



COMPUTER SCIENCE AND STATISTICS  
ENGINEER POLYTECH LILLE

## Documentation

Data Extraction from Publications and Statistics of  
the HAL Database

Anas Nay

---

School name and address:

POLYTECH Lille  
Boulevard Paul Langevin  
59655, VILLENEUVE D'ASCQ  
CEDEX  
03-28-76-73-60

School supervisor: Frédéric  
Hoogstoel

Company name and address:

Centre de Recherche en  
Informatique, Signal et  
Automatique de Lille (CRISTAL)  
Université de Lille, Sciences et  
technologies, Batiment Esprit,  
59655 Villeneuve-d'Ascq

Company supervisor: Mihaly  
Petreczky

Academic Year 2024-2025

# Contents

<b>1</b>	<b>Project Context and Objectives</b>	<b>4</b>
<b>2</b>	<b>Tool Usage Modes</b>	<b>5</b>
2.1	Interactive Graphical Interface ( <code>app.py</code> file)	5
2.2	Command-Line Interface ( <code>main.py</code> file)	5
<b>3</b>	<b>Interactive Application Overview</b>	<b>6</b>
3.1	Launching the Application	6
3.2	General Architecture	6
3.3	Data Extraction Module	7
3.3.1	Step 1: HAL Identifier Extraction	7
3.3.2	Verification and Correction of Extracted Identifiers	9
3.3.3	Step 2: Publication Metadata Extraction	11
3.4	Data Analysis Module	14
3.4.1	Automatic Generation of Graphical Visualizations	14
3.4.2	Interactive Dashboard Consultation	16
3.4.3	Compiled Report Production	17
3.4.4	Thematic Keyword Analysis	18
3.5	Matching Sensitivity Configuration	20
<b>4</b>	<b>Command-Line Interface Overview</b>	<b>22</b>
4.1	Main Functionalities	22
4.2	Input File Selection	22
4.3	Filtering Options	22
4.4	Sensitivity Configuration	23
4.5	Automatic Content Generation	23
4.6	Utility Commands	23
4.7	Usage Examples	23
4.7.1	Simple Extraction	23
4.7.2	Extraction with Filters	23
4.7.3	Extraction with Custom Sensitivity	23
4.7.4	Extraction with Automatic Generation	24
4.7.5	Information Commands	24
4.8	Progress Tracking and Results	24
4.9	Pre-extraction Summary	24
4.10	Output File Organization	24
<b>5</b>	<b>Duplicate and Homonym Detection Module</b>	<b>25</b>
5.1	Operating Principle	25
5.2	Module Access from Graphical Interface	25
5.3	Analysis Configuration and Launch	26
5.3.1	Input File Selection	26
5.3.2	Optional Laboratory File	26
5.4	Analysis Interface and Results	26
5.4.1	Summary Tab	26
5.4.2	Detailed Results Tabs	27

5.5	Supported Detection Types . . . . .	28
5.6	Automatic Data Processing . . . . .	28
5.6.1	Available Processing Options . . . . .	28
5.6.2	Processing Results . . . . .	28
5.7	Results Exportation . . . . .	29
5.7.1	Generated Files . . . . .	29
5.8	Command-Line Usage . . . . .	30
5.8.1	Command-Line Session Example . . . . .	30
5.9	Limitations and Considerations . . . . .	31
5.9.1	HAL API Dependency . . . . .	31
5.9.2	Special Cases . . . . .	31
5.10	Conclusion . . . . .	31
<b>6</b>	<b>Appendix</b>	<b>32</b>
6.1	Expected CSV File for Extraction . . . . .	32
6.2	Description of the Obtained CSV File . . . . .	32

# Foreword

This documentation is intended for users with basic computer skills, including the ability to execute commands in a terminal and manage Python file execution. Familiarity with version control systems, particularly GitHub, is also recommended.

Before using the scripts in this project, it is **imperative** to properly configure your working environment. The scripts, developed in Python, require the prior installation of specific dependencies that play an essential role in data processing, report generation, and graph visualization.

## Installation and Configuration

### 1. Repository Cloning

Open a terminal and execute the following commands:

```
git clone https://github.com/anasnay11/PROJET_HAL_.git
cd PROJET_HAL_
```

### 2. Virtual Environment Creation

Create and activate a Python virtual environment:

```
python -m venv venv
```

Then activate it according to your operating system:

- **Linux/macOS:** `source venv/bin/activate`
- **Windows:** `venv\Scripts\activate`

### 3. Dependencies Installation

Once the virtual environment is activated, install the required packages:

```
pip install -r requirements.txt
```

Verify that all packages install correctly before proceeding to script execution.

## Important Points to Remember

1. The main files `app.py` and `main.py` are located in the `python code` subfolder. Navigate to this directory before executing them.
2. The input CSV file must contain the columns `'nom'` and `'prenom'` to be usable by the application.
3. For optimal visualization of the interface screenshots, maximize your application window to faithfully reproduce the images presented in this documentation.

# 1 Project Context and Objectives

This project aims to develop an interactive and intuitive tool for the extraction, analysis, and visualization of scientific data from the HAL platform (Hyper Articles en Ligne). HAL constitutes the French national open archive that centralizes and disseminates scientific publications produced by research institutions, laboratories, and researchers across the territory.

Faced with the growing volumes of scientific data and the need to efficiently analyze research output, this tool offers a comprehensive solution built around three main functionalities:

- **Automated extraction:** targeted retrieval of author identifiers and publication metadata according to customizable criteria (time periods, scientific domains, document types).
- **Interactive visualization:** automatic generation of dynamic graphics (histograms, time series, bar charts, word clouds) enabling immediate understanding of trends and patterns in the data.
- **Professional reporting:** production of structured reports in PDF and LaTeX formats, integrating visualizations for clear and actionable presentation.

The tool’s architecture prioritizes accessibility and ease of use. A graphical interface allows any user, regardless of their technical skills, to quickly analyze the scientific work of a particular laboratory or research group. The system only requires a structured CSV file containing at minimum the names and first names of the authors of interest.

The development of this system was based on a real use case: the analysis of publications from the MACS research group<sup>1</sup>. This approach ensures functional relevance and tool robustness under authentic operational conditions.

The final ambition is to deliver a versatile and scalable solution, suited both to academic needs (project evaluations, scientific activity reports) and institutional requirements (evaluation reports, dashboards for laboratory management, strategic decision support).

---

<sup>1</sup>The test data file is available in section 5.1

## 2 Tool Usage Modes

The tool offers two complementary approaches to exploit HAL scientific data extraction and analysis functionalities, each adapted to distinct user profiles and usage contexts.

### 2.1 Interactive Graphical Interface (`app.py` file)

The application, developed with the Tkinter library, provides an intuitive and accessible user experience. This graphical interface structures the process into two functional modules:

- **Extraction module:** Configuration and launching of retrieval queries according to customizable criteria (time windows, scientific domains, document typologies).
- **Analysis module:** Generation of interactive visualizations and export of structured reports in PDF and LaTeX formats.

This approach prioritizes accessibility and is particularly suitable for occasional users or those unfamiliar with command-line environments.

### 2.2 Command-Line Interface (`main.py` file)

Terminal execution offers a robust alternative for advanced users. This method enables:

- **Configurable extraction:** Precise definition of criteria via command-line arguments, facilitating integration into scripts and processing pipelines.
- **Batch generation:** Automated production of visualizations and reports without manual intervention, optimizing processing of large volumes.
- **Complete automation:** Ability to chain extraction, analysis, and reporting in a single command, ideal for recurring analyses.

This approach maximizes efficiency and reproducibility, particularly suited to research environments requiring systematic analyses.

Both modes guarantee equivalent analysis quality and access the same functionalities. The choice between graphical interface and command line depends primarily on user preferences, technical level, and usage context.

## 3 Interactive Application Overview

The `app.py` file constitutes the core of the project's graphical interface. Developed with the `tkinter` library, this application offers an intuitive user experience for data extraction, analysis, and report generation based on HAL API data.

