Discovery of Brown Dwarfs mining the 2MASS and SDSS databases

Hendrik Heinl and Markus Demleitner

November 16, 2015

Abstract

Brown dwarfs are faint objects with a mass below 0.8 solar masses. In their stellar core, conditions don't suffice for hydrogene fusion and therefore the surface luminosity is very low. Even those brown dwarfs which are close to the solar system are faint objects, hard to detect and to identify as such. One method to identify brown dwarf candidates is using surveys as 2MASS and SDSS whoch provide colour filter bands. In this tutorial we will identify brown dwarf candidates by applying the method introduced by Zhang et.al. (2010). Therfore we use the VO tools TOPCAT and Aladin, and especially TAP-Services for 2MASS and SDSS. Eventually we crossmatch our results with SIMBAD to check our results for accuracy.
This tutorials is adressed to users who are already familiar with TOPCAT and now want to dive deeper into data mining with ADQL. It also gives a good overview on how you can combine different VO tools for your work.

Software: TOPCAT Version 4.3, Aladin Version 8.0, TAP, ADQL

1 Starting out

For this tutorial we need the VO tools TOPCAT and Aladin. If you do not have them already, this might be a good hint you are not familiar with either yet. In this case, a tutorial covering the basics of either VO tools might be more appropriate for you right now. In any case, you will find both clients here:

Aladin: `http://aladin.u-strasbg.fr/`

TOPCAT: `http://www.star.bris.ac.uk/~mbt/topcat/`

2 ADQL and TAP services

ADQL (Astronomical Data Query Language) is the language used in the IVOA to query remote TAP services. It is based on SQL and can be learned easily. We will use ADQL to post queries to different TAP services and especially to let the remote services do some of the work, so we receive only the data we are interested in and we do not have to download whole catalogs.

TAP (Table Access Protocol) services can be accessed by using ADQL. Therefor one needs a client software capable of understanding TAP. In this tutorial we use TOPCAT as a TAP client.

▷ **1** *Searching TAP services in the VO registry* – A good start point for our search for brown dwarfs is 2MASS, because it provides astrometry as well as magnitudes in infrared filter bands. To find an appropriat TAP service for 2MASS, we want to search in the VO registry. Therefor we open TOPCAT and go to VO →Table Access Protocol. In the TAP-Query window at Keywords we enter `2MASS`. In the line below, we check Description (you may need to adjust the window width to see all options). Then we click on Submit Query. In the section below now we can see a list of TAP services related to 2MASS. When searching for services, you may need to spent some time to find the one most appropriat for your needs. In our case we already have a good guess that the the **GAVO Data Center TAP service** will be the right choice. So in this list, we mark the line by clicking once. Then we either click on Use Service at the bottom or on the according tab at the top line.

Now we have to select the table we want to query. On the left side you see all tables which the GAVO data center TAP service provides. You can either scroll down until you find **twomass.data**, or instead use the search field above the list. Notice that you get a first glance of the meta data of this table in the field on the right. Here you find a good overview when you mine for data. We will have a close look at meta data a few steps ahead so we leave this aside for the moment.

▷ **2** *Submitting a first ADQL Query: cone search* – Within the VO the term of a **cone search** covers basically an area at the sky defined by a position and a radius in degrees around it. You may have performed cone searches already with TOPCAT, so here is a little different approach to give you an idea how ADQL works. Luckily TOPCAT comes with examples for the most common ADQL queries - or more precisely with those queries the TAP service mantainers assume to be common or at least useful.

