IAU Symposium 325

Astroinformatics 2016

Sorrento (Italy), October 20-24, 2016

# C$^3$: A Command-line Catalogue Cross-matching tool for modern astrophysical survey data

**Giuseppe Riccio**, Massimo Brescia, Stefano Cavuoti, Amata Mercurio

*INAF – Astronomical Observatory of Capodimonte, Napoli*

Anna Maria Di Giorgio, Sergio Molinari

*INAF – Istituto di Astrofisica e Planetologia Spaziali, Roma*

# The data-driven science era

The size of data has rapidly increased, reaching in many cases dimensions overcoming the human possibility to be handled in an efficient and comprehensible way.

In astronomy the data volumes, from the ongoing and next generation of multi-band and multi-epoch surveys, are expected to be so large that the ability of the astronomers to analyze, cross-correlate and extract knowledge from such data will represent the most challenge for scientists.

**Some example:**

- ESA Euclid space mission: ~800 Gbit/day over at least 6 years, collecting a minimum amount of ~200 TB of data;
- Pan-STARRS: more than 100 TB of data;
- GAIA space mission: a Milky Way 3D map, by collecting ~1 PB of data in 5 years;
- Large Synoptic Survey Telescope (LSST): ~30 TB/night of imaging data for 10 years and PB/year of radio data products;
- KiDS (Kilo-Degree Survey), DES (Dark Energy Survey), Herschel-ATLAS, Hi-GAL, E-ELT...

**To deal with such amount of information, also the data analysis techniques and facilities must quickly evolve.**

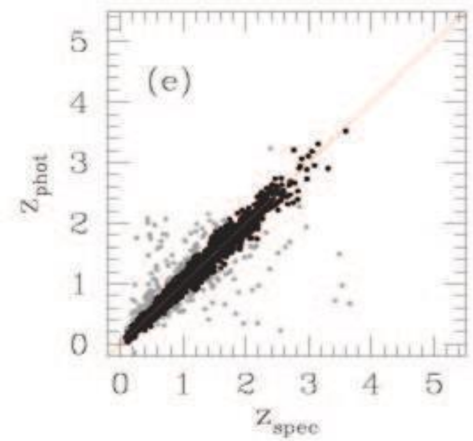
**Catalogues cross-match, fundamental prerequisite for combining multi-band data, is particularly sensible to the growing of the datasets dimensions**

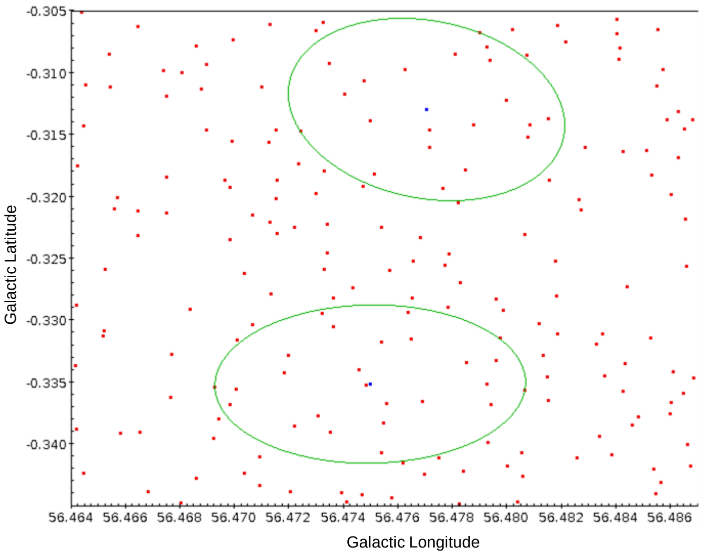
# Astronomical cross-matching

Data analysis techniques and technologies are rapidly evolving (data mining and machine learning algorithms, large scale distributed DBMS, parallel processing frameworks), but **Cross-matching** is still one of the core steps of any standard modern pipeline for data reduction/calibration/analysis

**Some Examples:**

Photometric redshifts evaluation for Quasars through *machine learning* algorithms (*Brescia et al. 2013*): dataset obtained by merging **4 different surveys** (SDSS, GALEX, UKIDSS and WISE)





Evolutionary classification tool for ViaLactea Project, based on *data mining* and *machine learning (Merello et al. in prep.):* it will catalogue a HI-GAL "clump" in terms of the evolutionary stage of the stellar sources. Data obtained by combining information from **UKIDSS, GLIMPSE** and **MIPSGAL surveys**.

