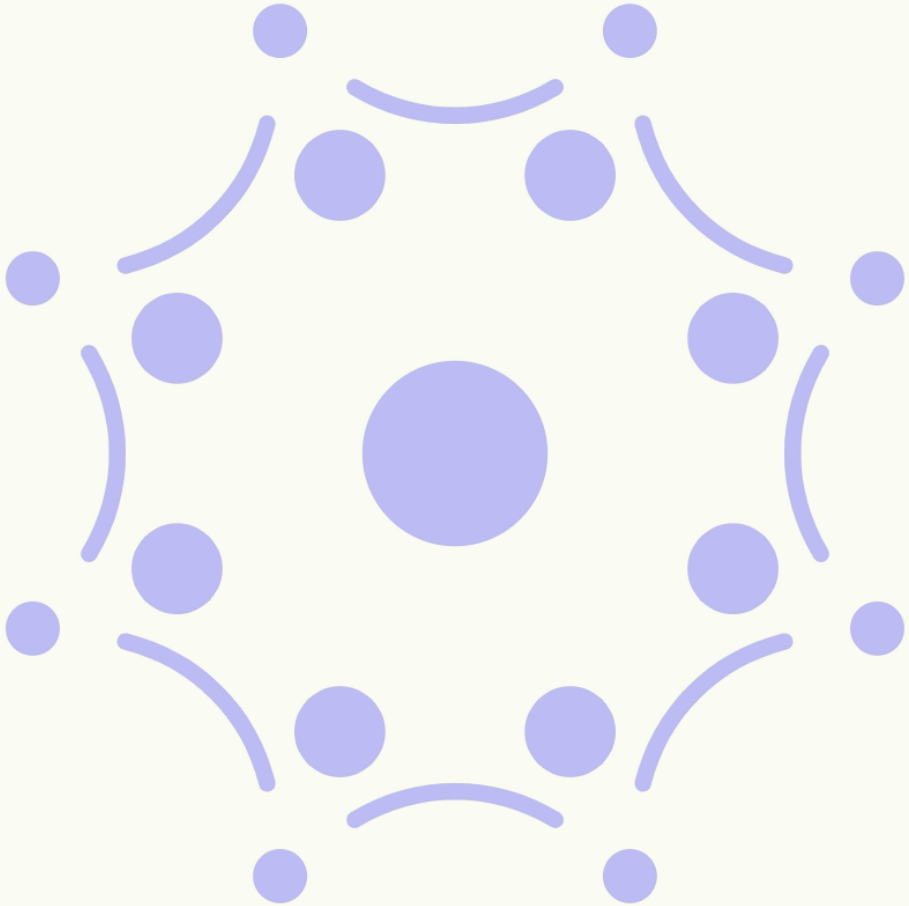
LPAP
status update

Frederik Duus Møller

Center for Genomic Epidemiology, DTU

November 2026



Today

01 What is LPAP?

02 Under the hood

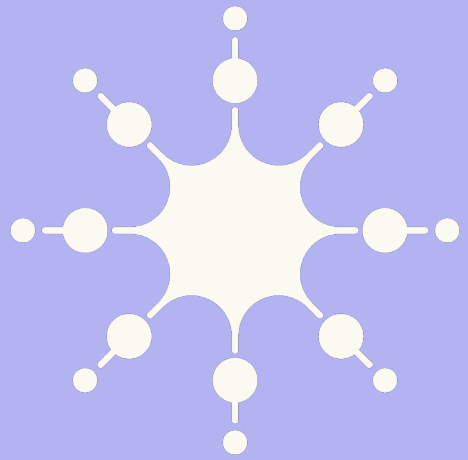
03 First release: v1.0

04 Demonstration

05 What is next

01

What is LPAP?



Local Pathogen Analysis Platform

The challenge



Sequencing went portable

A Nanopore MinION sequencer + laptop can sequence anywhere



The analysis did not

It still needs internet, a terminal, tools, databases and a server.

