A Platform for the Biomedical Application of Large Language Models

This manuscript (permalink) was automatically generated from biocypher/biochatter-paper@a197daf on February 6, 2024.

Authors

- **Sebastian Lobentanzer** ✉
  0000-0003-3399-6695 · slobentanzer · slobentanzer
  Heidelberg University, Faculty of Medicine and Heidelberg University Hospital, Institute for Computational Biomedicine, Heidelberg, Germany

- **Shaohong Feng**
  0009-0000-8124-3626 · fengsh27
  Department of Biomedical Informatics, The Ohio State University, Columbus, Ohio, USA

- **The BioChatter Consortium**

- **Andreas Maier**
  0000-0003-4408-0068 · andimajore
  Institute for Computational Systems Biology, University of Hamburg, Hamburg, Germany

- **Cankun Wang**
  0000-0002-0225-9855 · Wang-Cankun · Wang-Cankun
  Department of Biomedical Informatics, The Ohio State University, Columbus, Ohio, USA

- **Nils Krehl**
  TBD · nilskre
  Heidelberg University, Faculty of Medicine and Heidelberg University Hospital, Institute for Computational Biomedicine, Heidelberg, Germany

- **Qin Ma**
  0000-0002-3264-8392 · maqin2001 · QinMaBMBL
  Department of Biomedical Informatics, The Ohio State University, Columbus, Ohio, USA

- **Julio Saez-Rodriguez** ✉
  0000-0002-8552-8976 · saezrodriguez · saezlab
  Heidelberg University, Faculty of Medicine and Heidelberg University Hospital, Institute for Computational Biomedicine, Heidelberg, Germany

✉ — Correspondence possible via GitHub Issues or email to Sebastian Lobentanzer <sebastian.lobentanzer@gmail.com>, Julio Saez-Rodriguez <pub.saez@uni-heidelberg.de>.

Authors between consortium and last author are ordered alphabetically.

Abstract

Current-generation Large Language Models (LLMs) have stirred enormous interest in the recent months, yielding great potential for accessibility and automation, while simultaneously posing significant challenges and risk of misuse. To facilitate interfacing with LLMs in the biomedical space, while at the same time safeguarding their functionalities through sensible constraints, we propose a dedicated, open-source framework: BioChatter. Based on open-source software packages, we synergise the many functionalities that are currently developing around LLMs, such as knowledge integration / retrieval-augmented generation, model chaining, and benchmarking, resulting in an easy-to-use and inclusive framework for application in many use cases of biomedicine. We focus on robust and user-friendly implementation, including ways to deploy privacy-preserving local open-source LLMs. We demonstrate use cases via two multi-purpose web apps (https://chat.biocypher.org), and provide documentation, support, and an open community.

Introduction

Despite technological advances, major challenges remain to understand biological and biomedical systems [1,2]. We measure more and more data points with ever-increasing resolution to such a degree that their analysis and interpretation have become the bottleneck for their exploitation [2]. One reason for this challenge may be the inherent limitation of human knowledge [3]: Even seasoned domain experts cannot know the implications of every gene, molecule, symptom, or biomarker. In addition, biological events are context-dependent, for instance with respect to a cell type or specific disease.

Large Language Models (LLMs) of the current generation, on the other hand, can access enormous amounts of knowledge, encoded (incomprehensibly) in their billions of parameters [4,5,6]. Trained correctly, they can recall and combine virtually limitless knowledge from their training set. ChatGPT has taken the world by storm, and many biomedical researchers already use LLMs in their daily work, for general as well as research tasks [7,8,9]. However, the current, predominantly manual, way of interacting with LLMs is virtually non-reproducible, and their behaviour can be erratic. For instance, they are known to confabulate: they make up facts as they go along, and, to make matters worse, are convinced - and convincing - regarding the truth of their confabulations [9,10]. While current efforts towards Artificial General Intelligence manage to ameliorate some of the shortcomings by ensembling multiple models [11] with long-term memory stores [12], the current generation of AI does not inspire adequate trust to be applied to biomedical problems without supervision [10]. Additionally, biomedicine demands greater care in data privacy, licensing, and transparency than most other real-world issues [13].

Computational biomedicine involves many tasks that could be assisted by LLMs, such as the interpretation of experimental results, the design of experiments, the evaluation of literature, and the exploration of web resources. To improve and accelerate these tasks, we have developed BioChatter, a platform for communicating with LLMs specifically tuned to biomedical research (Figure 1). The platform guides the human researcher intuitively through the interaction with the model, while counteracting the problematic behaviours of the LLM. Since the interaction is mainly based on plain text (in any language), it can be used by virtually any researcher.

