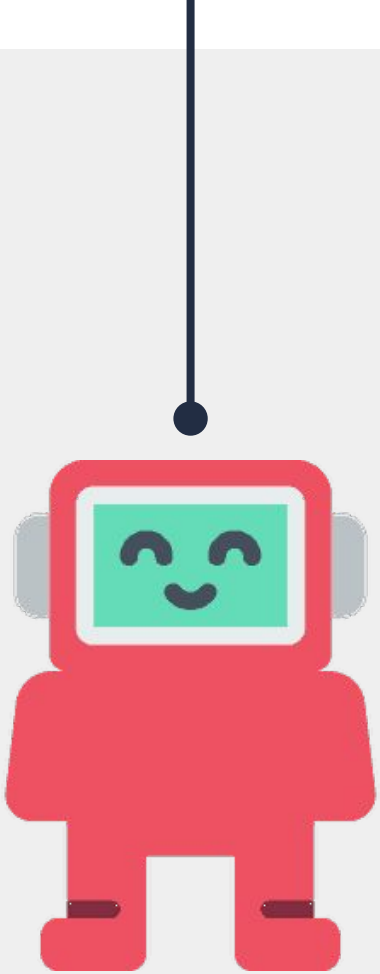
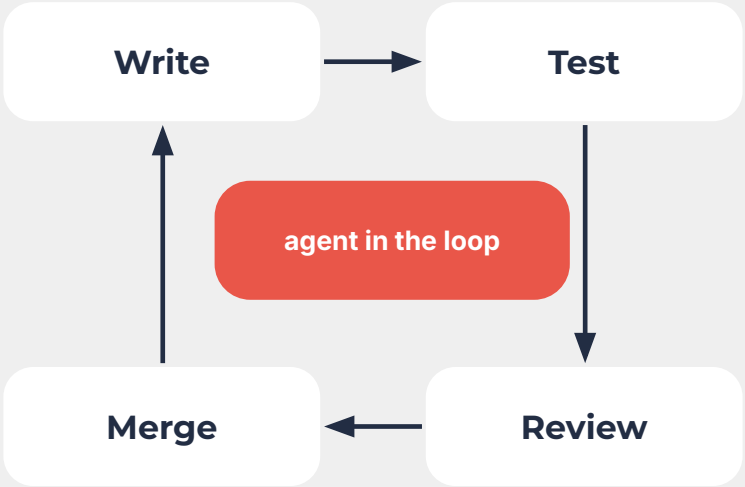
Scientific research using AI

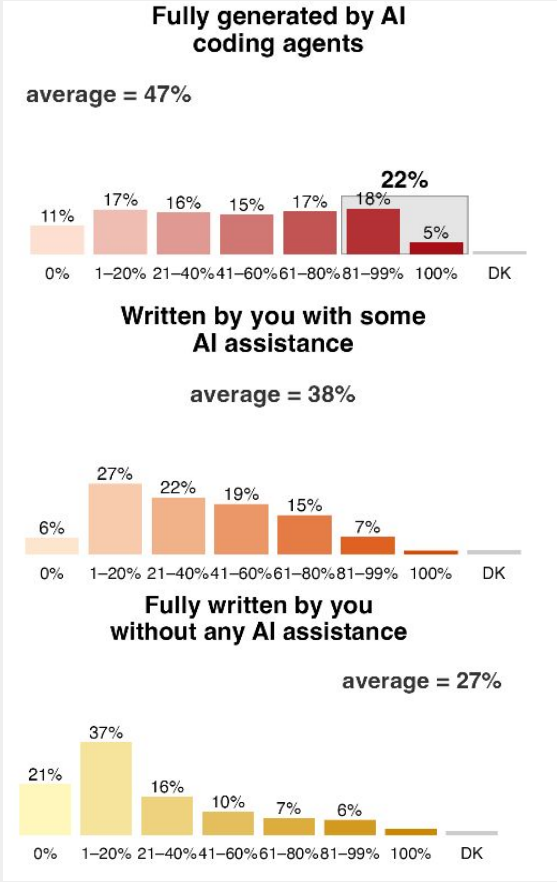
Letting AI agents do research, without losing control

Tomáš Pavlík, 2026



Developers trust agents





Study by Mikhail Bogdanov, JetBrains Research, 8/26

Researchers don’t trust agents

	SOFTWARE	RESEARCH
A bug looks like	a crash or a red test	**a plausible, wrong number**
You find out	in seconds	months later, from a reviewer
Undo means	git revert	a correction, or a retraction
Ground truth	the spec and the tests	biology, controls, statistics
Usual habits	tests, CI, review, version control	scripts in a folder, "it ran"
The data	usually yours to send anywhere	often sensitive; and enormous

“The most dangerous error does not crash. It looks like a result”

Researchers don’t trust agents

"If it is wrong, it is my name on the paper."

"It makes things up."

"Our data cannot leave the institute."

"I would not know where to start."

Only a small fraction of researchers use LLM agents - Petr Kulhánek’s poll at NCBR seminar 2026

Researchers don’t trust agents

The gap is know-how, not technology

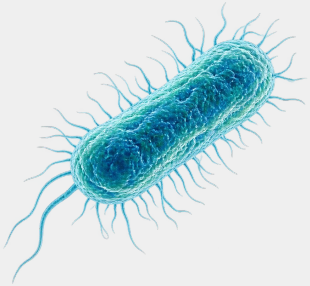
until then, research work remains mostly manual

Researchers don’t trust agents

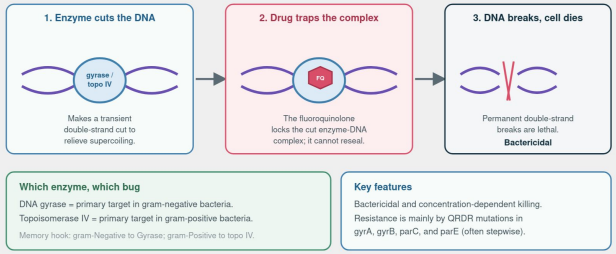
How to bridge it?

lead by example, share experience, build confidence

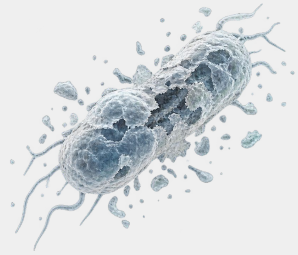
Antibiotics-resistant bacteria:
Fluoroquinolones case study

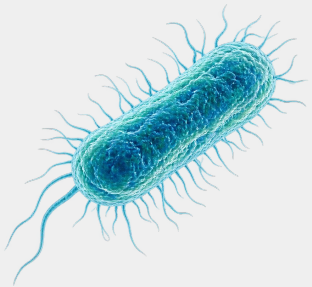






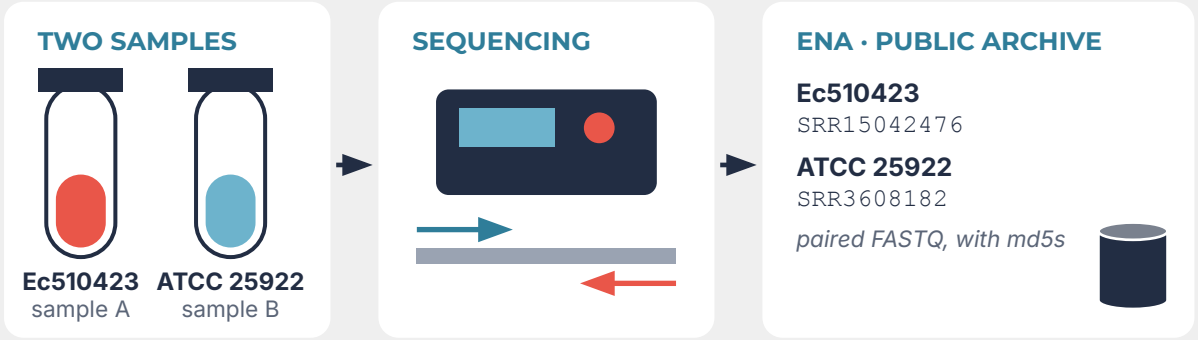






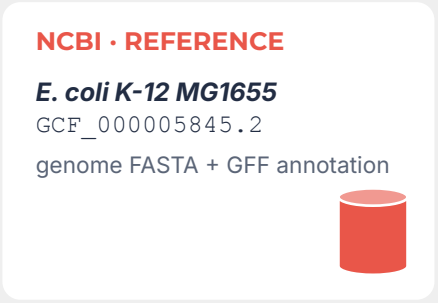
antibiotics-resistant

Antibiotics-resistant bacteria:
Fluoroquinolones case study



The reference: the classic lab sample everyone compares against.





