Automated Search for Tidal Flare Candidates In Chandra Archival Data

Matthew P. Wampler-Doty

Introduction

We present a catalog of 57765 point sources derived from archival Chandra data of distant ($z \in [.05, .3]$) galaxy clusters. For each of these point sources, fluxes for 5 energy bands are calculated, in many cases along with background data. The resulting catalog has over 4.6 million fluxes, weighing in at approximately 220MB. This catalog is suitable for the discovery of X-Ray transients as follow up to the research conducted by Maksym et al. [1] and Maksym et al. [2].

1 Overview

In this section we provide an overview of the data processing pipeline developed. Code for this is available at the following URL:

`https://github.com/xcthulhu/Galaxy-Clusters`

Pseudo-code outlining the entire system is given in Fig. 1. The results of the data processing pipeline are placed into an SQLite database, described in §2.

§1.1 `observations` ← Query HEASARC for all chandra observations

§1.2.1 `observation_tree` ← Use *complete linkage hierarchical agglomerative clustering* on `observations`

§1.2.2 `groups` ← Traverse `observation_tree` to find groups of observations seperated by $8'$

§1.3 `galaxy_clusters` ← For each group in `groups`, collect those groups that have a galaxy cluster in their vicinity according to NED

For each galaxy cluster `c` in `galaxy_clusters`:

`sources` ← initialize as empty []

For each observation `obs` in galaxy cluster `c`:

§1.4 Retrieve `obs` with `download_chandra_obsid`

§1.5 Run `chandra_repro` to repair `o`

§1.6 `sources` ← `sources` ∪ sources detected for `obs` using `wavdetect`

§1.7 `cluster_tree` ← Use *complete linkage hierarchical agglomerative clustering* on `sources`

`unique_sources` ← Traverse `cluster_tree` to find groups of sources separated by $5''$, and choose a random member from that group

For each `source` in `unique_sources`:

For each band `A` – `B` eV in {200 – 500, 500 – 1000, 1000 – 2000, 2000 – 4500, 4500 – 12000}:

For each observation `o` in galaxy cluster `c`:

§1.8 `D` ← Compute the 85% *encircling energy* (EE) diameter for energy $\frac{\texttt{B} - \texttt{A}}{2}$ eV

§1.9 `source_fluxes` ← Extract all of the fluxes for the events within $\frac{\texttt{D}}{2}$ of `source`

`background_fluxes` ← Extract all of the fluxes for the events which are between $\frac{3\,\texttt{D}}{4}$ and $\frac{3\,\texttt{D}}{2}$ away from `source`

Figure 1: Pseudo-code outlining the data processing system

1.1 HEASARC Queries

The first step in our data process pipeline is the automated retrieval of the listings for every Chandra observations in the HEASARC database[1]. We make use of python's urllib and urllib2 modules to carry out automated queries to a NASA's CGI script for interfacing this database. Some additional filtering is necessary afterwords - in particular, observation IDs corresponding to planned observations and unreleased data are discarded, as well as all observations within $15'$ of the galactic plane, and only Chandra observations using the pc are considered.

Once all of the observations have been acquired, we face the following difficulty: the labels assigned to objects are not strictly consistent between missions. It is not feasible to use the tags provided by researchers to classify observations of the same object.

1.2 Agglomerating Observations

Our solution to the lack of consistent labels in the Chandra data is to use an variety of unsupervised machine learning known as *hierarchical agglomerative clustering*. This allows us to classify groups of observations close to one another in the sky, eliminating the need rely on researcher labels.

1.2.1 Fast Complete Linkage Hierarchical Agglomerative Clustering

Hierarchical agglomerative clustering is an unsupervised learning algorithm for breaking data into groups of pairs of nearby data-points, and then into groups of pairs of nearby groups, and so on recursively. This technique is commonly employed in *computational phylogenetics,* a branch of evolutionary ecology [3].

[1]NASA's CGI script for interfacing with HEASARC, using `POST` methods, can be found here: `http://heasarc.gsfc.nasa.gov/db-perl/W3Browse/w3query.pl` For more information on http protocols: for more information on http protocols, see `http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html`.

We repurpose one of these algorithms for classifying groups of observations by position.

