Bioinformatics, 2026, **42**(5), btag166
https://doi.org/10.1093/bioinformatics/btag166
Published: 29 April 2026
Original Paper | Data and text mining

OXFORD

Metappuccino: large language model-driven reconstruction of sequence read archive metadata for cancer research

Fiona Hak[1,*], Camille Marchet[2], Daniel Gautheret[1,*], Mélina Gallopin[1,*]

[1]Université Paris-Saclay, CEA, CNRS, Institute for Integrative Biology of the Cell (I2BC), Gif-sur-Yvette 91190, France
[2]Université de Lille, CNRS, CRIStAL UMR 9189, Lille F-59000, France
*Corresponding authors. Fiona Hak, Universite Paris-Saclay, CEA, CNRS, Institute for Integrative Biology of the Cell (I2BC), Gif-sur-Yvette 91190, France. E-mail: fiona.hak@i2bc.paris-saclay.fr; Daniel Gautheret, Universite Paris-Saclay, CEA, CNRS, Institute for Integrative Biology of the Cell (I2BC), Gif-sur-Yvette 91190, France. E-mail: daniel.gautheret@universite-paris-saclay.fr; Mélina Gallopin, Universite Paris-Saclay, CEA, CNRS, Institute for Integrative Biology of the Cell (I2BC), Gif-sur-Yvette 91190, France. E-mail: melina.gallopin@i2bc.paris-saclay.fr.
Associate Editor: Jonathan Wren

Abstract

Motivation: **High-throughput RNA sequencing has significantly advanced transcriptomic profiling in oncology. Millions of RNA-seq datasets have accumulated in public databases such as the Sequence Read Archive (SRA). However, fragmented, ambiguous, or missing metadata can severely limit accurate cohort selection, introduce bias, and delay discoveries.**

Results: **To address these issues, we introduce 'Metappuccino', a hybrid metadata enrichment tool built on Mistral-7B-Instruct and specialized via low-rank adaptation (LoRA). Metappuccino reconstructs 19 metadata classes (e.g. organ, disease, cell type) by combining deterministic extraction/normalization with model-based completion: 4 submission-mandatory fields are read directly from SRA/API records, while the remaining 15 classes are obtained through validated rule-based extraction when explicitly supported by the context and otherwise predicted by the LoRA-specialized model when information is missing or ambiguous. To promote robust, context-aware inference rather than memorization, we designed training and data partitioning to minimize leakage and preserve generalization. When applicable, predicted values are mapped to standardized ontologies to ensure consistent, interoperable annotations. Across our benchmarks, Metappuccino substantially improves accuracy over the base model, matches or exceeds recent larger open-source LLMs, and reduces inference time by up to two-fold relative to these baselines. By enriching under-annotated public RNA-seq records, Metappuccino increases the usability of SRA datasets for large-scale reuse, with applications that extend beyond oncology transcriptomics.**

Availability and implementation: **Metappuccino source code is available on: github.com/chumphati/Metappuccino. The fine-tuned LLM, MetappuccinoLLModel, is available on: huggingface.co/chumphati/MetappuccinoLLModel. Both repositories are released under Apache-2.0 license.**

1 Introduction

High-throughput RNA sequencing (RNA-seq) has drastically improved our ability to characterize transcriptomic landscapes in various biological contexts, including oncology. At the same time, public repositories such as Sequence Read Archive (SRA) have accumulated massive datasets comprising millions of RNA-seq experiments, including >800 000 human RNA-seq samples produced with sufficient depth and quality for full transcriptome analysis between 2012 and 2025 (NCBI SRA Portal) (Leinonen et al. 2011). These data are highly valuable for the discovery of new clinical biomarkers and therapeutic targets and for inferring transcriptomic signatures, i.e. patterns of gene or transcript expression that shed light on oncogenic mechanisms.

In spite of their great potential value, these databases remain largely underexploited. A major bottleneck is the lack of high-quality, available sample-associated metadata. Several issues have been identified while analyzing this metadata. (i) Sample metadata are retrievable through APIs (such as NCBI) with multiple fields that can be selectively queried. Still, some fields are absent from the API because they were not submitted, despite being present in the associated publication or stored in custom, nonstandard fields. (ii) Many fields are very often filled with useless values such as "unknown/missing/etc." or are left empty. We found that 71.6% of the 192 fields across all run accessions

Received: 5 November 2025. **Revised:** 10 February 2026. **Accepted:** 12 March 2026

in the NCBI API are left empty (Fig. S1, available as supplementary data at *Bioinformatics* online). (iii) The fields can also be filled incorrectly: e.g. age can be filled as a cell type. (iv) Finally, even when annotations are correct, they may be ambiguous because there are no normalized rules to submit some fields such as cell type or cell line. This can be very problematic for downstream analysis when, e.g. trying to assemble a cohort. This can bias the extraction to always get the same well-annotated sample without exploiting the full richness of the databases (Hippen and Greene 2021, Rajesh et al. 2021, Caliskan et al. 2023). In this context, working with the current SRA metadata is very challenging and requires a substantial amount of human effort.

Several computational tools and methods have been developed to address these issues. Access and aggregation tools such as SRAdb and Grabseqs make SRA data easier to query and download but leave metadata untouched and inconsistent (Zhu et al. 2013, Taylor et al. 2020). MetaSRA normalizes BioSample attributes by mapping them to ontologies with rule-based methods. It still remains sensitive to heterogeneity and ambiguities in metadata, which still prevents fully exploiting SRA's potential (Bernstein et al. 2017). Newer systems move beyond field matching: SRA Database Navigator couples Natural Language Processing (NLP, automatic analysis of free text descriptions), Named Entity Recognition (NER, identifying and standardizing entities), and Network Analysis (graph-based linking and propagation) to reconstruct and enrich metadata for semantic clustering (grouping samples by meaning rather than keywords) (Guimarães et al. 2025). Txt2onto 2.0 uses large language model embeddings at prediction time to match unknown wording to known training terms, then applies Term Frequency–Inverse Document Frequency features (TF-IDF, which weights a term higher when it is frequent in a document but rare across the document collection, highlighting discriminative words) to assign standardized labels (tissue, disease, etc.). It is more robust to new phrasing but still uses a bag-of-words, ignoring context and limited to training labels (Yuan et al. 2024). PredictMEE uses NER on unstructured BioSample text (e.g. titles, sample names) to infer missing attributes across 11 classes, achieving high accuracy, e.g. Genus/Species (94%) and Condition/Disease (95%) (Klie et al. 2021). SGMC targets the geographic information by combining cloud queries, web scraping, and ChatGPT to standardize sequencing-institution and country fields across 2.3M SRA accessions, reaching 94.8% concordance for institutions and 93.1% for countries versus manual curation (Zhao et al. 2023). ChIP-GPT fine-tunes a LLaMA model to answer ChIP-seq–specific questions, handling noisy fields and achieving 90%–94% accuracy on targets/cell lines despite a narrow training scope (Cinquin 2024). Most recently, Ikeda et al. (2025) showed that LLM-assisted extraction plus ontology mapping can outperform rule-based pipelines. From BioSample text, the method identifies cell line names and maps them to the cell line reference database Cellosaurus (Bairoch 2018), surpassing MetaSRA both in accuracy and coverage (Ikeda et al. 2025). However, in all those methods, common gaps remain: (i) the use of small and not diverse datasets for LLM training that could lead to performance overestimation, (ii) the reliance on constrained ontologies or rules/heuristics that limits the full exploitation of SRA's potential, (iii) the sensitivity to noisy text/lack of context understanding, (iv) the presence of manual or semi-automated steps, and (v) the limits in the quantity of extracted information with often a maximum of 1–3 classes (e.g. cell line). Most approaches also overprivilege the metadata stored in BioSample fields while underusing the study/BioProject context. This leaves out a lot of important information that is not stored in BioSample but that is still available elsewhere in SRA.

