

# Onco-TTT: An Open-Source Platform for Automated Cancer Hypothesis Generation via Entity Extraction, Knowledge Graphs, and Multi-Source Validation

Ashish Makani

Ashoka University, Sonipat, India

[ashish.makani@ashoka.edu.in](mailto:ashish.makani@ashoka.edu.in)

February 2026 · Preprint (not peer-reviewed)

## Abstract

We present **Onco-TTT**, an open-source web platform that generates testable cancer research hypotheses from free-text queries. The system chains four stages: (1) zero-shot biomedical named entity recognition using GLiNER2 to extract genes, diseases, drugs, pathways, and six additional entity types; (2) knowledge graph construction enriched by the OpenTargets GraphQL API; (3) graph-based activation propagation to rank query-relevant nodes; and (4) structured hypothesis assembly from graph topology. In parallel, the platform retrieves supporting literature from Semantic Scholar, single-cell metadata from the CZI CELLxGENE Census, and runs a six-dimensional validation dashboard that cross-references DepMap gene essentiality, cBioPortal survival data, GTEx tissue expression, ClinicalTrials.gov trial counts, and OpenTargets tractability. Additional modules provide AlphaFold structure analysis with geometric pocket detection, USPTO patent landscape scoring, cell line recommendations, and CRISPR guide RNA design. Onco-TTT is deployed at <https://onco-hypothesis.up.railway.app> with source code at [https://github.com/inventcures/oncology\\_hypothesis\\_generation](https://github.com/inventcures/oncology_hypothesis_generation).

**Transparency statement.** Internal module names in this paper (“ARK” for the KG pipeline, “TTT” for activation propagation) are project codenames, not references to published methods. The activation propagation step is a simplified graph diffusion heuristic, not neural test-time training. Single-cell atlas coordinates are synthetically generated. Validation modules use curated fallback data when live APIs are unavailable.

## 1 Introduction

Cancer research generates vast quantities of genomic, transcriptomic, and clinical data across fragmented databases. A researcher investigating, for example, “*KRAS G12C resistance mechanisms in NSCLC*,” must manually query OpenTargets for gene-disease associations, search Semantic Scholar for relevant literature, check DepMap for gene essentiality, assess clinical trial competition on ClinicalTrials.gov, and evaluate druggability—each through a separate interface with its own query syntax.

Onco-TTT automates this workflow. Given a free-text oncology query, the platform extracts biological entities, constructs an enriched knowledge graph, generates ranked hypotheses, retrieves supporting evidence from multiple public databases, and presents all results in a unified interactive interface.

The system makes three practical contributions:

1. A modular, open-source pipeline that chains zero-shot NER, knowledge graph enrichment, literature search, and multi-source validation into a single query-to-hypothesis workflow.
2. A six-dimensional validation dashboard that provides immediate evidence grounding for generated hypotheses.

3. An interactive web interface with knowledge graph visualization, entity tables, paper retrieval, and feasibility assessment tools including structural analysis and CRISPR protocol generation.

We are transparent about what Onco-TTT is and is not. It is an *integration platform* that combines established methods and public data sources. It does not introduce novel machine learning architectures. Its hypotheses are heuristic, assembled from graph topology rather than from a fine-tuned language model, and have not been systematically benchmarked against ground truth. We describe the system accurately so that researchers can evaluate its utility for their own work.

## 2 System Architecture

Onco-TTT is a two-service web application: a Python/FastAPI backend and a TypeScript/Next.js 14 frontend, deployed on Railway. The backend exposes 12 REST endpoints. The main `/generate` endpoint runs three tasks concurrently via `asyncio.gather`: knowledge graph construction, literature search, and single-cell atlas retrieval. Individual task failures are isolated (`return_exceptions=True`), ensuring partial results are always returned.

Figure 1 shows the end-to-end pipeline architecture.