Abstractly, hierarchical agglomerative clustering algorithms compute a tree representing hierarchical relationships between points in some set. In agglomerative clustering, each point is grouped with its nearest neighbor according to some metric and rule system. The system then recurses, only now over points and previously computed groups of points, building a new stage of the tree. The process ends when all of the points are grouped together. A terminal node of the resulting tree represents one the original points, while branches represent groups of points. We always use the *complete linkage* rule in our applications, where groups of points in the tree are labeled with a value representing the maximum distance the points are apart, according to their *great circle distance*[2].

As of the time of the writing of this document, HEASARC reports that there are thousands of suitable archival Chandra observations. We have found that given the number of observations, agglomerative clustering is intractable without a fast algorithm. We use the efficient `fastcluster` python module[3] developed by Daniel Müllner.

1.2.2 Computing Agglomerations of Observations

After computing the agglomeration hierarchy, we find each group of observations separated by at most $8'$ and output the observation IDs of that group to a designated file. This is done via top-down recursive tree traversal[4]. As of the time of the writing of this document, the system computes 5846 groups of

[2]The *great circle distance* for two vectors $n_1 = \langle x_2, y_2, z_2 \rangle$ and $n_2 = \langle x_2, y_2, z_2 \rangle$ on the unit sphere S^3 can be computed via $arctan(|n_1 \times n_2|/(n_1 \cdot n_2))$, or via a special form of the *Vincenty Formula* [4] in the case of spherical coordinates.

[3]The `fastcluster` python module is available here: `http://math.stanford.edu/~muellner/fastcluster.html`. The code implements the CLINK algorithm in [5], which runs in $O(n^2)$ time (where n is the number of points being clustered). Other library implementations of agglomerative clustering, such as the one in the scientific python module `scipy`, often run in $O(n^3)$ time, which is too slow for our purposes.

[4]For an overview of top-down tree traversal algorithms, see [6, §3.4, pp. 77-83]

observations $8'$ apart.

1.3 NED Queries

Not every group found in §1.2.2 is suitable for analysis. We are only interested in observations of galaxy clusters. To find only those groups of observations corresponding to a galaxy cluster, we use Caltech's NED database[5] as the basis for further processing. For each agglomeration of observations, we query NED to find all of the galaxy clusters within the vicinity. An arbitrary observation o from each group of observations is selected, and a query is sent to NED for all galaxy cluster objects in a $15'$ of o. Queries are automated using a python script using `urllib` and `urllib2` just as in the case of HEASARC.

After all of the NED queries have been computed, the results are filtered such that only galaxy clusters with z-values in $[.05, .3]$ are kept. Any agglomeration which does not have a suitable galaxy cluster in its vicinity is discarded.

Figure 3 depicts the resulting positions of algorithmically identified observations of galaxy clusters.

1.4 Data Retrieval

In this section we detail how data is retrieved corresponding to the previously grouped observations.

After each group of suitable observations is found, we retrieve data files For each observation in those groups.

To date, hundreds of gigabytes of data have been downloaded as part of our survey. Each Chandra observation was retrieved using the `download_chandra_obsid` script[6], available in the CIAO[7] data analysis software for chandra.

[5] Caltech's NED database can be accessed using `GET` methods and the following CGI script: `http://ned.ipac.caltech.edu/cgi-bin/nph-objsearch`.