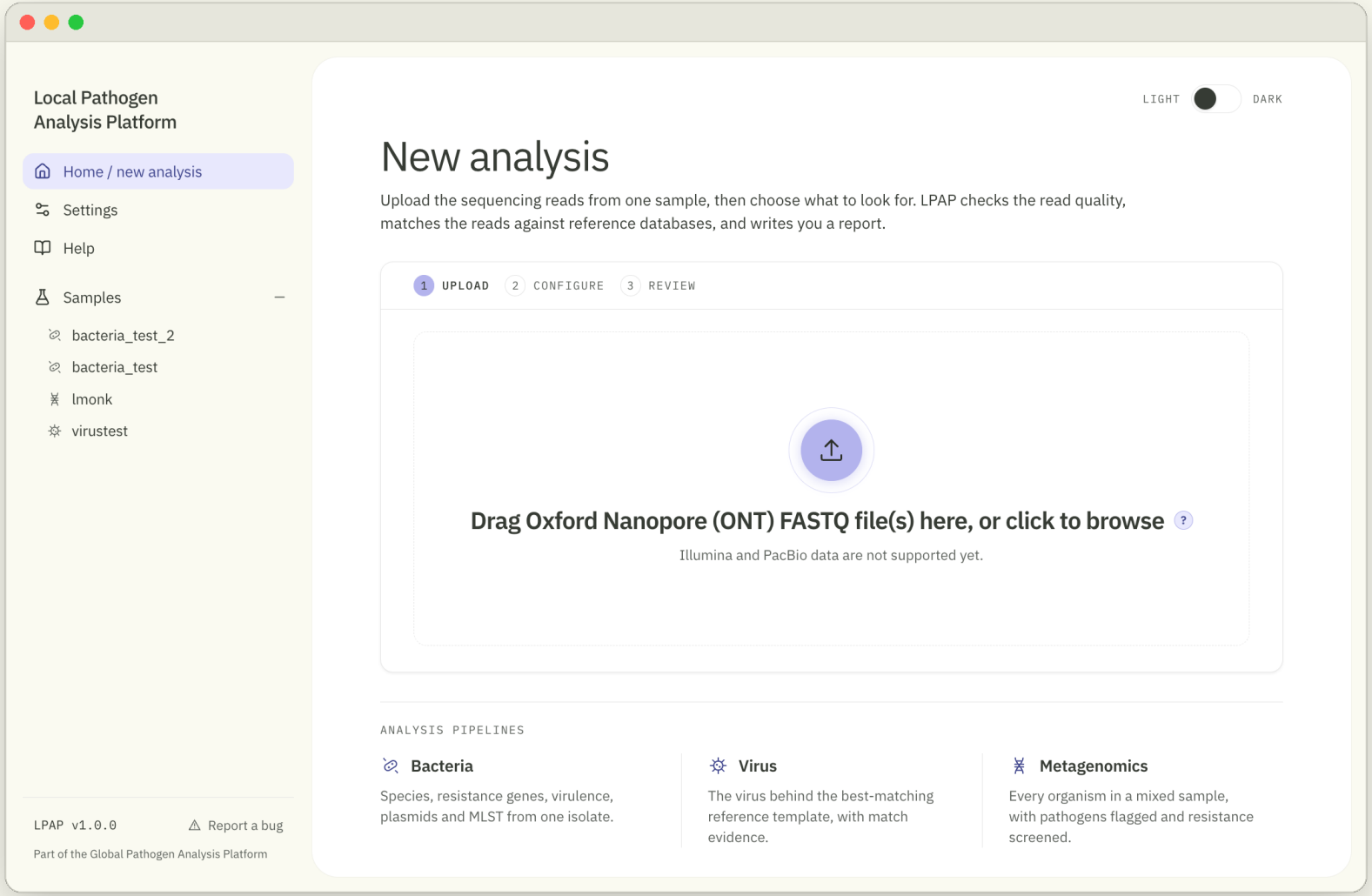


Remote settings make it harder

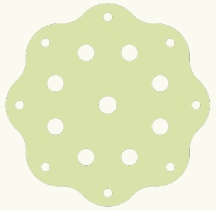
Unstable or no internet, no bioinformatician, data that cannot leave the site.

What is LPAP?

- A desktop app for Oxford Nanopore data
- Apple Silicon Macs and Ubuntu Linux
- Works offline once set up
- No command line, no admin rights required
- From raw data to results in 5 minutes



What can LPAP do?



Three pipelines, one tool

Bacteria, virus and metagenomics, with quality control first by default.



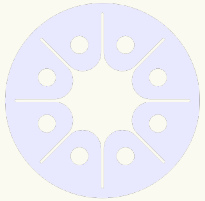
Reports you can share

Interactive HTML. Open in any browser, save as PDF.



AI interpretation

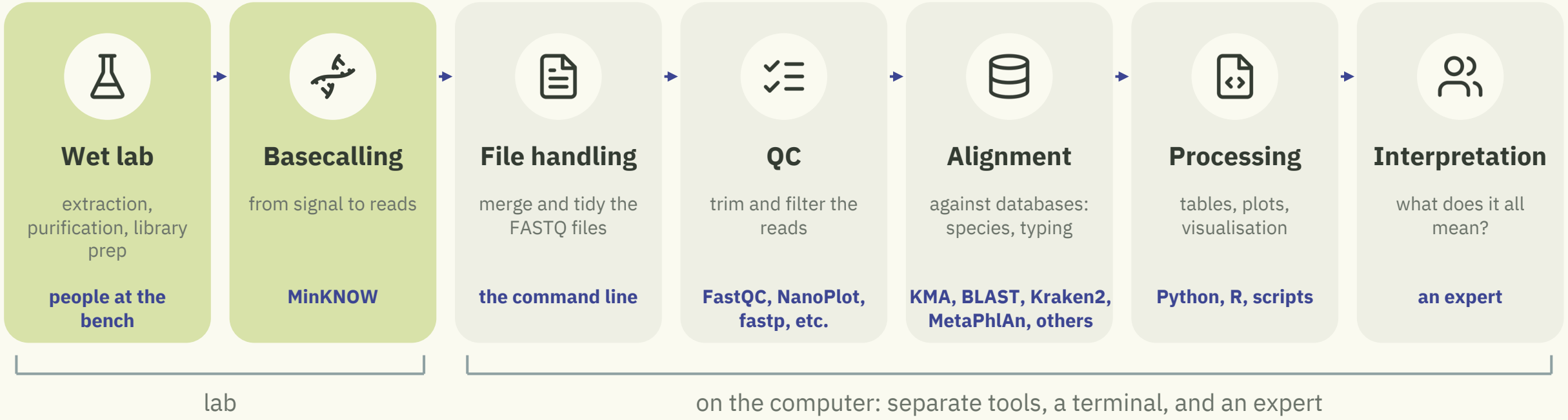
An optional local model explains the report in plain words.



