Curating the Huntingtin Interactome with Claude Code as an Agentic Extraction Workflow

Rancho bio · sciences | CHDI FOUNDATION | Bridlewood Consulting

Viktoria Andreeva (1), Oleg Stroganov (1), Irina Murasheva (1), Rick Blevins (2), Jeff Aaronson (3)

(1) Rancho BioSciences LLC, San Diego, CA, USA; (2) Bridlewood Consulting LLC, Flower Mound, TX, USA; (3) CHDI Management/CHDI Foundation, Princeton, NJ, USA

Introduction

Abstract

Background HDinHD (Huntington's Disease in High Definition; HDinHD.org) is an open, online data portal designed to support and accelerate HD research. It provides synthesized access to diverse HD experimental datasets, visualization and analysis tools developed by the HD research community, and curated, standardized data processed through reproducible analysis pipelines. The Huntingtin interactome (HINT) within HDinHD (add website link and perhaps one line explaining what it is) captures literature-derived protein–protein interactions relevant to Huntington's disease. Curating and integrating newly published articles into HINT requires substantial manual effort, creating a need for more efficient curation approaches.

Methods We performed a two-phase study. In the pilot phase, Claude Code operated directly as the extraction agent. Curation logic was encoded as Claude Code skills — persistent, reusable instructions that the agent follows during extraction. These skills were iteratively refined: after each extraction round, curator identified errors and edge cases, and skills were updated to correct parsing behavior, improve entity resolution, and handle ambiguous cases. Performance was assessed using precision, recall, and F1 metrics.

In the production phase, the optimized skill set was applied to 25 new publications. For each article, two independent extraction runs were performed, followed by intra-article comparison and manual review. Curators verified extracted interactions, resolved discrepancies, and harmonized metadata prior to integration into HDinHD.

Results In the pilot phase, LLM-assisted extraction showed high recall (~89–94%) and moderate precision (~69%) relative to manual curation. Running two independent extractions and combining their results improved coverage.

In the production phase, 22 of 25 assigned articles were curated (one was excluded by AI due to lack of PPI data, and two were excluded by a curator after data review). That resulted in over 1,100 experiments and approximately 470 interaction records (~80 unique interactors). AI-supported extraction reduced the time required for initial data capture, while manual review ensured consistency with curation standards.

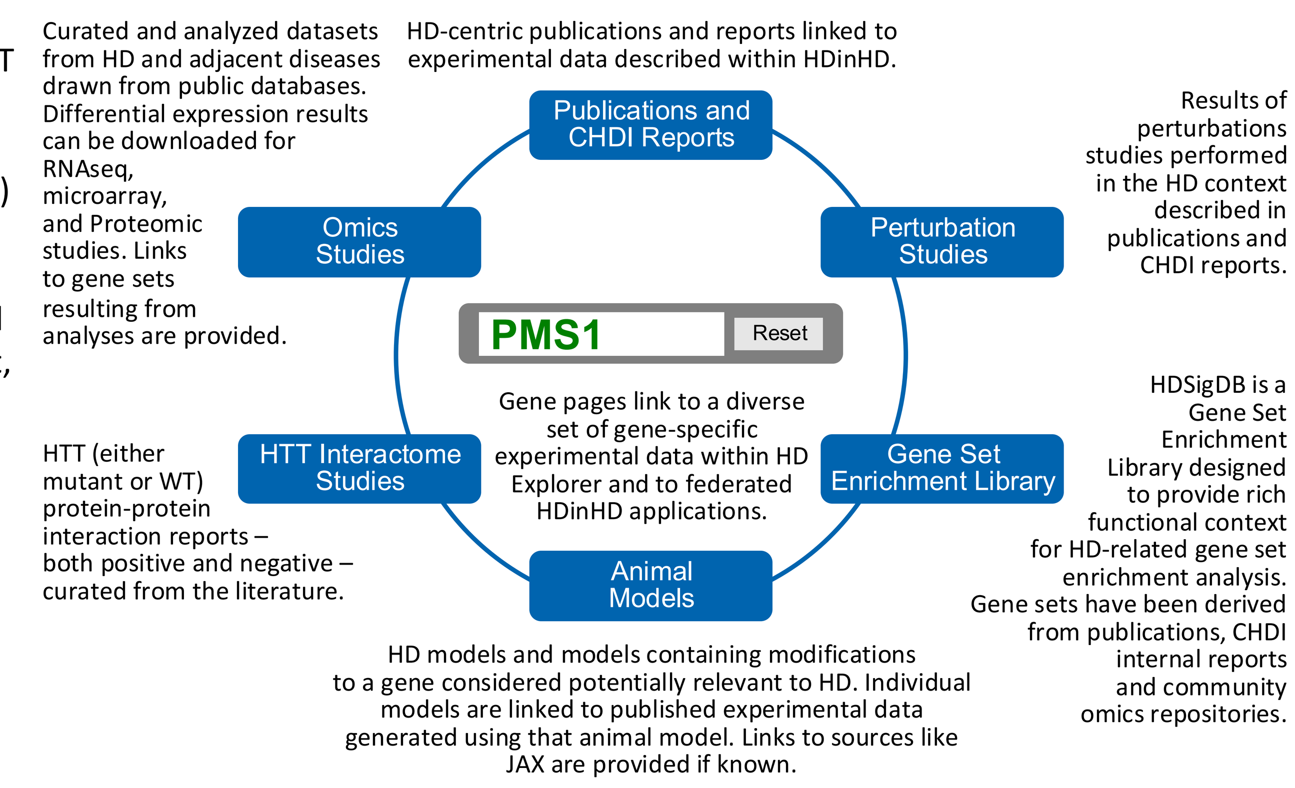
Conclusions Agentic workflows using Claude Code with feedback-refined skills are effective for initial extraction of PPI data but require expert validation. A hybrid workflow combining automated extraction with manual curation improves efficiency while maintaining data quality. This approach can be applied to ongoing updates of HINT and similar resources.