### 3.1 Launching the Application

Two methods allow executing the application:

- **From an IDE:** Direct execution of the Python file in an environment such as Spyder or PyCharm
- **From the terminal:** Navigate to the `PROJET_HAL_/python` code folder and execute the command `python3 app.py`

### 3.2 General Architecture

The application is structured around three complementary functional modules:

- **Extraction Module:** Retrieval of HAL identifiers and extraction of publication metadata
- **Analysis Module:** Generation of visualizations, keyword analysis, and report production
- **Detection Module:** Identification of duplicates and homonyms

Upon launch, the user directly accesses the Extraction section via the welcome interface:

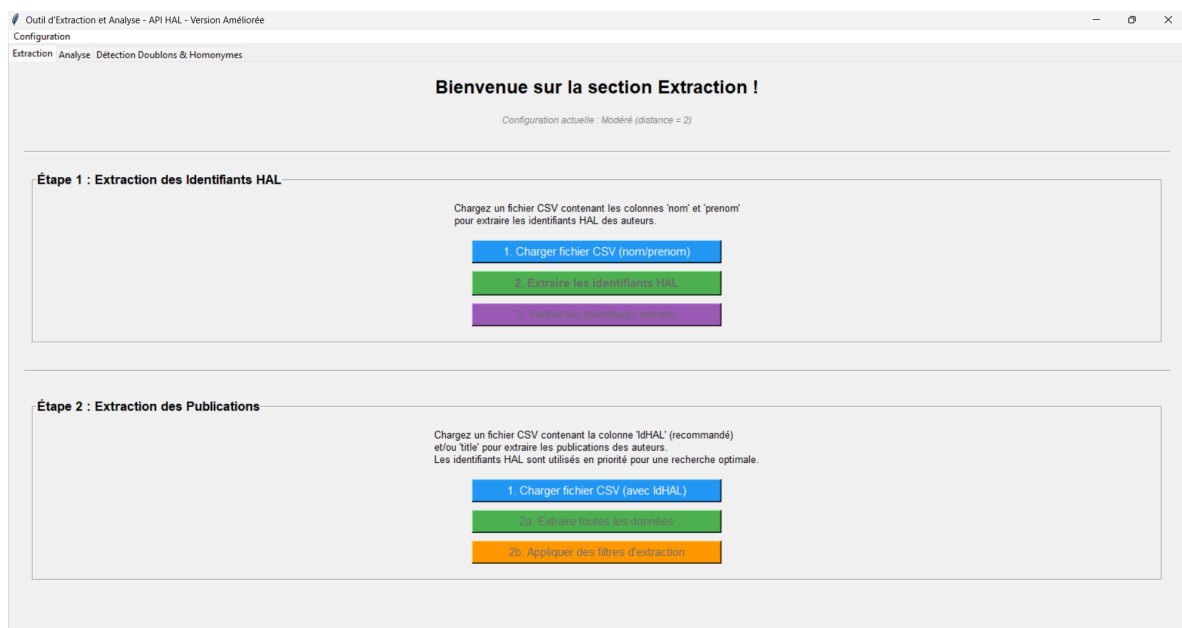


Figure 1: Welcome interface - Extraction Section

### 3.3 Data Extraction Module

The extraction process is performed in **two distinct steps** to ensure precision and quality of extracted data. The first step consists of identifying authors via their HAL identifiers, while the second retrieves their publications.

#### 3.3.1 Step 1: HAL Identifier Extraction

**Loading the Source File** The process begins by loading a CSV file containing the list of authors. The file must contain at minimum:

- Either a **title** column with the author's full name (e.g., "Jean-Luc DUPONT")
- Or the **nom** (last name) and **prenom** (first name) columns separately

The application accepts both formats and can even automatically generate the **nom** and **prenom** columns from the **title** column if it follows the convention: first name in mixed case and last name in UPPERCASE. However, it is still preferable for the input file to contain all three columns: **title**, **nom**, and **prenom**.

Once the file is validated, a confirmation message indicates the number of loaded authors and detected columns:

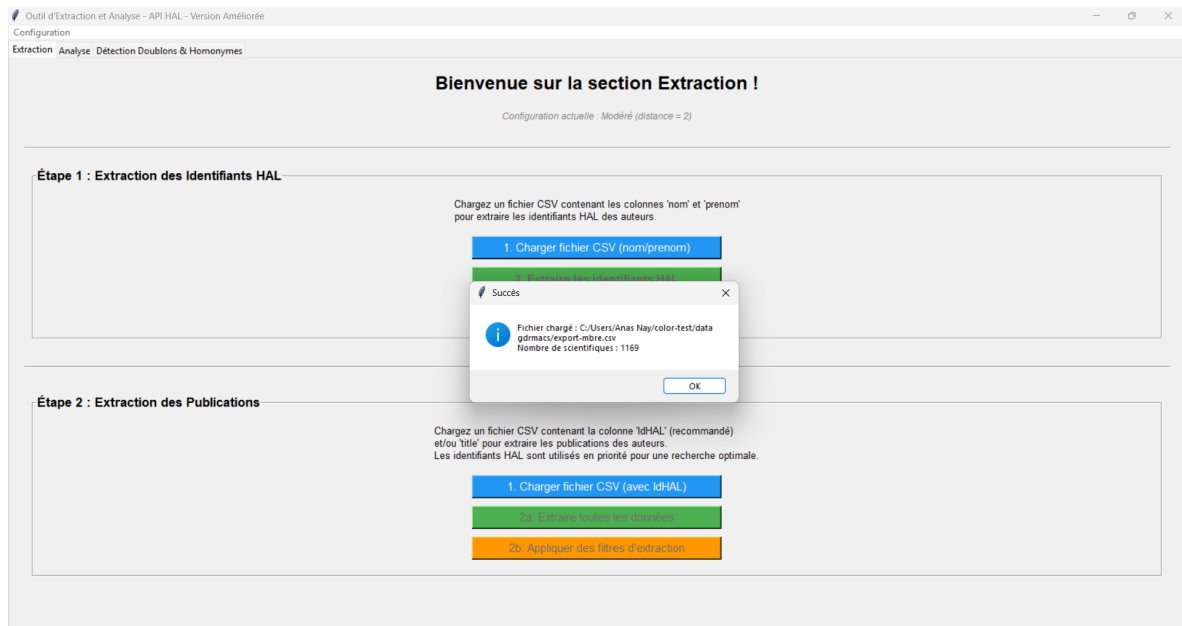


Figure 2: Loading the CSV file (nom/prenom) in Step 1 section

**Sensitivity Configuration** Before launching the extraction, it is possible to adjust the name matching sensitivity via the **Configuration > Matching Sensitivity** menu. This functionality, detailed in section 3.5, allows adapting the Levenshtein distance according to the quality of source data.



**Launching Identifier Extraction** Clicking the "2. Extract HAL identifiers" button triggers the analysis. A summary window then presents the extraction parameters.

