Epidemium Oncobase

Unifying Scientific Information on Cancer

The OncoBase Team

Challenge 4 Cancer
La Paillasse & Roche

Midterm presentation, March 12, 2016

Outline

1. Introduction
2. Architecture
3. Online databases
4. Scientific literature
5. Conclusion

The Epidemium Oncobase Project

What?

Collecting, cleaning, homogenizing and unifying heterogeneous data to produce accurate and reliable information on cancer.

Why?

To draw relevant conclusion, analyses require accurate, reliable, traceable and high-quality data but:

1. The amount of data is huge (21 000 datasets on data.epidemium.cc) and difficult to manage "by hand" without loosing information
2. Eclectic sources: national statistics, surveys, medical studies, scientific articles...
3. Data formats are diverse: html, csv, json, xls, xml, pdf, api...
4. Data sets are extremely heterogeneous: ≠ size, ≠ accuracy, ≠ reliability
5. 2 countries, institutions, organizations, sources can have ≠ metrics
6. Context and traceability → primary importance
7. Poor quality data → irrelevant or wrong conclusions

⇒ Uniformization

The Epidemium Oncobase Project

Who?

The OncoBase Team

When?

During the Challenge 4 Cancer timeframe (November 5, 2015 → May 5, 2016). . . and after

Where?

- Paris, France
- Urbana-Champaign, Illinois, USA

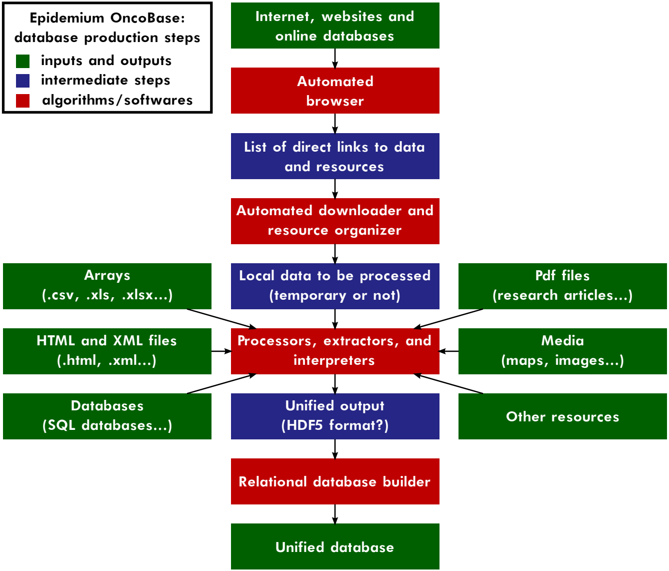
How?

- Using public and open access data
- Using the tools provided by La Paillasse
- Designing algorithms, implementing them in Python/C++
- Releasing our tools as free software

Goals

- Main goal: producing tools to create a uniform database on cancer
- Alternative goals: analyzing this database

Architecture overview



Epidemium OncoBase: database production steps

🟩 inputs and outputs

🟦 intermediate steps

🟥 algorithms/softwares

Internet, websites and online databases

Automated browser

List of direct links to data and resources

Automated downloader and resource organizer

Local data to be processed (temporary or not)

Arrays (.csv, .xls, .xlsx...)

Pdf files (research articles...)

HTML and XML files (.html, .xml...)

Processors, extractors, and interpreters

Media (maps, images...)

Databases (SQL databases...)

Unified output (HDF5 format?)

Other resources

Relational database builder

Unified database

Online databases

Goal

Being able to automatically search, download, process, clean, convert and compile information from a wide variety of sources and databases

First step: from heterogeneous data to unification

1. Automatically browse websites and databases
2. Identify relevant data
3. Automatically download them while keeping the context and the source
4. Process files and extract the relevant information
5. Convert it to a unified format

Second step: building meaningful information

1. Start from the unified format
2. Build relational databases on it
3. Exploit these relational databases

Online databases

Liberté • Égalité • Fraternité RÉPUBLIQUE FRANÇAISE data.gouv.fr

Epidemium Portail OpenData

Starting with data.gouv.fr

- Public data produced or received within public service tasks
- 21 841 data sets
- 93 193 resources
- 673 organizations

Why transforming data?

