**DrChicco: AI-powered Biomedical Research Assistant**

**🔬 Overview**

**DrChicco** è un sistema modulare progettato per facilitare la ricerca scientifica automatica su temi biomedici. L'obiettivo è supportare studenti, ricercatori e professionisti nella raccolta, filtraggio, analisi e sintesi di articoli scientifici, sfruttando API pubbliche (PubMed, EuropePMC) e modelli LLM (OpenAI).

Il sistema è costruito in ottica MVP (Minimum Viable Product) ma con una struttura estendibile verso un'app completa.

**🏛️ Architettura del Progetto**

DrChicco/

├── main.py # Entry point del programma

├── config/

│ └── api\_keys.json # Chiavi API (PubMed, OpenAI)

├── data/

│ ├── processed/ # Risultati filtrati e dati estratti

│ └── reports/ # Report finali generati

├── src/

│ ├── agents/

│ │ ├── filter.py # Filtro di rilevanza articoli via LLM

│ │ ├── miner.py # Estrazione dati strutturati dagli abstract

│ │ ├── synthesizer.py # Sintesi e generazione report

│ │ ├── retriever.py # Wrapper PubMed (via pymed)

│ │ └── retriever\_custom.py # Wrapper EuropePMC (fallback)

│ └── tools/

│ └── europepmc\_wrapper.py # Codice di supporto alla ricerca su EuropePMC

├── tests/

│ └── test\_agents.py # Test automatici con Pytest

└── README.md # Documentazione (questo file)

**🌐 Funzionalità Attuali**

**✅ Ricerca Articoli**

* Prima cerca su **PubMed** con l'API pymed.
* Se non trova nulla, cerca su **EuropePMC** (via REST API).

**✅ Filtro Rilevanza**

* Per ogni titolo + abstract: chiede all'LLM se l'articolo è rilevante.

**✅ Estrazione Dati**

* Estrae concetti chiave dagli abstract (studio, popolazione, outcome...)

**✅ Generazione Report**

* Crea un file Markdown con una sintesi coerente, leggibile e ben strutturata.

**✅ Salvataggio Locale**

* I dati vengono salvati su disco in /data/processed/ e /data/reports/.

**✅ Testing Automatico**

* I test Pytest validano output JSON, stringhe e logica dei moduli.

**⚖️ Dipendenze e Requisiti**

**Python:**

* Versione consigliata: 3.10+

**Librerie principali:**

pip install -r requirements.txt

openai

pyautogen

pymed

requests

pytest

**API keys necessarie (in config/api\_keys.json):**

{

"openai\_api\_key": "sk-xxxx",

"ncbi\_api\_key": "your-ncbi-key"

}

**📊 Diagramma di Flusso**

flowchart TD

A[main.py] --> B[Retriever (PubMed)]

B -->|se vuoto| C[EuropePMCWrapper]

B -->|se articoli| D[Filter Agent (LLM)]

C --> D

D --> E[Relevant Results]

E --> F[Data Miner (LLM)]

F --> G[Mined Data]

G --> H[Synthesizer (LLM)]

H --> I[Markdown Report]

**⚡ Cosa Funziona Bene**

* Modularità codice
* Fallback automatico EuropePMC
* Test unitari già attivi
* Risultati persistenti in JSON + report Markdown

**📈 Roadmap Migliorativa**

**✍️ Fase 1: Ricerca avanzata**

* Aggiungere filtri temporali (es. ultimi 5 anni)
* Ricerca per tipo di studio (es. "randomized trial")
* Interfaccia con query DSL semplificata (tipo builder logico)

**📅 Fase 2: Frontend Web**

* Interfaccia in Flask/Streamlit/Gradio
* Selezione keyword + filtri via form
* Download risultati + PDF viewer integrato

**🔮 Fase 3: Analisi approfondita**

* Classificazione semantica degli articoli
* Rilevamento bias o limiti nello studio
* Valutazione automatica forza dell'evidenza

**🔐 Fase 4: Auth + Database**

* Salvataggio utenti e progetti
* Versioning dei report

**🌟 Obiettivo Nirvana**

Creare uno **strumento AI-driven per la letteratura scientifica** in grado di:

* Ricercare articoli scientifici autorevoli
* Filtrarli per qualità e rilevanza
* Analizzarli semanticamente
* Generare un report leggibile e citabile
* Offrire un frontend web accessibile e veloce

Il tutto con API chiave, interfaccia snella e focus sulla **produttività scientifica**.

**📍 Prossimi Passi Immediati**

* Costruire interfaccia minimale (es. Streamlit)
* Integrare filtri temporali nella query PubMed
* Mostrare articoli trovati in console con metadati
* Espandere test con casi limite
* Snellire setup iniziale con script automatico

Per qualsiasi miglioria o contributo, sei nel posto giusto ✅.

Powered by Riccardo + LLM 🧑‍💻 + Python