HD Explorer & Huntingtin Interactome

HDinHD provides gene-centric access to interconnected HD data through the HD Explorer. From a single gene search, researchers can pivot across federated data sections:

Omics studies — differential expression from RNAseq, microarray, proteomics
Perturbation studies — results from HD-context perturbation experiments
Animal models — HD-relevant models linked to experimental data
Gene set enrichment library — HD-related gene sets for functional analysis
Publications & CHDI reports — linked to experimental data within HDinHD
Huntingtin **INT**eractome (HINT) — curated protein–protein interactions from literature

HINT captures literature-derived HTT protein–protein interactions (both positive and negative) curated from published articles. It records 70 structured fields per experiment, covering interactors, methods, antibodies, Q-lengths, model systems, and cellular context.



Keeping HINT current requires regular curation of new publications a labor-intensive process that motivated the agentic AI workflow described in this poster.

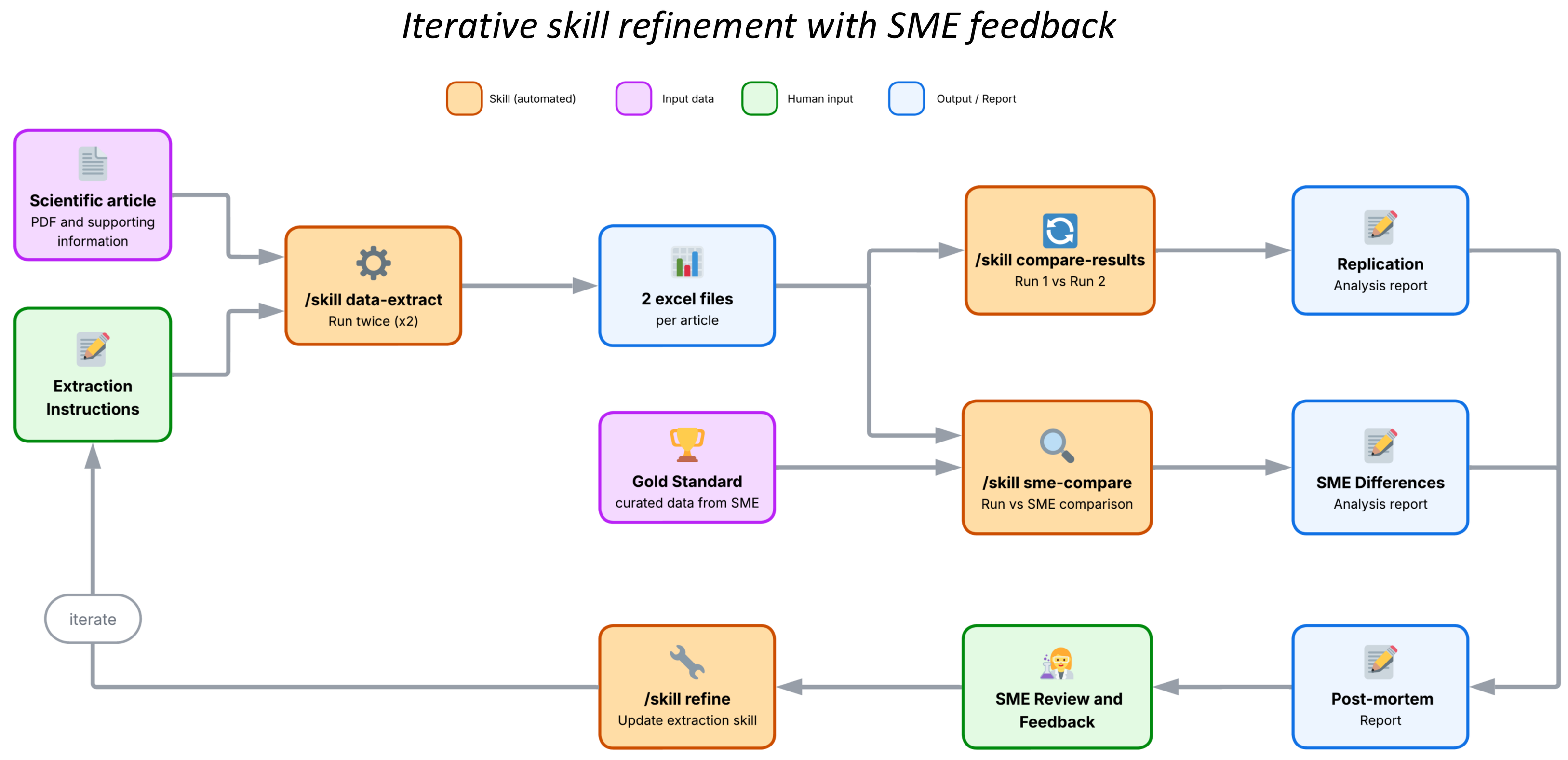
Methods