# Cross-matching Tools

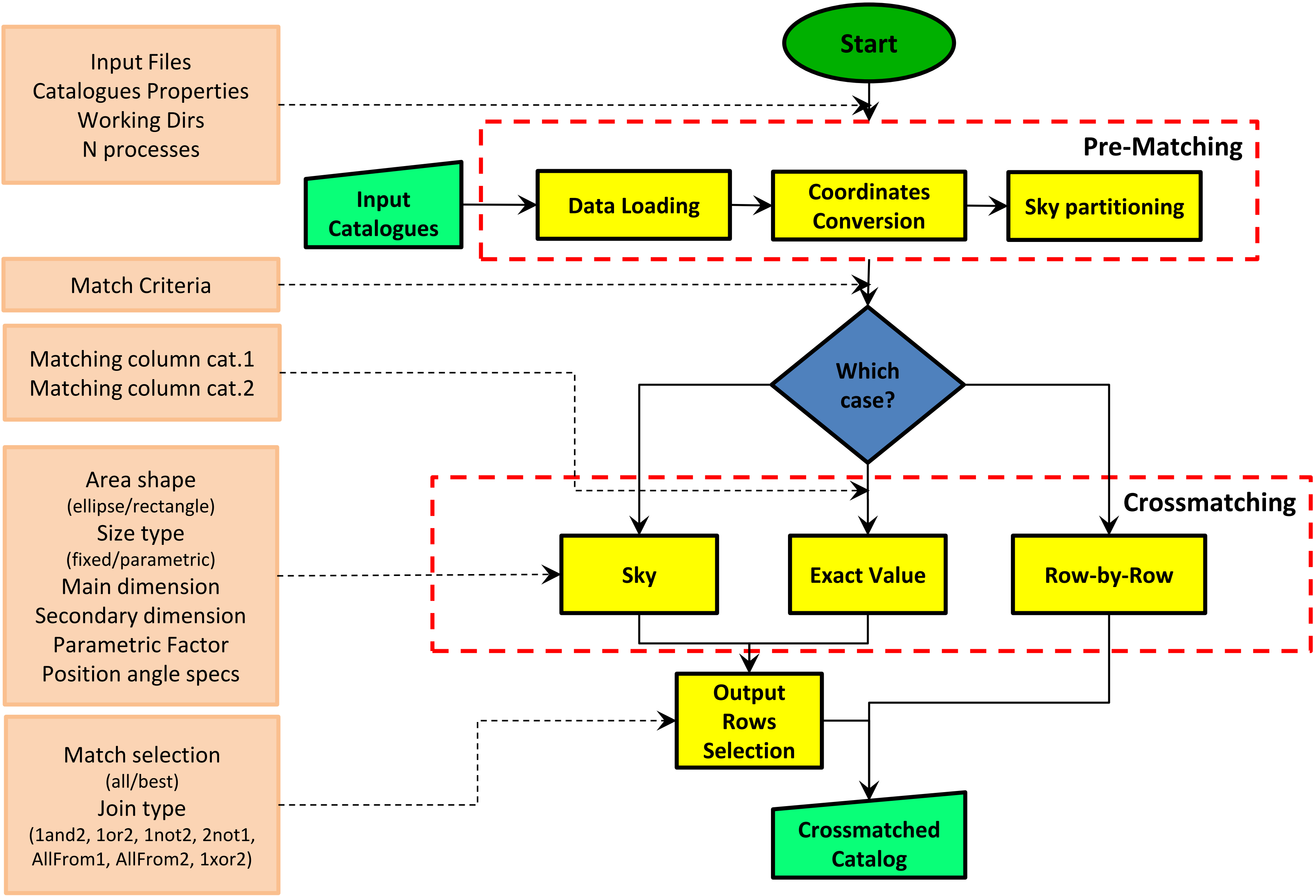
| Cross-matching strategy | PRO | CONS |
|---|---|---|
| **Web application (CDS-Xmatch, Arches)** | ✓ Portal to query and cross-match large datasets<br>✓ **Intuitive user interfaces**<br>✓ Black boxing<br>✓ Custom script languages | ● **No integration into data reduction/analysis pipelines**<br>● Tool scalability<br>● Limited choice of parameters |
| **Stand-alone (STILTS)** | ✓ **Command-line tools**<br>✓ Ready-for-use APIs<br>✓ **Easy application in a data reduction/analysis pipeline**<br>✓ Usage in distributed computing environments | ● Often platform-dependent<br>● Performance limited by hosting machine<br>● **Configuration not always easy** |
| **GUI (Topcat)** | ✓ **Intuitive user interfaces**<br>✓ Black boxing | ● **No integration into data reduction/analysis pipelines**<br>● Limited choice of parameters<br>● Local execution (Java heap memory limitations) |

# C$^3$: Command-Line Catalogue Cross-match

*C$^3$ is a command-line software, designed and developed to perform general cross-matching among astrophysical catalogues, trying to cope with the needs of new generations of astronomers, who must deal with very large datasets produced by independent surveys, to be combined together to extract new information and to increase knowledge about astronomical objects.*

- **Command-line tool**: as stand-alone program or integrated within complex pipelines;
- **Python Compatibility**: with ver. 2.7.x and 3.4.x;
- **Cross-platform**: tested on Ubuntu 14.04, Windows 7/10, Mac OS and Fedora;
- **Multi-process**: cross-matching process exploiting the multi-core parallel processing paradigm (nowadays the standard setup of desktop/laptop hardware);
- **Sky partitioning**: simple sky partitioning system to reduce computing time;
- **User-friendly**: easy to be configured and used.

# C$^3$ Architecture



Input Files
Catalogues Properties
Working Dirs
N processes

Match Criteria

Matching column cat.1
Matching column cat.2

Area shape
(ellipse/rectangle)
Size type
(fixed/parametric)
Main dimension
Secondary dimension
Parametric Factor
Position angle specs

Match selection
(all/best)
Join type
(1and2, 1or2, 1not2, 2not1, AllFrom1, AllFrom2, 1xor2)

Start

Pre-Matching

Input Catalogues

Data Loading

Coordinates Conversion

Sky partitioning

Which case?

Crossmatching

Sky

Exact Value

Row-by-Row

Output Rows Selection

Crossmatched Catalog

# C$^3$ Pipeline: Step 1

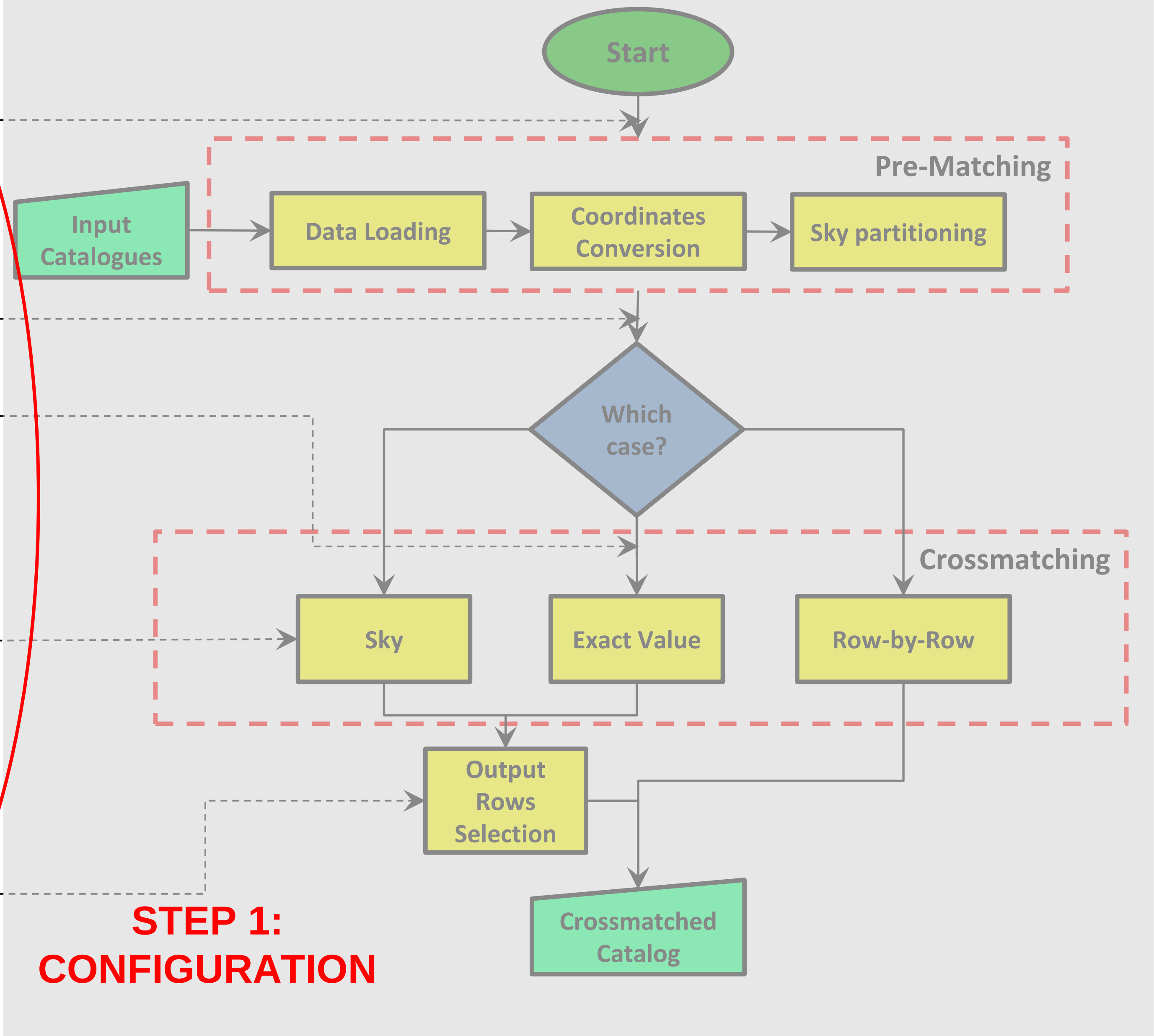
Input Files
Catalogues Properties
Working Dirs
N processes

