XXXII Symposium on Bioinformatics and Computer-Aided Drug Discovery (BCADD-2026)
October 19–21, 2026 |

ChatGPT-like tools in bioinformatics: architectural approaches and functionality of ExpasyGPT, STRING-Chat, and MEGA-GPT

Yana Syrovatskaya [1], **Yuriy Orlov** [1,2]

1 I.M. Sechenov First Moscow State Medical University (Sechenov University), Moscow, Russia
2 Novosibirsk State University, Novosibirsk, Russia

yana.syr@bk.ru

The Data Integration Challenge & Study Aim

1. THE PROBLEM

Biological data are highly heterogeneous:

- Different formats
- Different storage systems
- Diverse access methods

Data integration & reuse are complex

2. THE SOLUTION (SIB)

Semantic Web + public SPARQL endpoints

- Databases: UniProt, Bgee, STRING, OMA

Enables complex & federated queries across multiple sources

3. TECHNICAL BARRIER

Writing SPARQL queries requires specialized syntax & ontology knowledge

- Creates a significant technical barrier for biologists & clinicians

4. THE ROLE OF LLMs

ChatGPT-like tools act as a bridge:

- Convert natural language into structured queries or instructions
- No coding or syntax expertise required

STUDY AIM:

To conduct the first systematic comparative analysis of the architectural features and functionality of three ChatGPT-like tools — ExpasyGPT, STRING-Chat, and MEGA-GPT — and evaluate their prospects in bioinformatics, systems biology, and biomedical research (including drug target identification).

Architectural Approaches

1. ExpasyGPT

ROLE: "TRANSLATOR" (NL → SPARQL)

Architecture: RAG + LLM + SPARQL

- Scans 1000+ example queries (65 federated) on GitHub
- Converts plain language into executable SPARQL code
- Queries SIB graphs: UniProt, Bgee, OMA, STRING

1. User Question (Plain Language)

2. RAG + LLM (Retriever + Generator)

3. Federated SPARQL Endpoints (SIB)

4. Structured Data + SPARQL Code

2. STRING-Chat

ROLE: "TRANSLATOR" (NL → MCP Tools)

Architecture: LLM + MCP Server

- LLM acts as an orchestrator via Model Context Protocol
- Sequentially calls external STRING database tools
- Aggregates data via SPARQL backend into graphs

1. User Question (Plain Language)

2. LLM + MCP Server (Tool Orchestrator)

3. MCP Tools: network, homology, visual

4. Unified Response + Network Graph

3. MEGA-GPT

ROLE: "INSTRUCTOR" (NL → Protocol)

Architecture: RAG on Documentation

- Scans indexed MEGA documentation & protocols
- Grounds AI in software logic (reduces hallucinations)
- Does NOT execute queries — instructs the user

1. User Question (Plain Language)

2. RAG Retriever (MEGA Documentation)

3. LLM Generator (Workflow Synthesis)

4. Step-by-Step Guide

ExpasyGPT: Query Example

1. ExpasyGPT: Gene Expression Query

Test Query: "Where is the ACE2 gene expressed in humans?"

- **Automated Translation:** Converts plain English into an executable SPARQL query targeting the Bgee database.
- **Structured Output:** Returns anatomical entities, expression scores, and confidence metrics.
- **Detected Organs:** Intestine, Testis, Kidney, Heart, Lung, Thyroid.
- **Key Benefit:** Ready-to-use SPARQL code without writing a single line of syntax.

