Bioinformatics Advances, 2025, **00**, vbaf295
https://doi.org/10.1093/bioadv/vbaf295
Advance Access Publication Date: 18 November 2025
Application Note

OXFORD

Genome Analysis

GDC Cohort Copilot: an AI copilot for curating cohorts from the genomic data commons

Steven Song[1,2,3,†], **Anirudh Subramanyam**[1,†], **Zhenyu Zhang**[1], **Aarti Venkat**[1,4], **Robert L. Grossman**[1,2,4,*]

[1]Center for Translational Data Science, University of Chicago, Chicago, IL 60615, United States
[2]Department of Computer Science, University of Chicago, Chicago, IL 60637, United States
[3]Medical Scientist Training Program, Pritzker School of Medicine, University of Chicago, Chicago, IL 60637, United States
[4]Section of Biomedical Data Science, Department of Medicine, University of Chicago, Chicago, IL 60637, United States

*Corresponding author. Center for Translational Data Science, University of Chicago, 5454 S Shore Drive, Suite 2A/B, Chicago, IL 60615, USA. E-mail: rgrossman1@uchicago.edu.

†= equal contribution.

Associate Editor: Alex Bateman

Abstract

Motivation: The Genomic Data Commons (GDC) provides access to high quality, harmonized cancer genomics data through a unified curation and analysis platform centered around patient cohorts. While GDC users can interactively create complex cohorts through the graphical Cohort Builder, users (especially new ones) may struggle to find specific cohort descriptors across hundreds of possible fields and properties. However, users may be better able to describe their desired cohort in free-text natural language.

Results: We introduce GDC Cohort Copilot, an open-source copilot tool for curating cohorts from the GDC. GDC Cohort Copilot automatically generates the GDC cohort filter corresponding to a user-input natural language description of their desired cohort, before exporting the cohort back to the GDC for further analysis. An interactive user interface allows users to further refine the generated cohort. We develop and evaluate multiple large language models (LLMs) for GDC Cohort Copilot and demonstrate that our locally-served, open-source GDC Cohort LLM achieves better results than GPT-4o prompting in generating GDC cohorts.

Availability and implementation: We implement and share GDC Cohort Copilot as a containerized Gradio app on HuggingFace Spaces, available at https://huggingface.co/spaces/uc-ctds/GDC-Cohort-Copilot. GDC Cohort LLM weights are available at https://huggingface.co/uc-ctds. All source code is available at https://github.com/uc-cdis/gdc-cohort-copilot.

1 Introduction

The National Cancer Institute's (NCI) Genomic Data Commons (GDC) is a highly used resource for cancer research. With over 100 000 unique monthly users, the GDC provides access to high quality, harmonized, multimodal cancer data for over 45 000 patient cases (Heath *et al.* 2021). A typical user workflow using the GDC is to curate a cohort of cases before doing subsequent analysis, either using tools within the GDC Data Portal or through the GDC API (Jensen *et al.* 2017). Central to this workflow is the set of filters used to construct the cohort.

The GDC provides the Cohort Builder tool to allow users to interactively select their desired filters. The Cohort Builder is a powerful tool which allows users to select specific values from over 700 filter properties. While the Cohort Builder organizes commonly used filters into user-readable groupings, there are still dozens of properties, each with potentially a hundred or more possible values to filter by. This balance of allowing users to create specific and verbose filters while providing a user-friendly interface is complex. New users of the GDC may especially find it difficult to identify the filters most relevant for their research. However, such users may naturally be able to describe their desired cohort in natural language.