Match Criteria

Matching column cat.1
Matching column cat.2

Area shape
(ellipse/rectangle)
Size type
(fixed/parametric)
Main dimension
Secondary dimension
Parametric Factor
Position angle specs

Match selection
(all/best)
Join type
(1and2, 1or2, 1not2, 2not1, AllFrom1, AllFrom2, 1xor2)



**STEP 1: CONFIGURATION**

# C$^3$ Pipeline: Configuration

**I/O Files**

- Input Catalogues and their format (*CSV, FITS, ASCII, VOTable*);
- Output file (*CSV, FITS, ASCII, VOTable*), Log file.

**Match Criteria - ** matching algorithm types:

- **Sky**, the cross-match is done within sky areas defined by the catalogue parameters;
- **Exact Value**, objects matched in case of same value for a pair of columns;
- **Row-by-Row**, match done on a same row-ID of the two catalogues.

**Functional Case parameters - ** parameters used to perform the match:

- Shape and dimensions of matching area for "*Sky*";
- Matching columns for "*Exact Value*";
- No additional parameters for "*Row-by-Row*".

**Catalogues Properties**

- Coordinate system (*icrs, fk4, fk5, galactic*);
- Coordinate units (*degrees, radians, sexagesimal*).

**Thread Properties - ** number of parallel processes.

**Output Rows - ** matches to be stored in the output file:

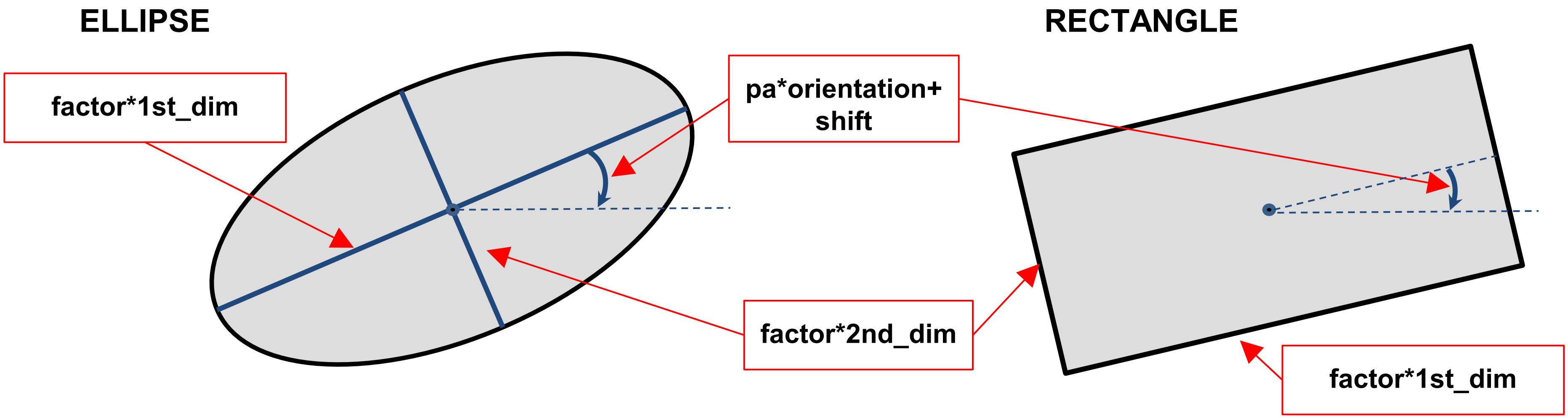
- Match selection, *all* or *best* matches;
- Join type, indicating which rows to include.