Once launched, the extraction executes with display of a progress bar and advancement counter:

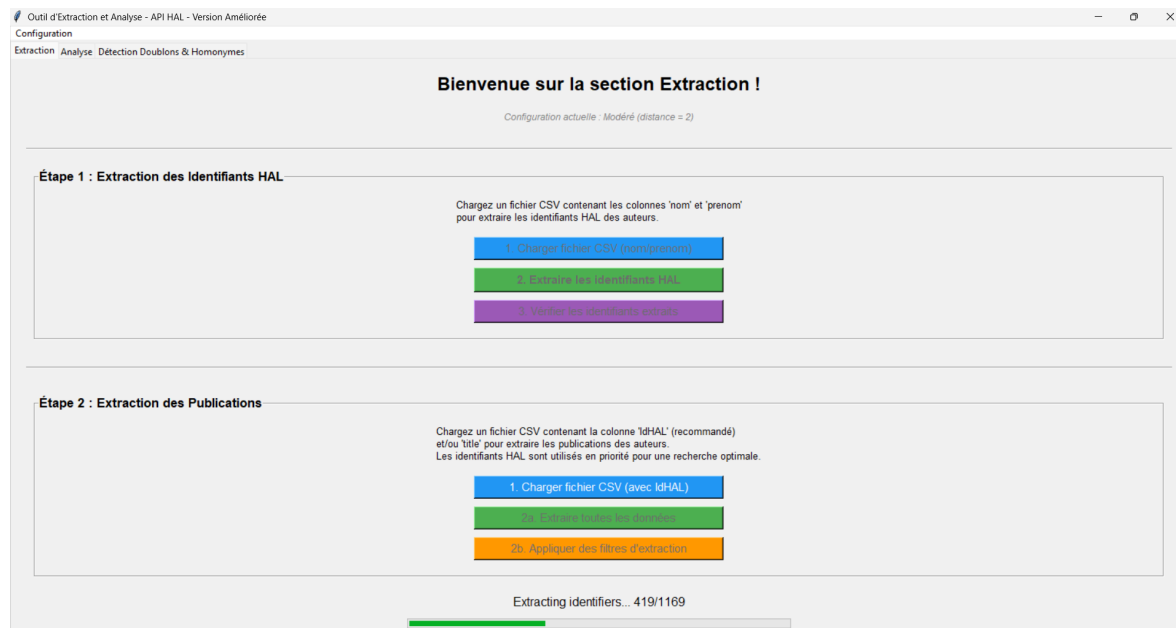


Figure 3: Author identifier extraction progress

Upon completion of extraction, a CSV file is automatically generated in the **extraction/** folder with the name: **filename\_hal\_id.csv**. This file includes all columns from the original file and adds:

- **IdHAL**: The author's HAL identifier (empty if not found)
- **Candidats**: List of alternative identifiers separated by commas
- **ID\_Atypique**: Indicator "OUI" (YES) or "NON" (NO) depending on whether the ID resembles the author's name
- **Details**: Technical debugging information in JSON format

A final message indicates the operation's success and signals any atypical identifiers requiring verification.

### 3.3.2 Verification and Correction of Extracted Identifiers

Once the first step is complete, it is possible to verify the extracted identifiers (cases of atypical identifiers or multiple candidates). The "3. Verify extracted identifiers" button then becomes accessible. This verification interface allows manual validation for cases requiring verification.

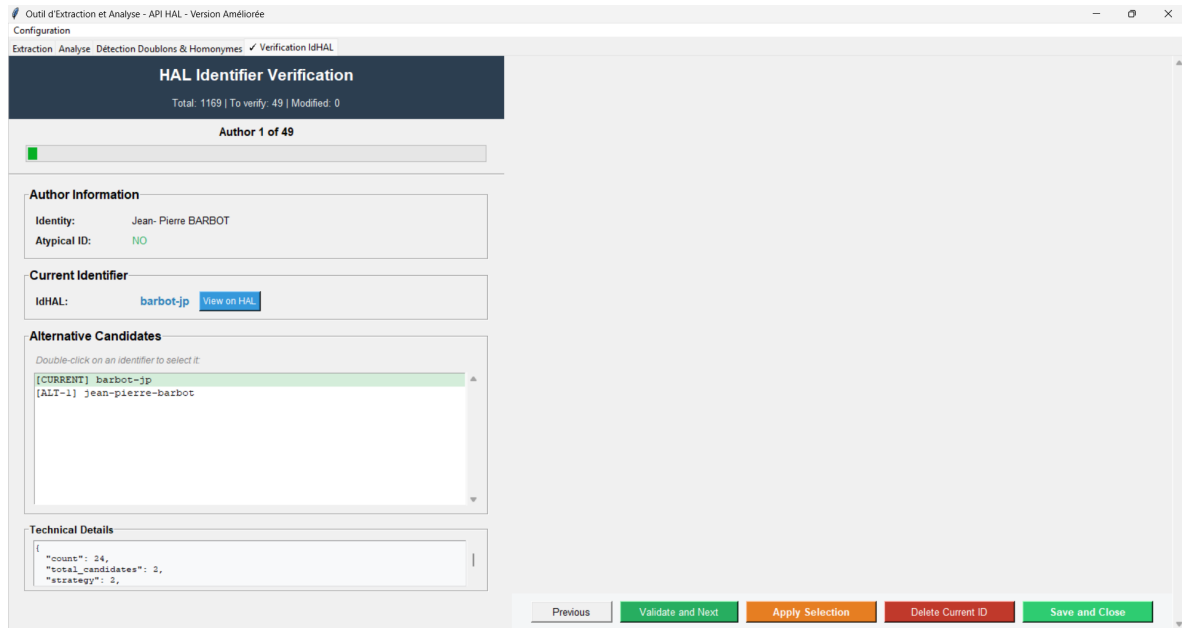


Figure 4: Identifier verification interface

The verification interface presents for each "problematic" author:

- The author's identity
- The ID status (atypical or not)
- The currently assigned identifier
- A link leading directly to the list of publications referenced under this identifier, allowing the user to verify it themselves
- The list of available alternative candidates
- Technical extraction details

Three actions are possible for each author:

1. **Validate and move to next:** Confirm the current identifier
2. **Select an alternative candidate:** Double-click on an ID in the list or use the "Apply Selection" button
3. **Delete the identifier:** "Delete Current ID" button if no ID is suitable

A progress counter indicates the verification advancement.

At the end of verification (after clicking the "Save and Close" button), a new file `filename_hal_id_verified.csv` is generated, containing all authors with applied corrections.

**Verification Use Cases** This verification step is particularly useful for:

- **Atypical identifiers:** IDs that do not resemble the author's name
- **Homonyms:** Multiple authors with the same name requiring distinction
- **Duplicates:** Authors present multiple times in the source file with different IDs
- **Multiple candidates:** Situations where multiple plausible IDs were found

Correcting duplicates at this stage prevents redundant publication extraction during step 2, thus ensuring coherence of subsequent analyses.

### 3.3.3 Step 2: Publication Metadata Extraction

**Loading the File with Identifiers** Once HAL identifiers are extracted (and optionally verified), step 2 consists of retrieving each author's publications. The process begins by loading the CSV file containing identifiers via the "1. Load CSV file (with IdHAL)" button.

The file must contain at minimum the `title` column. The presence of the `IdHAL` column is strongly recommended. If this column does not exist in your file, the fallback solution for extracting publication metadata will be to use the authors' names and first names (hence the `title` column). The application primarily uses the HAL identifier for queries (most precise method), then falls back on the full name if the ID is absent or empty.

**Available Extraction Modes** Two extraction approaches are offered according to analytical needs:

**Complete Extraction** This mode exhaustively retrieves all publications associated with listed authors, without temporal, thematic, or typological restrictions. Clicking 2a. **Extract all data** opens a summary window presenting the number of processed authors and applied sensitivity threshold:

