Master Thesis

An independent event identification pipeline for the CONNIE experiment

Author:
Iruatã Martins dos Santos Souza

Advisor:
Dr. Martín Makler

A thesis submitted in fulfillment of the requirements
for the degree of Master of Science

in the

Brazilian Center for Physics Research

Rio de Janeiro – April 2019

Declaration of Authorship

I, Iruatã Martins dos Santos Souza, declare that this thesis titled, 'An independent event identification pipeline for the CONNIE experiment' and the work presented in it is my own. I confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: _____

Date: _____

Abstract

Neutrinos are some of the most abundant and least interacting elementary particles in the Universe. They are leptons with zero electric charge, spin 1/2 and very small mass, and exist in three known active flavors – electron, tau and muon. Despite being theorized in 1930 and discovered in 1956, a number of fundamental questions about them remain unanswered, such as: What are the values of the masses of each of the flavors? Are neutrinos their own antiparticles? Do other neutrino flavors exist? These questions and the multitude of phenomena involving neutrinos are what drive Neutrino Physics, a field that have been awarded three Nobel prizes in the last 25 years. The coherent neutrino-nucleus scattering (CEνNS) is a process whereby a neutrino scatters off an atomic nucleus via neutral-current (weak) interaction; its cross section is proportional to the square of the number of neutrons in the nucleus and the square of the energy of the neutrino. This scattering covers a wide range of phenomena, and the understanding of its nature is useful to many areas of physics, from the existence of a new sterile neutrino flavor to the search for Dark Matter candidates. But even with the cross section enhancement provided by coherence, the energies for CEνNS are very low, requiring detectors with good sensitivity and a well characterized low background. The Coherent Neutrino-Nucleus Interaction Experiment (CONNIE) is the first experiment to apply the CCD imaging sensors to neutrino detection. Installed in and using electron antineutrinos from a Brazilian nuclear reactor, the detector is built for the measurement of nuclear recoils from CEνNS with energies lower than 10 eV. Since the detector produces digital images as output, the processing of these images is an essential step in the data analysis and can limit our capability of measuring the coherent scattering. Thus, CONNIE has developed its own image processing pipeline for that matter. In this work we present an independent pipeline for processing the CONNIE images. It is based on SExtractor, a widely used free software for processing astronomical images, and a number of new tools we developed and made available to the collaboration. The new pipeline was capable of reproducing the spectrum obtained with the standard pipeline. This attested the robustness of both the CONNIE and the new procedures, and showed that the spectra are essentially independent of the pipeline used. Based on the new pipeline, we implemented a methodology for assessing the efficiency and contamination in the identification of (simulated) neutrino events. We then applied this methodology to test whether parameters derived from a fit to an event's shape are helpful for separating neutrinos and spurious detections. Lastly, we applied cuts in these morphological parameters and were able to improve the contamination of our detections.

Dedicado à minha vó Josefa, que transformou a fome do sertão em amor infinito.

Acknowledgements

À Milena, minha esposa, pelo imensurável apoio intelectual, material e emocional sem os quais este trabalho não teria sido realizado. Por todo amor, sorriso, choro e luta. Por acreditar no melhor em mim, mesmo quando não acreditei. Por ser minha companheira de vida.

À Terezinha, minha mãe, por germinar em mim o amor pelo conhecimento e não medir esforços para que eu me desenvolvesse. Por me inspirar em palavra e ação. Pela coerência, referência e amor.

A Carlos, meu pai, pela referência, luta e amor.

A Ivan, também meu pai, pelo apoio material e moral. Pelo amor e dedicação.

A meu orientador, Prof. Dr. Martín Makler. Pelo apoio prático e moral, direto e indireto. Pela orientação e incentivo não só neste trabalho, mas para a Física. Pela referência intelectual e compromisso com a ciência.

À Prof. Dra. Carla Bonifazi. Pela orientação intelectual e de carreira desde a graduação. Pelas inúmeras discussões para este trabalho. Por ter me olhado e incentivado como colega. Pela referência de compromisso e rigor intelectual e prático.

À tia Lu e Malu, por me apoiarem sempre. Pelo amor e por me inspirarem.

A Cido e Ivanilde, pelo apoio e amor irrestrito. Por multiplicar as alegrias e diminuir as dificuldades.

A tio Lulu, pela doçura, amor e apoio. Por estar sempre disponível e presente.

À tia Chica, por seu amor e dedicação aqueles que a rodeiam, especialmente minha vó.

A Guilherme e Luiz, pelas alegrias, tristezas e lutas conjuntas no caminho científico e da vida. Por me apoiarem e estimularem até aqui.

A Clécio, pelas discussões sobre carreira e ciência. Pela disponibilidade e apoio, em especial durante minha estadia no Fermilab.

To Juan Estrada, for sharing knowledge generously. For all the discussion and availability. For supporting part of this research at Fermilab.

To CNPq and Fermilab, for the financial support that made this work possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

From the last decade of the 19th century to the first of the 20th century, a number of emissions from radioactivity were first observed. Among them is the β-decay, a reaction whereby a mother nucleus M transforms into a daughter nucleus D, keeping the same number of nucleons A but changing the number of protons Z:

$$M(A, Z) \rightarrow D(A, Z + 1) + e^{-}. \tag{1.1}$$

This expression suggests that the emitted electron carries the difference in energy between mother and daughter. That being true, the electron energy distribution would be narrow. However, contrary to this expectation, in 1914 Chadwick [1] observes a continuous electron spectrum. Meitner [2], in 1930, confirms that the continuity is a feature of the emitted electrons, not of possible secondary effects they may have gone through. Together, these discoveries imply energy is not conserved in formulation (1.1). Furthermore, it was observed that if M had an integer (fractional) spin, so did D. As electrons have spin 1/2, expression (1.1) also violates angular momentum conservation.

In order to solve these problems, in 1930 Pauli [3] proposed that a new particle with zero electric charge and spin 1/2 would be emitted alongside the electron in β-decay. The particle, later named *neutrino* by Fermi [4], would share the available energy with the electron, conserving total energy and angular momentum. In opposition to the common practice of the time, neutrinos were postulated *before* being observed. Indeed, observation proved to be a great challenge, to be surmounted only in 1956 by Cowan *et al.* [5], an achievement awarded with the 1995 Nobel Prize in Physics.

More than half a century after their first observation, the standard model (SM) of particle physics defines neutrinos as fundamental leptons with zero electric charge, zero mass, spin 1/2, and which only respond to the weak interaction. They exist in three distinct (active) flavors, each with its corresponding antiparticle: electron neutrino (ν_e), muon neutrino (ν_μ) and tau neutrino (ν_τ). In addition to the understanding of the particles themselves, *neutrino physics* can be used in the study

of the sources of neutrinos (*e.g.* stars), of the medium through which they propagate (*e.g.* Earth's atmosphere), and in technological applications (*e.g.* counting a nuclear reactor's live neutrino flux for safety purposes).

1.1 Neutrino sources

When cosmic rays, *e.g.* protons, interact with the Earth's atmosphere, the deposited energy is sufficient to produce secondary particles. The decay of the latter produce neutrinos predominantly via the following mechanisms [6]:

$$\pi^+ \to \mu^+ \nu_\mu \qquad \mu^+ \to e^+ \nu_e \bar{\nu}_\mu \tag{1.2}$$

$$\pi^- \to \mu^- \bar{\nu}_\mu \qquad \mu^- \to e^- \bar{\nu}_e \nu_\mu. \tag{1.3}$$

The neutrinos thus produced are called *atmospheric neutrinos*. Since many atmospheric processes triggered by cosmic rays can't be directly detected and studied, atmospheric neutrinos play an important role in providing information about them at ground-level experiments. One such experiment was Kamiokande [7], located in Japan and consisting of a large water Cherenkov detector. Originally designed to search for the proton decay, it was also sensitive to neutrinos of atmospheric origin. The first observation of the latter [8] earned Kamiokande's director the 2002 Nobel Prize. Based in Antarctica, the IceCube experiment [9] also uses the Cherenkov process and was able to detect atmospheric neutrinos [10].

The Sun creates energy by nuclear fusion, effectively transforming four hydrogen nuclei into a helium-4 nucleus, $4\mathrm{p} \to {}^4\mathrm{He} + 2\mathrm{e}^+ + 2\nu_e$. The electron neutrinos produced by this reaction are in the MeV energy range and their flux on Earth lies in the 10^{10} cm^{-2} s^{-1} scale [6], configuring the Sun as the most powerful neutrino source available on our planet. These neutrinos can be used to analyze the energy generation processes that take place in the Sun and in other stars, and were first detected in 1968 by Davis *et al.* [11] at the Homestake experiment. The detection was based in a neutrino being captured in the radiochemical process

$${}^A_N Z + \nu_e \to {}^A_{N-1} (Z+1) + e^-, \tag{1.4}$$

and the solar neutrino flux was measured. The number thus found was a factor of seven below the expected by the Solar Standard Model, a deficit that became known as the solar neutrino problem. Decades later, the Kamiokande [8] and Super-Kamiokande [12] experiments measured solar neutrinos fluxes to be inconsistent with both the Solar Standard Model and Davis' account.

Nuclear fusion processes taking place inside a star generate outward pressure, and an inward force is exerted by the star's own gravity. While these two mechanisms equate, hydrostatic equilibrium is maintained. When fusion can no longer counterbalance gravity, the star may collapse in the form of an explosion called a supernova. In supernovae of core collapse types, gravity compresses the star's core and disintegrates nuclei, consequently producing electron neutrinos. The remaining neutron core has a temperature of around 10^{11} K and thermally emits neutrino-antineutrino pairs of all flavors and $E_\nu < 100$ MeV. These particles carry 99% of the total energy released in the explosion. Powerful and short lived as they are, supernovae can produce in ten seconds as much neutrinos as the Sun in all its lifetime [6]. In 1987, three different experiments detected *supernova neutrinos* from the SN 1987a supernova. Kamiokande-II [13], IMB [14] and Baksan [15] respectively counted 12, 8 and 5 antineutrinos in the same time frame, around three hours before the visible light reached the Earth.

In addition to natural phenomena, neutrinos can also be produced artificially. Nuclear reactors perform fission of four distinct isotopes: ^{235}U, ^{238}U, ^{239}Pu and ^{214}Pu. The products of this process are unstable atoms, and may undergo a β-decay:

$$(A, Z) \rightarrow (A, Z+1) + e^- + \bar{\nu}_e \quad (\beta^-) \tag{1.5}$$

$$(A, Z) \rightarrow (A, Z-1) + e^+ + \nu_e \quad (\beta^+) \tag{1.6}$$

$$e^- + (A, Z) \rightarrow (A, Z-1) + \nu_e \quad \text{(electron capture)}. \tag{1.7}$$

An electron antineutrino can also be produced when a neutron is captured by a ^{238}U atom. These four processes are the dominant forms of neutrino production in a reactor. Their spectrum is well known and shown in Figure 1.1. Nuclear reactors are the strongest antineutrino sources on Earth, producing about 3.1×10^{16} of these particles per second per megawatt of thermal power [16]. Owing to this high flux and the control reactors allow, many experiments use them as the neutrino source. The first neutrino detection, by Cowan *et al.*, used as source the Savannah River reactor in the United States [5]. The CONNIE experiment, described in detail in Chapter 2, uses neutrinos from the Angra II reactor in Brazil.

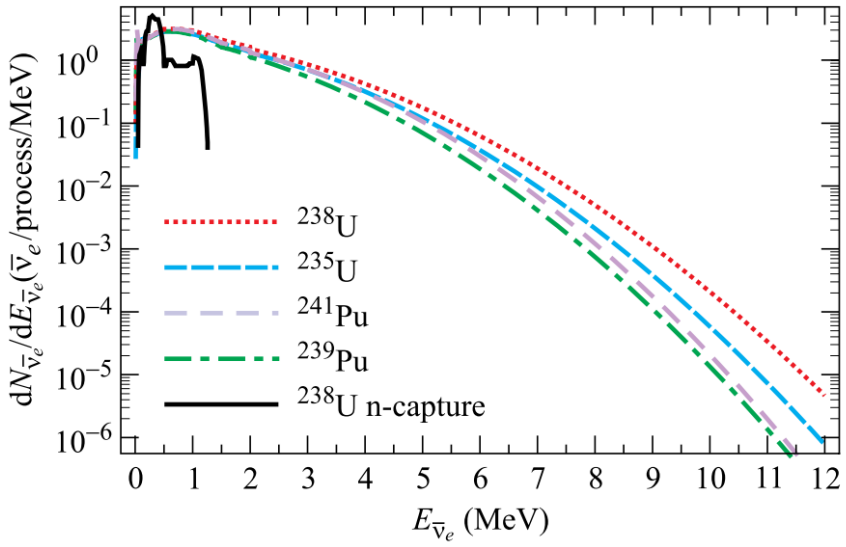

FIGURE 1.1: Electron antineutrino spectrum for the various processes in a nuclear reactor. Taken from [16].

Particle accelerators can offer another controlled environment for neutrino production. In the majority of setups of this type [17], a proton beam is accelerated and hits a target, producing mainly kaons and pions. These are focused and directed by magnetic fields to a region where they may decay into muons and neutrinos. By the end of this region, muons, electrons and hadrons are absorbed. Depending on the applications, the remaining neutrinos can then be detected nearby and/or far away, by what are respectively called short and long baseline detectors. Big collaborations are currently working on the next generation accelerator-based neutrino experiments. The Deep Underground Neutrino Experiment (DUNE) [18] will use an accelerator beam from Fermilab and have its short baseline detector also at installed Fermilab. The neutrino beam, planned to be the world's most powerful one at 1.2 MW with peak energy of about 3 GeV, will then travel 1300 km underground to the long baseline detector at the Sanford Underground Research Facility.

There are also less widespread processes and techniques worth mentioning. The COHERENT experiment [19] [20] uses the Spallation Neutron Source in the United States as a neutrino source. In this facility, pulses of accelerated protons collide with a liquid mercury target, producing neutrons and pions. The decay chain of the pions in turn generate muon, electron and antielectron neutrinos, with a flux of $\sim 2 \times 10^7$ cm^{-2} s^{-1} at 20 m from the target, and energies up to 60 MeV [21]. In 2017, COHERENT reported the first observation of the coherent elastic neutrino-nucleus scattering [22], the essential physical process that motivates this work, described in the next section. On another venue, isotopes with long lives present inside the Earth are also subject to β decay, and therefore produce electron antineutrinos naturally [23]. These so called geoneutrinos provide geological information directly across tons of rock and were detected by the Boxerino experiment in Italy [24] and KamLAND in Japan [25].

1.2 Neutrino flavor oscillations

There is no underlying principle preventing neutrinos to have mass. Their zero mass in the SM stems from the absence of evidence to the contrary by the time the framework consolidated, between the 1960s and 1970s. Let us suppose neutrinos do have mass and exist in n different flavors. In that case, we have n flavor eigenstates and n mass eigenstates, connected via a unitary mixing matrix U. The mass eigenstates are stationary and evolve according to the Schrödinger equation. Thus, if a neutrino is emitted in mass eigenstate i with momentum p by a source at $x = 0$, its evolution is given by

$$|\nu_i(x,t)\rangle = \mathrm{e}^{-i(E_i t - px)} \, |\nu_i\rangle \, . \tag{1.8}$$

On the other hand, a flavor eigenstate $|\nu_\alpha\rangle$ produced with momentum p at $t = 0$ will evolve as

$$|\nu(x,t)\rangle = \sum_i U_{\alpha,i} \mathrm{e}^{-iE_i t} \, |\nu_i\rangle = \sum_{i,\beta} U_{\alpha,i} U^*_{\beta,i} \mathrm{e}^{-i(E_i t - px)} \, |\nu_\beta\rangle \, , \tag{1.9}$$

where the Greek and Latin indices refer to flavor and mass eigenstates, respectively. The particle being relativistic, $p \gg m_i$ and $E \approx p$, so $E_i \simeq E + m_i^2/2E$. Thus, if the masses m_i are not all equal, the final state may be different from the initial one. In other words, a neutrino created in one flavor spontaneously transitions into other flavors and back, and may be detected as a different neutrino. This so called neutrino flavor oscillations were first predicted in 1957 by Pontecorvo [26]. From equation (1.9) we can derive the probability of flavor α transitioning to flavor β in time t[1]:

$$P(\alpha \to \beta)(t) = \langle P_{\alpha\to\beta} \rangle + 2 \, \mathrm{Re} \sum_{j>i} U_{\alpha i} U^*_{\alpha j} U^*_{\beta i} U_{\beta j} \exp\left(-i \frac{\Delta m^2_{ij}}{2} \right) \frac{L}{E}, \tag{1.10}$$

where $\Delta m^2_{ij} = m^2_i - m^2_j$ and L is the distance traveled by the particle. Hence, if we produce neutrinos in flavor α, put a detector at L, and count the number of flavor β particles detected, we can measure the difference of the squared masses of the flavors, but not the masses themselves.

In 1998, the Super-Kamiokande collaboration [27] was able to provide experimental evidence of oscillations in atmospheric neutrinos. They calculated the ratio of muon to electron neutrinos (ν_μ/ν_e) expected to arrive at the detector with energies from 0.1 to 10 GeV to be 2 with an uncertainty of less than 5%. By detecting the leptons produced by interactions of neutrinos with nuclei, they could measure the same ratio and compare model with data. This comparison showed a deficit in the detection, dependent on the zenith angle of the neutrino direction of arrival. No combination of experimental uncertainties or prediction of the neutrino fluxes was able to explain the findings. However, a model where muon neutrinos would oscillate to either tau neutrinos or a new type of

[1] See [6] for the complete derivation.

neutrino was consistent with both the ratios and the zenith angle dependence, providing the first strong evidence for oscillation.

Shortly after, in 2001, the SNO collaboration [28] presented more evidence of flavor oscillations using solar neutrinos at the Sudbury Neutrino Observatory (SNO), solving the solar neutrino problem. They used a charged-current reaction (CC), sensitive to only electron neutrinos, a neutral-current reaction (NC), and an elastic electron scattering reaction (ES), both sensitive to all active neutrino flavors. If the Sun's ν_es didn't oscillate, the flux measured from CC should be consistent with the one from ES, indicating a prevalence of that flavor in the detector. However, the flux from CC was significantly smaller, consistent with the oscillation of the electron neutrinos into other neutrino flavors.

Together the SK and SNO discoveries were responsible for the 2015 Nobel Prize, and led to a number of open questions. First, we do not know the values of the masses for the active neutrino flavors. Although we know that the masses of two neutrino flavors are almost degenerate and the other one is very different, we don't know if the latter mass is higher or lower: the *mass hierarchy* is still unknown. The answers to these two questions will provide essential information on the very mechanism that imbues neutrinos with mass. Still regarding the properties of neutrinos themselves, the mixing matrix U can be expressed in terms of angles, the values of which are yet unknown. Moreover, since neutrinos have mass but zero charge, are neutrinos and antineutrinos the same or not? In other words, are neutrinos Majorana or Dirac fermions? And related to that, does the leptonic sector violate CP-invariance? That would occur if the transition probability from one neutrino flavor to another is different from the probability for the respective antineutrinos. Also, on a more fundamental level, how many neutrino flavors are there? While there are three families in the SM, there could also be other ones that do not interact with the active flavors or other SM particles at all. These *sterile neutrinos* could, however, oscillate with the active ones, complicating even more the mixing matrix. These questions drive the majority of the current and planned future research in neutrino physics.

1.3 Coherent neutrino-nucleus scattering

The coherent elastic neutrino-nucleus scattering (CEνNS) is a process whereby a neutrino of any flavor scatters off an atomic nucleus via neutral-current (weak) interaction. The energy transfer has the form of a nuclear recoil, and the corresponding cross section is [29]

$$\frac{\mathrm{d}\sigma}{\mathrm{d}E_{\bar{\nu}_e}\mathrm{d}E_{rec}}(E_{\bar{\nu}_e}, E_{rec}) = \frac{G_F^2}{8\pi}\left[Z(4\sin^2\theta_W - 1) + N\right]^2 \times M\left(2 - \frac{E_{rec}M}{E_{\bar{\nu}_e}^2}\right)|f(q)|^2, \tag{1.11}$$

where $E_{\bar{\nu}_e}$ and E_{rec} are respectively the incident neutrino and the nucleus recoil energies, G_F is the Fermi coupling constant, θ_W the weak mixing angle, and $f(q)$ the nuclear form factor at momentum

transfer q. M, N, and Z are respectively the nucleus mass, neutron number and atomic number. The requirement for coherence is that the neutrino interacts with the nucleus as a whole, so if R is the nuclear radius, coherence is expressed as $q^2 \lesssim 1/R^2$, a condition generally satisfied for $E_{\bar{\nu}_e} < 50$ MeV [30]. In this process, the maximum recoil energy is $2E_{\bar{\nu}_e}^2/M$, corresponding to approximately 10 keV for silicon atoms and neutrinos of 12 MeV. These recoil energies allow us to take $|f(q)| \approx 1$, and the small contribution of protons to the form factor permits $\sin^2\theta_W \approx 1/4$ [31]. Integrating (1.11) over the possible recoil energies and using the mentioned approximations, we have a total cross section of

$$\sigma_T(E_{\bar{\nu}_e}) = \frac{G_F^2}{4\pi} N^2 E_{\bar{\nu}_e}^2 \approx 4.22 \times 10^{-45} N^2 (E_{\bar{\nu}_e}/\text{MeV})^2 \text{ cm}^2. \tag{1.12}$$

The processes involved in CEνNS cover a wide range of phenomena and can help to unravel answers in many frontiers of physics. For example, neutrinos from all types of supernovas, including the recently discovered Kilonovas [32] meet the requirement for coherence. As CEνNS proceeds via the neutral current, an event originated from it does not distinguish between the active neutrino flavors. Thus, if we measure a variation in neutrino flux with distance, it would mean the initial active flavors oscillated to a sterile neutrino. Extensions to the SM usually involve the addition of at least one sterile neutrino, so low energy neutrinos under coherence can also be used to test theories of physics beyond the Standard Model [33] [34].

Additionally, a number of experiments searching for dark matter (DM) candidates are based on the possibility of them interacting in the weak sector. Nonetheless, in recent years the *weakly interacting massive particle* (WIMP) paradigm of DM has been challenged, favoring searches for low mass candidates in an effort that requires low detection thresholds. The technology used for this type of detection can be and was applied to CEνNS research [35] [36]. As these dark matter experiments enhance their sensitivity, being capable of detecting lower energies, they reach the *neutrino floor*, a source of insurmountable background composed of neutrinos. Also, sterile neutrinos are being considered as DM candidates. Therefore, the study of the CEνNS can also provide essential information related to dark matter candidate search.

Despite the cross section enhancement provided by coherence, the very low recoil energies for CEνNS requires detectors with good sensitivity and a well characterized low background. Such practical challenges contributed for the 43 years it took between the theoretical prediction and its detection by the COHERENT experiment [20]. The Coherent Neutrino-Nucleus Interaction Experiment (CONNIE) is the first experiment to apply Charge-Coupled Devices (CCDs) to neutrino detection. Installed next to a Brazilian nuclear reactor, the detector is built for the measurement of nuclear recoils from CEνNS with energies as low as 30 eV. Owing to the small dimensions and weight of the setup, CONNIE also allows the detector to be moved with ease, opening the possibility for measuring sterile neutrino oscillation in the range of meters.

1.4 Outline

In the present work we developed a data pipeline for identifying specific events present in the CONNIE images. It was written and works independently of the current collaboration's code. Among other features, it segments images into events/objects, builds spectra, and identifies neutrino candidates. This new set of tools allows us to check the collaboration's analysis procedures for errors and improvements, and assess the robustness of their results by verifying that they do not depend on the pipeline used. Additionally, we tested novel ways of measuring neutrino-like events and obtained parameters that may be used for enhancing the selection of low energy events towards neutrino identification.

The thesis is organized as follows. Chapter 2 describes the CONNIE detector, its sensors and operation. Chapter 3 the new extraction of events with SExtractor and the respective production of spectra. Chapter 4 progresses to the analysis of the efficiency and contamination for extractions. Chapter 5 discusses improvements on contamination by selecting events using morphology fits. Chapter 6 summarizes our results and presents venues for future work. Lastly, the appendices serve as a reference on the software used and developed in the course of this work, and which is available for the CONNIE collaboration in its code repository.

Chapter 2

The CONNIE experiment

The Coherent Neutrino-Nucleus Interaction Experiment employs a Charge-Coupled Device (CCD) based detector to study the coherent neutrino-nucleus scattering. Although originally devised in 1969 as solid-state memories, CCDs can and have been largely used as imaging sensors in many different applications, from consumer electronics to space telescopes. Their energy and spatial resolution and low electronic noise make viable particle detectors. Specifically, their low detection thresholds appeal to the study of low energy neutrinos.

2.1 The operation

CONNIE is set up at the Almirante Alvaro Alberto Nuclear Power Plant, in the state of Rio de Janeiro, Brazil. The detector is located inside a shipping container 30 m from the core of the Angra-2 reactor, a Pressurized Water Reactor with a thermal power of 3764 MW. The total neutrino flux produced is 1.21×10^{20} $\bar{\nu}$/s, corresponding to a flux density of 7.8×10^{12} $\bar{\nu}/\mathrm{cm}^2/\mathrm{s}$ at the detector [37], which is large enough to justify its use as the neutrino source for the experiment.

The current detector is an array of 14 CCDs with 4112×4130 pixels (Fig. 2.1), designed by the *Lawrence Berkeley National Laboratory* (LBNL) and *Fermi National Laboratory* and produced at LBNL. Taking advantage of the latest progresses in production technology, each CONNIE sensor is 675 μm thick and possess a resistivity greater than 14 kΩcm [38]. Owing to the high resistivity and the application of a reverse bias, the CCDs are *fully depleted* of majority charge carriers (holes) in the whole sensitive area, so that the signal in a pixel is only due to particle hits and the sources of noise. The sensor has a mass of 6.1 g and pixel size of 15 μm $\times$ 15 μm. Its optimal root mean square (RMS) noise is $\sigma_{RMS} = 1.5e^-$ [39] or, equivalently, 5.54 eV. CONNIE considers a pixel has collected charge when it has a signal above $5\sigma_{RMS}$, *i.e.* the experiment's energy threshold is $\simeq$ 28

eV. Similar CCDs were first produced in large scale for the Dark Energy Survey (DES)[1], and later for the Dark Matter in CCDs (DAMIC) experiment [36].