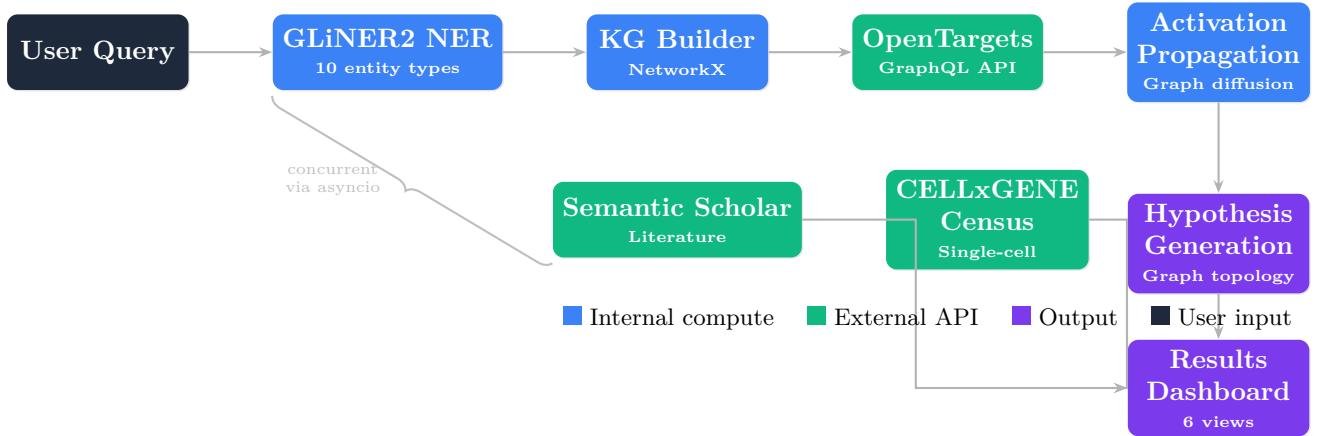


Figure 1: **Onco-TTT pipeline architecture.** A free-text oncology query flows through entity extraction, knowledge graph construction with OpenTargets enrichment, and graph-based activation propagation to produce ranked hypotheses. Literature search and atlas retrieval run concurrently. All results are merged into a six-view interactive dashboard.

## 3 Methods

### 3.1 Entity Extraction

Onco-TTT uses GLiNER2 [1], a generalist NER model based on a bidirectional transformer encoder, to extract biological entities from free-text queries in a zero-shot setting. We define a 10-type oncology entity schema (Table 1) and a 10-type relation schema covering interactions such as *targets*, *drives*, *mutated\_in*, and *synergizes\_with*.

The model (`fastino/gliner2-base-v1`) is loaded once as a thread-safe singleton. Extraction results are cached in an LRU cache (500 entries, 30-minute TTL) keyed by SHA-256 hash of normalized input text. The confidence threshold is 0.4 for entities and 0.3 for research focus classification.

Table 1: **Oncology entity types** extracted by GLiNER2 in zero-shot mode. Types are grouped by biological category.

Category	Entity type	Description
Molecular	Gene	Gene symbols (e.g., KRAS, TP53, EGFR)
	Mutation	Specific variants (e.g., G12C, T790M, V600E)
Clinical	Disease	Cancer types and subtypes
	Drug	Therapeutic agents (e.g., osimertinib, sotorasib)
Biological	Pathway	Signaling pathways (e.g., MAPK, PI3K/AKT)
	Mechanism	Biological processes (e.g., apoptosis, EMT)
Cellular	Cell type	Cell populations (e.g., T cells, macrophages)
	Biomarker	Predictive or prognostic markers
Anatomical	Anatomical site	Tissue/organ locations
	Clinical outcome	Endpoints (e.g., overall survival, response)

### 3.2 Knowledge Graph Construction

Extracted entities and relations are assembled into a directed graph (`networkx.DiGraph`). Each node stores its entity type, extraction confidence, and a deterministic color assignment (Table 1). Each edge stores its relation type, label, weight (from extraction confidence), and a corresponding color.