Works completely offline

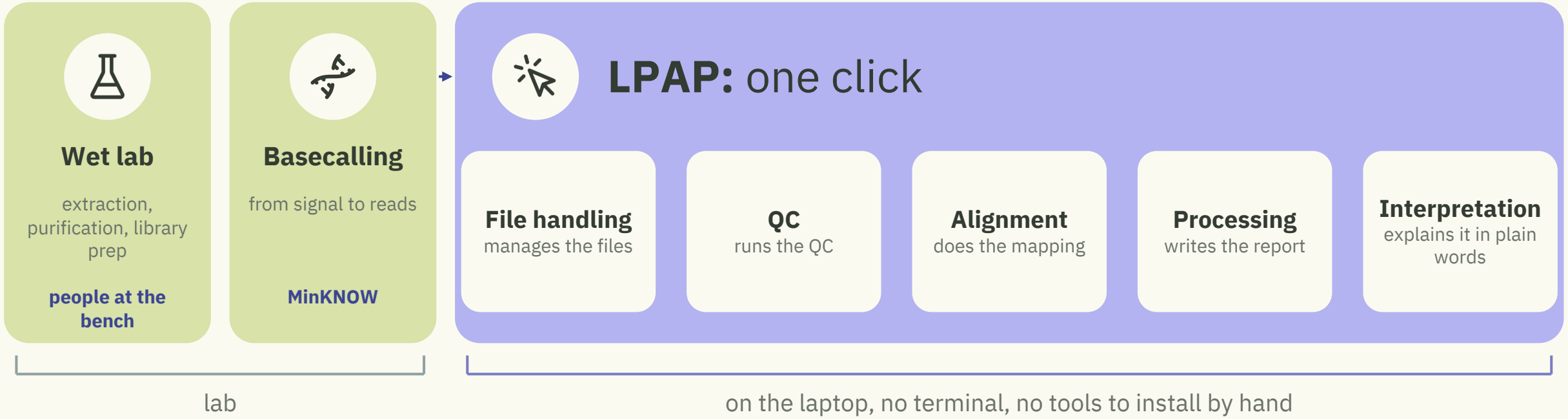
One switch keeps LPAP, and everything it starts, off the network

A typical workflow today



Many steps after basecalling, each requiring software and know-how

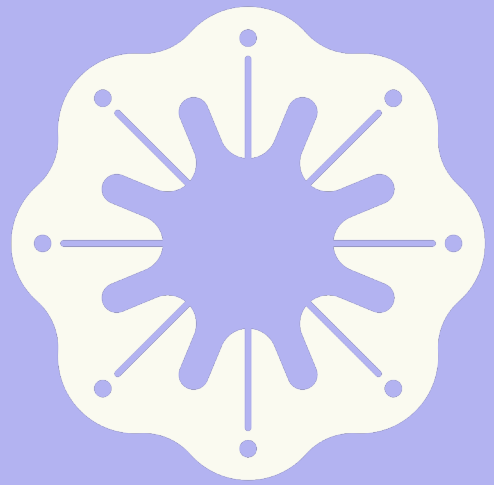
The same workflow, with LPAP



Only one click after basecalling

02

But what is LPAP actually?



A look under the hood, from the bottom up

Three layers, and one extra

LPAP app / UI layer
the desktop application people download

Electron

LPAP-tools / Pipeline layer
pipelines that run the commands in the right order

Python

Bioinformatics / Science layer
KMA and reference databases

(same as CGE services)

+ AI
optional, rides on top

Nothing new, on purpose

KMA aligning against reference databases

Same way SpeciesFinder, ResFinder and MLST do.

```
$ kma -t_db resfinder_db/all -i sample.fq -o amr
    -ont -md 5 -reassign -mem_mode -t 8
```

If you have the same file, same database, same KMA version: the same result, on any machine.