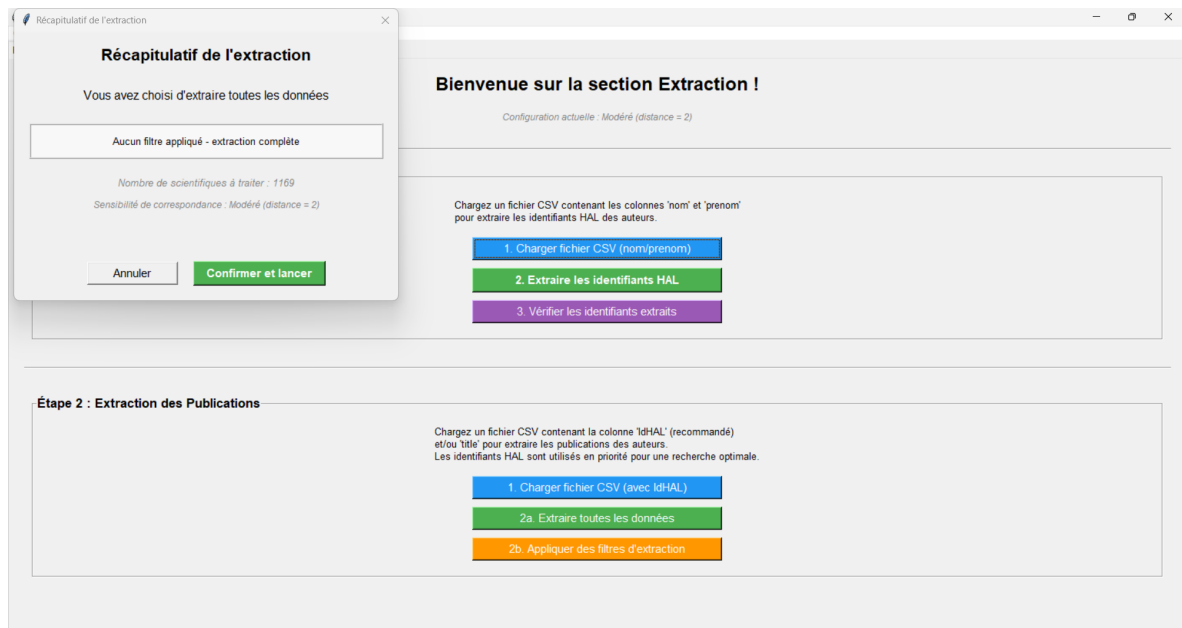


Figure 5: Complete extraction summary

**Extraction with Filters** This advanced option allows defining multiple filtering criteria via the **2b. Apply extraction filters** button:

- **Temporal window:** Restriction to a specific period in YYYY-YYYY format (e.g., 2019-2023)
- **Document typology:** Targeted selection (articles, theses, communications, reports, etc.)
- **Scientific domains:** Disciplinary filtering (computer science, mathematics, biology, chemistry, etc.)

The configuration interface centralizes these parameters:

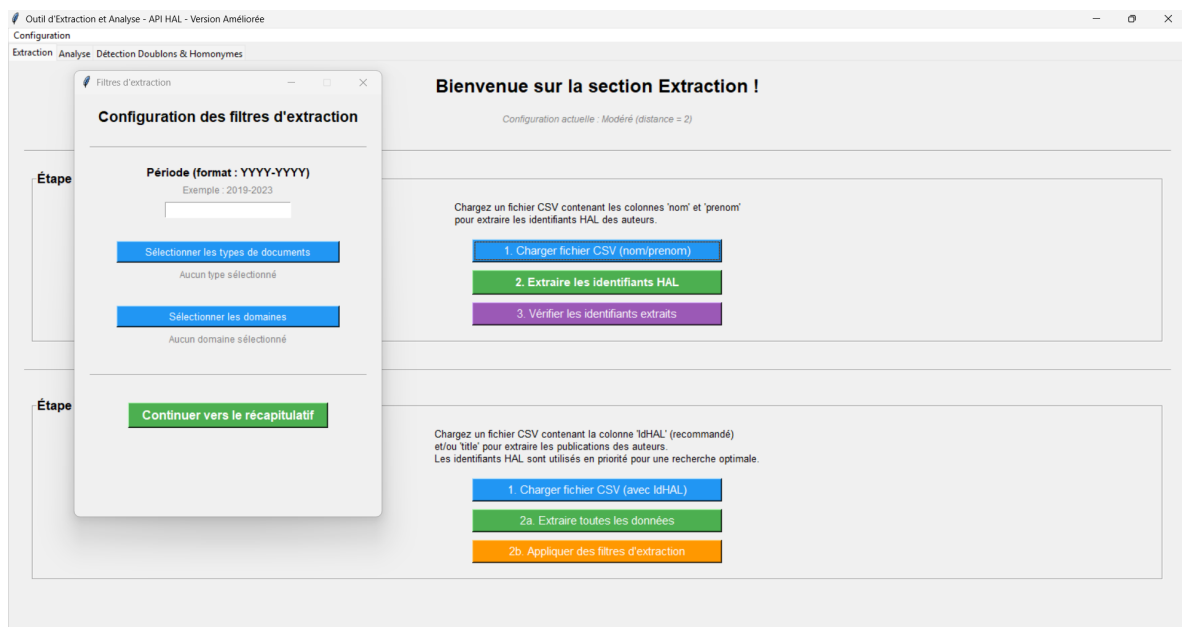


Figure 6: Filter configuration interface

A summary window validates the configuration before launch.

**Execution and Monitoring** Launching the extraction activates a progress bar with advancement counter.

Upon completion of extraction, results are automatically saved in the **extraction/** folder according to standardized nomenclature:

- Complete extraction: `all_data.csv`
- Filtered extraction: `all_data_{Domain}_{Period}_{Type}.csv`

Each line of the generated file corresponds to a publication and contains the following metadata (detailed in section 6.2):

- Author information: Last Name, First Name, Author's IdHAL
- Publication information: Title, Docid, Publication Year, Document Type, Domain, Keywords, Affiliated Research Laboratory
- Collaborations: IdHAL of publication authors (complete list)

A final message confirms the operation's success and indicates the location of the generated file.

**Advantages of the Two-Step Process** This two-phase architecture presents several benefits:

- **Traceability:** Preservation of extracted identifiers for later verification
- **Reusability:** Validated identifiers can serve for new extractions without re-searching
- **Precision:** Manual verification eliminates attribution errors due to homonyms
- **Performance:** Use of HAL IDs significantly accelerates publication extraction
- **Quality:** Reduction of duplicates and erroneous publications in final analyses

## 3.4 Data Analysis Module

The analysis module exploits CSV files generated during step 2 of extraction (publication metadata) to produce interactive visualizations, thematic analyses, and structured reports.

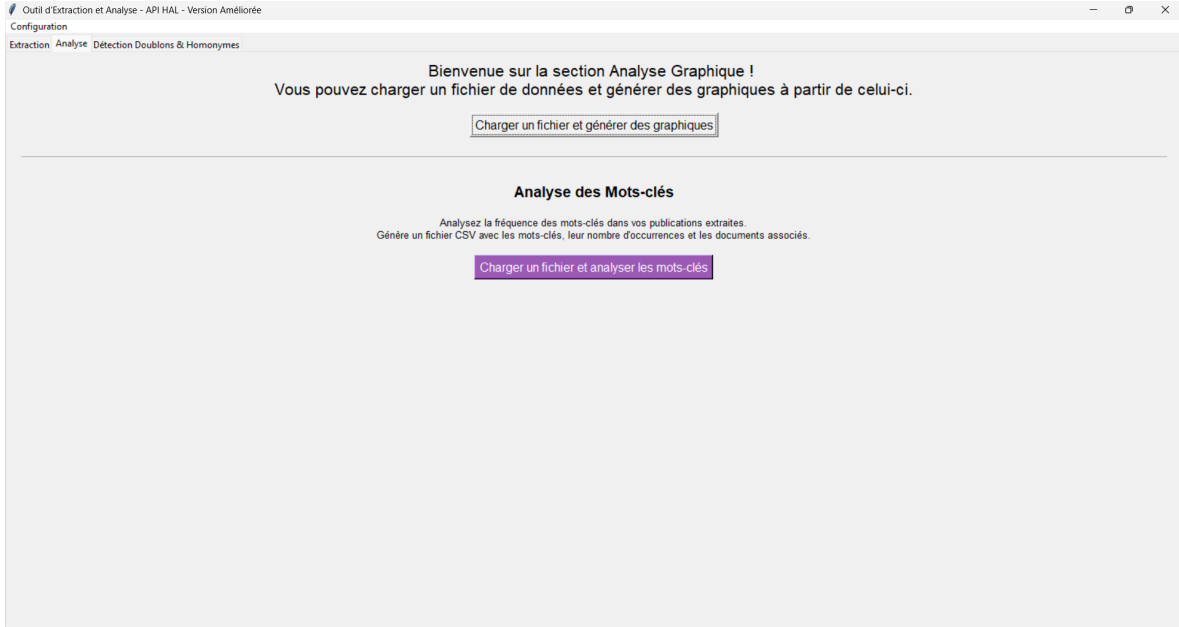


Figure 7: Analysis module interface

### 3.4.1 Automatic Generation of Graphical Visualizations

**Data Loading and Preparation** The process begins by loading a publications CSV file via the **Load file and generate graphics** button. This file must correspond to a publication extraction (file `all_data.csv` or filtered equivalent).

