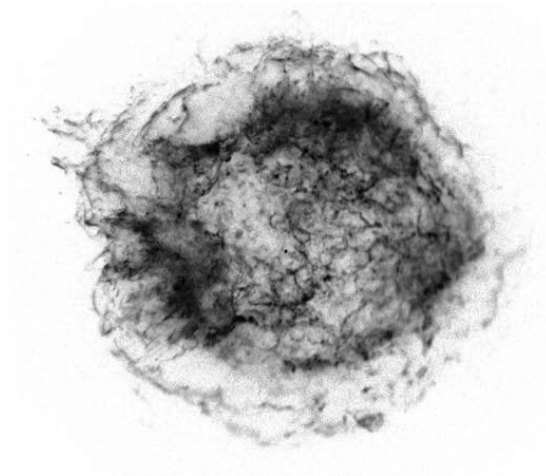


Київський національний університет імені Тараса Шевченка
Головна астрономічна обсерваторія НАН України

Бабик Ю.В., Вовк Є.Ю.

**ВІРТУАЛЬНА РЕНТГЕНІВСЬКА
І ГАММА-ОБСЕРВАТОРІЯ**
Обробка даних космічних обсерваторій
“Чандра” та “Фермі”



Київ, 2012

**Київський національний університет імені Тараса Шевченка
Головна астрономічна обсерваторія НАН України**

Бабик Ю. В., Вовк Є. Ю.

**Віртуальна рентгенівська і
гамма-обсерваторія.
Обробка даних космічних обсерваторій
“Чандра” та “Фермі”**

Методичний посібник для студентів фізичних факультетів
з обробки даних космічних обсерваторій “Чандра” та “Фермі”

Київ, 2012

УДК 524.7

Бабик Ю. В., Вовк Є. Ю. Віртуальна рентгенівська та гамма-обсерваторія. Обробка даних космічних обсерваторій “Чандра” та “Фермі”. Методичний посібник К.: ВАІТЕ. – 64 с.

В посібнику подано основні відомості щодо методів опрацювання даних космічних обсерваторій “Чандра” та “Фермі”. Приведено особливості аналізу даних на прикладі окремих галактик та скупчень галактик.

Посібник буде корисний для широкого кола фахівців при роботі в середовищах цих космічних телескопів.

Офіційні рецензенти:

Івченко Василь Миколайович, доктор фіз.-мат. наук, проф., завідувач кафедри астрономії та фізики космосу фізичного факультету Київського національного університету імені Тараса Шевченка.

Елійв Андрій Андрійович, кандидат фіз.-мат. наук, н.с. лабораторії астроінформатики ГАО НАН України.

Затверджено на засіданні вченої ради Головної астрономічної обсерваторії НАН України 1 березня 2012 р. (протокол # 5).

© Бабик Ю. В., Вовк Є. Ю., 2012.

Зміст

Вступ.....	4
1 Обробка спостережень обсерваторії “Чандра”	5
1.1 Історія створення космічної обсерваторії “Чандра”	5
1.2 Отримання даних та первинна обробка у пакеті CIAO .	9
1.3 Робота із зображеннями	11
1.3.1 Побудова кривої блиску	11
1.3.2 Пошук точкових джерел на матриці	14
1.3.3 Побудова карти експозиції	17
1.3.4 Отримання та моделювання профілю поверхне- вої яскравості	24
1.3.5 Створення зображень	30
1.4 Робота зі спектрами	32
1.5 Команди і додаткові посилання	38
1.6 Командні файли	39
2 Обробка спостережень супутника Fermi/LAT	43
2.1 Загальні відомості про телескоп Fermi/LAT	43
2.2 Завантаження даних	44
2.3 Застосування фільтрів	45
2.4 Створення зображень	47
2.5 Побудова кривої блиску	47
2.6 Створення спектру	51
Заключні зауваження.....	62
Література.....	63

Вступ

Багато відкриттів сучасної астрономії пов'язані з використанням спостережень у високоенергетичних діапазонах випромінювання. В астрофізиці високих енергій розглядається, головним чином, короткохвильова частина електромагнітного спектру: рентгенівське і гамма-випромінювання, а також нейтрино і космічні промені. Чим вище діапазон енергій, тим важче у ньому спостерігати, і тим менше можна виявити джерел на небі. Тому об'єктами дослідження саме рентгенівської астрономії є набагато більше джерел, ніж гамма-астрономії та нейтринної астрономії. У жорстких діапазонах випромінюють, в основному, акреційні диски і джети компактних об'єктів, пульсари і залишки Наднових. Рентгенівське та гамма випромінювання не проходить крізь атмосферу Землі, тому астрономічні рентгенівські та гамма джерела спостерігають з навколоземної орбіти. Перші рентгенівські джерела були відкриті у 1967 р. за допомогою балістичних ракет з рентгенівськими детекторами. Перший каталог рентгенівських джерел був зроблений за допомогою супутника "Ухуру", запущеного на орбіту у 1970 р. Цей каталог містив понад 300 джерел. Перший огляд всього неба в рентгенівському діапазоні був зроблений за допомогою космічного телескопа "ROSAT", який працював з 1990 по 1999 рр., і містить понад 150000 джерел в діапазоні до 2 кеВ. Зараз на орбіті працюють багато спеціалізованих рентгенівських телескопів (наприклад, "ХММ-Ньютон", "Чандра", "Suzaku" та ін.).

Найкращі космічні гамма-телескопи - це обсерваторія "Комптон", що працювала з 1991 по 2000 роки і обсерваторія "Фермі", що прийшла їй на зміну у 2008 р. Ці обсерваторії обладнані твердотільними детекторами, що реєструють частинки з енергіями від 20 МеВ до 300 ГеВ. Фокусування гамма-променів неможливе. Космічні гамма-обсерваторії можуть спостерігати всього кілька сотень джерел, але всі вони є дуже цікавими з наукової точки зору.

Спостереження сучасних телескопів зберігаються у електронній формі і часто надаються для широкого кола наукової спільноти, що дозволяє отримати більше наукових результатів. Міжнародний проект з розповсюдження астрономічних даних спостережень через інтернет називається "Віртуальна обсерваторія". Серед сервісів віртуальної обсерваторії слід назвати UkrVO, що була розроблена Українською астрономічною асоціацією, а також архіви наземного огляду "SDSS" і космічних телескопів ім. Габбла і "SOHO". В 2006

році в Києві була створена Українська віртуальна рентгенівська і гамма-обсерваторія VIRGO.UA. Ініціаторами її роботи стали Науковий центр обробки даних гамма-супутника INTEGRAL (Швейцарія), Київський національний університет імені Тараса Шевченка (КНУ) і Інститут теоретичної фізики ім. М. М. Боголюбова НАН України (ІТФ) (Київ). Зараз до складу VIRGO.UA входять 3 комп'ютерні класи: на фізичному факультеті КНУ, в ІТФ і в Головній астрономічній обсерваторії НАН України. На VIRGO.UA проводиться обробка спостережень даних гамма-супутників "Fermi", "INTEGRAL" і "Swift" та рентгенівських супутників "ХММ-Ньютон", "Чандра" і "RXTE". Додаткову інформацію можна знайти на сайті VIRGO.UA [7].

1 Обробка спостережень обсерваторії "Чандра"

1.1 Історія створення космічної обсерваторії "Чандра"

Рентгенівська астрономія бере свій початок із космічних апаратів, що виявляли місцеположення ядерних вибухів на Землі; тепер вона допомагає відкривати таємниці Всесвіту. Атмосфера надійно захищає поверхню нашої планети від короткохвильового електромагнітного випромінювання, тому реєстрація рентгенівських квантів неземного походження можлива лише на великих висотах. Рентгенівське випромінювання самих високих енергій можна віднайти за допомогою інструментів, розміщених на борту стратостатів, проте увесь рентгенівський діапазон відкривається лише при винесенні апаратури безпосередньо в космічний простір.

Перші експерименти такого плану були зроблені в кінці 1940-х років в США під егідою Пентагона. Працівник вашингтонської Військово-морської дослідницької лабораторії Герберт Фрідман побудував модифікований лічильник Гейгера, який детектував кванти високої енергії і тому міг виявити атмосферний ядерний вибух (це і було основною ціллю). В 1949 році Фрідман та його колеги почали відправляти нові лічильники в космос в носових конусах німецьких ракет "Фау-2". Відразу ж перший запуск приніс важливу інформацію — виявилось, що Сонце випускає рентгенівські промені. У 1964 році Фрідман та його співробітники першими виявили рентгенівське випромінювання Крабоподібної туманності, що виникло на місці вибуху наднової в 1054 році.

Такого роду лічильники встановлювались на геофізичних раке-

тах, проте для систематичного пошуку космічних рентгенівських джерел необхідні довготривалі неперервні спостереження. Тому на початку 1960-х років декілька американських астрофізиків дійшли висновку, що детектори рентгенівського випромінювання потрібно встановлювати на супутникових платформах. Серед них був випускник Міланського університету доктор Рікардо Джиакконі. В 1962 році його група відкрила Скорпіон X-1 — перше рентгенівське джерело за межами Сонячної системи. До 1966 року експерименти на ракетах та повітряних кулях дозволили астрофізикам із США та Англії виявити більше 30 таких джерел. Це переконало правління NASA в доцільності запуску рентгенівського супутника. В проєкті він називався “X-Ray Explorer”, проте після виходу на орбіту був перейменований в “Ухуру” (“свобода” - на мові суахілі). Він був запущений 12 грудня 1970 року, в день незалежності Кенії, з платформи поблизу її берегів. Ухуру пропрацював близько двох з половиною років, а отримані за його допомогою дані стали джерелом багатьох відкриттів. Цей апарат зареєстрував 339 рентгенівських джерел, в тому числі і об’єкт в сузір’ї Лебедя, який став першим в історії астрономії претендентом на роль чорної діри [6, 4, 5].

Ухуру був рентгенівським супутником, проте ніяк не телескопом. Будь-який такого типу інструмент повинен мати оптичну систему, що здатна збирати і фокусувати випромінювання, що проходить через апертуру. Нічого подібного на Ухуру не було. Все фокусування полягало в тому, що рентгенівські промені перед попаданням в газорозрядні детектори (фактично ті ж лічильники Гейгера) проходили через вузьку щілину-коліматор. Ця апаратура добре реєструвала точкові джерела, проте не була придатна для зйомки протяжних об’єктів. Джиакконі це прекрасно розумів. Ще в 1960 році він у співавторстві з американським визначним спеціалістом по космічним променям Бруно Россі опублікував статтю, що містила першу в світі принципову схему рентгенівського телескопа з фокусувальною дзеркальною системою. Побудована вона була нестандартно, так як рефлектори звичайних “оптичних” телескопів відбивають промінь, що падає під будь-яким кутом, причому вони створюють зображення з меншою аберацією, якщо цей кут не дуже великий. Тому головне дзеркало такого телескопа відкидає пучок світла назад до апертури. З рентгенівськими квантами ситуація дещо складніша: через свою високу енергію вони практично не заломлюються в речовині (а значить, неможливо створити рентгенівські “лінзи”) та не відбиваються, а поглинаються при будь-яких кутах падіння, окрім самих великих

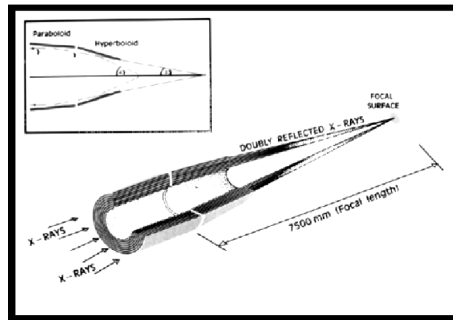


Рис. 1: Система дзеркал для рентгенівського телескопа ХММ-Ньютон.

(приблизно 90 градусів). Тому для створення ефективного рентгенівського дзеркала промені, що надходять на дзеркало, повинні приходити майже паралельно до відбиваючої поверхні. Дана система дзеркал показана на Рис. 1.

Розрахунки показали, що дзеркало для рентгенівського телескопа повинно мати форму параболічної чи гіперболічної труби, що звужується до одного кінця. Рентгенівський промінь проходить всередину з ширшої сторони і після відбиття виходить з вузького кінця. Ще в 1952 році фізик-оптик Ганс Волтер (H. Wolter) відмітив, що для потрібного фокусування необхідні дві послідовно встановлені відбиваючі поверхні. В свою чергу Джиаконні та Россі зрозуміли, що для збільшення чутливості телескопа кожний фокусуєчий блок повинен мати декілька вкладених одна в одну трубо-подібних дзеркал з одною центральною віссю.

Джиаконні разом з фахівцями у рентгенівській астрономії думав про великий рентгенівський телескоп ще до запуску "Ухуру". В травні 1970 року в NASA був направлений проект інструмента з 5 парами дзеркал діаметром до 105 сантиметрів. Однак у Вашингтоні вирішили зберегти кошти і затвердили більш скромний варіант з восьми дзеркал. Спочатку його назвали "HEAO-2" (High Energy Astronomy Observatory-2), а після запуску — на честь А. Ейнштейна. Саме ця обсерваторія, яка була запущена 13 листопада 1987 року та проработала до квітня 1991 року, і стала першим рентгенівським телескопом. "Ейнштейн" мав роздільну здатність близько 5 кутових секунд і реєстрував рентгенівські кванти в діапазоні 200 eV – 20 keV. Ця станція вперше здійснила високоякісне спектрографування залишків надно-

вих і відкрила велику кількість дуже слабких позагалактичних джерел рентгенівського випромінювання.

В 1980-90 роки деякі європейські країни, СРСР та Японія відправили в космос немало рентгенівських телескопів та супутників (самим потужним був німецький ROSAT з 80-см дзеркалами, що діяв у 1990-99 роках). Однак жоден з них не зміг достатньо покращити якість спостережень, зроблених “Ейнштейном”. Астрономії ХХІ століття потрібен був інструмент, який мав би більш широкі можливості. Саме тому в 1976 році Джаіаконні та його колега Харві Тананбаум направили в NASA нові пропозиції з цього приводу. Після цього Вашингтон виділив ресурси на їх дослідження, і проект набрав офіційного статусу. Майбутню обсерваторію назвали “AXAF” (Advanced X-Ray Astrophysics Facility). Задумка була грандіозною — телескоп з 6 парами дзеркал з максимальним діаметром 120 см, здатний реєструвати в сто раз менш яскраві рентгенівські джерела, ніж “Ейнштейн”. Контракт на виготовлення телескопа було затверджено лише в 1988 році. NASA остаточно затвердила скорочений проект лише в 1992 році, а після 4 років оптична система телескопа була змонтована і відправлена для перевірки в Центр космічних польотів імені Маршалла. Калібрування та випробування продовжувалися півроку, і було підтверджено, що дзеркала та реєструюча апаратура була зроблена ідеально і всі модулі повністю готові до роботи. До речі, ретельність виготовлення дзеркал цього телескопа досі залишається неперевершеною: похибка полірування не перевищує декількох розмірів атомів, а загальна точність позиціонування, майже триметрових, вкритих іридієм, площин, складає 1.3 мкм. Назву телескопу вирішили обрати на відкритому конкурсі, що зібрав майже 6 тисяч заявок із 61 країни. Переможцями виявилися школяр із штату Айдахо Тайлер Джонсон і каліфорнійська вчителька Джатіла ван дер Він. Саме вони запропонували назвати обсерваторію “Чандрою” на честь одного з найбільших астрофізиків ХХ століття Субрахманьяна Чандрасекара.