In the lower window at ADQL Text click on Examples →Cone Selection. You see an example ADQL query appearing in the field:

```
SELECT
TOP 1000
*
FROM twomass.data
WHERE
  1=CONTAINS(POINT('ICRS', raj2000, dej2000),
             CIRCLE('ICRS', 189.2, 62.21, 0.05))
```

Now at the first glance that may look confusing, so let's go through it step by step. Each ADQL query starts with `SELECT` followed by specifications of "what" to select, what to do with the selected records and how to return the results. `TOP 1000` means the first 1000 data records will be returned, which match the whole query. With `*` we select all columns of matching data records. If you just wanted certain columns from a table to be returned, you could specify that here. We will see in a later step how to do this. `FROM twomass.data` specifies the table of the TAP service we want to query. The next lines are the query conditions. In our example we use the ADQL built in functions that define a cone search. The `WHERE` clause extracts those data records that match. In our case these criteria shall be sources in a cone around a certain position. A little confusing may be the `1=CONTAINS`. This is due to the boolean result returned by the function `CONTAINS`. A result of `1` means "true" whereas `0` means "false". `POINT` expresses a point depending on the coordinate frame (soon to be depricated), the right ascension and the declination in degrees. Our query takes the coordinates from the table using the columns `raj2000` and `dej2000`. Analogously `CIRCLE` expresses a cone in space, except that we need to set an angle in degrees as last number.

Clicking OK will start the cone search and in between a few seconds we retrieve the first 1000 data records. Return to the TOPCAT mainwindow and play around with this data, if you like to. As soon as you feel ready, proceed.

▷ **3** *Metadata* – As mentioned above you can have a glance at the metadata of a TAP-service in TOPCATs TAP window. For our hunt of brown dwarfs we basically need colour indexes and therefore search the colour filter magnitudes. In the TAP window we click on Table →Table columns. Here we have an overview of the table metadata and get the names of the columns which we need later to specify our ADQL query. We are especially interested in H,J and K magnitudes, which we see are named hmag, jmag and kmag. Knowing the column names, we now can modify the ADQL query accordingly.

▷ **4** *Searching 2MASS with ADQL –* Now we want to search 2MASS for those objects, which are good candidates for brown dwarfs. Because Brown dwarfs are faint objects, we estimate they have a J magnitude of 15.3 or higher. We also assume brown dwarfs are red sources. So we want to search for sources with an index of

$$jmag - kmag > 0.8.$$

Running the query over the whole 2MASS catalog would take a long time so we don't search over the whole sky, but limit the search to a 2 degrees cone instead. Finally we change the search cone coordinates.

All this we can do with ADQL by modifying the query as following:

```
SELECT *
  FROM twomass.data
    WHERE 1=CONTAINS(
    POINT('ICRS', raj2000, dej2000),
    CIRCLE ('ICRS', 127.0000, 1.2000, 2.0))
      AND jmag > 15.3
      AND jmag-kmag > 0.8
```

Before we submit the query, we have to set a maximum so we receive all the matching data records. To do so look at **Service Capabilities** →**Max Rows** and select `max` in the drop down menu. We sent the query by clicking on OK and the result should be returned in a few seconds.

▷ **5** *Crossmatching with SDSS –* With the j,h and k filter magnitudes from 2MASS we found first candidates for brown dwarfs. For further data reduction we now apply criteria derived from Zhang et al. (2010). Therefor we need i,r and z filters which we take from SDSS. In the Topcat TAP window we now enter sdssdr7, again check Description and submit the query. Again our choice due to performance issues is the GAVO datacenter TAP service. You notice, that we have the same choices to query tables as when we searched for 2MASSS. This is because we use the same TAP service as before and therfore we have to specify the tables we want to query with `FROM`.

We could now do a similar search as for 2MASS and then compare the two tables locally. But we rather use a more elegant method and let the GAVO TAP service do some work for us remotely. We will use the possibility tap_upload function to upload local tables to a TAP service for more complex queries. In the same process we use the `JOIN... ON` command in ADQL to merge the uploaded data with the query results. This time we use the `CIRCLE` function to find matches for the data we received from 2MASS an therefore keep the cone very small at 1”. Finally we add `TOP 20000` after `SELECT` because our upload contains less rows and we expect to even reduce this data. This is the full query:

```
SELECT TOP 20000
  tm.* , sdss.*
  FROM sdssdr7.sources as sdss
  JOIN tap_upload.t2 AS tm
    ON 1=CONTAINS(
      POINT('ICRS', sdss.ra, sdss.dec),
      CIRCLE('ICRS', tm.raj2000, tm.dej2000, 1./3600.))
        AND sdss.i - sdss.z BETWEEN 1.5 AND 2
        AND sdss.r - sdss.i BETWEEN 1.5 AND 4.5
```

See that within `JOIN ... ON` we upload our table data from 2MASS. Take care that you use the right table number at `tap_upload.t2`. There are many reasons why this number might be different than 2. Just replace it with the according number of your table in TOPCAT.