**Important note:** To ensure coherence and relevance of visual analyses, only publications dating from 2005 and after are taken into account in the graphics. Earlier data, often incomplete or less representative in HAL, are automatically excluded from visualization.

**Generation Process** Once the file is loaded, a progress bar indicates the advancement of graphics generation.

The system automatically produces 9 distinct visualizations in two complementary formats:

- **Interactive HTML versions:** Stored in the `html/` folder, developed with `plotly` for interactivity (zoom, hover, filtering)
- **Static PNG exports:** Stored in the `png/` folder, intended for integration into reports and presentations

**Catalog of Generated Visualizations** The application produces the following graphics, each providing specific analytical insight:

1. **Publications by year**

Interactive histogram presenting the quantitative evolution of scientific production year by year.

2. **Document type distribution**

Pie chart displaying the distribution of publications by typology (articles, theses, communications, reports, etc.).

3. **Most frequent keywords**

Horizontal bar chart classifying the most used thematic terms in publications.

4. **Main scientific domains**

Histogram identifying the most represented disciplines in the corpus.

5. **Publication trends**

Line chart illustrating the temporal evolution of scientific productivity with trend identification.

6. **Laboratory distribution**

Stacked bar chart presenting the distribution of publications by research structures.

7. **Temporal evolution by team**

Multi-curve chart tracking the publication activity of each research team over time.

8. **Theses and HDR by year**

Bar chart counting thesis and habilitation defenses by year.

9. **Thesis keyword cloud**

Vertical hierarchical list of the 15 main keywords of doctoral works, with visual emphasis by size and color.

A confirmation message displays upon completion of generation, summarizing the number of created graphics, execution time, and file location.



### 3.4.2 Interactive Dashboard Consultation

The **Display graphics** button, activated after generation, automatically launches a consolidated HTML dashboard in the default browser:



Figure 8: Consultation and export interface

This dashboard centralizes all visualizations with advanced interactive features enabled by the **plotly** library:

- Dynamic zoom on areas of interest
- Display of exact values on hover
- Interactive filtering by categories (click on legend)
- Individual export of graphics

### 3.4.3 Compiled Report Production

The **Generate report** button initiates the creation of a consolidated document integrating all generated visualizations. The user selects the desired output format:

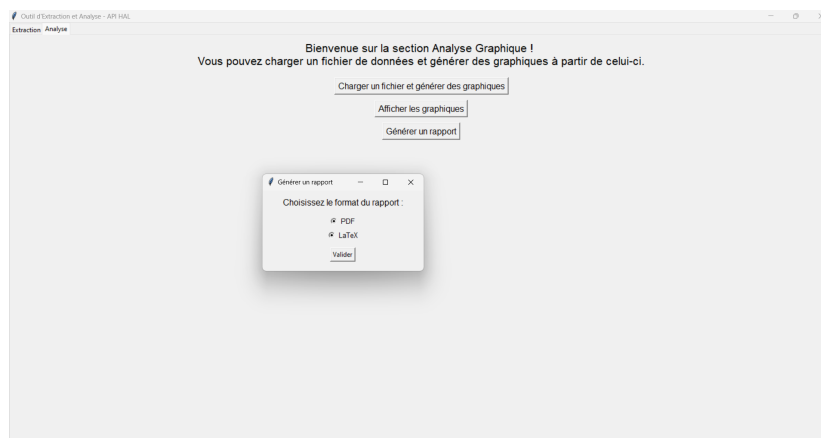


Figure 9: Report format selection

Two formats are offered:

- **PDF:** Directly consultable document, ideal for distribution
- **LaTeX:** Modifiable source code for advanced customization

Generated reports are automatically stored in the **rapports/** folder with integrated PNG images for optimal offline consultation. The report compiles all graphics into a structured and paginated document, accompanied by metadata about the analysis (period, number of publications, etc.).

### 3.4.4 Thematic Keyword Analysis

A complementary feature allows performing in-depth analysis of keywords present in publications via the **Load file and analyze keywords** button.

**Analysis Configuration** Loading a publications file opens a configuration window allowing analysis parameterization:

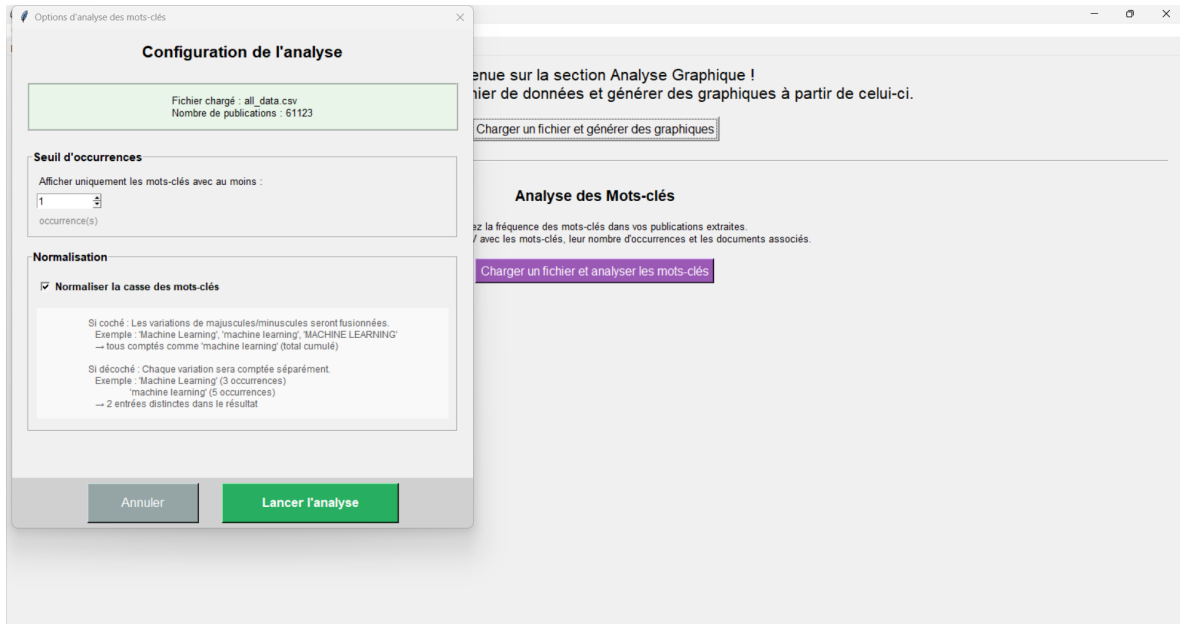


Figure 10: Keyword analysis configuration

Two parameters are configurable:

- **Minimum occurrence threshold:** Filters keywords appearing less than  $N$  times (default: 1)
- **Case normalization:** Option to merge uppercase/lowercase variants (e.g., "Machine Learning", "machine learning" → "machine learning")

**Analysis Execution** The analysis executes with display of a progress bar indicating the number of processed publications.

**Results File** Upon completion of analysis, a CSV file is automatically generated in the **extraction/** folder with the nomenclature:  
`filename_keywords_analysis.csv`

This structured file contains four columns:

- **Keyword:** The analyzed term (normalized if option was activated)
- **Occurrences:** Total number of keyword appearances in the corpus
- **Docids:** HAL identifiers of documents containing the keyword, separated by commas
- **Laboratories:** Unique list (no duplicates) of research structures associated with publications using this keyword, separated by commas

Results are sorted in descending order of occurrences, placing the most frequent keywords at the top of the list.