До моменту запуску “Чандри” на орбіті “трудилось” (або вже не функціонували) два десятки космічних апаратів, призначених для реєстрації рентгенівського випромінювання. Однак нова обсерваторія відразу ж довела, що вона може “працювати” як мінімум в 50 разів краще за своїх попередників. Її камера високої роздільної здатності може отримувати якісні зображення з точністю до 0.5 кутової секунди. І в цьому “Чандра” до цих пір не має аналогів (наприклад, європейський рентгенівський телескоп “ХММ-Ньютон”, запускений 4

місяцями пізніше має роздільну здатність в 5-14 кутових секунд).

Навіщо взагалі потрібна рентгенівська астрономія? І в нашій Галактиці, і в дальньому космосі є багато об'єктів настільки гарячих, що максимум їх випромінювання знаходиться в рентгенівському діапазоні. Такими є, наприклад, рентгенівські пульсари — нейтронні зорями з сильними магнітними полями, кожен з яких має в компаньйонах звичайну зорю. Газ, що перетікає з цієї зорі на темну нейтронну сусідку, нагрівається до сотень мільйонів градусів і випромінює рентгенівські кванти. Потужне рентгенівське випромінювання виникає і при попаданні речовини на чорну діру. Ще один цікавий об'єкт дослідження — рентгенівські барстери, космічні рентгенівські спалахи, які виникають при безпосередній участі чорних дір та нейтронних зір. Джерелом рентгенівського випромінювання може стати і гарячий рентгенівський міжгалактичний газ, і зорі будь-якого типу, в тому числі самі малопотужні коричневі карлики, і навіть планети. Тому зрозуміло, що без спостереження неба в рентгенівському діапазоні наші астрономічні знання будуть дуже обмеженими [1].

1.2 Отримання даних та первинна обробка у пакеті CIAO

Дані для обробки з обсерваторії “Чандра” знаходяться в електронній базі даних HEASARC [8]. Ця база містить дані, що були отримані багатьма обсерваторіями та телескопами. Для отримання даних потрібно лише вказати назву об'єкта в розділі Archive даного сайту та обрати назву телескопа, з якого ви плануєте замовити готові дані (в нашому випадку Chandra). Після цього необхідно вибрати робочу папку, в якій буде проводитись обробка, і завантажити в неї дані. Варто відмітити, що можна використовувати лише ті дані, що знаходяться більше 1 року в архіві, так як до новіших даних, доступу немає. В даному розділі буде представлено послідовність обробки на прикладі скупчення галактик A 2124.

Перш за все, потрібно провести первинну обробку даних, котра розпочинається з розархівування даних. В операційній системі Linux цю процедуру, як і багато інших, зручно виконати у командному рядку (терміналі) наступною командою:

```
tar -xf name.tar
unix% tar -xf w3browse-173292.tar
```

де `w3browse-173292.tar` – назва нашого архіву. Після виконання да-

ної операції подивимось, що ми отримали, за допомогою команди `ls`:

```
unix% ls
3238 cda.harvard.edu w3browse-173292.tar
```

Як бачимо, маємо три файли, точніше дві папки і наш попередній архів. Отож, надалі нам потрібна лише папка з цифрами, в даному випадку 3238 (номер спостереження даного об'єкта на телескопі Чандра, іншими словами `ObsID` даного об'єкта.) Перейти в дану папку можна за допомогою команди `cd`:

```
unix% cd 3238/
```

Знову використаємо `ls` для того, щоб переглянути її вміст:

```
unix% ls
oif.fits primary secondary
```

Там буде дві папки і файл `oif.fits`, який нам не потрібний. На відміну від обробки даних ХММ-Ньютон, для обсерваторії Чандра не потрібно виконувати калібровку файлів, так як в папці `primary` усі файли відкалібровані і приведені до нормального вигляду, тобто їх можна використовувати відразу. Окрім цього, папка `primary` містить основні файли, які потрібні для роботи в першу чергу. В папці `secondary` знаходяться допоміжні дані. Ми будемо використовувати папку `primary`, в котру перейдемо за допомогою команди `cd` і проглянемо її вміст за допомогою команди `ls`:

```
unix% cd primary
unix% ls
acisf03238_000N002_bpix1.fits.gz
acisf03238N002_evt2.fits.gz
acisf03238_000N002_fov1.fits.gz
acisf03238N002_full_img2.fits.gz
acisf03238N002_1_sum2.html
acisf03238N002_full_img2.jpg
acisf03238N002_1_sum2.ps
acisf03238N002_src2.fits.gz
acisf03238N002_2_sum2.html
acisf03238N002_src_img2.jpg
acisf03238N002_3_sum2.html
orbitf138024300N001_eph1.fits.gz
acisf03238N002_cntr_img2.fits.gz
pcadf138312521N002_asol1.fits.gz
acisf03238N002_cntr_img2.jpg
```

Бачимо безліч архівованих даних, котрі для початку роботи потрі-

бно розпакувати. Зробити це можна за допомогою команди `gunzip *.gz`, а для перегляду результату використаємо команду `ls`:

```
unix% gunzip *.gz
unix% ls
acisf03238N000_bpix1.fits
acisf03238N002_3_sum2.html
acisf03238N002_full_img2.jpg
acisf03238N000_fov1.fits
acisf03238N002_cntr_img2.fits
acisf03238N002_src2.fits
acisf03238N002_1_sum2.html
acisf03238N002_cntr_img2.jpg
acisf03238N002_src_img2.jpg
acisf03238N002_1_sum2.ps
acisf03238N002_evt2.fits
orbitf138024300N001_eph1.fits
acisf03238N002_2_sum2.html
acisf03238N002_full_img2.fits
pcadf138312521N002_asol1.fits
```

1.3 Робота із зображеннями

Відразу варто сказати про те, що для точкових джерел та протяжних аналіз в **STAO** проводиться по-різному. Обробка даних точкових джерел є складним процесом, проте, ще складніше обробляти дані протяжних об'єктів, таких, як скупчення галактик або залишки наднових. Процес ускладнюється через просторові зміни в детекторах. Велика кількість різного роду завдань з протяжними джерелами є непотрібними при аналізі точкових. Тому в даному розділі буде представлено методи обробки даних для протяжних джерел; приклади представлені для архівних даних **Chandra ACIS** [1].

1.3.1 Побудова кривої блиску

Для побудови зображень, кривої блиску і спектра, а саме це є нашою метою, нам потрібен лише один файл з усіх вище наведених. Основним є файл з суфіксом `evt2.fits`, котрий використовується для всіх можливих аналізів цих даних. Тепер поглянемо на саме зображення нашого скупчення. Для цього використовують програмне

забезпечення CIAO. Щоб його підключити, потрібно ініціалізуватись (запустити програмне забезпечення), тобто проробити наступні команди (якщо ви працюєте на фізичному факультеті КНУ імені Тараса Шевченка):

```
unix% csh
unix% source /virgo/scripts/login_ciao.csh
CIAO configuration is complete...
CIAO 4.2 Monday, November 30, 2009
bindir : /home/virgo/software/CIAO-4.2-x86_64/bin
```

Це означає, що програмне забезпечення CIAO підключено і тепер можливе його використання. Отож, подивимось на скупчення A 2124, для цього використаємо програмний пакет ds9, який запускається наступним чином:

```
unix% ds9 acisf03238N002_evt2.fits
```

На Рис. 2 наведено приклад отримано зображення A 2124.

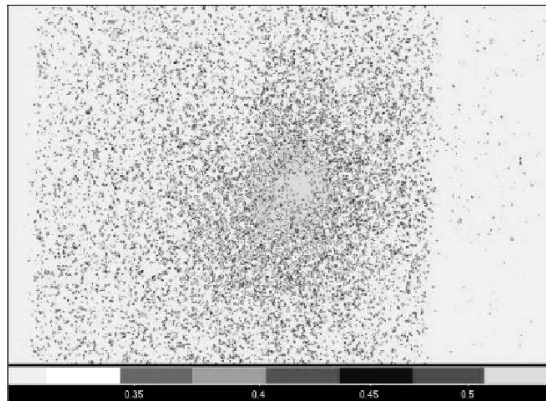


Рис. 2: Зображення 3C295 з розміром 512 на 512 пікселів.

Спочатку потрібно задати регіон, для якого буде проводитися аналіз. Наводимо курсор миші на зображення і при натиску лівою кнопкою миші з'явиться круг, а при подвійному натиску, можна переглянути координати центра в різних одиницях або зробити їх та радіус круга таким, яким потрібно в залежності від завдання. Зберігаємо регіон: знаходимо вкладку **Region**, вибираємо **Save** і зберігаємо під іменем A2124.reg. Натиснувши двічі по регіону, ви побачите

таблицю з координатами центра та радіусом круга у фізичних координатах телескопа. Радіус нам потрібен для створення аналогічного за розміром регіону для фону зображення. В фізичних координатах радіуси регіонів будуть складати 225 одиниць. Назва фонового файлу — `A2124_bkg.reg`. Тепер вікно `ds9` можна закривати і перейти до консолі. Проглянемо, що було створено нового:

```
unix% ls
A2124_bkg.reg
A2124.reg
acisf03238N002_2_sum2.html
acisf03238N002_full_img2.jpg
acisf03238N002_3_sum2.html
acisf03238N002_src2.fits
acisf03238_000N002_bpix1.fits
acisf03238N002_cntr_img2.fits
acisf03238N002_src_img2.jpg
acisf03238_000N002_fov1.fits
acisf03238N002_cntr_img2.jpg
orbitf138024300N001_eph1.fits
acisf03238N002_1_sum2.html
acisf03238N002_evt2.fits
pcadf138312521N002_asol1.fits
acisf03238N002_1_sum2.ps
acisf03238N002_full_img2.fits
```

З'явилися наші регіони `A2124_bkg.reg` та `A2124.reg`. Тепер перейдемо до кривої блиску. Вона буде створена за допомогою команди `dmextract` наступним чином:

```
unix% dmextract infile="acisf03238N002_evt2.fits \
[sky=region(A2124.reg)][bin time=:10]"
outfile=ltrcv.fits bkg="acisf03238N002_evt2.fits \
[sky=region(A2124_bkg.reg)]" opt=lrc1
```

Зверніть увагу, що все прописується через пробіл. Тут “\” позначає продовження рядка, `infile = “acisf03238...fits”` — назва вхідного файлу, в якому вказуємо назву регіону, для якого будемо визначати криву блиску, та кількість бінів при групуванні (для даного випадку було взято 10), `outfile=ltrcv.fits` — назва вихідного файлу. `bkg=“acisf03238...fits”` — команда для віднімання фону (необхідність виконання даної процедури визначається в залежності від задачі), `opt=lrc1` — допоміжна функція, якою сповіщається

програмному забезпеченню, що потрібно створити саме криву блиску (`ltc1`), так як дана команда використовується і для інших задач. Переглянути криву блиску можна за допомогою програмних пакетів `fv` або `Prism`. Це програми для перегляду та побудови `fits`-файлів. В даному випадку використовується `Prism`: відкриваємо у вікні `Prism` створений файл `ltsrv.fits` і будуємо криву блиску.

1.3.2 Пошук точкових джерел на матриці

При спостереженні протяжних джерел, на матриці часто міститься велика кількість точкових об'єктів, не пов'язаних безпосередньо з протяжними. Хоча такі об'єкти науково можуть бути цікаві самі по собі, проте вони можуть бути джерелом складності при аналізі дифузного випромінювання протяжних джерел. Тому при аналізі того чи іншого об'єкта є корисним відфільтрувати зображення на матриці від точкових джерел, наприклад, при визначенні радіального профілю поверхневої яскравості, щоб уникнути помилкових різких розривів. Також це потрібно при фітуванні спектрів різними моделями, так як це дозволить уникнути зсувів тієї чи іншої моделі від спостережуваного спектру. Видалення точкових джерел з файлу подій для будь-якого з цих завдань є дуже простим при використанні команди `dmcopy` у середовищі `CIAO`. Однак, перший крок полягає саме у виявленні цих джерел. Пакет `CIAO Detect` містить дві команди, які забезпечують можливість роботи в складних полях з точковими та протяжними об'єктами: `vtpdetect` та `wavdetect`. Проведемо наліз об'єкта на прикладі другої команди.

`Wavdetect` працює у два етапи. Спочатку він виявляє можливі пікселі джерел в наборі даних. Пікселі з досить великим позитивним значенням кореляції видаляються із зображення як передбачувані джерела, а наступні співвідношення виконуються в одному масштабі. На другому етапі генерується вихідний список з першого етапу в кожному масштабі. Ця команда відділяє близько розташовані точкові джерела і знаходить протяжні (див. Рис.3). Для отримання використовується лише файл із суфіксом `evt2`. Приклад показаний для об'єкта `3C295` (`ObsID 578`). Через велике навантаження дану команду рекомендується запускати для зображення розміром не більше 1024×1024 пікселів.

```
unix% dmcopy "acisf00578N003_evt2.fits[ccd_id=7] \  
[bin x=3848:4360:1.0,y=3860:4372:1.0]" s3_img.fits
```

Для перегляду файлу потрібно використати `ds9`:

```
unix% ds9 s3_img.fits &
```

Для запуску `wavdetect` з базовим набором параметрів необхідно ли-

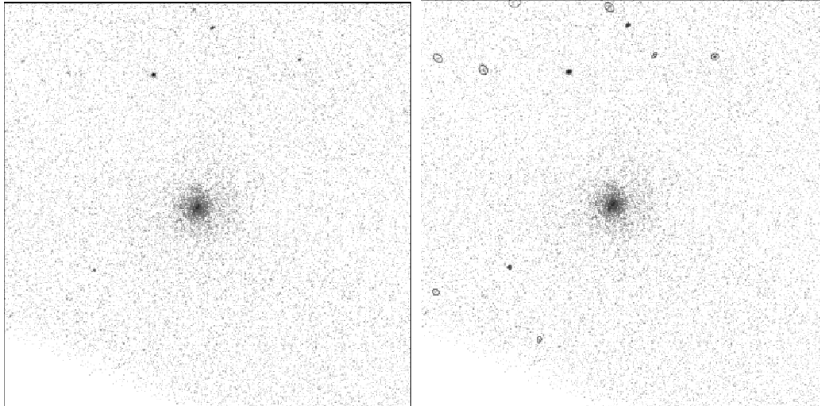


Рис. 3: Зображення 3C295 розміром 512 на 512 пікселів без (ліворуч) та з (праворуч) точковими джерелами.