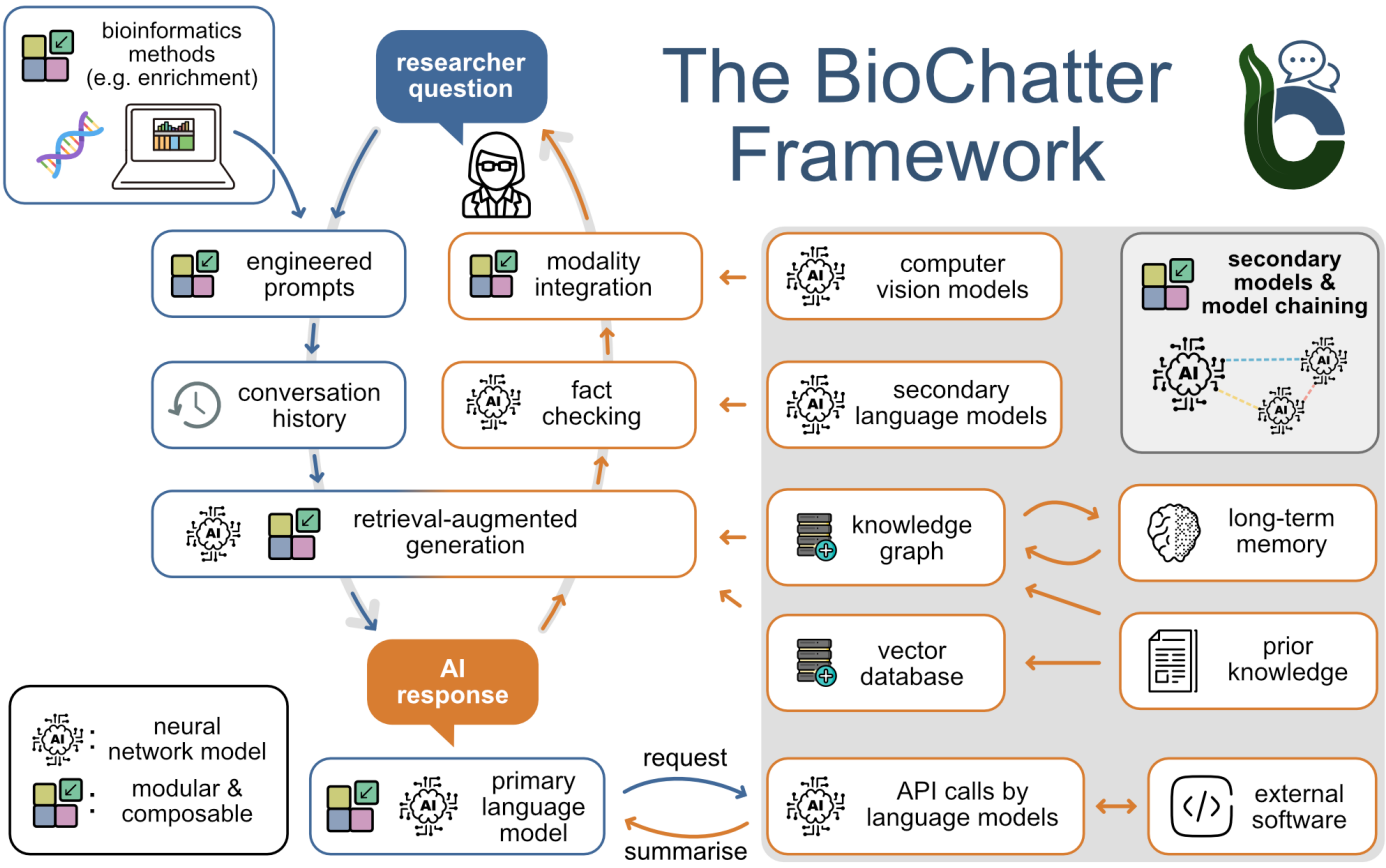

Figure 1: The BioChatter composable platform architecture (simplified). Many questions arise in daily biomedical research practice, for instance, interpretation of experimental results or the use of a web resource (top left). BioChatter's main response circuit (blue) composes a number of specifically engineered prompts and passes them (and a conversation history) to the primary LLM, which generates a response for the user based on all inputs. This response is simultaneously used to prompt the secondary circuit (orange), which fulfils auxiliary tasks to complement the primary response. In particular, using search, the secondary circuit queries a database as a prior knowledge repository and compares annotations to the primary response, or uses the knowledge to perform Retrieval-Augmented Generation (RAG). A knowledge graph such as BioCypher [14] can similarly serve as knowledge resource or long-term memory extension of the model. Further, an independent LLM receives the primary response for fact-checking, which can be supplemented with context-specific information by a RAG process. The platform is composable in most aspects, allowing arbitrary extensions to other, specialised models for additional tasks orchestrated by the primary LLM.

Results

BioChatter (https://github.com/biocypher/biochatter) is a python framework that provides an easy-to-use interface to interact with LLMs and auxiliary technologies via an intuitive API (application programming interface). This way, its functionality can be integrated into any number of user interfaces, such as web apps, command line interfaces, or Jupyter notebooks (Figure 2).

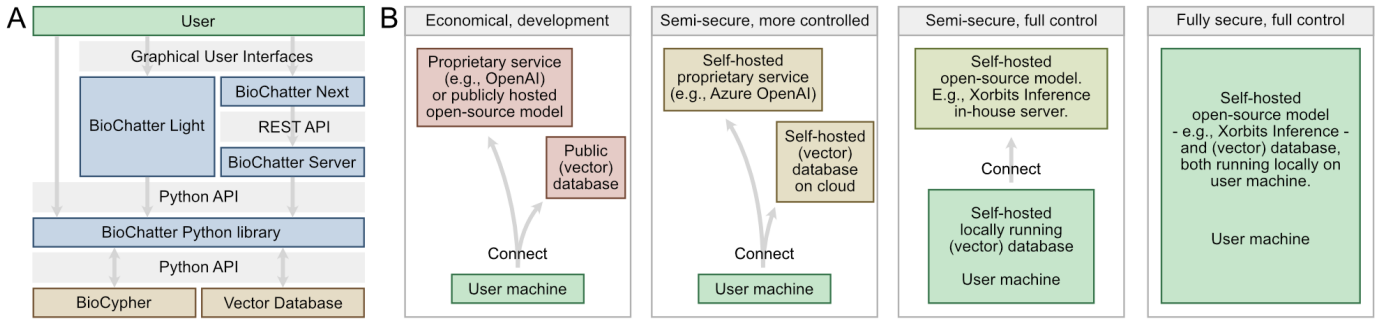

Figure 2: The BioChatter framework architecture. A) The BioChatter framework components (blue) connect to knowledge graphs and vector databases (orange). Users (green) can interact with the framework via its Python API, via the lightweight Python frontend using Streamlit (BioChatter Light), or via a fully featured web app with client-server