Here, we present GDC Cohort Copilot, an open-source AI copilot that enables users to curate GDC cohorts using natural language. Following the recent success of large language models (LLMs) in generating structured code (Chen *et al.* 2021) and database query languages (Ganesan *et al.* 2024, Pourreza *et al.* 2025), the GDC Cohort Copilot is powered by GDC Cohort LLM, an LLM trained to generate structured GDC cohort filters from free-text user input. We demonstrate that our locally-served, open-source model outperforms GPT-4o prompting in cohort construction accuracy. Once generated by the model, the tool automatically populates the cohort filter into a GDC Cohort Builder-like interface that allows the user to further refine their desired cohort. We provide a mechanism for exporting the curated cohort back to the GDC for further analysis. We release GDC Cohort Copilot as the overall framework presented in Fig. 1, the GDC Cohort LLM, and the containerized web app.

2 Materials and methods

GDC Cohort Copilot is comprised of both the generative GDC Cohort LLM and the containerized web app interface.

Received: September 2, 2025; **Revised:** November 7, 2025; **Accepted:** November 13, 2025

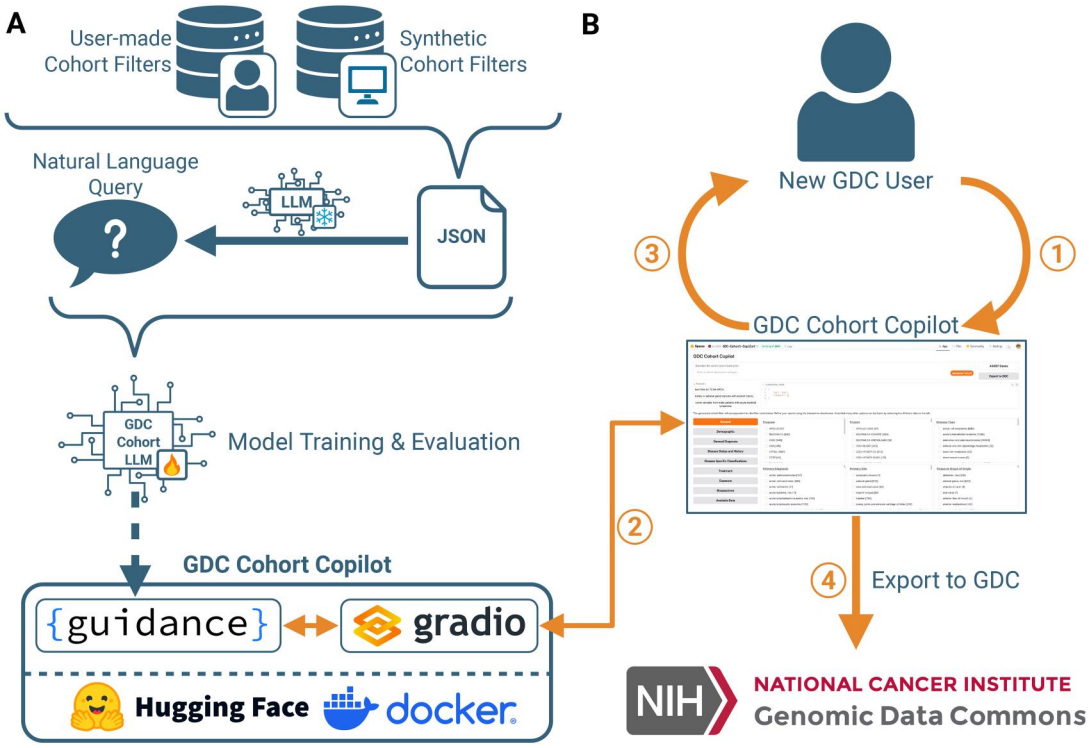

Figure 1. Overview of GDC Cohort Copilot implementation and user workflow. (A) Implementation of GDC Cohort Copilot involves training the GDC Cohort LLM to translate from a natural language query of a cohort to the cohort filter JSON. The cohort JSONs are derived from datasests of real user-made cohorts or synthetically generated cohorts. The paired natural language queries are generated by a frozen LLM using the cohort JSONs. The final trained GDC Cohort LLM model is served in a containerized web app that exposes a GDC Cohort Builder-like interface running on HuggingFace Spaces. (B) A user curates their desired cohort using the GDC Cohort Copilot by: (1) inputting a natural language description of a desired cohort (2) which is automatically passed to GDC Cohort LLM. The model is served using Guidance inside a Gradio app. (3) The resulting generated cohort filter is automatically populated back into the interface, allowing the user to manually refine their cohort before (4) exporting the curated cohort to the NCI GDC.

