Source Gross Counts: Linear 0 – 0.1

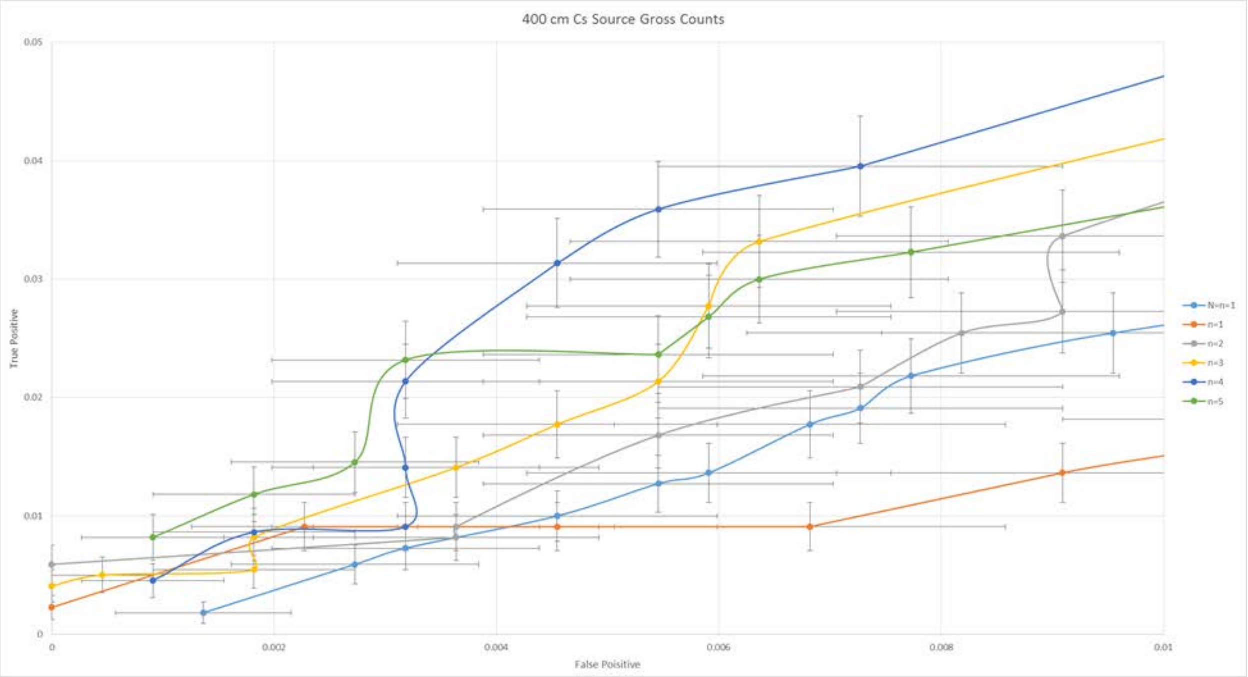


Figure H.4: 400 cm ^{137}Cs Source Gross Counts: Linear 0 – 0.01

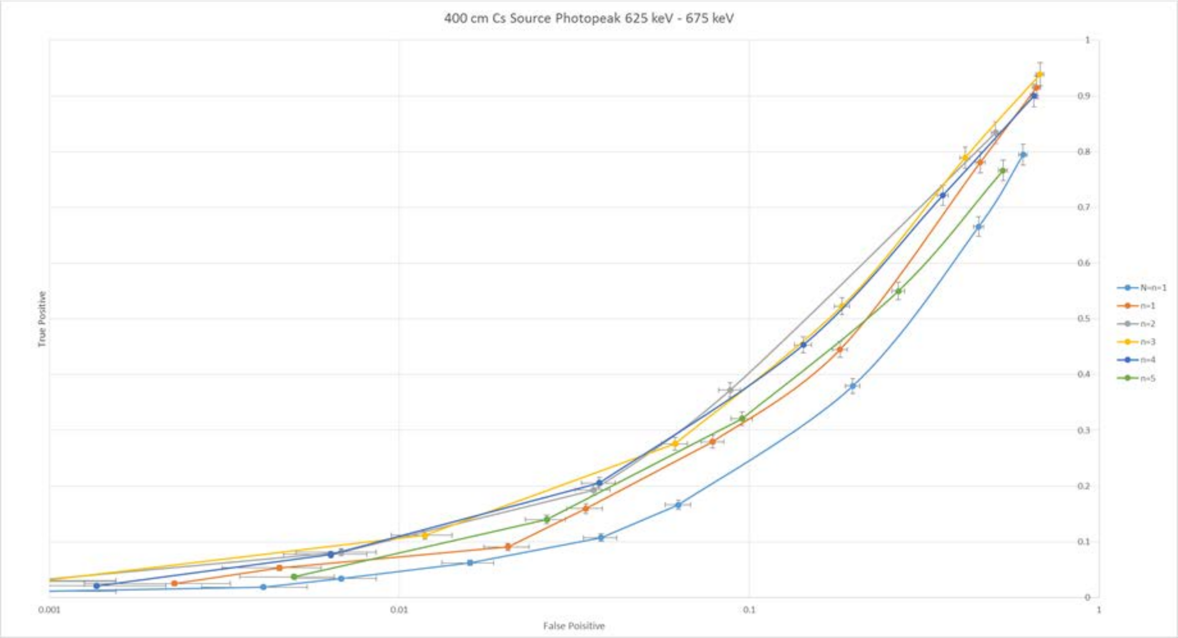


Figure H.5: 400 cm ^{137}Cs Source Photopeak Region 625 keV – 675 keV: Logarithmic

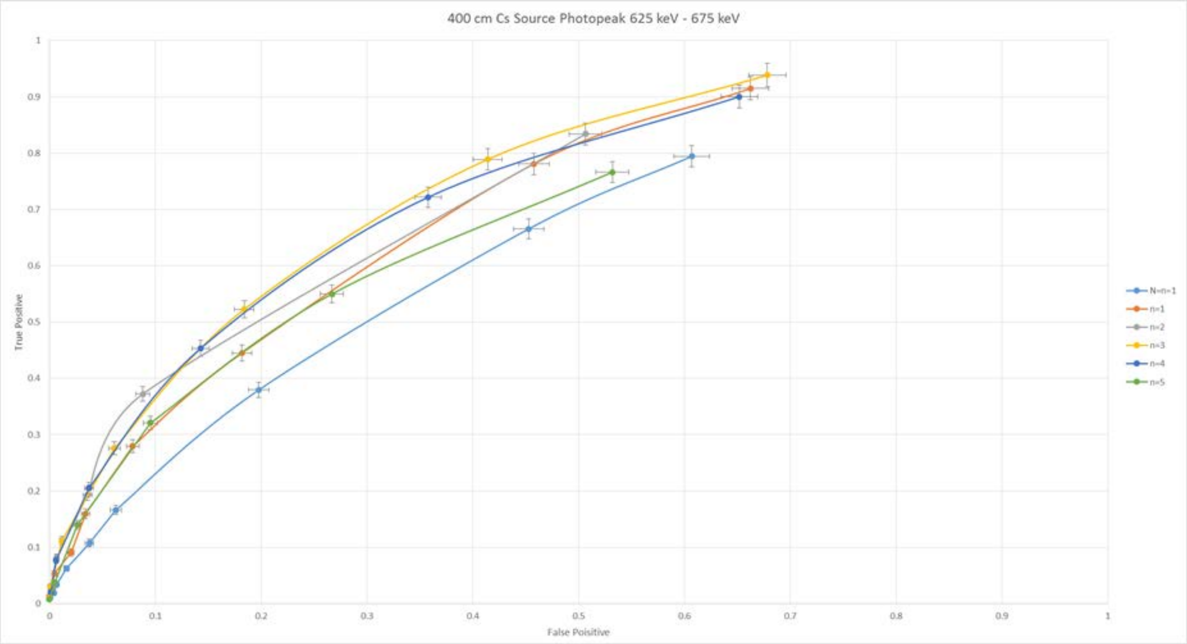


Figure H.6: 400 cm ^{137}Cs Source Photopeak Region 625 keV – 675 keV: Linear 0 - 1

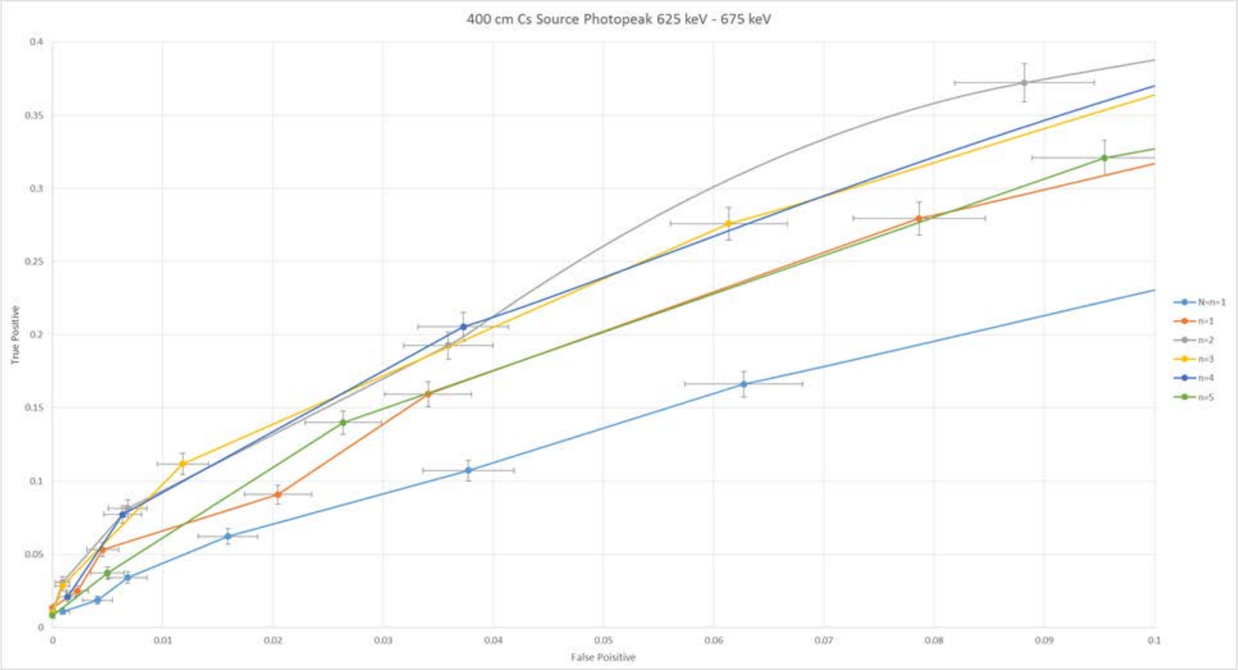


Figure H.7: 400 cm ^{137}Cs Source Photopeak Region 625 keV – 675 keV: Linear 0 – 0.1