Building on those previous efforts, Metappuccino is a hybrid and modular pipeline built around a fine-tuned Mistral model that goes beyond BioSample. It reads the full SRA record space by combining API fields, study and BioProject text, study description, and optional user notes. The model both extracts values and infers missing ones, understanding medical context rather than relying only on rules. To limit typical LLM issues such as hallucinations, it pairs the model with classic rule-based NLP similar to MetaSRA, then uses biomedical ontologies to harmonize terms and cut ambiguity. Unlike tools limited to a few attributes, Metappuccino now extracts 19 classes, including standard SRA fields and additional ones manually defined and useful for downstream cancer transcriptomics. These classes are not fixed: the training setup and pipeline design allow new ones to be added on request. Metappuccino can be applied to any kind of data beyond transcriptomics in oncology, as long as outputs from irrelevant classes are ignored.

2 Materials and methods

2.1 Metappuccino pipeline

The Metappuccino pipeline v.1.0.0 can extract 19 different types of information (discrete or open vocabulary) from a context, which we will refer to as classes (Table 1).

2.1.1 SRA extraction

Starting from a list of SRA-run accession numbers, Metappucino retrieves for each run an SRA XML file via the NCBI E-utilities API (1. XML; 2). This run-level record is structured into four blocks (<STUDY>, <SAMPLE>, <EXPERIMENT>, <RUN>) describing, respectively, the project or study, the sample, the experimental setup, and the sequencing run. The XML file is parsed into (i) reading study metadata from <STUDY> (title, first public date, project identifiers, abstract, etc.) (Fig. 1. Study information), (ii) reading sample metadata from <SAMPLE> (title, free-text description, age, biological source, etc.) (Fig. 1. Special fields), and (iii) getting a BioSample accession (SAMN.) from cross-references XREF_LINK and EXTERNAL_ID (if absent, a regular-expression search within the SRA XML is attempted). Using the resolved accession, the corresponding BioSample XML file is downloaded (Fig. 1. Biosample attributes). In parallel, an SRA API call returns values for fields that may be relevant to the targeted classes (details in Material, available as supplementary data at *Bioinformatics* online, section 'Extracted fields from the NCBI API during pipeline download'). All retrieved items (SRA XML, BioSample XML, and SRA API fields) are finally merged into a tab-separated file with one row per run. Metappuccino also accepts an optional tabular file with a required run accession column and any number of user-supplied text-metadata columns, which is useful if a user has access to additional data to complement SRA information.

2.1.2 Preprocessing

The preprocessing stage has two main steps (Fig. 2):

Table 1 Metappuccino extracted metadata fields.

Study accession number:	Unique study ID.
Instrument Platform:	Sequencing technology family (e.g. Illumina).
Number of base pairs:	Total number of sequenced bases.
Library strategy:	Assay type (e.g. RNA-seq).
Library selection:	Library enrichment strategy. 'Discrete values': polyA, inverse rRNA, hybrid selection, small RNA, other.
Sequencing source:	Sampling granularity of RNA-seq. 'Discrete values': bulk, single cell, spatial.
Biopsy site:	Organ, tissue, body part, or fluid from which the sample was extracted.
Biopsy type:	Clinical context of the sample 'if cancer related'. 'Discrete values': primary, metastasis, blood.
Cell line:	Standardized name of cultured/immortalized line.
Cell type:	Cell identity, that share morphological or phenotypical features (e.g. neuron, fibroblast, CD8 T cell, melanocyte).
Organ:	Organ affected (distinct from the sampling site in biopsy site).
Disease:	Specific associated condition (or healthy).
Treatment:	Intervention applied (drug, procedure, implant, event, etc.), including planned pre-treatments.
Treatment time:	Timing relative to the applied intervention (e.g. pre, on, post, or a duration).
Response:	Clinical or experimental outcome. 'Discrete values': no treatment, unknown, stable, progressive, success.
Age:	Donor age if human.
Sex:	Donor sex if human.
Ethnicity:	Donor ethnicity if human.
Is cancer:	Whether the disease is cancer-related. 'Discrete values': True, False.

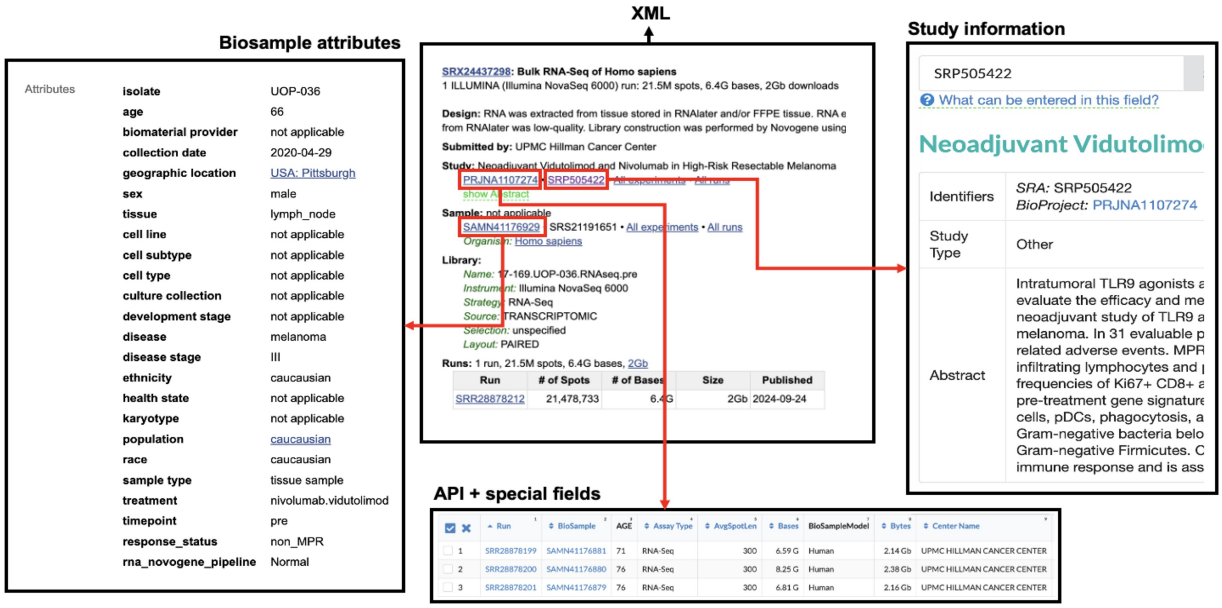

Figure 1 Example of SRA inputs collected by Metappuccino for SRR28878212. BioSample contains sample-centric information filled by the submitting team. Study information provides project-level metadata shared by all samples. The SRA XML aggregates and links study, sample, experiment, and run information, and it may include author free-text fields not exposed in some views/APIs. Some information appears in multiple sources, while other are specific to a single source.