A final message confirms the analysis success and provides some statistics:

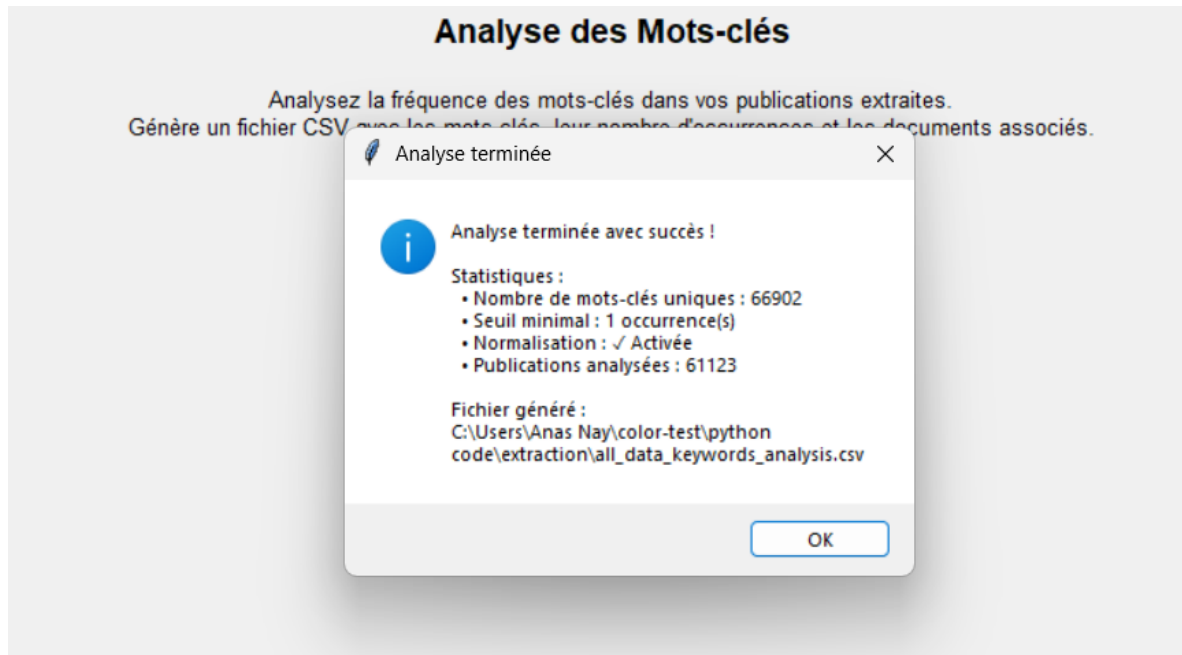


Figure 11: Keyword analysis statistics

**Analytical Applications** This thematic analysis notably allows:

- Identifying dominant research axes of a laboratory or team
- Mapping inter-laboratory collaborations on common themes
- Tracking the evolution of research topics by cross-referencing with temporal data
- Detecting thematic convergences between different structures

### 3.5 Matching Sensitivity Configuration

The application integrates an advanced functionality allowing customization of the sensitivity level used during the author name and first name matching process. This configuration is useful for optimizing extraction precision based on the quality and consistency of input data.

Access to these parameters is available from the main menu bar: **Configuration > Matching Sensitivity**....

The matching between names in the input CSV file and those in the HAL database relies on the Levenshtein distance, an algorithm that measures similarity between two character strings by calculating the minimum number of operations (insertions, deletions, or substitutions) required to make them identical. The lower this distance, the more similar the names are. This mechanism efficiently handles spelling variations, typos, accent differences, or common abbreviations.

The configuration window presents several predefined sensitivity levels, each corresponding to a specific distance value:

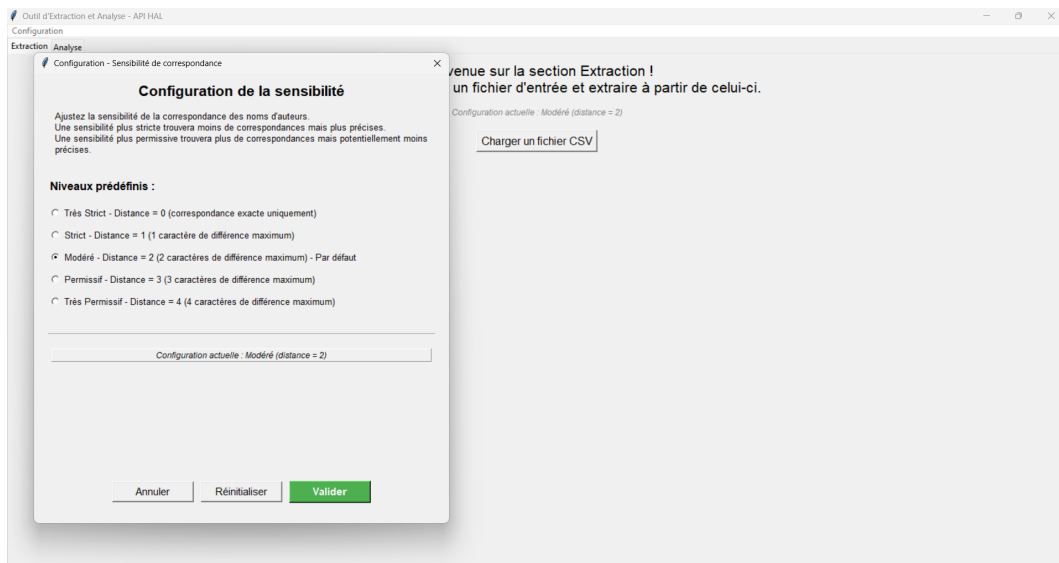


Figure 12: Sensitivity Configuration Interface

This interface offers five predefined levels:

- **Very Strict** (Distance = 0): Exact match only, no tolerance for variations
- **Strict** (Distance = 1): Tolerance for a single character difference maximum
- **Moderate** (Distance = 2): Default level, allows up to 2 character differences
- **Permissive** (Distance = 3): Extended tolerance for 3 character differences maximum
- **Very Permissive** (Distance = 4): Most tolerant level, accepts up to 4 character differences

**Application example:** For a name "Müller" in the CSV file:

- *Very Strict* level: only exact "Müller" will be found
- *Moderate* level: "Muller", "Müller" will be detected
- *Permissive* level: will also include "Miller", "Moller", etc.

An information box displays the current configuration in use.

In the previous image, the "Moderate" level is selected by default with a distance of 2. The interface also provides three action buttons: **Cancel** to close without saving, **Reset** to return to default parameters, and **Validate** to apply the new configuration before extracting data. Once validated, the new configuration will be confirmed by an informational message and will apply to all subsequent extractions.

## 4 Command-Line Interface Overview

The `main.py` file constitutes a complete command-line interface for extracting and analyzing scientific data from the HAL API. This approach offers a powerful alternative to the graphical interface, particularly suited for advanced users.

### 4.1 Main Functionalities

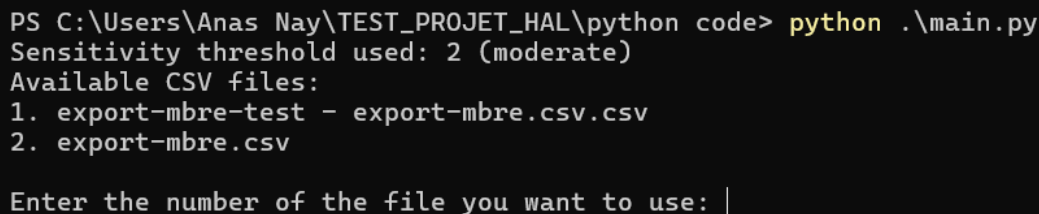
The script offers a comprehensive set of options allowing fine customization of data extraction and analysis:

- **Filtered extraction:** ability to target publications according to specific criteria
- **Sensitivity management:** configuration of author name matching threshold
- **Automatic generation:** creation of graphics and reports in a single command
- **Interactive interface:** guided selection of input files

### 4.2 Input File Selection

Unlike previous versions requiring manual file path modification, the current version offers an interactive interface to select the CSV file containing author names.