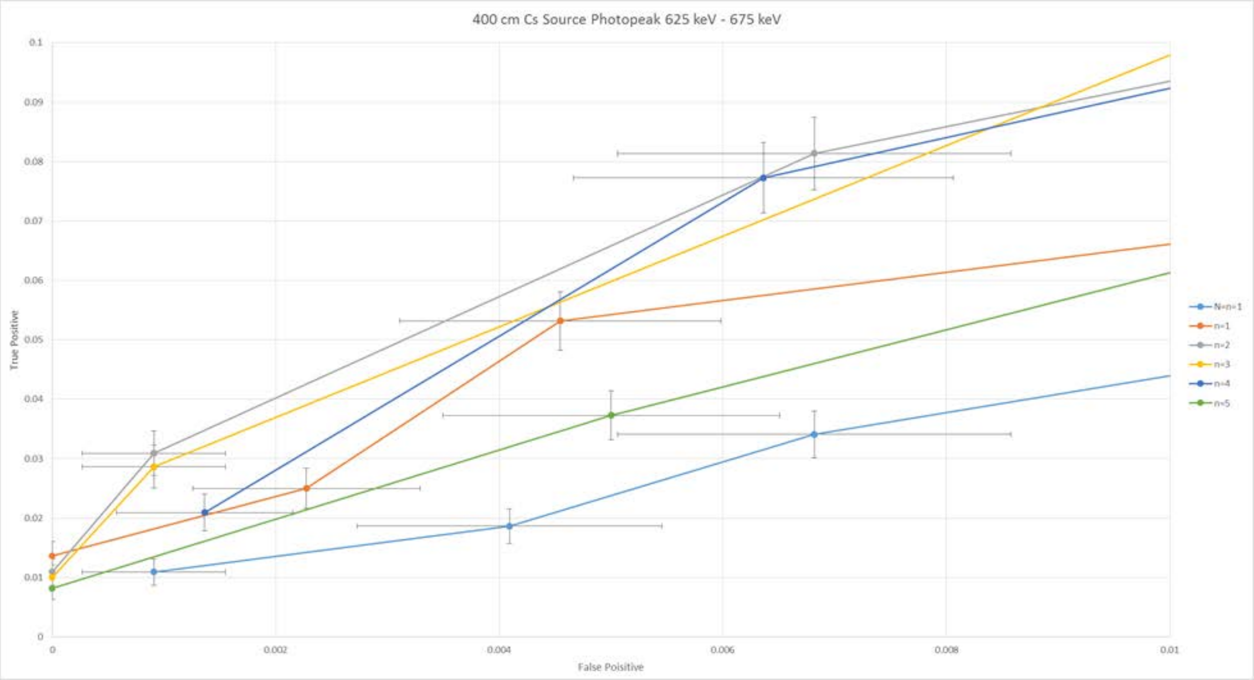


Figure H.8: 400 cm ^{137}Cs Source Photopeak Region 625 keV – 675 keV: Linear 0 – 0.01

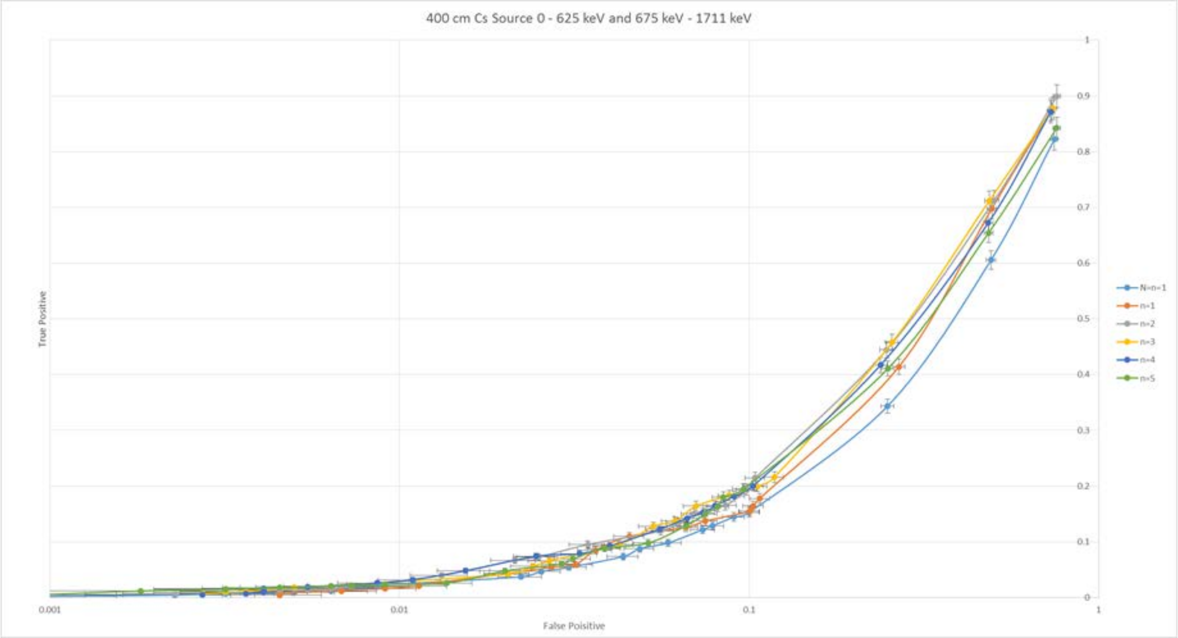


Figure H.9: 400 cm ^{137}Cs Source 0 keV – 625 keV and 675 keV – 1711 keV: Logarithmic

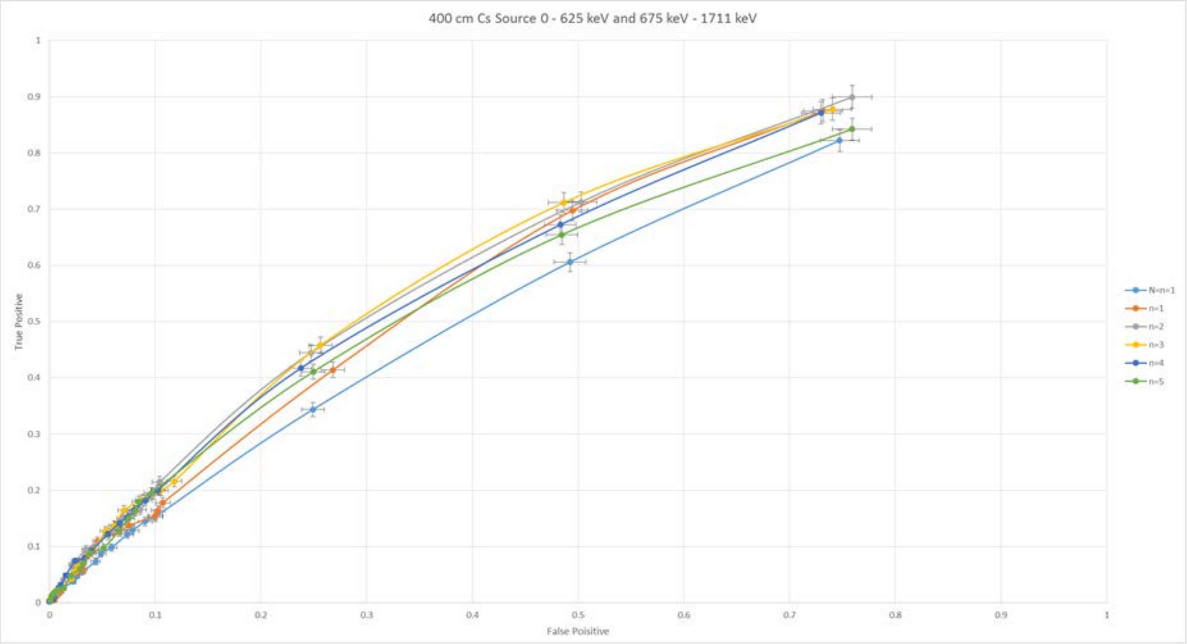


Figure H.10: 400 cm ^{137}Cs Source 0 keV – 625 keV and 675 keV – 1711 keV: Linear 0 - 1

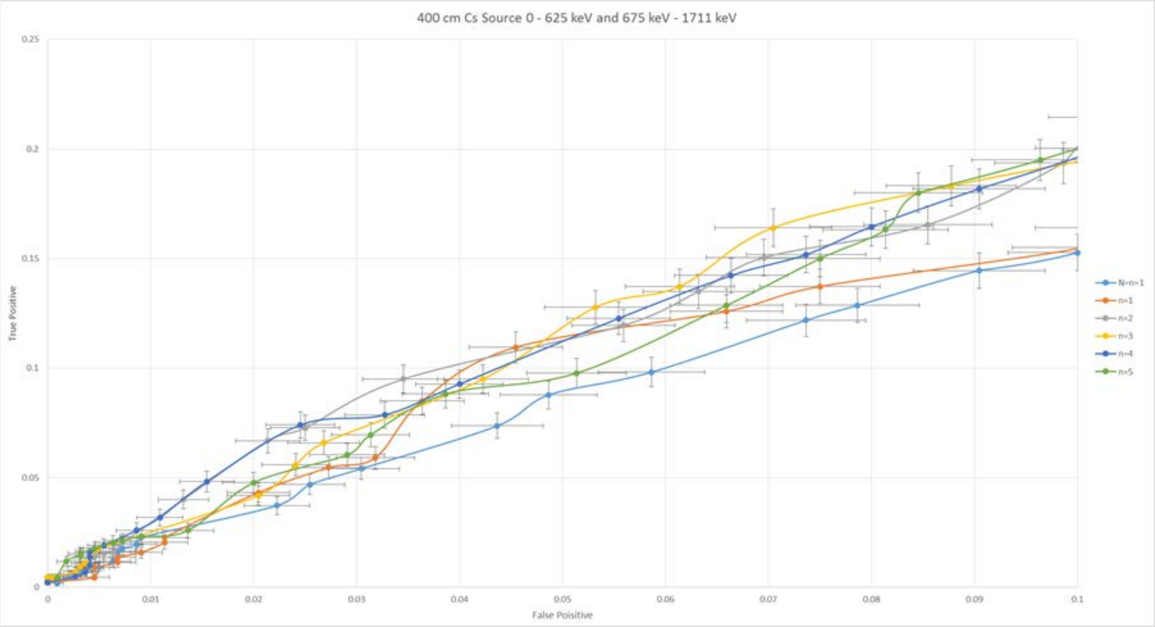


Figure H.11: 400 cm ^{137}Cs Source 0 keV – 625 keV and 675 keV – 1711 keV: Linear 0 – 0.1