You can retrieve the anatomical entities where **human ACE2** is reported as expressed using Bgee's SPARQL endpoint:

```
#+ endpoint: https://www.bgee.org/sparql/
PREFIX genex: <http://purl.org/genex#>
PREFIX obo:   <http://purl.obolibrary.org/obo/>
PREFIX orth:  <http://purl.org/net/orth#>
PREFIX rdfs:  <http://www.w3.org/2000/01/rdf-schema#>
PREFIX up:    <http://purl.uniprot.org/core/>

SELECT DISTINCT ?geneName ?anat ?anatName
WHERE {
  ?gene a orth:Gene ;
        rdfs:label ?geneName ;
        genex:isExpressedIn ?anat ;
        orth:organism ?organism .

  ?anat a genex:AnatomicalEntity ;
        rdfs:label ?anatName .

  ?organism obo:RO_0002162 ?species .
  ?species a up:Taxon ;
           up:scientificName "Homo sapiens" .

  FILTER(LCASE(?geneName) = "ace2")
}
LIMIT 100
```

Generated SPARQL

Expression Levels

Your output already answers the question with quantitative support: in Bgee, **ACE2** shows its **highest expression levels** (with *high confidence* in many cases) in **gastro-intestinal tissues** and **reproductive/renal/cardiac tissues**. From the first rows you pasted (ordered by `maxExpressionLevel`):

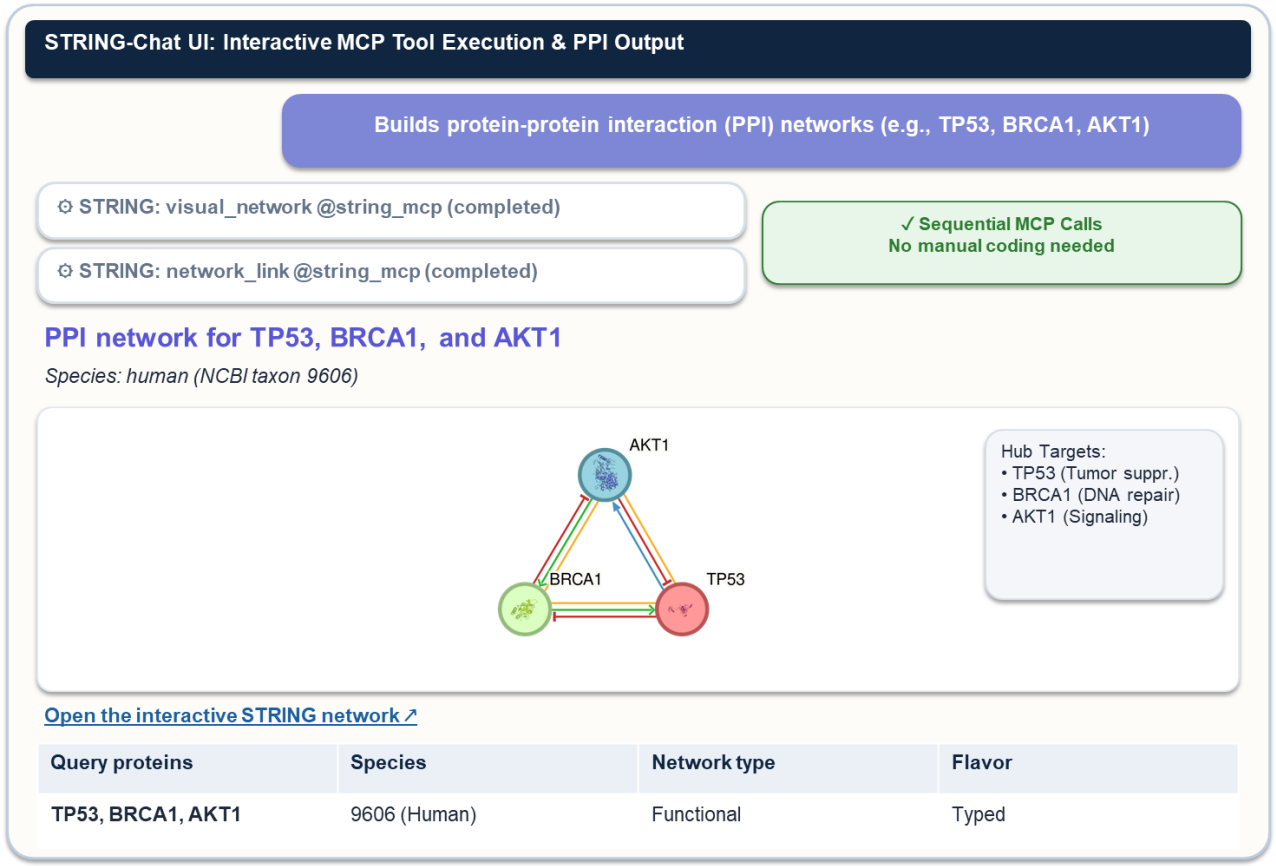
- **Intestine** (maxExpressionLevel ≈ 99.08; bestFDR ≈ 1e-14; high confidence)
- **Gonad / testis / male reproductive system** (≈ 98.38; bestFDR ≈ 1e-14; high confidence)
- **Ileal mucosa / ileum** (≈ 97.7–97.6; medium/high confidence depending on term)
- **Colon / large intestine / digestive tract** (≈ 97.37; high confidence)
- **Kidney** (≈ 94.82; high confidence)
- **Heart (ventricle/atrium)** (≈ 92.0–89.5; high confidence)
- **Thyroid gland** (≈ 74.2; high confidence)
- **Right lung** appears as well (≈ 73.0; high confidence)