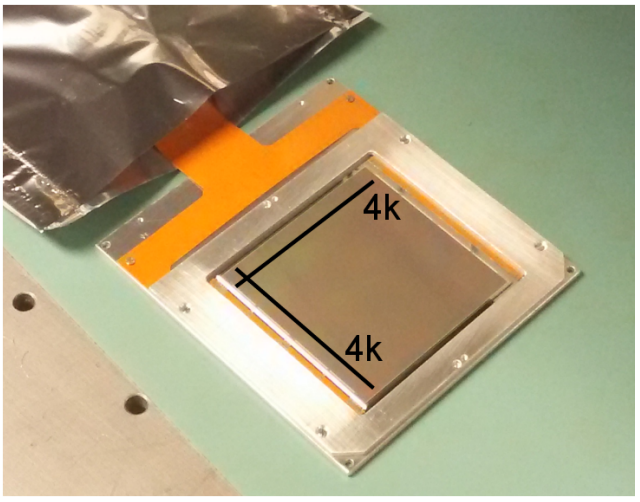

FIGURE 2.1: A CONNIE 4k × 4k pixels CCD. Picture from the collaboration.

The array is mounted in a copper box and enclosed in a vacuum vessel at 10^{-7} torr pressure and cryogenic temperatures $\sim$ 140 K. Inside the vessel, a lead cylinder 15 cm-long acts as a shield for radioactive contaminants coming from the readout electronics. To shield from external radiation, there is 30 cm of high density polyethylene to stop neutrons, 15 cm of lead to shield photons, and an inner 30 cm of polyethylene to stop neutrons created in the interaction of gammas with lead (see Figure 2.2).

[1]Initially CONNIE used the exact detectors made for DES [35], but currently they are produced specifically for the experiment.

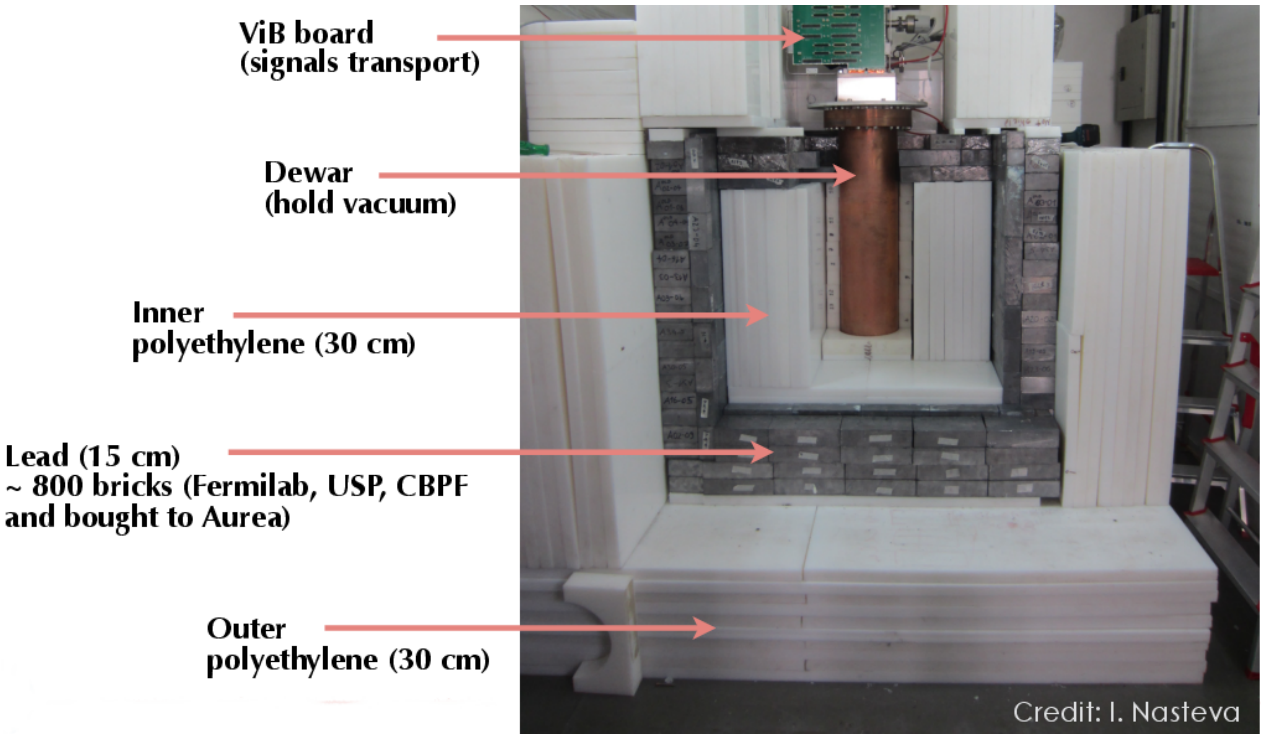

FIGURE 2.2: CONNIE montage scheme. Picture from Irina Nasteva irina@if.ufrj.br.

The energy calibration of the experiment uses the K_α and K_β lines of copper, respectively at 8.05 keV and 8.9 keV [40], produced by fluorescence from the detector enclosing material. After a set of images is stored, a histogram is built using the charge equivalent of the signal of each pixel. The signal's amplitude distribution is then fitted by a sum of two Gaussians, described by six parameters: amplitude, central value and width of each of the peaks [36]. The result of a calibration is shown in Figure 2.3. This technique allows calibration to be done in-situ and the monitoring of the gamma background in the detector.

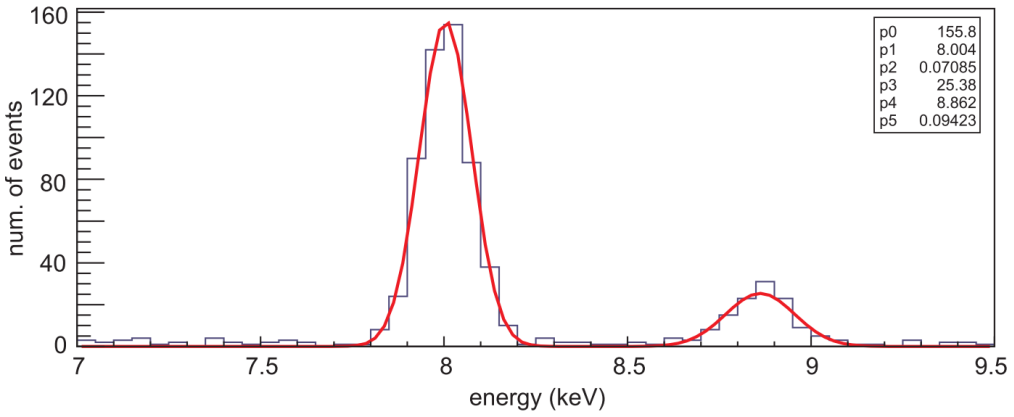

FIGURE 2.3: Result of an energy calibration with Cu the K_α and K_β fluorescence lines. The red line is a sum of two Gaussians fit with six parameters: amplitude, central value and width of each of the peaks. Taken from [36].

2.2 Charge-Coupled Devices

A CCD is composed of a grid of units called *pixels*. Figure 2.4 shows the cross section of a pixel. Its basic building block is a metal-oxide-semiconductor (MOS) capacitor, composed of a poly-silicon gate, an oxide layer and a silicon p-n junction. When on duty, a bias voltage is applied to the gate, forming a *buried channel*, *i.e.* a potential well in the p side of the junction. If a particle interacts with the atoms, nuclei or electrons in the sensitive volume of a pixel (Fig. 2.4), it may ionize the atoms and free electrons or holes, generating charge. For that to happen, the particle must have sufficient energy to make the carrier surmount the energy gap between the valence and conduction bands of the solid. In silicon, and so in CONNIE CCDs, the mean ionization energy is $E_{Si} = 3.69$ eV at $T \approx 140$ K [41]. The charge carriers thus created are attracted by the potential well, drift along the z direction in the figure, and get collected and held near the p-n junction, less then 1 μm below the gates. While moving towards the gates, the charge carriers diffuse in the directions perpendicular to the movement. This diffusion is a strictly monotonically increasing function of the event's depth z [42].

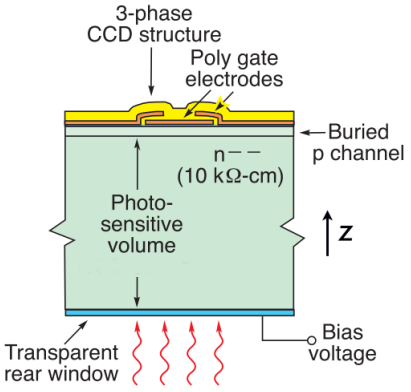

FIGURE 2.4: Diagram of the cross section of a pixel in a fully depleted buried-channel CCD [39].

Each type of particle hitting a CCD leaves a characteristic signature in the pixels, as in Figure 2.5. This happens because the particles induce charge generation in different ways. A heavy particle, such as an α, creates a region with a high density of electron-hole pairs, in which the external electric field cannot penetrate. This region then diffuses laterally, towards regions of smaller charge density [43], leaving a *circle* pattern in the pixels. When an incident electron travels through the CCD, its track is perturbed by the interaction with Si nuclei and electrons, and forges a *worm* pattern. For our purposes, muons interact mostly in the same ways as electrons, but their much higher mass prevents them from being greatly perturbed, thus impressing a *straight* track on the pixels. There are also events that deposit their whole energy within the volume of a single pixel, named point-like events. The signal for such an event is a very small collection of pixels originated solely from the diffusion process, and so the event is called a *diffusion-limited hit*. Different physical processes may produce this type of hit, nuclear recoils being the example we are most interested in.

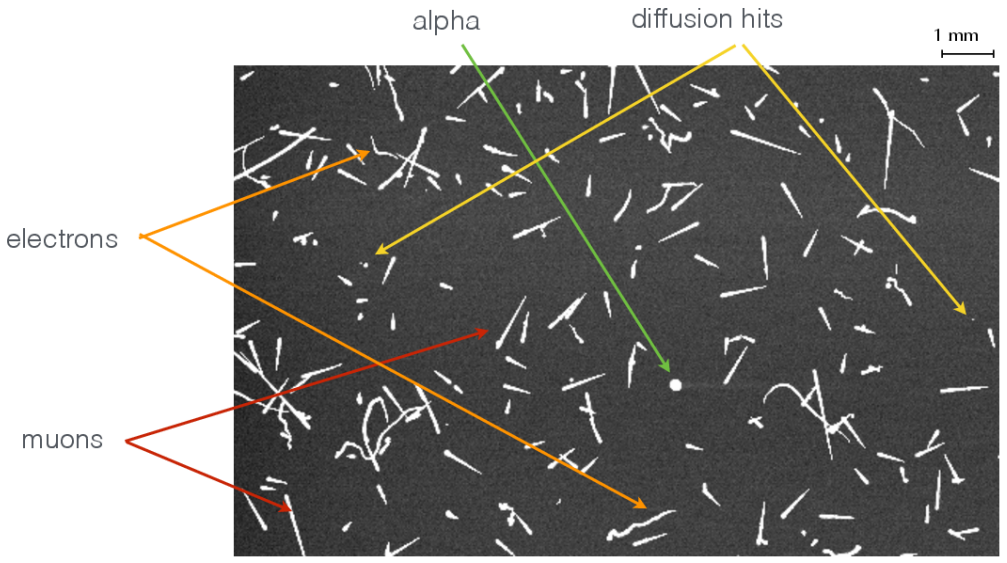

FIGURE 2.5: Characteristic particle tracks in a CCD image. Image from Carla Bonifazi bonifazi@if.ufrj.br.

As previously noted, the pixels in a CCD are arranged in a grid (Figure 2.6). Vertical strips form the vertical register, with each strip defining a row in the grid. Division into columns is achieved by the channel stops, with horizontal strips acting as charge barriers.

After a fixed exposure time, charge collection stops and the charge carriers generated in each pixel need to be transferred. The paradigm for this process is the 3-phase CCD, depicted in the right panel of Figure 2.6. In state 1, phases P_1 and P_3 have a repulsive potential (high) while phase P_2 has an attractive (low) one; this is the configuration of charge collection. In sequence, P_1 goes low and the collected charge spreads. State 3 has P_2 high, again squeezing the charge in a single phase, now P_1. The switching continues until in state 7 the charge has been completely transferred to the pixel on the left, which sits on the next row. Note that all the pixels in a row and all the rows in the grid are transferred at the same time.

The last row, leftmost in our depiction, is transferred to the horizontal register, at which point the transfers in the vertical register are paused. The same phase scheme is used to transfer the charge carriers in the horizontal register—now moving downwards relative to the figure—, and the pixel at the bottom is pushed to the sense node, where its signal is read. When all pixels in the horizontal register have been sequentially transferred and read, the vertical register transfer resumes. Only after the whole grid is read, then charge collection may resume.

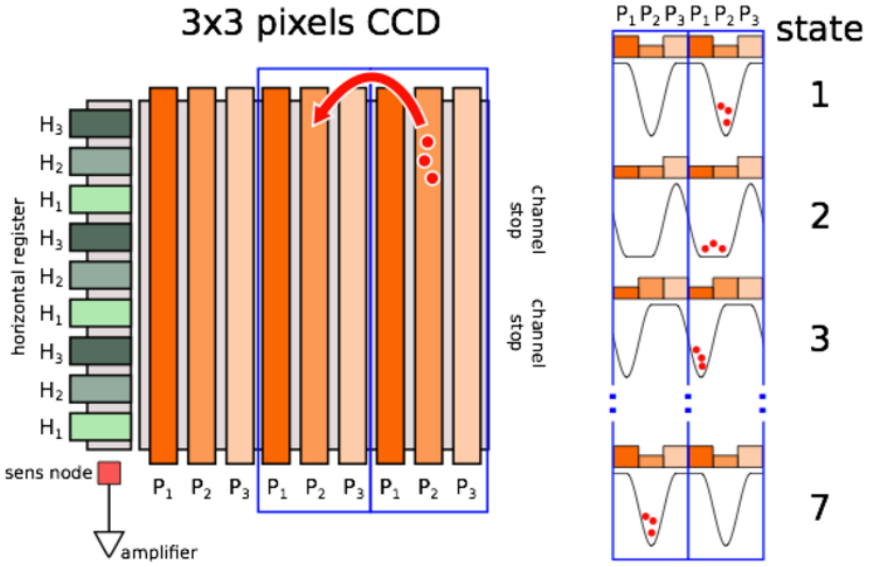

FIGURE 2.6: Diagrams of the top view of a 3×3 pixels CCD (left), and states of charge transfer between two pixels of a CCD. Image from Carla Bonifazi bonifazi@if.ufrj.br.

A major feature of CCDs is the linearity of their energy response, valid for orders of magnitude (Fig. 2.7). This means the amount of charge read per pixel is directly proportional to the energy deposited by the incident particle. CCDs also show a wide dynamic range, being capable of discerning energy values orders of magnitude apart. For that it is usual to have an amplifier after the sense node (Fig. 2.6), scaling the signal to a reasonable working range. Since one pixel is read at a time, knowing the dimensions of the grid and the time it takes for reading a single pixel, we can digitize, convert and store the CCD output as a digital image for latter analysis.

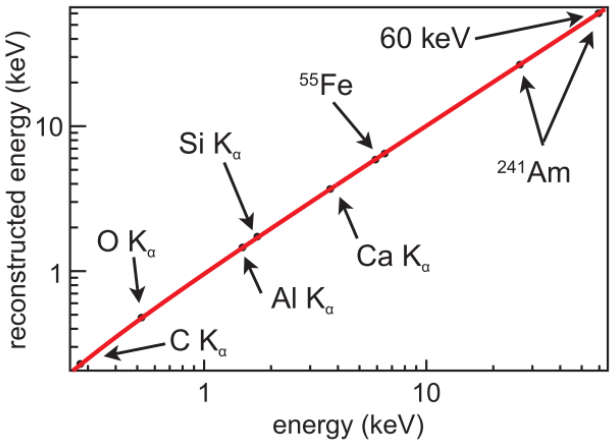

FIGURE 2.7: CCD linear response to various X-ray lines from 227 eV to 60 keV [35].

2.3 Data acquisition

The CONNIE CCDs are setup with two horizontal registers instead of one. As explained in the previous section, the left amplifier reads the collected charge. Simultaneously, the right amplifier reads from its sense node. However, as the rows of pixels are transferred to the left, their charges are emptied. Therefore, when the right amplifier reads a signal, it is actually reading the noise of the readout process itself.

In sequence, the signal output by both amplifiers is digitized and stored in four *parts* of 8540 columns and 1055 rows each. As shown in Figure 2.8, for each amplifier a row starts with a region of 8 columns—the *prescan*—, followed by an *active area* of 4112 columns, and a final region of 150 columns—the *overscan*. Not shown in the figure is the vertical overscan region, a 90–row region in the end of the last part. Note that only the active areas exist physically on the CCD, the prescan and overscan are created by reading from the sense node when no accumulated charge exists. The parts are stored in FITS files [44], a standard format for astronomical images. Files in this format are composed of textual headers and binary data representing an image or a table. A header and its corresponding data form a *Header Data Unit* (HDU), and there may be multiple HDUs (extensions) per file. A *raw image* (Fig. 2.8) is a FITS file resulting from 3 hours of exposition plus ∼15 min of reading of the 14 CCDs, and has one HDU per CCD. A batch of raw images taken with the same detection configuration is called an *image set*.

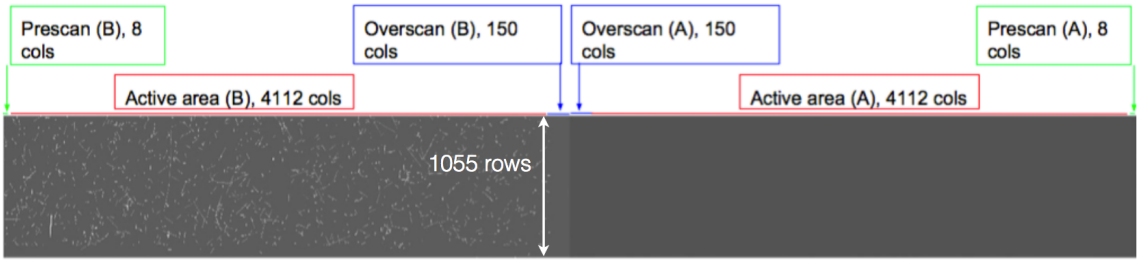

FIGURE 2.8: Dimensions and regions of a *raw image* from CONNIE. Left side corresponds to the accumulated charge, right side to the readout noise.

There are three major sources of noise in the CONNIE operation. First, a form of white noise generated by the electronics of the readout. Second, the electronic setup as a whole, including the external connections, also produces noise. When the left sense node reads a pixel, the right sense node also performs a read. However, in the latter's case the values read are not the accumulated charge, but the global *correlated noise* of the electronics. Lastly, we have the Poissonian noise called *dark current*. It is the signal corresponding to charge generated thermally and spontaneously in the CCD.

2.4 Image processing

The steps shown in Figure 2.9 compose the first part of the CONNIE image processing pipeline, and are applied to every image set.

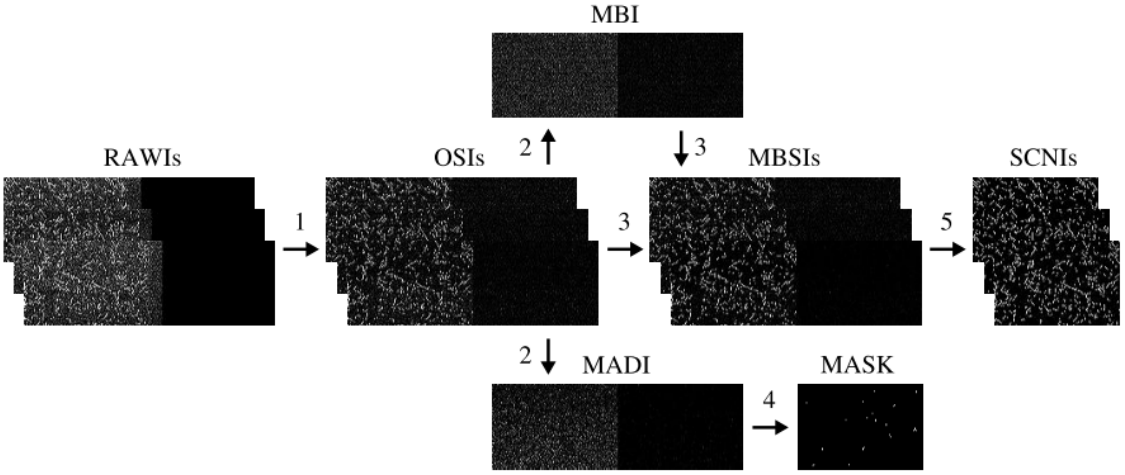

FIGURE 2.9: The steps of the CONNIE image processing pipeline [45].

The first step is the subtraction of the *readout noise*, which is applied to each raw image. We take the overscan pixel values, make a histogram and compute its mean μ. This value is then subtracted from the signal of every pixel in the raw image, producing the *overscan subtracted image* (OSI), which has both prescan regions removed. In other words, if I_{ij} is the ADU value for the pixel at position (i, j), for every pixel

$$I_{ij}^{osi} = I_{ij}^{raw} - \mu_{readout}. \tag{2.1}$$

The next step is the computation of the *master bias* for an image set. For a pixel (i, j), we take its value on all OSIs of an image set. The resulting distribution of values is a Gaussian. Repeating this process for all pixels, we can construct two images. In the *master bias image* (MBI), a pixel's value is the median of its values across the image set:

$$I_{ij}^{mb} = median(I_{ij}^{k}), \tag{2.2}$$

where k is the image number and runs through all images of the set.

And in the *median absolute deviation image* (MADI), a pixel's value is the standard deviation of its Gaussian:

$$I_{ij}^{mad} = \sigma_{ij}. \tag{2.3}$$

Note that this step consumes a set of images and produces just one MBI and one MADI. This is different from the other steps since they consume a single image and produce a single image, even if we apply them to every image of an image set.

The master bias obtained in the previous step is subtracted from an OSI, resulting in a *master bias subtracted image* (MBSI). From each pixel (i, j) in an OSI, we subtract its value from the MBI:

$$I_{ij}^{mbs} = I_{ij}^{osi} - I_{ij}^{mb}. \tag{2.4}$$

Some pixels in a CCD may need to be excluded from processing, *e.g.* the saturated ones. We define an anomalous pixel as one in which the standard deviation in the image set was greater than a threshold σ_{ref}. In other words, anomalous pixels are those that have their distribution of values wider than a reference. From the MADI we generate MASK images, with only half the width of the MADI, in which anomalous pixels have value one and non-anomalous zero:

$$I_{ij}^{mask} = \begin{cases} 1, & \text{if } \sigma_{ij} > \sigma_{ref} \\ 0, & \text{otherwise} \end{cases} \tag{2.5}$$

With the exception of MASKs, our images so far have their left halves corresponding to accumulated charge, and their right halves to electronic noise. The last cleaning step is the subtraction of the correlated noise from an MBSI, that produces a *subtracted correlated noise image* (SCNI), with half the width of the input one. First, we denote by R_{ij}^{mbs} and L_{ij}^{mbs} the pixel (i, j) on the right and left sides of an MBSI. Then we compute a linear combination of the R_{ij}^{mbs} in all the CCDs, and subtract it from the L_{ij}^{mbs}:

$$I_{ij}^{scn} = L_{ij}^{mbs} - \sum_{k=1}^{N_{CCD}} a_k R_{ij}^{mbs}. \tag{2.6}$$

This process is repeated varying the coefficients a_k and recomputing (2.6) until we minimize the variance of the resulting image.

All we have produced up to this point are parts of images, since that is how the data from the CCDs are read. Physically, though, the CCD has no such division. To generate images corresponding to all CCD's pixels, we merge the four parts into a *full SCN image*, also called a `data` image. Part 4 corresponds to the first strip of pixels, and part 1 to the last one. The merged images have 4112×4130 of active area, 150×4130 of overscan and 4262×90 of vertical overscan, totaling 4262×4220 pixels per HDU. If a file has 14 HDUs and its pixel data is encoded as `IEEE-754` single precision floating point numbers, $\sim$ 961 megabytes of storage is needed per image file.

After we have a clean merged image, the hit/event extraction program is run with it as input. The extractor scans the image for pixels with values greater than 4σ, where σ is the readout RMS noise.

A set of connected pixels passing this test configures a candidate event. The program then searches for pixels with values greater than 3σ and that are connected to a candidate's pixels. The final set after the two scans configure an event, and information such as its position, energy and number of pixels are saved to a *catalog* in ROOT [46] `TTree` format. The resulting extraction catalogs may be subsequently processed and analyzed, for example in the search for specific particle hits such as neutrinos.

Chapter 3

Extraction and spectra of events

The first step in the goal of identifying diffusion-limited hits in the CONNIE images is *event extraction*. This procedure searches an image for clusters of pixels that are above a background level, as they might be the product of particle hits in the CCD. The clusters found are called *events* and a *catalog* file with information about them is produced. For this task we use SExtractor [47] [48], a program widely established in the Astrophysics community for the extraction of astronomical objects in images, and summarized in Appendix A. The output of SExtractor is a catalog stored as a FITS table [44], where events are rows and their properties, such as position, are columns. The user has control over which information is computed and recorded for the events, as well as the configuration parameters that control the extraction and computations.

SExtractor has no knowledge about particles, and uses only information that can be derived from the pixel counts in *analog-to-digital units* (ADU). Using the energy in ADU produced by the extraction and set the tools described in Appendix B, we construct a number of *spectra*: histograms of the number of extracted events per energy bin. Each spectrum is produced by a different SExtractor configuration. We thus vary the latter in order to find a set of extraction parameters adequate for the further selection of events and discrimination of diffusion-limited hits.

3.1 Reproduction of the reference spectrum

Although SExtractor is well tested and robust, the contents of astronomical images differ greatly from CONNIE's. In order to understand how our extraction behaves in the CONNIE images, we first try to reproduce the CONNIE spectrum. To obtain it, the collaboration runs their image processing pipeline over an image set, producing a ROOT catalog of extractions for the whole set. From this catalog, a histogram of the number of events per energy bin, a *spectrum*, is created. Figure 3.1 shows the CONNIE (reference) spectrum for the HDU 2 CCD of an image set composed of images 246 to 302.