The overview of its implementation and user workflow is presented in Fig. 1.

2.1 Inputs and outputs

The primary input to the GDC Cohort Copilot is a natural language description of a GDC cohort, for example: "cases with gene expression data derived from RNA sequencing for lung adenocarcinoma." Upon submitting the query, the app uses GDC Cohort LLM model to generate and return the corresponding cohort filter JSON. The interface automatically populates the corresponding checkboxes for filter properties specified by the generated JSON. A user can interactively refine the cohort selections, before ultimately exporting and outputting a text file of GDC case identifiers. These case identifiers can be imported by the GDC for further analysis.

2.2 Core set of filter properties

In this initial release of the GDC Cohort Copilot, we simplify the development of the tool by considering only a subset of 68 filter properties from the GDC Cohort Builder. These are the default and most commonly used filter properties exposed by the GDC Data Portal v2.4.0 and additionally have predefined lists of possible values or value ranges (e.g. disease type or age at diagnosis). We refer to this subset of filters as the "core set." We create a JSON schema (from a Pydantic data model) to validate possible filters comprised of the core set.

2.3 GDC Cohort LLM

The GDC Cohort Copilot is powered by a generative LLM, GDC Cohort LLM, which translates natural language queries of cohorts into cohort filter JSONs. We describe here the development and evaluation of GDC Cohort LLM.

2.3.1 Training and evaluation data

GDC Cohort LLM is trained over paired natural language queries and cohort filter JSONs. This data is derived from real user-generated and synthetic cohort filters. About 68 209 user-generated cohort filters were supplied by the GDC User Services team from their database of GDC user-saved cohorts. Removing duplicates, null filters, filters with properties outside of the core set, and filters which fail schema validation results in 16 235 usable cohort filters. We additionally experiment with augmenting our dataset by randomly sampling synthetic cohorts filters. Specifically, we randomly sample fields and values from the core set of filter properties. Further details on our random sampling procedure are provided in Section A.1 (available as supplementary data at *Bioinformatics Advances* online). We experiment with augmenting our training data using 100 000 and 1 000 000 synthetic cohort filters.

One of the primary limitations of our cohort filter dataset is that it does not contain any user-generated natural language descriptions of the cohorts. To address this challenge, we prompt Mistral-7B-Instruct-v0.3 (Jiang *et al.* 2023) to generate a corresponding natural language query for a given cohort filter JSON. Our precise procedure for this reverse translation, including the prompt we use, is provided in Section A.4 (available as supplementary data at *Bioinformatics Advances* online). We apply this method to all real and synthetic cohort filters.

We finally split our paired samples derived from real user-generated data into 14 235 for training and 2000 for evaluation. For the evaluation split, we ensure that the model-specific token length of the natural language query and cohort filter for all samples fit within the minimum context length of the different models we experiment with. Additionally, we require that the evaluation samples do not result in empty cohorts (cohorts with 0 cases). This allows us to directly compare each experiment's results which were derived over precisely the same set of data samples.

We further derive a subset of 200 user-generated cohort filters, from the 2000 evaluation samples, for manual annotation. Motivated by the observation that LLMs tend to generate explicit and verbose text (Saito *et al.* 2023, Briakou *et al.* 2024), we manually write the natural language description for these cohort filters aiming to be less verbose. For example, if a cohort filter selects all lung lobes, the LLM generated synthetic query lists each of the lung lobes. However, a more natural way to describe this filter is simply "any lung lobe."