**All required parameters set by user through a simple configuration file**

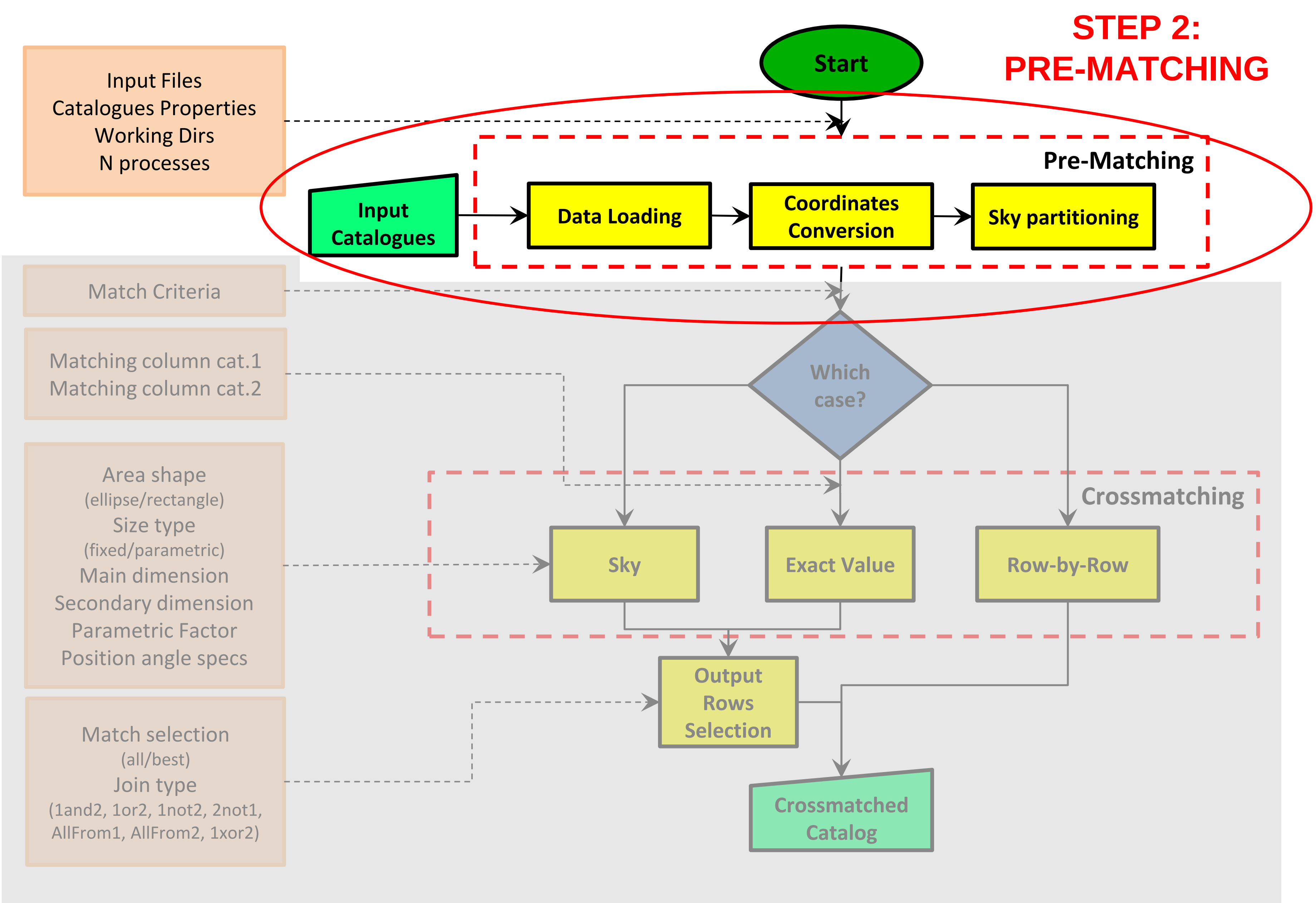
# C$^3$ Pipeline: *Sky* use case

"Sky" functional case requires additional parameters to characterize the shape and the dimensions of the matching areas

- **Area shape -** elliptical or rectangular (circular is a special elliptical case);
- **Size type -** valid entries are *fixed* or *parametric*;
- **Matching area dimensions -** semi-axes of the elliptical area or width and height of the rectangular area;
- **Parametric factor -** only for parametric ``Size type'', multiplicative factor for  dimensions;
- **Pa column/value -** value or column name/ID of the position angle;
- **Pa settings -** parameters indicating the orientation (clockwise/counterclockwise) and a shift angle (in degrees) to be taken into account for matching catalogues.



# C$^3$ Pipeline: Step 2

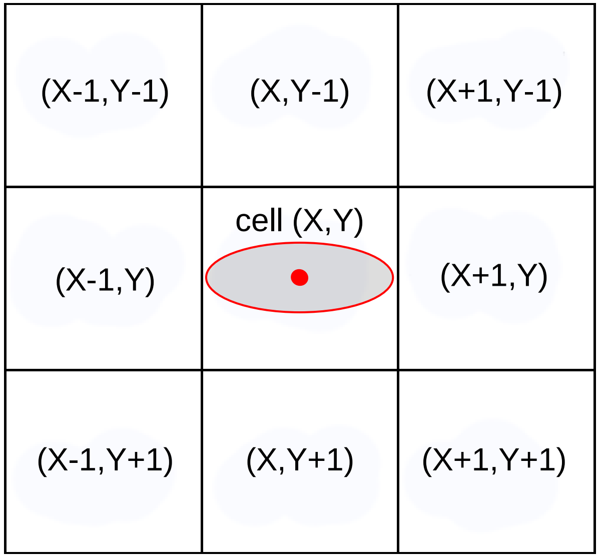


# C$^3$ Pipeline: Pre-Matching

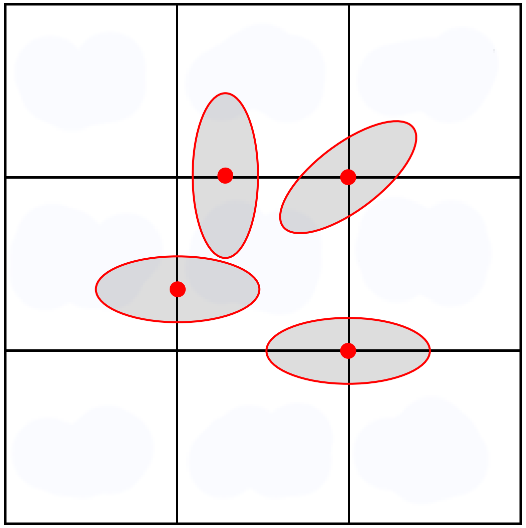
During the pre-matching phase, C$^3$ prepares data for the analysis. In particular, after configuration parameters checking and catalogues loading, it organizes data to exploit parallel computing in order to reduce the computing time.

**Catalogue 1:** uniform split in a number of groups equal to the defined number of parallel processes → **1 group for 1 process** (also for “Exact Value”);

**Catalogue 2 - Sky partitioning:** the sky is partitioned in square *cells* whose size is defined by the maximum dimension that the matching regions can assume, with a minimum value to avoid the cell generation redundancy.