ше вказати імена вхідних та вихідних файлів. Запускається дана команда наступним чином:

```
unix% punlearn wavdetect
unix% pset wavdetect infile=s3_img.fits
unix% pset wavdetect outfile=s3_src.fits
unix% pset wavdetect scellfile=s3_scell.fits
unix% pset wavdetect imagefile=s3_imgfile.fits
unix% pset wavdetect defnbgfile=s3_nbgd.fits
unix% pset wavdetect regfile=s3_src.reg
unix% wavdetect
Input file name (s3_img.fits):
Output source list file name (s3_src.fits):
Output source cell image file name (s3_scell.fits):
Output reconstructed image file name (s3_imgfile.fits):
Output normalized background file name (s3_nbgd.fits):
```

Для відображення вихідної картинки застосуємо `ds9`:

```
unix% ds9 s3_img.fits &
```


Також дану процедуру можна виконувати при наявності карти експозиції (exposure map):

```
unix% punlearn wavdetect
unix% pset wavdetect infile=s3_img.fits
unix% pset wavdetect outfile=s3_expmap_src.fits
unix% pset wavdetect scellfile=s3_expmap_scell.fits
unix% pset wavdetect imagefile=s3_expmap_imgfile.fits
unix% pset wavdetect defnbkgfile=s3_expmap_nbgd.fits
unix% pset wavdetect regfile=s3_expmap_src.reg
unix% pset wavdetect expfile=s3_0.71keV_expmap.fits
unix% wavdetect
Input file name (s3_img.fits):
Output source list file name (s3_expmap_src.fits):
Output source cell image file name \
(s3_expmap_scell.fits):
Output reconstructed image file name \
(s3_expmap_imgfile.fits):
Output normalized background file name \
(s3_expmap_nbgd.fits):
```

Щоб переглянути список усіх джерел потрібно використати команду dmlist:

```
unix% dmlist "s3_expmap_src.fits[cols RA,DEC,PSF_SIZE]" data
Data for Table Block SRCLIST.
  ROW  RA          DEC          PSF_SIZE
  ---  ---          ---          ---
  1    212.8558538789 52.1797387065 0.51768761873245
  2    212.8847083521 52.1878910087 0.61479145288467
  3    212.8640791885 52.1921762232 0.47023788094521
  4    212.8362796737 52.202181660  0.40190586447716
  5    212.8351087846 52.2029039026 0.40246158838272
  6    212.8475613372 52.2255903030 0.53206640481949
  7    212.8714240484 52.2259177730 0.62263220548630
  8    212.8842308305 52.227961680  0.75592726469040
  9    212.8067711419 52.2281781731 0.59245705604553
  10   212.8236615527 52.2284789414 0.55507236719131
  11   212.8310964079 52.2335976922 0.60252219438553
  12   212.8362660158 52.2366611370 0.65922313928604
  13   212.8626620537 52.2375255009 0.75592726469040
```

або це ж саме можна переглянути в Prism:

```
unix% prism s3_expmap_src.fits &
```

Для видалення цих джерел використовують команду `dmcopy` та параметр `exclude` у наступному вигляді

```
unix% dmcopу ‘‘acisf..._evt2.fits [exclude \  
sky=region(src.reg)]’’ without_ps_evt2.fits
```

де `acisf..._evt2.fits` — файл подій з якого потрібно видалити точкові джерела, `src.reg` — файл з списком точкових джерел, а назва вихідного файлу без точкових джерел — `without_ps_evt2.fits`.

1.3.3 Побудова карти експозиції

Визначення властивостей дифузного випромінювання від протяжних об’єктів може бути ускладнено просторовими варіаціями під час експозиції. Наприклад, детектор, як і програмний чіп, можуть створити “помилкове” просторове зображення в спостережуваному джерелі. Користувачі мають змогу вирішувати ці проблеми за допомогою карт експозицій. Карта експозиції для цього набору даних є ефективною площею при заданій позиції неба протягом спостереження.

Команда `mkeхrmap` створює карту експозиції, що може бути використана для перетворення відліків зображення в потік. Обчислення карти експозиції є ні що інше як образ ефективної площі при кожній позиції на небі. Розглянемо приклад. Для побудови карти експозиції потрібно наступні файли `evt2`, `asol1`, `msk1` та `pbk0`. Далі буде приведено приклад для побудови карти експозиції для об’єкта G21.5-09 (ObsID 1838).

Спочатку потрібно використати скрипт `fluximage`. При його запуску потрібно забезпечити тільки вхідний файл подій. Далі сам скрипт буде читати імена, пов’язані з даним файлом подій. Тому попередньо потрібно, щоб усі ці файли знаходились у одній папці.

В даному випадку скрипт було запущено для ACIS чіпу, `ccd_id=7` та діапазону енергій 0.5–7.0 кеВ:

```
unix% punlearn fluximage  
unix% pset fluximage infile= \  
"acisf01838N002_evt2.fits[ccd_id=7]"  
unix% pset fluximage outroot=flux  
unix% pset fluximage bands=broad  
unix% pset fluximage binsize=1
```

Запустивши даний скрипт з параметром `verbose=1`, можна спостерігати за самою побудовою карти експозиції в режимі он-лайн:

```

unix% fluximage verbose=1
Running fluximage...
Version: 29 September 2011
Input events or image files \
(acisf01838N002_evt2.fits[ccd_id=7]):
Output root of data products? (flux):
Image binning factor (0:) (1):
Energy bands, comma-separated list, min:max: \
center in keV
or ultrasoft, soft, medium, hard, broad, \
wide, CSC (broad):
Using CSC ACIS broad science energy band.
Bad pixel file acisf01838_000N002_bpix1.fits found.
Mask file acisf01838_000N002_msk1.fits found.
PBK file acisf084245776N002_pbk0.fits found.
Creating aspect histogram(s).
Creating binned images.
Creating instrument map(s).
Creating exposure map(s).
Exposure map limits: 0.000000e+00, 2.975940e+06
Writing exposure map to flux_7_2.3_bin1.exmap
Exposure-correcting image(s).
Cleaning up intermediary files.
The following files were created:
The binned counts image is:
    flux_0.5-7.0_bin1.img
The clipped counts image is:
    flux_0.5-7.0_bin1_thresh.img
The exposure map is:
    flux_2.3_bin1.exmap
The exposure-corrected image is:
    flux_0.5-7.0_2.3_bin1_flux.img

```

Або ж це все можна проробити по командам. Як вже зазначалось вище, використовуємо діапазон енергій 0.5–7.0 кеВ, тож створимо список подій в даному діапазоні, для цього використаємо команду `dmcopy`:

```

dmcopy "acisf01838N002_evt2.fits[energy=500:7000]" \
acis_1838_evt2.fits

```

Створимо зображення даного об'єкта, який у кінці буде нормований

картою експозиції, але перед цим нам потрібно знати список чіпів, що використовуються у спостереженні. Вони зберігаються в ключовому слові DETNAM в списку подій, тому:

```
unix% dmkeypar acis_1838_evt2.fits DETNAM echo+
ACIS-012367
```

У нашому випадку використовувались 6 чіпів для спостереження (0,1,2,3,6,7). Ми можемо знайти фізичні координати нашого джерела, використовуючи ds9 для візуалізації acis_1838_evt2.fits. Вони виявились близькими до $x = 4072$, $y = 4249$. Також ми можемо використати команду dmcoords для з'ясування чіпа, що знаходиться безпосередньо у цих координатах. Варто відмітити, що усі файли повинні бути введені або у вигляді списку або у вигляді таблиці. Далі робимо наступне:

```
unix% cat pcad_asol1.lis
pcadf084244404N002_asol1.fits
dmcoords acis_1838_evt2.fits asolfile=@pcad_asol1.lis
dmcoords>: sky 4072 4249
(RA,Dec):      18:33:33.594  -10:34:06.34
(RA,Dec):      278.38998    -10.56843 deg
THETA,PHI      1.267'      5.93 deg
(Logical):     4072.00     4249.00
SKY(X,Y):     4072.00     4249.00
DETX,DETY     4250.13     4112.47
CHIP ACIS-S3  370.58       383.72
TDET          4287.58     2085.72
dmcoords>: q
```

Це нам говорить про те, що потрібний нам чіп — ACIS-S, а ccd_id=7. Далі нам потрібно обрати регіон навколо центрального джерела і, використовуючи ds9, зберегти його у робочій директорії, в нашому випадку це obj.reg

```
unix% cat obj.reg:
# Region file format: CIAO version 1.0
circle(4072,4249,35)
```

Можемо використати цей файл для отримання спектра об'єкта в діапазоні енергій, а також знайти максимум випромінювання. Використовуємо dmextract для створення гістограми прийнятих подій як функція енергії:

```
unix% punlearn dmextract
```

```

unix% pset dmextract infile="acis_1838_evt2.fits \
[sky=region(obj.reg)][bin energy=500:7000:20]"
unix% pset dmextract \
outfile=obj_1838_histogram_energy.fits
unix% pset dmextract opt=generic
unix% dmextract
Input event file (acis_1838_evt2.fits \
[sky=region(obj.reg)] \
[bin energy=500:7000:20]):
Enter output file name \
(obj_1838_histogram_energy.fits):

```

Команда `dmstat` використовується для знаходження максимуму випромінювання на гістограмі. Використавши `dmlist`, можна знайти відповідну енергію для відповідних подій:

```

unix% dmstat
"obj_1838_histogram_energy.fits[cols count_rate]"
COUNT_RATE[count/s]
  min:          0          @:          3
  max:          0.015920916107 @:          62
  mean:         0.0055332286649
  sigma:        0.003904907356
  sum:          1.7982993161
  good:         325
  null:         0
unix% dmlist
"obj_1838_histogram_energy.fits[count_rate > 0.015]/
[cols energy,count_rate]" data,clean
# ENERGY          COUNT_RATE
      1730.0        0.01592091610690
      1750.0        0.01579354877805
      1770.0        0.01592091610690
      1810.0        0.01566618144919
      1830.0        0.01579354877805
      1850.0        0.01515671213377

```

Тобто, для цього набору даних пік виміряного спектру становить ~ 1.7 кеВ. Далі нам потрібно створити гістограму для чіпа, для цього використаємо команду `asphist`:

```

unix% cat pcad_asol1.lis
pcadf084244404N002_asol1.fits

```

```

unix% punlearn asphist
unix% pset asphist infile=@pcad_asol1.lis
unix% pset asphist outfile=asphist_7.fits
unix% pset asphist evtfile=\
"acis_1838_evt2.fits[ccd_id=7]"
unix% asphist
Aspect Solution List Files (@pcad_asol1.lis):
Aspect Histogram Output File \
(asphist_7.fits[ccd_id=7]):
Event List Files (acis_1838_evt2.fits):

```

Далі потрібно розрахувати інструментальну карту, оскільки ефективна площа дзеркала використовується для створення інструментальної карти, а сама площа залежить від енергії, то необхідно вирішити для якої енергії проводити розрахунки. В нашому прикладі ми будемо вважати моноенергетичний розподіл фотонів джерела як 1.7 keV:

```

unix% punlearn mkinstmap
unix% pset mkinstmap obsfile="asphist_7.fits\
[asphist]"
unix% pset mkinstmap outfile=instmap_7.fits
unix% pset mkinstmap maskfile=\
acisf01838_000N002_msk1.fits
unix% pset mkinstmap pbkfile=\
acisf084245776N002_pbk0.fits
unix% pset mkinstmap dafile=CALDB
unix% pset mkinstmap detsubsys=ACIS-7
unix% pset mkinstmap pixelgrid=\
"1:1024:#1024,1:1024:#1024"
unix% pset mkinstmap spectrumfile=NONE
unix% pset mkinstmap monoenergy=1.7
unix% mkinstmap
Output File Name (instmap_7.fits):
Energy Spectrum File (see docs) (NONE):
Energy for mono-chromatic map [keV] (0.1:10) (1.7):
Pixel grid specification x0:x1:#nx,y0:y1:#ny \
(1:1024:#1024,1:1024:#1024):
Name of fits file + extension with obs info \
(asphist_7.fits[asphist]):
Detector Name (ACIS-7):

```

```
Grating for zeroth order ARF (NONE|LETG|HETG) (NONE):  
NONE, or name of ACIS window mask file \  
(acisf01838_000N002_msk1.fits):  
NONE, or the name of the parameter block file \  
(acisf084245776N002_pbk0.fits):
```

І лише тепер використаємо `mkexpmap` для створення карти експозиції:

```
unix% get_sky_limits 1838_img_s3.fits verbose="1"  
Running: get_sky_limits  
version: 25 March 2011  
Checking binning of image: 1838_img_s3.fits  
Image has 1080 x 1080 pixels  
Pixel size is 1 by 1  
Lower left (0.5,0.5) corner is x,y= 3643.1, 3853.4  
Upper right (1080.5,1080.5) corner is \  
x,y= 4723.1, 4933.4  
DM filter is:  
x=3643.1:4723.1:#1080,y=3853.4:4933.4:#1080  
mkexpmap xygrid value is:  
3643.1:4723.1:#1080,3853.4:4933.4:#1080,
```

далі вкажемо наступне:

```
unix% punlearn mkexpmap  
unix% pset mkexpmap instmapfile=instmap_7.fits  
unix% pset mkexpmap outfile=expmap_7.fits  
unix% pset mkexpmap xygrid="3643.1:4723.1:\  
#1080,3853.4:4933.4:#1080"  
unix% pset mkexpmap asphistfile=asphist_7.fits  
unix% pset mkexpmap useavgaspect=no  
unix% pset mkexpmap normalize=no  
unix% mkexpmap  
Aspect Histogram File (asphist_7.fits):  
Output File Name (expmap_7.fits):  
Name of Instrument Map (instmap_7.fits):  
grid specification syntax x0:x1:#nx,x0:x1:ny /  
(3643.1:4723.1:#1080,3853.4:4933.4:#1080):  
Use Average Aspect Pointing (no):  
Exposure map limits: 0.000000e+00, 4.986713e+06  
Writing exposure map to expmap_7.fits$
```

Карту експозиції можна переглянути в `ds9` (Рис. 4). Колір точки відповідає потоку в фотонах/см²/с/піксель.

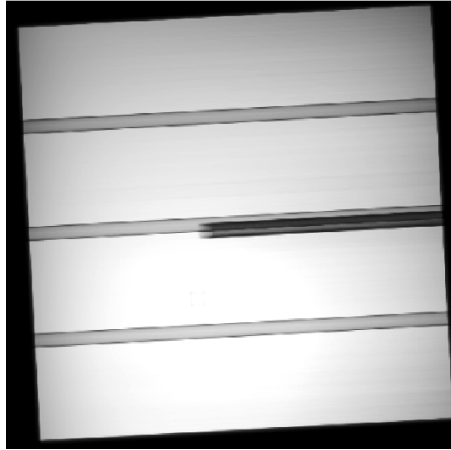


Рис. 4: Карта експозиції.