Table 1. Summary of tested LLM models. *After evaluating three LLM approaches, we selected Claude Code and developed an iterative skill-based extraction workflow*

Approach	Outcome Consistency	Speed	Outcome
Google NotebookLM	✗ Variable format, hallucinations	✗ Batch limit	✗ Rejected
OpenAI gpt-5 pipeline	✗ Frequent timeouts	✗ >15 min/article	✗ Abandoned
Claude Code	✓ Consistent	✓ ~2 min/article	✓ Selected

Figure 1. Skill Development workflow

Iterative skill refinement with SME feedback



Skill-Based Agentic Workflow

Skills are reusable prompt instructions in Claude Code (Claude 4.6 Opus) for specific extraction tasks. Our workflow:

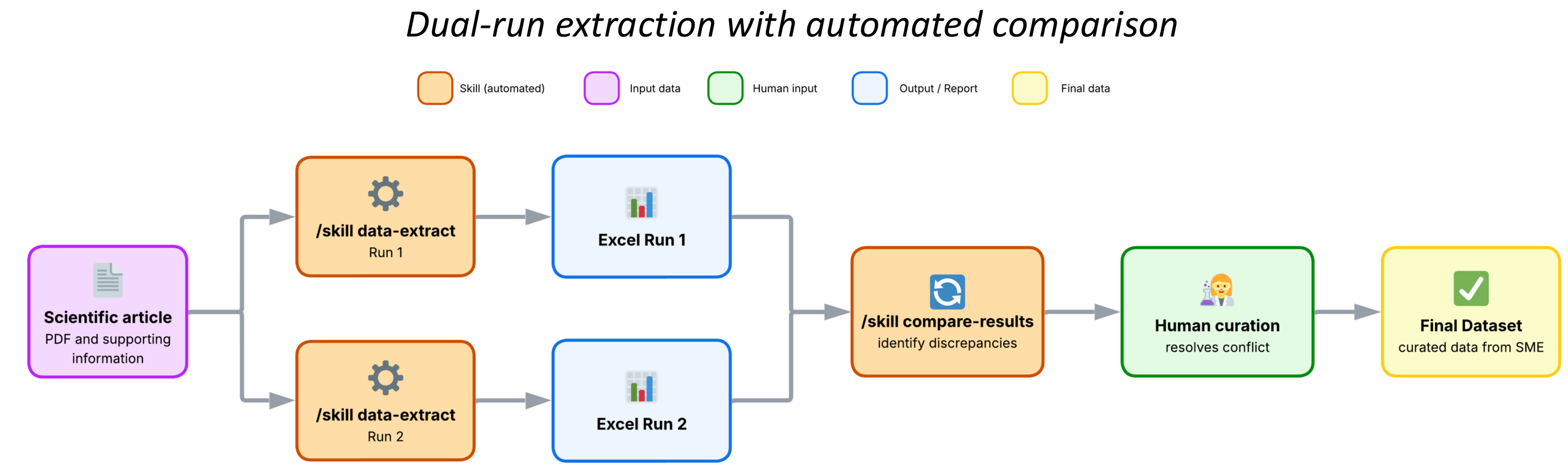
Dual-run extraction: Each article extracted twice independently via /skill data-extract (Figure 1, 2).
Replication check: Runs compared via /skill compare-results to flag inter-run discrepancies.
SME validation: Runs compared to SME gold-standard via /skill sme-compare.
Iterative refinement: SME feedback fed to /skill refine to update prompts and instructions (Figure 1).
Convergence: Cycle repeated until extraction quality stabilized.