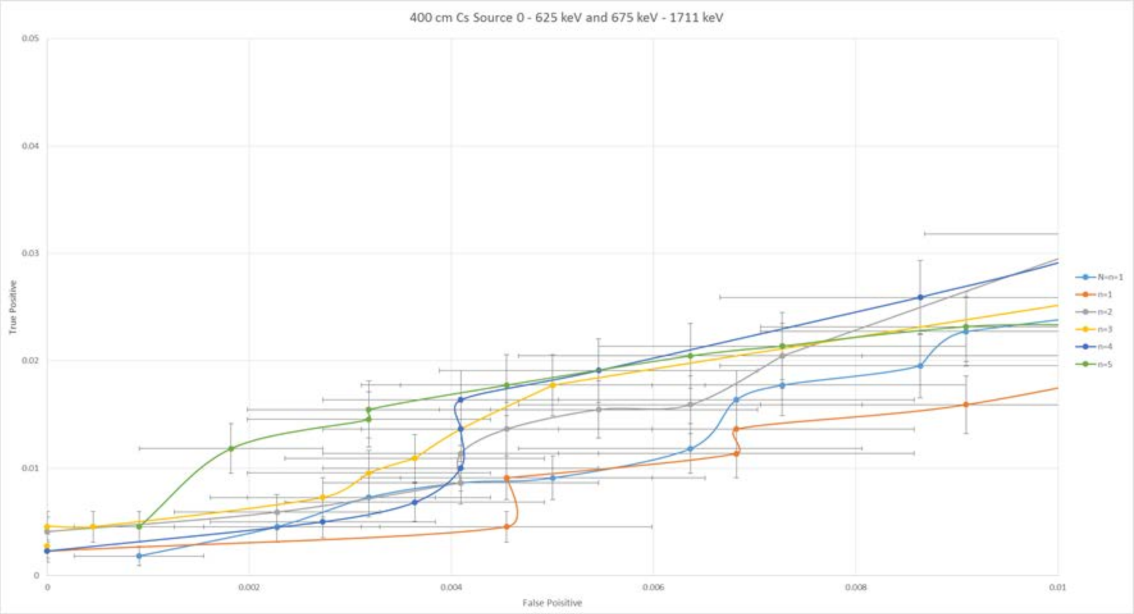


Figure H.12: 400 cm ^{137}Cs Source 0 keV – 625 keV and 675 keV – 1711 keV: Linear 0 – 0.01

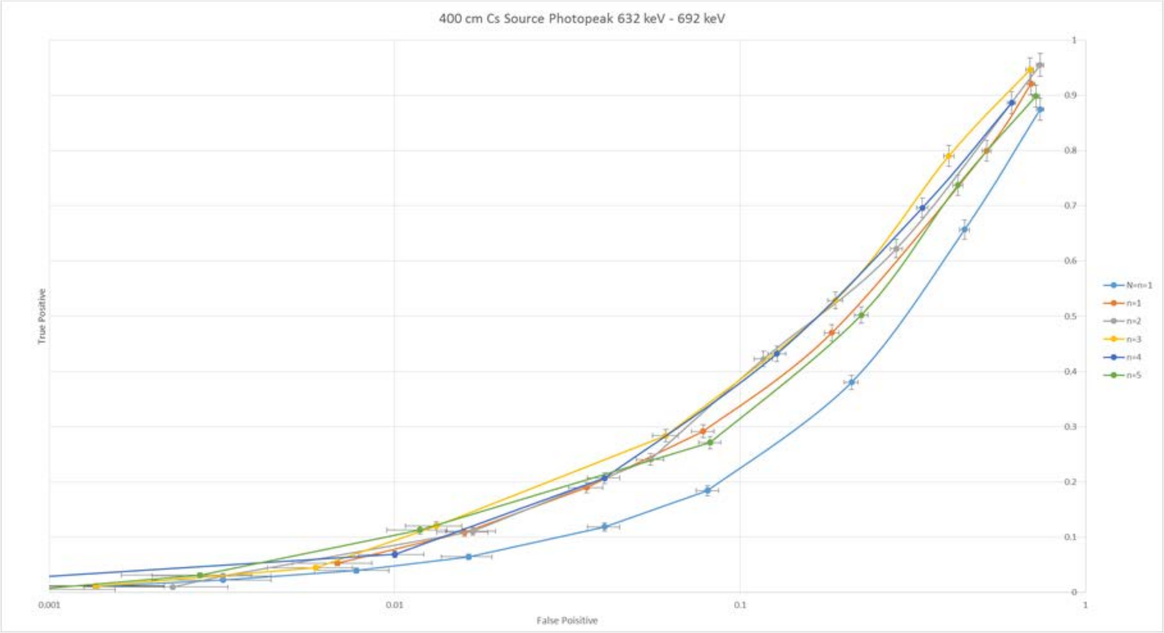


Figure H.13: 400 cm ^{137}Cs Source Photopeak Region 632 keV – 692 keV: Logarithmic

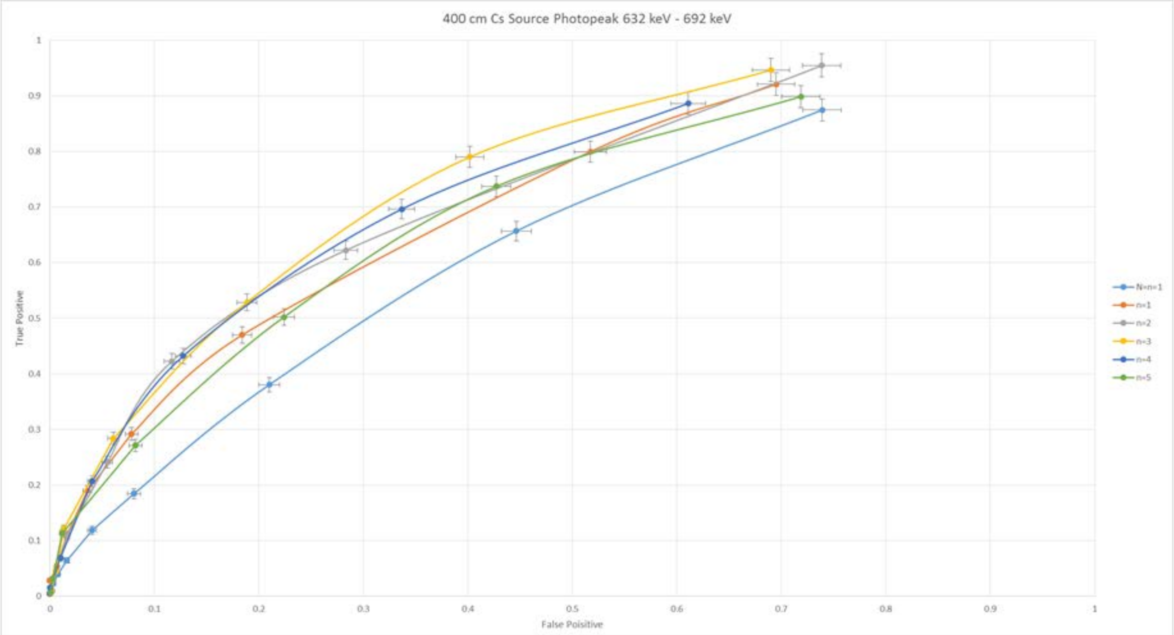


Figure H.14: 400 cm ^{137}Cs Source Photopeak Region 632 keV – 692 keV: Linear 0 - 1

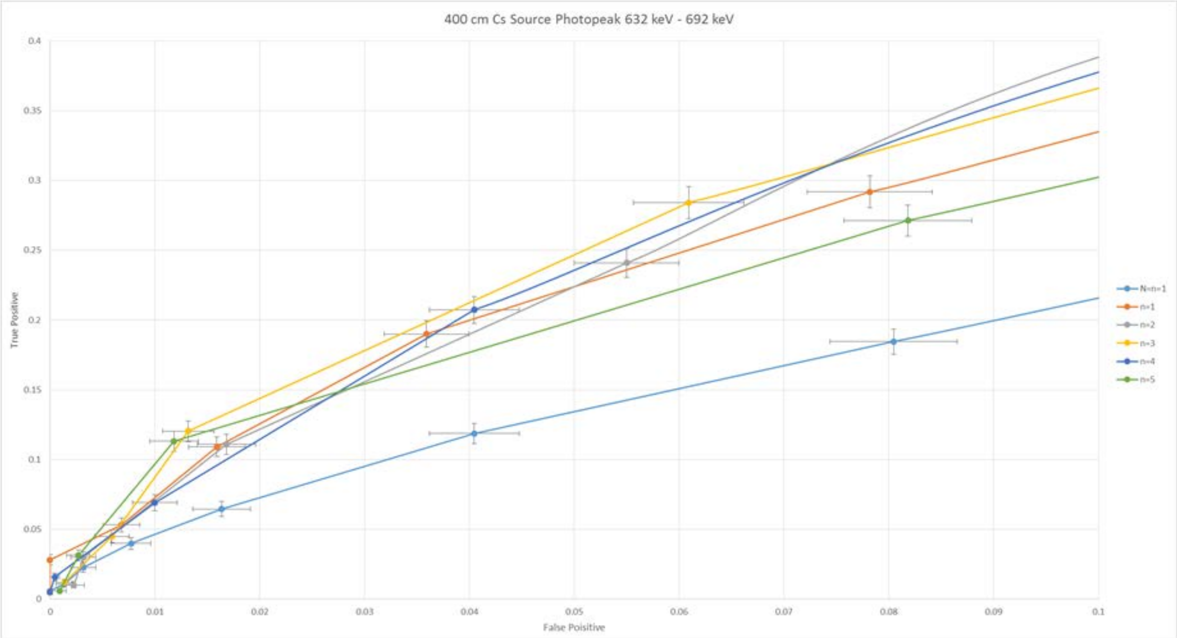


Figure H.15: 400 cm ^{137}Cs Source Photopeak Region 632 keV – 692 keV: Linear 0 – 0.1