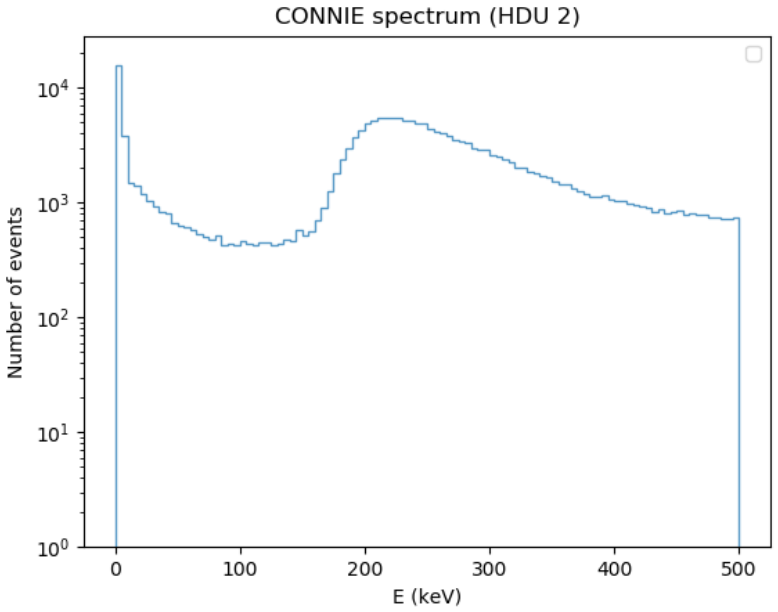

FIGURE 3.1: The CONNIE (reference) spectrum

To reproduce the CONNIE spectrum we used the default configuration in Table 3.1 to run SExtractor on the same image set used by the collaboration. This produced 57 catalogs, each with 14 HDUs, one for each CCD. Then, a *merged catalog* was constructed, also with 14 HDUs. Each HDU i in the latter catalog has the events of the ith HDU in all the 57 catalogs. The catalogs store the `FLUX_AUTO` output parameter,[1] a measurement of an event's energy that sums its pixel values in ADU, and thus should be proportional to the energy deposited by the hit. Using this parameter, we constructed our first spectrum. Figure 3.2 displays both spectra, ours (left) and CONNIE's (right), in ADU. Note that in this study we only use HDU 2, a CCD known to be stable and well behaved, where the characteristic Cu peaks explained in Chapter 2 are most pronounced. Comparing the spectrum from the default configuration to the reference, we see that they are very discrepant in the whole energy range presented. This is not surprising once we recall that the energy of each event is proportional to its ADU counts and the default SExtractor configuration used is optimized for astronomical images.

[1] See Section A.4.

Parameter	Value
DETECT_MINAREA	5
DETECT_THRESH	1.5
DEBLEND_NTHRESH	32
DEBLEND_MINCONT	0.005
FILTER	Y
FILTER_NAME	default.conv
CLEAN	Y
CLEAN_PARAM	1.0

TABLE 3.1: Default SExtractor configuration. See Appendix A for the definition of each parameter.

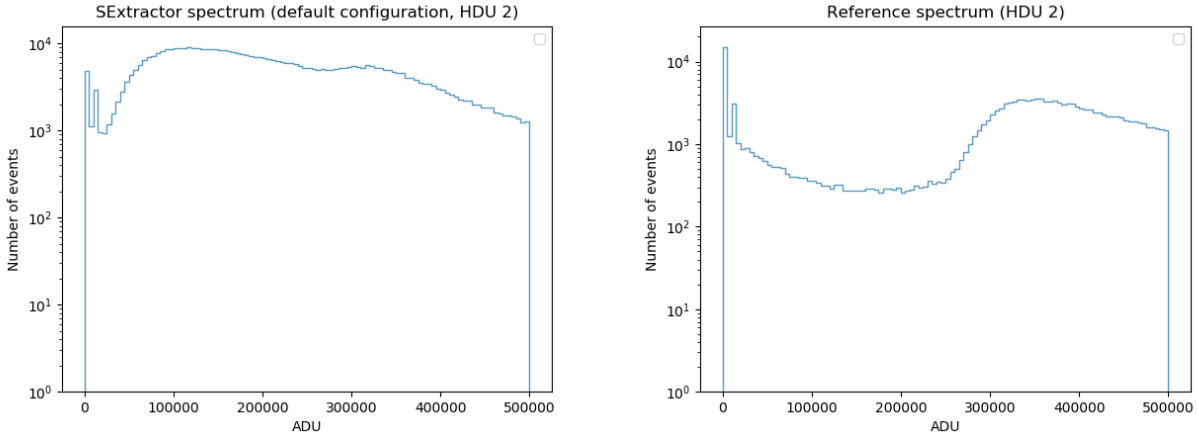

FIGURE 3.2: Comparison of spectra generated from SExtractor with its default configuration (left) and the CONNIE standard extractor (right). Both extractions were performed on the same image set, composed of images 256 to 302.

In the initial versions of CONNIE and DAMIC, before these experiments' pipelines were developed, SExtractor was used for preliminary event extraction. The SExtractor configuration used at that time is presented in table 3.2[2] with only the parameters that differ from Table 3.1 listed. Henceforth we will call it the CONNIE *fiducial* configuration. Comparison between the reference and the spectrum generated using the fiducial configuration (Fig. 3.3) demonstrate enhanced agreement, except for the lowest energies.

[2]Juan Estrada estrada@fnal.gov.

Parameter	Value
DETECT_MINAREA	3
DETECT_THRESH	1.0
DEBLEND_NTHRESH	3
DEBLEND_MINCONT	0.3
FILTER	N

TABLE 3.2: CONNIE fiducial SExtractor configuration, from CONNIE and DAMIC's preliminary extractions.

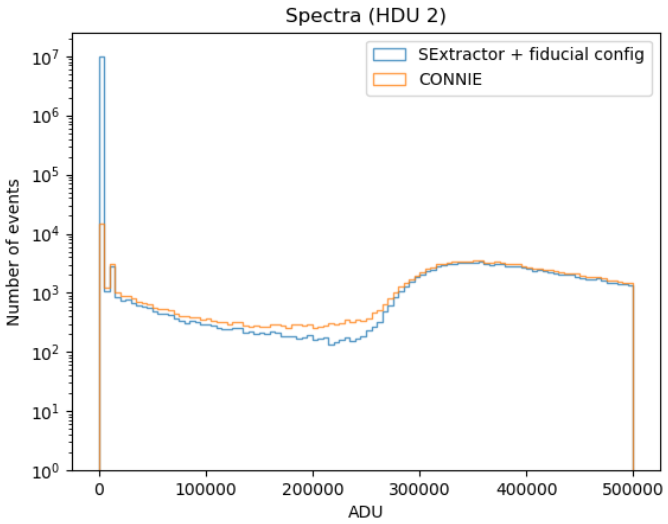

FIGURE 3.3: Reference and fiducial spectra for `data` images (see Section 2.3).

Due to the linearity of CCDs, an event's energy in ADU is proportional to the energy deposited by the hit. By visual inspection of Figure 3.4, we identified the K_α Cu peak, which corresponds to an energy of $E_{\mathrm{Cu}_\alpha} = 8.05$ keV [40], to be at $\mathtt{FLUX\_AUTO}_{\mathrm{Cu}_\alpha} = 12811$. Therefore we used the following calibration in our analysis:

$$E = \frac{\mathtt{FLUX\_AUTO}}{\mathtt{FLUX\_AUTO}_{\mathrm{Cu}_\alpha}} E_{\mathrm{Cu}_\alpha} \implies E = \mathtt{FLUX\_AUTO} \times 6.28 \times 10^{-4} \text{ keV}. \tag{3.1}$$

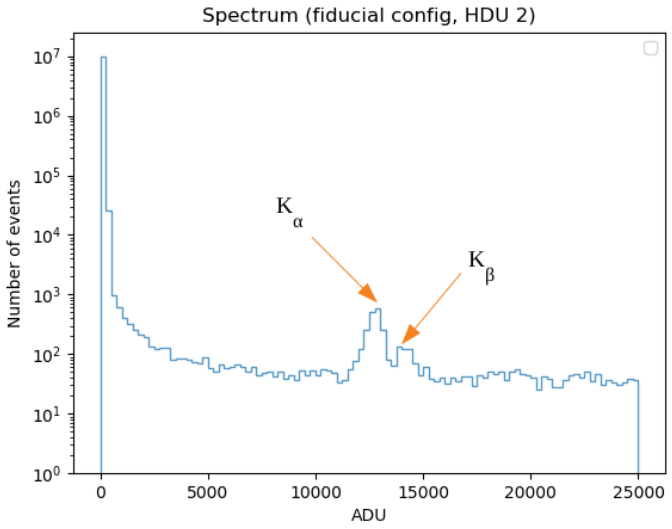

FIGURE 3.4: Spectrum in ADU from SExtractor with the fiducial configuration, showing the Cu K_α and K_β peaks.

The CONNIE catalogs provide their own calibration, represented by the `gainCu` variable in the catalogs. Using both calibrations, we constructed the reference and fiducial spectra in Figure 3.5. The only significant difference to Figure 3.3 is in the units of energy, from ADU to keV. Although the collaboration's calibration is more precise and accurate than ours, the latter was considered to be sufficient for the level of detail in our analysis.

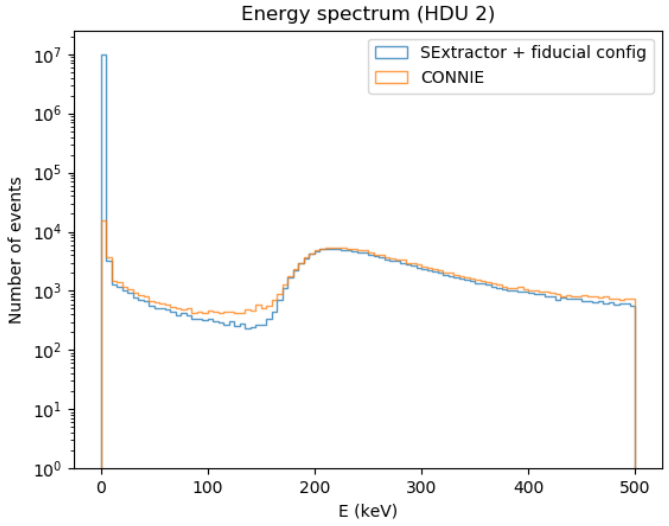

FIGURE 3.5: Reference and fiducial spectra for `data` images (see Section 2.3).

3.2 Simulated neutrinos

In order to study the efficiency of their extractor and selection criteria for neutrinos, the CONNIE collaboration simulated diffusion-limited nuclear recoil hits. The positions are uniformly distributed over the CCD volume, and the energies uniformly distributed over the range 28 eV $< E <$ 5 keV. Each hit is modeled as a two-dimensional Gaussian centered at position (x, y), with width σ determined by the hit's depth z, *i.e.* the distance to the charge collection region. A sample of 500 hits is simulated for each of the four SCN image parts. The parts are then merged, resulting in the so called `draw` images, containing 2000 hits—in contrast to the $\lesssim 2$ events expected for the three hours of exposure time [16]. Figure 3.6 displays reference spectra for both image types.

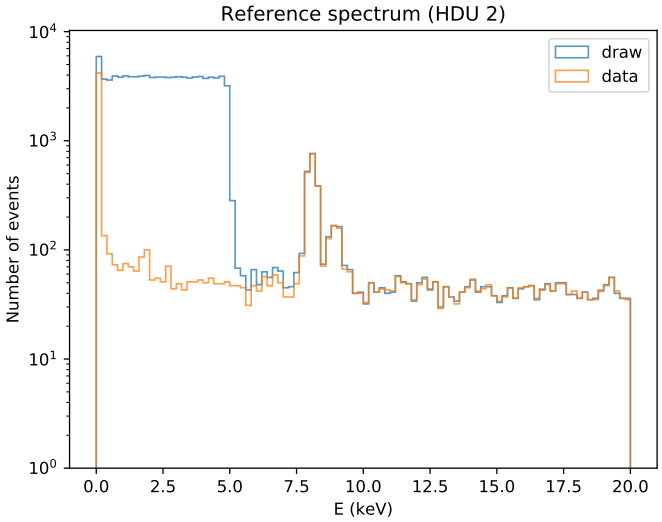

FIGURE 3.6: Reference spectra for images with (`draw`) and without (`data`) simulated hits in the energy range 28 eV $< E <$ 5 keV.

In Figure 3.7 we show the resulting spectrum from the SExtractor runs on the `draw` images using our fiducial configuration. The $E <$ 500 keV plot is used to assess the general features of the spectrum, such as the muon bump around 200 keV [37]; $E <$ 20 keV exhibits the two Cu peaks and the neutrinos; $E <$ 5 keV focuses on the simulated neutrinos, showing the expected plateau arising from the uniformity of the energy distribution of neutrinos; as shown in [16], $E <$ 300 eV is the energy range of the great majority of neutrinos arriving at the CONNIE detector. From the figure, we see that SExtractor was able to reproduce the main features of the reference spectrum down to ∼300 eV. For lower energies, though, a large departure from the reference is clear. In that range the signal to noise ratio is low, and SExtractor yields a peak of events composed predominantly of noise.

Since most neutrinos are expected to have $E <$ 300 eV, and due to the low threshold of the experiment's CCDs (see Chapter 2), neutrino detection is limited by the performance of event extraction and selection at low energies, *i.e.* by the capacity of extracting and discriminating events of interest.

Therefore, it is crucial that we optimize our SExtractor configuration for the lowest energies and find selection criteria capable of discerning neutrinos from noise.

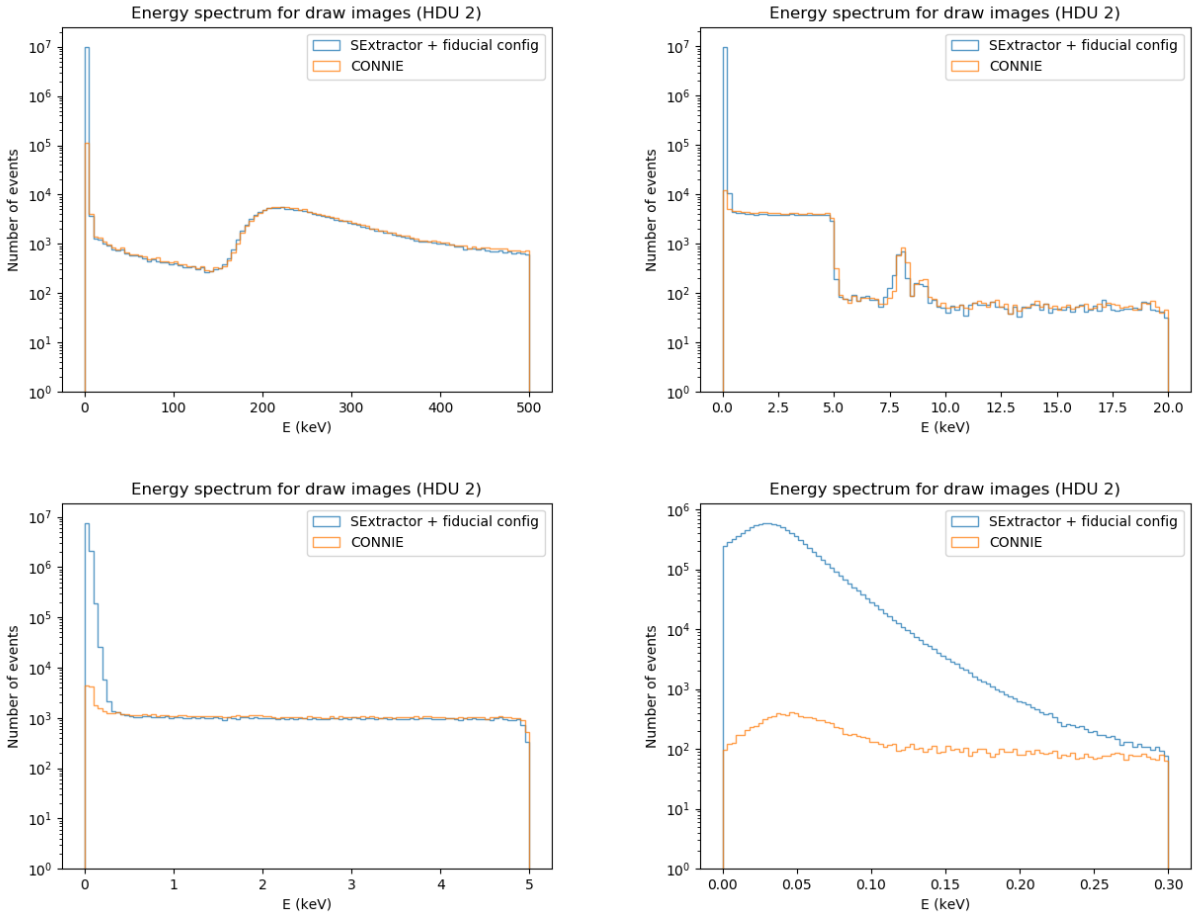

FIGURE 3.7: Spectra for `draw` images obtained by CONNIE and SExtractor. The meaning of the various ranges and the features of the plot are explained in the text.

3.3 Reproduction of the reference spectrum for low energies

Due to the uniform energy distribution of the simulated neutrinos, if we had an ideal extraction efficiency in the range 28 eV $< E <$ 5 keV, the plateau in Figure 3.7 would extend down to $E = 28$ eV. The plateau value was computed as the sum of the number of events extracted by SExtractor with the fiducial configuration in $1 < E < 4$ keV, divided by the energy range of 3 keV, resulting in $C = 20/\text{eV}$. Using this value, we can quantify the deviation from the perfect detection efficiency

using a reduced χ^2 with a Poisson error

$$\chi^2 = \frac{1}{N_{bin} - 1} \left[\sum_{i=1}^{N_{bin}} \frac{(N_i - C\Delta E)^2}{N_i} \right], \tag{3.2}$$

where N_i are the measured counts in each bin, ΔE the size of the bin in eV, and N_{bin} the number of bins. Large deviations from 1 in χ^2 mean either large incompleteness or contamination.

In order to try increasing this efficiency for 28 eV $< E <$ 300 eV, we ran SExtractor over the same set of `draw` images as in the previous section, changing its main detection parameters. Starting from the fiducial configuration, we varied the parameters one at a time, and analyzed their influence on the resulting spectra using both the metric 3.2 and visual inspection. For the latter, if the number of extractions is lower than the plateau, we *know* that simulated events are not being detected. On the other hand, having more events than the plateau may be an overestimation: background or noise may have been extracted as events when they should not. Since we will later apply criteria to select specific events, possibly excluding invalid extractions, we consider a number of events above the plateau to be better than one below it.

The metric was calculated twice for each configuration, corresponding to the $E <$ 300 eV and $1 < E <$ 4 keV energy ranges. The dashed blue lines in Figure 3.8 denote these regions, the gray arrows show the ranges from where the constant neutrino spectrum is determined and where the χ^2 at low energies is computed, and the dashed gray line represents the plateau.

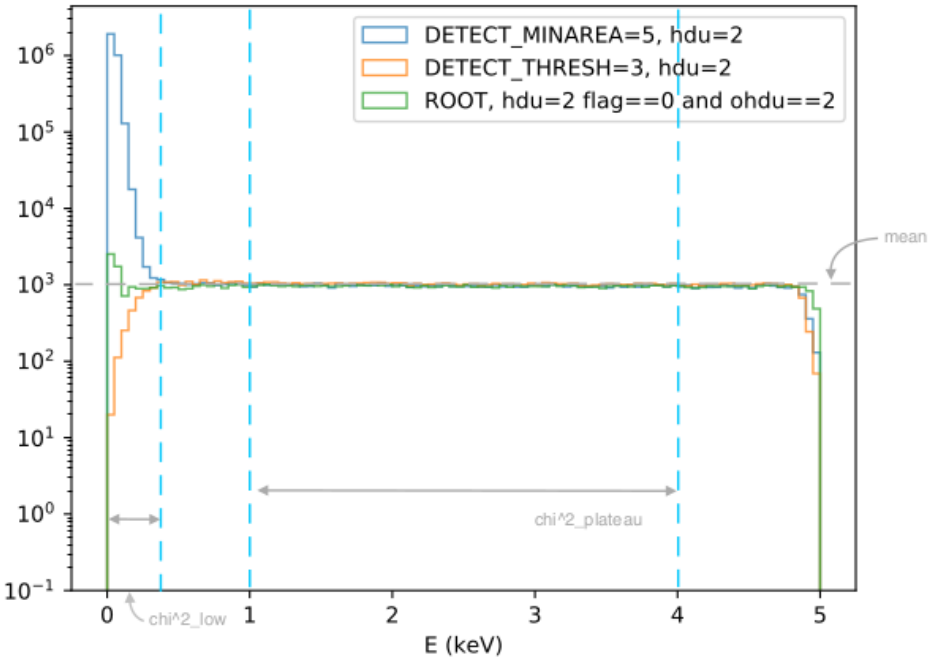

FIGURE 3.8: Spectrum at low energies in the draw images. The gray dashed line shows the constant number of simulated neutrinos per energy interval, which was determined from `draw` data. It also shows the energy range where the χ^2 is evaluated to quantify the deviations from a constant plateau at low energies.

The `FLAGS` output parameter from SExtractor tags an event with warning conditions encountered in its extraction, *e.g.* the occurrence of one or more saturated pixels. It is an 8-bit number, where zero means the extraction was completely normal. Figure 3.9 shows that the most restrictive value decreases the counted events. That is specially true for the lowest energy events, where the signal-to-noise ratio is small. On a similar note, the cleaning of spurious detections (see Section A.3), enabled by the `CLEAN` configuration option, also decreases the number of events at the lowest energies (Fig. 3.10), but not below the plateau, *i.e.* it does not get rid of the simulated neutrinos. On the other hand, after ∼100 eV the counts are increased. From these results alone, we cannot conclude whether the decrease and increase are due to losing hits or not. As performing the cleaning is the default behavior in SExtractor, it was kept in our analysis as well. In the future, a systematic study of the effects of cleaning and the variations of its `CLEAN_PARAM` parameter should be done.

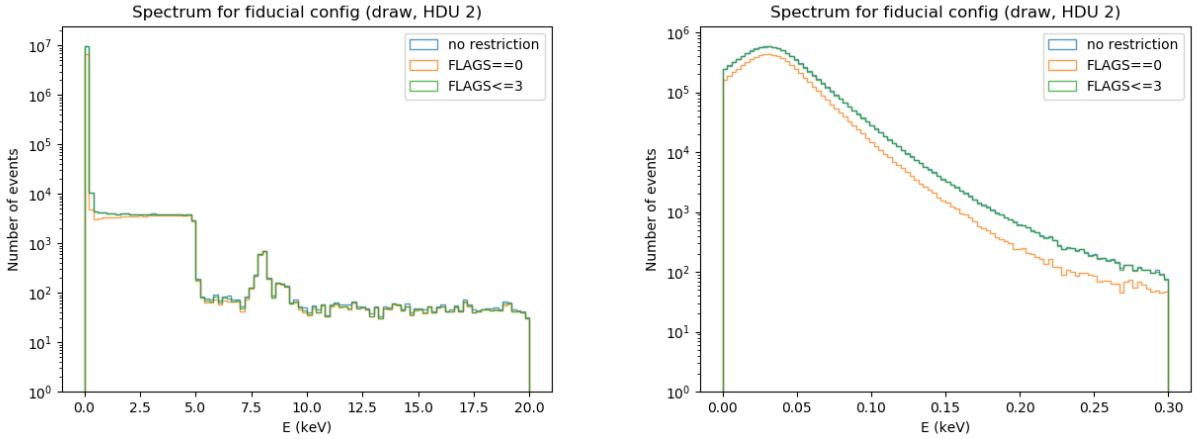

FIGURE 3.9: Energy spectrum for different cuts of the `FLAGS` parameter. Note that the curve with `FLAGS` = 0 is below the plateau, while the cut in `FLAGS` <= 3 preserves the plateau down to lower energies.

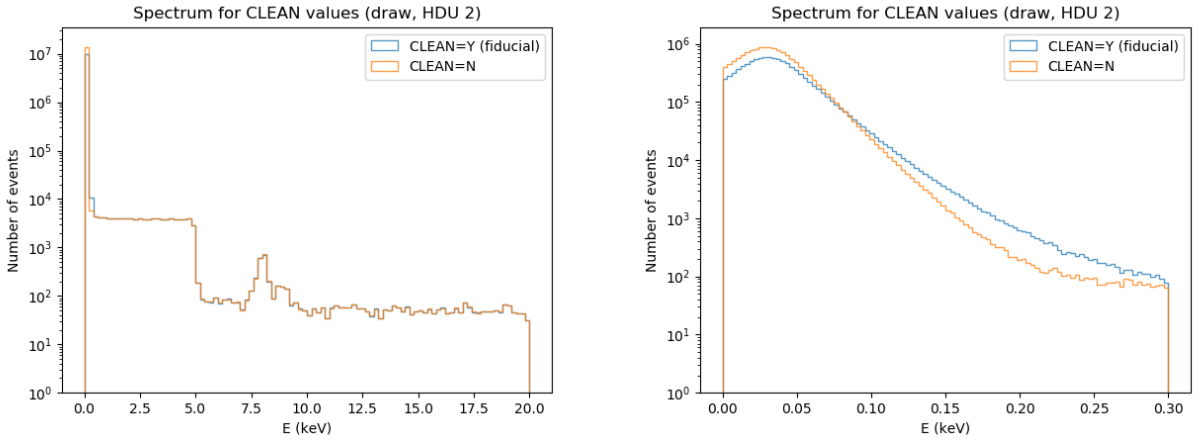

FIGURE 3.10: Energy spectrum with and without cleaning of spurious detections.

When compared to the fiducial configuration, the range of values explored for the deblending options, `DEBLEND_MINCONT` and `DEBLEND_NTHRESH` (see Section A.3), exert no significant influence on the spectrum (figures 3.11 and 3.12). Therefore the fiducial values were not changed.