architecture (BioChatter Next). Developers can write simple frontends using the Streamlit framework, or integrate the REST API provided by the BioChatter Server into their own bespoke solutions. B) Different use cases of BioChatter on a spectrum of tradeoff between simplicity/economy (left) and security (right). Economical and simple solutions involve proprietary services that can be used with low effort but are subject to data privacy concerns. Increasingly secure solutions require more effort to set up and maintain, but allow the user to retain more control over their data. Fully local solutions are available given sufficient hardware (starting with contemporary laptops), but are not highly scalable.

The framework is designed to be composable, meaning that any of its components can be exchanged with other implementations (Figure 1). Functionalities include:

- **basic question answering** with LLMs hosted by providers (such as OpenAI) as well as locally deployed open-source models
- **reproducible prompt engineering** to guide the LLM towards a specific task or behaviour
- **benchmarking** of LLMs, prompts, and other components
- **knowledge graph querying** with automatic integration of any KG created in the BioCypher framework [14]
- **retrieval-augmented generation** (RAG) using vector database embeddings of user-provided literature
- **model chaining** to orchestrate multiple LLMs and other models in a single conversation using the LangChain framework [11]
- **fact-checking** of LLM responses using a second LLM

In the following, we briefly describe these components, which are demonstrated in our web apps (https://chat.biocypher.org).

Question Answering and LLM Connectivity

The core functionality of BioChatter is to interact with LLMs. The framework supports both leading proprietary models such as the GPT series from OpenAI as well as open-source models such as LLaMA2 [15] and Mixtral 8x7B [16] via a flexible open-source deployment framework [17] (see Methods). Currently, the most powerful conversational AI platform, ChatGPT (OpenAI), is surrounded by data privacy concerns [18]. We address this issue in two ways. Firstly, we provide access to the different OpenAI models through their API, which is subject to different, more stringent data protection than the web interface [19], most importantly by disallowing reuse of user inputs for subsequent model training. Secondly, we aim to preferentially support open-source LLMs to facilitate more transparency in their application and increase data privacy by being able to run a model locally on dedicated hardware and end-user devices [20]. By building on LangChain [11], we support dozens of LLM providers, such as the Xorbits Inference and Hugging Face APIs [17], which can be used to query any of the more than 100 000 open-source models on Hugging Face Hub [21], for instance those on its LLM leaderboard [22]. Although OpenAI's models currently vastly outperform any alternatives in terms of both LLM performance and API convenience, we expect many open-source developments in this area in the future [23]. Therefore, we support plug-and-play exchange of models to enhance biomedical AI-readiness, and we implement a bespoke benchmarking framework for the biomedical application of LLMs.

Prompt Engineering

An essential property of LLMs is their sensitivity to the prompt, i.e., the initial input that guides the model towards a specific task or behaviour. Prompt engineering is an emerging discipline of practical AI, and as such there are no established best practices [24,25]. Current approaches are mostly trial-and-error-based manual engineering, which is not reproducible and changes with every new model [23]. To address this issue, we include a prompt engineering framework in BioChatter that allows the preservation of prompt sets for specific tasks, which can be shared and reused by the community. In addition, to facilitate the scaling of prompt engineering, we integrate this framework in the benchmarking pipeline, which allows the automated evaluation of prompt sets as new models are published.

Benchmarking

The increasing generality of LLMs poses challenges for their comprehensive evaluation. To circumvent this issue, we focus on specific biomedical tasks and datasets. For advanced assessment, we employ automated validation of the model's responses by a second LLM. For transparent and reproducible evaluation of LLMs, we implement a benchmarking framework that allows the comparison of models, prompt sets, and all other components of the pipeline. Built on the generic Pytest framework [26], it allows the automated evaluation of a matrix of all possible combinations of components. The results are stored and displayed on our website for simple comparison, and the benchmark is updated upon the release of new models and extensions to the datasets. We create a bespoke biomedical benchmark for multiple reasons: Firstly, the biomedical domain has its own tasks and requirements, and creating a bespoke benchmark allows us to be more precise in the evaluation of components [23]. Secondly, we aim to create benchmark datasets that are complementary to the existing, general purpose benchmarks and leaderboards for LLMs [22,27]. Thirdly, we aim to prevent leakage of the benchmark data into the training data of the models, which is a known issue in the general purpose benchmarks, also called memorisation or contamination [28]. To achieve this goal, we implemented an encrypted pipeline that contains the benchmark datasets and is only accessible to the workflow that executes the benchmark (see Methods).

Current results confirm the prevailing opinion of OpenAI's leading role in LLM performance (Table [**tab:benchmark?**]). Since the benchmark datasets were created to specifically cover functions relevant in BioChatter's application domain, the benchmark results are primarily a measure for the LLMs' usefulness in our applications. However, they generally reflect the results from general-purpose benchmarks (refs). OpenAI's GPT models (gpt-3.5-turbo and gpt-4) lead by some margin, and Meta's LLaMA2 is in second position. LLaMA2 in its 70B (70 billion parameters) version shows better performance than the smaller (7B and 13B) variants, but not for all quantisations (reductions in model size due to the reduction of bits representing each parameter). The 2- and 3-bit quantisations of the 70B model show worse performance than the 7B model; the 4-bit quantised 70B model performs best among all open-source models, roughly doubling the performance of the 3-bit 70B model. The Mixtral 8x7B model (46.7 billion parameters), a generally well-performing current open-source model, shows worse performance than all LLaMA2 models in our benchmark. We will update the benchmark (https://biochatter.org/benchmark/) as new models, benchmark datasets, and BioChatter functionalities are released.