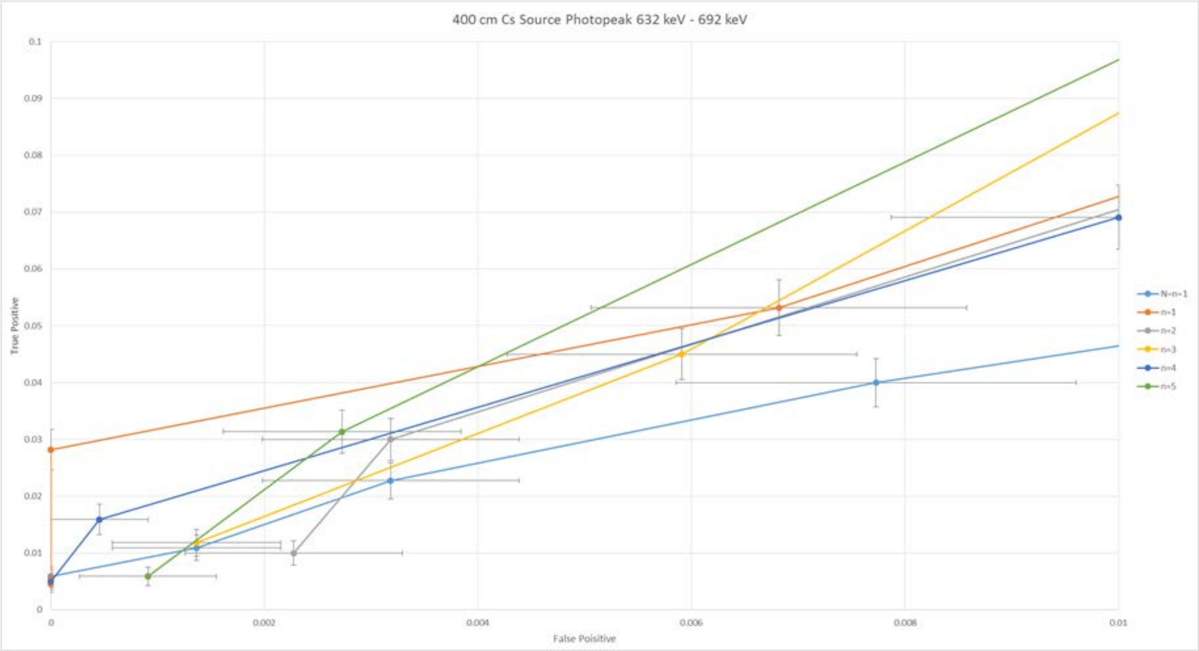


Figure H.16: 400 cm ^{137}Cs Source Photopeak Region 632 keV – 692 keV: Linear 0 – 0.01