Зміна експозиції поблизу країв матриці може утворюватись за рахунок так званих “гарячих” пікселів. Тому після створення карти експозиції її потрібно очистити від таких пікселів. Для цього використовують команду `dmimgthrest`:

```
unix% punlearn dmimgthresh
unix% pset dmimgthresh infile=1838_img_s3.fits
unix% pset dmimgthresh outfile=1838_img_s3_clean.fits
unix% pset dmimgthresh expfile=expmap_7.fits
unix% pset dmimgthresh cut=1.5%
unix% pset dmimgthresh value=0.0
unix% dmimgthresh
Input dataset/block specification (1838_img_s3.fits):
Output dataset/block specification (1838_img_s3_clean.fits):
```

Тут встановлено поріг 1.5% від максимального значення на карті експозиції. Всі пікселі з меншим значенням не будуть враховуватись. Даний поріг можна змінювати в залежності від поставленої задачі. Для створення зображення в одиницях фотони/см²/с/піксель потрібно розділити зображення на карту експозиції. Це робиться за допомогою команди `dmimgcalc`:

```
unix% punlearn dmimgcalc
unix% pset dmimgcalc infile=1838_img_s3_clean.fits
```



```
unix% pset dmimgcalc infile2=expmap_7.fits
unix% pset dmimgcalc outfile=1838_img_s3_norm.fits
unix% pset dmimgcalc operation=div
unix% dmimgcalc
Input file #1 (1838_img_s3.fits):
Input file #2 (expmap_7.fits):
output file (1838_img_s3_norm.fits):
arithmetic operation (div):
warning: CONTENT has 1 different values.
warning: DETNAM has different value...Merged...
```

Тепер можна подивитись на цей самий об'єкт в одиницях потоку (Рис. 5) і порівняти з попереднім.

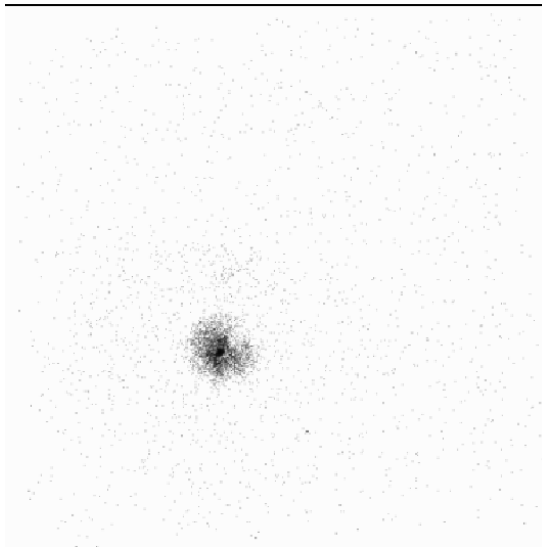


Рис. 5: Зображення скупчення галактик при врахуванні карти експозиції.

1.3.4 Отримання та моделювання профілю поверхневої яскравості

Одним із основних завдань при аналізі протяжних джерел є розрахунок радіального профілю поверхневої яскравості. Для скупчень

галактик даний профіль є основним для отримання таких характеристик, як густина газу та повна маса скупчення. У цьому розділі буде показано, як обчислити профіль поверхневої яскравості з використанням CIAO та при розбитті зображення на концентричні кільця. Отриманий профіль може бути далі промодельований за допомогою програмних пакетів *Sherpa* та *Xspec* для стандартних моделей, наприклад, β -моделлю, що і буде продемонстровано, або власною програмою.

Потік поверхневої яскравості можна визначити шляхом поділу числа відліків у відповідному регіоні на площу даного регіону. Для даного завдання нам потрібен файл із суфіксом *evt2*, використаємо вже звичне спостереження 1838 для об'єкта G21.5-0.9. Спочатку створимо список подій для діапазону енергій 0.5–8.0 кеВ:

```
unix% dmscopy \
"acisf01838N002_evt2.fits[energy=300:8000]" \
acis_1838_evt2.fits
```

Для побудови радіального профілю використовують *dmextract*, але перед цим потрібно розбити зображення об'єкта на концентричні кільця. Для цього використовуємо програму *ds9*:

```
unix% ds9 acis_1838_evt2.fits &
```

Далі в середовищі *ds9* потрібно перейти до **Select region**, потім до **Shape** і обрати **Annulus**. Після натиснення лівою кнопкою миші на самому рисунку з'явиться концентричне кільце (**annulus**), натиснувши по якому двічі лівою кнопкою миші, бачимо вікно даного регіону з координатами центра та радіусами внутрішнього та зовнішнього кілець. В даному вікні маємо змогу обирати власні координати. В нашому випадку це координати максимуму випромінювання. У правому вікні можна створити будь-яку кількість кілець, з якими завгодно радіусами. Саме вікно має вигляд, як показано на Рис. 6.

Для нашого випадку область зображення було розбито на 38 кілець з фізичними координатами $x = 4072$, $y = 4246$. Після того, як ви згенеруєте свої кільця, ви побачите щось на зразок нашого прикладу, зображеного на Рис. 7

Після цього потрібно зберегти ці кільця в робочу папку. Для цього потрібно натиснути в *ds9* по вкладці **Region** і перейти до **Save regions**, назва збереженого регіону **annuli.reg**. Важливо те, що після цього потрібно перезберегти даний файл у форматі CIAO, це виконується в тому ж *ds9*. Також нам потрібно створити регіон для врахування фону. Його було вибрано так, як показано на Рис. 7. Для

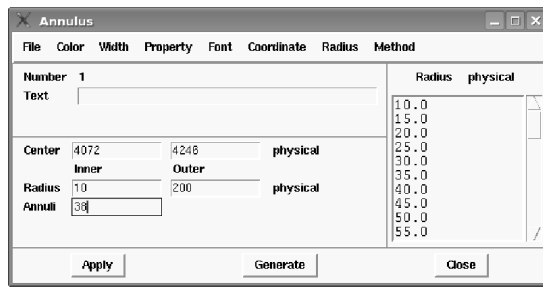


Рис. 6: Зображення вікна з характеристиками створених концентричних кілець.

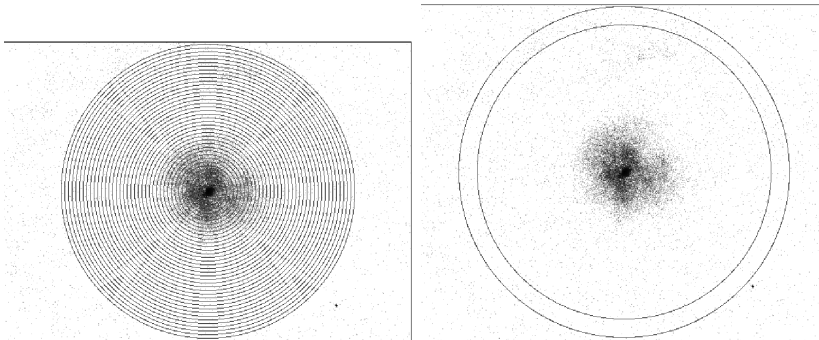


Рис. 7: Зображення джерела з кільцями (ліворуч) та регіон зображення для врахування фону (праворуч).

перегляду даних `.reg` файлів використовують команду `more`:

```
unix% more annuli.reg
# Region file format: CIAO version 1.0
annulus(4072,4246,10,15)
annulus(4072,4246,15,20)
annulus(4072,4246,20,25)
.
. (etc.)
.
annulus(4072,4246,190,195)
annulus(4072,4246,195,200)
```

так само і для фонового:

```
unix% more annuli\_bgd.reg
# Region file format: CIAO version 1.0
annulus(4070,4250,200,225)
```

Корисним є видалення точкових джерел, що потрапили в область кілець. Для нашого випадку це зображено на Рис.8.

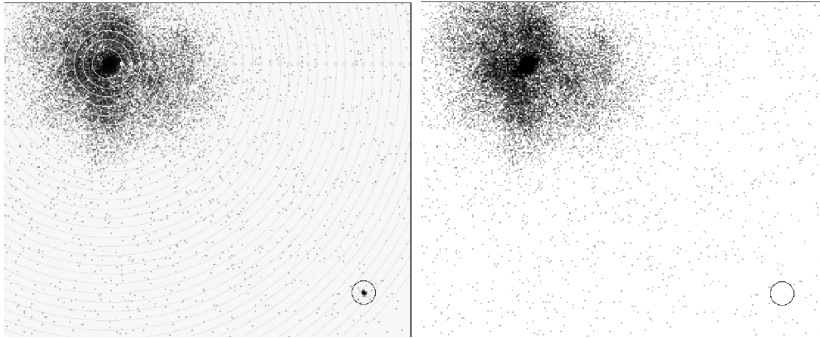


Рис. 8: Ліворуч: Виділення точкового джерела. Праворуч: Зображення без нього.

Даний регіон на правій панелі Рис. 8 був збережений як `cir.reg`, для його виділення використовуємо команди `dmcopy` та `exclude`:

```
unix% dmcopy "acis\_1838\_evt2.fits
[exclude sky=region(cir.reg)]"
acis_1838_excl_evt2.fits
```

Результат можна спостерігати на лівому зображенні Рис.8. А далі запускаємо команду `dmextract` для створення радіального профілю:

```
unix% punlearn dmextract
unix% pset dmextract infile="acis_1838_evt2.fits \
[bin sky=@annuli.reg]"
unix% pset dmextract outfile=1838_rprofile.fits
unix% pset dmextract bkg="acis_1838_evt2.fits \
[bin sky=@annuli_bgd.reg]"
unix% pset dmextract opt=generic
unix% dmextract
Input event file \
```

```
(acis_1838_evt2.fits[bin sky=@annuli.reg]):  
Enter output file name (1838_rprofile.fits):
```

Вміст файлів можна переглянути за допомогою:

```
plist dmextract: unix% dmlist 1838_rprofile.fits cols  
Columns for Table Block HISTOGRAM  
Col Name Unit Type  
20 NET_COUNTS count Real8  
21 NET_ERR count Real8  
22 NET_RATE count/s Real8  
23 ERR_RATE count/s Real8  
24 SUR_BRI count/pixel**2 Real8  
25 SUR_BRI_ERR count/pixel**2 Real8
```

SUR_BRI визначається як NET_COUNTS/AREA (колонки 20 і 8, відповідно); SUR_BRI_ERR — це NET_ERR/AREA (колонки 21 і 8). Відмітимо, що оскільки поверхнева яскравість визначається з колонки NET_COUNTS, то каунти фону були видалені наступним чином:

$$\text{NET_COUNTS} = \text{COUNTS} - [(\text{BG_COUNTS}/\text{BG_AREA}) * \text{AREA}].$$

Тому немає необхідності враховувати фон окремо при моделюванні даних. Нарешті нам потрібно додати ще одну колонку, що визначає середину кільця (rmid):

```
unix% punlearn dmtcalc  
unix% pset dmtcalc infile=1838_rprofile.fits  
unix% pset dmtcalc outfile=1838_rprofile_rmid.fits  
unix% pset dmtcalc expression="rmid=0.5*(R[0]+R[1])"  
unix% dmtcalc  
Input file (1838_rprofile.fits):  
Output file (1838_rprofile_rmid.fits):  
expression(s) to evaluate (rmid=0.5*(R[0]+R[1])):}
```

Далі переглянемо вміст даного файлу

```
dmlist 1838_rprofile_rmid.fits'[cols R,RMID]' data  
Data for Table Block HISTOGRAM}  
ROW R[2] RMID  
1 [10.0 15.0] 12.50  
2 [15.0 20.0] 17.50  
3 [20.0 25.0] 22.50  
4 [25.0 30.0] 27.50  
5 [30.0 35.0] 32.50
```

Далі використаємо пакети ChIPS для відтворення нашого профілю та Sherpa для фітування:

```
unix% chips
```

```
-----  
Welcome to ChIPS: CXC's Plotting Package  
-----
```

```
CIAO 4.3 ChIPS version 1 Thursday, December 2, 2010  
chips> make_figure("1838_rprofile_rmid.fits  
[cols rmid,sur_bri,sur_bri_err]")  
chips> log_scale(XY_AXIS)  
chips> quit
```

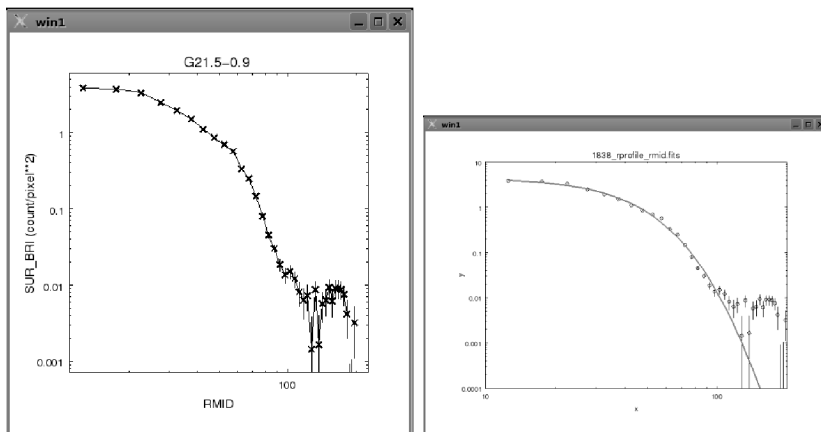


Рис. 9: Ліворуч: Профіль поверхневої яскравості. Праворуч: Моделювання β -моделю даного профіля.

Далі запусимо Sherpa для фітування профілю:

```
unix% sherpa
```

```
-----  
Welcome to Sherpa: C  
XC's Modeling and Fitting Package  
-----
```

```
CIAO 4.3 Sherpa version 1 Thursday, December 2, 2010  
sherpa> load_data(1, "1838_rprofile_rmid.fits  
[cols rmid,sur_bri,sur_bri_err]")
```

```

sherpa> set_source("beta1d.src")
sherpa> src.r0 = 105
sherpa> src.beta = 4
sherpa> src.ampl = 0.00993448
sherpa> freeze(src.xpos)
sherpa> fit()
Statistic value = 233.766 at function evaluation 88
Data points = 38
Degrees of freedom = 35
Probability [Q-value] = 3.08377e-31
Reduced statistic = 6.67902
    src.r0      121.817
    src.beta    3.94547
    src.ampl    4.47076
sherpa> plot_fit()
sherpa> log_scale(XY_AXIS)
sherpa> limits(X_AXIS, 10, 200)
sherpa> limits(Y_AXIS, 0.0001, 10)
sherpa> quit

```

Для фітування ми використали β -модель, як одну з найкращих для опису профілей поверхневої яскравості [3]. Результат моделювання профілю можна бачити на Рис. 9.

1.3.5 Створення зображень

Створення зображень є важливим при представленні своєї роботи чи на конференціях, та при написанні статей. Створення яскравих рисунків є останнім кроком при роботі з протяжними джерелами. Такі зображення ніякої наукової інформації не несуть, тому їх не варто інтерпретувати у фізичному ракурсі.