{#tab:benchmark}

Knowledge Graphs

Knowledge graphs (KGs) are a powerful tool to represent and query knowledge in a structured manner. With BioCypher [14], we have developed a framework to create KGs from biomedical data in a user-friendly manner while also semantically grounding the data in ontologies. BioChatter is an extension of the BioCypher ecosystem, elevating its user-friendliness further by allowing natural

language interactions with the data; any BioCypher KG is automatically compatible with BioChatter. We use information generated in the build process of BioCypher KGs to tune BioChatter's understanding of the data structures and contents, thereby increasing the efficiency of LLM-based KG querying (see Methods). In addition, the ability to connect to any BioCypher KG allows the integration of prior knowledge into the LLM's reasoning, which can be used to ground the model's responses in the context of the KG via in-context learning / retrieval-augmented generation (see below). We demonstrate the user experience of KG-driven interaction in Supplementary Note 1 and on our website (https://biochatter.org/vignette-kg/).

Retrieval-Augmented Generation

LLM confabulation is a major issue for biomedical applications, where the consequences of incorrect information can be severe. One popular way of addressing this issue is to apply "in-context learning," which is also more recently referred to as "retrieval-augmented generation" (RAG) [29]. Briefly, RAG relies on injection of information into the model prompt of a pre-trained model, and as such does not require retraining / fine-tuning; once created, any RAG prompt can be used with any LLM. While this can be done by processing structured knowledge, for instance from KGs, it is often more efficient to use a semantic search engine to retrieve relevant information from unstructured data sources such as literature. To this end, we allow the management and integration of vector databases in the BioChatter framework. The user is able to connect to a vector database, embed an arbitrary number of documents, and then use semantic search to improve the model prompts by adding text fragments relevant to the given question (see Methods). We demonstrate the user experience of RAG in Supplementary Note 2 and on our website (https://biochatter.org/vignette-rag/).

Model Chaining and Fact Checking

LLMs cannot only seamlessly interact with human users, but also with other LLMs as well as many other types of models. They understand API calls and can therefore theoretically orchestrate complex multi-step tasks [30,31]. However, implementation is not trivial and the complex process can lead to unpredictable behaviours. We aim to improve the stability of model chaining in biomedical applications by developing bespoke approaches for common biomedical tasks, such as interpretation and design of experiments, evaluating literature, and exploring web resources. While we focus on reusing existing open-source frameworks such as LangChain [11], we also develop bespoke solutions where necessary to provide stability for the given application. As an example, we implement a fact-checking module that uses a second LLM to evaluate the factual correctness of the primary LLM's responses continuously during the conversation (see Methods).

Discussion

The fast pace of developments around current-generation LLMs poses a great challenge to society as a whole and the biomedical community in particular [32,33,34]. While the potential of these models is enormous, their application is not straightforward, and their use requires a certain level of expertise. In addition, biomedical research is often performed in a siloed way due to the complexity of the domain and systemic incentives that work against open science and collaboration [35]. Inspired by the productivity of open source libraries such as LangChain [11], we propose an open framework that allows biomedical researchers to focus on the application of LLMs as opposed to engineering challenges. To keep the framework effective and sustainable, we focus on reusing existing open-source libraries and tools, while adapting the advancements from the wider LLM community to the biomedical domain. The transparency we emphasise at every step of the framework is essential to a sustainable application of LLMs in biomedical research and beyond [32].

To account for the requirements of biomedical research workflows, we take particular care to guarantee robustness and objective evaluation of LLM behaviour and their performance in interaction with other parts of the framework. We achieve this goal by implementing a living benchmarking framework that allows the automated evaluation of LLMs, prompts, and other components (https://biochatter.org/benchmark/). Even the most recent and biomedicine-specific benchmarking efforts are small-scale manual approaches that do not consider the full matrix of possible combinations of components, and many benchmarks are performed by accessing web interfaces of LLMs, which obfuscates important parameters, such as model version and temperature [23]. As such, a framework is a necessary step towards the objective and reproducible evaluation of LLMs. We prevent data leakage from the benchmark datasets into the training data of new models by encryption, which is essential for the sustainability of the benchmark as new models are released. The living benchmark will be updated with new questions and tasks as they arise in the community.

We facilitate access to LLMs by allowing the use of both proprietary and open-source models, and we provide a flexible deployment framework for the latter. Proprietary models are currently the most economic solution for accessing state-of-the-art models, and as such primarily suited for users just starting out or lacking the resources to deploy their own models. In contrast, open-source models are quickly catching up in terms of performance [23], and they are essential for the sustainability of the field [32]. We allow self-hosting of open-source models on any scale, from dedicated hardware with GPUs, to local deployment on end-user laptops, to browser-based deployment using web technology.

Limitations

The current generation of LLMs is not yet ready for unsupervised use in biomedical research. While we have taken steps to mitigate the risks of using LLMs, such as independent benchmarks, fact-checking, and knowledge graph querying, we cannot guarantee that the models will not produce harmful outputs. We see current LLMs, particularly in the scope of the BioCypher ecosystem, as helpful tools to assist human researchers, alleviating menial and repetitive tasks and helping with technical aspects such as query languages. They are not meant to replace human ingenuity and expertise, but to augment it with their complementary strengths.