Appendix I

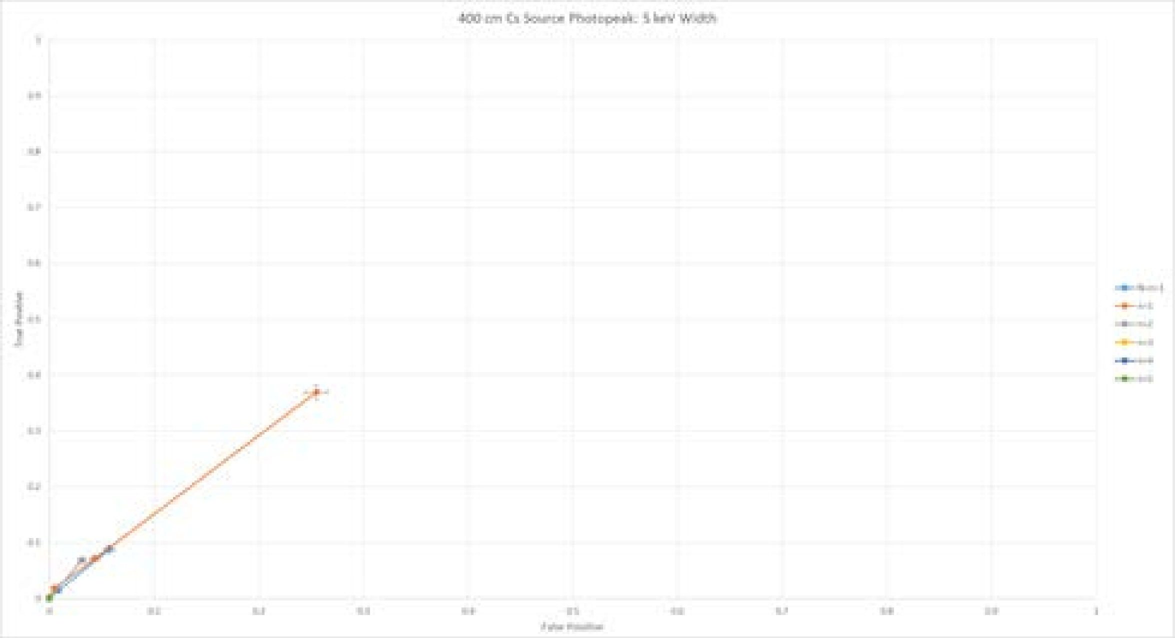


Figure I.1: 400 cm ^{137}Cs Source Photopeak Region: 5 keV Width

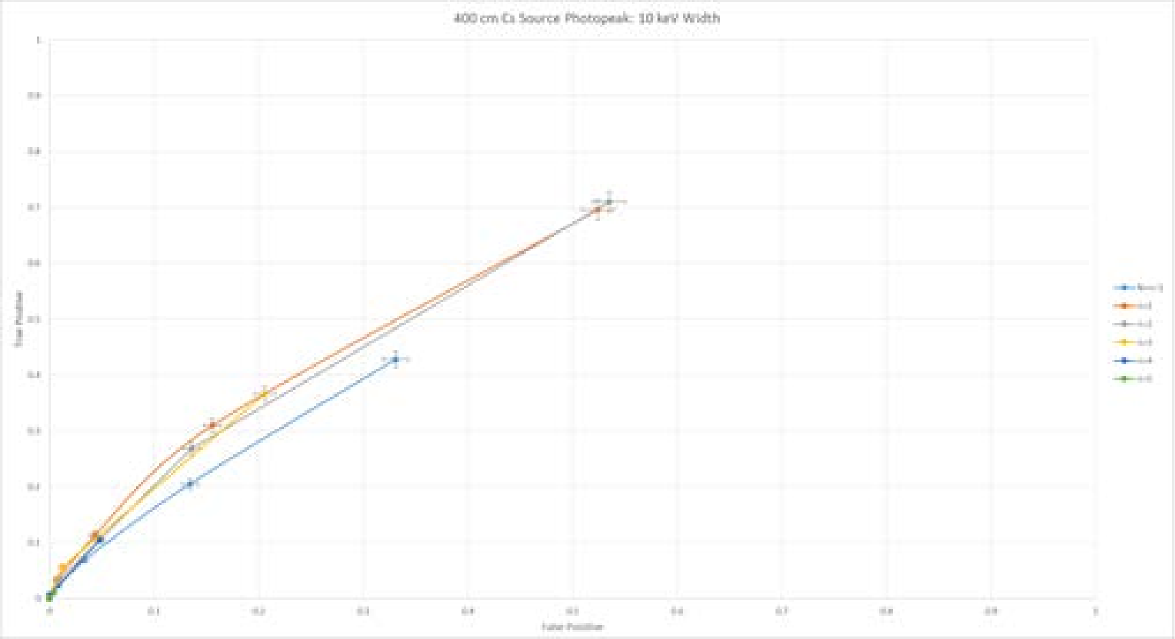


Figure I.2: 400 cm ^{137}Cs Source Photopeak Region: 10 keV Width

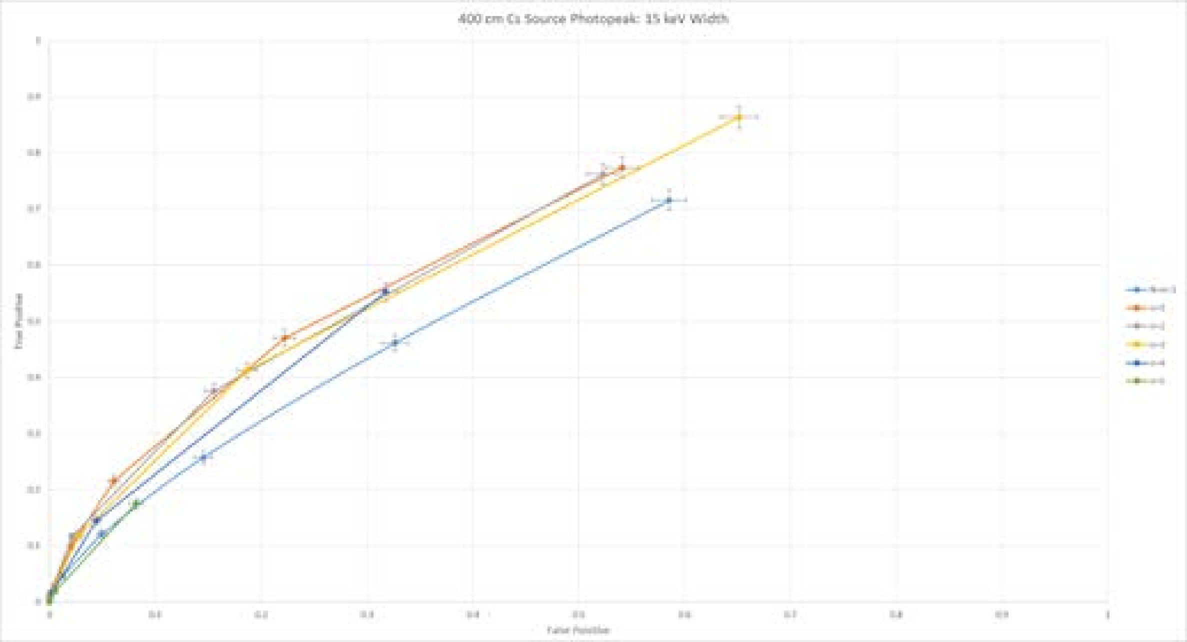


Figure I.3: 400 cm ^{137}Cs Source Photopeak Region: 15 keV Width

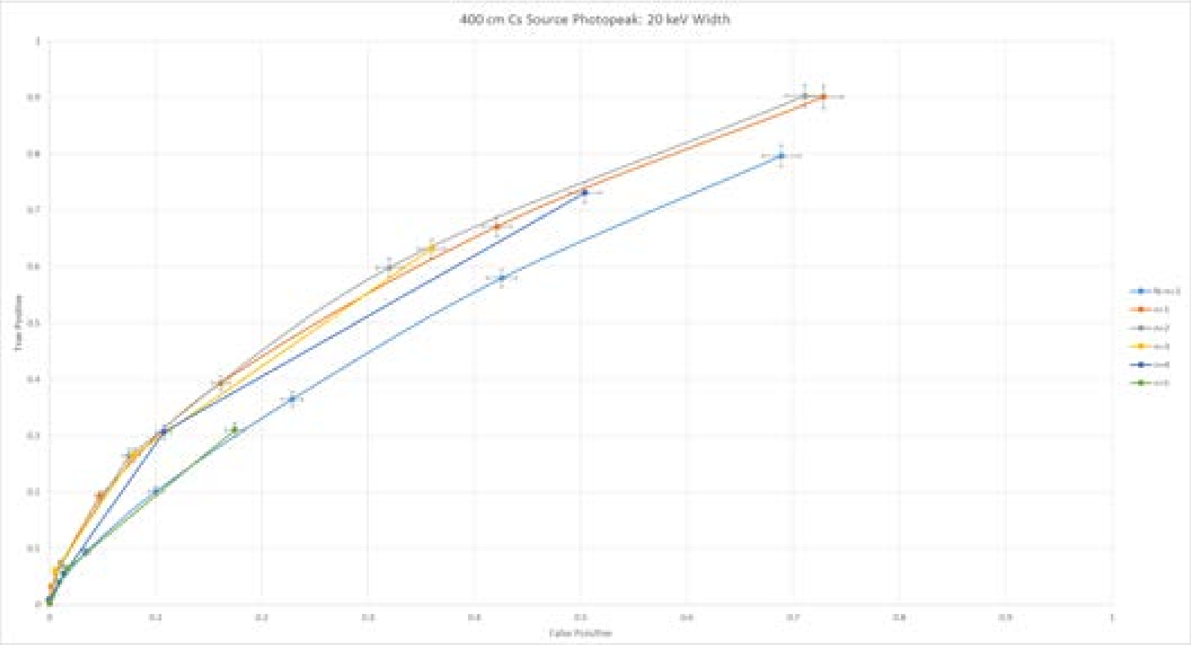


Figure I.4: 400 cm ^{137}Cs Source Photopeak Region: 20 keV Width

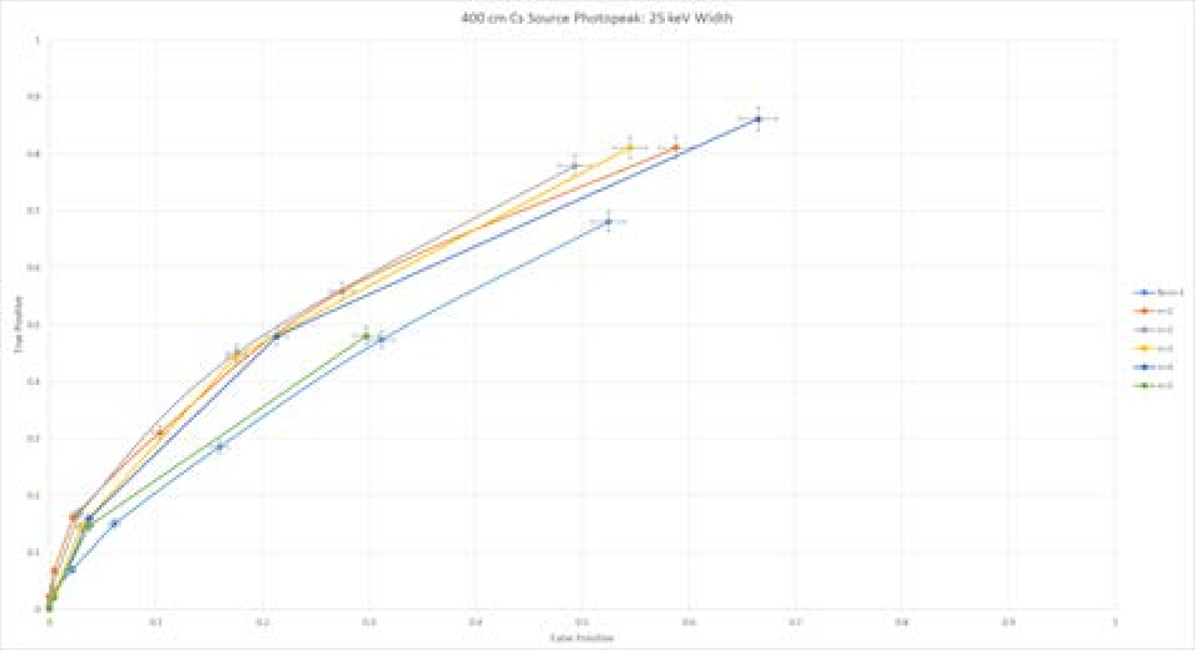


Figure I.5: 400 cm ^{137}Cs Source Photopeak Region: 25 keV Width