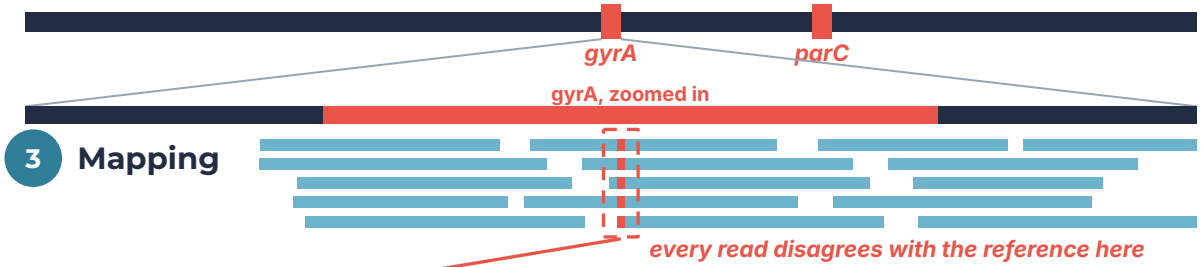
Antibiotics-resistant bacteria:
Fluoroquinolones case study

1 Reads



2 Reference genome

E. coli, 4.6 million letters



3 Mapping

4 Annotating: what does the change do?

reference	...	T	C	G	...	→ Ser · S
this isolate	...	T	T	G	...	→ Leu · L

codon 83 of gyrA
a classic resistance mutation: S83L

5 The call

- **Resistant**
- Susceptible
- Inconclusive

It’s completely free!

Czech national infrastructure and open-source tools

HARNESS

OpenCode

AGENTS

OpenAgentsControl

everything is open-source

MODELS

e-INFRA CZ

open-weight models, available to all researchers; **no data leaves the country**

COMPUTE

MetaCentrum

PBS jobs through an MCP connector

Giving agents autonomy

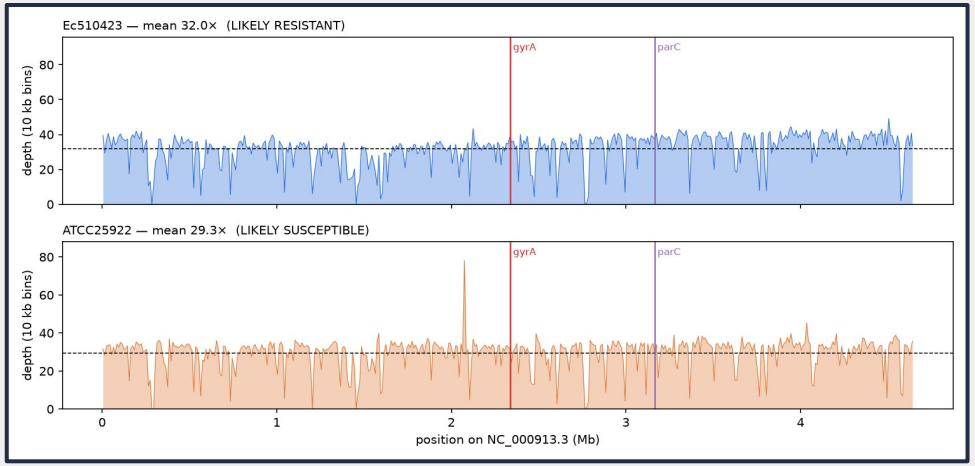


Giving agents autonomy

```
Create visualization of Ec510423 and ATCC25922 reads coverage on
NC_00913.3

Build · glm-5.3 LiteLLM · high

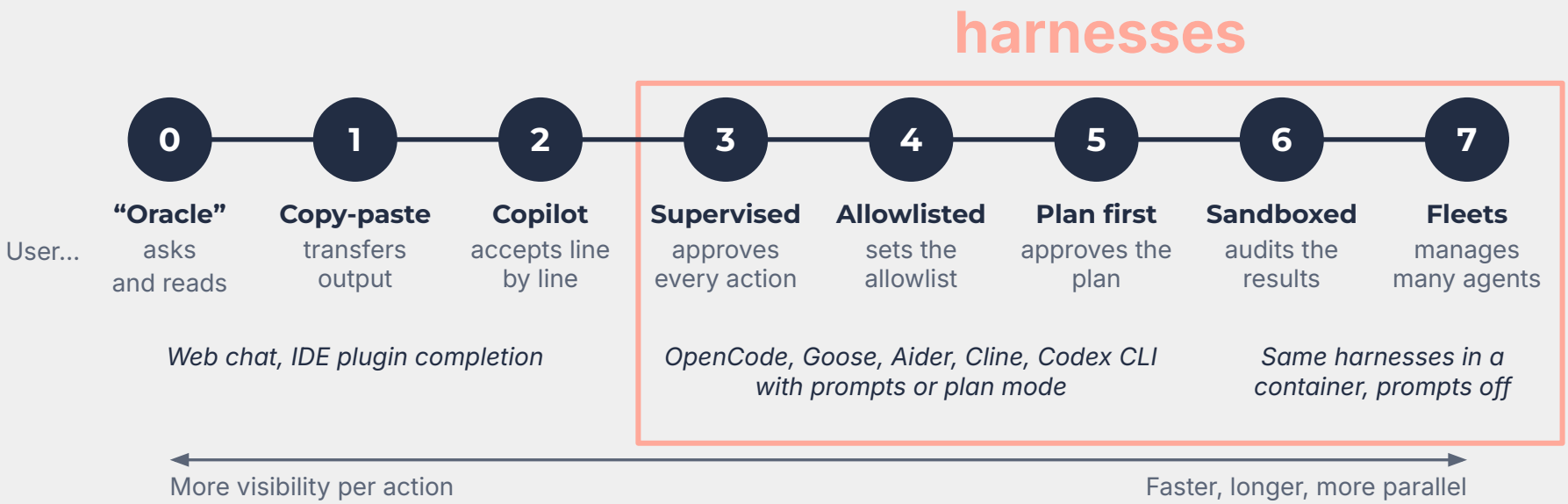
~/work/agentic_workshop_bioinf          shift+tab agents  ctrl+p commands
```



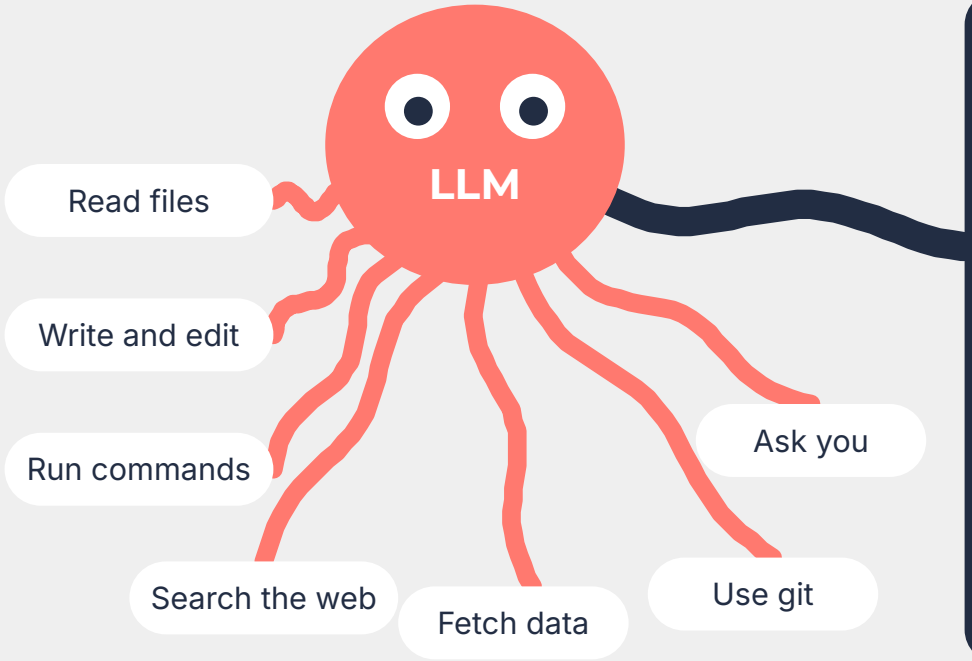
Autonomy = spectrum from doing to auditing



