SynBioKB-Agent: A Multi-Agent System for Automated Biosynthesis Knowledge Extraction and Integration

Vikas Upadhyay

1 Introduction

The rapid growth of synthetic biology has created an overwhelming amount of fragmented knowledge across journal articles, patents, and online repositories. Data critical for pathway design and optimization—such as yields, titers, enzyme variants, and fermentation parameters—is often buried within narrative text, tables, or figures. Manual curation efforts like MCF2Chem provide accuracy but are slow to update. Other platforms, such as SynBioHub, are primarily focused on DNA part design rather than production data, while ontology-driven initiatives like SBKS remain incomplete.

To address this gap, we developed **SynBioKB-Agent**, a modular multi-agent system that automates the process of searching, crawling, extracting, validating, and synthesizing biosynthesis knowledge into a structured, queryable database. The current system focuses on enzyme engineering, metabolic engineering, pathway synthesis, and production metrics—with future extensibility toward DNA-level design integration.

2 System Goals and Scope

The goal of SynBioKB-Agent is to create a continuously updated knowledge base of biosynthetic pathways and production metrics. The system is designed to:

- Extract structured data: yields, titers, productivities, host organisms, enzymes, substrates.
- Summarize literature into structured JSON and normalized database records.
- Orchestrate specialized agents for query planning, data gathering, summarization, validation, and synthesis.
- Store validated results in a relational database (SQLite) with schema support for metrics, organisms, enzymes, feedstocks, and modifications.

- Output both machine-readable knowledge (JSONL, DB) and human-readable reports (PDF, Markdown).
- Support reproducible workflows via CLI, Docker/Compose, and Nextflow integration.

3 Pipeline Architecture

The pipeline consists of sequential but modular stages, each with specific tools and agents:

1. **Plan (Strategist Agent)**: Expands queries into sub-queries, prioritizing open-access sources. Implemented with LLMs, tested on **Ollama** with **GPT-OSS-20B**, scalable to larger context models (GPT-4/5, Claude, DeepSeek).
2. **Gather (Crawler Agents)**:
   - Literature search: **SerpAPI**, **Brave API**.
   - Access filtering: **Unpaywall API**.
   - Retrieval: **httpx** (robots.txt compliant), **Playwright** for headless browsing.
3. **Generate (Summarizer Agent)**: Text extraction via **trafilatura** (fallback: BeautifulSoup). Summarization with LLM into structured JSON (`title, year, chemical, organisms, enzymes, metrics, results, summaries`). Handles long texts using chunking and filtering.
4. **Validate (Validator Agent)**:
   - Deterministic checks: years, units, EC numbers, numeric plausibility.
   - RAG validation: retrieves supporting text and confirms metrics with LLM.
   - Entity enrichment: **PubChem, UniProt, NCBI Taxonomy**.
5. **Store (KB Writer Agent)**: Inserts outputs into a normalized **SQLite DB**. Schema covers summaries, metrics, organisms, enzymes, substrates, feedstocks, reaction steps.
6. **Synthesize (Composer Agent)**: Merges multiple summaries into reports, building chronological narratives, comparison tables, and inline citations.
7. **Experimentalist Agent**: Extracts practical insights: host choices, pathway logic, fermentation strategies, and common bottlenecks.
8. **Compute Agent**: Normalizes units, aggregates metrics, identifies best-in-class values, and tracks trends over time.

4 Multi-Agent System and Orchestration

Agents are implemented as Python dataclasses (`base.py`) with attributes for name, role, expertise, and execution. They can run sequentially or as **crews** with shared context.
Core agents:

- Strategist – Query planner.
- Crawler – Document discovery and retrieval.
- Summarizer – Structured LLM-based extraction.
- Validator – Code + LLM-based checks.
- KB Writer – Database integration.
- Composer – Report synthesis.
- Experimentalist – Extraction of experimental logic.
- Compute – Normalization and comparative analytics.

Orchestration options:

- Custom Orchestrator (`orchestrator.py`) for sequential workflows.
- CrewAI-based orchestrator (`crewai_impl.py`) for richer collaboration.
- Nextflow for batch-scale reproducibility.

This modular design ensures upgradability: small local models can be swapped for larger LLMs without changing pipeline logic.

5 Connectors and Workflow Integration

Connectors bridge each stage:

- **Search**: SerpAPI, Brave.
- **Access**: Unpaywall.
- **Entity enrichment**: PubChem, UniProt, NCBI Taxonomy.
- **Database**: SQLite backend.
- **Workflow engines**: CLI, Docker/Compose, Nextflow.

Example query "isobutanol production in yeast": Strategist → Gather (Brave + Unpaywall) → Summarizer → Validator (EC + enrichment) → Store (SQLite) → Compute (best yield) → Experimentalist (fermentation insight) → Composer (final PDF).

6 Comparative Context

- **MCF2Chem**: Manually curated and accurate, but slow. Our system automates large-scale extraction.
- **SynBioHub**: Focused on DNA parts and designs, whereas SynBioKB-Agent currently targets biosynthesis, enzyme engineering, and pathway metrics. These are complementary directions.
- **SBKS**: Ontology-first but incomplete. Our approach emphasizes pragmatic extraction and validation, focusing on immediate usability.

7 Future Work

Planned extensions:

- Patent ingestion (USPTO/EPO).
- Table and figure extraction (Camelot, Tabula, Pix2Struct).
- Ontology alignment (ChEBI, KEGG, GO).
- Contradiction detection and trend analysis across reports.
- Graph database backend (Neo4j) for richer network queries.
- Human-in-the-loop validation interface (e.g., Streamlit/Flask) for expert approval.
- Integration with predictive tools (novoStoic2.0, RFdiffusion, BoltzDesign1).
- Conversational querying via RAG agents.
- **Memory and experimental intuition**: reconstructing experimental reasoning chains (why results improved) to build LLM-driven intuition that can guide future design.

8 Conclusion

SynBioKB-Agent is a multi-agent pipeline that automates biosynthesis knowledge extraction, validation, and synthesis. It combines deterministic checks, retrieval-based LLM validation, and specialized agents such as *Experimentalist* and *Compute* to deliver structured, evidence-backed insights. Tested with local LLMs (Ollama, GPT-OSS-20B), it is architected for scalability with larger models. With planned extensions in table/figure extraction, human-in-the-loop validation, and memory-driven intuition, SynBioKB-Agent has the potential to evolve into a comprehensive biosynthesis discovery platform.