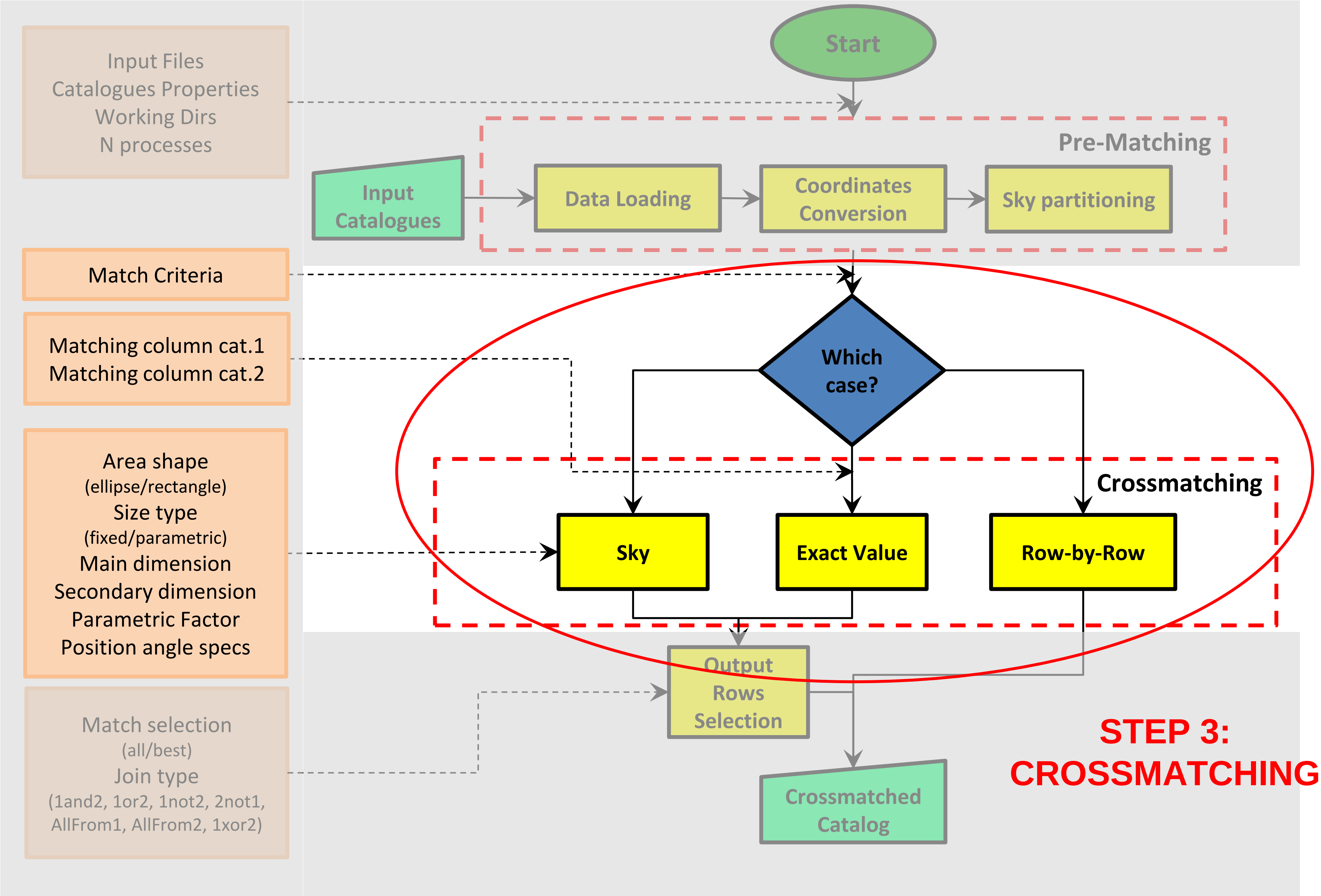


| | | |
|---|---|---|
| (X-1,Y-1) | (X,Y-1) | (X+1,Y-1) |
| (X-1,Y) | cell (X,Y) | (X+1,Y) |
| (X-1,Y+1) | (X,Y+1) | (X+1,Y+1) |



Once the partitioning is defined, each object of the 2nd catalogue is assigned to one cell, according to its coordinates. Having defined the cells, the boundaries of an elliptical region can fall at maximum in the eight cells surrounding the one including the object. **This prevents the well-known block-edge problem**.