Autonomy = spectrum from doing to auditing



The model talks; the harness gives it arms



THE ARM THAT MATTERS

Cluster connection

- **Data:** hundreds of GBs or TBs (!!)
- **Compute:** calculations take days, many CPU/GPU-hours, lots of RAM
- **Scale:** heavy parallelization

PROMPT: Use the metacentrum MCP tools to test the cluster connection

```
+ Thought · 9.1s

I'll call both tools now — connection_status first, then cluster_info.

✓ execute
› metacentrum.connection_status
› metacentrum.cluster_info

+ Thought · 15.4s

Connection is healthy. Here's the summary:

- MetaCentrum username: tomsko (principle tomsko@META, Kerberos ticket valid for ~8 more hours)
- Cluster working directory (project workspace): /storage/praha5-elixir/projects/bioinf-fi/tomsko —
  relative paths in all jobs/tools resolve here, with 149T free of 320T

A few extras worth noting:

- Frontend: skirit.grid.cesnet.cz, connected via a persistent SSH master (no active jobs or interactive
  sessions right now)
- Home: /storage/brno2/home/tomsko (readable, but do work in the workspace)
- Default queue: elixircz@pbs-m1.metacentrum.cz (defaults: 1 CPU, 4 GB mem, 1 h walltime; caps: 16
  CPUs, 64 GB, 200 GB scratch, 5 active jobs)
- A shared workshop conda env (samtools, bcftools, bwa-mem2, fastp, seqkit, etc.) is auto-sourced at
  the start of every batch job and interactive shell.
```

PROMPT:

We are starting a bacterial resequencing project on MetaCentrum. The question: does E. coli isolate Ec510423 carry fluoroquinolone-resistance mutations in gyrA/parC? ATCC 25922 is the susceptible control. Reference: E. coli K-12 MG1655, NCBI `GCF_000005845.2`. Paired-end reads in ENA: Ec510423 = `SRR15042476`, ATCC25922 = `SRR3608182`.

Step 1, research and plan, change nothing yet. Research how this analysis is usually done and what tends to go wrong. Then propose a project layout and a step list. Start from this layout and change it only with a reason: `data/ref` reference + annotation, `data/reads` raw reads, `scripts` every script we run (jobs must be rerunnable without you), `qc results logs`, plus README.md (question, samples, layout) and PROGRESS.md (a checklist of steps). Show me the proposal and wait for my approval.

Step 2, after I approve: using the metacentrum MCP tools, create `ecoli_fq` in my cluster workspace with the agreed layout. Show me the resulting tree. Do not download anything yet.

Step 3: fetch the inputs into `ecoli_fq/data` as a PBS batch job (not on the frontend): reference as genomic FASTA + GFF (use datasets from the shared env); reads from ENA (verify FASTQ md5s against ENA). Write `scripts/fetch.sh`, submit it, wait for it to finish. Paths: MCP jobs start in the workspace root, so the submitted body must export `PBS_O_WORKDIR=<abs path of ecoli_fq>`; keep plain `cd "$PBS_O_WORKDIR"` in the saved script so `qsub scripts/fetch.sh` works from ecoli_fq/. Don't guess formats: inspect the datasets zip (`unzip -l`) before copying FASTA/GFF; look up fastq_ftp/fastq_md5 in ENA's TSV by header (accession is column 1). The script writes md5 report, R1/R2 read counts and a FASTA/GFF seq-ID check into `logs/` and exits non-zero on any mismatch.

Step 4: verify from logs + listing: files non-empty, md5s match, R1 == R2, seq IDs equal. No trimming. Log all job ids (incl. failures) in PROGRESS.md and summarize.

If anything fails twice or does not match, stop and report what you saw, what you expected, and what you tried.

What the project is about

Structure, tools

Steps, next: fetch data

PROMPT:

We are starting a bacterial resequencing project on MetaCentrum. The question: does E. coli isolate Ec510423 carry fluoroquinolone-resistance mutations in gyrA/parC? ATCC 25922 is the susceptible control. Reference: E. coli K-12 MG1655, NCBI `GCF_000005845.2`. Paired-end reads in ENA: Ec510423 = `SRR15042476`, ATCC25922 = `SRR3608182`.

Step 1, research and plan, change nothing yet. Research how this analysis is usually done and what tends to go wrong. Then propose a project layout and a step list. Start from this layout and change it only with a reason: `data/ref` reference + annotation, `data/reads` raw reads, `scripts` every script we run (jobs must be rerunnable without you), `qc results logs`, plus README.md (question, samples, layout) and PROGRESS.md (a checklist of steps). Show me the proposal and wait for my approval.

Step 2, after approval: use the metacentrum-MCP skills, create `ecoli_fq` in my cluster workspace, and write the data-fetching tree. Do not download anything yet.

Step 3: fetch the datasets zip (on the frontend); run a small PBS job (not on the frontend): read the datasets from the shared env); reads from ENA (verify FASTQ md5s against ENA). Write `scripts/fetch.sh`, submit it, wait for it to finish. Paths: MCP jobs start in the workspace root, so the submitted body must export `PBS_O_WORKDIR=<abs path of ecoli_fq>`; keep plain `cd "$PBS_O_WORKDIR"` in the saved script so `qsub scripts/fetch.sh` works from ecoli_fq/. Don't guess formats: inspect the datasets zip (`unzip -l`) before copying FASTA/GFF; look up fastq_ftp/fastq_md5 in ENA's TSV by header (accession is column 1). The script writes md5 report, R1/R2 read counts and a FASTA/GFF seq-ID check into `logs/` and exits non-zero on any mismatch.

Step 4: verify from logs + listing: files non-empty, md5s match, R1 == R2, seq IDs equal. No trimming. Log all job ids (incl. failures) in PROGRESS.md and summarize.

If anything fails twice or does not match, stop and report what you saw, what you expected, and what you tried.

Don’t be afraid of long prompts!

Speech-to-text

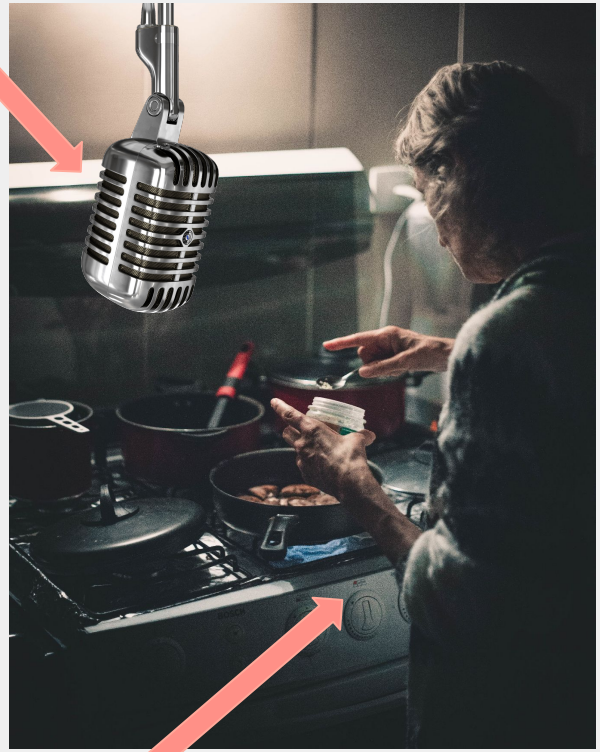


Illustration of me, cooking

PROMPT:

We are starting a bacterial resequencing project on MetaCentrum. The question: does E. coli isolate Ec510423 carry fluoroquinolone-resistance mutations in gyrA/parC? ATCC 25922 is the susceptible control. Reference: E. coli K-12 MG1655, NCBI `GCF_000005845.2`. Paired-end reads in ENA: Ec510423 = `SRR15042476`, ATCC25922 = `SRR3608182`.