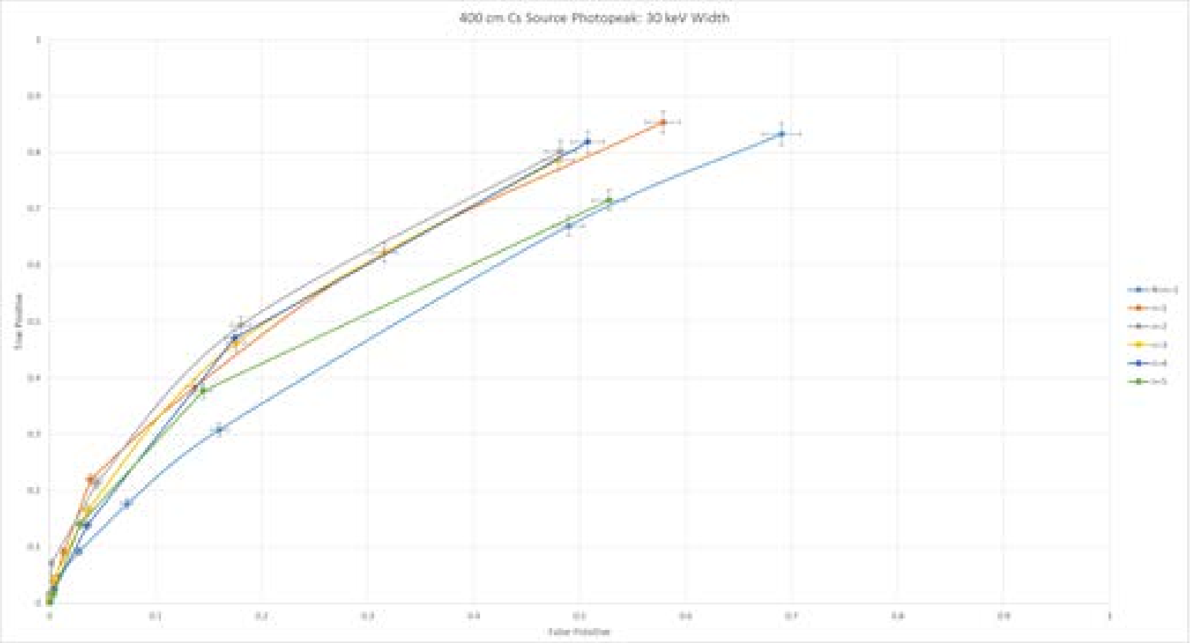


Figure I.6: 400 cm ^{137}Cs Source Photopeak Region: 30 keV Width

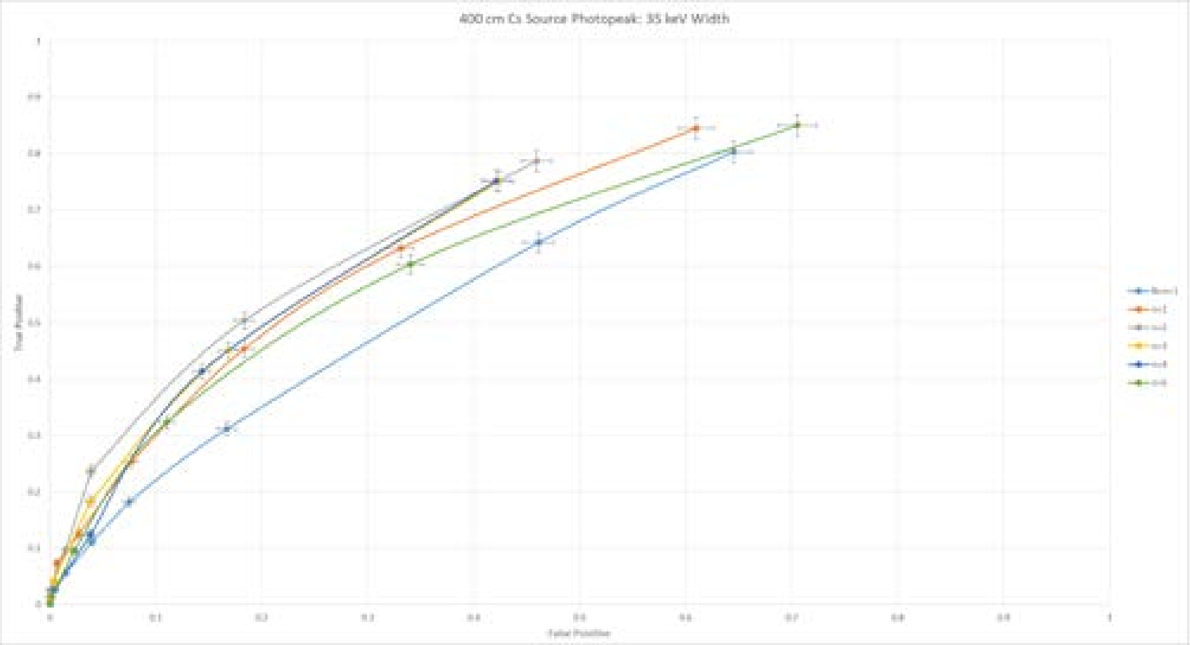


Figure I.7: 400 cm ^{137}Cs Source Photopeak Region: 35 keV Width

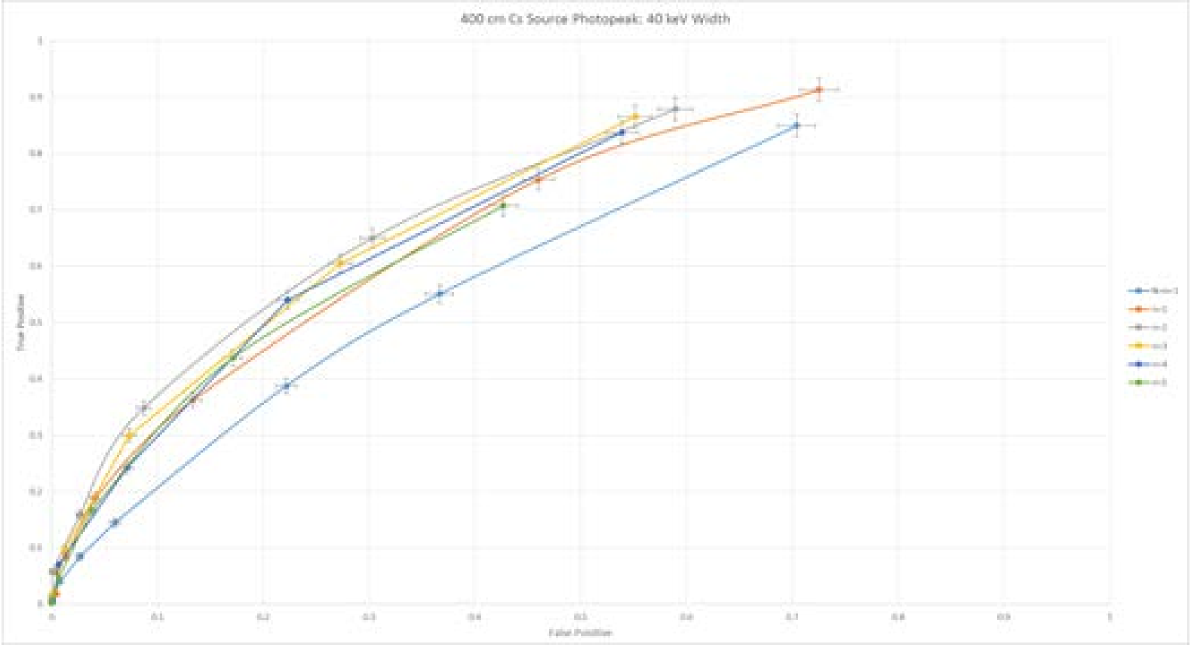


Figure I.8: 400 cm ^{137}Cs Source Photopeak Region: 40 keV Width

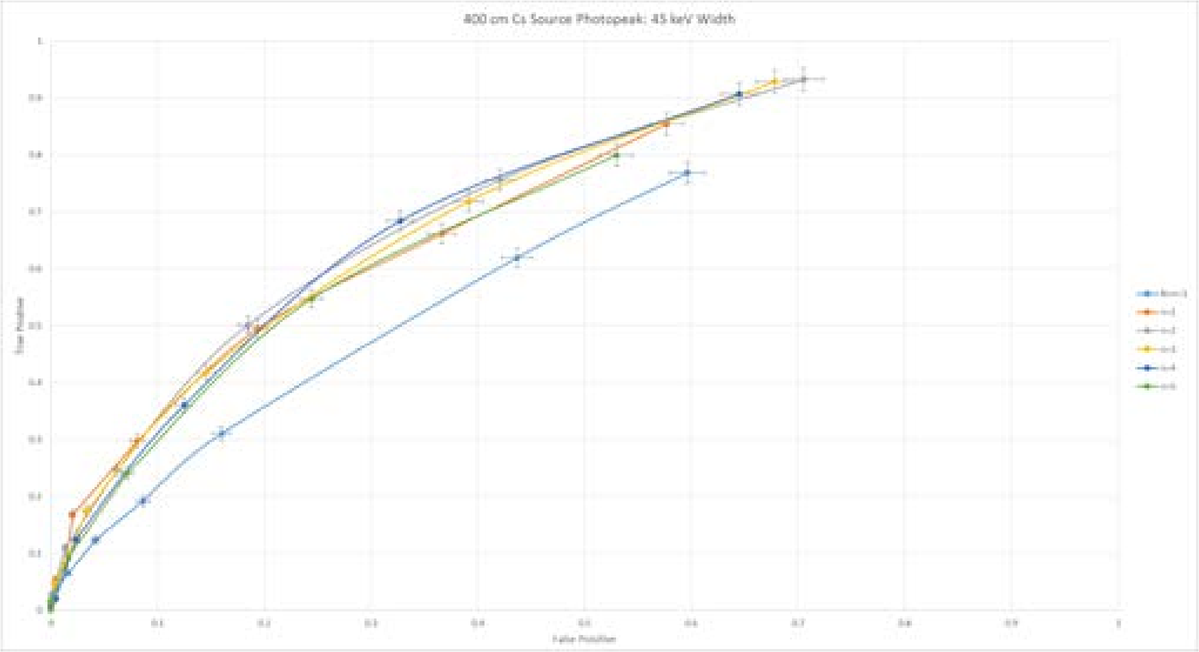


Figure I.9: 400 cm ^{137}Cs Source Photopeak Region: 45 keV Width

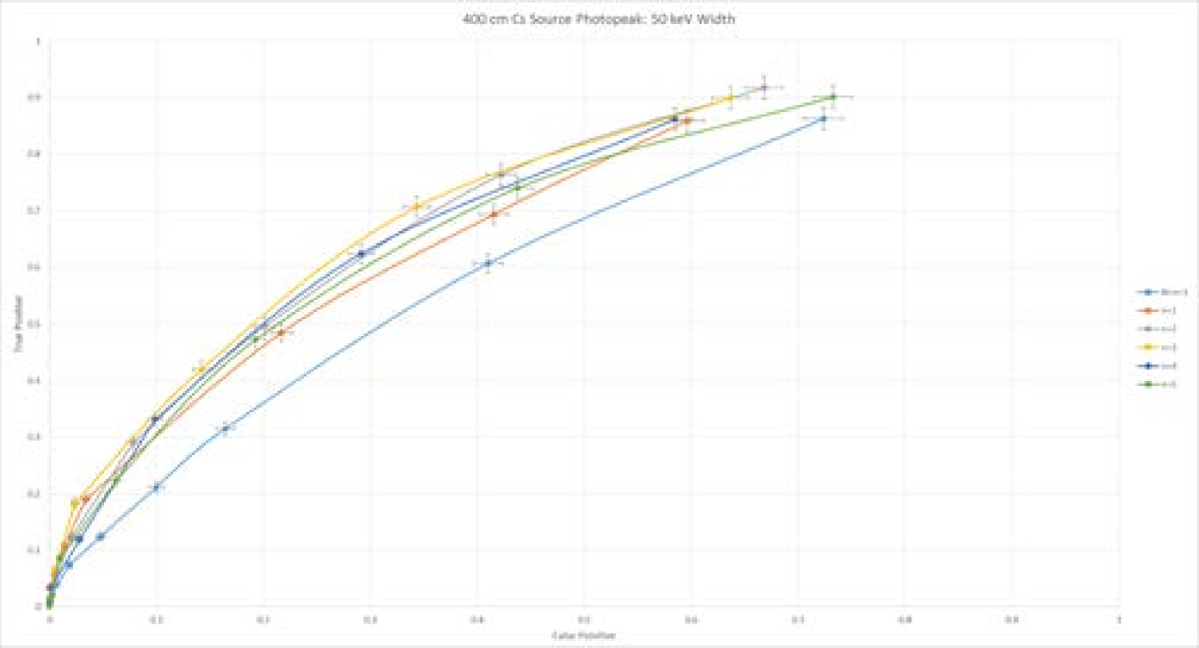


Figure I.10: 400 cm ^{137}Cs Source Photopeak Region: 50 keV Width