2.3.2 Model implementation

We experiment with three pretrained LLMs of different architectures and scales: GPT-2 (Radford *et al.* 2019), BART (Lewis et al. 2020), and Mistral-7B-Instruct-v0.3 (Jiang *et al.* 2023). We train each of these models using a causal language modeling (CLM—autoregressive) objective. For BART, the input to the encoder is the natural language query while the output of the decoder is the cohort filter JSON. For GPT-2 and Mistral, we concatenate the natural language query with the cohort filter JSON. Additionally, for Mistral, we use low rank adaptation (LoRA) (Hu *et al.* 2022) to efficiently train the model for our translation task. We load model weights from HuggingFace and use HuggingFace utilities for training our models. Further training details are described in Section A.2 (available as supplementary data at *Bioinformatics Advances* online).

At evaluation time, for efficient batched inferencing, we serve the trained models using vLLM (Kwon *et al.* 2023) with structured decoding using Outlines (Willard and Louf 2023) to ensure that our generated outputs are valid cohort filter JSON. One limitation of the JSON schema we develop (Section 2.2) is that it does not strictly enforce field and property strings; rather, our schema enforces the structure of the filter.

2.3.3 Evaluation metrics

After training the variations of GDC Cohort LLM, differing either in model type or data mixture, we evaluate the generated cohort filters. As we aim to enable accurate retrieval of cohorts, we do not directly evaluate the cohort filter; instead we compare the cases retrieved by the generated cohort filter to the cases retrieved by the true cohort filter. This allows flexibility in the actual content of the cohort filter as many filters may result in the same set of cases, for example selecting the TCGA program is equivalent to selecting all of the individual TCGA projects together. We thus compute three metrics for each filter's set of cases: sensitivity (true positive rate—TPR), Jaccard index (intersection over union—IoU), and a binary indicator for if the predicted and actual cases precisely match (Exact). If a generated filter is not valid (either due to context length truncation or imprecise generated field or value names) and cannot be used to retrieve cases using the GDC API, we use the empty set. TPR, IoU, and Exact are guaranteed to be finite and equal 0 if there are no predicted cases, as we ensure that the actual cases are never null (Section 2.3.1).

While we do not require the generated cohort filter precisely match the actual filter, we do evaluate whether they are semantically similar. To do so, we reverse translate the predicted cohort filters into natural language queries and compute the F1 BERTScore (Zhang *et al.* 2020) (BERT) between the original and derived queries. We specifically use SciBERT (Beltagy *et al.* 2019) in the computation of the BERTScore. Additionally, we compare all of our trained models against a prompting-based alternative using OpenAI's GPT-4o (Hurst *et al.* 2024) that requires an expensive, long context window of ∼15 000 tokens per prompt. Further details of this comparison implementation are provided in Section A.3 (available as supplementary data at *Bioinformatics Advances* online). Finally, we report the average of all metrics across all 2000 evaluation filters and apply paired *t*-tests (for TPR, IoU, BERT metrics) or McNemar's test (for Exact metric) with Bonferroni correction to evaluate statistical significance.

2.3.4 Web app, containerization, and deployment

We develop the web app for GDC Cohort Copilot as a Gradio (Abid *et al.* 2019) app deployed in a HuggingFace Space. HuggingFace Spaces provides out-of-the-box containerization with Gradio apps, enabling users to download and run GDC Cohort Copilot locally with docker. We package the GPT-2 variant of GDC Cohort LLM, trained over real and 1 million synthetic data samples, with GDC Cohort Copilot; in addition to its strong evaluation metrics, its architecture as a decoder-only, small-scale LLM enables it to be efficiently served. Specifically, we serve GDC Cohort LLM using Guidance (https://github.com/guidance-ai/guidance) for structured generation. While we utilize GPU acceleration in our HuggingFace Space for serving GDC Cohort LLM, the model only requires ∼1 GB of GPU VRAM and can even run efficiently on CPU. Our implementation of GDC Cohort Copilot allows it to be accessible to a wide variety of biomedical research users.