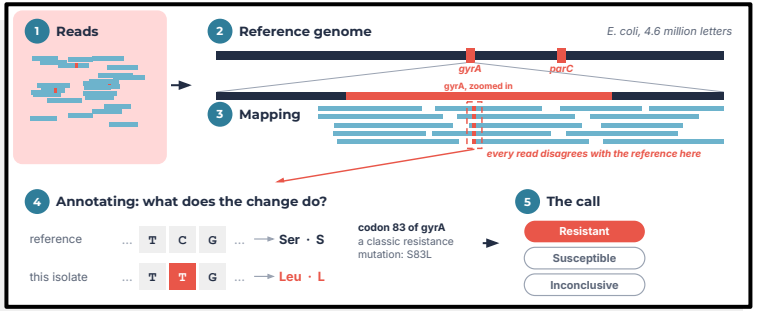
Step 1, research and plan, change nothing yet. Research how this analysis is usually done and what tends to go wrong. Then propose a project layout and a step list. Start from this layout and change it only with a reason: `data/ref` reference + annotation, `data/reads` raw reads, `scripts` every sc (jobs must be rerunnable without you), `qc results logs`, plus RE (question, samples, layout) and PROGRESS.md (a checklist of step the proposal and wait for my approval.

Step 2, after I approve: using the metacentrum MCP tools, create e my cluster workspace with the agreed layout. Show me the resultin not download anything yet.

Step 3: fetch the inputs into `ecoli_fq/data` as a PBS batch job (n frontend): reference as genomic FASTA + GFF (use datasets from t env); reads from ENA (verify FASTQ md5s against ENA). Write `scripts/fetch.sh`, submit it, wait for it to finish. Paths: MCP jobs workspace root, so the submitted body must export `PBS_O_WORKDI path of ecoli_fq>`; keep plain `cd "$PBS_O_WORKDIR"` in the s so `qsub scripts/fetch.sh` works from ecoli_fq/. Don't guess fo inspect the datasets zip (`unzip -l`) before copying FASTA/GFF; lo fastq_ftp/fastq_md5 in ENA's TSV by header (accession is column script writes md5 report, R1/R2 read counts and a FASTA/GFF seq- into `logs/` and exits non-zero on any mismatch.

Step 4: verify from logs + listing: files non-empty, md5s match, R1 IDs equal. No trimming. Log all job ids (incl. failures) in PROGRESS. summarize.

If anything fails twice or does not match, stop and report what you you expected, and what you tried.



Summary — Step 1 complete ✅

Job: `ecoli_fq_fetch` (25136442.pbs-m1) — exit 0, 3 min 48 s (after 3 debugged attempts; all fixes are baked into `scripts/fetch.sh`, so a manual `qsub scripts/fetch.sh` now reproduces the whole step).

What's in ecoli_fq/data/

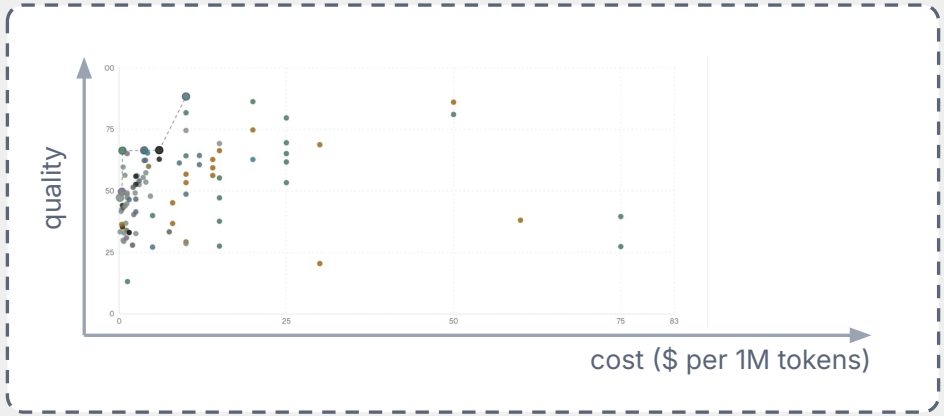
File	Size	Check
`ref/GCF_000005845.2_genomic.fna`	4.5 MB	non-empty; seq ID `NC_000913.3`
`ref/GCF_000005845.2_genomic.gff`	2.2 MB	identical seq ID set — **consistent with FASTA ✅**
`reads/SRR15042476_1/2.fastq.gz` (Ec510423)	144.4 / 180.9 MB	md5 **OK** vs ENA; **934,965 pairs**, R1=R2 ✅
`reads/SRR3608182_1/2.fastq.gz` (ATCC 25922)	155.7 / 155.4 MB	md5 **OK** vs ENA; **1,855,729 pairs**, R1=R2 ✅
`reads/md5_report.txt` + `logs/fetch_verify_*.txt`	—	full evidence trail in the workspace