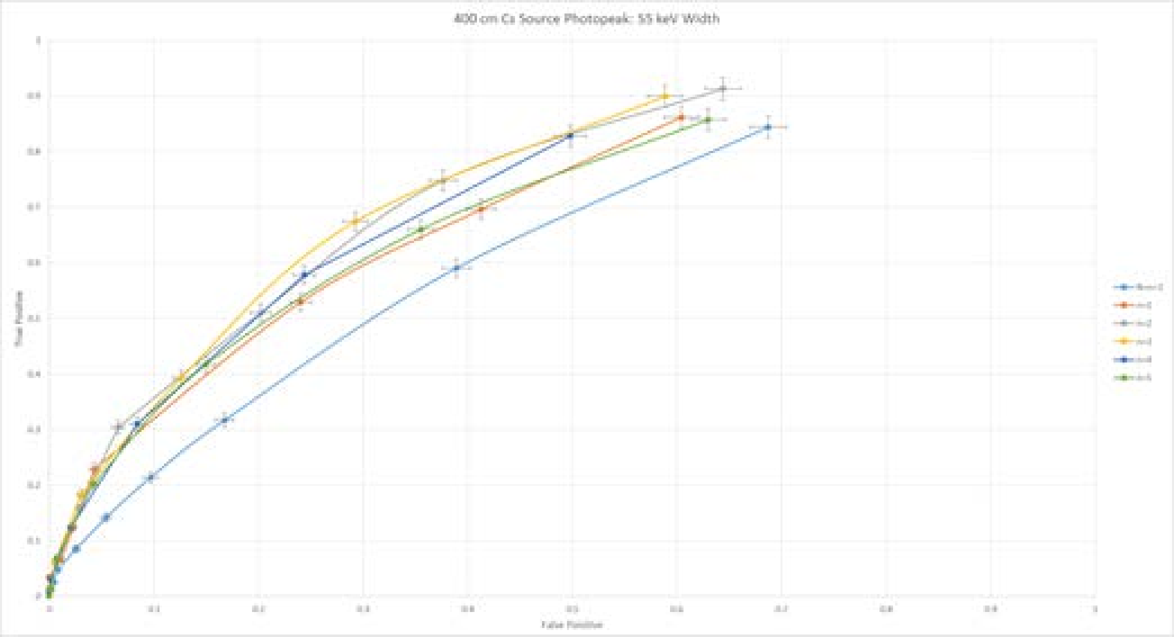


Figure I.11: 400 cm ^{137}Cs Source Photopeak Region: 55 keV Width

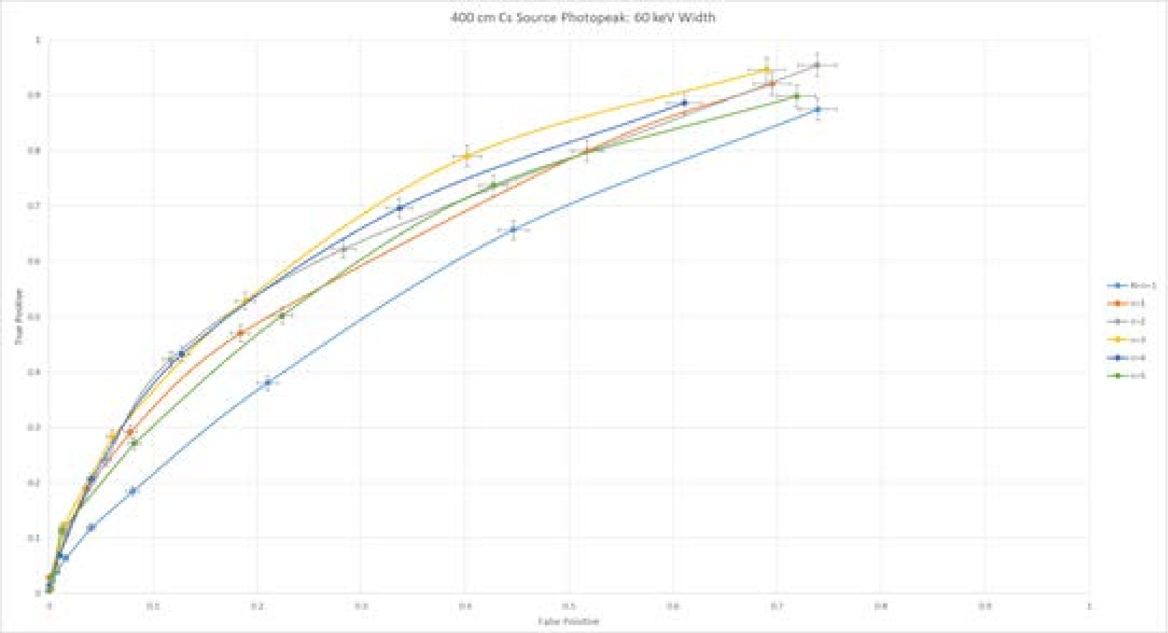


Figure I.12: 400 cm ^{137}Cs Source Photopeak Region: 60 keV Width

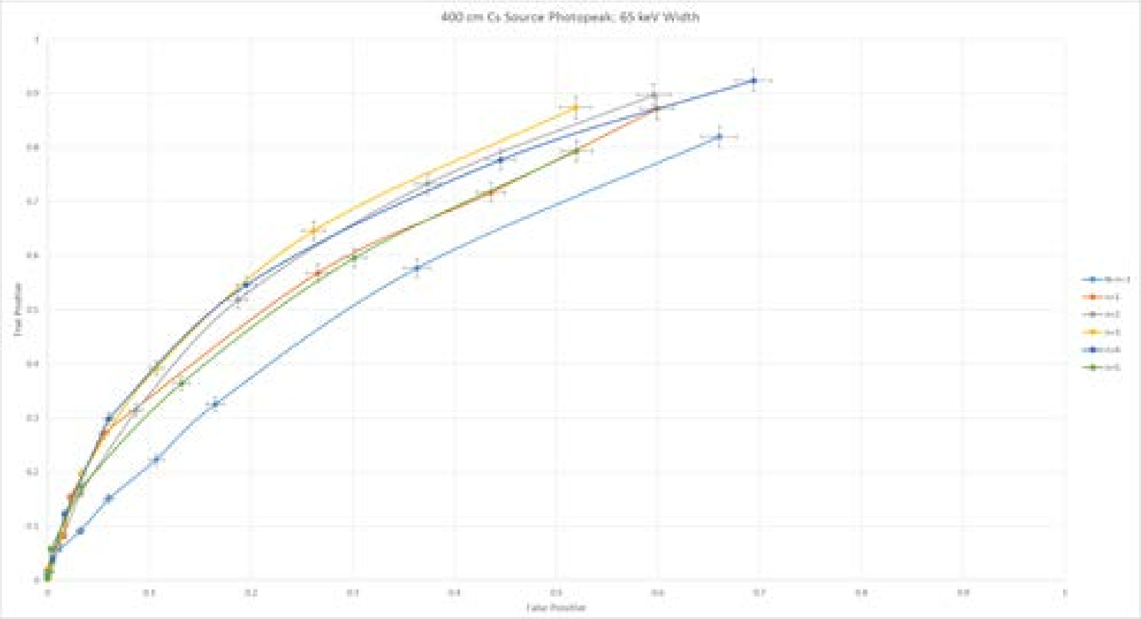


Figure I.13: 400 cm ^{137}Cs Source Photopeak Region: 65 keV Width

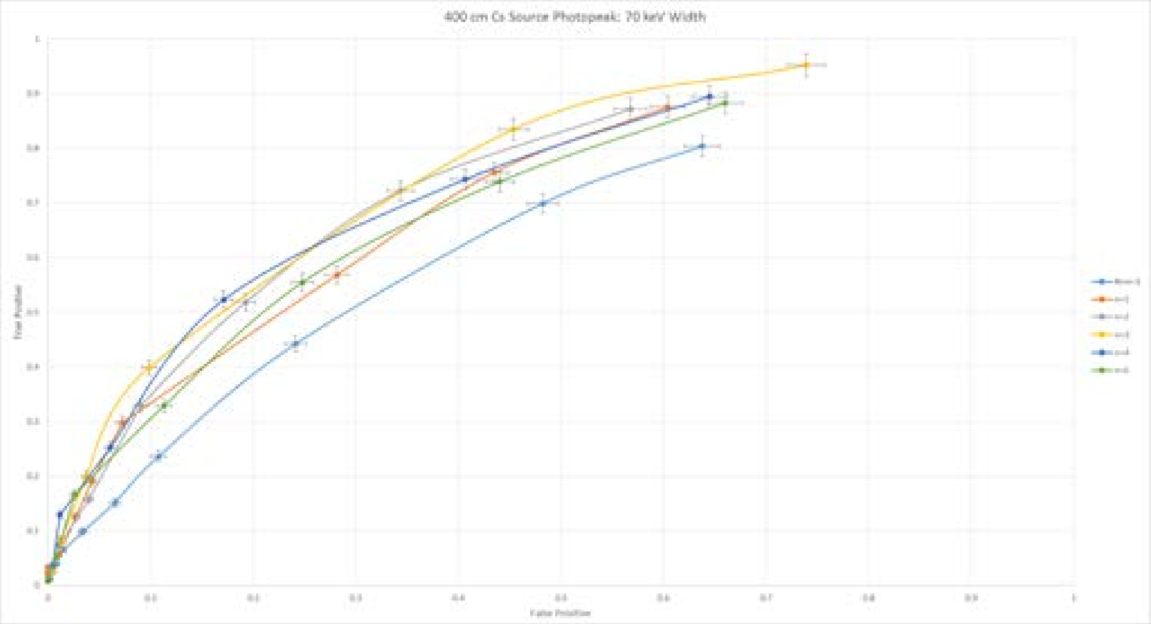


Figure I.14: 400 cm ^{137}Cs Source Photopeak Region: 70 keV Width

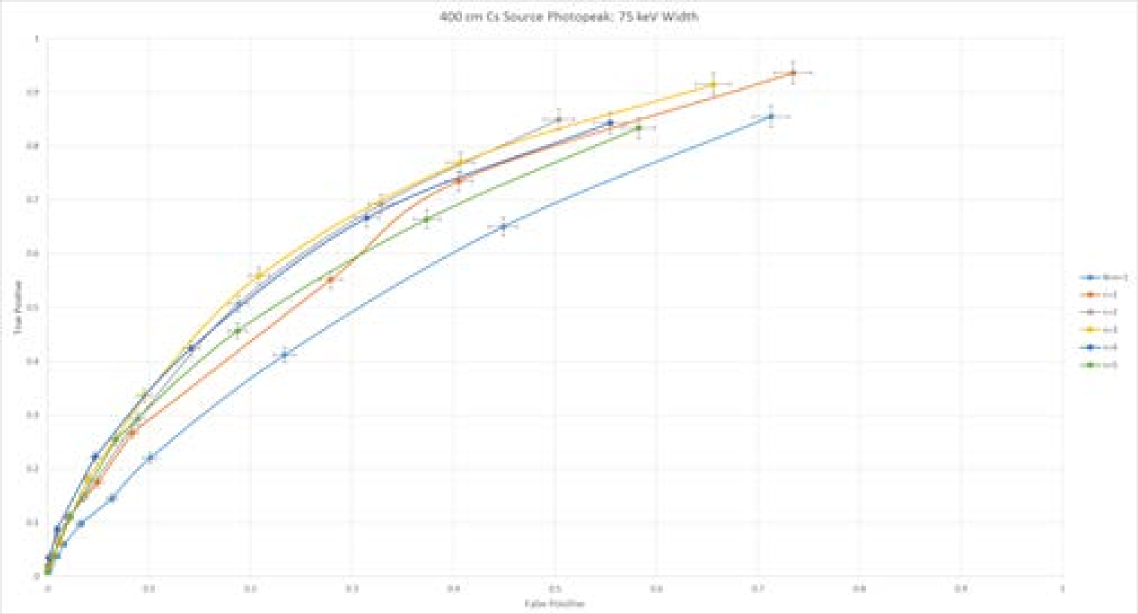


Figure I.15: 400 cm ^{137}Cs Source Photopeak Region: 75 keV Width