[6] `http://cxc.harvard.edu/ciao/ahelp/download_chandra_obsid.html`

[7] CIAO is available from the following website: `http://cxc.harvard.edu/ciao/`

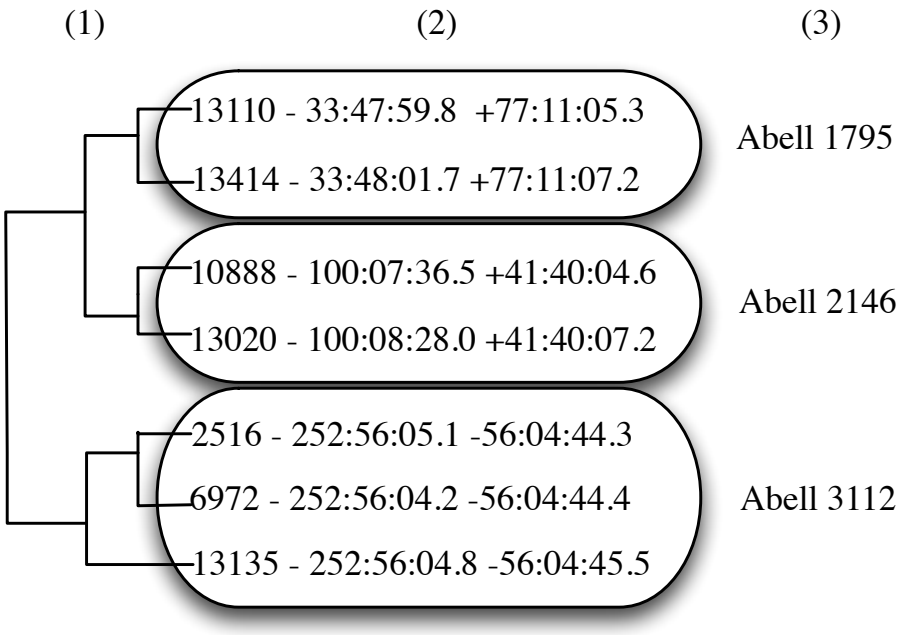

Figure 2: A small scale example of: (1) Complete linkage hiearchical agglomerative clustering, producing a phylogenetic tree (2) Using the phylogenetic tree to make groups of observations $8'$ apart and (3) cross referencing NED to find the corresponding galaxy cluster names

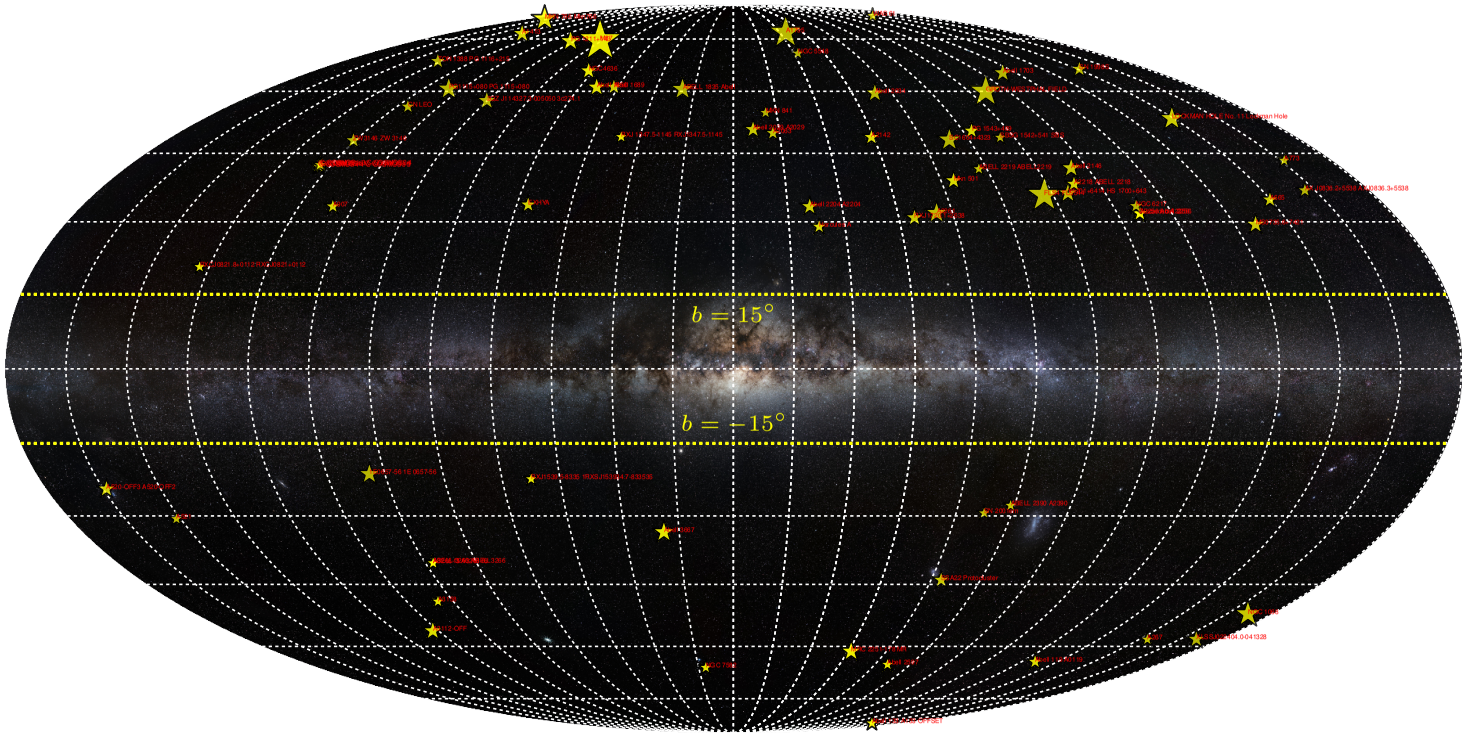

Figure 3: The result of hierarchical agglomerative clustering (§1.2.1), followed by cross-referencing with known galaxy clusters according to NED (§1.3)