# C$^3$ Pipeline: Step 3

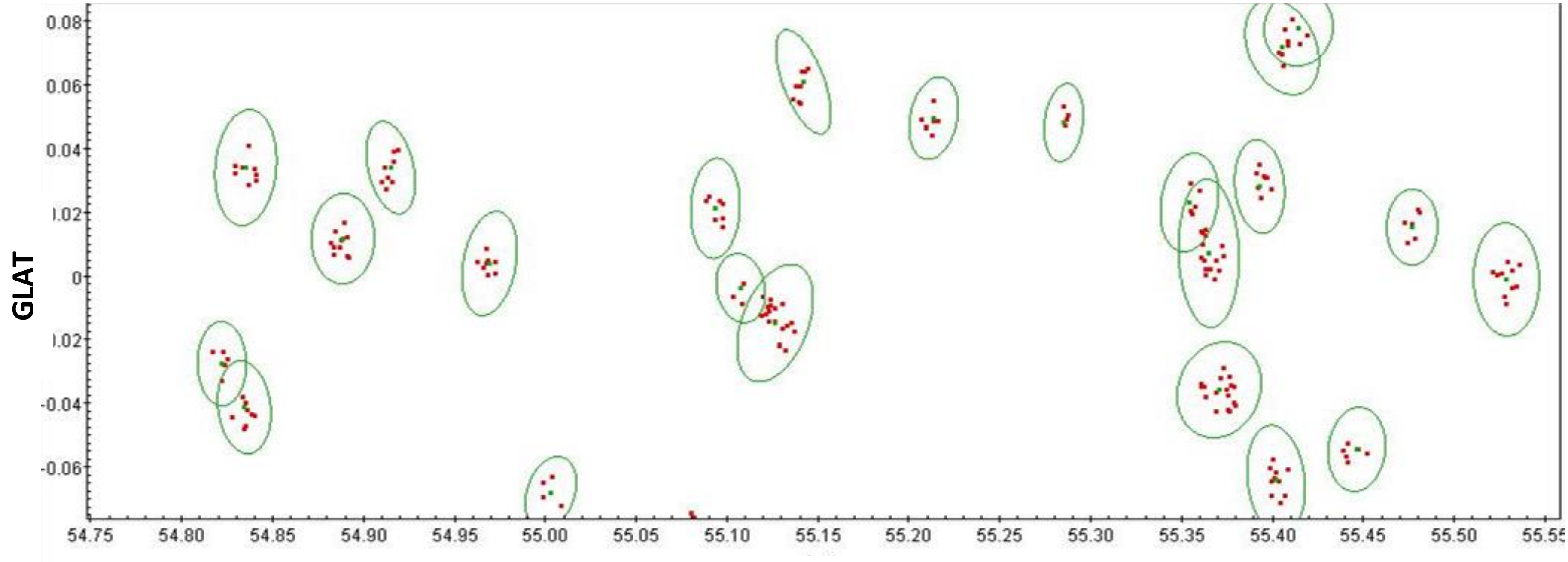


STEP 3: CROSSMATCHING

# C³ Pipeline: Cross-matching

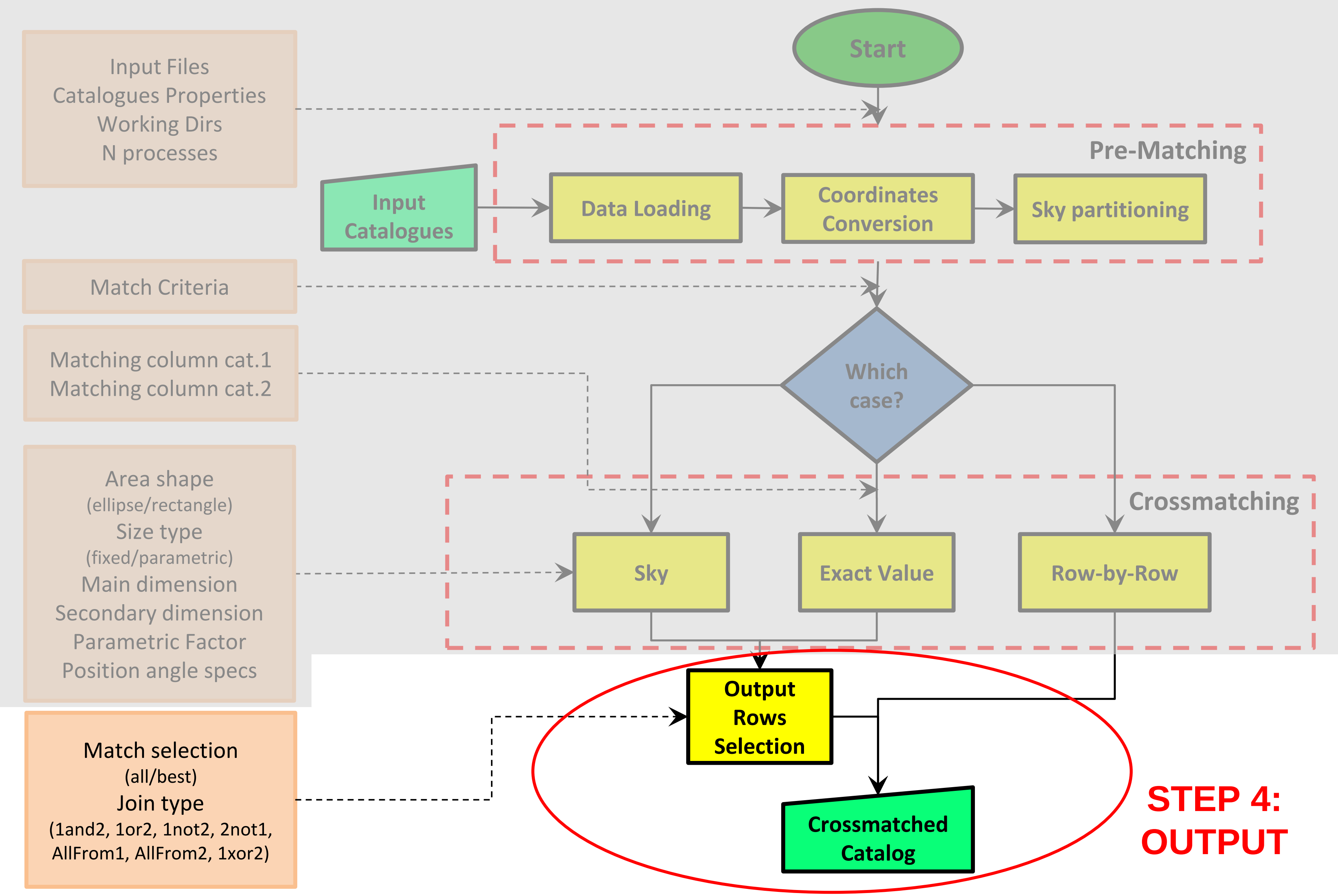
The positional cross-match is based on a simple concept:

- Definition of an elliptical/rectangular region centered on each object of catalogue 1, whose dimensions are limited by configuration parameters (fixed or cell value);
- Search for catalogue 2 sources within such region, by comparing their distance from the central object (in case of elliptical cross-match, the analytical equation of the ellipse is used).



**Green ellipses**: Matching Areas
**Green dots**: 1st Catalogue Objects
**Red dots**: 2nd Catalogue Objects

# $C^3$ Pipeline: Step 4

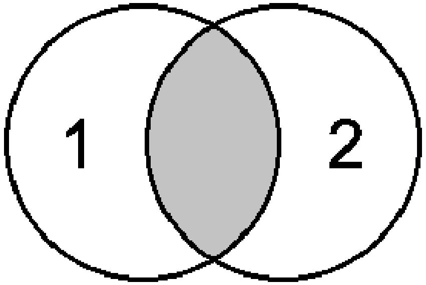


STEP 4:
OUTPUT

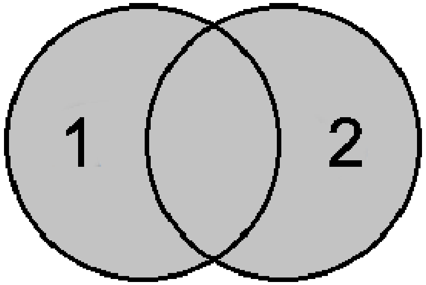
# C$^3$ Pipeline: Output Creation

C$^3$ provides a file (CSV, FITS, ASCII or VOTable) containing the results of the cross-match. For ``Exact value'' and ``Sky'' functional cases, the user can define the conditions to be satisfied by the matching output.

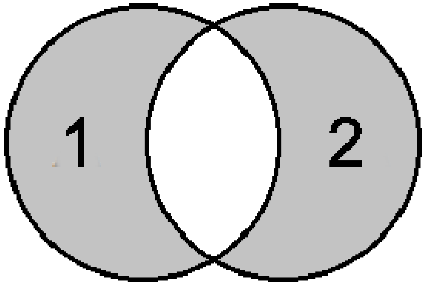
- **Match selection**, *all* matches or only the *best* matches;
- **Join type** in one of the combinations shown below.