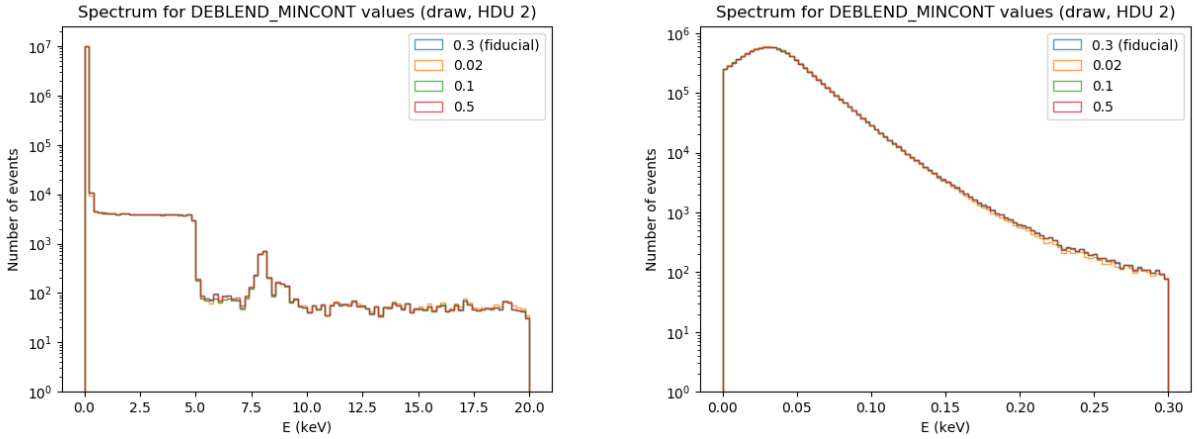

FIGURE 3.11: Energy spectrum for variations of the `DEBLEND_MINCONT` option.

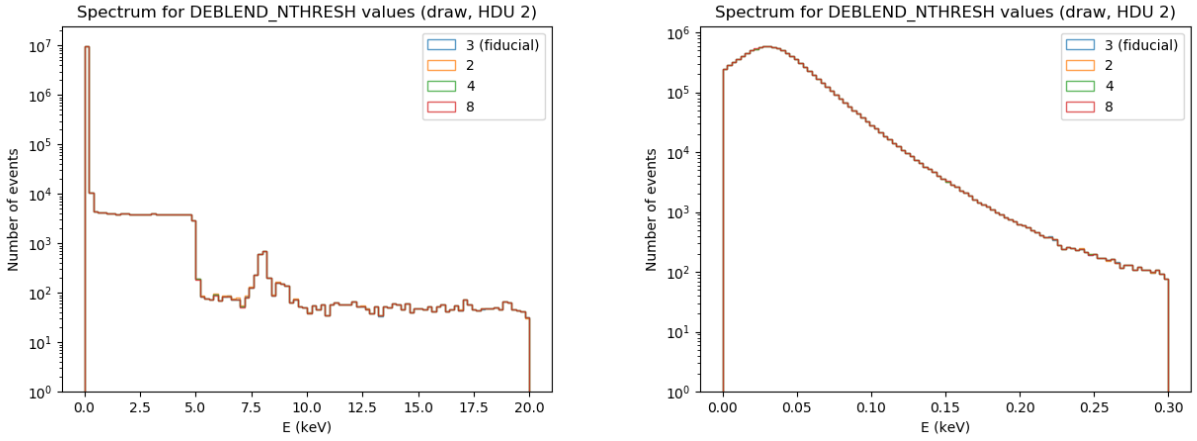

FIGURE 3.12: Energy spectrum for variations of the `DEBLEND_NTHRESH` option.

The application of SExtractor's default pyramidal filter [47] (Fig. 3.13) significantly affects the spectrum in some regions. For energies below 250 eV, applying the filter decreases the number of extracted events. Furthermore, for $E \lesssim 50$ eV the filtering discards valid hits from the simulation, and hence will not be used.

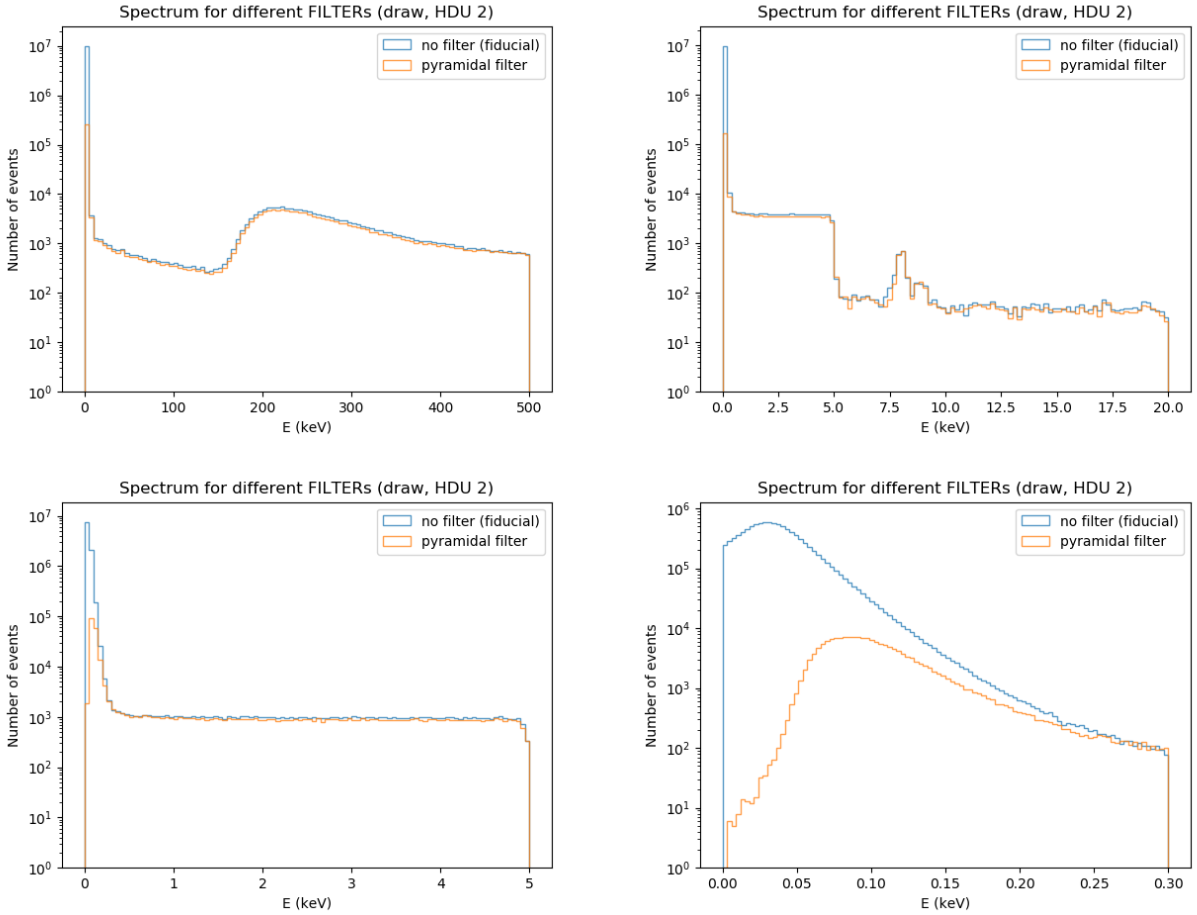

FIGURE 3.13: Energy spectrum with (orange) and without (blue) SExtractor's pyramidal filter.

Figure 3.14 shows that increasing `DETECT_MINAREA`, the minimum number of pixels above the background in an event, decreases the number of low energy events, an effect stronger for energies below 100 eV. However, from the diffusion of charge in the CCD (see Chapter 2), we would expect an increase in `DETECT_MINAREA` to decrease the number of hits closer to the charge collection region, independently of energy. Also in the low energy range, the number of standard deviations above the background RMS noise, `DETECT_THRESH`, is the most influential option of the fiducial configuration. Figure 3.15 points that $2 \leq$ `DETECT_THRESH` ≤ 3 marks a clear break in the extraction, from detecting noise/background as events to discarding valid hits.

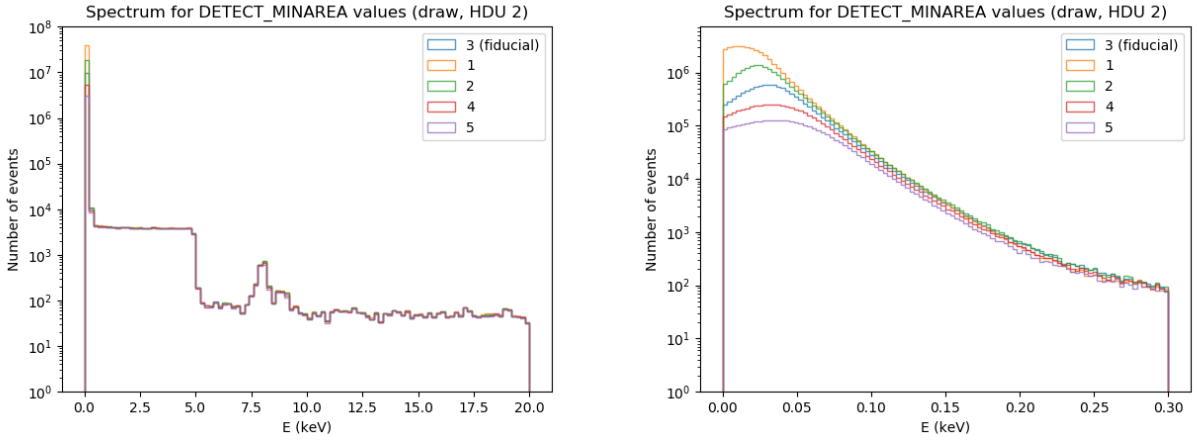

FIGURE 3.14: Energy spectrum for variations of the `DETECT_MINAREA` option.

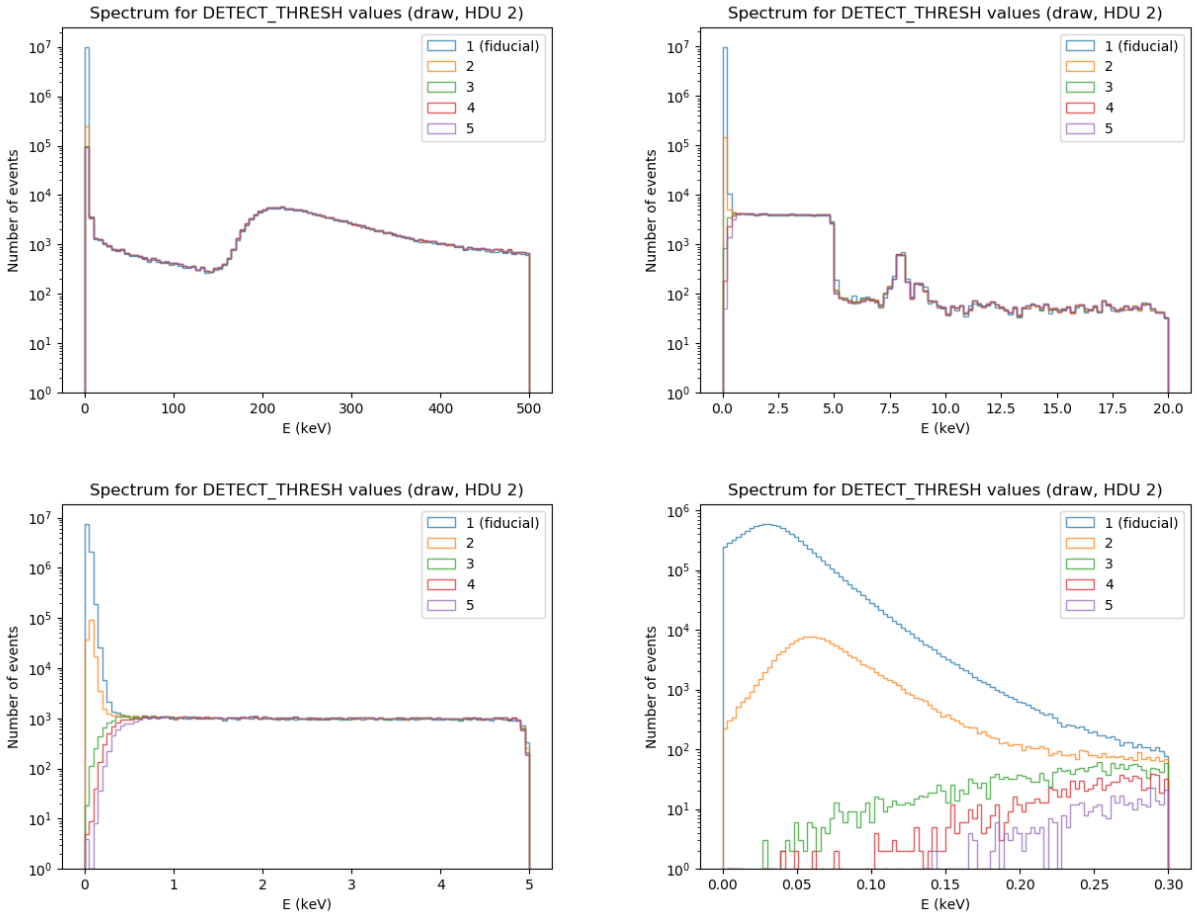

FIGURE 3.15: Energy spectrum for variations of the DETECT_THRESH parameter.

In addition to the qualitative analysis of the signal, for each variation from the fiducial value we applied metric 3.2 to events in the ranges $1 < E < 4$ keV and $28 < E < 300$ eV. Table 3.3 summarizes these results. In agreement with the previous analysis, `DETECT_THRESH` is the most influential variable for the lowest energies. Although in that energy range the spectrum for `DETECT_THRESH` $= 3$ is the one closest to the plateau, it discards valid hits and so decreases the completeness. Therefore, `DETECT_THRESH` $= 2$ is a better choice for our purpose of higher efficiency, and as such is the only value changed from the fiducial configuration discussed in Section 3.1. From the analysis undertaken in this chapter, the set of values optimized for low energy extraction are summarized in Table 3.4, and are henceforward called the `phase 1` configuration. Figure 3.16 summarizes the results of this chapter. The spectrum obtained from SExtractor and from CONNIE are consistent with the one from a reactor with added simulated neutrinos (see [37]). Also, the two pipelines are compatible for 300 eV $< E <$ 20 keV. However, for lower energies CONNIE extracts less noise and preserves the plateau down to $E \gtrsim 100$ eV, while our pipeline is flooded with spurious detections in $E \lesssim 200$ eV.

Parameter	$1 < E < 4$ keV	$28 < E < 300$ eV
`CLEAN` $=$ `N`	1.93	134984
`FILTER` $=$ `default`	18.47	2771
`DEBLEND_NTHRESH` $= 2$	2.13	102444
`DEBLEND_NTHRESH` $= 4$	1.96	102485
`DEBLEND_NTHRESH` $= 8$	1.90	102502
`DEBLEND_MINCONT` $= 0.5$	1.61	105252
`DEBLEND_MINCONT` $= 0.1$	1.73	102962
`DEBLEND_MINCONT` $= 0.02$	2.16	102317
`DETECT_MINAREA` $= 1$	1.65	199794
`DETECT_MINAREA` $= 2$	1.78	153555
`DETECT_MINAREA` $= 4$	2.34	58252
`DETECT_MINAREA` $= 5$	3.21	35225
`DETECT_THRESH` $= 2.0$	0.88	2296
`DETECT_THRESH` $= 3.0$	0.87	552
`DETECT_THRESH` $= 4.0$	0.95	1091
`DETECT_THRESH` $= 5.0$	0.96	1801

TABLE 3.3: Reduced χ^2 for each SExtractor configuration variation from the fiducial value.

Parameter	Value
DETECT_MINAREA	3
DETECT_THRESH	2.0
DEBLEND_NTHRESH	3
DEBLEND_MINCONT	0.3
FILTER	N

TABLE 3.4: SExtractor configuration values optimized for extraction at $E < 5$ keV, called the `phase 1` configuration. `DETECT_THRESH` is the only value changed from the fiducial configuration (Table 3.2).

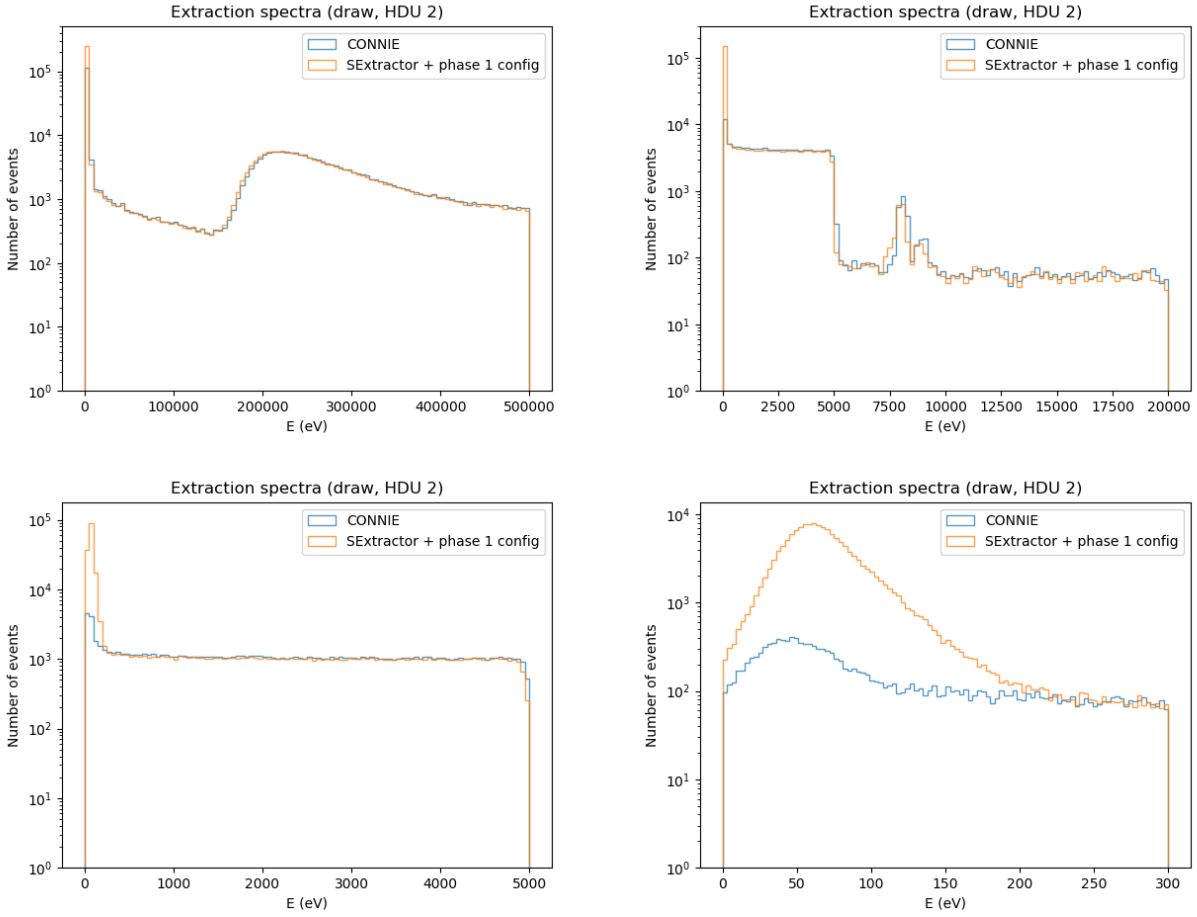

FIGURE 3.16: Comparison of spectra generated from the extractions by CONNIE and SExtractor with `phase 1` configuration.

Chapter 4

Analysis of extraction efficiency

In this chapter we are concerned with the efficiency of SExtractor in detecting simulated neutrinos. Specifically, for a given extraction we expect to address two questions: Which fraction of the simulated neutrinos are recovered from the extraction in a given energy bin? Which fraction of events *do not* correspond to simulated neutrinos in a given energy bin?

4.1 Simulation catalogs

The program for the simulation explained in Section 3.2 takes an image part (see Section 2.3), and produces both an image part with neutrino hits and a table in ROOT TTree format. A table entry corresponds to a hit, and contains hit information such as position, energy, charge and width of the Gaussian describing the event's signal. We then take four tables corresponding to one SCN image, convert each to a FITS table, and use the transformation explained in Section 2.3 to merge them into a single *simulation catalog* C_{sim}.

As detailed in Chapter 3, the `draw` images are processed by SExtractor, each producing one *extraction catalog*. The events with energy outside $28 < E < 5000$ eV are removed, creating a filtered extraction catalog C_{ext}. Using all C_{ext} and C_{sim} for an image set, we run `fitsposmatch`, a program we developed to match the positions of extracted events to simulated hits. For each simulated neutrino s in C_{sim}, we calculate its euclidean distance to every event in C_{ext}. As discussed in Section 2.3, the SCN image does not have the prescan region, hence its pixels start at position $x = 8$ in the CCD. The simulation program uses that information, and thus the simulated neutrino positions also start at $x = 8$. On the other hand, SExtractor is unaware of that property, and the extracted events start at $x = 0$. Therefore, in positional matching all extracted event positions must be translated as $x'_{ext} = x_{ext} + 8$. Then, if an event's distance is less than a configurable parameter d_{min}, it is flagged as a match candidate. After all events are looked up, the candidate with the minimum distance

is set as the *matched event* for s. The matched events for all s are then saved in a FITS *match catalog*, in which an entry contains the whole information from both the event's extraction and the hit simulation.

4.2 Completeness and contamination

From an extraction and its match catalog, we have the number of simulated neutrinos N_{sim}, extracted events N_{ext}, and matched events N_{match}. With these we compute the *completeness*, *i.e.* the fraction p of simulated neutrinos recovered by the extraction, and the *contamination*, *i.e.* the fraction q of extracted events that do not correspond to neutrinos:

$$p = \frac{N_{match}}{N_{sim}}, \tag{4.1}$$

$$q = \frac{N_{ext} - N_{match}}{N_{ext}}. \tag{4.2}$$

These quantities are computed in energy bins, with and without applying cuts on the several output parameters, aiming to increase the completeness (Chapter 4) or decrease the contamination (Chapter 5). To assign uncertainties to p and q in each energy bin, we consider Poisson errors in N_{match} and N_{ext}[1], and propagate them in the standard way. For completeness

$$\sigma_p = \frac{\sqrt{N_{match}}}{N_{sim}}, \tag{4.3}$$

and for contamination

$$\sigma_q = |q| \sqrt{\left(\frac{\sigma_{N_{ext}-N_{match}}}{N_{ext} - N_{match}}\right)^2 + \left(\frac{\sigma_{N_{ext}}}{N_{ext}}\right)^2} = |q| \sqrt{\frac{N_{ext}}{(N_{ext} - N_{match})^2} + \frac{1}{N_{ext}}}. \tag{4.4}$$

These quantities answer our two questions about an extraction. However, as seen in Chapter 3 and specifically in Figure 3.8, for the lowest energies the efficiency strongly depends on energy. Therefore, we divided the energy range in bins and calculated the completeness and contamination in each bin. As in the generation of spectra, to compute p and q we used a single CCD (HDU 2) and the image set composed of images 256 to 302. Each simulation adds 2000 neutrinos uniformly distributed in energy, so for an image set of 57 images the number of simulated neutrinos per unit energy is 22.93 eV^{-1}; this number multiplied by the width of the energy bin is the N_{sim} used in practice for calculating p.

In order to reduce the number of false positives, *i.e.* noise extracted as neutrinos, we applied one more procedure. For a given configuration, we ran the extraction and matching over both the `draw` and

[1] We do not consider errors in N_{sim} as their distribution in energy is homogeneous.

`data` images. If $N_{match}^{\texttt{draw}}$ and $N_{match}^{\texttt{data}}$ are their respective number of matches, the total number used in (4.2) for a given configuration/extraction is $N_{match} = N_{match}^{\texttt{draw}} - N_{match}^{\texttt{data}}$. In other words, if there are events in the `data` – with no neutrinos added – images matching with the position of simulated events, these are spurious detections and are not counted in N_{match}. Figure 4.1 shows the effect of using the `data` match catalog to the completeness and contamination of an extraction using the `phase 1` configuration and matching with $d_{min} = 2$. The first two bins have more matches in the `data` images, 16 and 31, than in the `draw` images, 6 and 23, leading to a negative efficiency/completeness for up to 70 eV. Consequently, the contamination for this same range is maximum. These results show that the configurations used for extraction and matching are not able to discern neutrinos from noise in the range $28 < E < 70$ eV.

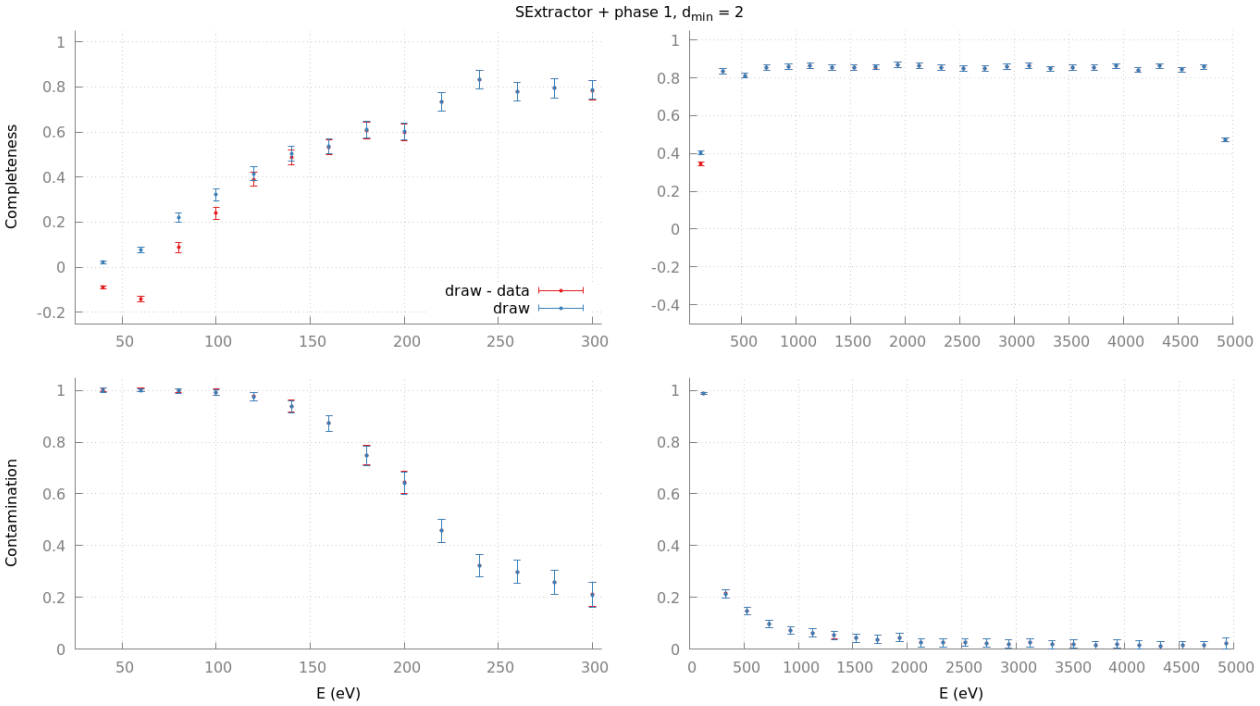

FIGURE 4.1: The effect of cleaning the matches in the images with added neutrinos using a `data` match catalog, for an extraction using the `phase 1` configuration. In blue, only $N_{match}^{\texttt{draw}}$ is considered in the calculations; in red, we use both $N_{match}^{\texttt{draw}}$ and $N_{match}^{\texttt{data}}$.