The SpeciesFinder database

Built from NCBI in five parts:

Bacteria

68,871

genomes, complete genomes

Viruses

205,070

genomes, phages included

Archaea

752

genomes, complete genomes

Eukaryotes

164

genomes, parasites and fungi

Human

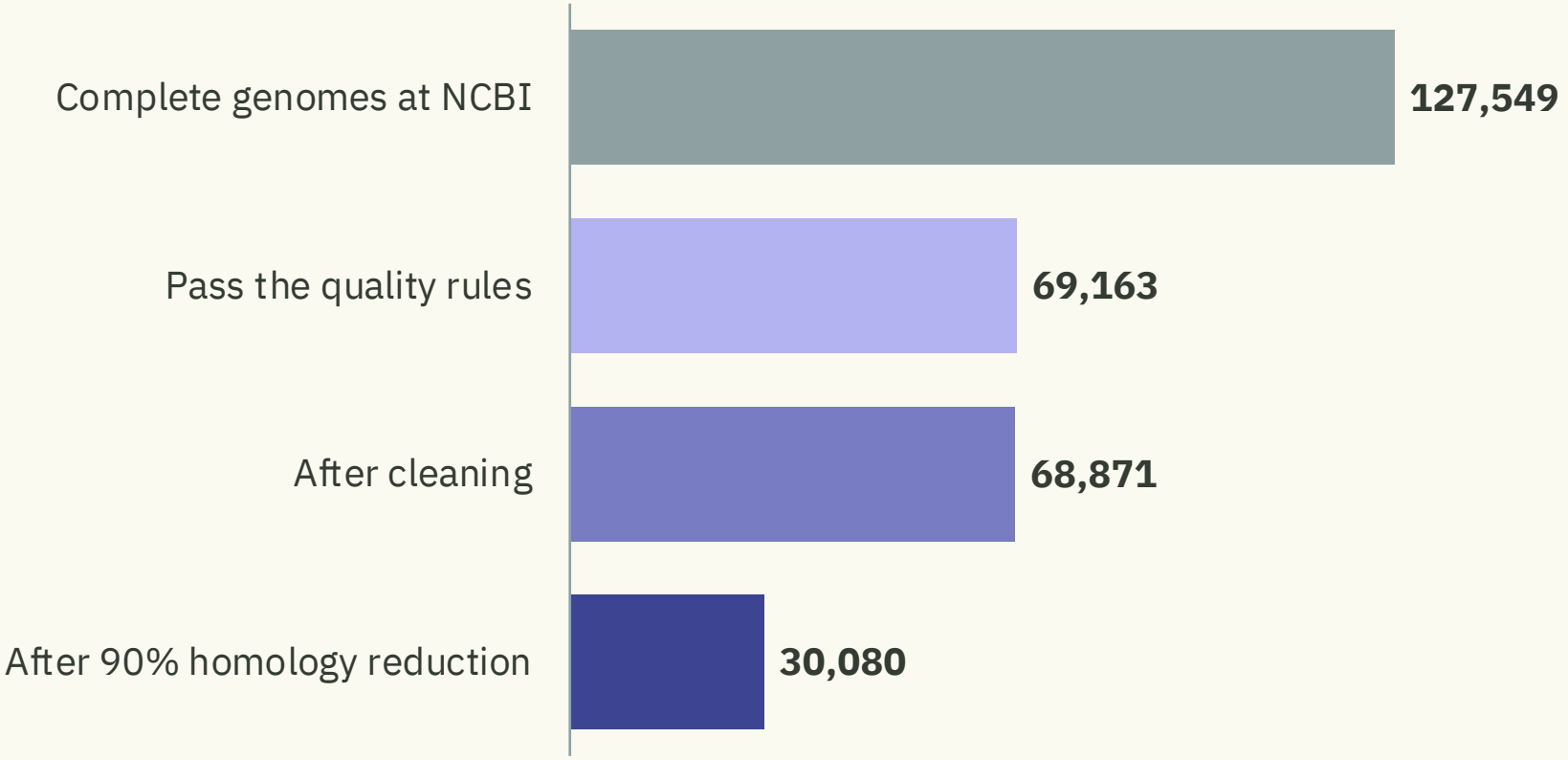
1

genome, to set human reads aside

Two database sizes: Large (AT) for 32 GB memory, Small (ATG) for 16 GB, ~46-74 GB on disk

Bacterial database filtering

September 2026 build, numbers in complete genomes:



Plasmids removed

so a plasmid cannot pass for the species

Near-duplicates merged

one genome stands in for relatives >90% identical

The other databases

ResFinder

3,212

resistance genes

VirulenceFinder

5,630

virulence genes

PlasmidFinder

488

plasmid replicons

MLST

151

typing schemes

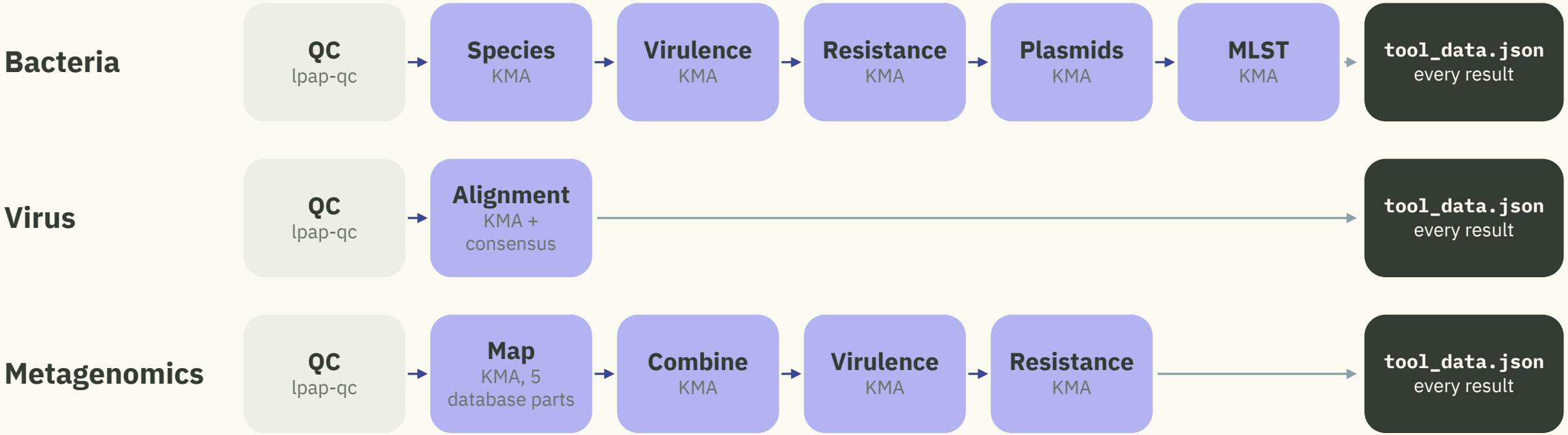
Pathogen watchlist

519 pathogens

439 bacteria, 80 viruses

LPAP-tools

Just a Python pipeline, similar to Nextflow or Snakemake workflow.
It runs the commands in order, and collects what they leave behind



Every step creates KMA output files; the JSON collects them, and **lpap-reports** turns it into the report.

The app

Building a desktop app like LPAP means solving two problems at once.

Good-looking interface

Modern buttons, progress bars, interactive tables and charts.

Browsers are the great tool for this: HTML for content, CSS for looks, JavaScript for behaviour.

Access to the machine

Start Python, run KMA, read databases, write result files.

A web page may not do any of this. It runs in a sandbox, so a random website cannot run programs or delete files.

Electron

One app that is a browser and a program at the same time.

Chromium

the engine inside Google Chrome

solves problem 01:
the interface is a modern web page

+

Node.js

JavaScript outside the browser

solves problem 02:
it can reach files and start programs

=



Electron

Built the same way

You probably have a few Electron apps open right now.