Upon script launch, a numbered list of available CSV files is displayed:



```
PS C:\Users\Anas Nay\TEST_PROJET_HAL\python code> python .\main.py
Sensitivity threshold used: 2 (moderate)
Available CSV files:
1. export-mbre-test - export-mbre.csv.csv
2. export-mbre.csv

Enter the number of the file you want to use: |
```

Figure 13: Interactive CSV File Selection

The user simply selects the number corresponding to the desired file. This improvement makes the tool more flexible and eliminates dependency on local configuration.

### 4.3 Filtering Options

The script accepts several optional arguments to filter extraction results:

- **-year:** specify a year range (YYYY-YYYY format)
- **-type:** filter by document type
- **-domain:** filter by scientific domain
- **-threshold:** configure name matching sensitivity (0-4)

## 4.4 Sensitivity Configuration

The new version integrates matching sensitivity management directly in command line via the `--threshold` argument. Available levels are:

- **0**: Very strict (exact match only)
- **1**: Strict (1 character difference maximum)
- **2**: Moderate (2 characters difference maximum) - *Default*
- **3**: Permissive (3 characters difference maximum)
- **4**: Very permissive (4 characters difference maximum)

## 4.5 Automatic Content Generation

Three new options enable automatic generation of visualizations and reports:

- `-graphs`: automatic graphics generation and dashboard opening
- `-reportpdf`: automatic PDF report creation
- `-reportlatex`: automatic LaTeX report generation

## 4.6 Utility Commands

The script also offers information commands:

- `-list-domains`: display all available scientific domains
- `-list-types`: list all supported document types
- `-list-sensitivity`: detail sensitivity levels

## 4.7 Usage Examples

### 4.7.1 Simple Extraction

```
python main.py
```

Extract all data without filters.

### 4.7.2 Extraction with Filters

```
python main.py --year 2019-2024 --domain "Mathematics" --type "Theses"
```

Extract mathematics theses published between 2019 and 2024.

### 4.7.3 Extraction with Custom Sensitivity

```
python main.py --threshold 1 --domain "Computer Science"
```

Extract with strict name matching for computer science domain.



#### 4.7.4 Extraction with Automatic Generation

`python main.py --graphs --reportpdf --reportlatex`  
Complete extraction with automatic graphics and reports generation.

### 4.7.5 Information Commands

```
python main.py --list-domains
python main.py --list-types
python main.py --list-sensitivity
python main.py -h
```

## 4.8 Progress Tracking and Results

The script displays a native progress bar during extraction, including:

- Completion percentage
- Number of processed elements
- Visual progress bar and Estimated time remaining (ETA)

## 4.9 Pre-extraction Summary

Before starting extraction, the program displays a formatted summary of selected parameters:

[illegible]

Figure 14: Pre-extraction Summary with Progress Display

## 4.10 Output File Organization

Results are automatically organized:

- **Extraction CSV files:** extraction/ folder
- **HTML graphics:** html/ folder
- **PNG images:** png/ folder
- **Reports:** rapports/ folder

## 5 Duplicate and Homonym Detection Module

This module constitutes an advanced functionality of the tool allowing automatic identification and processing of duplicate publications and homonymy cases in data extracted from HAL. This functionality relies on a method using author identifiers in the HAL database (`authIdPerson_i`) to guarantee maximum precision in detection.

### 5.1 Operating Principle

The detection system is based on a multi-step algorithm:

1. **Initial grouping:** Publications are grouped by author (name, first name) pairs
2. **HAL API enrichment:** For each publication, a query is performed to the HAL API to retrieve complete metadata, notably the `authIdPerson_i` identifiers
3. **Comparative analysis:** Publications from the same author are compared according to several criteria:
  - Title similarity (default threshold: 0.8)
  - Temporal gap between publications (default threshold: 2 years)
  - Official HAL identifiers
  - Author position in the co-author list
4. **Automatic classification:** Detected cases are classified into different categories

### 5.2 Module Access from Graphical Interface

The detection module is accessible via the "Duplicate and Homonym Detection" tab of the main interface:

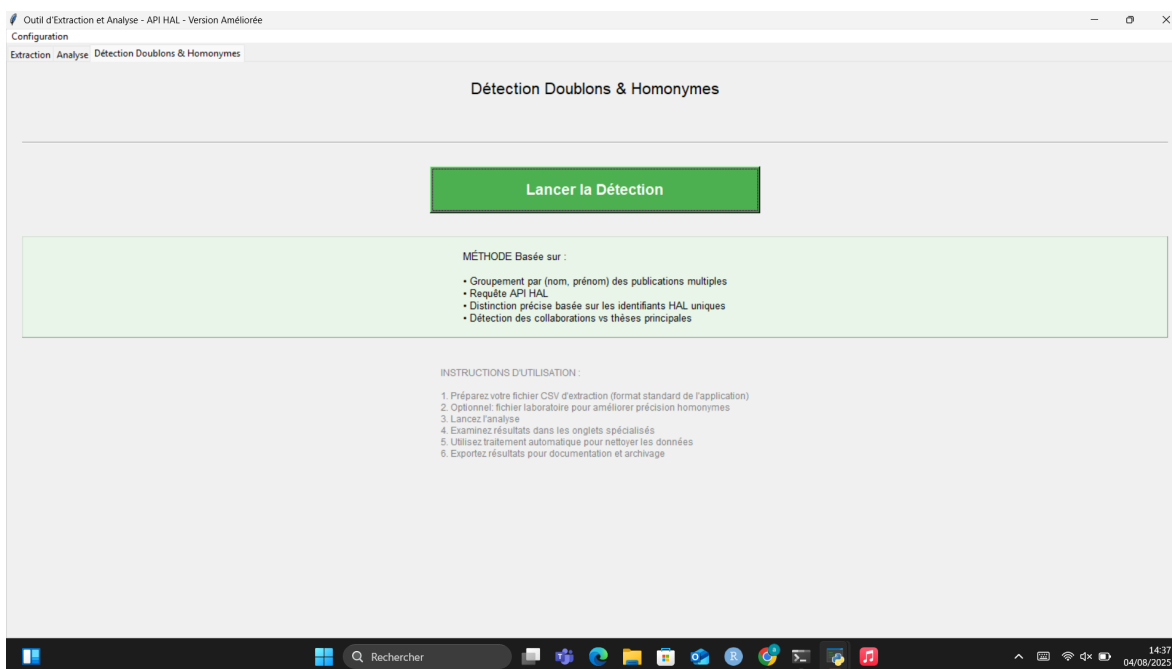


Figure 15: Detection Module Welcome Interface

The interface presents the main characteristics of the method used and provides a central button to launch the analysis. The process requires a CSV extraction file previously generated by the tool.

## 5.3 Analysis Configuration and Launch

### 5.3.1 Input File Selection

The analysis process begins with the selection of the CSV file to analyze. The system automatically proposes files from the **extraction** folder and asks if the user wishes to use an optional laboratory file.

### 5.3.2 Optional Laboratory File

The user can optionally provide a file containing laboratory information for authors. This file must contain the columns **nom**, **prenom**, and **unite.de.recherche**. This additional information significantly improves homonym detection precision (by referring to authors' laboratory affiliations, which allows better differentiation between them).

## 5.4 Analysis Interface and Results

Once the files are selected, the analysis starts automatically in a dedicated interface organized into thematic tabs.