Model, agent, and a harness

HARNESS

AGENT

MODEL
predicts the next token



benchlm.ai/llm-price-performance

quality · speed · context window · cost · where it runs

Model, agent, and a harness

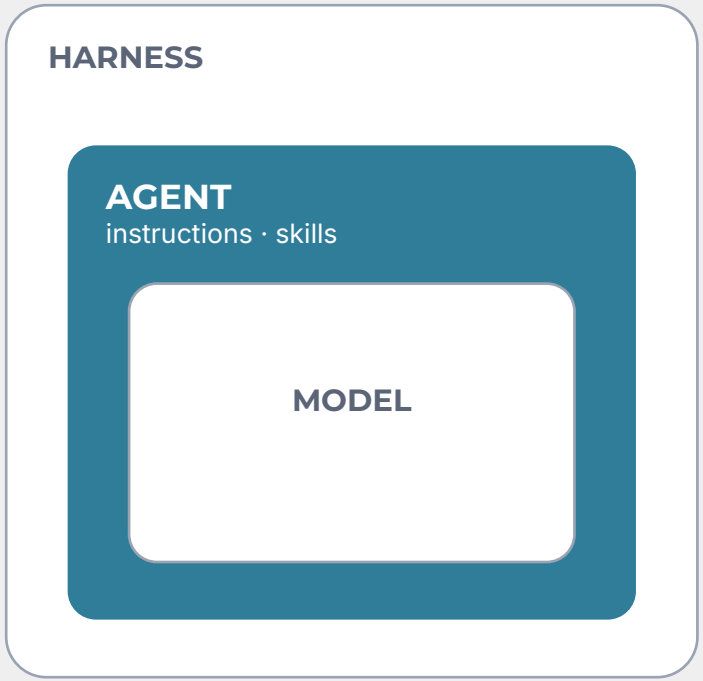


ChatGPT

e-INFRA CZ

Claude

Model, agent, and a harness



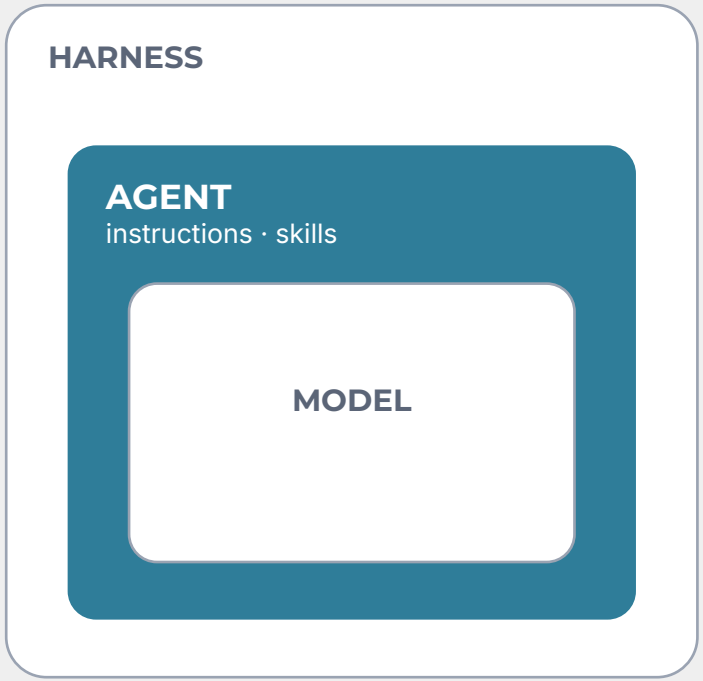
DEFINITION · who it is `agents/analyst.md`

```
---
name: bioinfo-analyst
description: runs and checks variant analyses
model: <your university-hosted model>
tools: read, write, bash, grid-mcp
permissions: allow *
---
You are a careful bioinformatics analyst.
Never print raw FASTQ, SAM or VCF files.
Log every decision in NOTES.md.
```

SKILL · loaded when needed `submit-pbs-job/SKILL.md`

```
---
name: submit-pbs-job
description: use when a step needs real compute
---
1. Fill the PBS template; cap at 2 h and 8 CPUs
2. Submit; log the job ID in PROGRESS.md
```

Model, agent, and a harness



DEFINITION · who it is `agents/analyst.md`

```
---
name: bioinfo-analyst
description: runs and checks variant analyses
model: <your university-hosted model>
tools: read, write, bash, grid-mcp
permissi
---
You are a careful bioinformatics analyst.
Never print raw FASTQ, SAM or VCF files.
Log every decision in NOTES.md.
```

modifies behaviour

SKILL · loaded when needed `submit-pbs-job/SKILL.md`

```
---
name: submit-pbs-job
descri
---
1. Fill the PBS template; cap at 2 h and 8 CPUs
2. Submit; log the job ID in PROGRESS.md
```

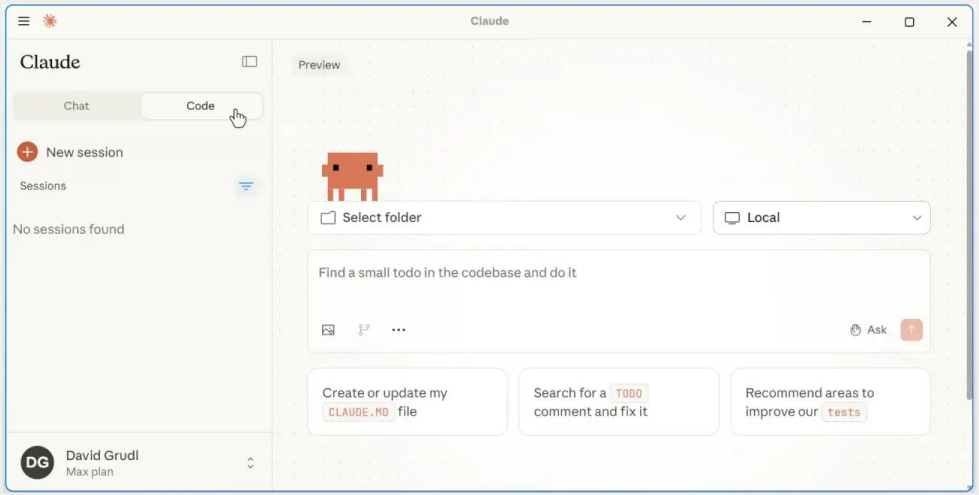
ensures reproducibility

Model, agent, and a harness

HARNESS
loop · tools · permissions · context

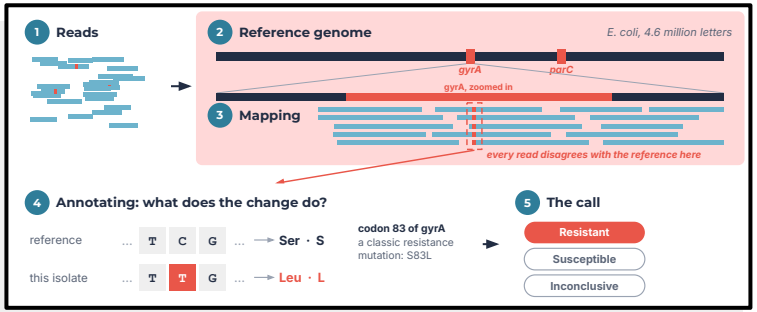
AGENT

MODEL



CONCEPT 2 · MODEL ≠ HARNESS

PROMPT: @align-reads Align both samples (Ec510423, ATCC25922) to the reference.



Alignment QC table

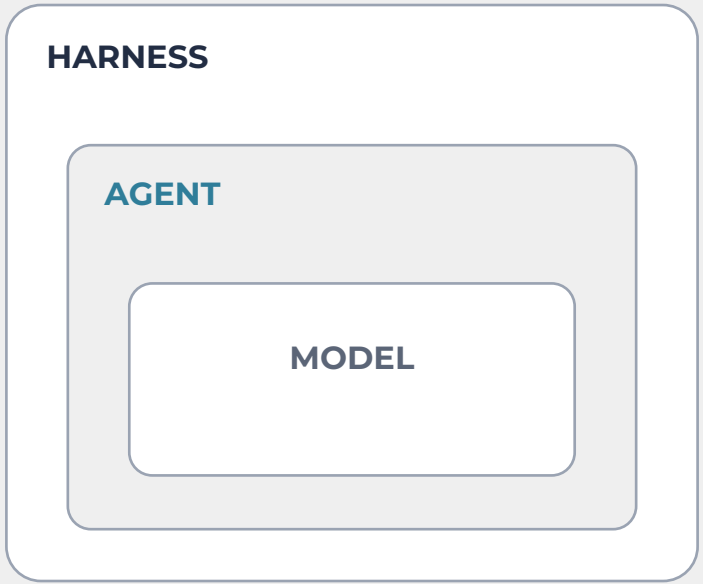
sample	total reads	% mapped	% properly paired	% dup	mean depth	% genome ≥10×	planned depth
Ec510423	574,854	68.31	67.38	0.31	10.46×	71.78%	~27.4×
ATCC25922	894,477	68.09	66.28	0.27	18.54×	90.31%	~28.9×

(Sanity: alignment input counts match the post-fastp read counts exactly — 570,644 / 891,482 primary — so the aligner used the right, synced QC reads.)