- Turning the raw SRA and BioSample downloads into a clean, compact paragraph-style summary for each run. First, a cleaning part converts the XML files into paragraphs. Then, the text is summarized only if needed (over 2000 tokens). In that case, obvious identifiers and boilerplate are removed, duplicates are deleted, and rare informative phrases and words are kept. This part is important in order to avoid giving the LLM too long contexts, which reduces performance and increases hallucinations, but the variety of the biological context has to be maintained.
- Extracting values for the 19 classes from the raw downloads whenever this is supported by our rule-based NLP methods: (i) When a class name appears as an explicit XML tag or a BioSample attribute, the value is directly extracted, including common aliases. (ii) When a class exists only in free text, it is extracted by phrase matching using a dictionary built from

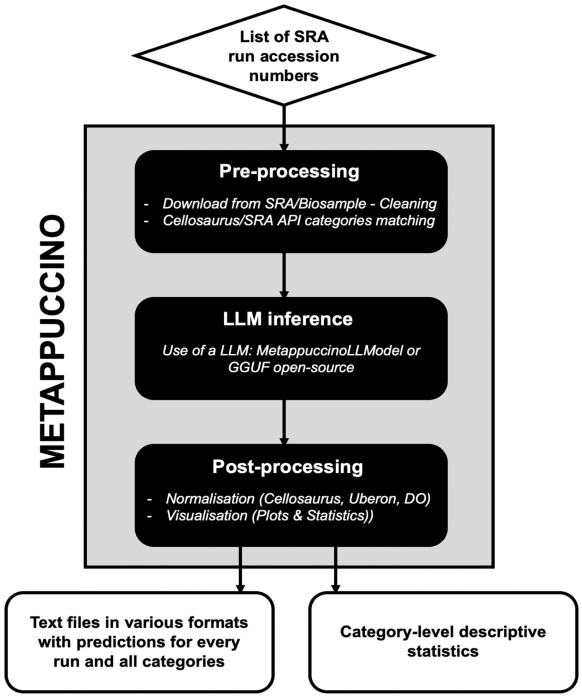

Figure 2 Metappuccino Workflow. Metappuccino first downloads and preprocesses SRA run records to extract as many class values as possible with rule-based NLP. LLM inference then completes any missing fields. Post-processing uses medical dictionaries and ontologies to infer related fields from new predictions and to normalize terminology. Outputs are tables (various formats) with per-run, per-class values, plus descriptive statistics.

Cellosaurus (Bairoch 2018), where controlled names and synonyms are lowercased, purged of generic terms, and compiled into a spaCy *PhraseMatcher* data structure (the text is parsed once, exact spans are returned, candidates are deduplicated while preserving order, and a lightweight regex fallback catches missed alphanumeric variants). (iii) When data is missing for library_selection, sequencing_source, and biopsy_type, simple heuristics try to infer them when their values are explicitly stated or via a list of their synonyms. (iv) When a cell line is found by one of the previous methods, the Cellosaurus database is used (Bairoch 2018), first to normalize the extracted term if it is a synonym to a canonical cell line name and then to fill the following classes based on its knowledge of the cell line: biopsy_site, cell_line, cell_type, organ, disease, age, sex, and ethnicity. Finally, in addition to Cellosaurus, a normalization step maps diseases to Disease Ontology names (Kibbe et al. 2015) and identifiers (diseases database) and organs or biopsy sites to Uberon names and identifiers (species anatomy database) (Mungall et al. 2012). Values obtained by steps (1)–(3) are validated before acceptance as follows: (i) 'type and range' checks (e.g. age must be numeric within plausible bounds, sex in {male, female, … }, etc.), (ii) 'ontology' checks (disease, organ, cell_type, and biopsy_site must map to valid terms in Disease Ontology, Cellosaurus, or Uberon), and (iii) 'cross-source consistency' across SRA XML, BioSample attributes, and API tables, with agreements preferred and disagreements flagged for downstream LLM review. Any value failing a check is rejected from the curated table. All extracted and normalized values are merged in a temporary file with one column per class and one run per row. Those values will not be overridden by the following LLM completion, as they are considered true.

2.1.3 LLM inference

LLM inference uses the prepared summary and the ambiguity notes from preprocessing to predict 15 classes out of 19. This is because four classes (study accession number, instrument platform, number of base pairs and library strategy) are always populated in their corresponding NCBI API fields. Two interchangeable LLM backends are supported: (i) **Hugging Face Transformers + PEFT (LoRA)**, our MetappuccinoLLModel mode, where Transformers is the PyTorch stack and PEFT (Parameter-Efficient Fine-Tuning) updates only small add-on modules instead of all weights (Hu et al. 2021). LoRA (Low-Rank Adaptation) inserts trainable low-rank matrices into attention/MLP projections (Hu et al. 2021). We provide one LoRA adapter per class (see 'Data availability') and activate the relevant adapter at inference on the base model `mistralai/Mistral-7B-Instruct-v0.3`. (ii) llama.cpp (github.com/ggml-org/llama.cpp), which runs local GGUF models (a quantized, CPU/GPU-friendly format downloadable from Hugging Face) without PEFT, letting the user swap in any compatible model, including non-open-source ones. This mode exists to give users the final say on the model, allowing easy model switching and enabling fully local deployment.

For class extraction, the prompt was developed empirically: different wording and tags were tested, the outputs were checked, and the structure that gave clear, direct answers was kept (Table 2). For each run and class, the run accession '(a)', the summary and the ambiguity notes '(b)', the class definition '(c)', other information '(d)' are included in the prompt template. The system outputs one JSON file per run containing all predicted class values and two confidence scores, the mean negative log-likelihood (NLL) and the perplexity (PPL) (Fig. S15, available as supplementary data at *Bioinformatics* online).

2.1.4 Post-processing

The post-processing stage converts the curated metadata extracted at preprocessing into a single table per run, using rule-based NLP and the LLM predictions, normalized with Cellosaurus, Disease Ontology, and Uberon. Values extracted at preprocessing are locked, and only missing fields are complemented with the LLM inferences. Additional cell lines detected during LLM inference are normalized with Cellosaurus and can propagate consistent attributes (e.g. disease, cell type, biopsy site). Free-text organ/disease/biopsy terms are normalized to preferred names and mapped to stable codes, with light cleaning (synonyms, formatting). The final dataset is exported into multiple formats (CSV, TSV, XLSX, Parquet, JSON, and Feather), alongside confidence tables, colored HTML views, and summary graphics to support quick quality checks and downstream analysis.