Отож, на початковому етапі при роботі, наприклад, з скупченнями галактик, потрібно позбутися точкових джерел. Проробимо це за допомогою `wavdetect` та `exclude`, як було описано вище. Для обробки потрібен лише головний файл з суфіксом `evt2`, для прикладу було взято групу галактик NGC 4038/39. Створимо зображення з потрібним нам розмірами, на якому зображена наша група галактик та її дифузний газ:

```

unix% punlearn dmcopу
unix% dmcopу "acisf00315N004_evt2.fits/

```

```
[energy=300:7000]/
[bin x=4004.5:4404.5:1,y=3625.5:4025.5:1]"
diff_image.fits
```

Даний набір команд створює зображення розміром 400 на 400 пікселів. Для швидкої обробки даних, при створенні дифузного зображення, рекомендується використовувати розмір зображення не більше, ніж 500 на 500 пікселів. Саме створене зображення можна побачити на Рис. 10.

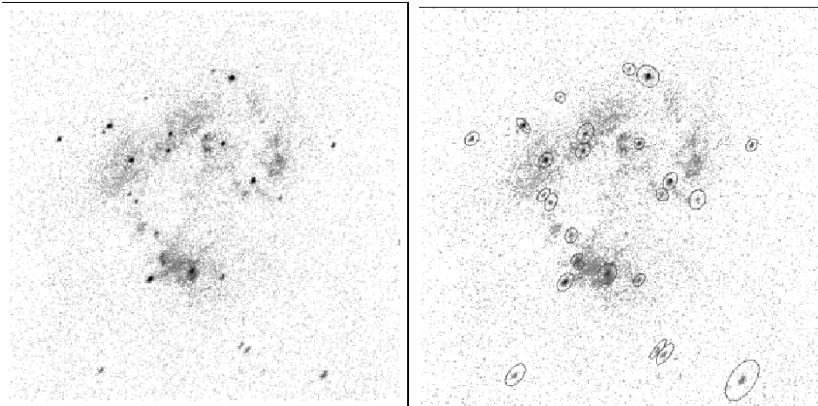


Рис. 10: Ліворуч: Зображення NGC 4038/39 розміром 400 на 400 пікселів. Праворуч: Виділення точкових об'єктів на рисунку.

Для пошуку та видалення точкових джерел запускаємо наступний блок команд:

```
unix% punlearn wavdetect
unix% pset wavdetect infile=diff_image.fits
unix% pset wavdetect outfile=sources.fits
unix% pset wavdetect scellfile=sources_scell.fits
unix% pset wavdetect imagefile=sources_image.fits
unix% pset wavdetect defnbkgfile=sources_bkg.fits
unix% pset wavdetect regfile=sources.reg
unix% pset wavdetect scales="1 2 4"
uni\% pset wavdetect ellsigma=4
unix% wavdetect
Input file name (diff_image.fits):
```



```

Output source list file name \
(sources.fits):
Output source cell image file name \
(sources_scell.fits):
Output reconstructed image file name \
(sources_image.fits):
Output normalized background file name \
(sources_bkg.fits):

```

Даний набір команд створить файл `sources.fits`, в який згенерує, тобто виділить, усі точкові джерела. Далі потрібно просто відкрити дане зображення у `ds9` і зберегти виділені області у файл з розширенням `.reg`. Для видалення їх потрібно використати команду `exclude`, так, як було показано у попередньому розділі. В результаті отримаємо зображення, яке можна бачити на лівій панелі Рис. 11. Для створення “красивого” зображення даний рисунок потрібно згладити. Роблять це за допомогою команди `aconvolve`:

```

unix% punlearn aconvolve
unix% pset aconvolve infile=diff_image.fits
unix% pset aconvolve outfile=smoothed_fill.fits
unix% pset aconvolve
kernelpec="lib:gaus(2,5,1,10,10)"
unix% pset aconvolve method=fft
unix% aconvolve
Input file name (diff_image.fits):
Kernel specification (lib:gaus(2,5,1,10,10)):
Output file name (smoothed_fill.fits):

```

1.4 Робота зі спектрами

Використовуючи різні комбінації команд у програмному забезпеченні CIAO, ми маємо змогу створити спектри для фону та самого джерела. Команда `specextract` дозволяє автоматизувати усі кроки разом для швидкого створення спектрів як для протяжних, так і для точкових об’єктів. Для цього нам потрібно мати файли з суфіксами `evt2`, `asol1`, `bpix1`, `pbk0` та `msk1` в робочій директорії. Приклад представлений для ARP 220 (ACIS-S, ObsID 869). Команда `specextract` має дуже простий синтаксис:
Для CIAO 4.2

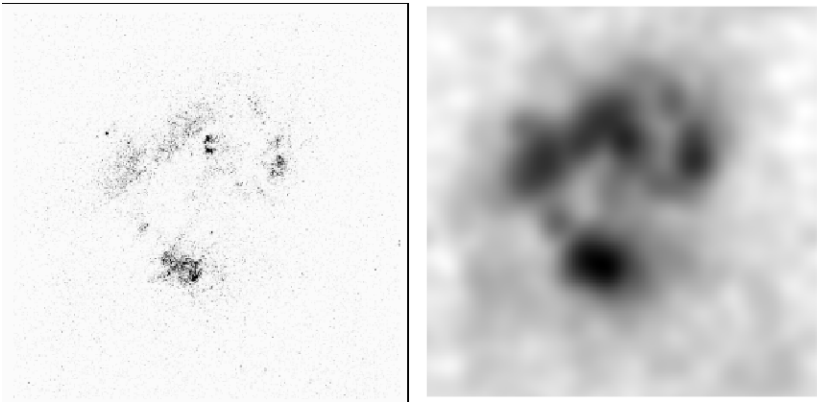


Рис. 11: Зображення групи галактик NGC 4038/39 без точкових джерел (ліворуч) та з визначеними точковими об'єктами (праворуч).

```
specextract infile="evt2.fits[sky=region(src.reg)]" /
bkgfile="evt2.fits[sky=region(bkg.reg)]"
outroot=extended pbkfile=pbk0.fits grouptype=NUM_CTS
binspec=15
```

або для CIAO 4.3:

```
specextract infile="evt2.fits[sky=region(src.reg)]" /
bkgfile="evt2.fits[sky=region(bkg.reg)]" bkgresp=yes
outroot=extended pbkfile=pbk0.fits
asp=pcad_asol1.fits
mskfile=msk1.fits
grouptype=NUM_CTS binspec=15}
```

Спочатку перевіримо свій aspect solution файл:

```
unix% cat pcad_asol1.lis
pcadf078246822N003_asol1.fits
unix% punlearn specextrac
unix% pset specextract infile=\
"acisf00869N003_evt2.fits/
[sky=region(simple.reg)]"
unix% pset specextract outroot=simple
unix% pset specextract bkgfile=\
"acisf00869N003_evt2.fits/
```

```

[sky=region(simple_bkg.reg)]"
unix% pset specextract asp=@pcad_asol1.lis$
unix% pset specextract pbkfile=\
acisf078247287N003_pbk0.fits
unix% pset specextract mskfile=\
acisf00869_000N003_msk1.fits
unix% pset specextract badpixfile=\
acisf00869_000N003_bpix1.fits
unix% pset specextract weight=yes correct=no
unix% pset specextract grouptype=NUM_CTS binspec=15

```

Після цього потрібно лише запустити наступний скрипт з параметром `verbose=2` для явного перегляду, в результаті отримаємо:

```

unix% specextract verbose=2
Source event file(s) \
(acisf00869N003_evt2.fits\
[sky=region(simple.reg)]):
Output directory path + root name for \
output files (simple):
Should response files be weighted? (yes):
Apply point source aperture \
correction to ARF? (no):
Combine ungrouped output spectra \
and responses? (no):
Background event file(s) \
(acisf00869N003_evt2.fits\
[sky=region(bg_simple.reg)]):
Create background ARF and RMF? (yes):
Source aspect solution or histogram file(s) \
(@pcad_asol1.lis):
pbkfile input to mkwarf \
(acisf078247287N003_pbk0.fits):
mskfile input to mkwarf \
(acisf00869_000N003_msk1.fits):
Running: specextract
Version: 6 October 2011
Checking source input files for readability...
Checking background input files for readability...
Extracting src spectra for item \
1 of 1 in input list

```

```

Running tool dmextract using:
>>> dmextract "acisf00869N003_evt2.fits\
[sky=region(simple.reg)]
[bin PI]" simple.pi wmap=\
"[energy=300:2000][bin det=8]"
verbose=2
Creating src ARF for item 1 of 1 in input list
Running tool sky2tdet using:
>>> sky2tdet "acisf00869N003_evt2.fits[sky=\
region(simple.reg)]
[energy+ 300:2000][bin sky=1]" simple.asphist
"simple_tdet.fits[wmap]" clobber=yes
Running: remove_extra_time_keywords
version: 21 Apr 2011
infile=simple_21Apr_tdet.fits[wmap]
no keywords were deleted from \
simple_21Apr_tdet.fits[wmap]
Running tool mkwarf using:
>>> mkwarf "simple_tdet.fits[wmap]" \
simple.warf simple.wfef
spectrumfile= egridspec=0.3:11.0:0.01
pbkfile=acisf078247287N003_pbk0.fits
mskfile=acisf00869_000N003_msk1.fits verbose=2
Creating src RMF for item 1 of 1 in input list
Searching for P2_RESP calibration file...
Running tool calquiz using:
>>> calquiz "simple.pi[WMAP]" telescope= instrument= \
product=SC_MATRIX
calfile=CALDB outfile=y echo=yes
Using mkacisrmf...
Running tool mkacisrmf using:
>>> mkacisrmf CALDB simple.wrmf "simple.pi[WMAP]" \
0.3:11.0:0.01 1:1024:1 PI 0 INDEF INDEF \
CALDB verbose=2
Grouping src spectrum for item 1 of 1 in input list
Running tool dmgroup using:
>>> dmgroup "simple.pi[SPECTRUM]" \
simple_grp.pi NUM_CTS 15.0 \
binspec= xcolumn=channel ycolumn=counts verbose=2
Updating header of simple.pi with RESPFILE and \

```

```

ANCRFILE keywords.
Running tool dmhedit using:
>>> dmhedit simple.pi none add RESPFILE \
simple.wrmf verbose=2
Running tool dmhedit using:
>>> dmhedit simple.pi none add ANCRFILE \
simple.warf verbose=2
Updating header of simple_grp.pi with RESPFILE and
ANCRFILE keywords.
Running tool dmhedit using:
>>> dmhedit simple_grp.pi none add RESPFILE \
simple.wrmf verbose=2
Running tool dmhedit using:
>>> dmhedit simple_grp.pi none add ANCRFILE \
simple.warf verbose=2
Extracting bkg spectra for item 1 of 1 in input list
Running tool dmextract using:
>>> dmextract "acisf00869N003_evt2.fits\
[sky=region(bg_simple.reg)]
[bin PI]" simple_bkg.pi wmap="[energy=300:2000]\
[bin det=8]"
verbose=2
Creating bkg ARF for item 1 of 1 in input list
Running tool sky2tdet using:
>>> sky2tdet
"acisf00869N003_evt2.fits[sky=region\
(bg_simple.reg)][energy+
300:2000][bin sky=1]" simple.asphist \
"simple_bkg_tdet.fits[wmap]"
clobber=yes
Running: remove_extra_time_keywords
version: 21 Apr 2011
infile=simple_21Apr_bkg_tdet.fits[wmap]
no keywords were deleted from \
simple_21Apr_bkg_tdet.fits[wmap]
Running tool mkwarf using:
>>> mkwarf "simple_bkg_tdet.fits[wmap]" \
simple_bkg.warf
simple_bkg.wfef spectrumfile= egridspec=\
0.3:11.0:0.01 \

```

```

pbkfile=acisf078247287N003_pbk0.fits
mskfile=acisf00869_000N003_msk1.fits verbose=2
Creating bkg RMF for item 1 of 1 in input list
Searching for P2_RESP calibration file...
Running tool calquiz using:
>>> calquiz "simple_bkg.pi[WMAP]" telescope= \
instrument= product=SC_MATRIX \
calfile=CALDB outfile=y echo=yes
Using mkacisrmf...
Running tool mkacisrmf using:
>>> mkacisrmf CALDB simple_bkg.wrmf \
"simple_bkg.pi[WMAP]" 0.3:11.0:0.01 1:1024:1 \
PI 0 INDEF INDEF CALDB verbose=2
Updating header of simple_bkg.pi with RESPFILE and \
ANCRFILE keywords.
Running tool dmhedit using:
>>> dmhedit simple_bkg.pi none add RESPFILE \
simple_bkg.wrmf verbose=2
Running tool dmhedit using:
>>> dmhedit simple_bkg.pi none add ANCRFILE \
simple_bkg.warf verbose=2
Updating header of simple.pi with BACKFILE keyword.
Running tool dmhedit using:
>>> dmhedit simple.pi none add BACKFILE \
simple_bkg.pi verbose=2
Updating header of simple_grp.pi with \
BACKFILE keyword.
Running tool dmhedit using:
>>> dmhedit simple_grp.pi none add \
BACKFILE simple_bkg.pi verbose=2

```

В результаті ми отримуємо набір файлів, використовуючи які, ми можемо створити спектр нашого джерела. Візуалізація та подальша обробка проводяться за допомогою пакетів Sherpa або Xspec [2].

```

Source:
simple.pi      ungrouped spectrum
simple_tdet.fits  WMAP from sky2tdet
simple.warf     weighted ARF
simple.wfef     FEF weight file
simple.wrmf     weighted RMF

```

```

simple_grp.pi grouped spectrum
Background:
simple_bkg.pi      ungrouped spectrum
simple_bkg_tdet.fits WMAP from sky2tdet
simple_bkg.warf    weighted ARF
simple_bkg.wfef    FEF weight file
simple_bkg.wrmf    weighted RMF

```

1.5 Команди і додаткові посилання

Нижче наведено список команд, що найчастіше використовуються при обробці даних.

Назва	Пояснення
aconvolve	Використовується для створення згладжених зображень.
csmooth	Використовується для створення зображень та згладження.
dmcopy	Використовується для фільтрування, перейменування та редагування списку подій та самих зображень.
dmextract	Використовується для створення спектрів та кривих блиску.
mkexrmap	Використовується для створення карти експозиції.
specextract	Використовується для протяжних об'єктів при генерації arf та gmf файлів.
wavdetect	Використовується для виявлення точкових джерел на матриці для подальшого їх усунення.

Додаткову інформацію з установки програмного забезпечення, обробки даних та аналізу отриманих результатів можна знайти веб-сторінках телескопу “Чандра” [9, 10, 11].