In most cases, raw data needs to be processed to fit the requirements of analyses:

- missing information
- wording, names, titles need to be standardized
- data needs to be converted or completed (e.g. GPS coordinates → ZIP code)

Short term task

Automatically process these data sets, cleaning and enriching them

Analyzing scientific literature

Goal

Being able to automatically extract information from scientific literature.

PubMed as a starting point

PubMed

- Medicine and life sciences
- More than 25 million bibliographic references
- Of which about 1 250 000 are in open access

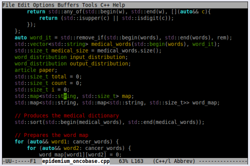
Analyzing scientific literature

Analyzing open access articles

We are creating a tool to automatically analyze the open access scientific literature available on PubMed.

About our tool

- C++14 for max performance
- Generic programming
- Multithreading



```
        return std::any_of(std::begin(w), std::end(w), [](auto&& c){
            return (std::isupper(c) || std::isdigit(c));
        });
    };
    auto word_it = std::remove_if(std::begin(words), std::end(words), rem);
    std::vector<std::string> medical_words(std::begin(words), word_it);
    std::size_t medical_size = medical_words.size();
    word_distribution input_distribution;
    word_distribution output_distribution;
    article paper;
    std::size_t total = 0;
    std::size_t count = 0;
    std::size_t i = 0;
    std::map<std::string, std::size_t> map;
    std::map<std::string, std::map<std::string, std::size_t>> word_map;

    // Produces the medical dictionary
    std::sort(std::begin(medical_words), std::end(medical_words));

    // Prepares the word map
    for (auto&& word1: cancer_words) {
        for (auto&& word2: cancer_words) {
            word_map[word1][word2] = 0;
```

-UU-:----F1 epidemium_oncobase.cpp 63% L163 (C++/l Abbrev) ----------------

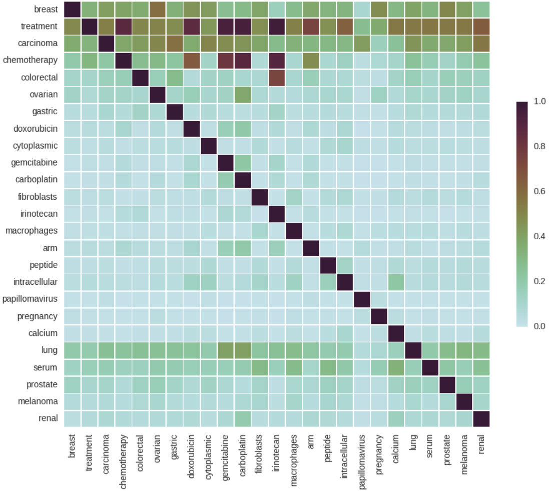
1. Downloads all the open access articles in `.txt` and `.nxml` when available
2. Automatically updates the local files when necessary
3. Iterates over the files, load them in memory and parses them
4. Extracts relevant information
5. Try to give a meaning to the article
6. Automatically detects result tables and extract them with context

Analyzing scientific literature

Preliminary illustration

- Analysis of a subset of 250 000 articles
- Creation of correlation-like matrices of medical words to see how they are related
- Can create 2D matrices of 10 000×10 000 or 3D matrices of 1000×1000 × 1000 words

Probability that a cancer-related article discussing [column] also discusses [row]



Midterm conclusion

Challenges

- Organizational challenges France/USA
- Mixing algorithmic, technical and medical problems
- Understanding the structure of medical data

Achievements

- Clear technical architecture of the project
- Review of main data sources
- Experimental tests of automated internet searches with Python Mechanize
- Functional implementation in C++ of a scientific literature analyzer
- Analysis of word correlations in 250 000 articles from PubMed

What's next?

- Automated data aggregation from data.epidemium.cc and data.gouv.fr
- Production of HDF5 files
- New algorithms for the scientific literature analyzer (meaning from nxml files)
- 3D word correlation analysis for the 1 250 000 PubMed open access articles
- And more. . .

Thank you for your attention

Any question?

Join us at www.epidemium.cc

- Project page: http://www.epidemium.cc/project/25/view
- Wiki page: http://wiki.epidemium.cc/wiki/OncoBase
- GitHub link: https://github.com/vreverdy/epidemium_oncobase
- Slack page: https://epidemium-oncobase.slack.com/messages/general