

A basic cross-match tool

T. Boch, S. Derrière [CDS]

May '04 27



T. Boch – VOTable2 Session
IVOA Interop meeting Cambridge MA



Motivations

- Need for a positional cross-match for the '04 AVO demo
- Being able to merge information coming from multiple catalogues
- Not intended for very large catalogues, but suitable for medium-sized lists (few 10,000)

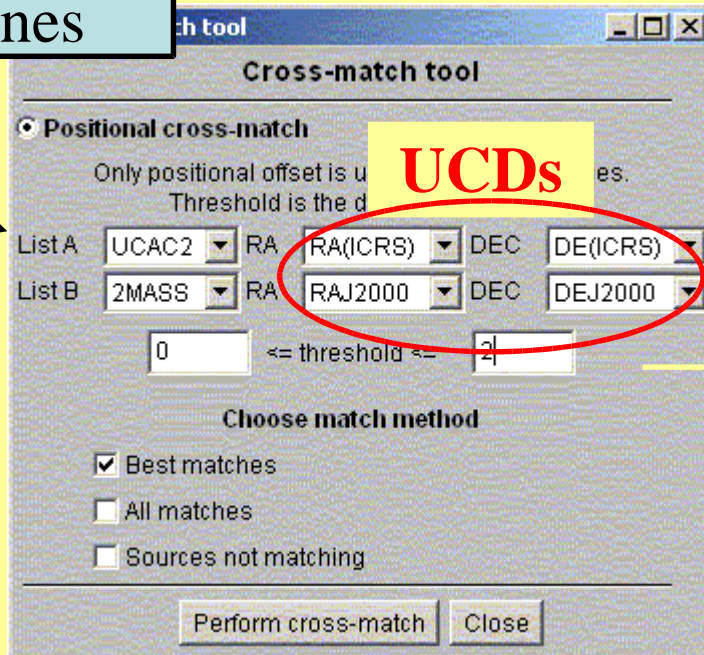
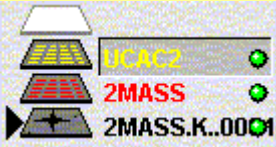


Architecture

- For the AVO demo: remote service (HTTP post) with VOTable as input parameter, and VOTable as output



2 catalogue planes



AVO
prototype

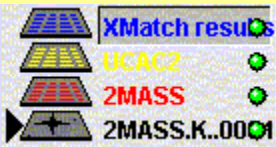
Data exchange
in VOTable

VOTable (2 tables)
+ thresholds

Remote
Cross-match
Service at CDS

VOTable
(merged data + distance)

New catalogue plane



Architecture

- For the AVO demo: remote service (HTTP post) with VOTable as input parameter, and VOTable as output
- In next release of Aladin: built-in Java class
 - Aladin uses SAVOT parser to ingest VOTable catalogue files
 - Cross-match computed internally
 - Ability to export result of the cross-match as a VOTable file

Demo

Cross-match service

Cross-match service

Choose planes for x-match

List A: 2MASS

List B: UCAC2

Positional cross-match

Only positional offset is used to find the matches.

2MASS RA: RAJ2000 DEC: DEJ2000

UCAC2 RA: RA(ICRS) DEC: DE(ICRS)

Threshold is the distance in arcsec

<= threshold <=

Choose match method

Best matches

All matches

Sources not matching

Cross-ID

The join is made on a common field

2MASS XID Field: RAJ2000

UCAC2 XID Field: 2UCAC

Perform cross-match Close

Input parameters

- List A : reference objects, for which we search counterparts
- List B, in which counterparts are to be found
- d_{\min} and d_{\max} : objects are considered matching if separated by a distance between d_{\min} and d_{\max}
- Best match : only the best match (with smallest distance) is returned
- All matches : all matching sources from list B returned
- Non-matches : returns sources without counterpart in list B

Cross-match algorithm

- Sort lists by declination
- Foreach source from A, scan relevant subset of B (within d_{\max}) and compute distance
- Return either
 - All matches
 - Best match
 - Objects from A with no match

Output

- All FIELDS from list A
- All FIELDS from list B
- Distance (in arcsec) between 2 matching sources
- FIELDS are suffixed by `_tab1` (resp. `_tab2`) to ensure column name unicity

Possible improvements

- Take into account positional errors
(will use the new <GROUP > feature to find appropriate error columns)
- Extend the algorithm to compute a score, function of distance on the sky + other parameters
- In the output, group similar quantities by using UCDs to ease comparison
- Make it independent from Aladin, and distribute it as a cross-match package (input : 2 VOTable streams, output : 1 VOTable stream)