STRING-Chat: Query Example

2. STRING-Chat: Interaction Networks

Test Task: Reconstructing PPI & Disease Networks

- **MCP Orchestration:** Sequentially calls string_network, string_homology, and string_visual_network.
- **Graph Reconstruction:** Builds protein-protein interaction (PPI) networks (e.g., TP53, BRCA1, AKT1).
- **Drug Discovery Value:** Identifies highly connected hub genes (potential drug targets) & pathways.
- **Key Benefit:** Aggregates multi-tool outputs and visual graphs in one dialogue.

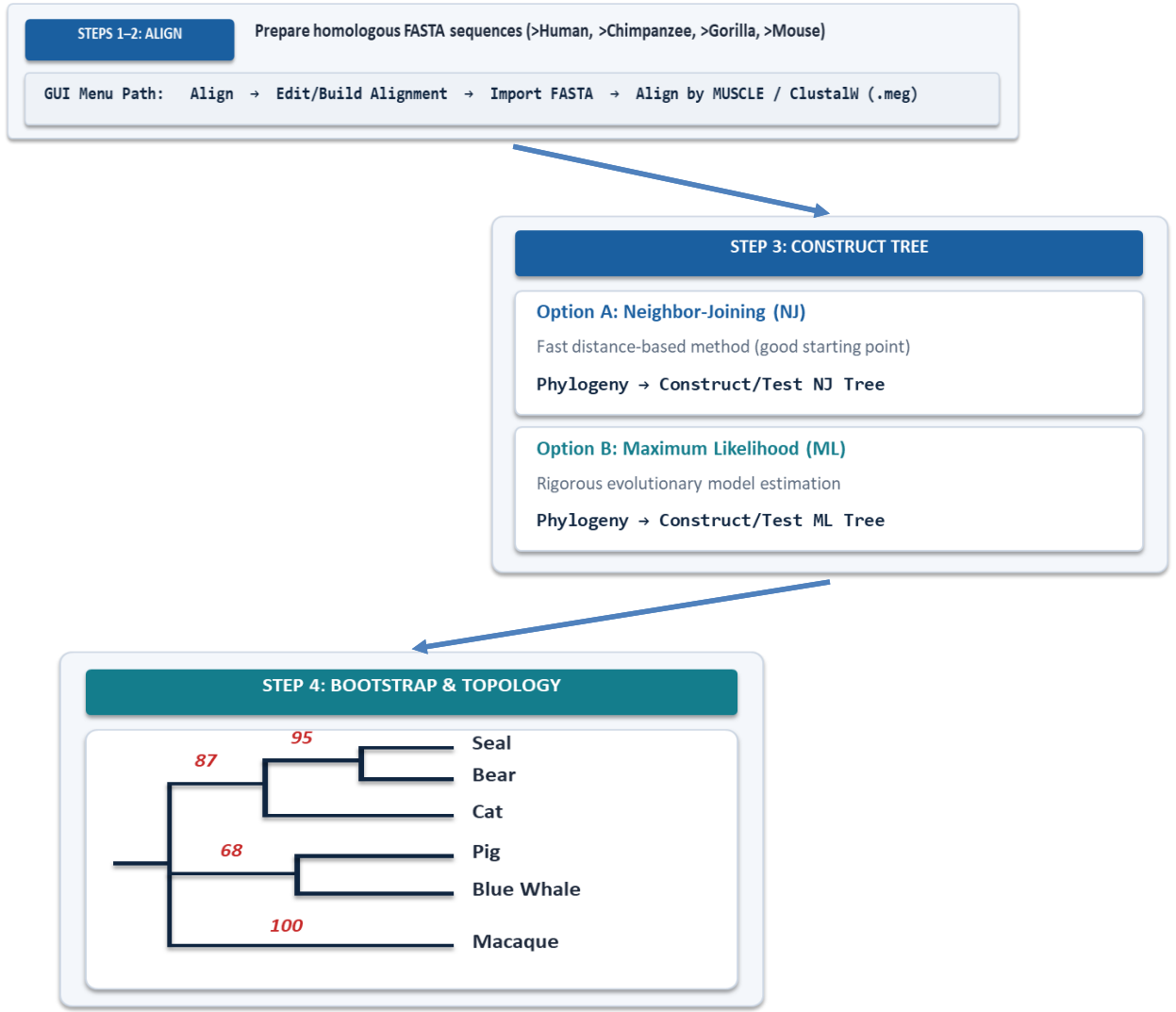


MEGA-GPT: Query Example

3. MEGA-GPT : Phylogenetic Workflow Guidance

Test Task: How do I build a phylogenetic tree in MEGA?

- **Architecture:** RAG-augmented ChatGPT grounded in official MEGA software documentation & protocols.
- **Core Function:** Does NOT execute code directly — instructs users through multi-step evolutionary analyses.
- **Example Output:** Provides step-by-step UI navigation guides, parameter recommendations.
- **Key Advantage:** Reduces LLM hallucinations and allows updating the knowledge base without retraining the model.



Systematic Comparative Analysis

Systematic Comparative Analysis of the Studied Tools

Comparison Criterion	ExpasyGPT	STRING-Chat	MEGA-GPT
Functional Role	**"Translator" (Plain language → SPARQL code)**	**"Translator" (Plain language → MCP tool calls)**	**"Instructor" (Plain language → Workflow guide)**