The graph is then enriched via the OpenTargets Platform GraphQL API [2]. Starting from the highest-confidence gene entity (or a regex fallback `[A-Z][A-Z0-9]{2,7}`), the system:

1. Queries `search(queryString, entityNames: ["target", "disease"])` to resolve the entity to an Ensembl or EFO identifier.
2. Fetches the top-10 associated diseases (for gene seeds) or targets (for disease seeds) via `associatedDiseases` or `associatedTargets`.
3. Adds curated pathway edges for 15 common oncogenes (e.g., KRAS→RAS/MAPK, BRCA1→Homologous Recombination) and cell-type context for 7 genes.

Node visual importance is computed as  $0.4 \times \text{degree\_centrality} + 0.6 \times \text{confidence}$ , mapped to a radius between 22 and 38 pixels. Layout is computed via the Fruchterman–Reingold spring algorithm (80 iterations, seed 42 for reproducibility).

### 3.3 Activation Propagation

After graph construction, a graph diffusion step ranks nodes by query relevance. Nodes whose labels contain query terms receive an initial activation of 1.0. For  $k = 5$  iterations, activation propagates to neighbors weighted by edge weight and a learning rate  $\eta = 0.1$ :

$$a_v^{(t+1)} = a_v^{(t)} + \eta \sum_{u \in \mathcal{N}^-(v)} a_u^{(t)} \cdot w_{uv}$$

where  $\mathcal{N}^-(v)$  denotes in-neighbors of node  $v$  and  $w_{uv}$  is the edge weight. The top-10 nodes by final activation score are returned.

**Transparency note.** This module is internally named “TTT” after the Test-Time Training concept [3], which adapts neural network parameters at inference time. Our implementation is a simplified graph diffusion heuristic—it performs label propagation on a static graph, not gradient-based parameter adaptation. The name reflects an aspirational design direction.

### 3.4 Hypothesis Generation

Hypotheses are assembled deterministically from the knowledge graph topology using four strategies:

1. **Gene–disease driver:** For each gene node, find linked disease nodes; the strongest association (by edge weight) becomes a “Gene X as Driver in Disease Y” hypothesis.
2. **Drug–target:** For each drug node, find gene nodes connected by *targets* or *inhibits* relations.
3. **Mutation–resistance:** For each mutation node, link to related mechanisms or genes to propose resistance hypotheses.
4. **Pathway involvement:** For each pathway node, find connected genes to suggest coordinated signaling hypotheses.

Each hypothesis receives a confidence score derived from edge weights ( $\min(0.95, 0.6 + w \times 0.3)$ ) and a novelty score inversely related to confidence. Results are capped at five hypotheses per query.

### 3.5 Literature Search

The platform queries the Semantic Scholar Academic Graph API [4] for papers matching the user’s query. Up to six papers are returned per request, sorted by citation count. Retrieved fields include title, abstract (with TLDR fallback), authors (truncated to three plus “et al.”), year, citation count, journal, DOI, PubMed ID, and open-access PDF URL. An optional API key (`S2_API_KEY`) provides higher rate limits.

### 3.6 Single-Cell Atlas Integration

Tissue-relevant single-cell metadata is retrieved from the CELLxGENE Census [5] via the `cellxgene_census` Python SDK. The system infers tissue type from the query text (9 tissue mappings, defaulting to “lung”) and fetches up to 300 cells with metadata including cell type and disease annotation.

**Caveat.** The UMAP coordinates displayed in the frontend are *synthetically generated*—cell types receive random cluster centers with Gaussian noise—not projections from a reference atlas embedding. Gene expression values are random placeholders. This module provides metadata exploration, not quantitative single-cell analysis.