Depending on generic open-source libraries such as LangChain [11] and Pytest [26] allows us to focus on the biomedical domain but also introduces technical dependencies on these libraries. While we support those upstream libraries via pull requests, we depend on their maintainers for future updates. In addition, keeping up with these rapid developments is demanding on developer time, which is only sustainable in a community-driven open-source effort.

Future directions

Multitask learners that can synthesise, for instance, language, vision, and molecular measurements, are an emerging field of research [36,37,38]. To remain accessible in the face of ever increasing complexity of these models, we will focus on the usability improvements that allow broad adoption in biomedical research. Autonomous agents for trivial tasks have already been developed on the basis of LLMs, and we expect this field to mature in the future [31]. As research on agent behaviour progresses, we will integrate these developments into the BioChatter framework to allow the creation of helpful assistants for biomedical research.

All framework developments will be performed in light of the ethical implications of LLMs, and we will continue to support the use of open-source models to increase transparency and data privacy. While we focus on the biomedical field, the concept of our frameworks can easily be extended to other scientific domains by adjusting domain-specific prompts and data inputs, which are accessible in a composable and user-friendly manner in our frameworks [14]. Our Python library is developed openly

on GitHub (https://github.com/biocypher/biochatter) and can be integrated into any number of user interface solutions. We develop under the permissive MIT licence and encourage contributions and suggestions from the community with regard to the addition of bioinformatics tool integrations, prompt engineering, benchmarking, and any other feature.

(Supplementary / Online) Methods

BioChatter is a Python library, supporting Python 3.10-3.12, which we ensure with a continuous integration pipeline on GitHub (https://github.com/biocypher/biochatter). We provide documentation at https://biocypher.github.io/biochatter, including a tutorial and API reference. All packages are developed openly and according to modern standards of software development [39]; we use the permissive MIT licence to encourage downstream use and development. We include a code of conduct and contributor guidelines to offer accessibility and inclusivity to all that are interested in contributing to the framework.

Applications

To demonstrate basic and advanced use cases of the framework, we provide two web apps, BioChatter Light and BioChatter Next.

BioChatter Light is a web app based on the Streamlit framework (version 1.21.0, https://streamlit.io), which is written in Python and can be deployed locally or on a server (https://github.com/biocypher/biochatter-light). The ease with which Streamlit allows the creation of interactive web apps in pure Python enables rapid iteration and agile development of new features, with the tradeoff of limited customisation and scalability. This framework is suited for rapid prototyping of bespoke solutions for specific use cases. For an up-to-date overview and preview of current functionality of the platform, please visit the online preview.

BioChatter Next (https://github.com/biocypher/biochatter-next) is a modern web app with server-client architecture, based on the open-source template of ChatGPT-Next-Web (https://github.com/ChatGPTNextWeb/ChatGPT-Next-Web). It is written combining Typescript and Python and utilises Next.js (v13.4.9) for a sleek frontend and Flask (v3.0.0) as backend. It demonstrates the use of BioChatter in a modern web app, including full customisation and scalability and localisation in 18 languages. However, this comes at the cost of increased complexity and development time. To provide seamless integration of the BioChatter backend into existing frontend solutions, we provide the server implementation at https://github.com/biocypher/biochatter-server and as a Docker image in our Docker Hub organisation (https://hub.docker.com/repository/docker/biocypher/biochatter-server).

We invite all interested researchers to select the framework that best suits their needs, or use the BioChatter server or library in their existing solutions.

Benchmarking

The benchmarking framework implements a matrix of component combinations using the parameterisation feature of Pytest [26]. This allows the automated evaluation of all possible combinations of components, such as LLMs, prompts, and datasets. As a default, we run each test five times to account for the stochastic nature of LLMs. We generally set the temperature to the lowest value possible for each model to decrease fluctuation. The results are stored in a database and displayed on the website for easy comparison. The benchmark is updated upon the release of new models and extensions to the datasets. The individual dimensions of the matrix are:

- **LLMs**: Testing proprietary (OpenAI) and open-source models (commonly using the Xorbits Inference API and HuggingFace models) against the same set of tasks is the primary aim of our benchmarking framework. We facilitate the automation of testing by including a programmatic way of deploying open-source models.

- **prompts**: Since model performance can dramatically rely on the used prompts, a set of prompts for each task with varying degrees of specificity and fixed as well as variable components is used to evaluate this variability.

- **datasets**: We test various tasks using a set of datasets for each task in question-answer-style.

- **data processing**: Some data processing steps can have great impact on the downstream performance of LLMs. For instance, we test the conversion of numbers (which LLMs are notoriously bad at handling) to categorical text (e.g., low, medium, high).

- **model quantisations**: We test a set of quantisations for each model (where available) to account for the trade-off between model size and performance.

- **model parameters**: Where suitable, we test a set of parameters for each model, such as "temperature," which determines the reproducibility of model responses.

- **integrations**: We write dedicated tests for specific tasks that require integrations, for instance with knowledge graphs or vector databases.

- **stochasticity**: To account for variability in model responses, we include a parameter to run each test multiple times and generate summary statistics.

- **sentiment and behaviour**: To assess whether the models exhibit the desired behaviour patterns for each of the personas, we let a second LLM evaluate the responses based on a set of criteria, including professionalism and politeness.