Core Architecture	RAG + LLM + SPARQL generation	LLM Orchestrator + MCP Server	RAG-augmented ChatGPT on documentation
Target Resource / DB	SIB Knowledge Graphs (UniProt, Bgee, OMA, STRING)	STRING Database (Protein interaction networks)	MEGA Software (Phylogenetic analysis)
Primary Output	Executable SPARQL queries & structured tables	Aggregated network data & visual PPI graphs	Step-by-step protocols & ASCII phylogenetic trees
Biomedical & Drug Discovery Value	Federated multi-omics & gene expression querying	Reconstructing disease networks & finding hub targets	Accurate evolutionary & sequence conservation analysis

Conclusions & Future Directions

1. Key Conclusions

- **Lower Barrier:** Intuitive access to complex databases without coding.
- **Two Roles:** "Translators" (queries & graphs) vs. "Instructor" (workflows).
- **High Utility:** Bridges raw biological data and researcher questions.

2. Current Challenges

- **Query Accuracy:** Ensuring 100% valid & reproducible SPARQL code.
- **Ontology Alignment:** Harmonizing terms across different databases.
- **Data Heterogeneity:** Integrating diverse multi-omics formats.

3. Drug Discovery Prospects

- **Unified Platforms:** Combining RAG and MCP in multi-agent systems.
- **Target Discovery:** Fast-tracking hub gene & pathway identification.
- **Clinical Translation:** From molecular networks to pharmacotherapy.