1.5 Reprocessing

Due to a number of errors in the preprocessing of the Chandra archival data, it is necessary to run a reprocessing script on each observation before any further data manipulation is performed[8]. Reprocessing is carried out using the `chandra_repro` script[9] from the CIAO software suite.

1.6 Point Source Detection

After reprocessing, sources for a particular observation are extracted using `wavdetect`[10]. This implements the following algorithm:

1. `wtransform` detects probable source pixels within a dataset by repeatedly correlating it with *Mexican hat* wavelet functions with different scale sizes.

2. `wrecon` generates a source list with information from each wavelet scale. For each source, a cell is computed that contains the majority of the source flux, and source properties are computed within that cell.

1.7 Point Source Agglomeration

After sources are detected for each observation, they are combined using hierarchical agglomerative clustering to remove duplicates, using the same algorithm used in §1.2. To make computation more efficient, only sources from observations of the same galaxy cluster are considered. All of the sources in all of the observations of a particular galaxy cluster are grouped into agglomerations of size $5''$. From each group of point sources, an arbitrary source is selected.

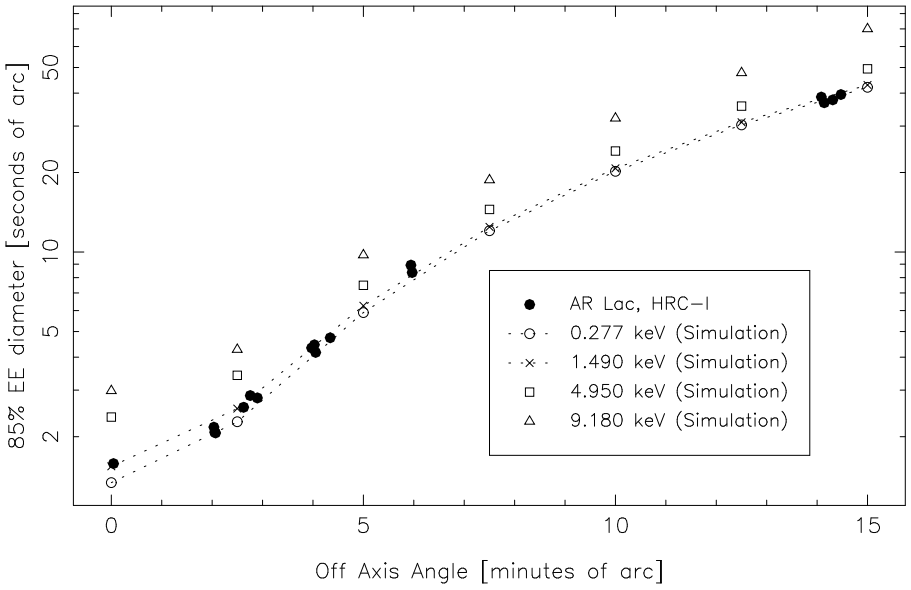

Figure 4: Off-axis 85% EE diameters (from simulations) [7]

1.8 Energy Octave Extraction

For each point source in each galaxy cluster, for each observation in that galaxy cluster, fluxes in the 85% *effective energy* (EE) circle are extracted. The octaves we extract energy from are:

- 200 – 500 eV
- 500 – 1000 eV
- 1000 – 2000 eV
- 2000 – 4500 eV
- 4500 – 12000 eV

[8] A detailed discussion of the reasons for reprocessing Chandra data files can be found here: `http://cxc.harvard.edu/ciao4.4/threads/createL2/index.html`

[9] `http://cxc.harvard.edu/ciao/ahelp/chandra_repro.html`

[10] `http://cxc.harvard.edu/ciao/ahelp/wavdetect.html`

For each octave, the 85% EE diameter is computed using the *average* energy of that octave (ie, 350 eV for the 200–500 eV octave, 750 eV for the 500 – 1000 eV octave, etc). We compute the 85% EE using `psffrac.py`[11], a utility which calls CIAO's model of the chandra *point spread function* PSF. The 85% EE off-axis circle diameters for point sources of varying energy levels are given in Fig. 4. Events in the observation are extracted using the CIAO program `dmcopy`[12]. For each band, for each source, once its 85% EE diameter `D` is computed, background events are extracted in an annulus with inner diameter $\frac{3D}{2}$ and outer diameter `3D`, again using `dmcopy`.