Table 2 Dynamic prompt template used for LLM inference.[a]

(a) Run accession: {**run**}
(b) Context summary: {**summary**}
(c) Class definition line: {- {**class name**}: {**definition**}}
Classes and definitions (one used at runtime):
library_selection: based on the given context, determine the library selection fixed class by searching for specific keywords or synonyms that match one of the five strict classes: "polyA," "inverse rRNA," "hybrid selection," "small RNA," or "other." Assign "polyA" if the context contains any of the following terms or similar meaning that can be inferred: "PolyA," "poly.A," "oligo.dT," "oligodT," "truseq.mrna," "truseq.stranded.mrna," "truseq.standard.mrna," "smarter.mRNA," "stran ded.mRNA." Assign "inverse rRNA" if the context mentions depletion of ribosomal RNA with any of these terms or similar meaning that can be inferred: "ribominus," "ribodep," "ribozero," "ribo.zero," "riboerase," "ribogone," "ribocop," "ribo-dep," "ribo-mi," "ribo minus," "depleted ribosom," "remove ribosom," "TruSeq.Stranded.Total," "TruSeq.Total," "SMARTer.Stranded.Total," "SMARTer.Total." Assign "hybrid selection" if the context refers to hybrid capture or exon selection using any of these terms or similar meaning that can be inferred: "Hybrid.Selection," "Exon.capture," "Exome.capture," "RNA.Exome," "geoMX." Assign "small RNA" if the context refers to small RNA isolation with keywords such as "TruSeq.Small," "size.fraction" or similar meaning that can be inferred. Assign "other" if none of the above terms are found. Return only the exact class name.
sequencing_source: one of: "spatial," "bulk," "single cell." Spatial preserves in-tissue localization; bulk sequences a mixture of cells; single cell profiles individual cells. biopsy_site: organ, body part or fluid where tissue was sampled (same as organ if not cancer). Mention xenograft here if applicable. Not the disease; the sampled tissue. If possible, deduce from the cell line origin.
biopsy_type: "metastasis" if cancer and metastasis mentioned, or "blood" if blood-related and no metastasis; otherwise "primary." Do not assign metastasis or blood unless explicitly stated. Always choose one of the three.
cell_line: standardized laboratory cell line name; otherwise "Primary tissue" if not a cell line.
cell_type: the cellular type (e.g. neuron, fibroblast, CD8 T cell, NK cell, melanocyte, dendritic cell). If not provided, deduce from context; otherwise "primary tissue". organ: organ studied or affected (different from biopsy_site if applicable). Not the disease. disease: associated disease (be specific) or "healthy" (watch for cues like "adjacent," "normal"). treatment: applied intervention (drug, molecule, surgery, implant, event). Infer if implicit. Do not restate the disease. Include planned pre-treatments.
treatment_time: time or phase relative to treatment; if quantitative, indicate pre/on/post. response: reaction to treatment: "no treatment," "unknown," "stable," "progressive," "success". age: human donor age (numeric range/exact or qualitative like child/teen/adult/senior). If unknown, return "unknown". sex: human donor sex; "unknown" if not inferable.
ethnicity: human donor ethnicity (e.g. caucasian, black, asian); "unknown" if not inferable.
is_cancer: "True" if the disease is cancer-related, "False" otherwise.
(d) Instruction block:
- Extract information from the summary if possible.
- If one value is impossible to extract, even by deducing it, return "unknown."

Be careful: sometimes the information concerns several samples from the same study. Distinguish what applies to the current run so everything remains consistent. For each class, several answers can be possible; cite them all with a "," or ";" separator.

[a] (a) run accession, (b) context summary, (c) the class definition line (one picked at runtime), and (d) the instruction block are injected for each run and class for individual inference.

2.2 MetappuccinoLLModel

A task-specialized fine-tuned model is provided with Metappuccino: MetappuccinoLLModel, trained to outperform available open-source LLMs on our task.

2.2.1 Fine-tuning

`Mistral-7B-Instruct-v0.3` is our backbone, selected for its fastest native inference among the open-source models we tested (see Section 2.3.1). We freeze the base weights and train lightweight LoRA adapters (Hu et al. 2021), one per class, to keep deployment fast, modular, and memory-efficient. Training uses the same prompt format as inference, runs at 16-bit mixed precision on two A6000 GPUs (80 GB VRAM), and is designed to teach the model where to extract each class's information within different context structures, maximizing use of context while reducing hallucinations and extraction errors while producing concise outputs. All hyperparameters and settings are documented in the Hugging Face configs of the released adapters (https://huggingface.co/chumphati/MetappuccinoLLModel). Further details of the training method are available in the Material, available as supplementary data at *Bioinformatics* online, section 'MetappuccinoLLModel: in-depth methods, training procedure and final adapters selection'.

2.2.2 Data for training and evaluation

Because well-annotated SRA data are limited, especially for our target classes, we created large-scale synthetic texts based on Metappuccino summaries that closely matched real inputs. We found that the best results per class used ∼600–3000 training samples and ∼200–500 validation samples, with balanced classes (including unknown). For open-vocabulary classes, values are strictly different across training, validation, and test sets to promote generalization, not memorization. Models are evaluated with three test sets per class: **ID (identical) with 400 samples similar to training data**, **OOD (out of distribution) with 400 samples very different from training data**, and **MID (middle) with 1200 samples that mix both**. Full details on data

generation and splitting, as well as data leakage evaluation, are given in Material, available as supplementary data at *Bioinformatics* online, section 'MetappuccinoLLModel: in-depth methods, data generation, data leakage'. To complement synthetic data, we randomly selected 400 human RNA-seq SRA records and 100 melanoma records, also randomly selected among melanoma-annotated samples (see Fig. S5, available as supplementary data at *Bioinformatics* online), that we manually annotated for all target classes. These annotations are available in the Data section of the GitHub repository. The 100 melanoma records were annotated twice, once using only SRA information to fairly evaluate the LLM given the same context it sees, and once with added Cellosaurus knowledge for the 'Study case with Metappuccino, 100 melanoma samples', which shows how pre- and postprocessing can improve performance alongside LLM inference.

2.3 Benchmarking

The performance of Metappuccino and MetappuccinoLLModel was assessed against commonly used open-source LLMs.

2.3.1 Baseline-LLM selection

The following instruction-tuned open-weight models were chosen as reference baselines for this study: Mistral-7B-Instruct-v0.3 (https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3) (Jiang et al. 2023, Mistral-AI 2024), Llama-3.1-8B-Instruct (https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md) (AI@Meta 2024), Llama-3.1-70B-Instruct (AI@Meta 2024), DeepSeek-V2-Instruct (https://huggingface.co/deepseek-ai/DeepSeek-Coder-V2-Instruct) (DeepSeek-AI 2024, Liu et al. 2024), Gemma-3n-E4B-Instruct (https://huggingface.co/google/gemma-3n-E4B) (Google 2025), and Qwen3-30B-A3B-Instruct (https://huggingface.co/Qwen/Qwen3-30B-A3B) (Qwen-Team 2025). These models span a spectrum from small, efficient LLM variants suited to modest hardware (Mistral-7B, Llama-3.1-8B, Gemma-3n-E4B) to larger, higher-capacity systems for higher accuracy and longer-context reasoning (Llama-3.1-70B, Qwen3-30B, DeepSeek-V2). "XB" designates the number of parameters of the model. All are instruction-tuned to follow task prompts reliably. They were all downloaded via Hugging Face on their official repositories. They were selected to evaluate the most widely used open-source originals. The latest versions were tested, and only those that produced responses consistent with our prompt structure were retained. Q8 quantization was used for Llama-3.1-70B to meet memory and throughput constraints (2 GPUs = 48 GB × 2 VRAM, NVIDIA-A6000). All other models were loaded unquantized in floating-point: FP16 by default. The same prompt and datasets were used for inference with all the models to produce comparable results.