### 3.7 Validation Dashboard

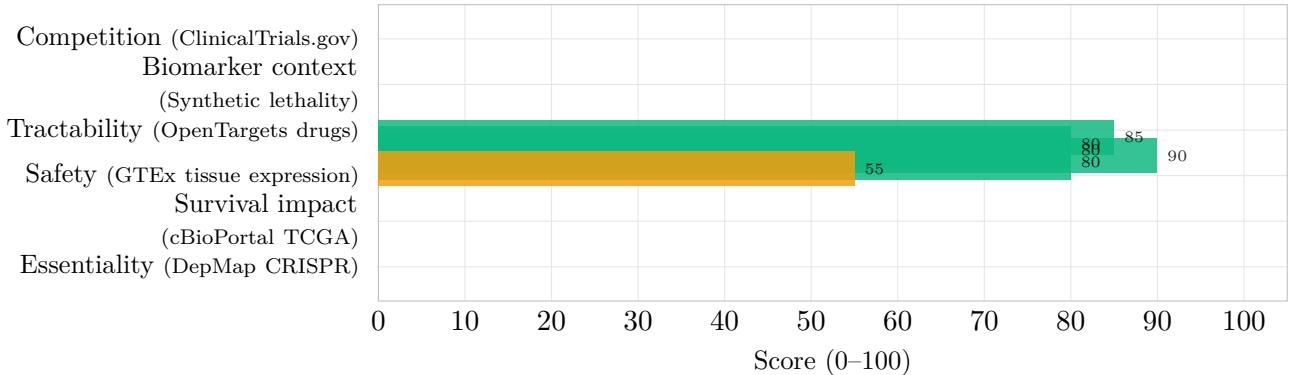
The validation module runs six checks concurrently via `asyncio.gather` (Figure 2), each scoring 0–100. The overall score is the arithmetic mean.

The six checks are:

1. **Essentiality** (DepMap): CRISPR dependency scores (Chronos). Scores below  $-1.0$  indicate strong essentiality; selectivity is the difference between target and pan-cancer lineage scores.
2. **Survival impact** (cBioPortal/TCGA): Hazard ratios from expression-survival correlation.  $HR > 1.5$  with  $p < 0.05$  indicates a poor-prognosis marker.
3. **Safety** (GTEx): Normal tissue expression in 8 vital organs. Genes with  $TPM > 10$  in  $\geq 3$  vital tissues receive a “fail” status.
4. **Tractability** (OpenTargets): Counts of approved drugs, clinical-stage compounds, and preclinical candidates.
5. **Biomarker context**: Curated synthetic lethality partners (e.g.,  $PARP1 \leftrightarrow BRCA1/2$ ,  $PRMT5 \leftrightarrow MTAP$ ).
6. **Competition** (ClinicalTrials.gov): Active trial counts via the v2 API.  $> 10$  active trials or Phase 3 entries indicate a crowded landscape.

When live APIs are unavailable, the system falls back to curated reference data derived from published literature (10 genes with cancer-type-specific dependency scores, 8 genes with hazard ratios, 10 genes with drug data). An LLM-powered rationale synthesis (GPT-4o-mini) integrates

## Six-dimensional hypothesis validation



Example: KRAS in lung adenocarcinoma. Scores use curated reference data when live APIs are unavailable. Color indicates status: *green* = pass, *amber* = caution, *red* = fail, *gray* = unknown.

Figure 2: **Validation dashboard scores.** Each bar represents one of six parallel validation checks with its data source shown in parentheses. The overall score is the arithmetic mean across all checks.

all six check summaries into a 2–3 sentence scientific rationale; a template-based fallback is used when no API key is configured.

### 3.8 Additional Modules

Four additional modules are available for hypothesis feasibility assessment:

**Structural analysis.** Fetches AlphaFold [6] predicted structures via UniProt ID resolution, then performs custom geometric pocket detection using neighbor density at 5/10 Å cutoffs, concavity estimation, hierarchical clustering at 15 Å, and convex hull volume computation. Drug-gability is scored as  $0.3 \times \text{size} + 0.3 \times \text{hydrophobicity} + 0.2 \times \text{confidence} + 0.2 \times \text{enclosure}$  (ideal pocket  $\approx 500 \text{ Å}^3$ ).

**Patent landscape.** Queries the USPTO PatentsView API for patents mentioning the target gene and disease over the past 10 years. A “scooped score” (0–100) combines filing volume (0–40), filing trend (0–30), and competitor count (0–30).