After submitting the query we receive a table of hopefully good candidates for brown dwarfs, so let’s take a look at the data. At first glance we recognise that we do have some identical data records in raj2000 and dej2000 columns. This is due to the TAP search which returned all matches around a source from the 2MASS data, and not the single best match. As we scroll down the records, we see that for all 2MASS positions at least 2 records in SDSS matched our query. How come that as we performed a cone search around 1” around our given positions? It may be helpful to take a look at the observed images and thankfully the the VO provides us with the perfect tool for that: Aladin.

▷ **6** *Resolving suspicious results with Aladin and TOPCAT* – In this step we want to see the images in 2MASS and SDSS of our brown dwarf candidates, especially we are interested to find out, why we have several matches in SDSS for each of the candidates identified in 2MASS. For this we use the SAMP protocol to send data from TOPCAT to Aladin. We start Aladin and select the 2MASS catalog. Then we go back to the TOPCAT main window and check the table with our candidates from SDSS. We then click in Current Table Properties →Activation Action →Transmit Coordinates. Thus we activate TOPCAT to send data to Aladin. Aladin will automatically ”listen” and wait for data from the SAMP hub, that is running in the background. Still in Topcat we open the table data and click on any table row. In Aladin we can see the position change. If we compare the images from our doubled data records, we see that the sources point to the same object in the catalog. We can switch between 2MASS and SDSS images in Aladin and see the same effect. This is simply because we received all matching datarecords from our crossmatch with SDSS and in the SDSS catalog the different observations to a single object are not merged. So we have to do this on our own.

We solve this in TOPCAT. In the main window we mark the table and click on Joins →Internal match. In the new window at Table we chose our current table and below we check Eliminate All But First of Each Group. Of course for real science this would be inappropriate, but for our purpose it works well. Now we reduced our brown dwarfs candidates to 11 and it's about time to check the accuracy of this kind of these methods.

▷ **7** *Multicone Search in SIMBAD* – Now we want to know how many of our candidates are already confirmed brown dwarfs. A good location to check for this of course is SIMBAD, but instead of accessing it via the web interface, we use the Multi Cone Search in TOPCAT.

In TOPCAT's main window go to Join →Multicone. At **Keywords** enter `SIMBAD` and then click on Find Services. After a few seconds you will see three results. Select the one with **Shor Name** SIMBAD. Now in the lowe part of the window we need to select the **Input Table** according to the result of our internal match from the previous step. TOPCAT will complete the positions. As **Search Radius column** 1" will suffice. The other configurations can stay as they are. Click on Go and recieve the Matches from SIMBAD.

If you take a look at the data, you will see in the second last column that these 5 data records are indeed confirmed brown dwarfs.

2.1 More to think about

As we saw in the last step, this method seems to be accurate to identify brown dwarfs - but what about completeness ? Think about how to check our results for completeness with SIMBAD and which functions in ADQL you could use for this.

3 References

Fernique, P. http://aladin.u-strasbg.fr/java/AladinManual6.pdf

Ortiz, I., Lusted, J., Dowler, P. et al. http://www.ivoa.net/documents/REC/ADQL/ADQL-20081030.pdf

Taylor, M. http://www.star.bris.ac.uk/~mbt/topcat/#docs

Zhang, Z. H., Pinfield, D. J., Day-Jones, A. C., et al. 2010, MNRAS, 404, 1817, 2010MNRAS.404.1