1.6 Командні файли

Перший командний файл (скрипт) використовується для отримання дифузного зображення.

```
#!/bin/csh
csh
source /virgo/scripts/login_ciao.csh
echo "Create of DIFFUSE EMISSION IMAGE"
echo "Input files:evt2.fits, coordinates
[bin x=.....:1,y=.....:1] "
echo "Create an image of the region (dmcopу)"
punlearn dmcopу
dmcopу "acis_evt2.fits[energy=500:7000]
[bin x=.....:1,y=.....:1]"
diff_image.fits
echo "Detect the point sources (wavdetect)"
punlearn wavdetect
pset wavdetect infile=diff_image.fits
pset wavdetect outfile=point_sources.fits
pset wavdetect scellfile=point_sources_scell.fits
pset wavdetect imagefile=point_sources_image.fits
pset wavdetect defnbkgfile=point_sources_bkg.fits
pset wavdetect regfile=point_sources.reg
pset wavdetect scales="1 2 4"
pset wavdetect ellsigma=4
wavdetect
echo "Remove the point sources from matrix"
punlearn dmcopу
dmcopу "acis_evt2.fits \
[exclude sky=@point_sources.reg]"
acis_excl_evt2.fits
echo "Remove the point sources diffuse image"
punlearn dmcopу
dmcopу "diff_image.fits\
[exclude sky=@point_sources.reg]"
diff_image_excl.fits
echo "Smooth the diffuse image.."
punlearn aconvolve
pset aconvolve infile=diff_image_excl.fits
pset aconvolve outfile=smoothed_diff_excl_image.fits
```



```

pset aconvolve kernelspec="lib:gaus(2,5,1,10,10)"
pset aconvolve method=fft
aconvolve
echo "DONE."

```

При роботі із зображеннями часто є корисним накладання того ж зображення одне на одне при різних енергіях фотонів. Наступний командний файл розбиває зображення на три з різними енергіями фотонів і накладає один на другий, що допомагає покращити виявлення певних структур в складі об'єкта.

```

#!/bin/csh
csh
source /virgo/scripts/login_ciao.csh
echo "This script build true_colour \
image of cluster
in three range of energy"
echo "input acisf_evt2.fits, and \
coordinates of image
[bin x=...,y=...]"
echo "START BUILD THE IMAGE (RESIDUALS IMAGE) \
OF CLUSTER"
echo "Create the soft image"
punlearn dmcopу
pset dmcopу infile=\
"acisf_evt2.fits[energy=500:1500]
[bin x=...,y=...]"
pset dmcopу outfile=soft_image.fits
dmcopу
echo "Create the medium image"
punlearn dmcopу
pset dmcopу infile=\
"acisf_evt2.fits[energy=1500:2500]
[bin x=...,y=...]"
pset dmcopу outfile=medium_image.fits
dmcopу
echo "Create the hard image"
punlearn dmcopу
pset dmcopу infile=\
"acisf_evt2.fits[energy=2500:7000]
[bin x=...,y=...]"

```

```

pset dmcopу outfile=hard_image.fits
dmcopу
echo "Build true colour image"
punlearn dmimg2jpg
pset dmimg2jpg infile=soft_image.fits
pset dmimg2jpg greenfile=medium_image.fits
pset dmimg2jpg bluefile=hard_image.fits
pset dmimg2jpg outfile=truecolour.jpg
pset dmimg2jpg maxred=1 maxblue=1 maxgreen=1
pset dmimg2jpg showgrid=yes
pset dmimg2jpg gridsize=60
pset dmimg2jpg fontsize=1
dmimg2jpg
echo "_-----DONE-----_"

```

Наступний командний файл буде спектр джерела.

```

#!/bin/csh
csh
source /virgo/scripts/login_ciao.csh
echo "Building the source and background
spectrums for the cluster"
echo "infile acis_evt2.fits, \
sources.reg, sources_bkg.reg, "
echo "'_asol1.fits, _msk1.fits, _bpix1.fits'"
punlearn specextract
pset specextract infile=
"acis_evt2.fits[sky=region(sources.reg)]"
pset specextract outroot=source
pset specextract bkgfile=
"acis_evt2.fits[sky=region(source_bkg.reg)]"
pset specextract asp=@pcad_asol1.fits
pset specextract pbkfile=acis_pbk.fits
pset specextract mskfile=acis_msk1.fits
pset specextract badpixfile=acis_bpix1.fits
pset specextract weight=yes correct=no
pset specextract grouptype=NUM_CTS binspec=50
specextract verbose=2
### The output files are the next:
### Source:
### source.pi -> ungrouped spectrum

```

```

### source_tdet.fits -> WMAP from sky2tdet
### source.warf -> weighted ARF
### source.wrnf -> weighted RMF
### source.wfef -> FEF weight file
### source_grp.pi -> grouped spectrum
### Background:
### source_bkg.pi -> ungrouped spectrum
### source_bkg_tdet.fits -> WMAP from sky2tdet
### source_bkg.warf -> weighted ARF
### source_bkg.wrnf -> weighted RMF
### source_bkg.wfef -> FEF weight file
### source_bkg_grp.pi -> grouped spectrum
echo "Group all files...."
grppha clobber=yes source.pi source.grp comm='chkey
RESPFILE source.wrnf & chkey ANCRFILE source.warf &
chkey BACKFILE source_bkg.pi & group min 50 & exit'
xspec
echo "DONE."

```

Далі в середовищі Xspec можна проаналізувати даний спектр.

2 Обробка спостережень супутника Fermi/LAT

2.1 Загальні відомості про телескоп Fermi/LAT

Космічний телескоп Фермі було запущено 11-го червня 2008-го року. Спочатку він називався GLAST (The Gamma ray Large Area Space Telescope), але пізніше був перейменований у Фермі. Два основних інструменти на його борту – LAT (Large Area Telescope) та GBM (GLAST Burst Monitor). LAT використовується для звичайних спостережень джерел, в той час як GBM – в основному для спостережень гамма-спалахів.

Отже, LAT спостерігає небо в діапазоні енергій від 30 MeV до ~ 1 TeV, хоча “нормальним” для нього вважається діапазон 100 MeV – 300 GeV. Нижче 100 MeV зростаючий фон сильно ускладнює обробку даних, а вище 300 GeV можливі проблеми із калібруванням інструмента. Фермі був спеціально запущений таким чином, щоб постійно сканувати все небо: повний оберт навколо своєї осі він робить за ~ 1.6 години, а огляд всього неба – за два оберти, тобто 3.2 години. Телескоп інколи перестає обертатися і наводиться тільки на один об’єкт, але таке трапляється лише в окремих випадках. Таким випадком може бути раптовий спалах або різка зміна поведінки якогось із об’єктів, що являє особливий інтерес для науковців (так звані Targets of Opportunity, або ToO). Прикладом такого явища є спалах “стандартної свічки” – пульсару у Крабовидній туманності у вересні 2010-го року. Також можливим є автоматичне наведення телескопу на гамма-спалахи, зареєстровані інструментами LAT та GBM..

Як відомо, будь-який телескоп будує зображення точкового джерела не у вигляді точки, а трохи його розмазує. Профіль цього “розмиття” носить назву PSF (від англійського Point-Spread Function), і його ширина визначає роздільну здатність телескопу. У випадку Fermi/LAT PSF дуже сильно залежить від енергії:

$$\theta_{PSF} = 2(E/1\text{GeV})^{-0.8} \text{ градусів (95\% довірчий інтервал)},$$

таким чином, якщо на 100 MeV-ах він складає 10 градусів, то на 100 GeV-ах – всього лиш 0.2 градуси.

Ці особливості супутника слід враховувати при аналізі отриманих з нього даних.

2.2 Завантаження даних

Дані Fermi/LAT можна завантажити з офіційного веб-сайту телескопу [12]. Дана сторінка для завантаження достатньо проста у користуванні.

Для аналізу необхідно завантажити файл із зареєстрованими фотонами (**photon file**), та файл, що включає історію положення та орієнтації супутника в просторі (**spacecraft file**). Слід мати на увазі, що якщо вміст фотонного файлу залежить від ділянки неба, яку ви обрали, то файл супутника залежить лише від часу, тобто якщо потрібно проаналізувати спостереження двох об'єктів за один і той же проміжок часу, то файл історії супутника достатньо завантажити лише один раз.

Для прикладу, можна завантажити дані для відомого пульсару у Крабовидній туманності, наприклад, з 1-го серпня по 24-те вересня 2010-го року. Для цього в полі **Object name or coordinates** пишемо **Crab**, а в **Observation dates** ставимо 2010-08-01 00:00:00, 2010-09-24 00:00:00. Завантажимо дані для інтервалу енергій 1-300 GeV, для чого в полі **Energy range (MeV)** пишемо 1000,300000. Також обираємо звичайний тип даних (**LAT data type: photon**) та ставимо галочку навпроти **Spacecraft data**, щоб завантажити історію позиціонування супутника.

Далі тиснемо **Start search**. На наступній сторінці сервер повідомить нам, що запит успішно прийнятий, додасть коротку інформацію щодо нього та запропонує посилання, за яким можна буде завантажити дані. Цю коротку інформацію варто зберегти кудись, так як вона може пізніше знадобитися – як мінімум, щоб нагадати, що саме ми завантажили. Наприклад, можна зберегти її у файл **Query.txt**, який матиме наступний вигляд:

```
Equatorial coordinates (degrees) (83.6331,22.0145)
Time range (MET) (302313600,306979200)
Time range (Gregorian) (2010-08-01 00:00:00,2010-09-24
00:00:00)
Energy range (MeV) (1000,300000)
Search radius (degrees) 15
```

Тепер можна перейти по запропонованому посиланню, зачекати, поки сервер закінчить формування замовлених нами файлів та завантажити останні.

2.3 Застосування фільтрів

Сервер Fermi називає файли зручними для комп'ютера, але не для користувача, іменами на кшталт

```
L100924055816E0D2F37E91_PH00.fits.
```

Краще змінити їх на щось простіше, наприклад, `ph.fits` та `sc.fits` для фотонних файлів та файлів історії позиціювання, відповідно. Варто зауважити, що якщо даних багато, сервер Fermi любить розбивати їх на окремі невеликі (після перейменування нехай вони називаються `ph0.fits`, `ph1.fits`, `ph2.fits` і т.д.). Це може бути зручно для завантаження даних, але додає багато дискомфорту при обробці, тому радимо склеювати їх в один файл `ph.fits`, наприклад, наступним чином:

```
mv *PH00* ph0.fits
mv *PH01* ph1.fits
mv *PH02* ph2.fits
mv *SC* sc.fits

echo -n "Processing EVENTS extension... "
ftmerge 'ph0.fits[events],\
ph1.fits[events],ph2.fits[events]' \
events.fits clobber=yes lastkey="TStop"
fdelhdu events.fits[2] confirm=no proceed=yes
echo "Done!"
echo -n "Processing GTI extension... "
ftmerge 'ph0.fits[gti],ph1.fits[gti],\
ph2.fits[gti]' gti.fits \
clobber=yes
fdelhdu gti.fits[1] confirm=no proceed=yes
echo "Done!"
echo "Combining files..."
fappend gti.fits[gti] events.fits
echo "Removing temporary files..."
mv -f events.fits ph.fits
rm -f gti.fits
echo "Done!"
```

Ви можете записати ці команди у деякий скрипт (командний файл) та запускати його при потребі.

Тепер можна, власне, і відфільтрувати отримані дані. Навіщо це необхідно? В перше чергу, це потрібно зробити тому, що не всі по-

дії, що їх реєструє супутник, насправді є фотонами – є ще космічні промені та інше (можливе) “сміття”, яке також потрапляє на детектор. Fermi сам визначає, які події чому відповідають, але оскільки він не знає, що саме нам знадобиться, то у фотонний файл потрапляє все. Взагалі, рекомендації щодо фільтрів можна знайти на сайті Fermi/LAT [13]. Звичайним фотонам відповідає клас 2, саме його ми і відфільтруємо.

```
gtselect evclass=2
Input FT1 file[] ph.fits
Output FT1 file[] ph_f.fits
RA for new search center (degrees) (0:360) \
[] 83.6331
Dec for new search center (degrees) (-90:90) \
[] 22.0145
radius of new search region (degrees) (0:180) \
[] 10
start time (MET in s) (0:) [] 302313600
end time (MET in s) (0:) [] 306979200
lower energy limit (MeV) (0:) [] 1000
upper energy limit (MeV) (0:) [] 300000
maximum zenith angle value (degrees) (0:180) \
[] 105
Done.
```

Значення координат, початкового та кінцевого моментів часу, відповідають нашому запиту (от навіщо був потрібен `Query.txt`).

Тепер потрібно врахувати, що супутник міг працювати не весь час. Можливі перебої в штатній роботі інструмента враховуються шляхом використання `GTI (Good Time Interval)`. Щоб дописати `GTI` до наших даних, потрібно використати команду `gtmktime:[-0.5ex]`

```
gtmktime
Spacecraft data file[] sc.fits
Filter expression[] DATA_QUAL==1 && LAT_CONFIG==1 \
&& ABS(ROCK_ANGLE)<52
Apply ROI-based zenith angle cut[] yes
Event data file[] ph_f.fits
Output event file name[] ph_f_g.fits
```

Тепер можна приступати власне до аналізу даних.

2.4 Створення зображень

Щоб подивитись, як виглядає пульсар в гамма-діапазоні – використаємо команду `gtbin`, використавши відфільтровані дані.

```
gtbin
This is gtbin version \
ScienceTools-v9r17p0-fssc-20100906
Type of output file (CCUBE|CMAP|LC|PHA1|PHA2) \
[] CMAP
Event data file name[] ph_f_g.fits
Output file name[] ph_cMap.fits
Spacecraft data file name[] sc.fits
Size of the X axis in pixels[] 220
Size of the Y axis in pixels[] 220
Image scale (in degrees/pixel)[] 0.1
Coordinate system (CEL - celestial, GAL -galactic) \
(CEL|GAL) [] CEL
First coordinate of image center in degrees \
(RA or galactic l)[] 83.6331
Second coordinate of image center in degrees \
(DEC or galactic b)[] 22.0145
Rotation angle of image axis, in degrees[] 0
Projection method e.g. \
AIT|ARC|CAR|GLS|MER|NCP|SIN|STG|TAN:[] CAR
```

Отримане зображення (Рис. 12) можна подивитися за допомогою програмного пакету `DS9`.

Звичайно, її можна ще трохи покращити. На енергії в 1 GeV 68% фотонів, завдяки PSF, розмиваються у кружечок розміром 0.8 градуса, так що отримане зображення варто згладити гаусіаною (із доступного у `DS9` вона найближче до істинної форми PSF) з ядром у 8 пікселів – адже ми будували картинку в масштабі 0.1 градуса/піксель. Також варто додати координатну сітку, скажімо, у галактичних координатах. Результат представлено на Рис. 13

Отримане зображення набагато приємніше оку, на ньому навіть видно площину Галактики.