**Cell line recommendations.** Integrates Cellosaurus metadata and Cell Model Passports (Sanger/DepMap) data to recommend cell lines, with match scoring based on data richness, mutation concordance, and known problematic line flagging (e.g., HeLa cross-contamination).

**CRISPR protocol generation.** Retrieves coding sequences from the Ensembl REST API and designs SpCas9 guide RNAs by scanning for NGG PAM sites on both strands. gRNA scoring incorporates GC content (40–70% optimal), position-specific nucleotide weights (simplified Doench et al. 2016 Rule Set 2 [7]), and penalties for poly-T termination signals and restriction enzyme sites.

## 4 Implementation

### 4.1 Backend

The backend consists of 13 Python modules totaling approximately 6,200 lines. Key architectural decisions include:

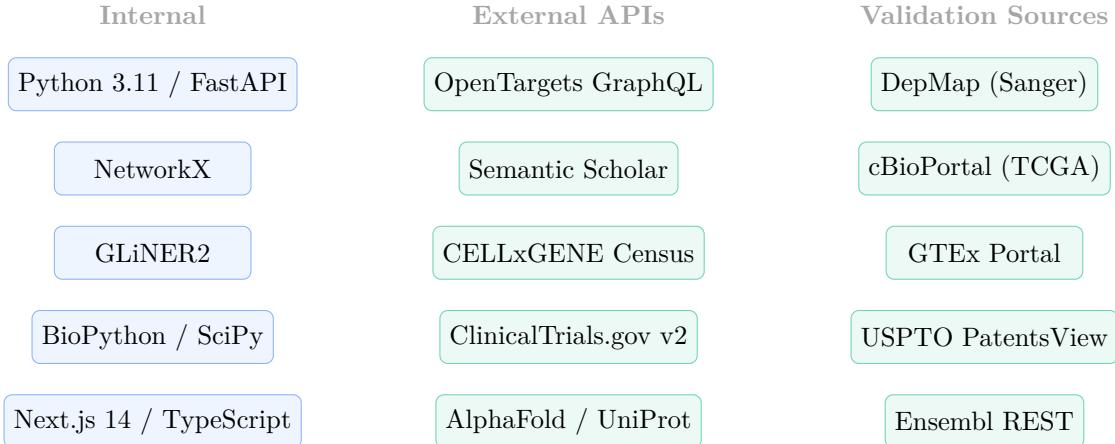


Figure 3: **Technology stack and data sources.** Internal components (left) communicate with 10 external APIs (center, right) via asynchronous HTTP clients (`httpx.AsyncClient`). All external calls have curated fallback data for resilience.

- **Request-scoped graph state:** Each `/generate` request creates a fresh `OncoGraph` instance to prevent concurrent state corruption, while sharing a persistent `httpx` connection pool for OpenTargets.
- **Concurrency:** Knowledge graph construction, literature search, and atlas retrieval run in parallel via `asyncio.gather` with `return_exceptions=True`. Synchronous atlas operations are wrapped in `asyncio.to_thread`.
- **Graceful degradation:** Every external API call has a typed exception handler and curated fallback data. A complete failure of all external APIs still produces a functional (mock-data) response.
- **Input validation:** All text inputs are bounded to 2,000 characters via Pydantic `Field(max_length=2000)`.
- **Semantic caching:** An orchestrator module caches API results with keyword-based fuzzy matching (Jaccard similarity > 0.8, 1-hour TTL, 1,000 entries).

## 4.2 Frontend

The frontend is a single-page application built with Next.js 14 (App Router) and TypeScript. The knowledge graph is rendered as pure inline SVG (no external graph library); node positions are computed server-side. Interactive features include node/edge hover with neighbor highlighting, six switchable view modes, CSV/BibTeX export, and a Mol\*-based 3D protein structure viewer. State is managed via 21 `useState` hooks with `useCallback/useMemo` optimizations.

## 5 Example Walkthrough

We illustrate the system with the query “*role of TP53 in cancer*.”

**Step 1: Entity extraction.** GLiNER2 identifies TP53 (gene, confidence 0.92) and cancer (disease, confidence 0.88) with a *mutated\_in* relation.