The above mentioned procedure also helped in choosing d_{min}. Using an extraction with the `phase 1` configuration, and another from the CONNIE extractor, we set d_{min} to 1, 2 and 3 pixels, and computed p and q for each variation in each extraction. As displayed in figures 4.2 and 4.3, $d_{min} = 1$ results in the best efficiency and contamination for SExtractor, and the worse of both for the CONNIE extractor. This suggests the latter fails to determine the centroids of the events with good accuracy, which makes the positional matching unable to find many of the events in the catalogs. On the other hand, $d_{min} = 2$ introduces many spurious detections in our extraction, but not in CONNIE. Since the effect for $d_{min} = 2$ is not as bad in SExtractor as $d_{min} = 1$ is in CONNIE, we choose $d_{min} = 2$ for both. In the future, the analysis should be repeated for CONNIE with $d_{min} = 2$ and SExtractor with $d_{min} = 1$ to compare the best approaches for each extractor.

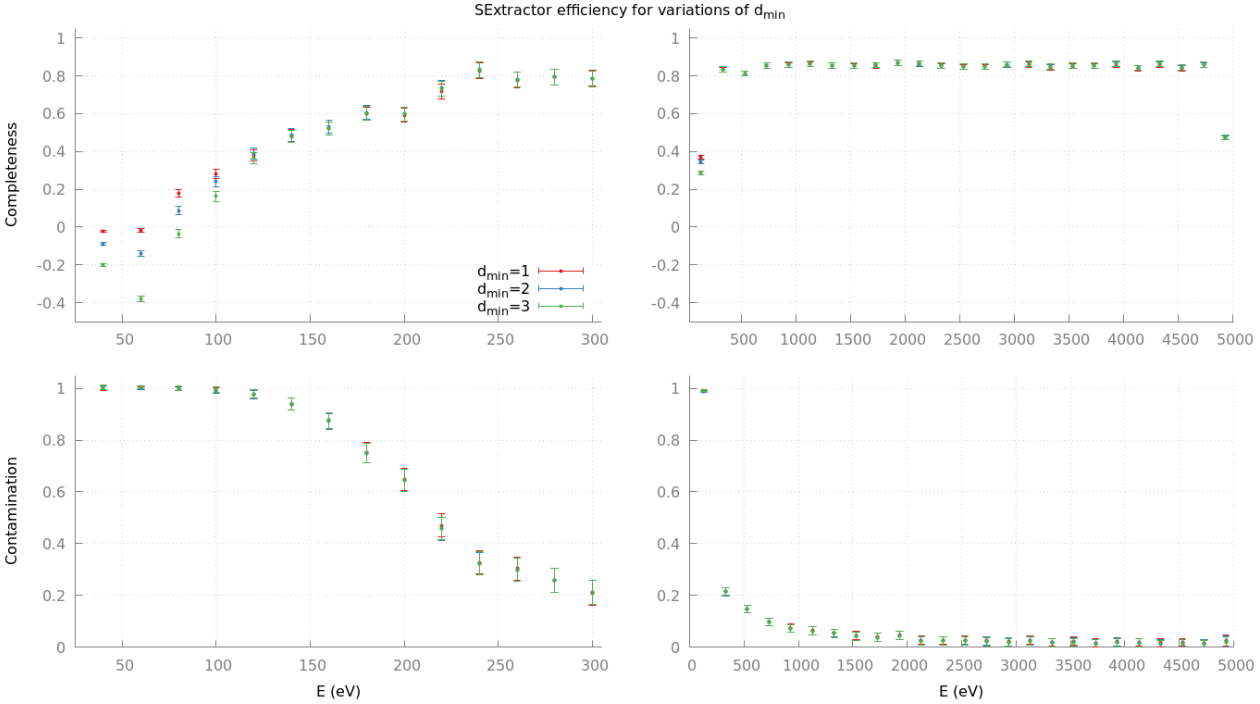

FIGURE 4.2: Completeness and contamination for variations of d_{min} in matching an extraction using `phase 1` configuration.

In brief, for each extraction the following steps are executed:

1. Run the extraction for the `data` image set.
2. Run the extraction for the `draw` image set.
3. Since the simulated neutrino energies are in the range $28 < E < 5000$ eV, remove all events not in this range from the `draw` and `data` catalogs. The resulting catalogs are taken as the new `draw` and `data` catalogs.
4. Merge all the `draw` catalogs of an image set into one.
5. With $d_{min} = 2$, match the `data` catalogs to the respective simulation catalogs, creating a match `data` catalog for the image set.
6. With $d_{min} = 2$, match the `draw` catalogs to the respective simulation catalogs, creating a match `draw` catalog for the image set.
7. Using the merged `draw` catalog, the `draw` match catalog, and the `data` match catalog, compute p and q for each bin in the energy range.

In order to increase the completeness of the extractions, we used the `phase 1` configuration as base and varied the `DETECT_THRESH` and `DETECT_MINAREA` parameters. The results in Figure 4.4 show, as expected, that increasing the detection threshold decreases the completeness and contamination

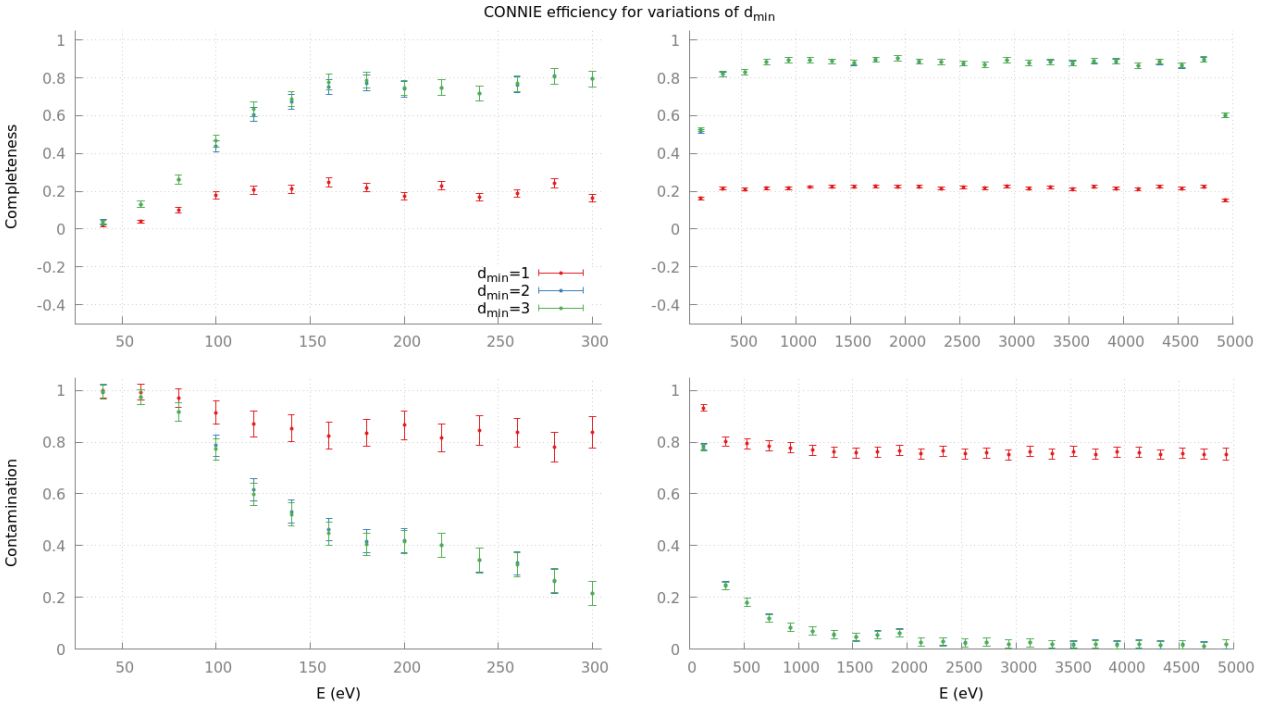

FIGURE 4.3: Completeness and contamination for variations of d_{min} in matching a CONNIE extraction.

altogether. The `DETECT_THRESH = 1.5` plot shows once more the importance of using the `data` match catalog: although in general its completeness is increased in relation to the other configurations, from 30 to 70 eV we can see the extraction is heavily dominated by false positives. Since the random nature of the noise prevents it from clustering into more than a few pixels, decreasing the minimum area for an object also has the effect of increasing contamination, as pictured in Figure 4.5. The variations from `phase 1` could not improve the detection of neutrinos without bringing together a significant amount of false positives. Thus, in order to proceed in the investigation with a balance between completeness and contamination, we choose to keep using `phase 1` as the configuration for detection with SExtractor. Figure 4.6 presents a comparison between the efficiency and contamination of that setup and the CONNIE extractor.

In the current chapter we implemented a method for computing the completeness and contamination for an extraction, asserted the method's robustness, and used it to choose a SExtractor configuration. We have shown that, at least prior to the application of any selection of events, the detections from the CONNIE pipeline are more complete and less contaminated than those obtained with SExtractor. In the next chapter we apply such procedure to find and analyze event selection criteria, with the goal of improving the detection of diffusion-limited hits.

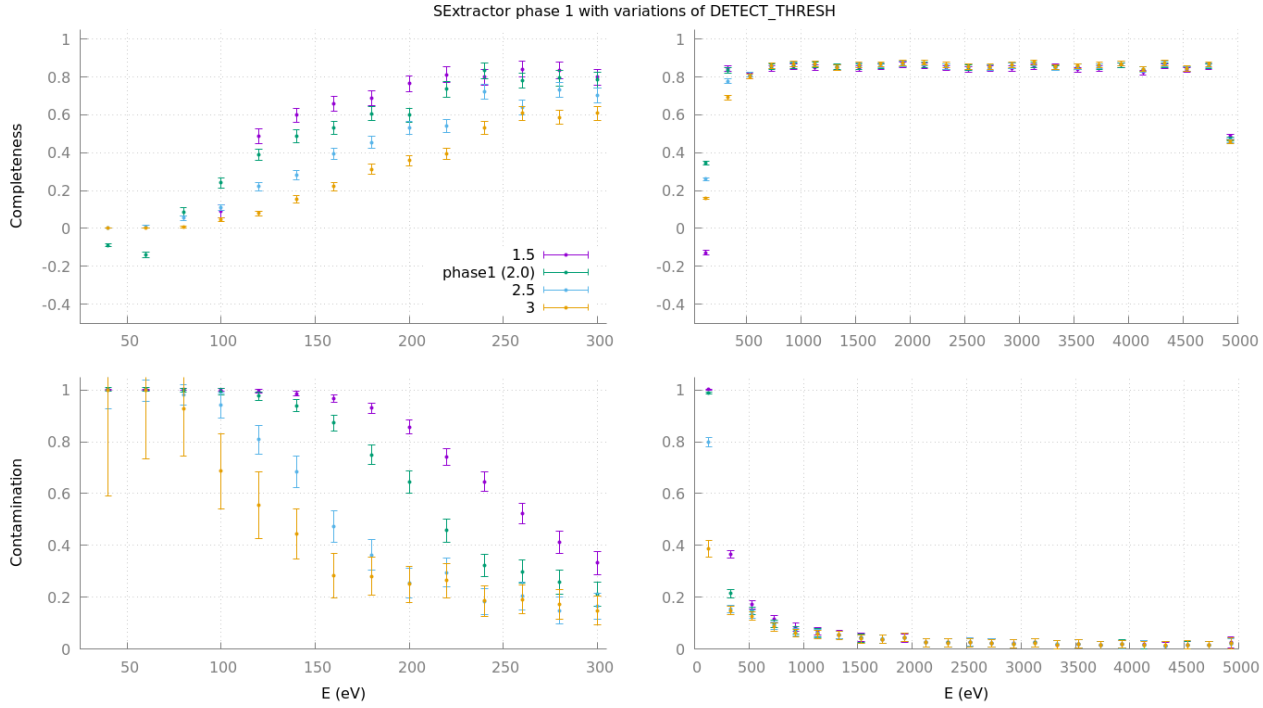

FIGURE 4.4: Completeness and contamination for variations of `DETECT_THRESH` using the `phase 1` configuration as base.

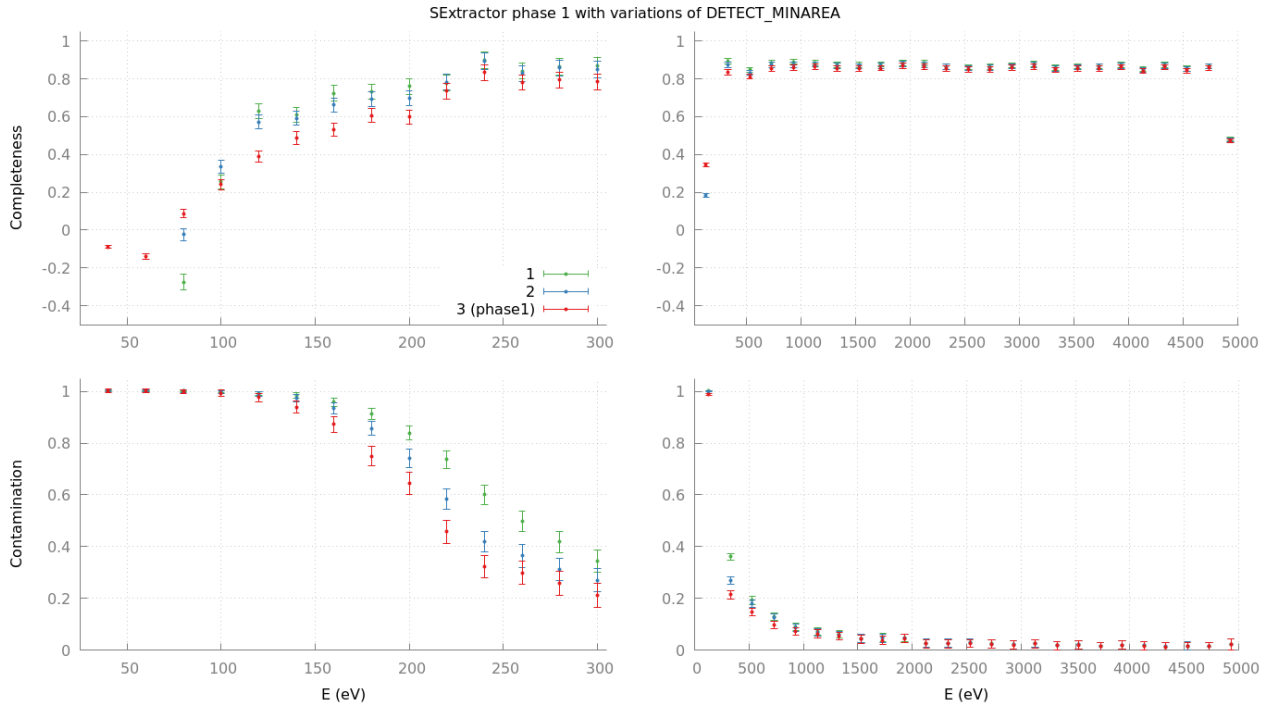

FIGURE 4.5: Completeness and contamination for variations of `DETECT_MINAREA` using the `phase 1` configuration as base.

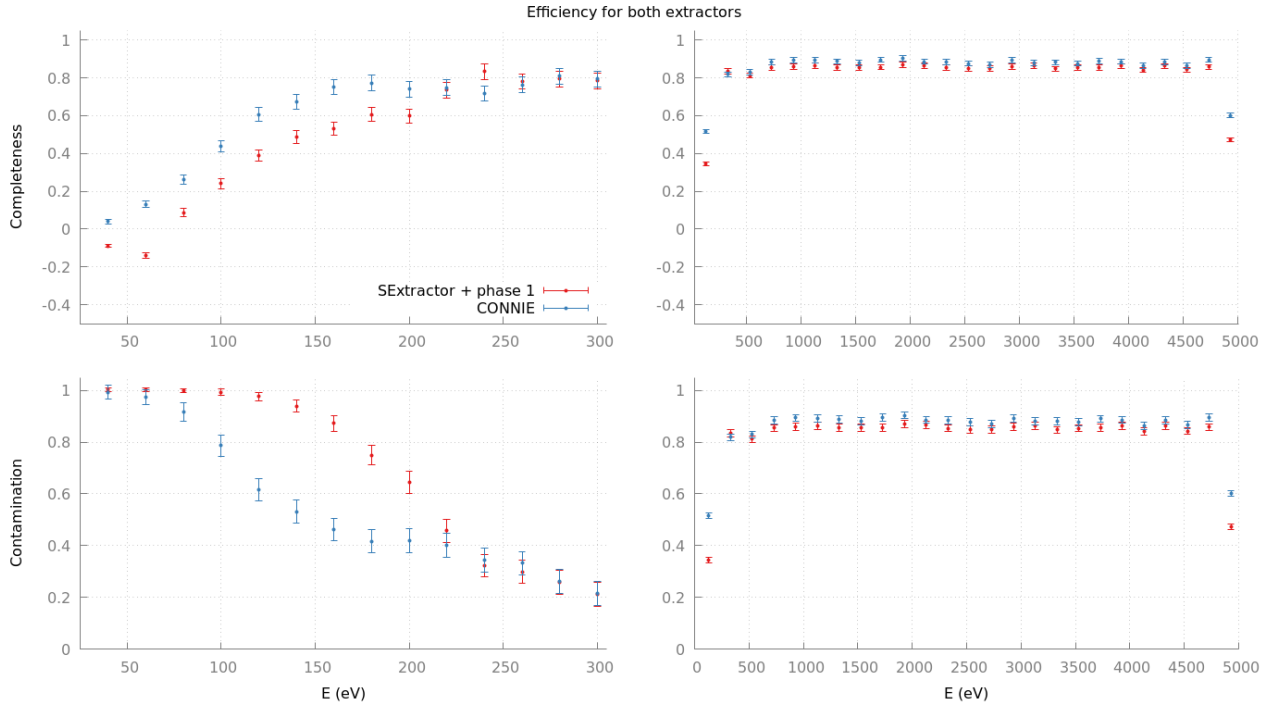

FIGURE 4.6: Comparison of p and q values for the `phase 1` SExtractor configuration and the CONNIE extractor.

Chapter 5

Event morphology and selection of diffusion-limited hits

In the previous two chapters we used SExtractor for event extraction and assessed the effect of varying the detection parameters in Table 3.4 on the completeness and contamination of the neutrino sample using images containing simulated neutrinos. In that exploratory study we found out that at the lowest energies we are limited by a low detection efficiency and high contamination rate. In this chapter we explore several cuts on SExtractor's output parameters, aiming at decreasing the contamination. So far our analyses used only three event quantities measured by SExtractor: the energy in ADU, expressed in the `FLUX_AUTO` parameter and related to the energy in eV by the calibration (3.1); the barycenter, given by `X_IMAGE` and `Y_IMAGE`; and the `FLAGS`. None of these parameters alone or in combination are expected to be able to discriminate diffusion-limited hits. As discussed in chapters 2 and 3, this is the type of event corresponding to the neutrinos arriving at the CONNIE detector. Their shape can be modeled by a two-dimensional Gaussian distribution due to the diffusion of charge along their trajectory from the interaction point to the charge collection region. Therefore, if we measure the shape of events, the associated parameters might be useful for selecting the hits of interest. For this sake we used SExtractor's morphological parameters and model-fitting features (see Section A.5) to try and find event selection criteria capable of decreasing contamination without throwing away neutrino hits, *i.e.* preserving the efficiency.

5.1 Changes to SExtractor

As SExtractor was created for the processing of astronomical images, its model-fitting procedure requires that one of its inputs be a file representing a point spread function (PSF): a description of the output signal generated by a point source. The signal of the input image is then convolved with

the PSF to produce the output signal to which measurements are performed. In our case, however, the photons are generated inside the CCD proper and a PSF has no physical sense.

To overcome this limitation, one possibility is to generate a PSF describing a Dirac delta at each point, so that the end result of the convolution would be the same as performing no convolution, that is, having no PSF at all. The delta description could be generated by PSFex [49], and an appropriate choice of its configuration parameters, or by writing a separate generation program. With respect to the human effort, both solutions would require that we understand how to represent the delta in the file and take the time to generate it. In addition to that, even though the end result should not be affected by our delta PSF, the computationally intense convolution would still have to be performed.

The other possible solution, suggested by E. Bertin and chosen by us, was to modify SExtractor to skip the convolution with the PSF if the latter is not provided. This prevents unneeded computation and provided an opportunity to get acquainted with the source code for an essential piece of external software in our pipeline. We tried to keep the changes to the bare minimum so as to not interfere in the processing more than what was absolutely needed. Also, the changes were conditioned to execute only when no PSF file is provided. Namely, we modified SExtractor to:

1. Stop reading a default PSF file. Although this is fine for our purposes, from the user experience perspective it needs to be better engineered if it is to be merged in the public SExtractor repository.
2. Use a model sampling step of 1.0 instead of relying on the PSF sampling step.
3. Zero the `psf_fwhm` variable, for its value is summed in the computation of each object's dimension.
4. Do not compute a local PSF.
5. Do not use the PSF FWHM on the best guess for the modeled sources half-light radius.
6. Do not convolve with the PSF.

Albeit these modifications amount to less than 20 lines of code, we needed to ensure they didn't impact negatively in the model-fitting results. For that, we used SExtractor to process images with known objects and analyzed the fitted parameters. In order to generate the test images we wrote `expdisks`, a program that simulates objects in which pixel values are given by an exponential disk profile [47]. This model is supported by SExtractor fitting and is defined as

$$I(\boldsymbol{r}) = I_0 \exp\left(-\frac{r}{r_0}\right), \tag{5.1}$$

where r is the distance to the object's center, I_0 the intensity at $r = 0$, and r_0 the disk's scale factor. Given these parameters and a background value as input, `expdisks` uses (5.1) to compute the pixel intensities in the image. After all objects are added, Poisson noise is applied to each pixel in the image. An example image with 1000 disks, background of 10 counts, $4 \leq I_0 \leq 1000$ counts, and $1 \leq r_0 \leq 10$ pixels, is shown in Figure 5.1.

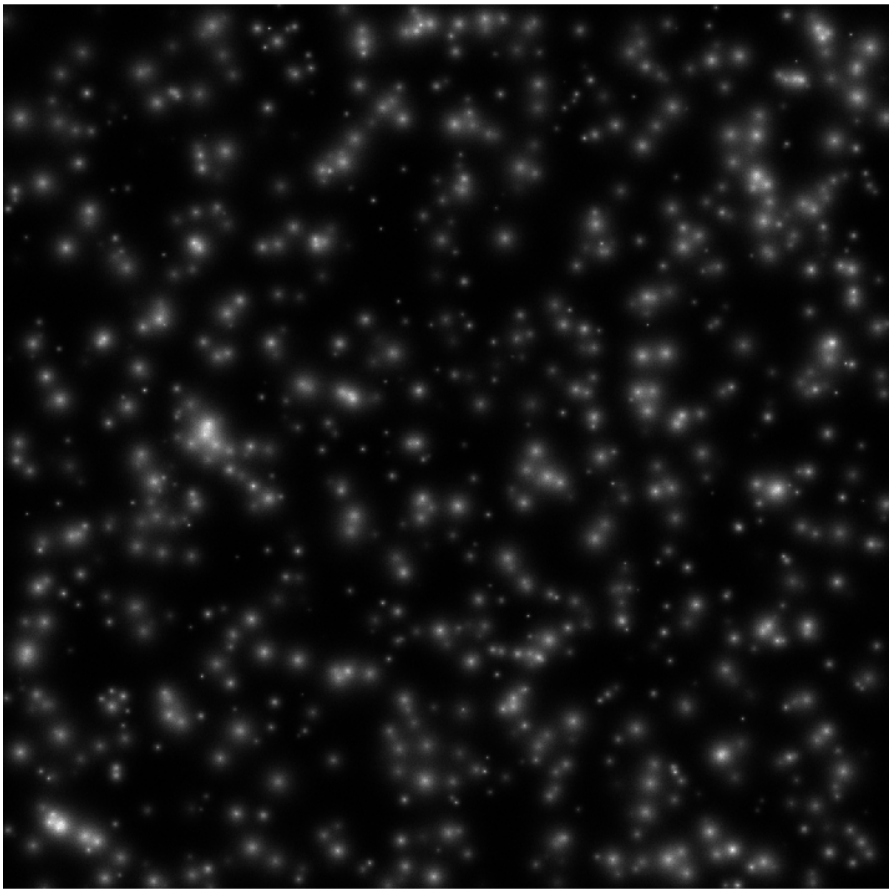

FIGURE 5.1: Image generated by `expdisks` with 1000 disks, background of 10 counts, $4 \leq I_0 \leq 1000$ counts, and $1 \leq r_0 \leq 10$ pixels.