Interpretation — the striking thing is the *equality*, not the depth

- **Both samples map at ~68%**, and cleanly so: of mapped reads, ~97% are properly paired, singletons 0.07–0.76%, duplicates 0.3% (unsurprising after subsampling from only 28–29×), no secondary mappings. The alignments that *happen* are textbook-quality.
- **My key discriminator fired against the contamination hypothesis:** ATCC25922 is the *bona fide* susceptible control — if a pure reference-quality *E. coli* also maps at 68.09%, then 68% is what "this pipeline + this reference" yields for our data, and Ec510423's identical 68.31% is *not* evidence of sample-side

Making agents more capable



1. **Tools**
   deep research
2. **MCP servers**
   MetaCentrum grid, Playwright
3. **Plugins**
   CC-Safety-Net
4. **Agent description**
5. **Skills**
6. **Permissions, questions**
   allow, ask, deny

Making agents more capable

PROMPT: Do a deep research of the relevant publications available online:

Does E. coli isolate Ec510423 carry fluoroquinolone-resistance mutations in gyrA/parC

1. **Tools**
   deep research
2. **MCP servers**
   MetaCentrum grid, Playwright
3. **Plugins**
   CC-Safety-Net
4. **Agent description**
5. **Skills**
6. **Permissions, questions**
   allow, ask, deny

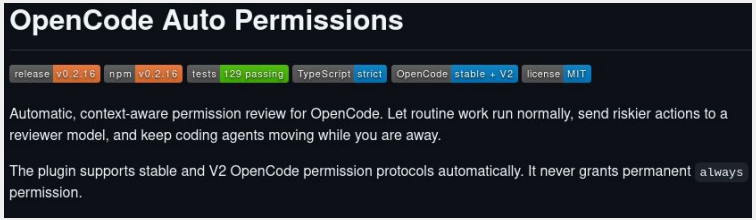
Making agents more capable

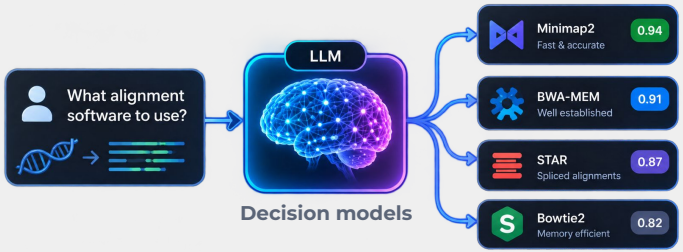


https://github.com/kenryu42/cc-safety-net

1. **Tools**
   deep research
2. **MCP servers**
   MetaCentrum grid, Playwright
3. **Plugins**
   CC-Safety-Net
4. **Agent description**
5. **Skills**
6. **Permissions, questions**
   allow, ask, deny

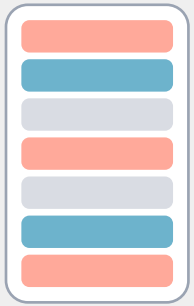
Making agents more capable



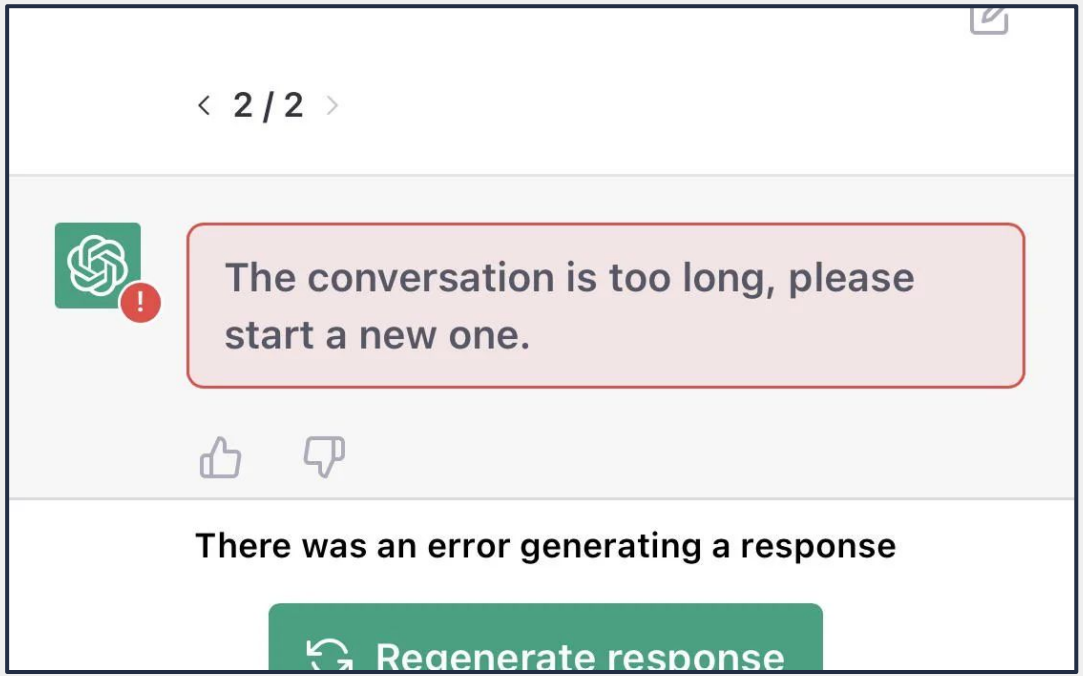


1. **Tools**
   deep research
2. **MCP servers**
   MetaCentrum grid, Playwright
3. **Plugins**
   CC-Safety-Net
4. **Agent description**
5. **Skills**
6. **Permissions, questions**
   allow, ask, deny