1.9 Flux Calculation

After photons around each source events are extracted, the `eff2evt` command[13] is used to extract fluxes from each detected photon p. The flux ($\frac{\text{ergs}}{\text{cm}^2 \times \text{sec}}$) for each photon is computed using the following formula:

$$flux_p = \frac{1}{QE \times EA \times DA \times LIVETIME}$$

Where

- QE is the *quantum efficiency*, determined by the ACIS instrument and position where the photon hit.
- EA is the *effective area* of the photon, governed by the ASIS instruments point spread function.
- DA is the *dead area* correction factor. Over time, the cosmic ray flux incident on the ACIS instrument reduces its usable area, making it necessary

[11] `psffrac.py` is available through github: `https://raw.github.com/xcthulhu/Galaxy-Clusters/master/bin/psffrac.py`

[12] `http://cxc.harvard.edu/ciao/ahelp/dmcopy.html`

[13] `http://cxc.harvard.edu/ciao/threads/eff2evt/index.html#acis.phflux`

to take in to account when measuring photon flux. The correction ranges from approximately -2.2% at the readout to -4% at the top of the chip[14].

- $LIVETIME$ is the total exposure time for an observation. A number of correction factors need to be taken into account. For the ACIS instrument this excludes:

  (a) the time it takes to transfer charge from the image region to the frame store region (0.04104 seconds per frame)

  (b) time during preflushes when they are necessary

  For more information on how $LIVETIME$ is calculated, see `http://cxc.harvard.edu/ciao/ahelp/times.html`.

In Fig. 1 the `total_flux` for an observation `obs` is $\sum_{p \in \texttt{obs}} flux_p$.

2 Data Base Summary

Our analysis covers over 1.9×10^4 sources taken from 19 agglomerative clusters (as described in §1.2.1). Each agglomerative cluster is in the vicinity of a number of galaxy clusters according to the NED catalog. This information is summarized in Table 1.

The database schema is summarized in Table 1. Notably:

- `AGGLOMERATIONS` contains the metadata in Table 1 apart from the catalog information.
- `SOURCES` does not have `UNIQUE(RA,DEC)` as a database constraint, for efficiency concerns when generating the table.
- As discussed in the introduction, `SOURCES` has 57765 entries, and `FLUXES` has 4600670 entries

[14] For details on how the dead area correction factor is calibrated, see [8]