3 Results

We first evaluate the adaptability of various model types to our filter generation task as GDC Cohort LLM (Table 4, available as supplementary data at *Bioinformatics Advances* online). Training over user-derived data, we find that GPT-2 (TPR = 0.365; IoU = 0.331; Exact = 0.221) significantly outperforms BART (TPR = 0.117, $P = 8.56e{-}89$; IoU = 0.078, $P = 7.73e{-}114$; Exact = 0.028, $P = 3.69e{-}94$) and Mistral (TPR = 0.124, $P = 2.33e{-}90$; IoU = 0.117, $P = 3.62e{-}80$; Exact = 0.092, $P = 9.35e{-}39$) models over case-retrieval metrics. Over query-based metrics, GPT-2 (BERT = 0.819) outperforms BART (BERT = 0.735, $P = 4.03e{-}106$). While GPT-2 is statistically worse than Mistral (BERT = 0.835, $P = 6.47e{-}5$), the difference is relatively small and not meaningful in the context of poor case-retrieval capabilities.

Given GPT-2's strong adaptability to GDC Cohort LLM, we next explore how to improve its performance by training over synthetic data mixtures (Table 5, available as supplementary data at *Bioinformatics Advances* online). We find that, compared to a baseline using only user-derived data (TPR = 0.365; IoU = 0.331; Exact = 0.221; BERT = 0.819), incorporating 100 000 synthetically generated records with our real user data (TPR = 0.783, $P = 9.59e{-}217$;

Table 1. GDC Cohort LLM is significantly better at generating GDC cohorts across all reported metrics compared to GPT-4o ($P < .05$).[a]

Model	TPR	IoU	Exact	BERT
GDC Cohort LLM	**0.855**	**0.832**	**0.702**	**0.919**
GPT-4o	0.720	0.698	0.558	0.894

[a] Significantly better results are bolded. TPR: true positive rate; IoU: intersection over union; Exact: exact match; BERT: F1 BERTScore using SciBERT.

IoU = 0.748, $P = 7.19e{-}227$; Exact = 0.607, $P = 1.00e{-}188$; BERT = 0.902, $P = 7.67e{-}145$) significantly improves all metrics. We further train over a mixture of 1 million synthetic records with user records and find that this provides significantly stronger results (TPR = 0.855, $P = 6.03e{-}18$; IoU = 0.832, $P = 1.85e{-}23$; Exact = 0.702, $P = 1.20e{-}23$; BERT = 0.919, $P = 3.74e{-}16$) than using only 100 thousand synthetic samples.

Our final GDC Cohort LLM model is thus trained from a GPT-2 foundation over a mixture of 1 million synthetic and real user data. Importantly, GDC Cohort LLM (TPR = 0.855; IoU = 0.832; Exact = 0.702; BERT = 0.919) significantly outperforms a prompting-based implementation of cohort filter generation using GPT-4o (TPR = 0.720, $P = 8.01e{-}37$; IoU = 0.748, $P = 2.08e{-}36$; Exact = 0.607, $P = 2.12e{-}37$; BERT = 0.894, $P = 3.57e{-}26$) across all metrics (Table 1). Because GDC Cohort LLM is specifically trained for this task, to provide a more fair comparison, we prompt GPT-4o with a list of all possible field-value pairs which consume 15K tokens. This reduces the potential for GPT-4o to hallucinate invalid field or value names. Despite this, we find that our open-source, small-scale GDC Cohort LLM model achieves better results than GPT-4o. We conceptually compare GDC Cohort LLM to GPT-4o in Table 2.