**Step 2: KG enrichment.** OpenTargets resolves TP53 to Ensembl ID ENSG00000141510 and returns the top-10 associated diseases including breast carcinoma (score 0.73), colorectal carcinoma (score 0.69), and lung adenocarcinoma (score 0.65). Curated pathway (p53 signaling) and regulatory (MDM2→TP53) edges are added.

**Step 3: Hypothesis generation.** The system generates two hypotheses: “TP53 as Driver in Breast Carcinoma” (confidence 0.82) based on the gene–disease edge weight, and “p53 Pathway Involvement” (confidence 0.70) linking TP53 and MDM2 through the shared pathway node.

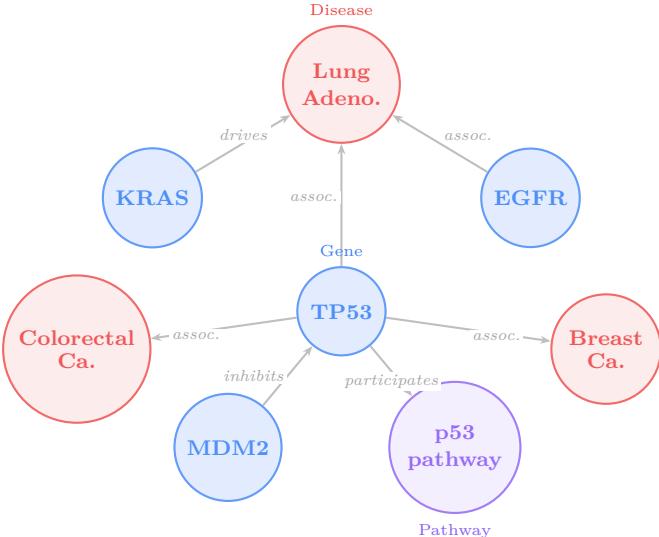


Figure 4: **Example knowledge graph** for the query “role of TP53 in cancer.” Nodes are [genes](#) (blue), [diseases](#) (red), and [pathways](#) (purple). Edge labels show extracted relation types. TP53 is the highest-degree hub, connected to three cancer types, a pathway, and a regulatory gene. This graph was enriched by OpenTargets top-10 disease associations for TP53. Node sizes are proportional to  $0.4 \times \text{degree} + 0.6 \times \text{confidence}$ .

**Step 4: Parallel retrieval.** Semantic Scholar returns six papers sorted by citation count. CELLxGENE Census provides lung tissue cell metadata (300 cells with cell type and disease annotations).

## 6 Limitations

We believe transparency about limitations is essential for a tool intended for scientific use.

1. **No systematic benchmarking.** Hypotheses have not been evaluated against curated ground-truth corpora (e.g., COSMIC, CIViC). We do not report precision, recall, or any quantitative performance metric. The utility of the platform should be evaluated by domain experts on a case-by-case basis.
2. **Heuristic hypothesis generation.** Hypotheses are assembled from graph topology (node degree, edge weight) rather than from a trained generative model. Quality varies with query specificity and the coverage of upstream NER and API enrichment.
3. **Synthetic atlas data.** UMAP coordinates and expression values in the single-cell atlas view are synthetically generated, not computed from real expression matrices. This module provides metadata browsing, not quantitative analysis.
4. **Curated fallback data.** When live APIs (DepMap, GTEx, cBioPortal) are unavailable, the system uses hardcoded reference values for approximately 10 common oncogenes. Users should verify validation results against primary sources.
5. **Simplified structural analysis.** Pocket detection uses custom geometry (neighbor density + hierarchical clustering), not established tools such as fpocket or SiteMap. Druggability scoring weights are ad hoc.
6. **Simplified gRNA scoring.** Off-target estimation is heuristic (10-mer uniqueness ratio), not based on genome alignment. Researchers should validate guide designs with established tools (e.g., CRISPOR, Benchling).
7. **Stateless sessions.** The platform has no user accounts or saved sessions. All queries are independent and cannot be compared or revisited.
8. **Oncology-only scope.** Entity schemas, knowledge graph structure, and curated data are

tailored to cancer biology. The system does not generalize to other disease areas without modification.