Extraction skill architecture:

Extraction is implemented as a five-phase pipeline: **Discovery** (PDF enumeration) → **Survey** (sequential figure-by-figure inventory of all experiments) → **Commitment** (extraction targets locked before extraction begins, preventing post-hoc exclusion bias) → **Extraction** (one record per distinct experiment) → **Validation** (programmatic schema checks). Three mandatory QC gates between phases require explicit pass/fail documentation, catching missed figures, arithmetic errors, and invented exclusion criteria. A structured audit log records all decisions, enabling reproducibility and iterative skill refinement.

Figure 2. Production Curation Workflow

Dual-run extraction with automated comparison



Results

Figure 3. LLM Extraction Performance by Article

F1 Score per article (sorted descending) with mean reference line

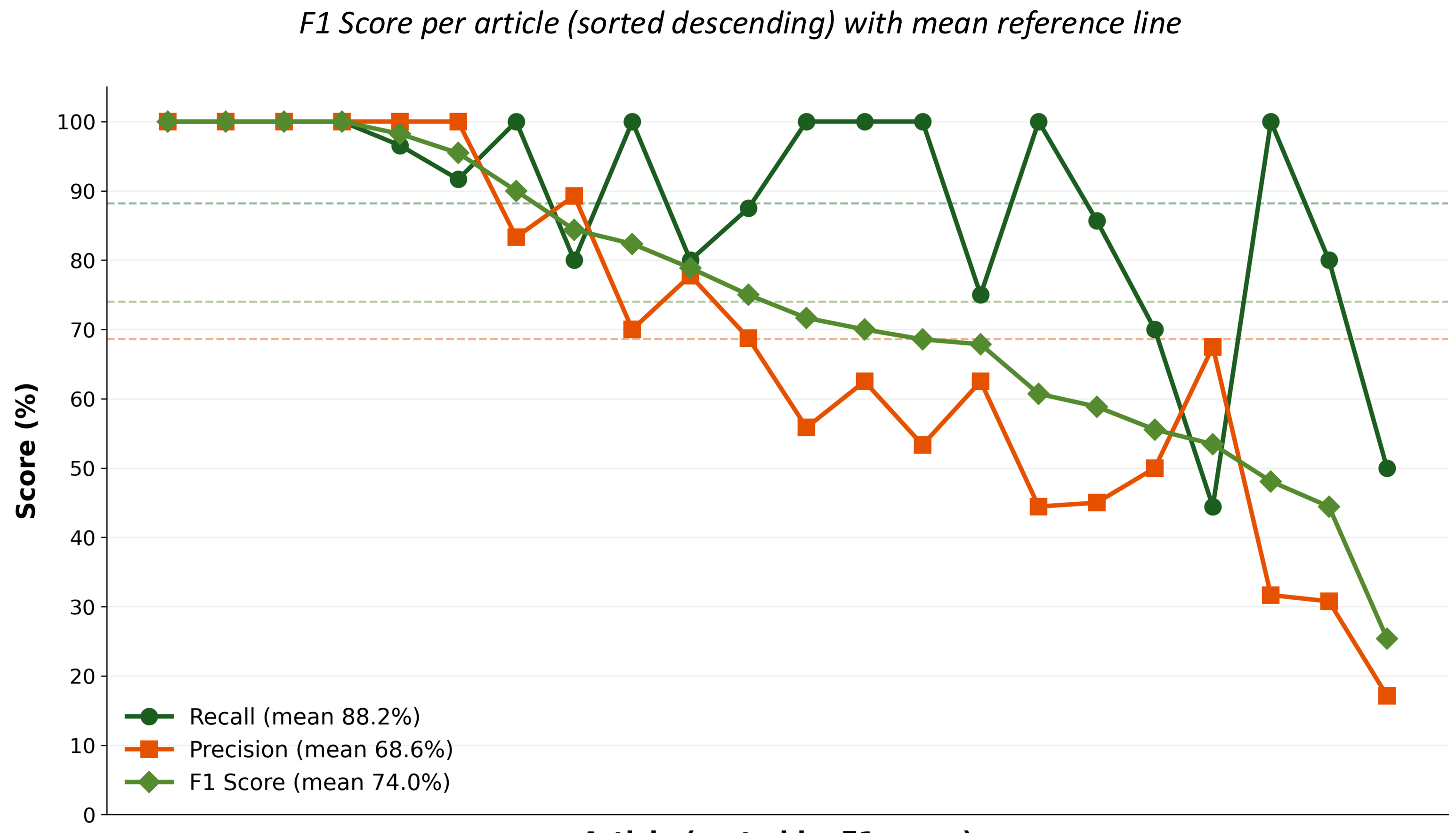
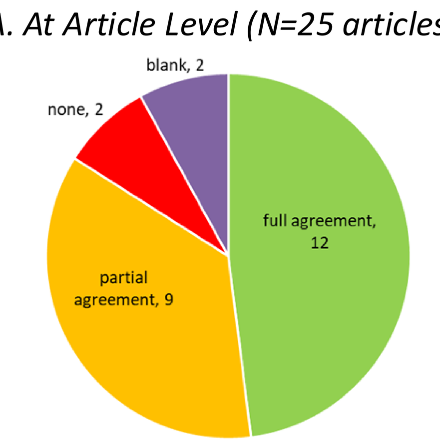
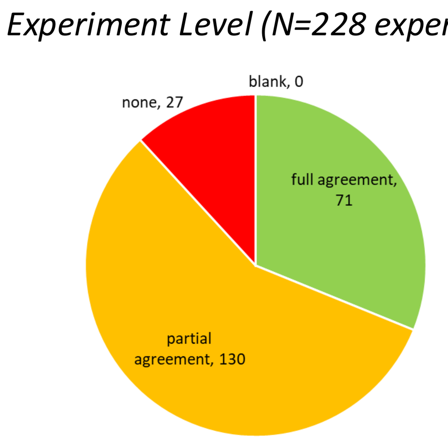


Figure 4. Between-Run Agreement

A. At Article Level (N=25 articles)



B. At Experiment Level (N=228 experiments)



Key Results at a Glance

Based on 122 SME experiments across 25 articles

- **89.5%** Mean Recall — LLM finds ~89% of SME experiments
- **68.6%** Mean Precision — ~69% of LLM extractions are valid
- **0.745** Mean F1 Score — Balanced performance measure
- **93.7%** Union Recall — Combined recall from both runs
- **+4.2%** Dual-Run Lift — Additional recall from second independent run ★ Key Finding
- **$5–8** Cost per Article — Dual-run extraction plus comparison

Conclusions

Claude Code as extraction agent. Operating Claude Code directly as the curation agent enabled rapid iteration. Skills encoded domain knowledge in persistent, human-readable files that accumulated institutional expertise over 29 refinement cycles.

Dual runs surface ambiguity. Two independent extractions per article raised union recall to 93.7% and flagged scientific interpretation differences that helped make curation criteria more explicit.

Expert validation is essential. Mean precision of 68.6% and field-level agreement below 70% for most fields confirm that LLM extraction is a first-pass tool. Curators shifted from extracting data to verifying and refining AI output.

Disagreements improve standards. Many LLM-vs-curator mismatches reflected genuine interpretive differences (method classification, curation scope, direct vs. indirect binding). Resolving these cases forced explicit documentation of curation rules that had previously been implicit.

Cost-effective at scale. At $5–8 per article for dual-run extraction plus comparison, the agentic workflow compares favorably to fully manual curation while producing structured output that curators can review directly.