Visual Studio Code

Figma

Discord

Signal

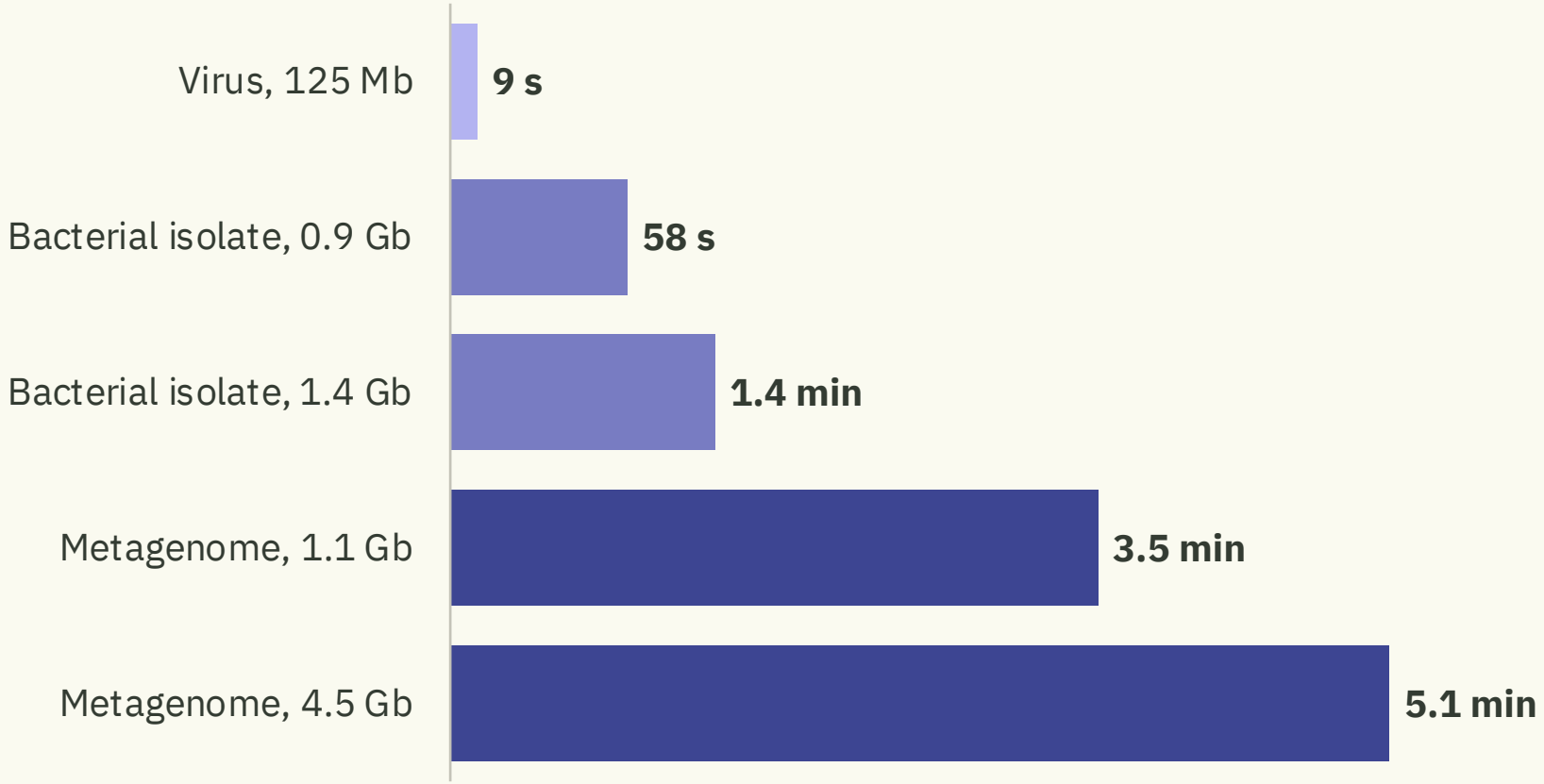
... and LPAP

Electron, and the alternatives

Approach	Pros	Cons
Electron	+ looks identical everywhere: same browser built in + HTML, CSS, JavaScript; mature	– large: about 200 MB per app, has to be kept up to date with Chromium security releases
Tauri	+ tiny; also HTML and CSS, runs on every system	– uses each system's own browser: looks and bugs differ per system
Qt	+ runs on every system	– looks different on each system unless restyled by hand; no HTML or CSS
SwiftUI	+ truly native on the Mac	– Mac only: no Linux, no Windows
Flutter	+ looks the same everywhere	– not HTML: the reports would need a separate built-in browser
A web app	+ nothing to install	– cannot work offline: needs a server, and the data leaves the lab

Timing a LPAP run

Real runs, LPAP 1.0 on a MacBook Pro (M5 Pro, 24 GB)



~3 min

approximate average for a metagenomic sample

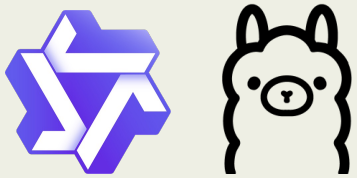
+20 s

for the optional AI explanation

A small model, on the laptop

Qwen3.5 9B (open model, Apache 2.0) running in **Ollama** on this laptop. Optional, and it only writes text.

5.6 GB



model weights download, plus 0.5 GB for Ollama

6–7 GB

memory while it writes

~20 s

per answer on an M-series Mac

Local

nothing leaves the laptop; cloud models are refused

Step 1: the report

1 Report
what was found

↓

2 Packet
findings as words

↓

3 Answer
written, then checked

```
virus alignment, top hits                identity  coverage   depth
Monkeypox virus DRC Yandongi 1985          99.8 %   100.0 %   520 x
Cowpox virus Kostroma_2015                 94.0 %     2.0 %   0.2 x
Cowpox virus HumLit08/1                    90.5 %     3.3 %   0.2 x

data quality: good (Q17.5, 117 Mb)
```

Step 2: the packet

LPAP turns each finding into words, and sends only what actually ran.

1 Report
what was found

↓

2 Packet
findings as words

↓

3 Answer
written, then checked

```
Data quality: good.
- Monkeypox virus DRC Yandongi 1985
    [identity identical, coverage full, depth deep]
- Cowpox virus Kostroma_2015
    [identity related, coverage scattered, depth trace]
```

Why words?
Fewer false statements
Model doesn't try to copy results as numbers

Step 3: the answer

Written on the laptop, checked as it streams, shown as plain text.