We note that the evaluation presented in Table 1 utilizes LLM generated queries for real user-generated filters. As LLM generated text tends to be verbose and less natural than human written text (Saito *et al.* 2023, Briakou *et al.* 2024), we further evaluate GDC Cohort LLM and GPT-4o on a subset of $N = 200$ manually written, less verbose queries (described in Section 2.3.1). We find that when using manually written queries, GDC Cohort LLM and GPT-4o are not significantly different (Table 3), despite GDC Cohort LLM being orders of magnitude more efficient (Table 2).

Finally, we package GDC Cohort LLM with our GDC Cohort Copilot tool as a containerized Gradio app running on HuggingFace Spaces. GDC Cohort Copilot can additionally be downloaded and run locally using docker. We serve a GDC Cohort Builder-like interface to allow users to interactively curate cohorts using both natural language based descriptions and graphical checkboxes. We integrate with NCI GDC by providing utilities to export curated cohorts back to the GDC for further analysis.

3.1 Limitations

While we aim to provide a helpful tool to new users of the GDC, we note key limitations in this initial study. First, while we utilize the core set of filter properties (Section 2.2) that are the default filters shown in the GDC Data Portal, there are ~700 total filter properties available in the GDC. Indeed, these hidden-by-default filters are likely even more difficult for a new user to find and so future work will expand the GDC Cohort Copilot to these filter properties. Next, we note that in our experiment utilizing human written queries for human generated cohort filters (Table 3), the writer of the query and cohort filter are separate individuals. This is again a limitation of the dataset (Section 2.3.1), where the result may be that the intent of the original user who generated the filter may not have been accurately captured by the manual annotator who wrote the free-text query. Lastly, we nonetheless observe that over manually written queries, model performance on filter generation degrades when compared to using LLM written queries. It is active, ongoing work to further improve the model, however, we emphasize that no model will ever be perfect. We therefore designed the GDC Cohort Copilot tool as not only the model, but also the interactive interface that allows a user to dynamically refine the generated cohort filter. In this way, the GDC Cohort Copilot is a collaborative AI tool.

Table 2. Conceptual comparison of GDC Cohort LLM and GPT-4o as LLMs to power GDC Cohort Copilot.

Comparison	GDC Cohort LLM	GPT-4o
Achieves SOTA	✓	✗
Open source	✓	✗
Deploy locally	✓	✗
Runs on CPU-only	✓	✗
Structured outputs	✓	✓
No training	✗	✓
Required tokens	≤ 1024	$> 15k$

Table 3. In GDC cohort filter generation on a subset of 200 evaluation samples, GDC Cohort LLM still outperforms GPT-4o when using model generated queries ($P < .05$). However, GDC Cohort LLM is not significantly different than GPT-4o when using manually written queries ($P > .05$), yet remains orders of magnitude more computationally efficient.[a]

Queries	Model	TPR	IoU	Exact	BERT
Synthetic	GDC Cohort LLM	**0.919**	**0.887**	**0.780**	**0.933**
	GPT-4o	0.720	0.718	0.585	0.899
Manual	GDC Cohort LLM	0.711	0.665	0.520	0.762
	GPT-4o	0.736	0.707	0.500	0.755

[a] Significantly better results are bolded. TPR: true positive rate; IoU: intersection over union; Exact: exact match; BERT: F1 BERTScore using SciBERT.

4 Conclusion

GDC Cohort Copilot is a novel copilot tool to use natural language to assist in the curation of cohorts from the NCI GDC. Users can interactively use the copilot and a graphical interface to discover and refine their cohort. We experiment with various model types and data mixtures to develop GDC Cohort LLM, and demonstrate that our open-source, small-scale model is better able to accurately translate natural language descriptions of cohorts into their corresponding GDC cohort filter. We share GDC Cohort Copilot as a containerized Gradio app deployed on HuggingFace Spaces, ultimately providing an accessible tool to aid biomedical cancer researchers in their data curation efforts.

Acknowledgements

We thank Bill Wysocki and Wendy Teo from the GDC User Services team for their assistance in gathering data and testing

GDC Cohort Copilot. The overview figure was created using BioRender.