2.3.2 Performance evaluation

A hybrid evaluation was implemented to handle both classes requiring: (i) context-tolerant matches (e.g. organ: 'oral cavity' versus 'mouth' = True) and (ii) exact matches (e.g. is_cancer: true/false). Details are given in Material, available as supplementary data at *Bioinformatics* online, section 'General performance benchmarking: performance evaluation'.

2.3.3 System requirements

Metappuccino runs with PBS and Slurm schedulers and on local Linux/MacOS systems. With MetappuccinoLLModel, we recommend 8 CPU cores and one GPU (at least 40 GB VRAM); CPU-only execution is supported but significantly slower (see the project README, release v1.0.0, commit 136, for technical details).

3 Results

3.1 Predictive performance of MetappuccinoLLModel versus standard open-source LLMs

The performances of DeepSeek, Gemma, two Llama LLMs of different numbers of parameters, Qwen3, and untuned Mistral were compared to MetappuccinoLLModel. Two trivial classifiers were also included: the "Majority class" that reports the accuracy obtained by always predicting the most frequent label in a class; and the "Unknown class" that reports the accuracy obtained by always answering unknown. This way, majority-class bias and a tendency to answer "unknown" frequently by a model can be detected more easily.

3.1.1 Accuracy achieved on the synthetic sets

On ID sets, MetapuccinoModel outperforms all untrained models for all classes with an accuracy $>$ 80% for 8 of them and never below 60% (Fig. 3A). These results are expected because the ID test dataset is similar to the training dataset (Fig. S9, available as supplementary data at *Bioinformatics* online).

To assess generalization under distribution shift, we evaluated models on the MID split, which contains contexts slightly structurally different from the train set (length, vocabulary, etc.) (Fig. S9, available as supplementary data at *Bioinformatics* online). Accuracy drops moderately yet remains high and above both untuned Mistral and the strongest baselines for almost all classes (except treatment): 11 classes have around 80% accuracy or greater (Fig. 3B). Model rankings mirror those on ID, indicating predictable generalization; MetappuccinoLLModel remains first in classes where context understanding is important (e.g. biopsy type, treatment time, response). Because MID has more samples than ID (1200 versus 400), slight gains occur—for instance, organ at 0.9617 (MID) versus 0.9275 (ID)—partly due to reduced variance and the split procedure from a single balanced pool by embedding proximity (Fig. S4, available as supplementary data at *Bioinformatics* online), which can mildly overrepresent easier cases. Overall, performance degrades slightly in a predictable way with distance to the training set or stays stable depending on the class.

One important aspect to verify is that MetappuccinoLLModel does not rely on memorization and can process inputs very different from its training data. In the OOD split, the strongest shift, accuracy, remains robust (Fig. 3C). Nine classes still reach $>$80% accuracy. MetappuccinoLLModel still outperforms all models except for biopsy site (Llama 70B) and disease (Gemma). Treatment remains poorly predicted at 0.2725 across all models, indicating an intrinsically hard class.

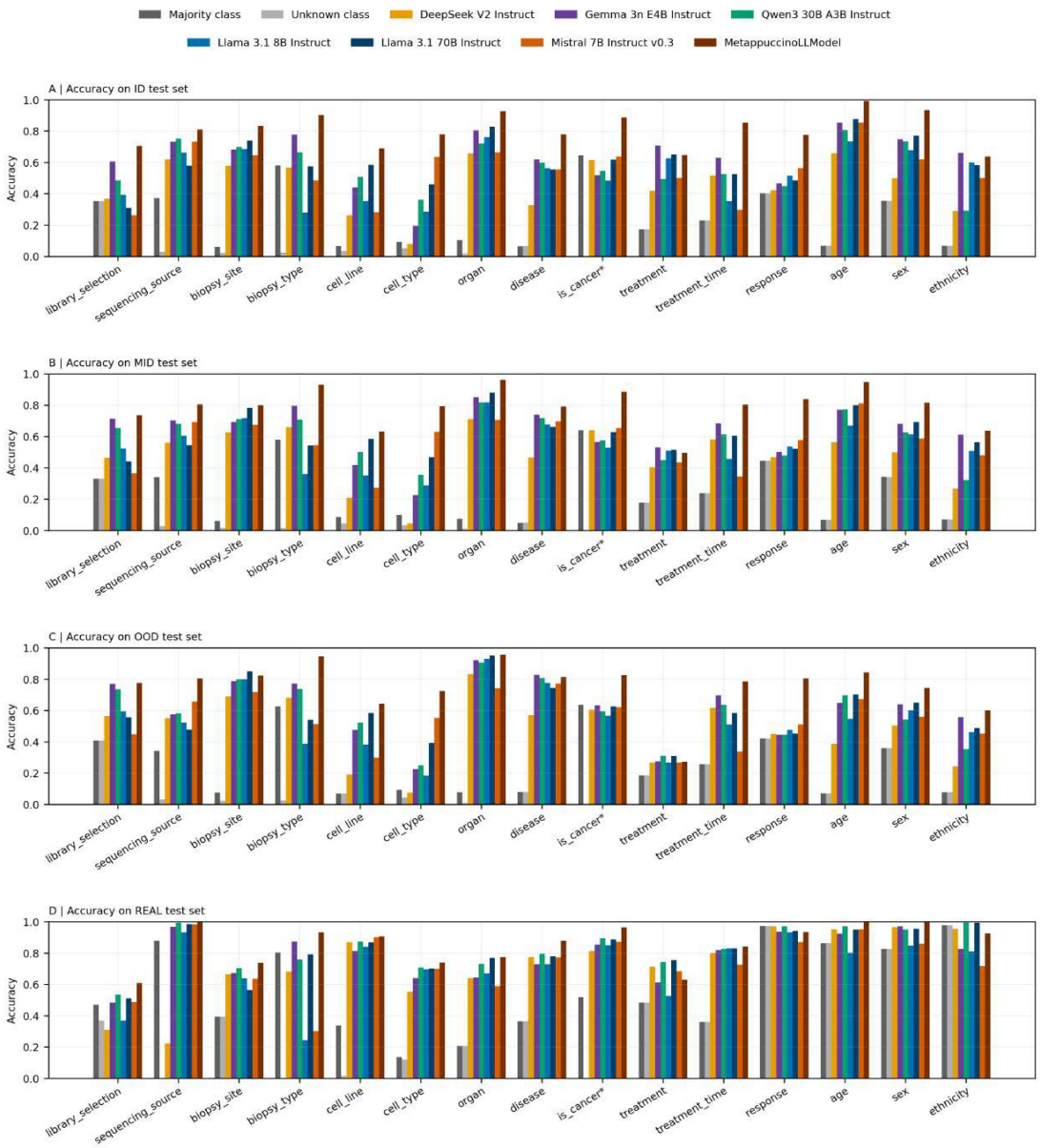

Figure 3 Per-class accuracy across the different test splits. (A–C) Accuracy on synthetic datasets constructed from an initial set of 2000 samples, split by embedding proximity to the training set: ID (400 samples) for close points, OOD (400 samples) for distant points, and MID (1200 samples) for intermediate points. (D) Dataset of 500 samples randomly extracted from human RNA-seq SRA sample metadata, excluding any samples used for model training.