To quantify the accuracy with which SExtractor recovers the simulation parameters after our changes, we define the discrepancy between a simulated (X_{sim}) and a recovered (X_{rec}) quantity as $dX = |X_{sim} - X_{rec}|/X_{sim}$. The histograms for the χ^2 of the exponential disk fit and for the discrepancies of r_0 and I_0 are presented respectively in figures 5.2, 5.3 and 5.4. In them we see the great majority of events are in the first bin, with a discrepancy of less than 50% and $0 < \chi^2 < 2$. This indicates our changes do not impact significantly the SExtractor model-fitting for the exponential disk profile. In the future, more thorough testing should be performed for the exponential disk and for the other profiles supported by SExtractor.

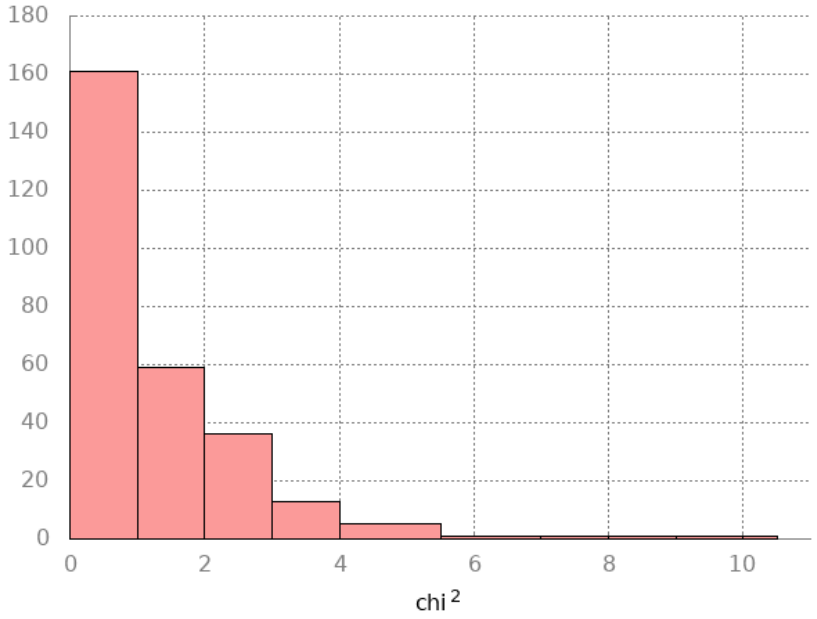

FIGURE 5.2: Histogram of the χ^2 of the exponential disk fitted by SExtractor to the events generated by `expdisks`.

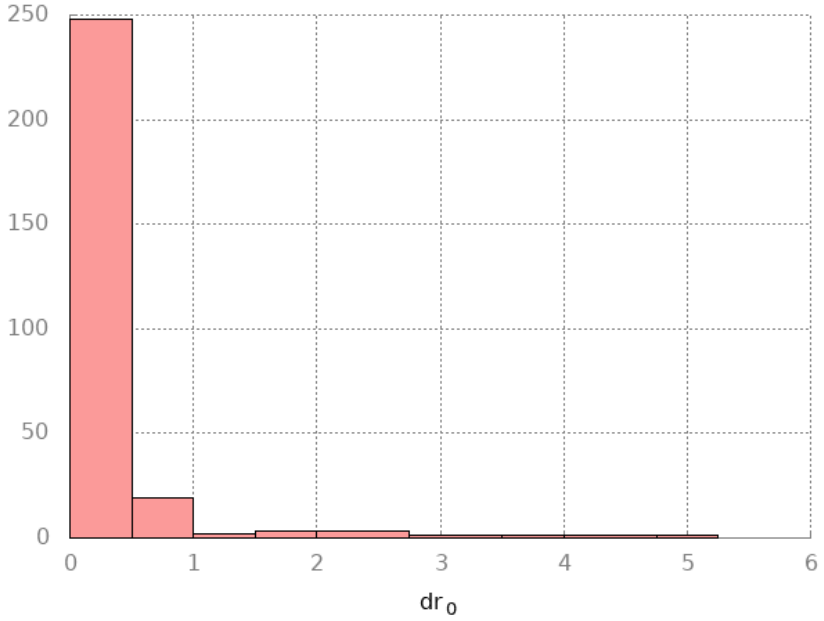

FIGURE 5.3: Histogram of the discrepancy $dr_0 = |r_{0_{sim}} - r_{0_{rec}}|/r_{0_{sim}}$ of the exponential disk fitted by SExtractor to the events generated by `expdisks`.

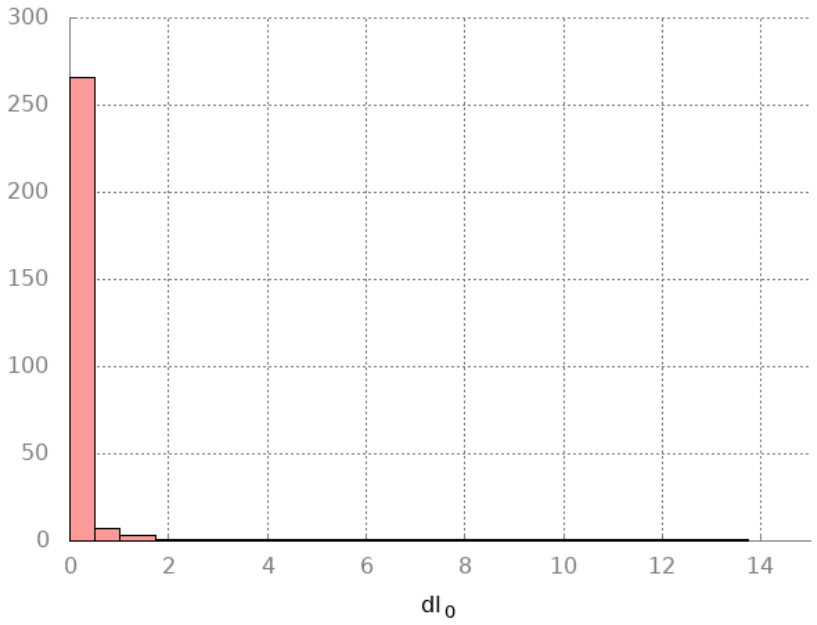

FIGURE 5.4: Histogram of the discrepancy $dI_0 = |I_{0_{sim}} - I_{0_{rec}}|/I_{0_{sim}}$ of the exponential disk fitted by SExtractor to the events generated by `expdisks`.

5.2 Model-fitting and event morphology

The extracted events were modeled by SExtractor according to the Sérsic profile [50] [47], a model usually applied to describe the intensity of light of galaxies, in which the Gaussian distribution is a special case. As a function of the distance $r = \sqrt{x^2 + y^2}$ from the event's center, the profile is defined as

$$I(r) = I_0 \exp\left[-b(n)\left(\frac{r}{R_e}\right)^{1/n}\right], \tag{5.2}$$

where I_0 is a normalization, $b(n) \approx 2n - 1/3$, R_e is an effective radius, and n is the Sérsic index.

The model-fitting related parameters that we used for this chapter are the reduced[1] χ^2 of the fitting, `CHI2_MODEL`, the fitted effective radius R_e, `SPHEROID_REFF_IMAGE`, the fitted Sérsic index n, `SPHEROID_SERSICN`, and the ellipticity ϵ of the fitted model. The latter is computed from the second moments of the pixel values m_x^2, m_y^2 and m_{xy}, via the parameters `ELLIP1MODEL_IMAGE` and `ELLIP2MODEL_IMAGE`:

[1] χ^2 per degree of freedom

$$\texttt{ELLIP1MODEL\_IMAGE} = \frac{m_x^2 - m_y^2}{m_x^2 + m_y^2} \tag{5.3}$$

$$\texttt{ELLIP2MODEL\_IMAGE} = \frac{2m_{xy}}{m_x^2 + m_y^2} \tag{5.4}$$

$$\epsilon = \sqrt{\texttt{ELLIP1MODEL\_IMAGE}^2 + \texttt{ELLIP2MODEL\_IMAGE}^2}. \tag{5.5}$$

We then expect neutrino events to be well modeled by the Sérsic index $n = 1/2$, corresponding to a Gaussian distribution. Also, since the hits are limited by diffusion, the effective radius is expected to be smaller for neutrinos than other types of events. Last, diffusion-limited hits are expected to have a more circular shape, with an ellipticity close to zero.

5.3 Selection

Before we apply cuts on several morphological quantities and explore their effect on the completeness and contamination, we discuss an improvement for the identification of the simulated neutrinos in the `draw` images. The matching discussed in Section 4.1 is purely positional; it does not consider any other extraction or simulation information. In doing so, in principle it is possible that a spurious detection has a measured barycenter that causes it to match with a simulated neutrino. To decrease this superposition effect, we computed the energy discrepancy between the simulated and the recovered events on catalogs produced by positional matching. Using the matched event energy E_{match} and simulated energy E_{sim}, we defined $dE = (E_{match} - E_{sim})/E_{sim}$. Then, we removed from the match catalogs all events for which $dE > 50\%$. This allows matched events to have an extracted energy lower than the simulated one, but not greater by 50% or more. This latter case would mean we are extracting events together with much of their surrounding noise. Figure 5.5 shows a histogram of dE before and after the filtering. Numerically, the cut removed 0.3% of events in the range $28 < E < 5000$ eV, and 3% of events in the range $28 < E < 300$ eV, the latter being the range in which we have the largest contamination. The next processing steps all use the dE-filtered catalogs as the match catalogs.

We identified the events with positional and energy matching as neutrinos, and the remaining ones as the background. From these two sets of events we produced histograms and scatter plots of the parameters. Our goal was to see if the matched events and the background would each cluster in different regions of the parameter space.

Plotting $R_e \times \chi^2$ (Fig. 5.6) shows no distinct clustering regions for matches and background. From the histograms in Figure 5.7, we see that a cut $1/2 \leq R_e \leq 2$ pixels would reduce the contamination, while keeping most neutrino events. However, we see from Figure 5.8 that the impact of the cut

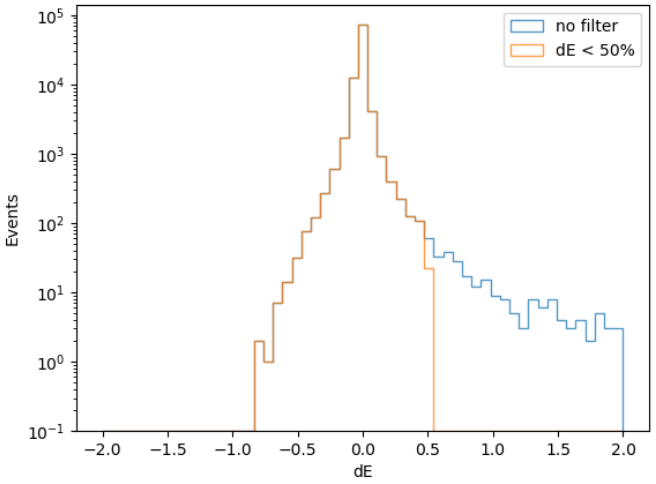

FIGURE 5.5: The effect of removing the events for which $dE > 50\%$ from the match catalogs.

affects both the completeness and the contamination in a roughly equal manner. It is interesting to note that up to 70 eV, this cut is able to increase the completeness by excluding hits detected in `data` images. Hence, the R_e cut is taken for further consideration.

The histograms in Figure 5.9 exhibit a weak relation between the reduced χ^2 of the Sérsic fit and the matched events. If we force a cut on $\chi^2 < 5$, we not only decrease the completeness up to 5 keV, but also increase the contamination in the range $E < 300$ eV that we are most interested in, as can be seen from Figure 5.10.

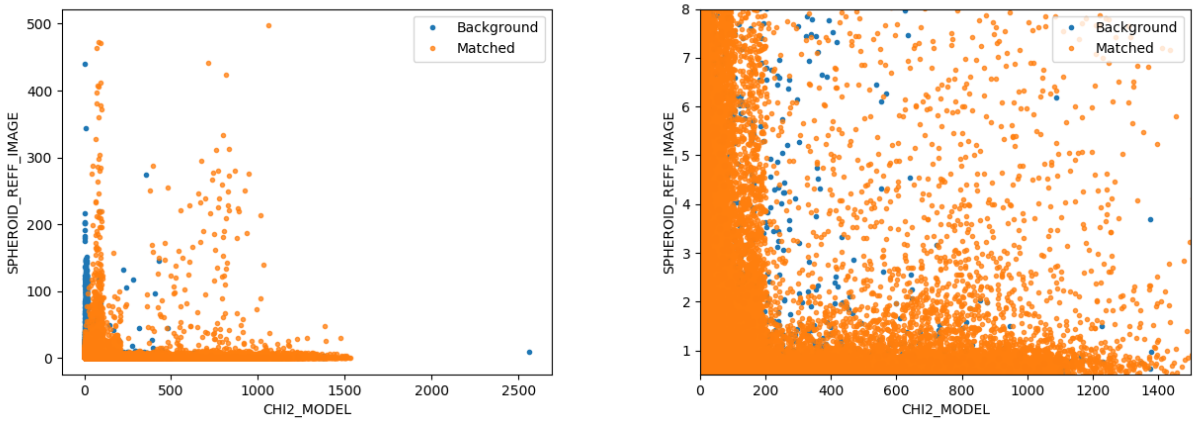

FIGURE 5.6: Scatter plot of $R_e \times \chi^2$ showing no clear separation between background and matched events.

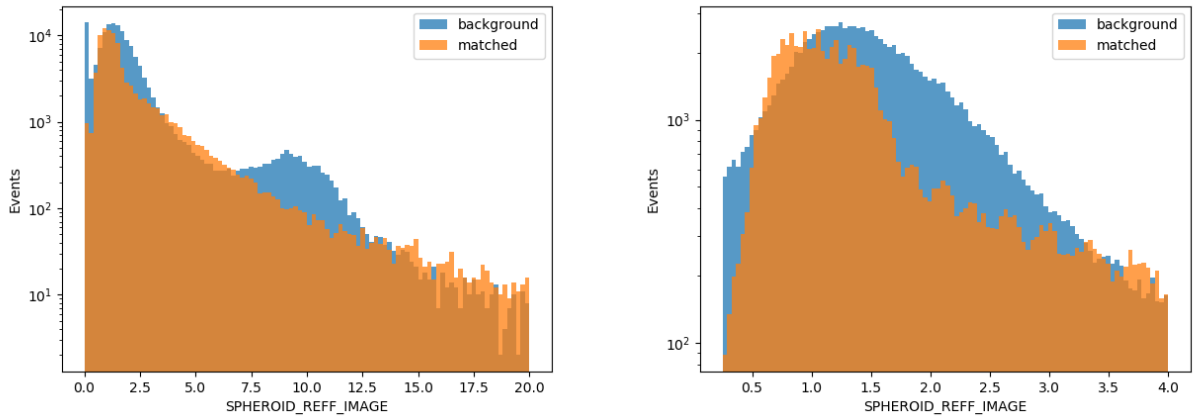

FIGURE 5.7: Histograms of the effective radius R_e for matched and background events.

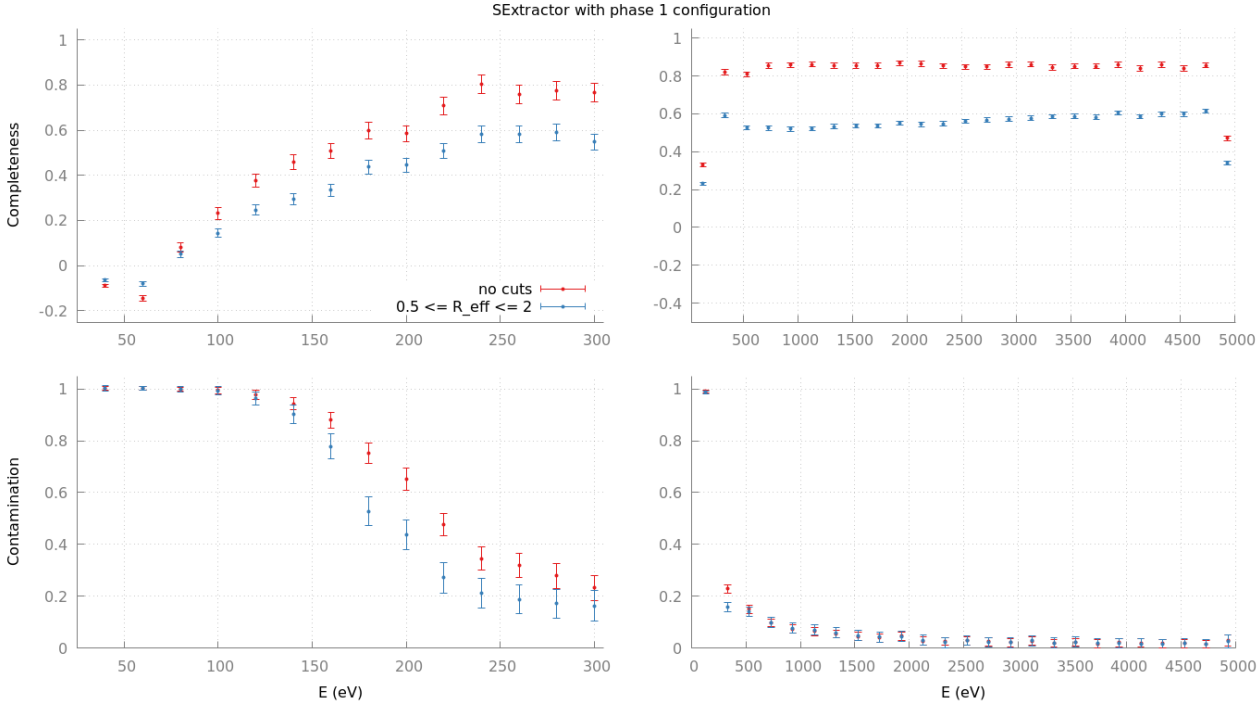

FIGURE 5.8: Efficiency plot with and without the cut $1/2 \leq R_e \leq 2$.

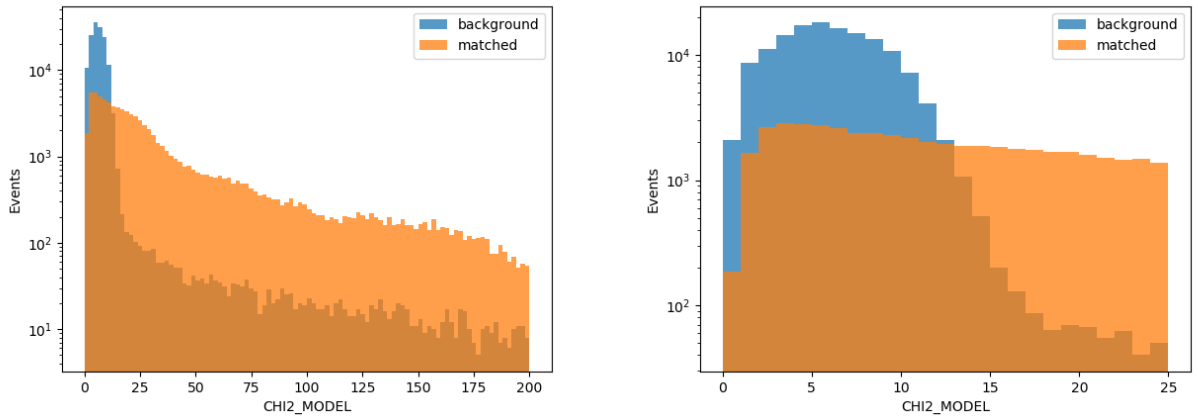

FIGURE 5.9: Histograms of the reduced χ^2 of the Sérsic fit for matched and background events.

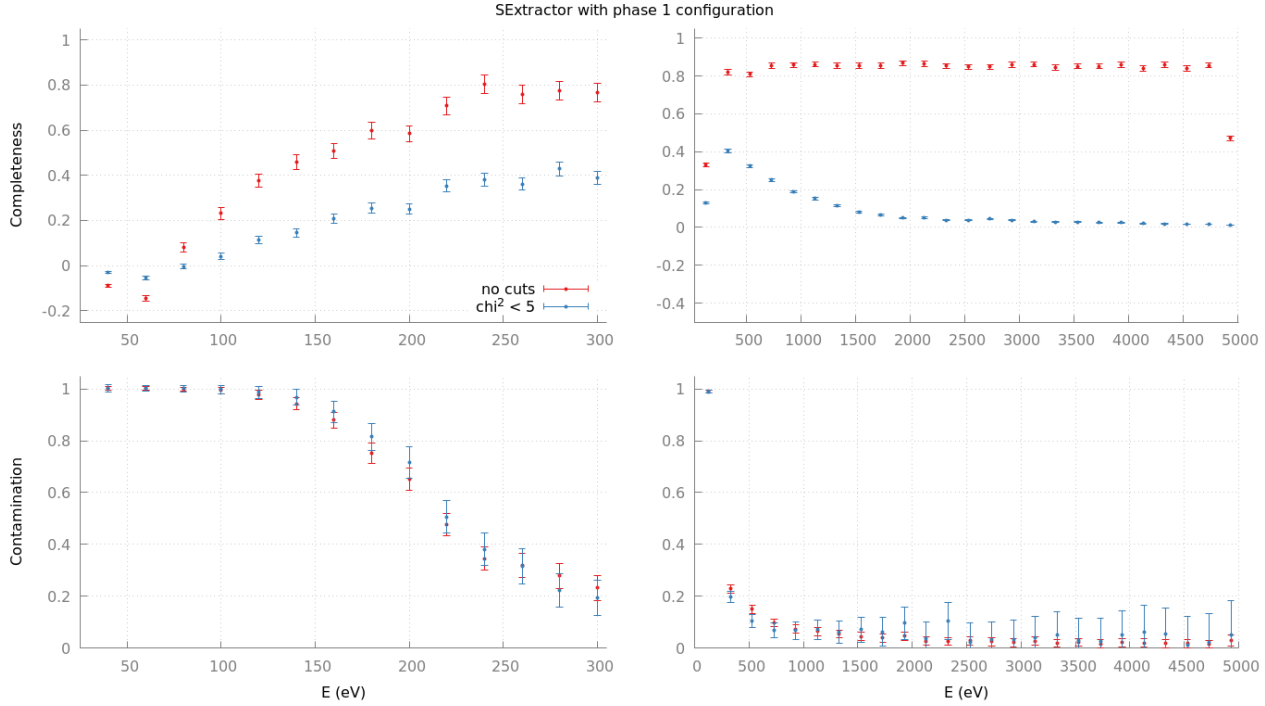

FIGURE 5.10: Efficiency plot with and without the cut $\chi^2 < 5$.

In SExtractor, extracting and fitting an event may trigger some special conditions, such as an event being blended with another. To warn the user of a condition, a flag is stored in the output catalog. The histograms of the flags for both the extraction (`FLAGS`) and the model-fitting (`FLAGS_MODEL`) are presented in Figure 5.11. From them we see that most events, matched or background, raise no SExtractor warnings, and for all flags there are more background than neutrino events. However, we also see that the fraction of neutrinos is higher for the events in which the flags are zero. Thus, we

exclude all events with non-zero flags of any type. Figure 5.12 shows that selecting `FLAGS = 0` has a stronger impact in the contamination than in the completeness, and therefore could be useful in selecting a purer neutrino sample. On the other hand, Figure 5.13 shows that setting `FLAGS_MODEL = 0` decreases the completeness, while having almost no impact on the contamination. Therefore, there is no advantage in using this cut.

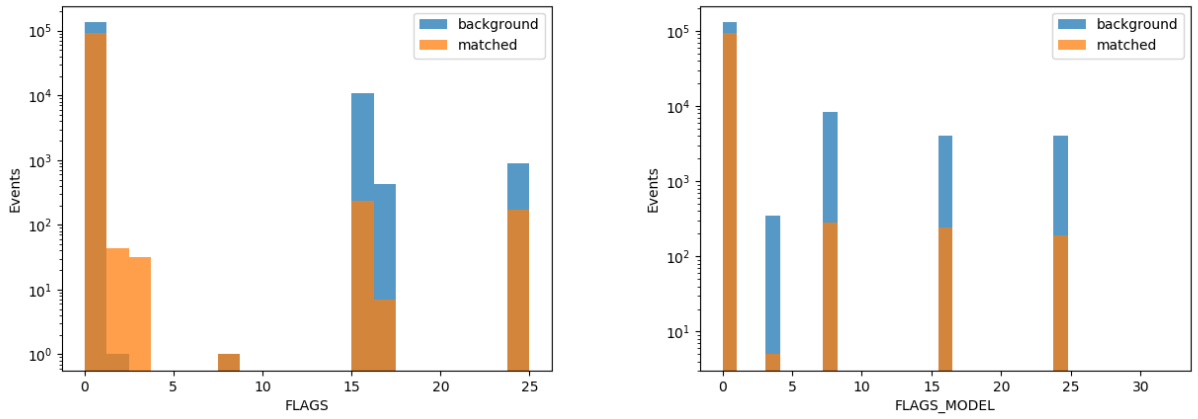

FIGURE 5.11: Histograms of the extraction flags `FLAGS` and fitting flags `FLAGS_MODEL` for matched and background events.