1 Report
what was found

↓

2 Packet
findings as words

↓

3 Answer
written, then checked

```
## Organisms
Monkeypox virus (strain DRC Yandongi 1985): Firmly identified based on identical
identity, full coverage, and deep depth. This organism is known to cause monkeypox
disease in humans and animals and is a public health concern due to its potential
for transmission and severity.

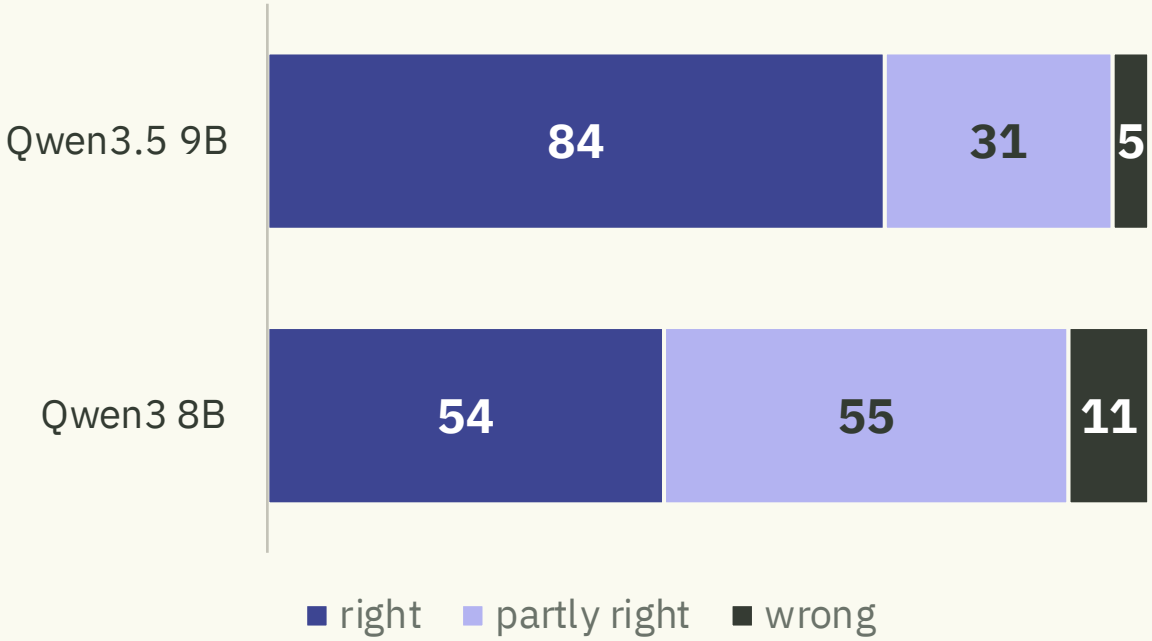
Cowpox virus (strains HumLit08/1 and Kostroma_2015): Weakly detected with related
identity, scattered coverage, and trace depth. These hits are likely artifacts or
shared sequences rather than distinct infections, especially given the strong signal
from Monkeypox virus. No specific clinical significance is attributed to these weak
signals in this context.
```

Checked as it is written

Numbers or names not in the packet get flagged; code or a link stops the answer.

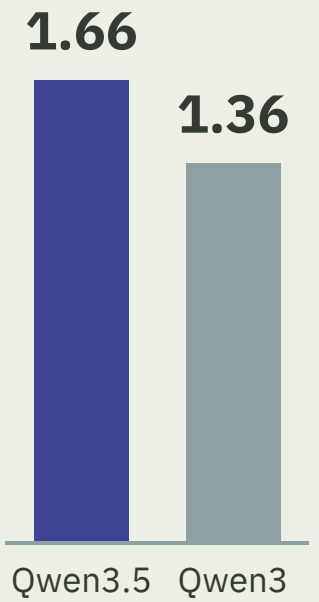
Tested, blind

Main conclusion, number of answers out of 120:



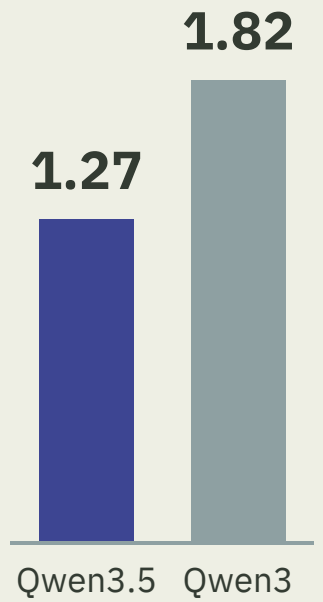
1,140 answers graded across 40 samples and 4 models

Mean grade (0-2)



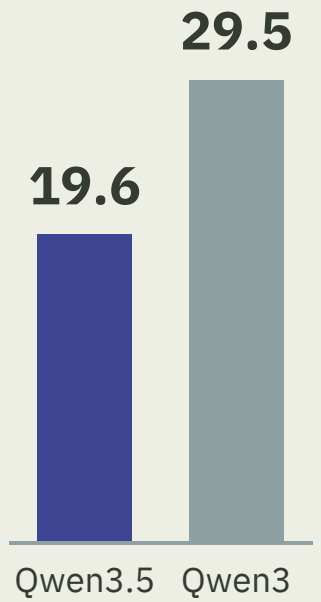
higher is better

Key findings missed



lower is better

Seconds per answer



lower is better

Safe by design

Text in, text out

Cannot run tools, cannot run commands, cannot run code, cannot browse or touch files