### 5.4.1 Summary Tab

The results are initially presented in a tab summarizing the analysis. Here is an example:

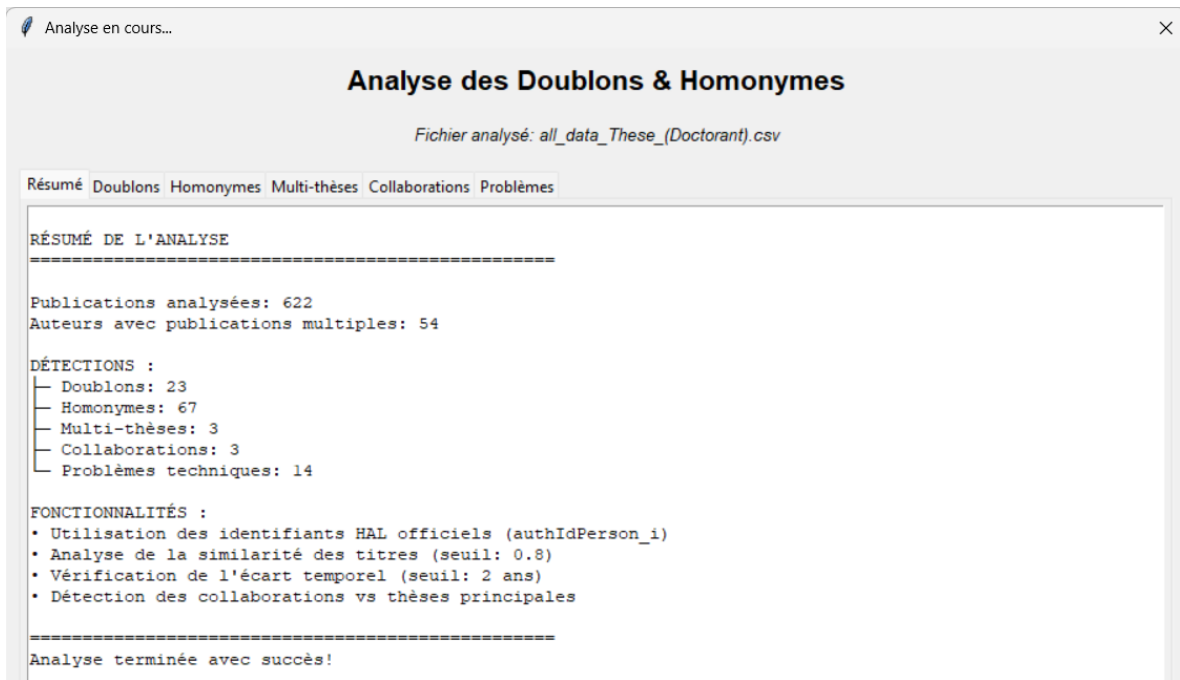


Figure 16: Duplicate and Homonym Detection Summary

This tab presents an overview of analysis results with global statistics including:

- Total number of publications analyzed
- Number of authors with multiple publications
- Detections by category (duplicates, homonyms, multi-theses, collaborations)
- Technical information about the method used

#### 5.4.2 Detailed Results Tabs

The different tabs present detected cases organized by category:

- **Duplicates Tab:** Publications identified as potential duplicates with similarity scores, compared titles, and publication years
- **Homonyms Tab:** Homonymy cases with differentiation criteria (different HAL identifiers, distinct scientific domains, different laboratories)
- **Collaborations Tab:** Distinction between main theses and collaborations of the same author
- **Multi-theses Tab:** Rare cases of authors with multiple theses
- **Issues Tab:** Publications without `authIdPerson_i` or incomplete metadata

## 5.5 Supported Detection Types

The module identifies several problem categories:

Table 1: Detection Types and Criteria

Type	Detection Criteria
Duplicates	Same <code>authIdPerson_i</code> , title similarity > 0.8, temporal gap < 2 years
Homonyms	Different <code>authIdPerson_i</code> , same (name, first name), distinct domains/laboratories
Multi-theses	Same <code>authIdPerson_i</code> , temporal gap > 3 years, low title similarity
Collaborations	Non-principal position in author list, different domains
Technical issues	Absence of <code>authIdPerson_i</code> , incomplete HAL metadata

## 5.6 Automatic Data Processing

The module offers automatic processing functionalities to clean detected data via a dedicated interface accessible after analysis.

### 5.6.1 Available Processing Options

Several processing levels are offered:

- **Duplicate removal:** Automatic elimination of duplicated publications (keeping the first occurrence)
- **Homonym marking:** Addition of a `Homonyme_Potentiel` column to flag ambiguous cases
- **Collaboration removal:** Elimination of collaborations while keeping only main theses
- **Multi-thesis flagging:** Marking of rare cases of multiple theses by the same author

### 5.6.2 Processing Results

After processing, the system generates:

- A cleaned CSV file (suffix `_nettoye.csv`)
- A detailed report of actions performed
- Comparative statistics before/after processing

## 5.7 Results Exportation

The module offers comprehensive export functionalities to document and archive analysis results. The user can choose the destination folder and the system automatically generates several specialized files.

### 5.7.1 Generated Files

The export produces several specialized files:

- `*_doublons_detecte.csv`: Detailed list of duplicates with similarity scores
- `*_homonymes_detecte.csv`: Homonymy cases with differentiation information
- `*_multi_theses.csv`: Rare cases of multiple theses per author
- `*_collaborations.csv`: Main theses/collaborations distinction
- `*_resume_detecte.txt`: Complete analysis report with methodology

## 5.8 Command-Line Usage

The detection module is also accessible via the command-line interface with the `--analyse` argument:

```
python main.py --analyse
```

This approach offers the same functionalities in an interactive textual environment:

1. Interactive selection of the CSV file to analyze
2. Optional configuration of the laboratory file
3. Analysis with real-time results display
4. Integrated processing and export options

### 5.8.1 Command-Line Session Example

```
# Launch analysis
python main.py --analyse

# File selection (interactive interface)
Available files in 'extraction/':
1. all_data_These_Doctorant.csv
2. publications_2020-2024.csv
Select a file (1-2): 1

# Laboratory file configuration
Use a laboratory file? (y/n): y
Laboratory file selected: laboratoires.csv

# Analysis in progress with progress bar
Analysis in progress... (patience required - HAL API querying)
Analyzing Dupont Jean (3 publications) - 15/120...

# Detailed results display
=====
ANALYSIS RESULTS
=====
Publications analyzed: 450
Authors with multiple publications: 120
Duplicates detected: 23
Homonyms detected: 8
Collaborations detected: 15
```

## 5.9 Limitations and Considerations

### 5.9.1 HAL API Dependency

Detection precision depends on HAL metadata quality:

- Publications without `authIdPerson_i`: processing by title similarity only
- Incomplete metadata: classification in "technical issues"
- HAL data evolution: possible variations over time

### 5.9.2 Special Cases

Some cases require manual validation:

- Author name changes (marriage, etc.)
- International collaborations with spelling variations
- Publications with metadata errors on the HAL side

## 5.10 Conclusion

The duplicate and homonym detection module constitutes a powerful tool for improving bibliometric analysis data quality. By combining the use of official HAL identifiers with advanced heuristics, it enables reliable automatic detection of most problematic cases while providing appropriate processing solutions.

The integration of this module into the global tool ensures a complete processing chain, from initial extraction to final analysis, through a data cleaning and validation phase that significantly improves the relevance of obtained results, and consequently the statistics generated thereafter.



## 6 Appendix

### 6.1 Expected CSV File for Extraction

To download and view the type of CSV file required for extraction, please use the following link:

[Download the CSV file here](#)

### 6.2 Description of the Obtained CSV File

The CSV file obtained after data extraction, whether via the application interface or by executing the `main.py` file, presents a uniform structure. The file will contain the same information regardless of the method used for extraction. Each line in the CSV file represents a publication. Here is a description of the typical columns found in this file:

Table 2: CSV File Column Description

Column	Description
Nom	The author's last name.
Prénom	The author's first name.
IdHAL de l'Auteur	HAL identifier of the author.
IdHAL des auteurs de la publication	HAL identifiers of the publication authors.
Titre	The publication title.
Docid	The publication identifier.
Année de Publication	The publication year.
Type de Document	The document type, for example article, thesis, etc.
Domaine	The scientific domain of the publication.
Mots-clés	The keywords associated with the publication.
Laboratoire de Recherche	The laboratory or research center associated with the publication author.