Author contributions

Steven Song and Anirudh Subramanyam conceived the experiments; Steven Song and Anirudh Subramanyam conducted the experiments; Steven Song and Anirudh Subramanyam implemented the software; Zhenyu Zhang provided data; Steven Song wrote the manuscript; Zhenyu Zhang, Aarti Venkat, and Robert L. Grossman supervised the work; Steven Song, Anirudh Subramanyam, Zhenyu Zhang, Aarti Venkat, and Robert L. Grossman reviewed and edited the manuscript.

Supplementary data

Supplementary data are available at *Bioinformatics Advances* online.

Conflict of interest

None declared.

Funding

This work was supported by the Advanced Research Projects Agency for Health [75N92020D00021/5N92023F00002] and the National Institutes of Health [T32GM007281 to S. S.]. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Government.

Data availability

The data underlying this article will be shared on reasonable request to the corresponding author.

References

Abid A, Abdalla A, Abid A *et al.* Gradio: Hassle-free sharing and testing of ml models in the wild. arXiv, arXiv:1906.02569, 2019, preprint: not peer reviewed.

Beltagy I, Lo K, Cohan A. Scibert: a pretrained language model for scientific text. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, 3615–20.

Briakou E, Liu Z, Cherry C *et al.* On the implications of verbose llm outputs: a case study in translation evaluation. arXiv, arXiv:2410.00863, 2024, preprint: not peer reviewed.

Chen M, Tworek J, Jun H *et al.* Evaluating large language models trained on code. arXiv, arXiv:2107.03374, 2021, preprint: not peer reviewed.

Ganesan B, Ghosh S, Gupta N *et al.* LLM-powered GraphQL generator for data retrieval. In: *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*. Jeju, Korea: IJCAI, 2024, 8657–60.

Heath AP, Ferretti V, Agrawal S *et al.* The NCI genomic data commons. *Nat Genet* 2021;**53**:257–62.

Hu EJ, Shen Y, Wallis P *et al.* LoRA: low-rank adaptation of large language models. *ICLR* 2022;**1**:3.

Hurst A, Lerer A, Goucher AP *et al.* GPT-4o system card. arXiv, arXiv:2410.21276, 2024, preprint: not peer reviewed.

Jensen MA, Ferretti V, Grossman RL *et al.* The NCI genomic data commons as an engine for precision medicine. *Blood* 2017;**130**:453–9.

Jiang AQ, Sablayrolles A, Mensch A *et al.* Mistral 7B. arXiv, arXiv:2310.06825, 2023, preprint: not peer reviewed.

Kwon W, Li Z, Zhuang S *et al.* Efficient memory management for large language model serving with PagedAttention. In: *Proceedings of the 29th Symposium on Operating Systems Principles*. Koblenz, Germany: Association for Computing Machinery, 2023, 611–26.

Lewis M, Liu Y, Goyal N *et al.* BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 2020, 7871–80.

Pourreza M, Li H, Sun R *et al.* CHASE-SQL: multi-path reasoning and preference optimized candidate selection in text-to-sql. In: *The Thirteenth International Conference on Learning Representations*. Singapore: ICLR, 2025.

Radford A, Wu J, Child R *et al.* Language models are unsupervised multitask learners. *OpenAI Blog* 2019;**1**:9.

Saito K, Wachi A, Wataoka K *et al.* Verbosity bias in preference labeling by large language models. arXiv, arXiv:2310.10076, 2023, preprint: not peer reviewed.

Willard BT, Louf R. Efficient guided generation for large language models. arXiv, arXiv:2307.09702, 2023, preprint: not peer reviewed.

Zhang T, Kishore V, Wu F *et al.* BERTScore: Evaluating text generation with BERT. In: *8th International Conference on Learning Representations*. Online: ICLR, 2020.

Bioinformatics Advances, 2025, 00, 1–5
https://doi.org/10.1093/bioadv/vbaf295
Application Note