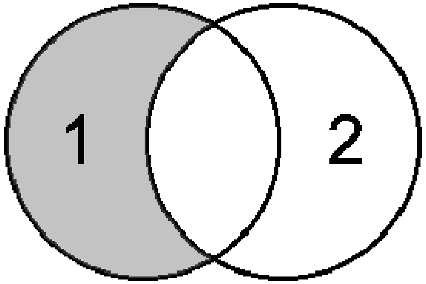
**1 AND 2**



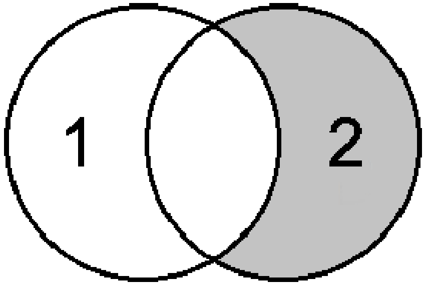
**1 OR 2**



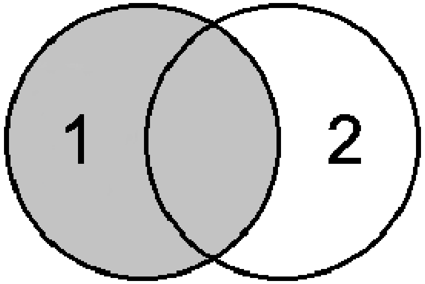
**1 XOR 2**



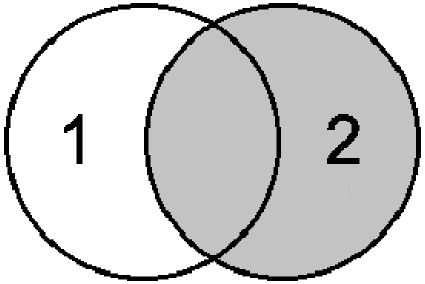
**1 NOT 2**



**2 NOT 1**

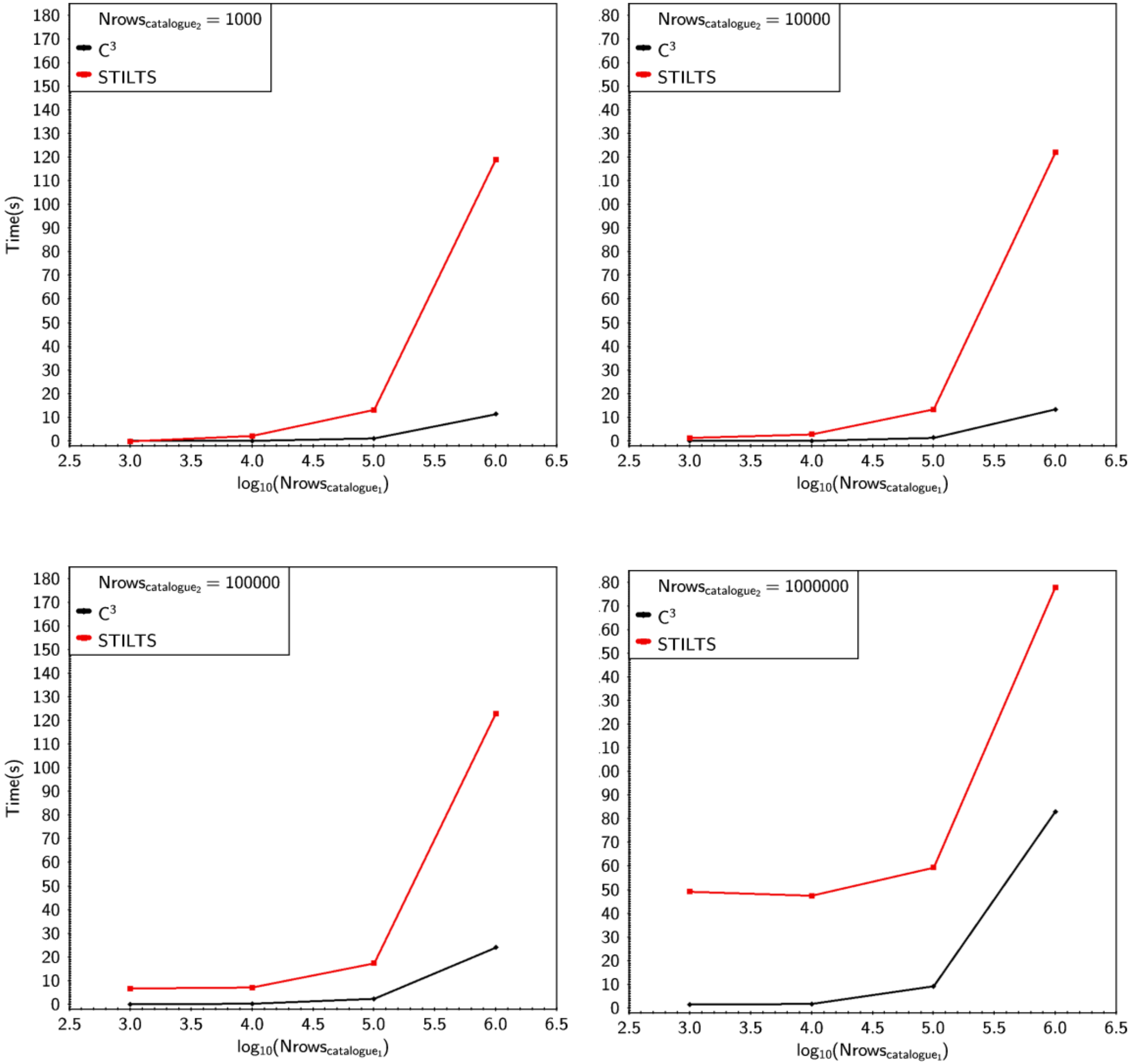


**ALL FROM 1**



**ALL FROM 1**

# $C^3$ vs STILTS



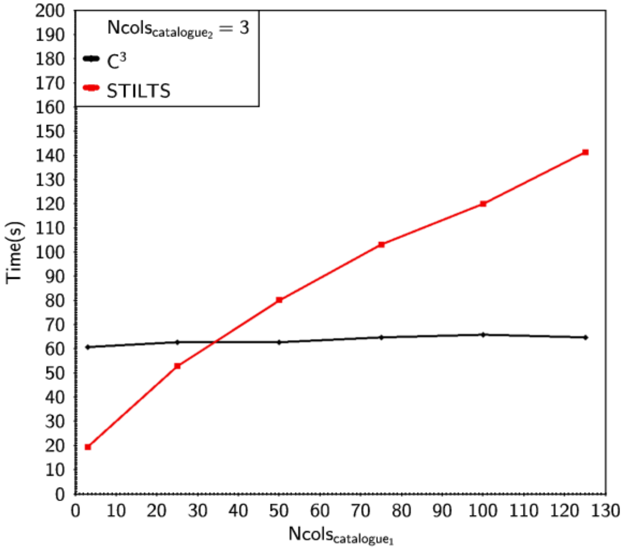
**Comparison between $C^3$ Cross-matching phase and STILTS by varying the number of catalogue objects (rows).**