In addition to these accuracy results, we further showed that they are not explained by trivial strategies or data leakage. MetappuccinoLLModel keeps similarly high performance when the information to recover is written explicitly or only hinted at through synonyms or paraphrased wording, with noticeable drops mainly in the already difficult categories such as library selection and treatment (Fig. S11). In cases where the correct answer is ‘unknown’, the model usually refrains from guessing and predicts ‘unknown’ appropriately but can still overuse this option when the relevant information is present only indirectly, again mostly for the most challenging attributes (Section 3.1.3, Fig. S12). Finally, large margins over the majority and always-unknown baselines, disjoint label values between train, validation, and test, and embedding-based diagnostics showing almost no near-duplicates across splits together support that the ID/MID/OOD benchmarks capture genuine generalization rather than memorization (Figs S9, S4, and S10, available as supplementary data at *Bioinformatics* online).

3.1.2 On real-context sets

To verify performance on unmodified real data, we evaluated 500 manually annotated SRA records across all 15 classes. It can be observed that when metadata is explicit in the given context, strong models all perform equally well. For instance classes, age, and cell line are often stated explicitly (e.g. age = x years). When information is this clear, any LLM can extract it reliably

(Fig. 3D). Where wording varies and domain knowledge matters (library selection, biopsy site, biopsy type, organ, disease, cancer flag, and cell type), MetappuccinoLLModel stays ahead. Treatment remains poorly predicted and shows no clear fine-tuning gains, especially when context differs from training. A key limitation of this real dataset is class imbalance: many classes have a dominant answer, such as sequencing source (often bulk), and response with 97.4% of cases annotated as unknown, which mechanically inflates baseline accuracies because untuned LLMs often default to "unknown" or prompt defaults when the information is not clearly stated (Fig. 3D). Along with natural variation in how easily values can be found, this explains why gaps between MetappuccinoLLModel and untuned models are smaller than on synthetic data. Nonetheless, MetappuccinoLLModel maintains strong predictive performance across all classes on real data.

3.1.3 Main prediction error types

The prediction errors made by each model for each class were investigated based on the errors made on all the test sets (Fig. 3).

3.1.3.1 MetappuccinoLLModel

About 30.6% of errors are abstentions ("unknown"). The remaining 69.4% are substitutions, as follows: library_selection = other (8.9%), cell_line = primary tissue (3.9%), biopsy_type = primary (3.1%), is_cancer = true/false (4.3% total), sequencing_source = spatial/bulk/single-cell (5.5% total), treatment_time = not applicable (2.0%), and response = stable/progressive (2.3%); the rest are many low-frequency substitutions. Additional experiments providing deeper insights into missing data, one of the main sources of LLM errors, are reported in the Materials, available as supplementary data at Bioinformatics online (section 'Additional results').

3.1.3.2 Mistral

About 42.3% of the errors are abstentions. Among substitutions, the dominant patterns are library_selection = polya (8.6%), cell_line = primary tissue (6.1%), is_cancer = true (2.7%), sequencing_source = bulk (2.4%), with smaller but recurring cases like sex = male (0.8%), sequencing_source = spatial (0.6%), biopsy_type = primary (0.6%), and cell_type = primary tissue (0.5%).

3.1.3.3 Other baseline LLMs (average)

About 58.6% of errors are abstentions. Substitution patterns are more diffuse; the largest include cell_type = primary tissue (4.5%), is_cancer = true (2.7%), library_selection = polya (2.1%), sequencing_source = bulk (2.1%), and biopsy_type = primary (2.1%), with smaller contributions from library_selection = other (1.0%), sex = male (0.9%), is_cancer = false (0.7%), and sequencing_source = spatial (0.7%).

In conclusion, thanks to fine-tuning, MetappuccinoLLModel abstains less, even if it keeps some error patterns from its Mistral backbone. When Metappuccino is wrong, it is more likely to make a plausible, structured substitution than to fall back to abstention.

3.2 Inference time across models

We measured inference times for 2000 samples on one NVIDIA A6000 GPU with a context size of a minimum of 589 tokens, a maximum of 3500 tokens, and a mean of 2337 tokens. MetappuccinoLLModel is the fastest model, with a median of around 4 s per sample and a tight dispersion, with most runs processed between 3 and 8 s (mainly depending on context size). Untuned Mistral is slightly behind with a median of around 6 s. Llama-3.1-8B and Qwen have intermediate performance with medians of 13 and 15 s. DeepSeek has a median of around 14 s Llama-3.1-70B moves to a median of 23 s. Gemma is even slower, with a median of 26 s. Finally, MetappuccinoLLModel is clearly the fastest and most stable, with intermediate sizes offering compromise and very large models being more expensive and more variable in inference time (Fig. 4). On CPU, the inference time for MetappuccinoLLModel averages 54 s, compared to over 15 minutes for Llama-3.1-70B.

3.3 Case study on 100 manually annotated melanoma samples

Class values are exactly extracted when possible by Metappuccino's rule-based preprocessing and post-processing (see Section 2). We tested whether these deterministic methods alone could match or surpass LLM inference. In Fig. 5, accuracy per class is computed only on runs whose reference is not "unknown" or "not applicable," comparing "Preprocessing Only" to "Pre/post-processing + MetappuccinoLLModel" (full pipeline). LLM-only inference was not evaluated here, as it lacks ontology access, especially to Cellosaurus, and is therefore not comparable without the post-processing dictionary match. Results show consistent gains with the full pipeline. The largest LLM improvements appear for cell type, treatment, treatment time, organ, and disease. Library selection improves slightly, although LLM predictions there remain less reliable than for other classes. Overall, LLM processing improves every class, and the full pipeline delivers the strongest performance.

The combination of LLM inference and rule-based NLP extraction therefore improves overall information retrieval, but we also showed on a 500-run real SRA benchmark that MetappuccinoLLModel alone is generally more accurate than deterministic extraction for most categories, with the notable exceptions of library selection and treatment. This motivated the per-field resolution strategy implemented in Metappuccino, where the final output file selects either the LLM or the rule-based value depending on the category, while intermediate files still expose raw rule-based and LLM-only predictions so users can inspect or override them (Fig. S14, available as supplementary data at *Bioinformatics* online).