The Pytest framework is implemented at https://github.com/biocypher/biochatter/blob/main/benchmark, and more information and results are available at https://biocypher.github.io/biochatter/benchmarking. The Pytest matrix uses a hash-based system to evaluate whether a model-dataset combination has been run before. Briefly, the hash is calculated from the dictionary representation of the test parameters, and the test is skipped if the combination of hash and model name is already present in the database. This allows automatic running of all tests that have been newly added or modified.

To prevent leakage of benchmarking data (and subsequent contamination of future LLMs), we implement an encryption routine on the benchmark datasets. The encryption is performed using a hybrid encryption scheme, where the data are encrypted with a symmetric key, which is in turn encrypted with an asymmetric key. The datasets are stored in a dedicated encrypted pipeline that is only accessible to the workflow that executes the benchmark. These processes are implemented at https://github.com/biocypher/llm-test-dataset and accessed from the benchmark procedure in BioChatter.

Knowledge Graphs

We utilise the close connection between BioChatter and the BioCypher framework [14] to integrate knowledge graph (KG) queries into the BioChatter API. In the BioCypher KG creation, we use a configuration file to map KG contents to ontology terms, including information about each of the entities. For instance, we detail the properties of a node and the source and target classes of an edge.

Additionally, during the KG build process, we enrich this information and save it to a YAML file and, optionally, directly to the KG. This information is used by BioChatter to tune its understanding of the KG, which allows the LLM to query the KG more efficiently. By understanding the context of the KG, the exact contents, and the exact spelling of all identifiers and properties, we effectively support the LLM in generating correct queries. To illustrate the usage of this feature, we provide a demonstration repository at https://github.com/biocypher/pole including a KG build procedure and web app, which can be run using a single Docker Compose command. The pole KG can also be used in conjunction with the BioChatter Next app by using the `docker-compose-incl-kg.yaml` file to build the application locally.

Retrieval-Augmented Generation

While current LLMs possess extensive internal general knowledge, they may not know how to prioritise very specific scientific results, or they may not have had access to some research articles in their training data (e.g., due to their recency or licensing issues). To bridge this gap, we can provide additional information from relevant publications to the model via the prompt. However, we frequently cannot add entire publications to the prompt, since the input length of current models still is restricted; we need to isolate the information that is specifically relevant to the question given by the user. To find this information, we perform a semantic similarity search between the user's question and the contents of user-provided scientific articles (or other texts). The most efficient way to do this mapping is by using a vector database [40].

The contextual background information provided by the user (e.g., by uploading a scientific article of prior work related to the experiment to be interpreted) is split into pieces suitable to be digested by the LLM, which are individually embedded by the model. These embeddings (represented by vectors) are used to store the text fragments in a vector database; the storage as vectors allows fast and efficient retrieval of similar entities via the comparison of individual vectors. For example, the two sentences "Amyloid beta levels are associated with Alzheimer's Disease stage." and "One of the most important clinical markers of AD progression is the amount of deposited A-beta 42." would be closely associated in a vector database (given the embedding model is of sufficient quality, i.e., similar to GPT-3 or better), while traditional text-based similarity metrics probably would not identify them as highly similar.

By comparing the user's question to prior knowledge in the vector database, we can extract the relevant pieces of information from the entire background. Even better, we can first use an LLM to generate an answer to the user's question and then use this answer to query the vector database for relevant information. Regardless of whether the initial answer is correct, it is likely that the "fake answer" is more semantically similar to the relevant pieces of information than the user's question [40]. Semantic search results (for instance, single sentences directly related to the topic of the question) are then sufficiently small to be added to the prompt. In this way, the model can learn from additional context without the need for retraining or fine-tuning. This method is sometimes described as in-context learning [29] or retrieval-augmented generation [41].

To provide access to this functionality in BioChatter, we implement classes for the connection to, and management of, vector database systems (in the vectorstore_host.py module), and for performing semantic search on the vector database and injecting the results into the prompt (in the vectorstore.py module). To demonstrate the use of the API, we add a "Retrieval-Augmented Generation" tab to the preview apps that allows the upload of text documents to be added to a vector database, which then can be queried to add contextual information to the prompt sent to the primary model. This contextual information is transparently displayed. Since this functionality requires a connection to a vector database system, we provide connectivity to a Milvus server, including a way to start the server in conjunction with a BioCypher knowledge graph and the BioChatter Light app in one Docker Compose workflow.

Deployment of Open-Source Models

To facilitate access to open-source models, we adopt a flexible deployment framework based on the Xorbits Inference API [17]. Xorbits Inference includes a large number of open-source models out of the box, and new models from Hugging Face Hub [42] can be added using the intuitive graphical user interface. We use Xorbits Inference version 0.8.4 to deploy the benchmarked models, and we provide a Docker Compose repository to deploy the app on a Linux server with Nvidia GPUs (https://github.com/biocypher/xinference-docker-builtin/). This Compose uses the multi-architecture image (for ARM64 and AMD64 chips) we provide on our Docker Hub organisation (https://hub.docker.com/repository/docker/biocypher/xinference-builtin).

Model Chaining

The ability of LLMs to control external software, including other LLMs, opens up a wide range of possibilities for the orchestration of complex tasks. A simple example is the implementation of a correcting agent, which receives the output of the primary model and checks it for factual correctness. If the agent detects an error, it can prompt the primary model to correct its output, or forward this correction to the user directly. Since this relies on the internal knowledge base of the correcting agent, the same caveats apply, as the correcting agent may confabulate as well. However, since the agent is independent of the primary model (being set up with dedicated prompts), it is less likely to confabulate in the same way.