	RA	DEC	Sources	# of Obs.	First Observation	Final Observation	Duration	Catalog Name	z-value
1	12:30:49.50	12:23:28.0	5566	92	2000-04-20 20:30:26	2010-05-14 14:03:53	10.1 years	[HB84] Cluster	0.086019
2	13:48:52.70	26:35:27.0	2216	42	2000-03-21 13:53:45	2012-04-08 17:02:36	12.1 years	WHL J134905.4+263153	0.2765
								MaxBCG J207.26074+26.54908	0.26465
								[EAD2007] 148	0.0896
								WHL J134852.5+263534	0.069
								ABELL 1795	0.062476
3	2:42:40.70	-0:00:48.0	2144	13	2000-02-21 21:47:15	2008-12-05 14:23:41	8.8 years	SDSS CE J040.633698+00.003835	0.219981
								SDSS CE J040.633717-00.057119	0.07248
4	14:17:43.60	52:28:41.2	894	18	2002-08-13 03:30:56	2005-12-11 14:30:03	3.3 years	RCS J141803+5223.1	0.271
5	11:18:17.00	7:45:59.4	581	8	2000-11-03 14:17:10	2010-02-15 17:45:29	9.3 years	PG 1115+080:MWKZ GROUP 3	0.277
6	4:54:05.40	2:53:35.0	549	6	2003-12-04 19:48:38	2008-01-12 04:18:18	4.1 years	1RXS J045408.8+025506	0.203
								ABELL 0520	0.199
7	12:58:41.30	-1:45:41.0	323	7	2005-08-10 12:19:40	2007-02-25 01:01:36	1.5 years	MaxBCG J194.67892-01.77814	0.11615
								[EAD2007] 027	0.10486
								WHL J125841.5-014541	0.1023
								SDSS-C4 1041	0.08392
								ABELL 1650	0.083838
								NSC J125844-014226	0.0806
8	13:11:29.50	-1:20:30.0	255	5	2001-01-07 14:18:34	2007-03-07 17:59:43	6.2 years	SDSS CE J197.823410-01.237907	0.197288
								NSC J131122-012108	0.1904
								ABELL 1689	0.1832
								WHL J131132.1-011946	0.1793
								GMBCG J197.89393-01.30305	0.153
9	15:10:13.40	33:30:43.0	213	5	2007-05-07 19:34:06	2010-11-28 11:35:01	3.6 years	MaxBCG J227.67170+33.51785	0.28355
								NSCS J150948+333548	0.13
								SDSS +227.6+33.5+0.12	0.12155
								NSCS J150945+332826	0.12
								ABELL 2034	0.113
								WHL J150959.8+332746	0.1093
								NSC J151001+332906	0.1092
10	12:42:50.00	2:41:17.0	210	3	2000-01-26 15:19:12	2003-02-15 08:54:11	3.1 years	SDSS-C4 1106	0.08583
								ABELL 1599	0.0855
11	3:17:57.60	-44:14:17.0	143	5	2001-09-15 05:33:55	2011-03-14 06:19:24	9.5 years	ABELL 3112	0.075252
12	4:54:22.10	-10:16:15.9	115	3	2000-10-13 13:45:21	2010-11-27 05:31:06	10.1 years	ABELL 0521	0.2533
13	1:02:41.80	-21:52:50.0	68	3	2002-06-24 11:33:56	2008-08-29 21:24:07	6.2 years	ABELL 0133	0.0566
14	8:30:59.25	65:50:26.0	67	4	2002-12-28 16:28:52	2011-01-09 18:53:10	8.0 years	ZwCl 0826.1+6554	0.21
								ABELL 0665	0.1819
								NSC J083017+654939	0.167
15	13:47:31.00	-11:45:11.0	24	3	2000-04-29 15:26:12	2003-09-03 21:49:41	3.3 years	[BGV2006] 043	0.086
16	11:55:18.10	23:24:17.0	13	3	2001-05-16 14:32:54	2007-07-10 05:12:12	6.2 years	WHL J115518.0+232417	0.1499
								ABELL 1413	0.1427
								NSCS J115519+231840	0.11
								GMBCG J178.81296+23.46510	0.105
17	15:58:15.10	27:14:43.0	13	3	1999-08-20 20:23:42	2007-05-07 17:48:52	7.7 years	NSC J155821+271549	0.1276
								SDSS +239.6+27.2+0.11	0.10922
								MaxBCG J239.58334+27.23341	0.10265
								WHL J155820.0+271400	0.0952
								ABELL 2142	0.0909
18	17:20:09.91	26:37:30.0	12	4	1999-10-19 15:26:35	2002-10-03 20:45:28	3.0 years	NSC J172013+264028	0.2206
								SDSS-C4 3072	0.164
								FSVS_CL J172015+263858	0.161
								WHL J172010.0+263732	0.1573
19	16:35:52.80	66:12:50.4	2	3	1999-10-19 20:29:16	2007-06-13 12:25:28	7.7 years	ABELL 2218	0.1756

Table 1: Agglomerative clusters, along with galaxy clusters in their vicinity from the NED catalog