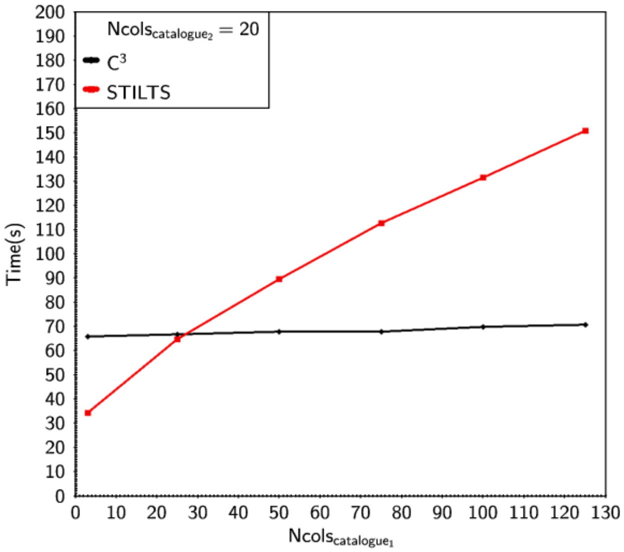
$C^3$ always faster than STILTS (in particular when the size of datasets increases).

Latency time due to Pre-matching and Output creation phases not yet been considered (serialized in the current release).

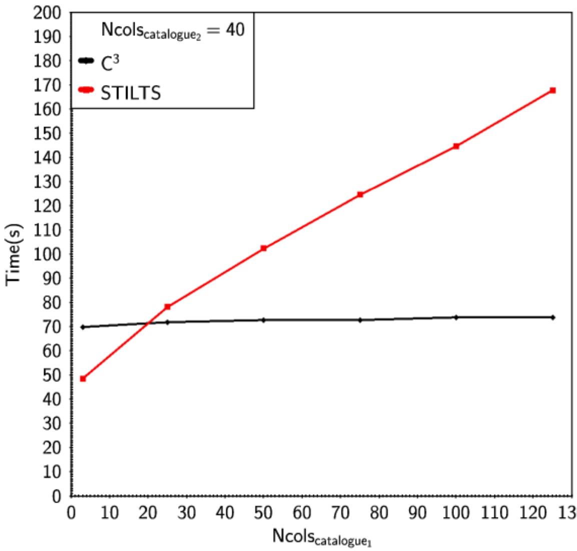
**They will be optimized in the next releases.**

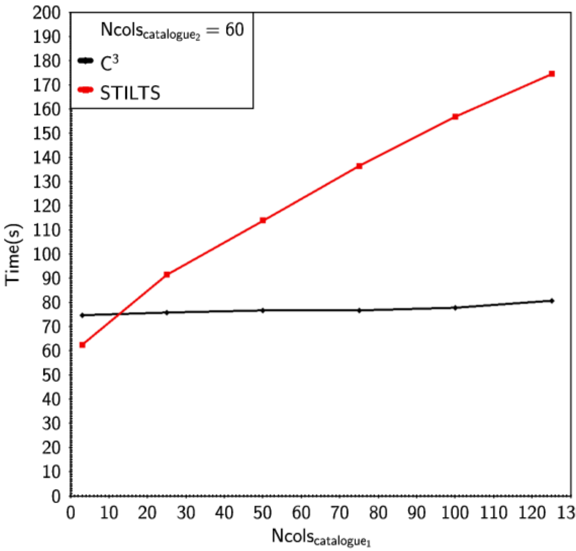
# $C^3$ vs STILTS

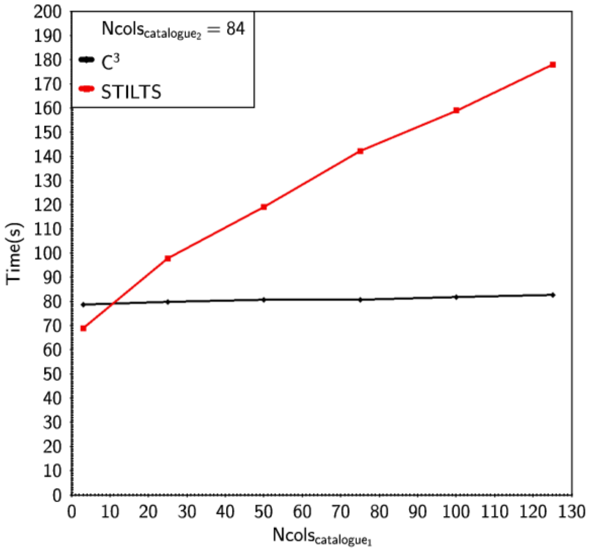




**Comparison between $C^3$ Cross-matching phase and STILTS by varying the number of columns of the catalogues.**







Increasing the number of columns of input datasets, $C^3$ is approaching STILTS computational time. When the number of columns is sufficiently high, $C^3$ becomes faster.

# C$^3$ Summary

- C$^3$ is a **multi-platform** tool designed to efficiently **cross-match** massive catalogues, reaching high-performance capabilities through a **multi-core parallel processing paradigm**;
- The tool has been conceived to be a **stand-alone command-line application** or an **integrated** package within any generic data reduction/analysis pipeline;
- It provides the **maximum flexibility** to the end user, in terms of parameter configuration, coordinates and cross-matching types;
- It is **user-friendly**, easy to be configured and used;
- Preliminary tests show its scientific reliability in terms of cross-matching quality.

**TODO in next releases:**

- **Increasing performance:**
  - by optimizing Pre-matching and Output creation steps;
  - by refining the Sky partitioning algorithm;
- **Providing additional functionalities**:
  - new matching algorithms;
  - sub-regions definition;
  - new coordinate systems
  - User demanding...

C$^3$ **Rel. 1.0** is available @:

**http:/dame.dsf.unina.it/c3.html**

THANKS!