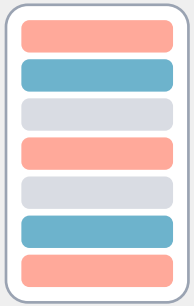
Making agents smarter



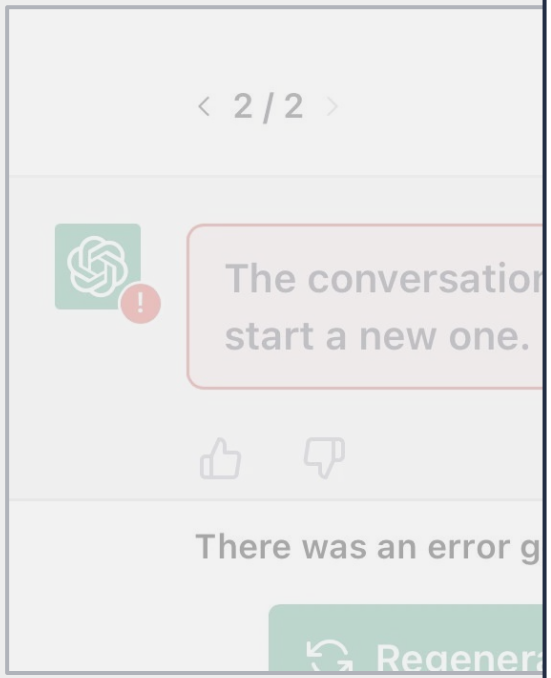
Long session

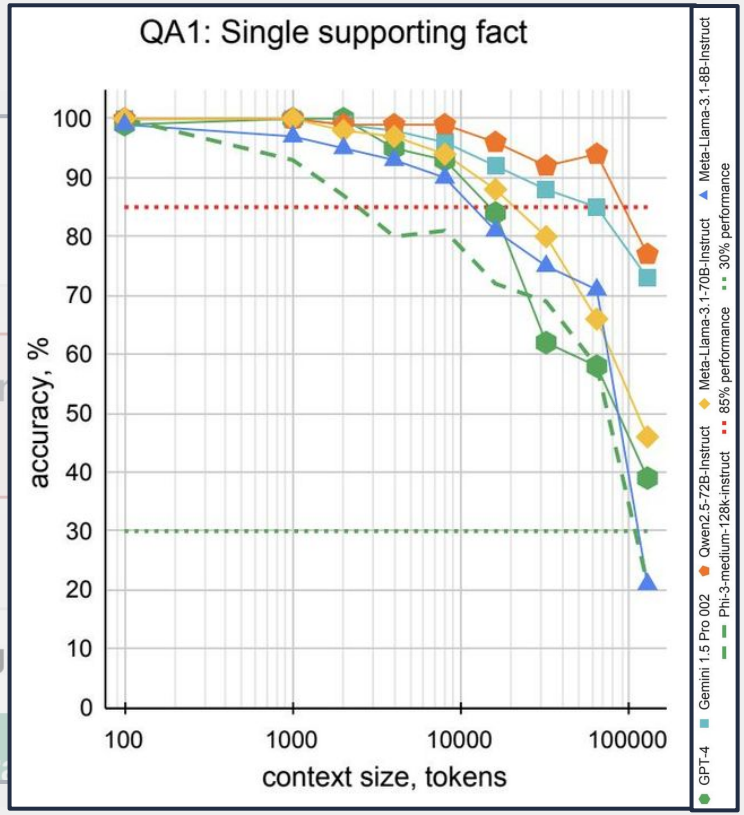


Making agents smarter



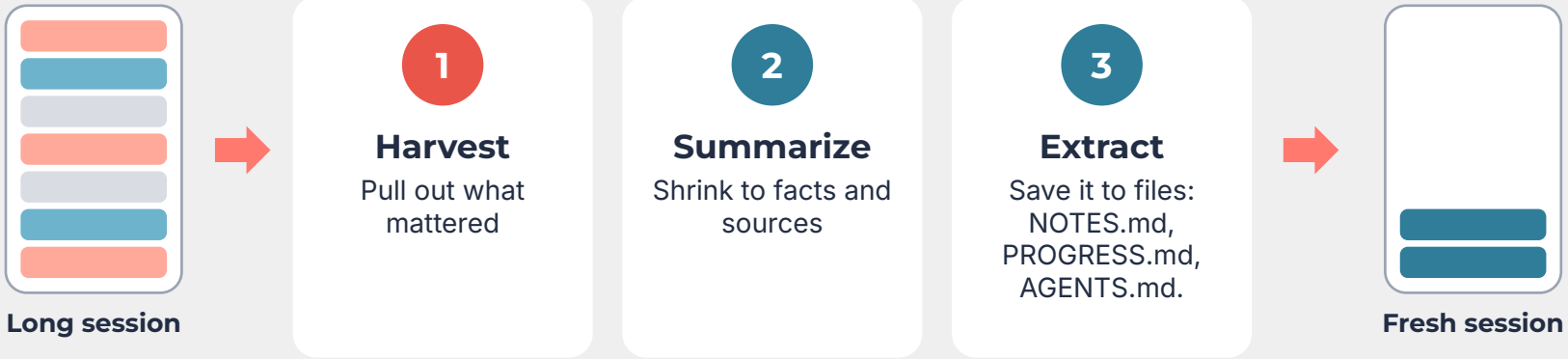
Long session





Kuratov et al., 2024

Making agents smarter



Compact context often
Keep it about ⅓ of maximum context window (⅓ → ⅔ is usually low attention zone)

Polish & clean up the documentation
Prevent stale information from piling up

In research, context is your lab diary: lose it and you repeat the same attempts.

Making agents smarter

Structure: a frame, then freedom

YOU SET, AT THE START	THE AGENT DECIDES
high-level structure	folder names
conventions	script layout
rules	file naming

person = very specific way of thinking
model = average of millions of people

every restriction = `translation overhead`

you: `aln_v2_FINAL/` the model expects: `results/align/`

Long-term memory

MANUALLY PER PROJECT

- NOTES.md, or a memory/ folder
- In git, so every change is visible
- What every future session should know

PLUGINS



https://github.com/tickernelz/opencode-mem

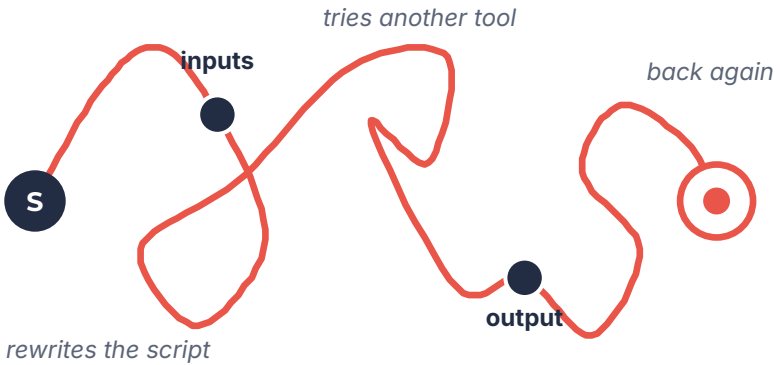
Personalized Benchmarking: Evaluating LLMs by Individual Preferences, Garbacea et al., 2026

Staying on track



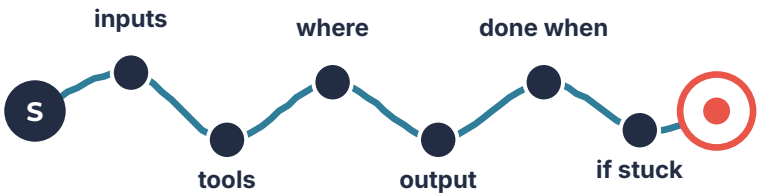
Staying on track

SPARSE MAP



Two anchors. Detours, loops, quiet guesses.

DENSE MAP



Six anchors. Project description, agent's research, back-and-forth with data

Every run is full of decisions: yours, or its: make the ones that matter on your terms.

Staying on track



Plan it (together)

Agree on the **steps and checkpoints** before anything runs.

Make each one **detailed and concrete**

LLMs tend to please and force results



Tests it can't edit

Write the **tests** and **expected results** before the run.

Use **positive and negative controls**.

A **failing test means stop** and talk to you. Never “fix” the test.

Biological data can be forgiving… and misleading



Version everything

A **git commit** at every checkpoint.

Each run in its own folder, with its command and tool versions.

Be able to roll back without much damage

Staying on track

Plan it (together)

Agree on the **steps and checkpoints** before anything runs.

Make each one **detailed and concrete**

LLMs tend to please and force results

Tests it can't edit

Write the **tests** and **expected results** before the run.

Use **positive and negative controls**.

A **failing test means stop** and talk to you. Never “fix” the test.

Biological data can be forgiving… and misleading

Version everything

A **git commit** at every checkpoint.

Each run in its own folder, with its command and tool versions.

Be able to roll back without much damage

Staying on track

Plan it (together)

Agree on the **steps and checkpoints** before anything runs.

Make each one **detailed and concrete**

LLMs tend to please and force results

Tests it can't edit

Write the **tests** and **expected results** before the run.

Use **positive and negative controls**.

A **failing test means stop** and talk to you. Never “fix” the test.

Biological data can be forgiving… and misleading

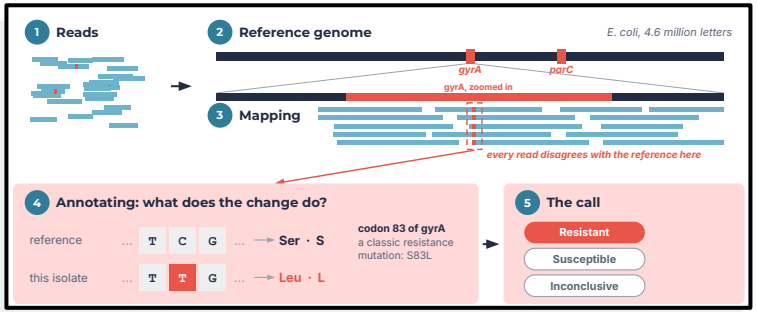
Version everything

A **git commit** at every checkpoint.

Each run in its own folder, with its command and tool versions.

Be able to roll back without much damage

PROMPT: Call variants, create subset of those in quinolone-resistance-determining regions and using rules from project's start decide which sample is resistant. Make plots auditing every step.



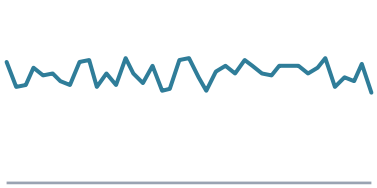
Final step complete ✅

Decisions (mechanically applied from DECISION.md — the sign-off line in results/report.md awaits you):

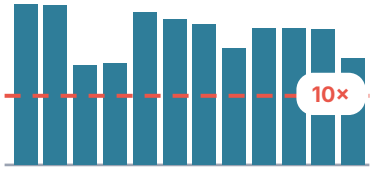
sample	decision	determinants	gates
Ec510423	likely resistant	gyrA S83L, gyrA D87N, parC S80I — all AF=1.000, haploid	3/3 pass
ATCC25922	likely susceptible	none — 7 QRDR SNPs all silent (incl. S80S at a key codon)	3/3 pass

Validity gates (both samples): positive control recovered a planted parC E84G through the exact lookup code; every determinant codon has per-base min depth ≥12× (Ec510423: 12–26×) / ≥13× (ATCC25922: 13–22×); contig name NC_000913.3 consistent across FASTA/GFF/VCFs.