2.5 Побудова кривої блиску

Для того, щоб побудувати криву блиску, слід вибрати тільки ті фотони, які прийшли від нашого джерела. Будувати її ми будемо в

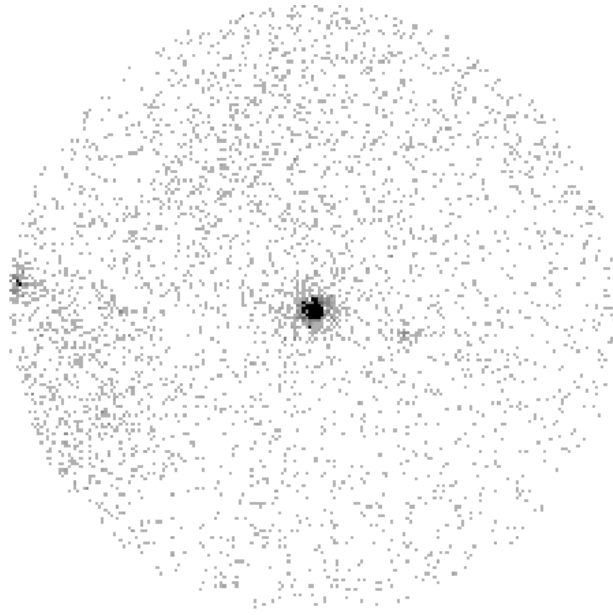


Рис. 12: Зображення ділянки неба навколо пульсару у Крабовидній туманності.

тому ж інтервалі енергій, що й знімки 12 та 13, тобто 1–300 GeV. На енергії в 1 GeV PSF телескопу Fermi/LAT складає 2 градуси (по 95-ти процентах фотонів) і лише спадає із збільшеннями енергії, таким чином буде достатньо зібрати фотони з кружечка радіусом у 2 градуси навколо нашого пульсару.

```
gtselect evclass=2
Input FT1 file[ph.fits]
Output FT1 file[ph_f.fits]
RA for new search center (degrees) (0:360) \
[83.6331]
Dec for new search center (degrees) (-90:90) \
[22.0145]
radius of new search region (degrees) (0:180) \
[10] 2
start time (MET in s) (0:) [302313600]
```

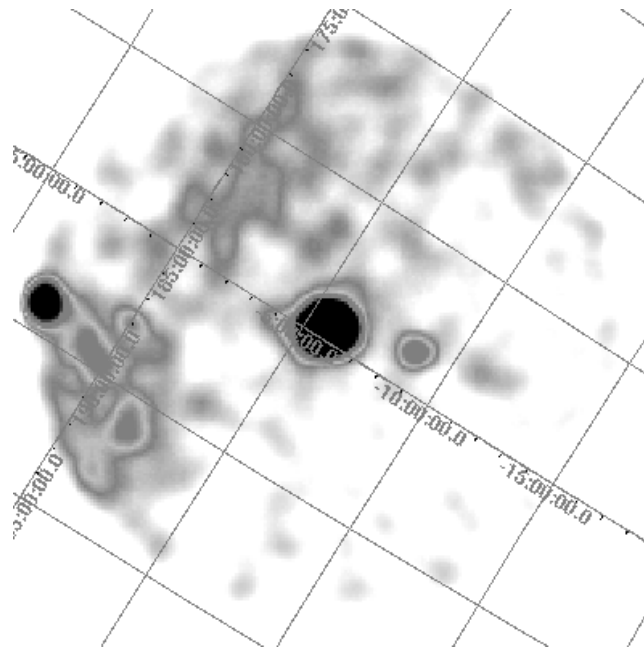


Рис. 13: Зображення ділянки неба навколо пульсару у Крабовидній туманності, згладжене по 8-ми пікселям та з накладеною координатною сіткою в галактичних координатах.

```
end time (MET in s) (0:) [306979200]
lower energy limit (MeV) (0:) [1000]
upper energy limit (MeV) (0:) [300000]
maximum zenith angle value (degrees) (0:180) [105]
Done.
```

Як видно, вказані при попередньому виклику команди, параметри запам'ятовуються, так що можна явно вказати лише те, що потрібно змінити, в нашому випадку – радіус кружечка. Тепер застосуємо фільтри.

```
gtmktime
Spacecraft data file[sc.fits]
Filter expression[DATA_QUAL==1 &&\
LAT_CONFIG==1 && ABS(ROCK_ANGLE)<52]
```

```
Apply ROI-based zenith angle cut[yes]
Event data file[ph_f.fits]
Output event file name[ph_f_g.fits]
```

І, нарешті, побудуємо криву блиску, скажімо, із кроком у 5 днів (432000 секунд). Для цього скористаємось командою `gtbin` із параметром `LC` (що означає Light Curve – крива блиску):

```
gtbin
This is gtbin version \
ScienceTools-v9r17p0-fssc-20100906
Type of output file (CCUBE|CMAP|LC|PHA1|PHA2) \
[] LC
Event data file name[] ph_f_g.fits
Output file name[] ph_lc.fits
Spacecraft data file name[] sc.fits
Algorithm for defining time bins (FILE|LIN|SNR) \
[] LIN
Start value for first time bin in MET[] 302313600
Stop value for last time bin in MET[] 306979200
Width of linearly uniform time bins in seconds[] \
432000
```

Для того, щоб правильно врахувати, скільки саме часу супутник реально спостерігав наш пульсар, потрібно розрахувати експозиції для кожного моменту часу (у “біні”) на кривій блиску. Це робиться командою `gtexposure`, яка допише результат у файл кривої блиску. Для того, щоб правильно обчислити експозицію, програмі `gtexposure` потрібно також вказати середній спектральний індекс джерела. В даному випадку ми використаємо значення, наведене у 2-му каталозі точкових джерел Fermi/LAT [14] – -2.2 .

```
gtexposure
Light curve file[] ph_lc.fits
Spacecraft file[] sc.fits
Response functions[] P7SOURCE_V6
Source model XML file[] none
Photon index for spectral weighting[] -2.2
```

Тут ми також вказали, яку модель чутливості інструменту використовувати – `P7SOURCE_V6`. Це важливо, адже ці моделі постійно покращуються. `P7SOURCE_V6` рекомендована для обробки спостережень точкових джерел [13], тому ми її і обрали.

Якщо в полі зору багато близьких джерел, тоді для правильної оцінки потоку потрібно вказати їхні спектри через параметр `Source model XML file`. В нашому випадку це не потрібно, тому параметр залишений пустим (`none`).

Тепер можна намалювати отриману криву. У файлі `ph_lc.fits`, який ми отримали, є п'ять колонок: `Time`, `TimeDel`, `Counts`, `Error` та `Exposure`. Перша і друга — це середина та ширина біну відповідно, третя і четверта — кількість фотонів в біні та пуасонова похибка їх числа, а остання — експозиція. Криву блиску варто будувати в одиницях [фотони/($\text{cm}^2 \cdot \text{сек}$)], для чого кількість фотонів у біні слід поділити на відповідну експозицію. Результат можна побачити на Рис. 14.

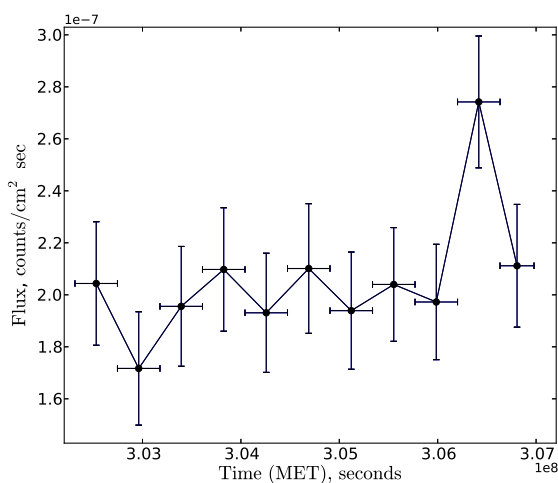


Рис. 14: Крива блиску пульсару у Крабовидній туманності.

2.6 Створення спектру

У порівнянні із процедурою отримання спектрів від рентгенівських телескопів ХММ-Ньютон та Чандра, у Fermi/LAT все трохи складніше. Складність полягає в тому, що PSF у Fermi/LAT великий і залежний від енергії. На практиці це означає, що у кожній окремій

точці картинка зміщується випромінювання від кількох сусідніх об'єктів і розділити його можливо далеко не завжди. Та сама проблема виникає і з фоновим випромінюванням – для слабких об'єктів немає способу сказати, чи прийшов певний фотон від об'єкту чи від фону.

Щоб вийти із цієї ситуації, у програмному забезпеченні **Fermi Science Tools** реалізований більш загальний підхід, заснований на використанні принципу максимальної правдоподібності (англ. likelihood).

Головна відмінність від аналізу рентгенівських спектрів тут полягає в тому, що, оскільки випромінювання окремих джерел не можна розділити, аналізувати їх потрібно разом. Оскільки в загальному випадку невідомо які/скільки джерел є у полі зору, ми мусимо висунути певне припущення щодо цього, виходячи із візуального аналізу отриманих зображень та/чи із більш строгих статистичних методів пошуку джерел. Фактично програмному забезпеченню потрібно передати список об'єктів у полі зору із їхніми спектральними моделями. Якщо досліджуваний об'єкт значно яскравіший за своїх сусідів, то ними часто можна знехтувати. На практиці гарною ідеєю є починати із останнього доступного каталогу точкових джерел, що випускається командою **Fermi** [14].

Для того, щоб задавати спектральні моделі джерел, пакет **Fermi Science Tools** використовує файли формату XML. Для аналізу доступні кілька простих типів спектрів, такі як степеневий, степеневий із експоненційним обрізанням та кілька інших. Повний їх список можна переглянути на відповідній веб-сторінці телескопу Фермі [15].

Для більшості джерел гамма-випромінювання спектри є, за своєю природою, нетепловими. Механізми прискорення високоенергетичних частинок, що створюють це випромінювання, зазвичай призводять до ступеневої форми спектру. Відхилення від останньої, пов'язані, наприклад, із досягненням максимальної енергії прискорення чи поглинанням гамма-променів на їх шляху до спостерігача, є також можливими і залежать від типу об'єкта (наприклад, пульсари та активні ядра, зазвичай, мають істотно різні форми спектру).

Для спрощення, тут ми будемо використовувати модель ступеневого спектру, яка має наступний загальний вигляд:

$$\frac{dN}{dE} = P \cdot \left(\frac{E}{E_0} \right)^{-\Gamma} \quad (1)$$

Тут P є нормою спектру, E_0 — характерною енергією, на якій потік

чисельно дорівнює нормі, а Γ є спектральним індексом, що визначає нахил спектру.

Давайте створимо файл із спектральною моделлю нашого пульсару. Як ми бачили раніше із Рис. 13, поблизу пульсару є ще одне яскраве джерело (2FGL J0521.7+2113, згідно з 2-им каталогом точкових джерел Fermi/LAT). Також присутнє суттєве фонове випромінювання представлене двома компонентами – неізотропною галактичною та ізотропною позагалактичною. Fermi Science Tools має моделі для обох цих компонент. Отже, наш файл `src_model.xml` виглядатиме наступним чином (“\” означає перенос):

```
<?xml version="1.0" ?>
<source_library title="source library">

  <source name="iso_p7v6source" type="DiffuseSource">
    <spectrum \
    file="$(FERMI_DIR)/refdata/fermi/diffuseModels/
    iso_p7v6source.txt" \
    type="FileFunction">
      <parameter free="1" max="1000" min="1e-05"
      name="Normalization" \
      scale="1" value="0.74"/>
    </spectrum>
    <spatialModel type="ConstantValue">
      <parameter free="0" max="10.0" min="0.0" \
      name="Value" scale="1.0" value="1.0"/>
    </spatialModel>
  </source>

  <source name="gal_2yearp7v6_v0" type="DiffuseSource">
    <!-- diffuse source units are\
    cm^-2 s^-1 MeV^-1 sr^-1 -->
    <spectrum type="ConstantValue">
      <parameter free="1" max="10.0" min="0.0" name="Value" \
      scale="1.0" value="1.20"/>
    </spectrum>
    <spatialModel \
    file="$(FERMI_DIR)/refdata/fermi/diffuseModels/
    gal_2yearp7v6_v0.fits" \
    type="MapCubeFunction">
      <parameter free="0" max="1000.0" min="0.001"
```

```

name="Normalization" \
scale="1.0" value="1.0"/>
</spatialModel>
</source>"

<source name="2FGL J0521.7+2113" type="PointSource">
<spectrum type="PowerLaw">
<parameter free="1" max="1000.0" min="1e-8" \
name="Prefactor" scale="1e-09" value="1.0"/>
<parameter free="1" max="-0.7" min="-5.0" name="Index" \
scale="1.0" value="-1.93"/>
<parameter free="0" max="2000.0" min="30.0" name="Scale" \
scale="1.0" value="100.0"/>
</spectrum>
<spatialModel type="SkyDirFunction">
<parameter free="0" max="360" min="-360" name="RA" \
scale="1.0" value="80.449"/>
<parameter free="0" max="90" min="-90" name="DEC" \
scale="1.0" value="21.22"/>
</spatialModel>
</source>

<source name="Crab pulsar (2FGL J0534.5+2201)"
type="PointSource">
<spectrum type="PowerLaw">
<parameter free="1" max="1000.0" min="1e-8"
name="Prefactor" \
scale="1e-09" value="1.0"/>
<parameter free="1" max="-0.7" min="-5.0" \
name="Index" scale="1.0" value="-2.14"/>
<parameter free="0" max="2000.0" min="30.0" \
name="Scale" scale="1.0" value="100.0"/>
</spectrum>
<spatialModel type="SkyDirFunction">
<parameter free="0" max="360" min="-360" name="RA" \
scale="1.0" value="83.632"/>
<parameter free="0" max="90" min="-90" name="DEC" \
scale="1.0" value="22.02"/>
</spatialModel>
</source>

```

</source_library>

Як видно, для наших двох точкових об'єктів ми задали тип спектру степеневий (параметр `spectrum type`), відповідну норму спектру P (параметр `Prefactor`) в одиницях 10^{-9} та спектральний індекс Γ (параметр `Index`). E_0 (параметр `Scale`) залишимо рівним 100 MeV — цей параметр не впливає на визначення спектру; формально необхідних для обезрозмірювання, цей параметр змінюють лише у деяких специфічних випадках, що ми їх тут розглядати не будемо. Зверніть увагу, що якщо спектральним параметрам ми дозволили варіюватися під час підгонки (аргумент `free = "1"`), то координати об'єктів є фіксованими (аргумент `free = "0"`). Програмне забезпечення **Fermi Science Tools** не дозволяє одночасно виконувати підгонку і того, і іншого.

Тепер, коли ми підготували інформацію про спектри, давайте виберемо дані для аналізу. Для кращого визначення спектру завантажимо всі доступні дані для нашого пульсару. Для цього у формі на сайті завантаження замість дати напишемо `START,END`. Фільтрацію зробимо так само, як ми робили у попередніх розділах, тільки змінимо кінцевий і початковий моменти часу.

```
gtselect evclass=2
Input FT1 file[ph.fits]
Output FT1 file[ph_f.fits]
RA for new search center (degrees) (0:360) \
[83.6331]
Dec for new search center (degrees) (-90:90) \
[22.0145]
radius of new search region (degrees) (0:180) \
[2] 10
start time (MET in s) (0:) [302313600] 239557417
end time (MET in s) (0:) [306979200] 341973363
lower energy limit (MeV) (0:) [1000]
upper energy limit (MeV) (0:) [300000]
maximum zenith angle value (degrees) (0:180) [105]
Done.
```

```
gtmktime
Spacecraft data file[sc.fits]
Filter expression[DATA_QUAL==1 && LAT_CONFIG==1 \
```



```

&& ABS(ROCK_ANGLE)<52]
Apply ROI-based zenith angle cut[yes]
Event data file[ph_f.fits]
Output event file name[ph_f_g.fits]

```

Сама процедура підгонки є достатньо ресурсоемною, і щоб хоч трохи її прискорити, команда Фермі розбила її на кілька етапів.