Table Name	Column Name	Type	Database Constraints
BANDS			
	ID	INTEGER	PRIMARY KEY NOT NULL
	STARTEV	FLOAT	NOT NULL
	ENDEV	FLOAT	NOT NULL
Additional Constraints:	UNIQUE(STARTEV, ENDEV) ON CONFLICT IGNORE		
AGGLOMERATIONS			
	ID	INTEGER	PRIMARY KEY NOT NULL
	RA	FLOAT	NOT NULL
	DEC	FLOAT	NOT NULL
Additional Constraints:	UNIQUE(RA,DEC) ON CONFLICT IGNORE		
OBSIDS			
	ID	INTEGER	PRIMARY KEY NOT NULL
	RA	FLOAT	NOT NULL
	DEC	FLOAT	NOT NULL
	DATE	INTEGER	NOT NULL
	DURATION	INTEGER	NOT NULL
	AGGLOMERATIONID	INTEGER	NOT NULL
	TYPE	STRING	NOT NULL
	CATALOG_NAME	STRING	NOT NULL
	INSTRUMENT	STRING	NOT NULL
Additional Constraints:	FOREIGN KEY(AGGLOMERATIONID) REFERENCES AGGLOMERATIONS(ID) UNIQUE(RA,DEC,DATE) ON CONFLICT IGNORE		
SOURCES			
	ID	INTEGER	PRIMARY KEY NOT NULL
	RA	FLOAT	NOT NULL
	DEC	FLOAT	NOT NULL
	AGGLOMERATIONID	INTEGER	NOT NULL
Additional Constraints:	FOREIGN KEY(AGGLOMERATIONID) REFERENCES AGGLOMERATIONS(ID)		
FLUXES			
	ID	INTEGER	PRIMARY KEY NOT NULL
	RADIUS	FLOAT	NOT NULL
	FOREGROUND	FLOAT	
	BACKGROUND	FLOAT	
	OBSID	INTEGER	NOT NULL
	SOURCEID	INTEGER	NOT NULL
	BANDID	INTEGER	NOT NULL
Additional Constraints:	UNIQUE(SOURCEID,OBSID,BANDID) ON CONFLICT IGNORE FOREIGN KEY(OBSID) REFERENCES OBSIDS(ID) FOREIGN KEY(SOURCEID) REFERENCES SOURCES(ID) FOREIGN KEY(BANDID) REFERENCES BANDS(ID))		

Table 2: Schema for tables in SQL database

- While `FOREGROUND` data is always available, sadly some `BACKGROUND` data is missing in the `FLUXES` table. This was due to a fatal disk error on a computer that was not properly backed up.

- The data is presented in a form known in database engineering as *renormalized*. As a result, no data is duplicated; when querying for a FLUX, a reference to its source is associated with it rather than an RA and DEC. This means that the data is naturally compact; in fact uncompressed the data takes up only 220 MB; compressed it takes up 102 MB.

Final Remarks

We have outlined a large catalog of point sources and fluxes, suitable for the discovery of tidal flare disruption events. This work opens up the possibility of follow up research into analyzing the time series from arising from these fluxes.

References

[1] P. Maksym, M. Ulmer, Constraining the tidal flare rate with rich galaxy clusters, Bulletin of the American Astronomical Society 42 (2010) 665.

[2] W. Maksym, M. Ulmer, M. Eracleous, A tidal disruption flare in a1689 from an archival x-ray survey of galaxy clusters, The Astrophysical Journal 722 (2010) 1035.

[3] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical Recipes: The Art of Scientific Computing, 3rd Edition, Cambridge University Press, 2007.

[4] T. Vincenty, Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations, Survey review 23 (176) (1975) 88–93.

URL http://www.maneyonline.com/doi/abs/10.1179/sre.1975.23.176.88

[5] D. Defays, An efficient algorithm for a complete link method, The Computer Journal 20 (4) (1977) 364–366. doi:10.1093/comjnl/20.4.364.
URL http://comjnl.oxfordjournals.org/content/20/4/364

[6] S. S. Skiena, The Algorithm Design Manual, Springer London, London, 2008.
URL http://link.springer.com/10.1007/978-1-84800-070-4

[7] D. Jerius, R. H. Donnelly, M. S. Tibbetts, R. J. Edgar, T. J. Gaetz, D. A. Schwartz, L. P. Van Speybroeck, P. Zhao, Orbital measurement and verification of the chandra x-ray observatory's PSF, in: Astronomical Telescopes and Instrumentation, 2000, pp. 17–27.
URL http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=899828

[8] A. Fruscione, J. C. McDowell, G. E. Allen, N. S. Brickhouse, D. J. Burke, J. E. Davis, N. Durham, M. Elvis, E. C. Galle, D. E. Harris, CIAO: chandra's data analysis system, in: Astronomical Telescopes and Instrumentation, International Society for Optics and Photonics, 2006.
URL http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1288644