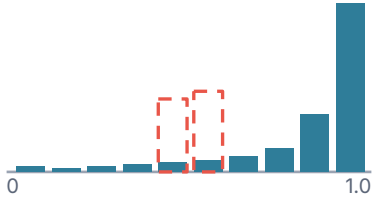
Validation: seeing mistakes



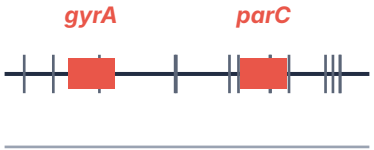
Coverage along the genome



Depth across the QRDRs



Allele frequency of calls



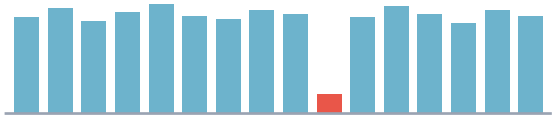
Variants along the genome

Validation: seeing mistakes

Plot every step

Various visualizations for every step of the project. You have domain experience to judge what's off.

The plots come from scripts, not from the chat.



coverage along the genome: **a dip you see at once**

Red-team audit

A fresh session, another model, attacks the work: is the method sound, do the results match the literature?

Ask it to find what is wrong, not to confirm.

"Is this the right method?
Do our numbers match published ones?
What is the strongest case we're wrong?"

Reproducible: one pipeline

From raw data to report in one run: Snakemake, Nextflow or run_all.sh. Data, environment and parameters pinned, all in git.

If it can't rerun without the agent, it isn't finished.

fetch → align → call → report

one command reruns it all

Validation: biology is unforgiving

LLM result
in silico



some of it
wrong

Plot the raw data

Directly from experiments with **minimal normalisation or filtering. Transforms hide errors.**

Orthogonal methods

Check with a different method that has **different blind spots and limitations**

Wet lab

An experiment is crucial: **a wrong claim doesn't reproduce at lab.**

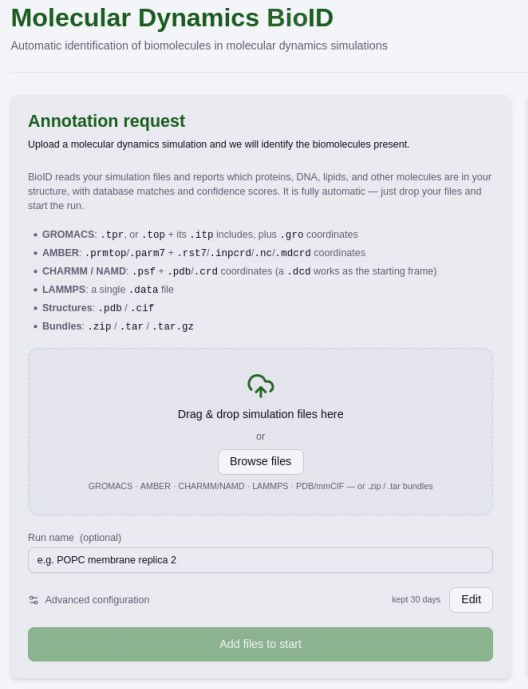
Holds up

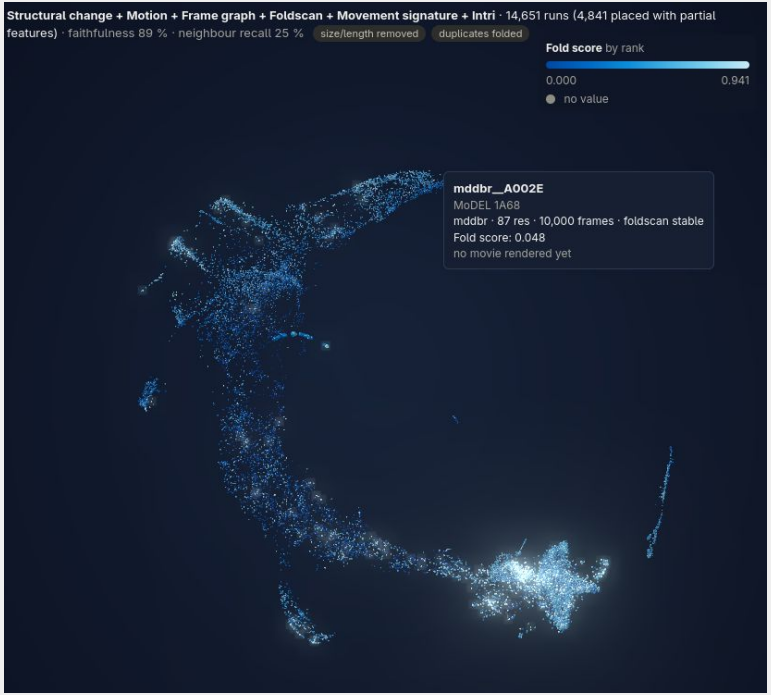


only what biology agrees with

Blindly trusting in-silico results is plainly wrong. But biology is complex, so a wrong result is easier to get caught.

Does it work on large projects?





Yes! if you design it right

Current limitations, and what's in front of us?



scientific reports

View all journals | Saved research

Explore content | About the journal | Publish with us

nature > scientific reports > articles > article

Generative AI lacks the human creativity to achieve scientific discovery from scratch

Article | Open access | Published: **20 March 2025** | Volume 15, article number 9587 (2025) | Cite this article

You have full access to this open access article

Download PDF | Save article

Amy Wenxuan Ding & Shibo Li

Current limitations, and what's in front of us?

ARXIV · 2025 (BIXBENCH)

Blindly failing

53 real bioinformatics analyses with ~300 questions. **Frontier agents of early 2025 answered only 17% of the open questions correctly.**

Mitchener et al., BixBench: a comprehensive benchmark for LLM-based agents in computational biology

NATURE · 2024

The illusion trap

AI tools can make us believe we understand more than we do, and **narrow science to the questions and methods AI handles well.**

Messeri & Crockett, Artificial intelligence and illusions of understanding in scientific research

It's fundamental to know limitations of AI - and put human exactly there

Current limitations, and what's in front of us?

NATURE · JUL 2025

Agents as a lab team

An AI "PI" leads a team of AI scientists; **a human gives high-level feedback**.

Swanson et al., The Virtual Lab of AI agents designs new SARS-CoV-2 nanobodies

NATURE · MAY 2026

The full research loop

Agents read the literature, picked a drug to repurpose for dry macular degeneration, designed the experiments and analysed the results. **Humans did lab work.**

Ghareeb et al., A multi-agent system for automating scientific discovery (Robin)

SCIENCE · JUL 2026

Tools make the agent

A general biomedical agent with tools, software and databases mined from thousands of papers. **Capable of research on its own.**

Huang et al., Autonomous biomedical research with an artificial intelligence agent (Biomni)

From using AI as work assistant to designer of research systems

Takeaway

1 AUTONOMY

Choose autonomy level

More autonomy means more auditing. Build trust by design.

2 MODEL ≠ HARNESS

Chose models appropriately

Know the limitations, advantages, and plan within them.

3 MORE CAPABLE AGENTS

Harness gives LLM arms

MCP servers, skills, plugins, and clear allow / ask / deny rules.

4 CONTEXT

Keep context clean

Compact on purpose. Organize project well. Keep long-term notes.

5 PLAN AND CHECKPOINTS

Plan, check

Concrete steps, tests it can't edit, a commit at every checkpoint.

6 VALIDATION

Plot, audit, rerun

Plots at every step, a red-team audit, one pipeline that reruns.

Thanks!
Q&A

Letting AI agents do research, without losing control
Tomáš Pavlík, 2026