Спочатку будується деякий куб даних, який містить в собі час, який телескоп проводить, спостерігаючи на певну ділянку неба, як функцію кута між напрямком на об'єкт та віссю Z телескопу. Цей куб даних називається `livetime cube`.

Потім розраховується експозиція як функція координат на небі в межах заданої ділянки.

Коли завершено вищенаведені кроки, виконується підгонка даних заданою моделлю.

`Livetime cube` розрахувати просто, але достатньо довго. Робиться це програмою `gtlrcube`:

```

gtlrcube
Event data file[] ph_f_g.fits
Spacecraft data file[] sc.fits
Output file[] ph_expCube.fits
Step size in cos(theta) (0.:1.) [] 0.025
Pixel size (degrees)[] 1
Working on file sc.fits
.....!

```

Тепер побудуємо карту експозиції (фактично, розподіл експозиції як функції координат на небі для кількох значень енергії фотонів) за допомогою програми `gtexpmap`:

```

gtexpmap
The exposure maps generated by this tool are meant
to be used for *unbinned* likelihood analysis only.
Do not use them for binned analyses.
Event data file[] ph_f_g.fits
Spacecraft data file[] sc.fits
Exposure hypercube file[] ph_expCube.fits
output file name[] ph_expMap.fits
Response functions[] P7SOURCE_V6
Radius of the source region (in degrees)[] 22
Number of longitude points (2:1000) [] 120
Number of latitude points (2:1000) [] 120

```

```

Number of energies (2:100) [20]
The radius of the source region, 22, should
be significantly larger (say by 10 deg)
than the ROI radius of 15
Computing the ExposureMap using ph_expCube.fits
.....!

```

Розмір ділянки, для якої будується карта, повинен бути значно більшим розміру ділянки, яку ми аналізуємо, щоб можна було правильно врахувати вклади об'єктів за межами поля зору. Саме це нам правильно підказує `gtexpmap` при запуску. Тут ми вказали використати 20 значень енергії (між максимальним і мінімальним, обраними на попередніх кроках). Це впливає на точність апроксимації ефективної площі, що використовується при підгонці спектру. Число 20 рекомендоване розробниками Fermi/LAT.

Тепер можна запустити підгонку спектру (службові повідомлення в кінці опущено):

```

gtlike
Statistic to use (BINNED|UNBINNED) [] UNBINNED
Spacecraft file[] sc.fits
Event file[] ph_f_g.fits
Unbinned exposure map[] ph_expMap.fits
Exposure hypercube file[] ph_expCube.fits
Source model file[] src_model.xml
Response functions to use[] P7SOURCE_V6
Optimizer (DRMNFB|NEWMINUIT|MINUIT|DRMNGB|LBFGS)\
[] DRMNFB
...

```

`gtlike` дозволяє попередньо згрупувати фотони на кілька інтервалів по енергії, що може прискорити підгонку. Ціна за прискорення – можлива гірша точність визначення параметрів спектру. В нашому випадку спектр розраховується достатньо швидко, і необхідності у такому групуванні немає. Саме тому ми при аналізі використовуємо статистику `Unbinned`. Оптимізатор можна обрати на свій розсуд – його вибір впливає на якість підгонки та необхідний для неї час. Команда Фермі рекомендує використовувати `DRMNFB` для швидкого попереднього пошуку параметрів підгонки і `NEWMINUIT` для їх уточнення.

Через деякий час `gtlike` повідомить нас про результати своєї роботи (початок виводу опущено):

....

Photon fluxes are computed for the energy range
1000 to 300000 MeV

2FGL J0521.7+2113:
Prefactor: 0.7295 +/- 0.105018
Index: -1.9423 +/- 0.0381131
Scale: 100
Npred: 975.072
ROI distance: 3.06483
TS value: 2309.86
Flux: 8.81455e-09 photons/cm²/s

Crab pulsar (2FGL J0534.5+2201):
Prefactor: 52.1329 +/- 1.69684
Index: -2.34417 +/- 0.0101579
Scale: 100
Npred: 19226.3
ROI distance: 0.00559374
TS value: 105643
Flux: 1.75882e-07 photons/cm²/s

gal_2yearp7v6_v0:
Value: 1.00561 +/- 0.00379973
Npred: 88995.6
Flux: 4.75454e-05 photons/cm²/s

iso_p7v6source:
Normalization: 1e-05 +/- 0.0267264
Npred: 0.0592768
Flux: 7.09122e-11 photons/cm²/s

WARNING: Fit may be bad in range [1000, 1330.01]\
(MeV)
WARNING: Fit may be bad in range [2352.71, 3129.13]\
(MeV)
WARNING: Fit may be bad in range [127513, 225562] \
(MeV)

Total number of observed counts: 109197

Total number of model events: 109197

$-\log(\text{Likelihood})$: 1371734.45

Elapsed CPU time: 316.98

Як бачимо, для кожного джерела із файлу моделей `gtlike` виводить підігнані параметри спектру та їх похибки. Ці параметри також автоматично записуються у файл `results.dat`. Величина TS (від англ. Test statistics), яка наведена для кожного об'єкту, окрім двох компонент фону, визначає достовірність детектування джерела. Для достатньо великих значень TS (~ 10) достовірність детектування (в одиницях σ) дорівнює \sqrt{TS} , тобто наш пульсар зареєстровано на рівні $\sqrt{105643} \approx 325\sigma$.

Насправді спектр нашого об'єкта може і не бути степеневим, це було лише наше припущення. У нашому випадку це справді так: в GeV-ному діапазоні світить як сам пульсар, так і туманність, що його оточує. Але для того, щоб побачити форму спектра, треба виконати спектральний аналіз, для якого вже потрібно припустити шукану форму спектру.

Для того, щоб вийти із цього зачарованого кола (знання форми спектру необхідне для того, щоб виконати підгонку, але сама підгонка необхідна, щоб отримати уявлення про вигляд спектру) ми зробимо наступне. Оскільки спектральних ліній у гама-діапазоні не буває (окрім, можливо, ліній розпаду/анігіляції темної матерії), то ми можемо очікувати, що спектр змінюється з енергією плавно. Таким чином, якщо ми розіб'ємо наш інтервал енергії (1–300 GeV) на деяку кількість менших, то можемо очікувати, що в кожному з них спектр буде дуже близький до степеневого. Якщо це виявиться не так, ми завжди можемо збільшити кількість цих маленьких інтервалів, зменшивши їх ширину. Слід мати на увазі, що таку процедуру не має сенсу повторювати нескінченно, адже Fermi/LAT має скінченну енергетичну роздільну здатність, приблизно рівну 10%.

Для прикладу, розіб'ємо інтервал 1–300 GeV на 15 менших у логарифмічному масштабі — так зручніше, адже маємо два порядки по енергії. Оскільки нас цікавить не форма спектру в кожному окремому інтервалі, а лише сумарний потік, то ми “заморозимо” спектральні індекси джерел у файлі з моделями, дозволивши підгонку лише норм спектрів. Це робиться присвоюючи параметру `free` значення нуль у стрічках файлу моделей, де визначено спектральні індекси.

Таким чином ми пришвидшимо роботу підгонки і одразу отримаємо оцінку потоку на кожному інтервалі – вона тепер прямо визначається похибкою норми, що її нам повідомить `gtlike`. Звичайно, для такої процедури зручніше огорнути команди у деякий скрипт, що дозволить зручно будувати спектри з заданим наперед числом енергетичних інтервалів. Читачам пропоную написати такий скрипт самостійно, якщо є така необхідність.

Тепер, маючи на руках результати підгонки спектру у всіх 15 інтервалах, можна намалювати сумарний спектр. `gtlike` повернув нам параметри, які можна прямо підставити у рівняння 1, щоб отримати диференційний спектр dN/dE . Слід пам'ятати, що норма при підгонці відраховується в одиницях 10^{-9} . Одиниці виміру отриманого таким чином спектру – [фотони/($\text{см}^2 \cdot \text{сек} \cdot \text{MeV}$)]. Зазвичай малюють $E^2 dN/dE$, так зробимо і ми. Також ми переведемо MeV-и у eV-и. Результат представлено на Рис. 15.

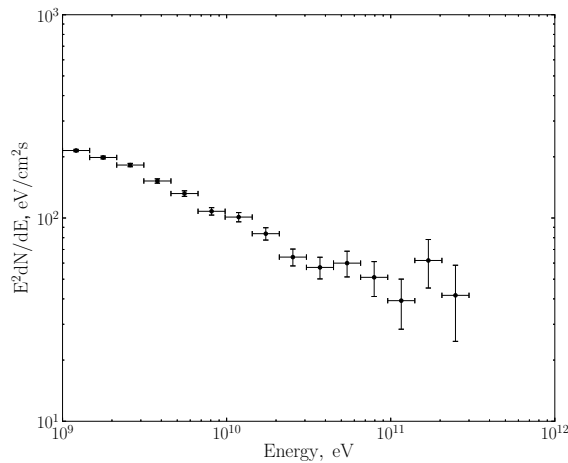


Рис. 15: Спектр пульсару у Крабовидній туманності (насправді, спектр не самого пульсару, а пульсару та туманності, що його оточує)

Оскільки ми використали малі інтервали по енергії, то могло так статися, що в деяких з них опинилося б замало фотонів. Це можна контролювати величиною `TS`, яка повідомляється після кожної під-

гонки. Якщо TS стає меншим за 10, то джерело зареєстроване в такому інтервалі на рівні $\sim 3\sigma$ і оцінка потоку разом із своєю похибкою може бути невірною. Таким чином, результати нашої підгонки в таких інтервалах використовувати не можна. Якщо є необхідність хоча б отримати обмеження на потік від джерела в такій ситуації, то слід використовувати більш складний аналіз, який потребує використання мови програмування `Python` (саме для неї існує інтерфейс `Fermi Science Tools`, який дозволяє більш гнучке оперування даними).

На останок рекомендується регулярно проглядати наступні інтернет-ресурси:

- Сторінка Fermi/LAT віртуальної обсерваторії VIRGO.UA [16];
- Офіційна документація телескопу Фермі [17].

Заключні зауваження

Аналіз даних — часто не проста справа, але не слід забувати, що головне не дані, а те, що за ними стоїть. Невірно виконаний аналіз може викривити результати, і призвести до цілком інших, скоріше за все неправильних висновків. Але не слід робити аналіз просто заради аналізу. Надмірне його ускладнення (у випадку Fermi/LAT, скажімо, включення багатьох джерел до аналізу, використання занадто вузьких енергетичних інтервалів) також може привести до викривлених результатів. Тому, експериментуйте! Експериментуйте, оцінюючи результати зі здоровим глуздом. Пам'ятайте, програмне забезпечення писали люди, тож воно не застраховане від можливих помилок.

У цьому посібнику продемонстровано основні елементи обробки даних Чандра та Fermi/LAT, які найчастіше використовуються у повсякденній роботі. Проте, різні ситуації і питання, що на них шукають відповіді спостереженнями, можуть диктувати свої умови, істотно змінюючи загальну процедуру, наведену вище.

Подяки

Автори висловлюють щире подяку рецензентам даного рукопису, без вагомих зауважень котрих даний посібник не був би зроблений належним чином. Також висловлюється подяка особисто Вавиловій Ірині Борисівні за сприяння у друці та редагування даного посібника, а також Добричевій Дар'ї Вікторівні за детальну перевірку та Іващенко Ганні Юріївні за технічне редагування посібника.

Посібник підготовлено в рамках проекту Цільової комплексної програми НАН України з наукових космічних досліджень.

Література

- [1] Aldcroft T. L., Karovska M., Cresitello-Dittmar M. L., Cameron R. A., Markevitch M. L. Initial performance of the aspect system on the Chandra Observatory: postfacto aspect reconstruction // Proc. SPIE. X-Ray Optics, Instruments, and Missions III, eds.: Truemper J. E., Aschenbach B. — 2000. — V. 4012. — P. 650-657.
- [2] Arnaud K. A. XSPEC: The First Ten Years // Astronomical Data Analysis Software and Systems V, ASP Conference Series, eds.: Jacoby J. H., Barnes J. — 1996. — V. 101. — P. 17.
- [3] Cavaliere A., Fusco-Femiano R. X-rays from hot plasma in clusters of galaxies // Astron. & Astrophys. — 1976. — V. 49. — P. 137-144.
- [4] Gelb A. “Applied Optimal Estimation”, 1974, M.I.T. Press, Cambridge, MA.
- [5] Ma C., Arias E. F., Eubanks T. M. et al. The International Celestial Reference Frame as Realized by Very Long Baseline Interferometry // Astrophys. J. — 1998. — V. 116. — P. 516-546.
- [6] <http://icxc.harvard.edu/icd/FITS/ascts2.0.ps>
- [7] <http://virgo.org.ua>
- [8] <http://heasarc.gsfc.nasa.gov>
- [9] <http://cxc.harvard.edu/>
- [10] <http://cxc.harvard.edu/ciao/>
- [11] <http://cxc.harvard.edu/cal/>
- [12] <http://fermi.gsfc.nasa.gov/cgi-bin/ssc/LAT/LATDataQuery.cgi>
- [13] http://fermi.gsfc.nasa.gov/ssc/data/analysis/documentation/Cicerone/Cicerone_Data_Exploration/Data_preparation.html
- [14] http://fermi.gsfc.nasa.gov/ssc/data/access/lat/2yr_catalog/
- [15] http://fermi.gsfc.nasa.gov/ssc/data/analysis/scitools/-source_models.html#spatialModels
- [16] http://wiki.virgo.org.ua/index.php/Starting_with_Fermi/LAT

[17] <http://fermi.gsfc.nasa.gov/ssc/data/analysis/documentation/>

Додаткова література

- [18] Бисноватый-Коган Г. С. Релятивистская астрофизика и физическая космология. КРАСАНД. — 2010. — 362 с.
- [19] Засов А. В., Постнов К. А. Общая астрофизика. — Фрязино: Век-2, 2006. — 494 с.
- [20] Соболев В. В. Курс теоретической астрофизики. — М.: Наука. — 1985. — 504 с.
- [21] Тугай А. В. Рентгенівська астрономія. Методи отримання зображень, кривих блиску і спектрів за супутниковими спостереженнями: методичний посібник для студентів кафедри астрономії та фізики космосу. К.: ВПЦ “Київський університет”. — 2012. — 42 с.
- [22] Чеснок Н. Г. Обробка спектрів зір та позагалактичних об’єктів за допомогою програмного забезпечення VOSpec, Методичний посібник. К.: ВАІТЕ. — 2010. — 44 с.