4 Discussion

4.1 MetappuccinoLLModel training

We tested two training setups: one model learning all 15 classes with a single loss (negative log-likelihood of the true label under the model's predicted probabilities, allowing measurement of

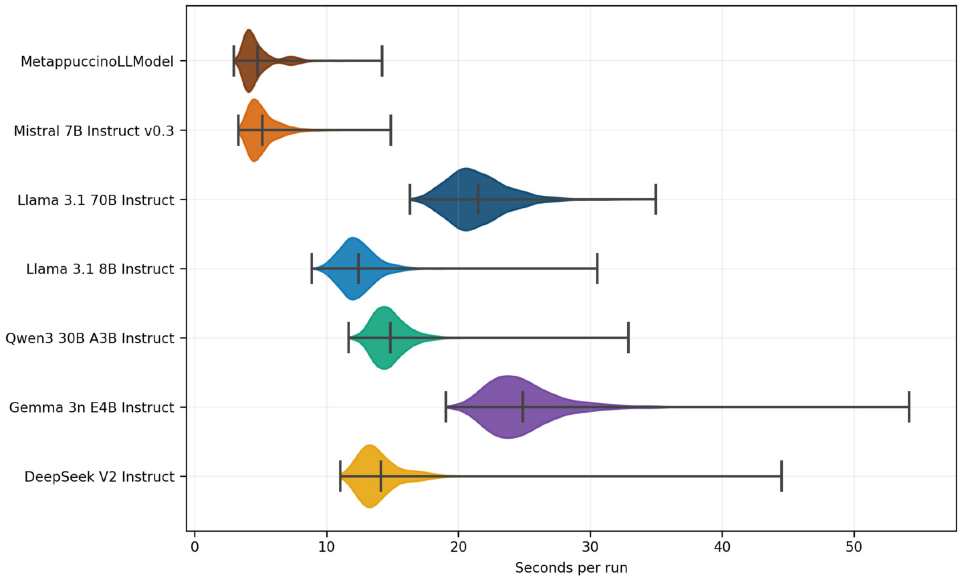

Figure 4 Distribution of per-sample inference times (seconds) for each model. The central tick marks the median, whiskers indicate the outer range.

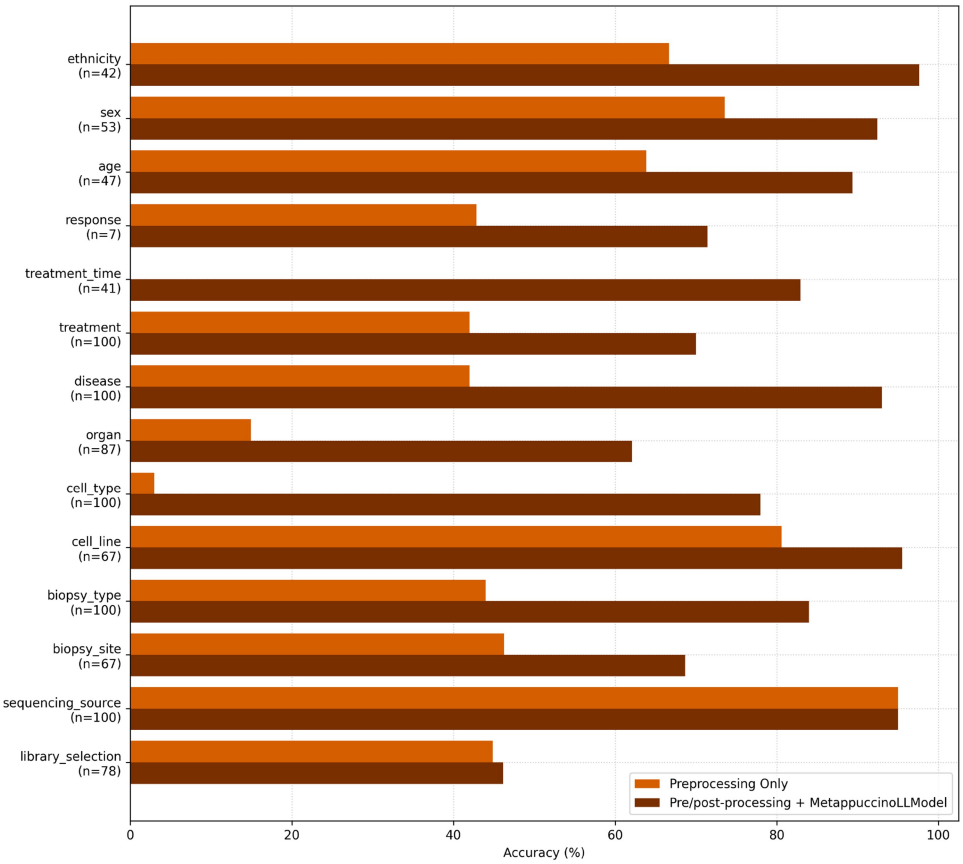

Figure 5 Metappuccino performances on the known values of the 100 melanoma-real dataset. Grouped accuracies (%) for "Preprocessing Only" and "Pre/post-processing + MetappuccinoLLModel". Each bar is computed only over non-unknown references for that class (*n*).

performance progression during training) and separate training for each class. The single loss setup was unstable; when a few classes improved, others stalled or got worse because the hardest ones held back the rest. At inference, asking for one class at a time worked better than asking for all 15 at once, no matter which LLM was used. We therefore use per-class training: it reduced interference and helped each class to specialize. Overall, performance depended most on data quality; some classes improved with more varied phrasing in the training texts, others with cleaner, hand-checked examples.

4.2 Why are there performance differences between classes?

Performance varies with the class, the clarity of the prompt, and the quality of the input. Discrete classes have few values, so the model can learn patterns and be guided to detect them rather than generate text. Open vocabulary classes rely on semantic matching to learn, so we must define what counts as correct. For example, “TB” versus “tuberculosis” can be considered correct, but special attention is needed to prevent drift and hallucinations during training caused by allowing different values than the target. Data composition also matters. In SRA or BioSample, the target information to extract is rarely written clearly, or it follows writing patterns that differ across studies. For example, the class “sex” is almost always stated explicitly, while the class “organ” may be absent yet inferable from a disease (e.g. lung is inferred when pneumonia is present). Difficulty increases even more with text diversity and distance from training examples, which explains gaps between ID, MID, and OOD tests: when learned patterns are missing, similar performance is harder to achieve. For example, if the model does not know all the ways to express “bulk,” it may only recognize patterns seen during training. Moreover, prompt engineering strongly affects results. Some classes are hard to annotate even for humans. For example, treatment time is often unclear (e.g. whether time is pre, post, or on treatment), or contexts may describe treatments applied during a study without confirming that they apply to the specific sample that has to be annotated. Such biases make extraction difficult overall, and LLMs do not outperform humans on such cases.

Regarding prediction errors, MetappuccinoLLModel often makes near-miss substitutions that a human might accept, e.g. “surgery” instead of “lung surgery,” whereas other models more often produce unrelated substitutions, an effect that reflects progress with the fine-tuning improving the fallback to simple default answers.

4.3 A standard LLM with adapters

We chose an LLM approach with per-class LoRA adapters because it is flexible: adding a new class does not require retraining the entire model but just creating a new dedicated adapter and training it locally without affecting the others. This reduces cost, limits regressions, simplifies maintenance, and allows quick evolution of the tool. This solution also avoids using multiple models for each class during inference, which would be inefficient in practice: even if some baseline LLMs sometimes achieve a similar or slightly higher score on a given class, MetappuccinoLLModel keeps the advantage across all classes, with a homogeneous and stable behavior.

4.4 Already open-source fine-tuned LLMs

We tried to use already fine-tuned models on medical data, but they are often optimized for other tasks (clinical QA, summarization), use training sets that are hard to audit, and can introduce bias. Moreover, on our task, they did not perform better than the general LLMs.