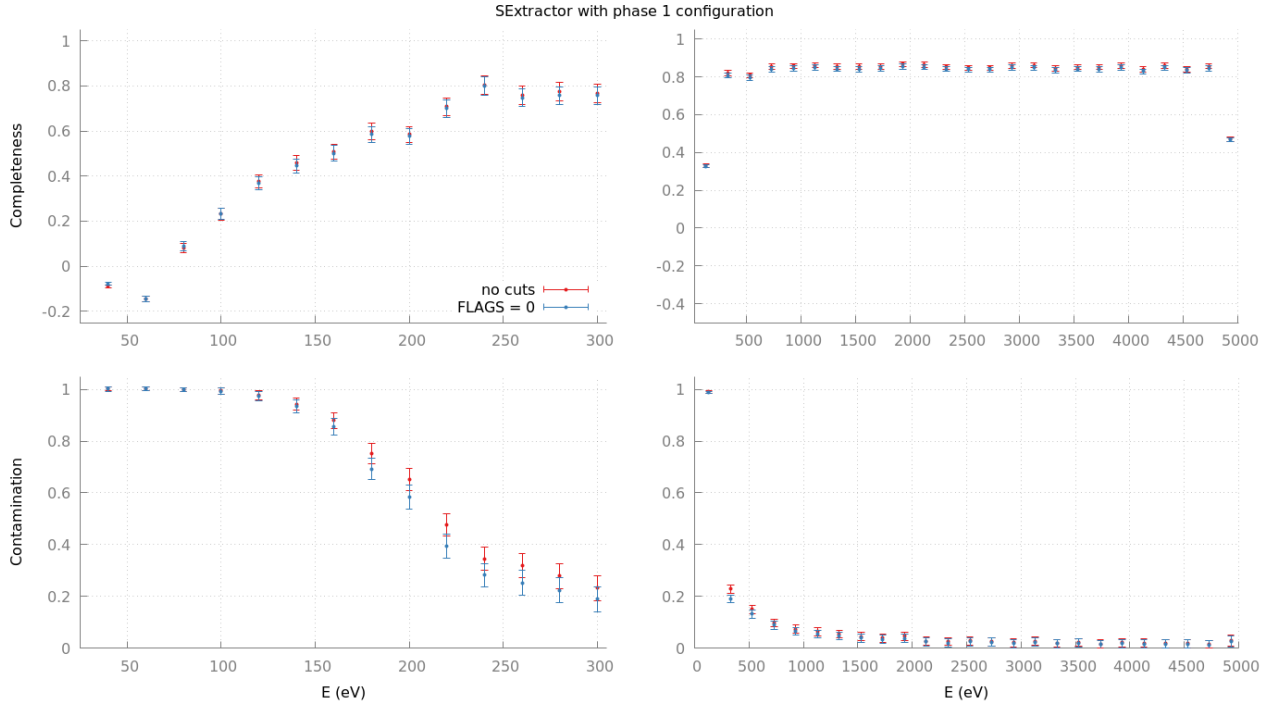

FIGURE 5.12: Efficiency plot with zero and non-zero extraction flags.

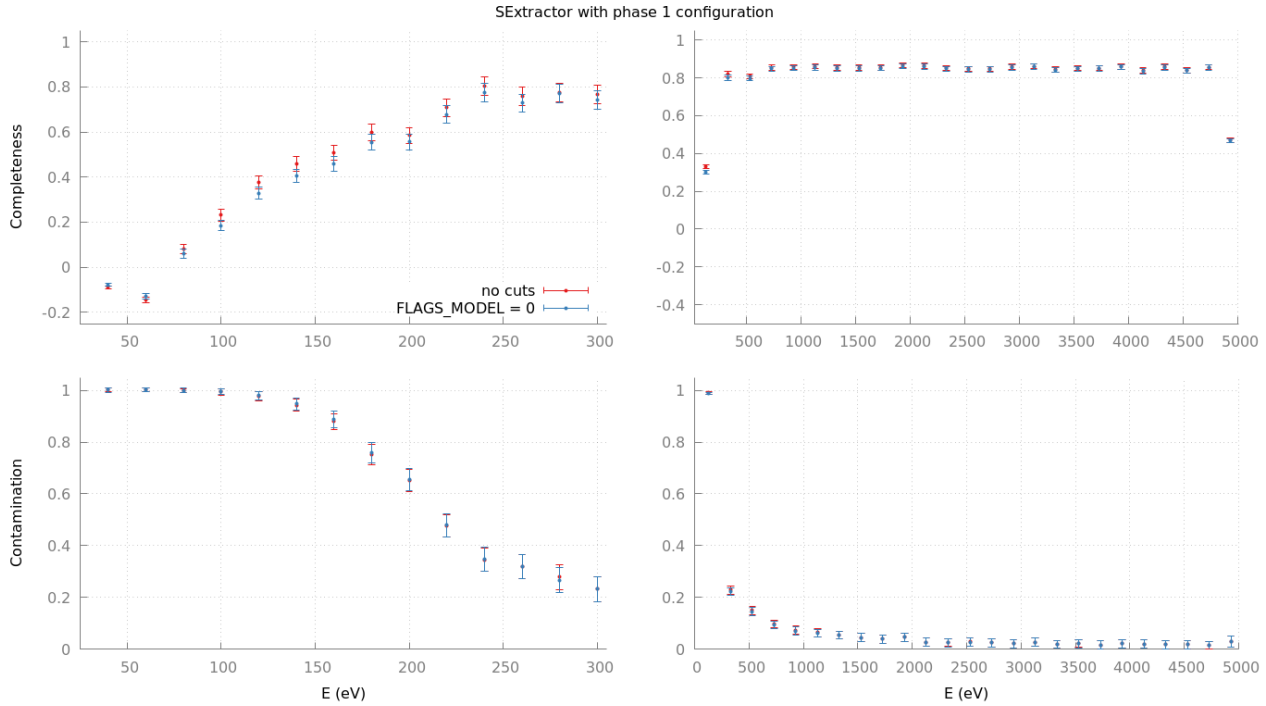

FIGURE 5.13: Efficiency plot with zero and non-zero model-fitting flags.

In Figure 5.14 we show the scatter plot of $\epsilon \times n$. This plot shows no separation between matched and background events. As discussed, this is in stark contradiction to our expectations; if a clear separation does not exist, at least we would expect the matched events to be clustered around $\epsilon = 0$, $n = 1/2$. We see from the histogram on the right panel of Figure 5.15 that the number of neutrinos decreases significantly for $n < 1/2$ and for $n > 6$, while the background is less affected by these limits. Therefore, we tested the cut $1/2 < n < 6$. Nevertheless, Figure 5.16 shows that the restriction on the Sérsic index has a totally negative impact, decreasing the efficiency and increasing the contamination.

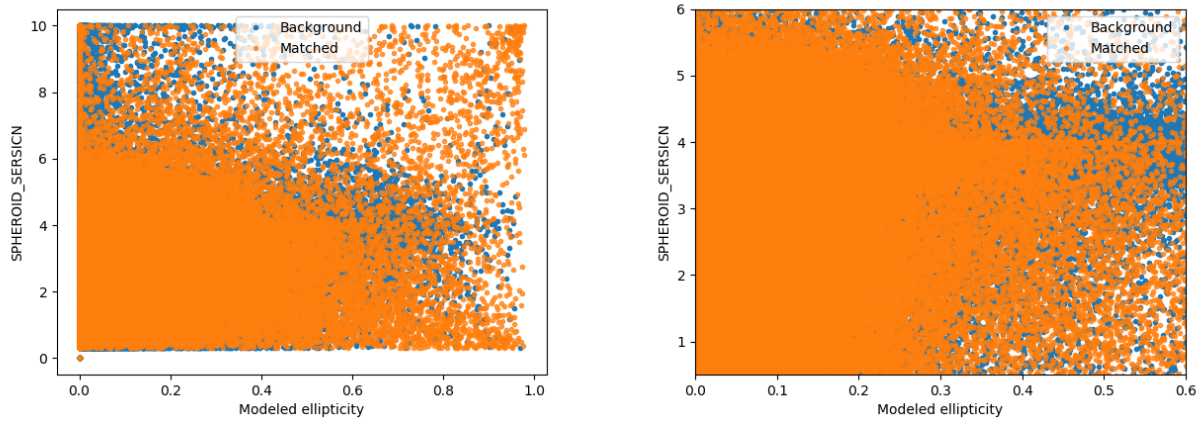

FIGURE 5.14: Scatter plot of $\epsilon \times n$, again showing no clear separation between background and matched events.

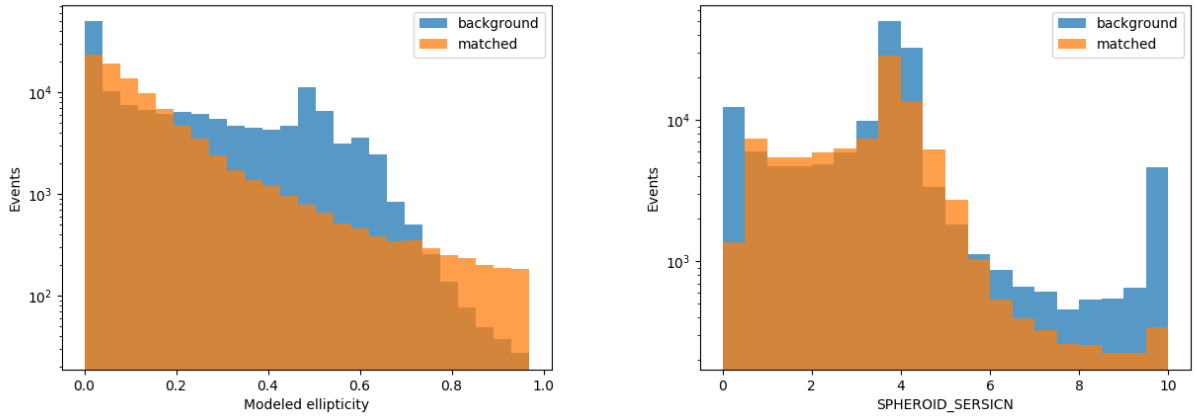

FIGURE 5.15: Histograms of the ellipticity (left) and the Sérsic index n (right) for the matched and background events.

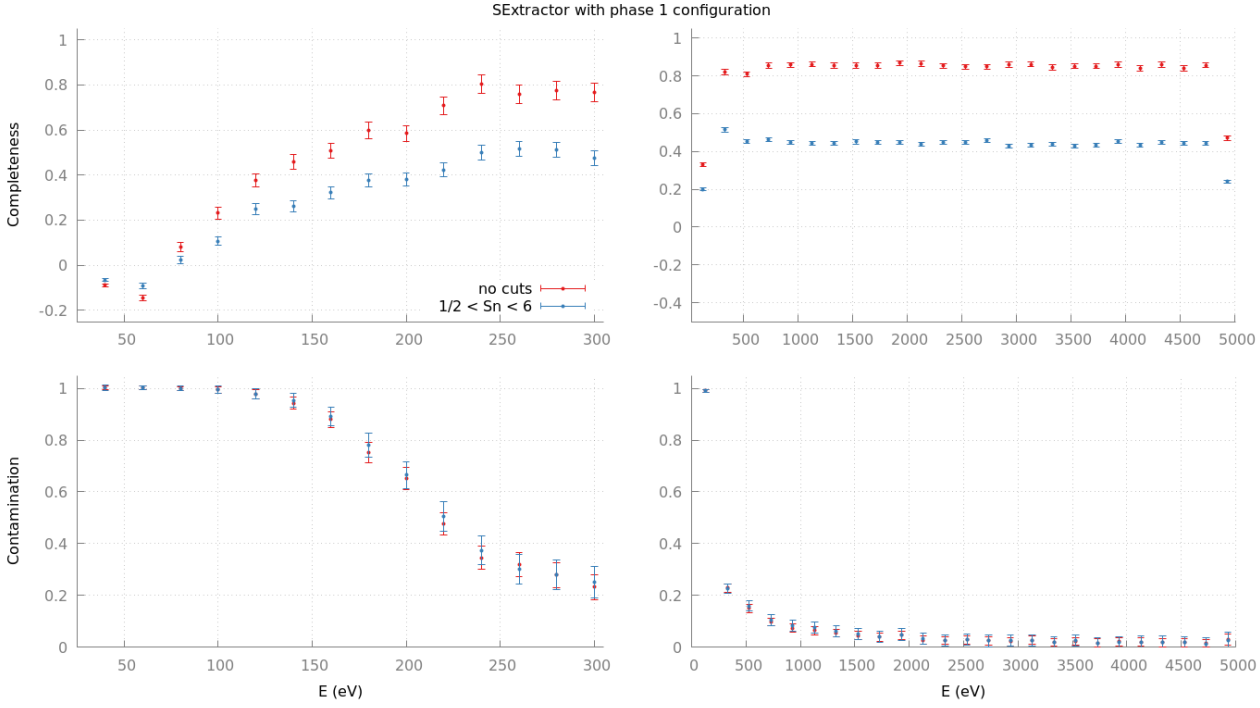

FIGURE 5.16: Efficiency plot with and without the cut $1/2 < n < 6$.

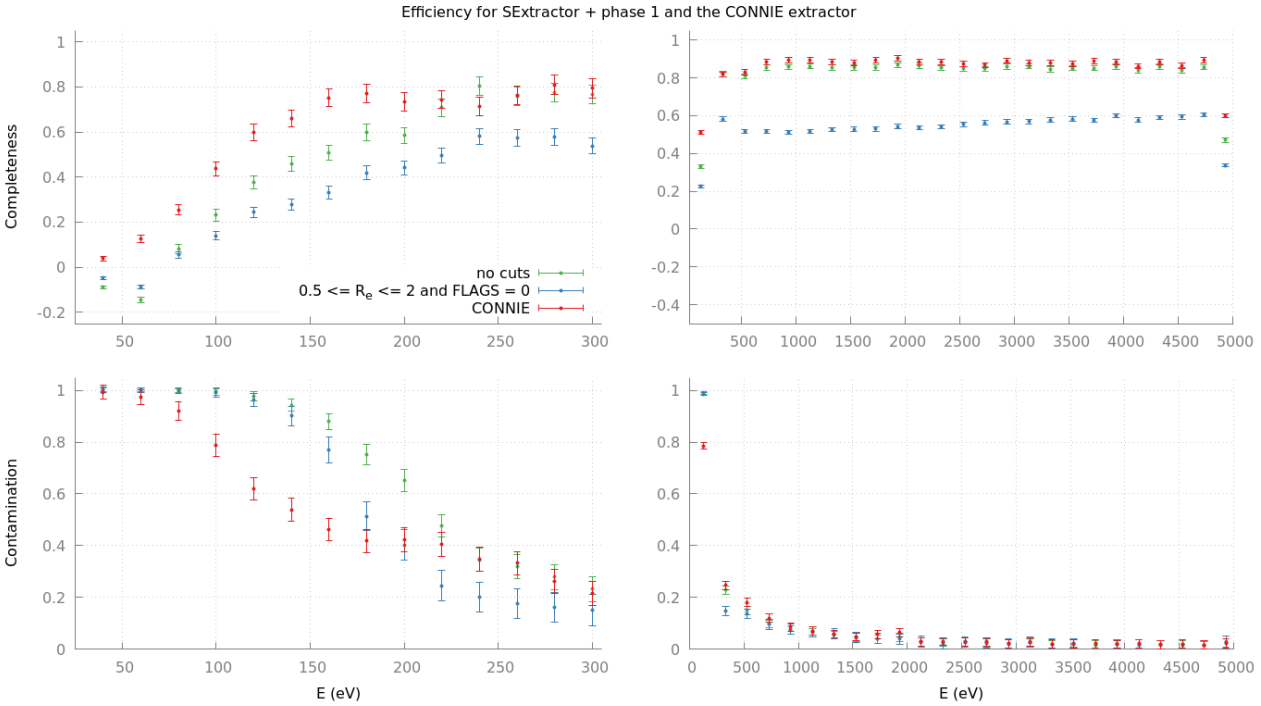

FIGURE 5.17: Comparison between the efficiency and contamination of SExtractor with no cuts, SExtractor with both R_e and `FLAGS` cuts applied, and the CONNIE extractor.

Lastly, we combined the cuts $1/2 \leq R_e \leq 2$ and `FLAGS = 0`, resulting in Figure 5.17. From this and the past figures, we see that no single parameter or combination of parameters allow us to differentiate between simulated neutrinos and background events. The cuts that manage to decrease the contamination, our main goal in the chapter, do so at the cost of also decreasing the efficiency. In other words, our use of event morphology did not enhance the selection of neutrinos.

Chapter 6

Discussion and concluding remarks

Despite being discovered more than 60 years ago, neutrinos remain one of the most mysterious particles in the Standard Model of particle physics. As varied as the neutrino sources and energies are, so are the efforts to understand them using the wide range of methods, techniques and tools reviewed in Chapter 1. The Coherent Neutrino-Nucleus Interaction Experiment stands out as currently having the lowest energy threshold for neutrino detection (see Chapter 2). The CCD technology used in its detector, although well understood in general, is being applied for the first time to the search of neutrinos. The processing of the detection images is an essential step for analyzing the data, and so CONNIE has developed its own image processing pipeline.

Our main motivations for this work are:

1. To cross check the CONNIE event extraction with a well established extraction software used in astronomical CCD image processing
2. To enhance event extraction for energies below 300 eV
3. To test if event morphology is able to enhance the selection of neutrinos from the extracted events

To address these issues, we developed a new pipeline for processing the CONNIE images. From the SCN (our "science grade") images described in Chapter 2 as main input data, the pipeline performs event extraction, event model-fitting, manipulation of FITS catalogs, generation of spectra, filtering of ROOT and FITS catalogs, event matching, and other procedures. In short, it is a set of software tools to perform several of the tasks related to the analysis of CONNIE images. The programs, some of which are described in Appendix B, work independently of the CONNIE pipeline. They are available to the collaboration in the CONNIE's code repository. Event extraction and model-fitting, the most computationally intensive parts of the pipeline, are performed by SExtractor, a free

software originally written for and widely applied to astronomical images. Moreover, the SExtractor model-fitting code was modified to not perform the convolution of pixel values with a point-spread-function, a step required in astronomical applications but inapt to our case. Such changes are also available publicly, in [51].

In accordance with our first motivation, as discussed in Chapter 3, we applied our pipeline to the CONNIE images and generated spectra from the catalogs produced both by the CONNIE extractor and SExtractor. The results were consistent in the wide range 300 eV $< E <$ 20 keV, attesting the robustness of the two pipelines, and showing that the spectra are essentially independent of the extractor used. Nonetheless, improvements can still be made on the generation of spectra from our pipeline. The differences we see in Figure 3.16 between the energies reconstructed by the two pipelines can be attributed to our simplistic calibration. While CONNIE uses a sum of two Gaussians fitted to two Cu peaks, we use the linear relation (3.1) calculated from a single Cu peak identified visually. In the future, a more sophisticated calibration procedure can easily be implemented.

Another motivation for our work was to understand the efficiency in the detection of the diffusion-limited hits present in the CONNIE images. In Chapter 4 we implemented a methodology for measuring the ability of the image processing procedures to identify such events. The methodology takes into account the goal of detecting neutrinos in CONNIE, and does not depend on a specific pipeline. As such, it can be applied to different implementations of the various processing steps. The result is a pair of quantities: the binned efficiency and the binned contamination. Applying such methodology to the extraction catalogs, we could state that below 300 eV, and using a matching radius of $d_{min} = 2$, the CONNIE pipeline has higher efficiency and lower contamination than our pipeline.

As discussed in Chapter 5, for our third motivation we used SExtractor's model-fitting and event morphology features. However, no single morphological parameter or combination of parameters was helpful in decreasing the contamination of our extraction. In other words, morphology was unable to help discriminate between neutrino and background events. An intriguing result is the disagreement between the fitted Sérsic index n and ellipticity ϵ and their expected values. Further investigation should be done to ensure that the neutrino simulations are correct and produce events with a Gaussian intensity profile in two dimensions. In that regard, it would be useful for SExtractor to have a way of fixing the Sérsic index and fit only the effective radius. The modifications we have done in SExtractor allowed us to gain familiarity with its source code, a useful skill if there is a need for more changes.

In summary, this work provides CONNIE with three assets. First and foremost, a general methodology that takes the experiment's goals into account to measure the quality of image processing procedures. Second, the set of software tools we developed are simple and general enough to be readily used by the collaboration members in a number of different situations related or not to this work's specific goals. And last, an independent pipeline for processing the experiment's images.

In the future, a number of directions may be taken to follow up on our results. We may incorporate selection criteria already used by CONNIE: exclusion of the image borders, to avoid spurious detections; exclusion of regions with strong noise; mask CCD columns known to be bad. After the extraction, we could also compute and CONNIE's `NoiseLike` variable, a quantity that accounts for the probability of an event being compatible with noise. Furthermore, taking advantage of our software tools and SExtractor's flexibility, it is possible to use the CONNIE extraction catalogs as input for SExtractor model-fitting and morphological measurements. Moreover, in the subsequent selection, a combination of `NoiseLike` with morphology may improve selection.

So far, the exploration of the SExtractor configurations and their respective extraction results was done by a manual process. There are a number of ways to automatize and improve such steps. Genetic algorithm techniques can be used to evolve various configurations with the aim of increasing efficiency. Supervised machine learning, *e.g.* artificial neural networks, can be used to either select diffusion-limited hits from an extraction catalog, or to directly search the input images for those events.

Lastly, any venues to be taken by CONNIE in the processing and analysis of the collaboration's images might build upon the results achieved, and the tools, the methodology and the techniques developed, implemented or used in the present work.

Appendix A

Object detection with SExtractor

SExtractor [47] [48] is a program whose main goal is to extract, perform measurements on, and classify objects present in FITS image files [44]. As input data, it requires an image where to perform the extraction and measurement. Optionally, it can use a different image for the measurements, a weight map with the relative weights of pixels, and a flag map marking specific pixels as unusable. Being a command-line program, SExtractor is controlled and configured by command-line arguments and configuration files. These may adjust input parameters for algorithms, select different output files, and control which results are stored for latter analysis. In practice, SExtractor works as a pipeline comprising background estimation and subtraction, object segmentation, object deblending, and measurement of object properties. Each phase uses the results from the previous ones and emits data for the next ones, as pictured in Figure A.1. The final result is a catalog in which each entry contains the extracted event's properties, such as position, flux, elongation, etc. Optionally, some of the processing steps may produce *check images*, FITS images with some step's results, such as the computed background. The following sections describe the most relevant SExtractor's processing steps in greater detail.

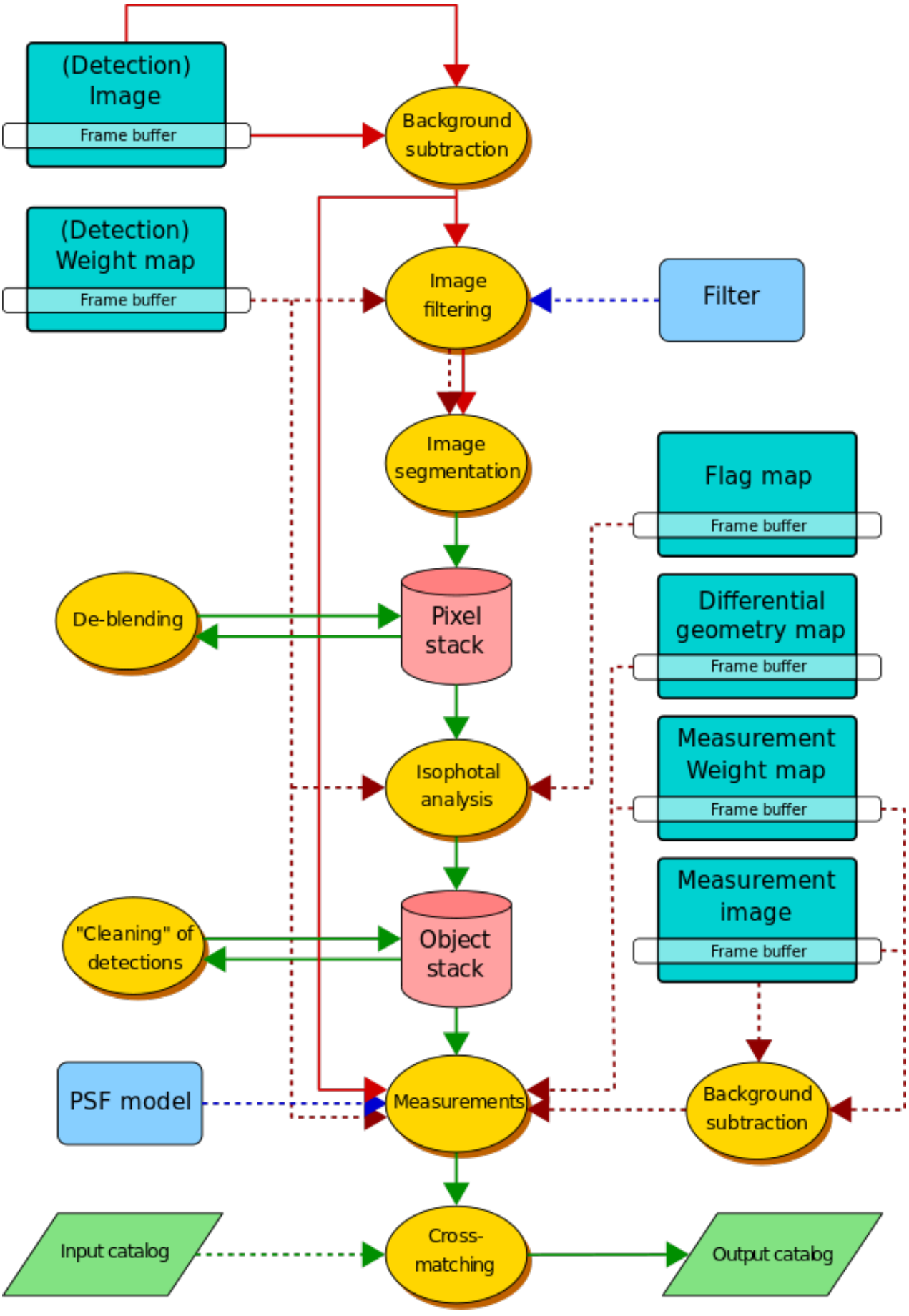

FIGURE A.1: SExtractor flow diagram from [47]. Dashed lines are optional steps.

A.1 Background estimation

As its first step, SExtractor computes the background map of the input image. It divides the image in a grid of equally spaced meshes and calculates the background signal in each mesh [47]. To avoid overestimations local to meshes, a filter defined by the user may smoothen the resulting map. Finally, the background value for each pixel is subtracted from the total pixel's signal. In parallel, a map of the background noise standard deviation is also produced. The background and background RMS maps, and the resulting image after background subtraction, can be output respectively to the check images `BACKGROUND`, `BACKGROUND_RMS`, and `-BACKGROUND`.

The user can control background estimation by choosing input parameters such as the mesh size (`BACK_SIZE`) and filter size (`BACKFILTER_SIZE`). As a special case, by setting `BACK_TYPE` to `MANUAL` it is also possible to set the background map to the constant defined by `BACK_VALUE`.