## 7 Related Work

Automated hypothesis generation in biomedicine has been explored through literature-based discovery [8], link prediction on biomedical knowledge graphs [9], and LLM-driven scientific reasoning [10]. Onco-TTT differs from these approaches in scope: it is an *integration platform* rather than a novel method, combining zero-shot NER with API-based evidence retrieval into an interactive workflow.

For entity extraction, domain-specific models such as PubMedBERT [11] and BioBERT [12] require supervised fine-tuning per entity type. GLiNER2’s zero-shot schema composition allows rapid addition of new entity types without retraining.

For hypothesis validation, tools such as Open Targets Genetics, DepMap Portal, and cBio-Portal each provide individual evidence dimensions. Onco-TTT’s contribution is aggregating these into a single scored dashboard with programmatic access.

## 8 Future Work

Planned improvements include:

- LLM-powered hypothesis refinement with citation grounding.
- Systematic evaluation against CIViC and COSMIC benchmarks.
- Real single-cell atlas embeddings via scVI or scArches reference mapping.
- Persistent user sessions with hypothesis comparison.
- Replacing the graph diffusion heuristic with actual test-time training on the knowledge graph [3].
- Expansion to additional disease domains.

## 9 Availability

Onco-TTT is open-source under the MIT license.

- **Live demo:** <https://onco-hypothesis.up.railway.app>
- **Source code:** [https://github.com/inventcures/oncology\\_hypothesis\\_generation](https://github.com/inventcures/oncology_hypothesis_generation)
- **Backend API:** <https://backend-production-baa6.up.railway.app/docs>

## Acknowledgments

Onco-TTT builds on the work of the GLiNER team (Zaratiana et al.), the OpenTargets consortium, the Allen Institute for AI (Semantic Scholar), the Chan Zuckerberg Initiative (CEL-LxGENE Census), the Broad Institute (DepMap), Memorial Sloan Kettering (cBioPortal), the GTEx consortium, and the AlphaFold team at DeepMind. We gratefully acknowledge these groups for making their tools and data publicly available.

## References

- [1] U. Zaratiana, N. Golde, and D. Wangmo. GLiNER: Generalist model for named entity recognition using bidirectional transformer. *arXiv preprint arXiv:2311.08526*, 2023.
- [2] D. Ochoa, A. Hercules, B. M. Bento, et al. The next-generation Open Targets Platform: reimaged, redesigned, rebuilt. *Nucleic Acids Research*, 51(D1):D1353–D1359, 2023.

- [3] Y. Sun, X. Li, K. Dalal, et al. Learning to (learn at test time): RNNs with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.
- [4] R. Kinney, C. Anastasiades, R. Authur, et al. The Semantic Scholar Open Data Platform. *arXiv preprint arXiv:2301.10140*, 2023.
- [5] CZI Single-Cell Biology, et al. CZ CELLxGENE Discover: A single-cell data platform for scalable exploration, analysis and modeling of aggregated data. *bioRxiv*, 2023.
- [6] J. Jumper, R. Evans, A. Pritzel, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596:583–589, 2021.
- [7] J. G. Doench, N. Fusi, M. Sullender, et al. Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9. *Nature Biotechnology*, 34:184–191, 2016.
- [8] D. R. Swanson. Fish oil, Raynaud’s syndrome, and undiscovered public knowledge. *Perspectives in Biology and Medicine*, 30(1):7–18, 1986.
- [9] S. Bonner, I. P. Barrett, C. Ye, et al. A review of biomedical datasets relating to drug discovery: A knowledge graph perspective. *Briefings in Bioinformatics*, 23(6):bbac404, 2022.
- [10] H. Wang, J. P. Gonzalez-Brenes, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620:47–60, 2023.
- [11] Y. Gu, R. Tinn, H. Cheng, et al. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23, 2021.
- [12] J. Lee, W. Yoon, S. Kim, et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.