Stays on the laptop

All cloud models are refused, only the LPAP local model works

Shown as plain text

No links, no HTML, text planted in a report can only change words

Checked as it writes

Anything that looks like code or a link stops the answer

What it can and cannot do

It can

- ✓ explain findings in plain words
- ✓ say how far the evidence goes
- ✓ flag a signal worth confirming

It cannot

- ✗ act on the computer
- ✗ see the exact numbers
- ✗ diagnose or prescribe (forbidden, and checked)

Main risk: it can be wrong. So it is labelled AI, not a diagnosis, and read next to the report.

03

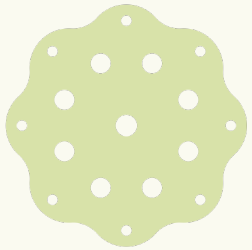
First release: v1.0



What makes it 1.0

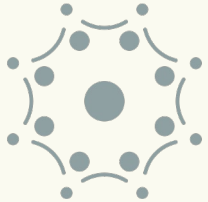
What makes it 1.0

A ”promise” of how LPAP behaves from now on



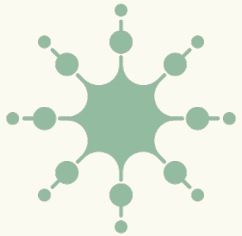
One tested bundle

The tools ship inside the app, pinned to the versions tested. Every 1.0 install runs the same thing



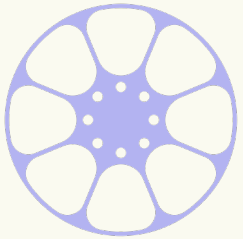
Old results keep working

Every saved file carries a version and is upgraded automatically. A newer LPAP always opens your earlier analyses.



Updates find you

LPAP tells you when a new version is out, and small tool fixes/updates can arrive without a new LPAP.



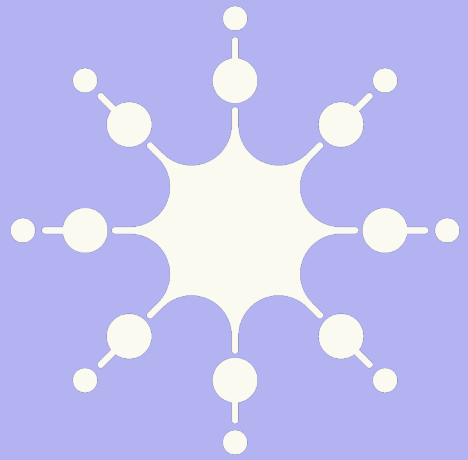
Installs like any app

Signed and notarized on the Mac, a normal package on Ubuntu, now in final testing on clean machines.

Try it, break it, tell me

04

Demonstration



1 Setup finds what is missing

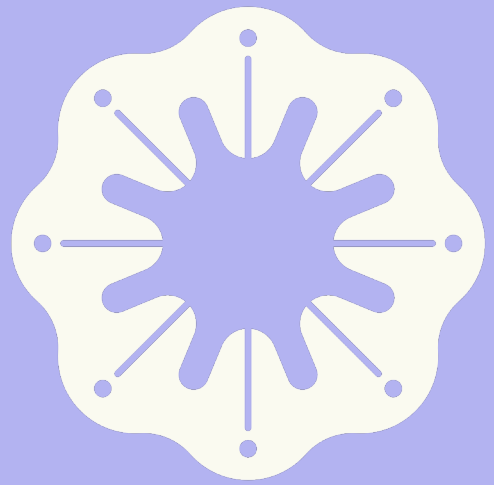
2 The interface

3 Run a sample

4 The reports

05

What is next



New features, and LPAP on the road

Features in the works

01	Application note	the paper describing LPAP, right after 1.0
02	Phylogenetic trees	a tree per template that grows with each sample
03	Salmonella serotyping	automatically, when the species is Salmonella
04	Illumina reads	paired-end input in all pipelines
05	Updating	one-click updates of LPAP itself
06	Collections	save groups of samples and compare them
07	Windows	LPAP on Windows, further out

LPAP on the road

Today

Nov 2026
LPAP 1.0

in final testing on clean machines

Mexico

7–8 Dec 2026
SENASICA, Chihuahua

LPAP introduction

Thailand

late Feb 2027
CU-VET, Chulalongkorn University, Bangkok

co-creator workshop

Senegal

early Apr 2027
Institut Pasteur de Dakar (IPD), Dakar

half-day hands-on with your own data

Thank you
Questions?