A.2 Segmentation

After background estimation, SExtractor proceeds to execute its segmentation procedures, *i.e.* grouping pixels as objects. There are three criteria by which a collection of pixels must pass in order to be grouped as an object. First, each pixel value must be above a certain detection threshold, configured by the `DETECT_THRESH` parameter. This threshold can be specified via `THRESH_TYPE` in absolute ADU and applied equally to all pixels, or in sigmas relatively to the background RMS, thus depending on the background estimation local to each pixel. When processing all the pixels above the detection threshold, if the current pixel has neighboring pixels also above the threshold (Fig. A.2), they are said to be connected. A set of pixels must be connected to form an object. At last, a minimum number of connected pixels is required, configured by the `DETECT_MINAREA` parameter. Also, a maximum number of pixels before an object is forcefully split can be configured by the `DETECT_MAXAREA` parameter.

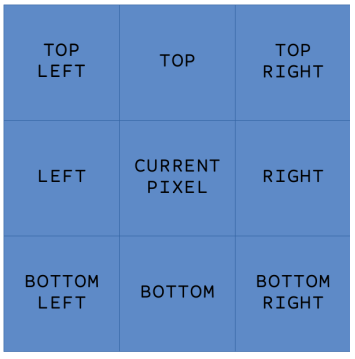

FIGURE A.2: Pixel neighbors as considered by SExtractor

A.3 Deblending and cleaning

Due to physical effects or previous image processing, distinct sources of signal may have been segmented as a single object. SExtractor then proceeds on applying a deblending procedure. The first step is dividing the intensity interval, from the detection threshold up to the maximum intensity for the object, in exponentially spaced levels, the number of which is controlled by the `DEBLEND_NTHRESH` parameter.

The procedure continues by examining pixel values in relation to each of the increasing levels. If a group of pixels is above one level and surrounded by pixels below that level, the group is a candidate for being split into a new object. The new candidate group not only must have pixels above a certain level, but also provide a minimum contribution to the original object's total intensity. This contribution, or candidate contrast, is defined as percentage in the `DEBLEND_MINCONT` parameter. However, even if a group of pixels passes the deblending criteria, they are still not separated into a new object. It is only after two or more groups of pixels from the same original object qualify as candidates, that they are split into many new objects.

If the `CLEAN` parameter is set as true/yes, for each resulting object after the deblending, SExtractor computes the contributions of its neighbors to its signal. Those contributions are then subtracted from the object's pixels. If the object's new signal is not enough for it to be segmented/detected, the object is a is marked as a spurious detection and removed from the list of objects. The efficiency of the cleaning is controlled by `CLEAN_PARAM`.

By the end of this process we have the final list of objects extracted from the image, each identified by a unique number. Two check images are also produced: SEGMENTATION, in which all the pixels pertaining to an object have the values set to the object identification number; and OBJECTS, where pixels from objects have their original values, but any other pixel is blank. Using SEGMENTATION to identify an object's pixels and OBJECTS to assign these pixels their values, we can construct a *postage stamp*: an image of a single object.

A.4 Measurements

With the list of objects ready, SExtractor starts computing quantities for each of them separately, storing then in the output catalog's entries. The majority of these measurements are either positional or photometric. The first group comprise quantities derived from the spatial distribution of pixels in an object, such as its barycenter `X` and `Y`; minimum and maximum x and y coordinates of a rectangle enclosing the object `XMIN`, `YMIN`, `XMAX` and `YMAX`; second moments `X2`, `Y2`, `XY` and, derived from them, `ELLIPTICITY` and `ELONGATION`. The second group, quantities such as magnitudes (`MAG_`)

and fluxes (`FLUX_`), grouped into four distinct types: isophotal, corrected-isophotal, fixed-aperture and adaptive-aperture. The definitions for the measurement parameters are given in [47].

One output parameter of special importance in real world usage, *e.g.* selection of the objects for further processing, is `FLAGS`. It provides information about the quality of an object: whether it was originally blended, has saturated pixels or is too close to an image boundary.

A.5 Model-fitting

Before proceeding to the measurement phase, SExtractor can optionally fit one or a combination of various models to the detected object. A model is defined as a function that takes a pixel position $\boldsymbol{x}$ and gives the pixel's value. As usual, the model also depends on a set of parameters. An example is the exponential disk, which depends on $\boldsymbol{q} = (m_0, h, \boldsymbol{x})$ and is defined as

$$m_{\texttt{DISK}}(\boldsymbol{q}, \boldsymbol{r}) = m_0 \exp\left(-\frac{|\boldsymbol{r} - \boldsymbol{x}|}{h}\right). \tag{A.1}$$

Taking the ith pixel value above the background p_i, this value's uncertainty σ_i, and the resampled model value $\widetilde{m}(\boldsymbol{q})$, the actual fitting procedure is performed by minimizing the loss function

$$\lambda(\boldsymbol{q}) = \sum_i \left[g\left(\frac{p_i - \widetilde{m}_i(\boldsymbol{q})}{\sigma_i}\right)\right]^2 + \sum_j \left(\frac{f_j(Q_j) - \mu_j}{s_j}\right)^2. \tag{A.2}$$

The first term in (A.2) is a modified weighted sum of squares, and is used for minimizing the residuals of the fit. The $g(u)$ function acts to reduce the influence exerted by possible large discrepancies between the pixel and model values, and the second term is related to the bounding and regularization of the model parameters.

After the fitting, every object has a corresponding modeled object. Hence, in addition to the fitting coefficients `VECTOR_MODEL`, SExtractor can output quantities derived from the modeling, such as the flux in the modeled object `FLUX_MODEL`, reduced χ^2 of the fit `CHI2_MODEL`, and model-fitting flags `FLAGS_MODEL`.

Appendix B

CONNIE pytools

In the course of processing CONNIE's data, a set of unix style tools were developed from scratch using Python [52], Astropy [53] and Matplotlib [54]. The *CONNIE pytools* are available for the collaboration and compose part of the independent pipeline devised in this work, being used daily by the author. In special, `fitsmerge` was used for merging all the FITS tables mentioned in the text; all histograms were produced by `fitshist`; all the catalogs produced by the CONNIE extractor and simulations were converted to FITS tables by `root2fits`.

B.1 expr.py

`Expr.py` is a module providing the use of Python expressions for filtering and computing data based on FITS tables columns. It is included in all the following tools, for example for providing the filtering and calibration features.

B.1.1 Examples

- Select all table rows where the events positions, given by the column names `X` and `Y`, lie in a square region of 10 pixels:

```
newdata = expr.filter(table.hdu[1].data,\
  '(X >= 10 and X <= 20) and (Y >= 10 and Y <= 20)')
```

- Generate the energy data from the `FLUX_AUTO` columns, using the calibration 3.1:

```
newdata = expr.compute(table.hdu[1].data, 'FLUX_AUTO*0.628')
```

B.2 extinfo

`Extinfo` calculates the completeness and contamination in each of a number of energy bins for an extraction, as defined in (4.2). The output is a text file with the following space separated columns: energy of the bin, completeness, error in completeness, contamination, error in contamination, number of matches, number of matches in the `draw` extraction, and number of matches in the `data` extraction, if a `data` match catalog was provided. See Chapter 4 for the definition of the catalogs and usage scenario.

B.2.1 Examples

- Calculate completeness and contamination from 30 to 300 eV, divided in bins of 20 eV, with the calibration 3.1 in all catalogs, and using both match catalogs.

```
$ extinfo -bin 20 -range 30 300\
    draw_merged_extraction_catalog,param="FLUX_AUTO*0.628"\
    draw_match_catalog,param="FLUXEXT*0.628"\
    data_match_catalog,param="FLUXEXT*0.628"\
    > info.dat
```

- Calculate completeness and contamination from 30 to 5000 eV, divided in bins of 200 eV, with the calibration 3.1 in the catalogs, and using only `draw` match catalogs.

```
$ extinfo -bin 20 -range 30 5000\
    draw_merged_extraction_catalog,param="FLUX_AUTO*0.628"\
    draw_match_catalog,param="FLUXEXT*0.628"\
    > info_nodata.dat
```

B.3 fits2text

`Fits2text` outputs the specified columns of a FITS [44] table to the standard output in a space separated plain text format. It is mostly useful for quick and small interactive processing tasks, where the unix text processing tools can be leveraged to aid in the analysis.

B.3.1 Examples

- Output the maximum value of the `FLUX_AUTO` column in the second HDU:

```
$ fits2text -HDU 2 -col FLUX_AUTO cat.fits |
    awk 'BEGIN{max = 0}
         {if($1 > max) max = $1}
         END{print max}'
6399.45
```

- Output the average values of the `FLUX_AUTO` and `MAG_AUTO` columns in the second HDU:

```
$ fits2text -HDU 2 -col FLUX_AUTO,MAG_AUTO cat.fits |
    awk 'BEGIN{avgflux = 0; avgmag = 0}
         {avgflux += $1; avgmag += $2}
         END{print avgflux/NR" "avgmag/NR}'
2.39201 14.5006
```

B.4 fitscat

`Fitscat` concatenates the specified HDUs from FITS files and outputs the result to the standard output in FITS format. It is useful for any task where scattered FITS HDUs need to be gathered into a single multi-extension FITS file.

B.4.1 Examples

- Concatenating HDU 2 of two files into HDUs 2 and 3 of another file:

```
$ fitscat -hdu 2 a.fits b.fits > a2b2.fits
```

- Concatenating HDU 3 from one file and 4 from another into, respectively, HDUs 2 and 3 of a third one:

```
$ fitscat a.fits,3 b.fits,4 > a3b4.fits
```

B.5 fitschi2

Given a expected value and a FITS table, `fitschi2` computes the χ^2 of a histogram of the table in relation to the expected value, as defined in (3.2). The tool is useful for comparing the efficiency of extraction catalogs against each other and a reference value, *e.g.* Table 3.3.

B.5.1 Examples

- Calculate the χ^2 using 100 bins in the energy range $0 < E < 5$ keV, with expected value 1000:

```
$ fitschi2 -bins 100 -expect 1000 -filter 'E >= 0 and E <= 5000'
   table.fits
0.8976
```

B.6 fitsfilter

`Fitsfilter` is a wrapper around `expr.py`'s filtering feature. Given a FITS table, it creates a new table with the same structure, but with only the rows selected by the filter expression. It is useful when the same filtered rows are to be used in a number of sequential processes, as in Chapter 4.

B.6.1 Examples

- Create a new table with the rows having energy in the range $28 < E < 5000$ eV, where the energy is calculated on the fly from the `FLUX_AUTO` column:

```
$ fitsfilter\
    'FLUX_AUTO*0.628 >= 28 and FLUX_AUTO*0.628 <= 5000'\
    oldtable.fits > newtable.fits
```

B.7 fitshdus

`Fitshdus` outputs each HDU from a multi-extension FITS file to a separate file with the same structure.

B.7.1 Examples

- Separate a 14-HDU FITS table into 14 FITS tables:

```
$ fitshdus MEFtable.fits
MEFtable-1.fits
MEFtable-2.fits
...
MEFtable-14.fits
```

B.8 fitshist

`Fitshist` plots histograms of FITS tables. It supports the filtering and processing of values before constructing the final histogram. Since it is based on the matplotlib library for Python, it can output to screen or an image file supported by the installed matplotlib.

B.8.1 Examples

- Plot the histogram for the `FLUX_AUTO` column in the third hdu

```
$ fitshist -o out.pdf cat.fits,param=FLUX_AUTO,hdu=3
```

- Plot the log histogram for the `FLUX_AUTO` column, but exclude those values for which `FLUXERR_AUTO` is less than one.

```
$ fitshist -log -o out.pdf\
    cat.fits,param=FLUX_AUTO,hdu=3,\
      filter='FLUXERR_AUTO < 1'
```

- Plot the log histogram for the `FLUX_AUTO` column in a different unit

```
$ fitshist -log -o out.pdf\
    cat.fits,param='FLUX_AUTO/13.6',hdu=3
```

- Plot the log histograms for the `FLUX_AUTO` column from two different files

```
$ fitshist -log -o out.pdf\
    cat.fits,param=FLUX_AUTO,hdu=3\
    cat2.fits,param=FLUX_AUTO,hdu=3
```

- Plot the log histograms for the `FLUX_AUTO` column from the same file, with and without the values for which the `FLAGS` column is zero

```
$ fitshist -log -o out.pdf\
    cat.fits,param=FLUX_AUTO,hdu=3,\
      filter='FLAGS == 0'\
    cat.fits,param=FLUX_AUTO,hdu=3
```

B.9 fitsmerge

`Fitsmerge` merges the values from equal columns of different FITS tables. It allows for coalescing values of quantities from multiple sources.

B.9.1 Examples

- Merge the values of the `FLUX_AUTO` column from two different HDUs into the `FLUX_AUTO` column of a new HDU:

```
$ fitsmerge -col FLUX_AUTO\
    cat.fits,hdu=1 cat.fits,hdu=2 > all.fits
```

- Merge the `FLUX_AUTO` and `FLAGS` columns from HDU 2 in a set of FITS tables, into the `FLUX_AUTO` and `FLAGS` columns of a new table:

```
$ fitsmerge -hdu 2 -col FLUX_AUTO,FLAGS\
    catalog*.fits > all.fits
```

B.10 match+info

`Match+info` executes the steps described in Section 4.2, using `fitsfilter`, `fitsmerge`, `fitsposmatch` and `extinfo`.

B.10.1 Examples

- Run `match+info` on a `phase 1` configuration extraction:

```
$ match+info phase1
```

B.11 mergesimcat

`Mergesimcat` merges the 4 parts of the CONNIE ROOT simulation into a single corresponding FITS table, transforming the position coordinates as described in Section 4.1. It uses `root2fits`, `fitsmerge` and `fitscat`.

B.11.1 Examples

- Merge the parts of a simulation catalog onto one:

```
$ mergesimcat simcat1_p*.root  # output to simcat1_merged_cat.fits
```

B.12 plateau

`Plateau` computes the plateau explained in Section 3.3 for a given table.

B.12.1 Examples

- Compute the plateau using 100 bins, in the energy range $1 < E < 4$ keV, computed from the `FLUX_AUTO` parameter:

```
$ plateau -bins 100\
    -filter 'FLUX_AUTO*6.28e-4 >= 1 and FLUX_AUTO*6.28e-4 <= 4'\
    table.fits
20.27
```

B.13 root2fits

`Root2fits` converts selected branches from a ROOT [55] tree to FITS table columns. It is essential when applying astrophysics techniques, whose programs usually accept FITS input, to particle physics data, which is usually stored in ROOT trees.

B.13.1 Examples

- Select the ohdu, flag and E1 branches of the hitSumm tree to columns of the same name in a FITS table

```
$ root2fits -tree hitSumm\
    -b runID,ohdu,flag,E1,gainCu cat.root > cat.fits
```

B.14 root2text

`Root2text` is the same as `fits2text` but with ROOT trees as input.

Bibliography

1. Chadwick, J. Intensitätsverteilung im magnetischen Spektrum von β-Strahlen von Radium B+C. Verhandlungen der Dtsch. Phys. Gesellschaft **16,** 383–391 (1914).
2. Meitner, L. & Orthmann, W. Über eine absolute Bestimmung der Energie der primären β-Strahlen von Radium E. Z. Phys **60,** 143–155 (1930).
3. Brown, L. M. The idea of the neutrino. Phys. Today **31,** 23–28. ISSN: 0031-9228 (Sept. 1978).
4. Fermi, E. Versuch einer Theorie der β-Strahlen. I. Zeitschrift für Phys. **88,** 161–177 (Mar. 1934).
5. Cowan, C. L., Harrison, F. B., Langer, L. M. & Reines, F. A test of neutrino-antineutrino identity. Nuovo Cim. **3,** 649–651. ISSN: 0029-6341 (Mar. 1956).
6. Zuber, K. Neutrino Physics 2nd. ISBN: 9781420064728. doi:10.1201/b11065. <http://adsabs.harvard.edu/cgi-bin/nph-data%7B%5C_%7Dquery?bibcode=2011neph.book.....Z%7B%5C&%7Dlink%7B%5C_%7Dtype=EJOURNAL%7B%5C%%7D5Cnpapers3://publication/doi/10.1201/b11065> (2011).
7. Hirata, K. S. et al. Experimental study of the atmospheric neutrino flux. Phys. Lett. B **205,** 416–420. ISSN: 03702693 (Apr. 1988).
8. Totsuka, Y. Solar and atmospheric neutrinos observed at Kamiokande. Nucl. Phys. B (Proceedings Suppl. **28,** 67–74. ISSN: 09205632 (July 1992).
9. Ahlers, M. & Halzen, F. Opening a new window onto the universe with IceCube Sept. 2018. doi:10.1016/j.ppnp.2018.05.001. arXiv: 1805.11112. <https://www.sciencedirect.com/science/article/pii/S0146641018300346?via%7B%5C%%7D3Dihub>.
10. Adams, J. et al. Measurement of the $\nu\mu$energy spectrum with IceCube-79: IceCube Collaboration. Eur. Phys. J. C **77,** 692. ISSN: 14346052 (Oct. 2017).
11. Davis, R., Harmer, D. S. & Hoffman, K. C. Search for neutrinos from the sun. Phys. Rev. Lett. **20,** 1205–1209. ISSN: 00319007 (May 1968).
12. Collaboration, S.-K. et al. Solar Neutrino Measurements in Super-Kamiokande-IV. arXiv: 1606.07538. <http://arxiv.org/abs/1606.07538> (June 2016).

13. Hirata, K. S. et al. Observation in the Kamiokande-II detector of the neutrino burst from supernova SN1987A. Phys. Rev. D **38,** 448–458. ISSN: 05562821 (July 1988).

14. Bionta, R. M. et al. Observation of a neutrino burst in coincidence with supernova 1987A in the Large Magellanic Cloud. Phys. Rev. Lett. **58,** 1494–1496. ISSN: 0031-9007 (Apr. 1987).

15. Alexeyev, E., Alexeyeva, L., Krivosheina, I. & Volchenko, V. Detection of the neutrino signal from SN 1987A in the LMC using the INR Baksan underground scintillation telescope. Phys. Lett. B **205,** 209–214. ISSN: 0370-2693 (Apr. 1988).

16. Fernandez Moroni, G. et al. Charge coupled devices for detection of coherent neutrino-nucleus scattering. Phys. Rev. D - Part. Fields, Gravit. Cosmol. **91,** 1–9. ISSN: 15502368 (2015).

17. Dore, U., Loverre, P. & Ludovici, L. History of accelerator neutrino beams. arXiv Accel. Phys. doi:arXiv:1805.01373v1. arXiv: 1805.01373. <http://arxiv.org/abs/1805.01373> (May 2018).

18. Acciarri, R. et al. Long-Baseline Neutrino Facility (LBNF) and Deep Underground Neutrino Experiment (DUNE) Conceptual Design Report Volume 1: The LBNF and DUNE Projects. arXiv: 1601.05471. <http://arxiv.org/abs/1601.05471> (Jan. 2016).

19. Akimov, D. et al. COHERENT 2018 at the Spallation Neutron Source. arXiv: 1803.09183. <http://arxiv.org/abs/1803.09183> (Mar. 2018).

20. Akimov, D. et al. Observation of coherent elastic neutrino-nucleus scattering. Science (80-.). **357,** 1123–1126. ISSN: 10959203 (Sept. 2017).

21. Efremenko, Y. & Hix, W. R. Opportunities for neutrino physics at the Spallation Neutron Source (SNS) in J. Phys. Conf. Ser. **173** (Nov. 2009). ISBN: 1742-6588. doi:10.1088/1742-6596/173/1/012006. arXiv: 1211.5199. <http://arxiv.org/abs/1211.5199>.

22. Akimov, D. et al. Observation of coherent elastic neutrino-nucleus scattering. Science (80-.). **357,** 1123–1126. ISSN: 10959203 (Aug. 2017).

23. Ludhova, L. & Zavatarelli, S. Studying the earth with geoneutrinos Oct. 2013. doi:10.1155/2013/425693. arXiv: 1310.3961. <http://arxiv.org/abs/1310.3961%20http://dx.doi.org/10.1155/2013/425693>.

24. Agostini, M. et al. Spectroscopy of geoneutrinos from 2056 days of Borexino data. Phys. Rev. D - Part. Fields, Gravit. Cosmol. **92.** ISSN: 15502368. doi:10.1103/PhysRevD.92.031101. arXiv: 1506.04610. <http://arxiv.org/abs/1506.04610%20http://dx.doi.org/10.1103/PhysRevD.92.031101> (June 2015).

25. Araki, T. et al. Experimental investigation of geologically produced antineutrinos with KamLAND. Nature **436,** 499–503. ISSN: 00280836 (July 2005).

26. Pontecorvo, B. Mesonium and Antimesonium. Sov. Phys. JETP **33,** 549–551 (1957).

27. Fukuda, Y. et al. Evidence for oscillation of atmospheric neutrinos. Phys. Rev. Lett. **81,** 1562–1567. ISSN: 10797114 (Aug. 1998).

28. Ahmad, Q. R. et al. Measurement of the Rate of ν e + d $\rightarrow$ p + p + e $-$ Interactions Produced by B 8 Solar Neutrinos at the Sudbury Neutrino Observatory. Phys. Rev. Lett. **87,** 071301. ISSN: 0031-9007 (July 2001).

29. Freedman, D. Z. Coherent effects of a weak neutral current. Phys. Rev. D **9,** 1389–1392. ISSN: 05562821 (Mar. 1974).

30. Scholberg, K. Prospects for measuring coherent neutrino-nucleus elastic scattering at a stopped-pion neutrino source. Phys. Rev. D **73,** 033005. ISSN: 1550-7998 (Feb. 2006).

31. Patton, K., Engel, J., McLaughlin, G. C. & Schunck, N. Neutrino-nucleus coherent scattering as a probe of neutron density distributions. Phys. Rev. C - Nucl. Phys. **86,** 024612. ISSN: 1089490X (Aug. 2012).

32. Tanvir, N. R. et al. A 'kilonova' associated with the short-duration γ-ray burst GRB 130603B. Nature **500,** 547–549. ISSN: 0028-0836 (Aug. 2013).

33. Dutta, B., Mahapatra, R., Strigari, L. E. & Walker, J. W. Sensitivity to Z -prime and nonstandard neutrino interactions from ultralow threshold neutrino-nucleus coherent scattering. Phys. Rev. D **93,** 013015. ISSN: 24700029 (Jan. 2016).

34. Adhikari, R. et al. A white paper on keV sterile neutrino dark matter Jan. 2017. doi:10.1088/1475-7516/2017/01/025. arXiv: 1602.04816. <http://stacks.iop.org/1475-7516/2017/i=01/a=025?key=crossref.bada2c09a5e03691e66d8233961ff157>.

35. Collaboration, T. D. et al. DAMIC: a novel dark matter experiment. arXiv: 1310.6688. <http://arxiv.org/abs/1310.6688> (Oct. 2013).

36. Collaboration, C. et al. The CONNIE experiment. arXiv: 1608.01565. <http://arxiv.org/abs/1608.01565> (Aug. 2016).

37. Aguilar-Arevalo, A. et al. Results of the engineering run of the coherent neutrino nucleus interaction experiment (CONNIE). J. Instrum. ISSN: 17480221. doi:10.1088/1748-0221/11/07/P07024. arXiv: 1604.01343 (2016).

38. Haro, M. S. Sensores Multipixel CCD de Ultra Bajo Ruido de Lectura para Detección de Partículas PhD (Centro Atómico Bariloche, 2017).

39. Chavarria, A. E. et al. DAMIC at SNOLAB. Phys. Procedia **61,** 21–33. ISSN: 18753892 (Jan. 2015).

40. Deutsch, M. et al. X-ray spectrometry of copper: New results on an old subject. J. Res. Natl. Inst. Stand. Technol. **109,** 75. ISSN: 1044677X (2004).

41. Canali, C., Martini, M., Ottaviani, G. & Quaranta, A. A. Measurements of the average energy per electron-hole pair generation in silicon between 5-320°K. IEEE Trans. Nucl. Sci. **19,** 9–19. ISSN: 15581578 (Aug. 1972).

42. Holland, S. E., Groom, D. E., Palaio, N. P., Stover, R. J. & Wei, M. Fully depleted, back-illuminated charge-coupled devices fabricated on high-resistivity silicon. IEEE Trans. Electron Devices **50,** 225–238. ISSN: 00189383 (Jan. 2003).

43. Estrada, J., Molina, J., Blostein, J. J. & Fernández, G. Plasma effect in silicon charge coupled devices (CCDs). Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip. **665,** 90–93. ISSN: 01689002 (Feb. 2011).

44. Hanisch, R. J. et al. Definition of the Flexible Image Transport System (FITS). Astron. Astrophys. **376,** 359–380. ISSN: 0004-6361 (2001).

45. Vergara, B. A. C. Estudio de Los Retrocesos de Núcleos de Si de un CCD Debidos a la Dispersión de Neutrones, Considerando Efectos de "Channeling" (2018).

46. Brun, R. & Rademakers, F. ROOT — An object oriented data analysis framework. Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip. **389,** 81–86. ISSN: 0168-9002 (Apr. 1997).

47. Bertin, E. & Arnouts, S. SExtractor User Manual — SExtractor 2.24.2 documentation <https://sextractor.readthedocs.io/en/latest/> (2018).

48. Holwerda, B. W. Source Extractor for Dummies v5. ISSN: 0365-0138. doi:10.1051/aas:1996164. arXiv: 0512139 [astro-ph]. <http://arxiv.org/abs/astro-ph/0512139> (2005).

49. Bertin, E. Automated Morphometry with SExtractor and PSFEx. Analysis **442,** 435–438 (2011).

50. Sérsic, J. L. Influence of the atmospheric and instrumental dispersion on the brightness distribution in a galaxy 1963. <https://ui.adsabs.harvard.edu/%7B%5C#%7Dabs/1963BAAA...6...41S>.

51. <https://github.com/iru-/sextractor> (2019).

52. Python Language Reference; version 2.7 Python Software Foundation. <https://www.python.org/> (2018).

53. Collaboration, T. A. et al. The Astropy Project: Building an inclusive, open-science project and status of the v2.0 core package. doi:10.3847/1538-3881/aabc4f. arXiv: 1801.02634. <http://arxiv.org/abs/1801.02634%20http://dx.doi.org/10.3847/1538-3881/aabc4f> (Jan. 2018).

54. Hunter, J. D. Matplotlib: A 2D Graphics Environment. Comput. Sci. Eng. **9,** 90–95. ISSN: 1521-9615 (2007).

55. Brun, R. & Rademakers, F. ROOT — An object oriented data analysis framework. Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip. **389,** 81–86. ISSN: 01689002 (Apr. 1997).