4.5 Evaluation without bias

Prediction accuracy depends on the dataset: how much information is present, how varied the phrasing is, and how balanced the values are, so an absolutely unbiased metric is hard to give. Still, the trend is clear: MetappuccinoLLModel generally outperforms baseline models, and model rankings stay stable as context patterns and values change, showing good generalization. A baseline model can match or beat it on a specific class, but the overall advantage remains. Pre and postprocessing further improve answers, and normalization with Cellosaurus, UBERON, and Disease Ontology keeps outputs consistent and medically grounded. The weaknesses of rule-based NLP are offset by the strengths of the LLM, and vice versa. Taken together, *Metappuccino* delivers a strong metadata reconstruction.

4.6 Limitations: treatment and library selection

Treatment and library selection remain harder to predict; they depend more on pre- and postprocessing. Treatment details are often diluted in study text; the model may mix treatments across samples or list all study treatments instead of the sample’s. Library selection fields blend protocols, abbreviations, and technical notes, which confuses any LLM and even humans (e.g. when ribodepletion and polyA selection are described in succession). These limits come not only from the model but also from unclear inputs and, at times, the prompt.

4.7 Future directions

The use of adapters makes Metappuccino easy to extend to new classes and domains. It is currently specialized for transcriptomic human cancer metadata, yet already usable beyond oncology when class-specific fields are ignored, and it can be expanded to bacteria, plants, or other organisms by adding new adapters. Enriching the inputs given to Metappucino, beyond SRA metadata, in order to reduce the number of unanswered predictions is another path for improvement. Users can already provide their own supplementary metadata documents, but exploring automatically new sources would give, we believe, a significant boost. We envision using longer documents such as full research articles and even switching to other extraction methods beyond LLMs, such as vision transformers. Moreover, beyond text, a hybrid strategy could use information from the sequencing reads to recover some attributes. The main challenge is scalability: applying read-level analyses across SRA is computationally expensive, so practical adoption will require efficient methods that make such read-informed inference feasible at scale.

Pending these improvements, Metappuccino already offers a robust, scalable solution for extracting and completing metadata from a simple list of SRA run accessions, combining rule-based NLP with LLM inference, requiring a few seconds per run on GPU while remaining operable on minimal CPU infrastructure. While extracted classes are currently oncology-centered, Metappuccino is applicable across diverse data types.

Author contributions

Fiona Hak (Conceptualization [equal], Software [equal], Writing—original draft [equal], Writing—review & editing [equal]), Camille Marchet (Conceptualization [supporting], Methodology [supporting], Supervision [supporting]), Daniel Gautheret (Conceptualization [equal], Funding acquisition [equal], Supervision [equal], Writing—review & editing [equal]), and Mélina Gallopin (Conceptualization [equal], Supervision [lead], Writing—review & editing [equal])

Supplementary material

Supplementary material is available at *Bioinformatics* online.

Conflicts of interest

The authors declare that they have no conflicts of interest.

Funding

Agence Nationale de la Recherche grant ANR-22-CE45-0007 and financial support from 2021–2030 Cancer Control Strategy, on funds administered by Institut national de la Santé et de la Recherche Médicale (INSERM).

Data availability

Metappuccino and all the materials used in this study including the fine-tuning and evaluation scripts are available in GitHub, at https://github.com/chumphati/Metappuccino. The associated fine-tuned adapters are available in Hugging Face, at https://huggingface.co/chumphati/MetappuccinoLLModel. The version v1.0.1 of Metappuccino is available in Zenodo, at doi.org/10.5281/zenodo.18417472.

References

AI@Meta. 2024. Llama 3 model card.

Bairoch A. The Cellosaurus, a cell-line knowledge resource. *J Biomol Tech* 2018;**29**:25–38.

Bernstein MN, Doan A, Dewey CN. MetaSRA: normalized human sample-specific metadata for the sequence read archive. *Bioinformatics* 2017;**33**:2914–23.

Caliskan A, Dangwal S, Dandekar T. Metadata integrity in bioinformatics: bridging the gap between data and knowledge. *Comput Struct Biotechnol J* 2023;**21**:4895–913.

Cinquin O. ChIP-GPT: a managed large language model for robust data extraction from biomedical database records. *Brief Bioinform* 2024;**25**:1–18.

DeepSeek-AI. 2024. Deepseek-coder-v2-instruct model card.

Google. 2025. Gemma-3n-e4b model card.

Guimarães PAS, Carvalho MGR, Ruiz JC. A computational framework for extracting biological insights from SRA cancer data. *Sci Rep* 2025;**15**:8117.

Hippen AA, Greene CS. Expanding and remixing the metadata landscape. *Trends Cancer* 2021;**7**:276–8.

Hu EJ, Shen Y, Wallis P *et al.* LoRA: low-rank adaptation of large language models. In: *ICLR 2022 - 10th International Conference on Learning Representations* 2021, 1–26.

Ikeda S, Zou Z, Bono H *et al.* Extraction of biological terms using large language models enhances the usability of metadata in the BioSample database. *Gigascience* 2025;**14**:1–13.

Jiang AQ, Sablayrolles A, Mensch A *et al.* Mistral 7B. Large language model, 2023, 1–9, preprint: not peer reviewed.

Kibbe WA, Arze C, Felix V *et al.* Disease ontology 2015 update: an expanded and updated database of human diseases for linking biomedical knowledge through disease data. *Nucleic Acids Res* 2015;**43**:D1071–D1078.

Klie A, Tsui BY, Mollah S *et al.* Increasing metadata coverage of SRA BioSample entries using deep learning–based named entity recognition. *Database* 2021;**2021**:1–11.

Leinonen R, Sugawara H, Shumway M; International Nucleotide Sequence Database Collaboration The sequence read archive. *Nucleic Acids Res* 2011;**39**:D19–D21.

Shao Z, Dai D, Guo D *et al.* Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model, 2024, arXiv, preprint: not peer reviewed.

Mistral-AI 2024. Mistral-7b-instruct model card.

Mungall CJ, Torniai C, Gkoutos GV *et al.* Uberon, an integrative multi-species anatomy ontology. *Genome Biol* 2012;**13**:R5.

Qwen-Team 2025. Qwen3-30b-a3b model card.

Rajesh A, Chang Y, Abedalthagafi MS *et al.* Improving the completeness of public metadata accompanying omics studies. *Genome Biol* 2021;**22**:106.

Taylor LJ, Abbas A, Bushman FD. Grabseqs: simple downloading of reads and metadata from multiple next-generation sequencing data repositories. *Bioinformatics* 2020;**36**:3607–9.

Yuan H, Hicks P, Ahmadian M *et al.* Annotating publicly-available samples and studies using interpretable modeling of unstructured metadata. *Brief Bioinform* 2024;**26**:1–12.

Zhao K, Farrell K, Mashiku M *et al.* A search-based geographic metadata curation pipeline to refine sequencing institution information and support public health. *Front Public Health* 2023;**11**:1254976.

Zhu Y, Stephens RM, Meltzer PS *et al.* SRAdb: query and use public next-generation sequencing data from within R. *BMC Bioinformatics* 2013;**14**:19.