This approach can be extended to a more complex model chain, where the correcting agent, for example, can query a knowledge graph or a vector database to ground its responses in prior knowledge. These chains are easy to implement, and some are available out of the box in the LangChain framework [11]. However, they can behave unpredictably, which increases with the number of links in the chain, and as such should be tightly controlled. They also add to the computational burden of the system, which is particularly relevant for deployments on end-user devices.

Author Contributions

SL conceptualised and developed the platform and wrote the manuscript. AM implemented the local deployment functionality. NK implemented benchmarking procedures. CW architected the BioChatter Next server infrastructure. QM oversaw the development and deployment of the BioChatter Next server environment. SF integrated BioChatter continuous integration pipelines and developed both front-end and back-end components for the BioChatter Next server. JSR supervised the project, revised the manuscript, and acquired funding. All authors read and approved the final manuscript.

Acknowledgements

We thank Hanna Schumacher, Daniel Dimitrov, Pau Badia i Mompel, and Aurelien Dugourd for feedback on the original draft of the manuscript and the software.

This work was supported by funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 965193 (SL), award U54-AG075931 (QM) from the National Institutes of Health, award NSF1945971 (QM) from the National Science Foundation, and the Pelotonia Institute of Immuno-Oncology (PIIO).

Conflict of Interest

JSR reports funding from GSK, Pfizer and Sanofi and fees/honoraria from Travere Therapeutics, Stadapharm, Pfizer, Grunenthal, and Astex Pharmaceuticals.

References

1. **Study reveals cancer's 'infinite' ability to evolve**
James Gallagher
BBC News (2023-04-12) https://www.bbc.com/news/health-65252510

2. **Current progress and open challenges for applying deep learning across the biosciences**
Nicolae Sapoval, Amirali Aghazadeh, Michael G Nute, Dinler A Antunes, Advait Balaji, Richard Baraniuk, CJ Barberan, Ruth Dannenfelser, Chen Dun, Mohammadamin Edrisi, … Todd J Treangen
Nature Communications (2022-04-01) https://doi.org/gp26xk
DOI: 10.1038/s41467-022-29268-7 · PMID: 35365602 · PMCID: PMC8976012

3. **Capacity limits of information processing in the brain**
René Marois, Jason Ivanoff
Trends in Cognitive Sciences (2005-06) https://doi.org/d5gmqt
DOI: 10.1016/j.tics.2005.04.010 · PMID: 15925809

4. **PaLM: Scaling Language Modeling with Pathways**
Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, … Noah Fiedel
arXiv (2022) https://doi.org/kfxf
DOI: 10.48550/arxiv.2204.02311

5. **LaMDA: Language Models for Dialog Applications**
Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, … Quoc Le
arXiv (2022) https://doi.org/kmfc
DOI: 10.48550/arxiv.2201.08239

6. **GPT-4 Technical Report**
OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, … Barret Zoph
arXiv (2023) https://doi.org/grx4cb
DOI: 10.48550/arxiv.2303.08774

7. **Autonomous chemical research with large language models**
Daniil A Boiko, Robert MacKnight, Ben Kline, Gabe Gomes
Nature (2023-12-20) https://doi.org/gs9v4v
DOI: 10.1038/s41586-023-06792-0 · PMID: 38123806 · PMCID: PMC10733136

8. **Assessing GPT-4 for cell type annotation in single-cell RNA-seq analysis**
Wenpin Hou, Zhicheng Ji
Cold Spring Harbor Laboratory (2023-04-21) https://doi.org/gsznzg
DOI: 10.1101/2023.04.16.537094 · PMID: 37131626 · PMCID: PMC10153208

9. **How will generative AI disrupt data science in drug discovery?**
Jean-Philippe Vert
Nature Biotechnology (2023-05-08) https://doi.org/gsznzd
DOI: 10.1038/s41587-023-01789-6 · PMID: 37156917

10. **Foundation models for generalist medical artificial intelligence**
Michael Moor, Oishi Banerjee, Zahra Shakeri Hossein Abad, Harlan M Krumholz, Jure Leskovec, Eric J Topol, Pranav Rajpurkar

Nature (2023-04-12) https://doi.org/gr4td4
DOI: 10.1038/s41586-023-05881-4 · PMID: 37045921

11. **Langchain** https://python.langchain.com/

12. **AutoGPT Official**
AutoGPT Official
(2024-01-29) https://autogpt.net/

13. **Towards Conversational Diagnostic AI**
Tao Tu, Anil Palepu, Mike Schaekermann, Khaled Saab, Jan Freyberg, Ryutaro Tanno, Amy Wang, Brenna Li, Mohamed Amin, Nenad Tomasev, … Vivek Natarajan
arXiv (2024) https://doi.org/gtdmpj
DOI: 10.48550/arxiv.2401.05654

14. **Democratizing knowledge representation with BioCypher**
Sebastian Lobentanzer, Patrick Aloy, Jan Baumbach, Balazs Bohar, Vincent J Carey, Pornpimol Charoentong, Katharina Danhauser, Tunca Doğan, Johann Dreo, Ian Dunham, … Julio Saez-Rodriguez
Nature Biotechnology (2023-06-19) https://doi.org/gszqjr
DOI: 10.1038/s41587-023-01848-y · PMID: 37337100

15. **Llama 2: Open Foundation and Fine-Tuned Chat Models**
Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, … Thomas Scialom
arXiv (2023) https://doi.org/ktkj
DOI: 10.48550/arxiv.2307.09288

16. **Mixtral of Experts**
Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, … William El Sayed
arXiv (2024) https://doi.org/gtc2g3
DOI: 10.48550/arxiv.2401.04088