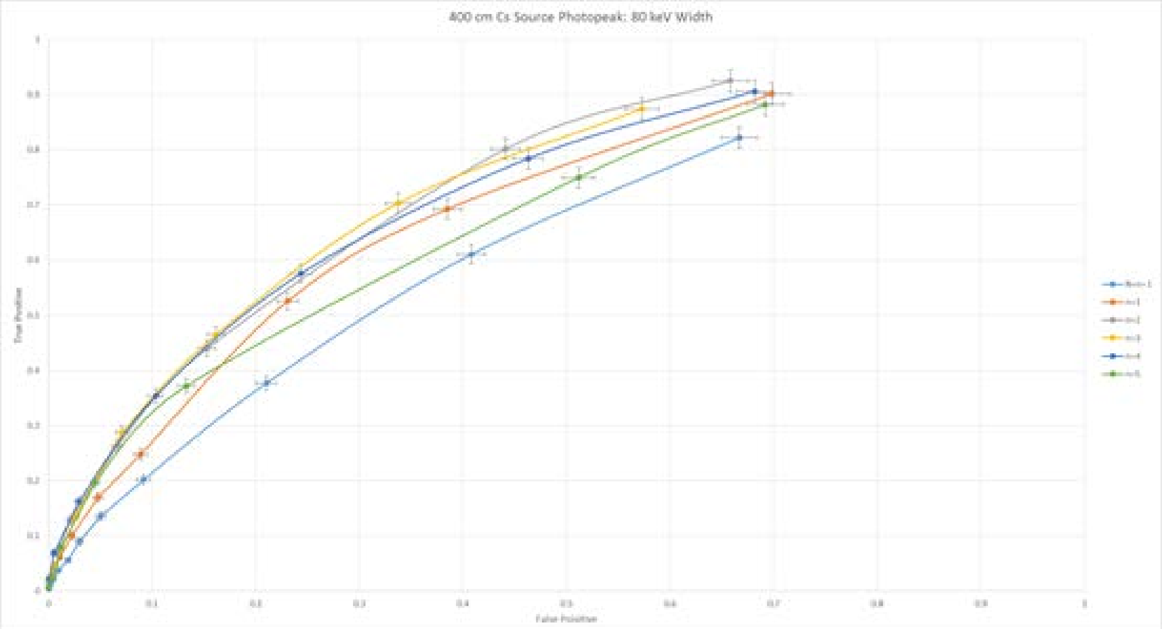


Figure I.16: 400 cm ^{137}Cs Source Photopeak Region: 80 keV Width

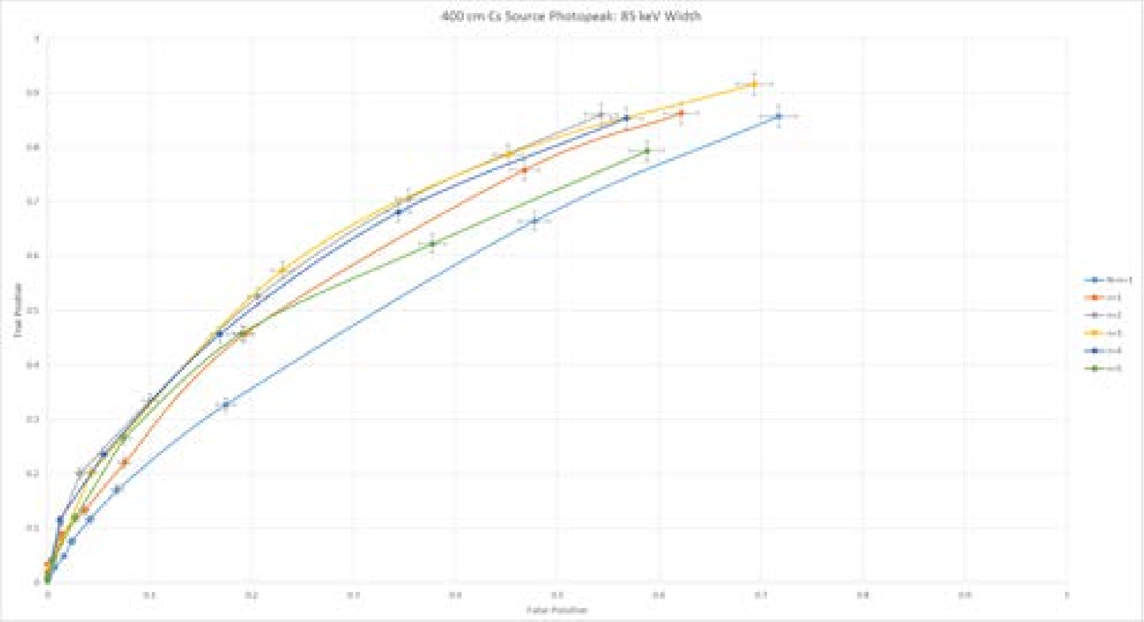


Figure I.17: 400 cm ^{137}Cs Source Photopeak Region: 85 keV Width

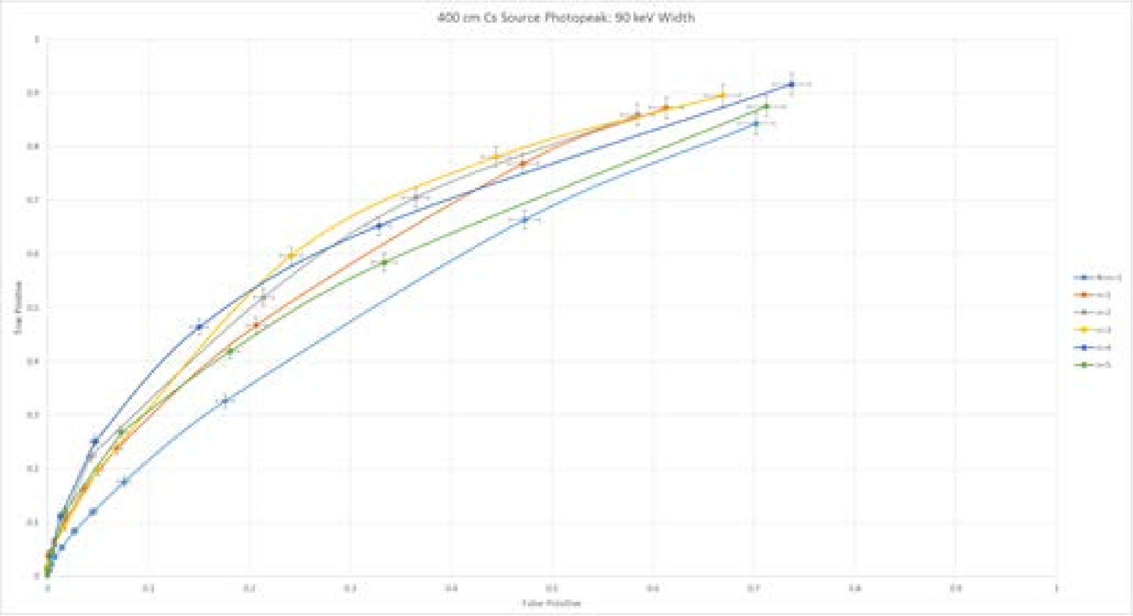


Figure I.18: 400 cm ^{137}Cs Source Photopeak Region: 90 keV Width

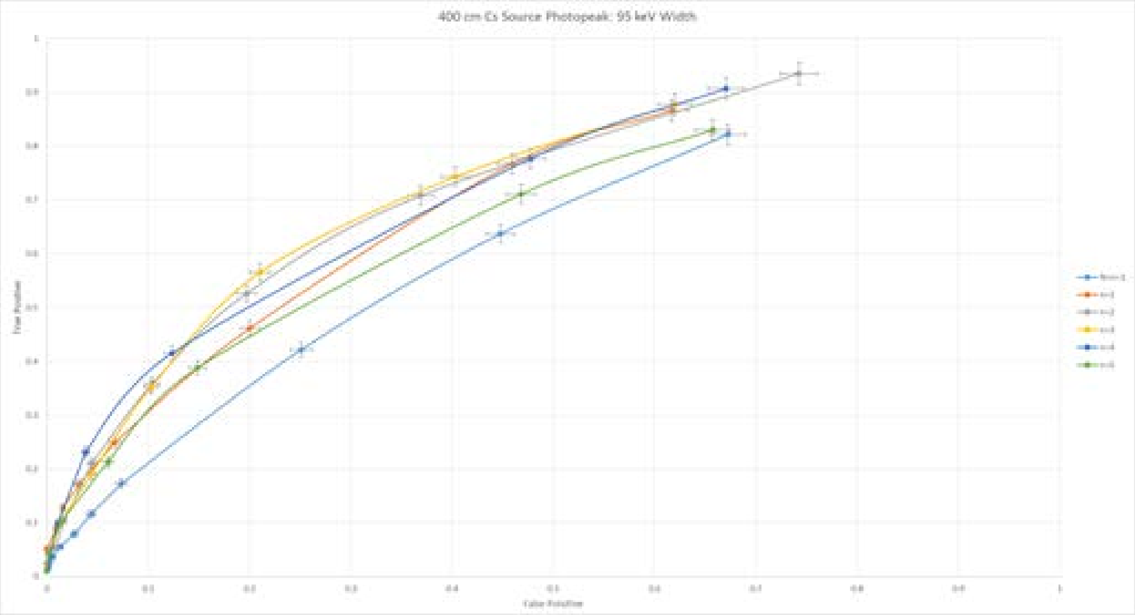


Figure I.19: 400 cm ^{137}Cs Source Photopeak Region: 95 keV Width

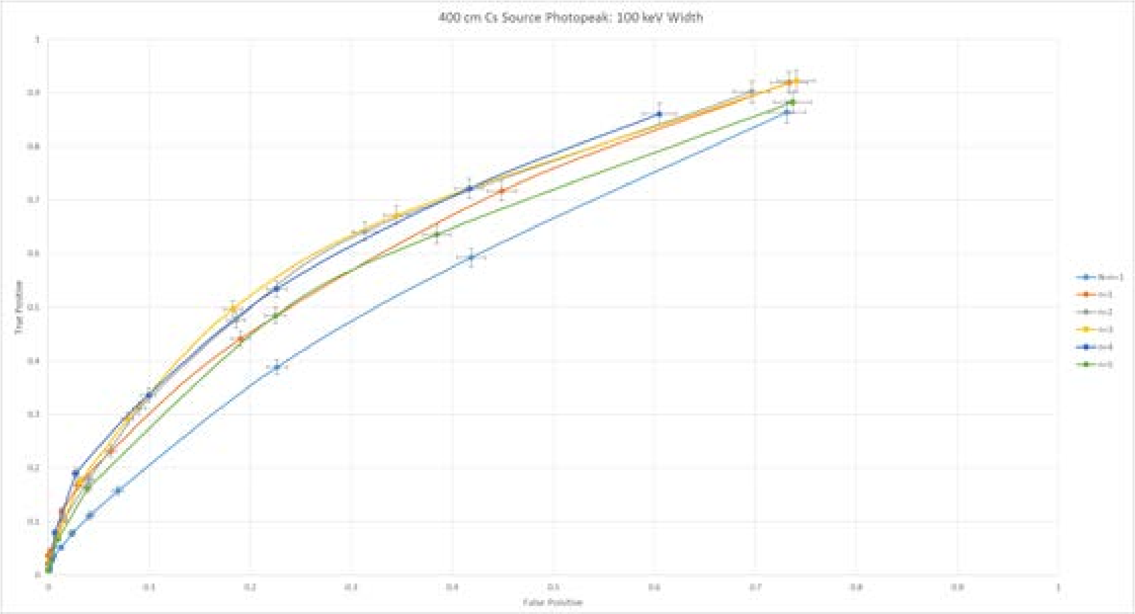


Figure I.20: 400 cm ^{137}Cs Source Photopeak Region: 100 keV Width