17. **xorbitsai/inference**
Xorbits
(2024-02-06) https://github.com/xorbitsai/inference

18. https://www.reuters.com/technology/european-data-protection-board-discussing-ai-policy-thursday-meeting-2023-04-13/

19. **Terms of use** https://openai.com/policies/terms-of-use

20. **Why open-source generative AI models are an ethical way forward for science**
Arthur Spirling
Nature (2023-04-18) https://doi.org/gsqx6v
DOI: 10.1038/d41586-023-01295-4 · PMID: 37072520

21. **Hugging Face Hub documentation** https://huggingface.co/docs/hub/index

22. **Open LLM Leaderboard - a Hugging Face Space by HuggingFaceH4**
https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

23. **BioLLMBench: A Comprehensive Benchmarking of Large Language Models in Bioinformatics**

Varuni Sarwal, Viorel Munteanu, Timur Suhodolschi, Dumitru Ciorba, Eleazar Eskin, Wei Wang, Serghei Mangul
Cold Spring Harbor Laboratory (2023-12-20) https://doi.org/gtbgvk
DOI: 10.1101/2023.12.19.572483

24. **A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT**
Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, Douglas C Schmidt
arXiv (2023) https://doi.org/grxct8
DOI: 10.48550/arxiv.2302.11382

25. **Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4**
Sondos Mahmoud Bsharat, Aidar Myrzakhan, Zhiqiang Shen
arXiv (2023) https://doi.org/gtdnfg
DOI: 10.48550/arxiv.2312.16171

26. **pytest-dev/pytest**
pytest-dev
(2024-02-06) https://github.com/pytest-dev/pytest

27. **Large language models encode clinical knowledge**
Karan Singhal, Shekoofeh Azizi, Tao Tu, SSara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, … Vivek Natarajan
Nature (2023-07-12) https://doi.org/gsgp8c
DOI: 10.1038/s41586-023-06291-2 · PMID: 37438534 · PMCID: PMC10396962

28. **NLP Evaluation in trouble: On the Need to Measure LLM Data Contamination for each Benchmark**
Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, Eneko Agirre
arXiv (2023) https://doi.org/gtbgvp
DOI: 10.48550/arxiv.2310.18018

29. **HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face**
Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, Yueting Zhuang
arXiv (2023) https://doi.org/gskd97
DOI: 10.48550/arxiv.2303.17580

30. **Gorilla: Large Language Model Connected with Massive APIs**
Shishir G Patil, Tianjun Zhang, Xin Wang, Joseph E Gonzalez
arXiv (2023) https://doi.org/gtbgvm
DOI: 10.48550/arxiv.2305.15334

31. **A Survey on Large Language Model based Autonomous Agents**
Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, … Ji-Rong Wen
arXiv (2023) https://doi.org/gsv93m
DOI: 10.48550/arxiv.2308.11432

32. **There are holes in Europe's AI Act — and researchers can help to fill them**
Nature
(2024-01-10) https://doi.org/gtdnfb
DOI: 10.1038/d41586-024-00029-4 · PMID: 38200306

33. **Is AI leading to a reproducibility crisis in science?**
Philip Ball

Nature (2023-12-05) https://doi.org/gs8bmk
DOI: 10.1038/d41586-023-03817-6 · PMID: 38052897

34. **Generative AI could revolutionize health care — but not if control is ceded to big tech**
Augustin Toma, Senthujan Senkaiahliyan, Patrick R Lawler, Barry Rubin, Bo Wang
Nature (2023-11-30) https://doi.org/gtdnd9
DOI: 10.1038/d41586-023-03803-y · PMID: 38036861

35. **Scientific Utopia**
Brian A Nosek, Jeffrey R Spies, Matt Motyl
Perspectives on Psychological Science (2012-11) https://doi.org/f4fc2k
DOI: 10.1177/1745691612459058 · PMID: 26168121 · PMCID: PMC10540222

36. **Git-Theta: A Git Extension for Collaborative Development of Machine Learning Models**
Nikhil Kandpal, Brian Lester, Mohammed Muqeeth, Anisha Mascarenhas, Monty Evans, Vishal Baskaran, Tenghao Huang, Haokun Liu, Colin Raffel
arXiv (2023) https://doi.org/gtdnfd
DOI: 10.48550/arxiv.2306.04529

37. **Crosslingual Generalization through Multitask Finetuning**
Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, MSaiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, … Colin Raffel
arXiv (2022) https://doi.org/gtdnfc
DOI: 10.48550/arxiv.2211.01786

38. **MiniGPT-v2: large language model as a unified interface for vision-language multi-task learning**
Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, Mohamed Elhoseiny
arXiv (2023) https://doi.org/gtdnff
DOI: 10.48550/arxiv.2310.09478

39. **The TRUST Principles for digital repositories**
Dawei Lin, Jonathan Crabtree, Ingrid Dillo, Robert R Downs, Rorie Edmunds, David Giaretta, Marisa De Giusti, Hervé L'Hours, Wim Hugo, Reyna Jenkyns, … John Westbrook
Scientific Data (2020-05-14) https://doi.org/ggwrtj
DOI: 10.1038/s41597-020-0486-7 · PMID: 32409645 · PMCID: PMC7224370

40. **Large Language Models for Information Retrieval: A Survey**
Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zhicheng Dou, Ji-Rong Wen
arXiv (2023) https://doi.org/gtbgvn
DOI: 10.48550/arxiv.2308.07107

41. **Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks**
Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, … Douwe Kiela
Advances in Neural Information Processing Systems (2020)
https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf

42. **Hugging Face – The AI community building the future.** (2024-02-04) https://huggingface.co/