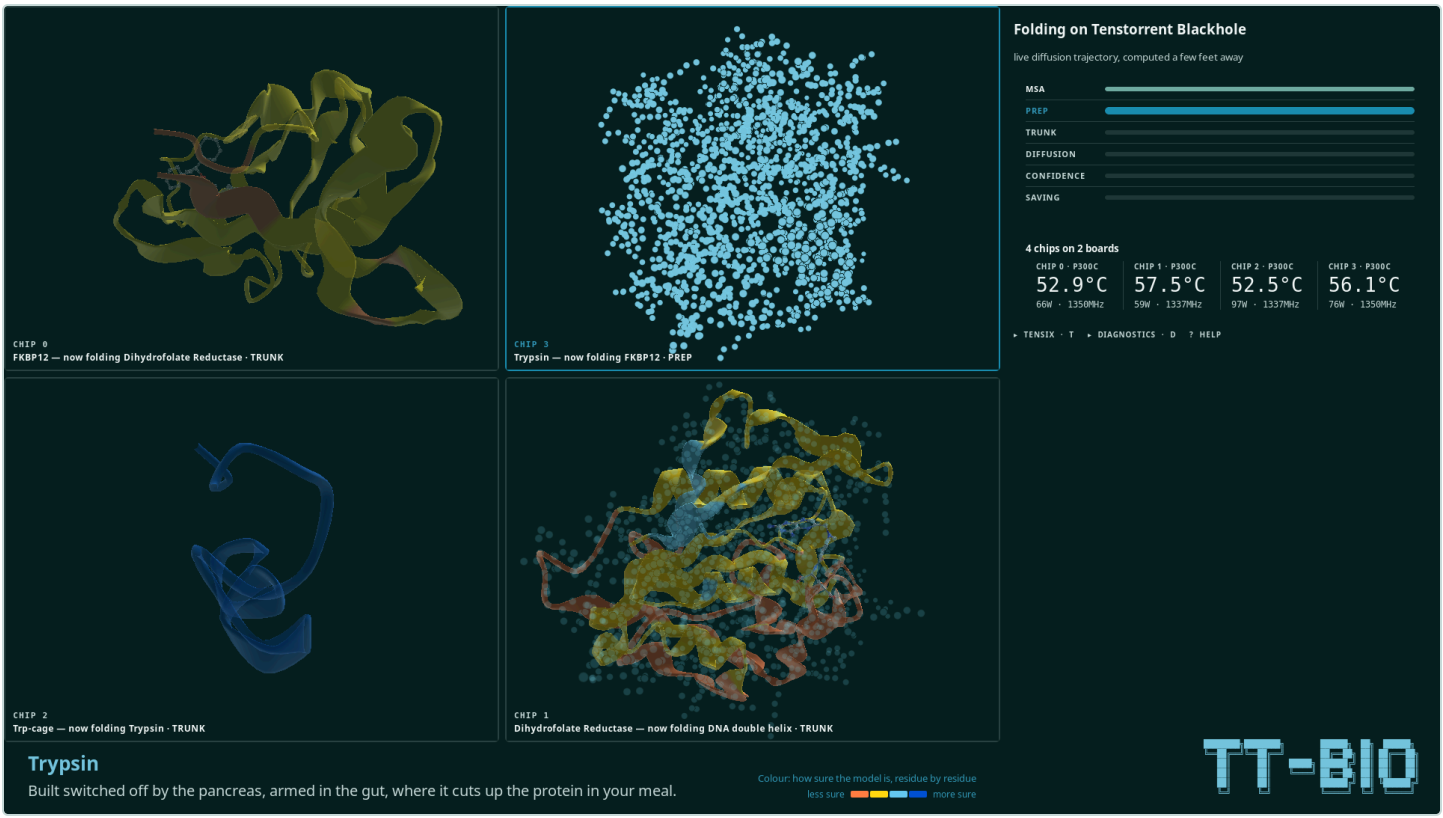
Watch a protein fold

Live on **Tenstorrent Blackhole**
tt-bio-demo · v0.4.0

A protein is a chain of amino acids that only works once it has folded into a specific three-dimensional shape. Predicting that shape from the sequence alone was an open problem in biology for fifty years. This booth does it in front of you — **on the four Blackhole chips a few feet away**, in seconds, with nothing pre-recorded.



The real application, on real silicon. *Four Blackhole chips (two p300c boards), each folding a different molecule at its own point in the pipeline. Structures are drawn as a cartoon — helices as ribbons, sheets as arrows. The loose dots are the model's actual denoising trajectory, streamed one step at a time as it condenses into structure.*

WHAT YOU ARE WATCHING

The model starts from **pure noise** — a cloud of atoms in random positions — and removes a little of that noise at a time, guided by what it has learned about how real proteins are built. Every dot on screen is a genuine intermediate step, sent from the chip as it happens.

Nothing here is an animation of a finished answer. When a fold has no coordinates yet, the booth holds the **previous real structure**, dimmed and labelled, rather than inventing something to show you.

COLOUR = THE MODEL'S CONFIDENCE

very high	confident	low	very low

Each residue is coloured by **pLDDT** — the model scoring its own certainty. Blue regions it is sure about; orange it is guessing. An honest model tells you where it is weak, and this one is shown doing exactly that.

THE SIX STAGES ON THE PANEL

`MSA`	related sequences from other organisms — skipped here, which is why some targets come back orange
`PREP`	builds the model's input features
`TRUNK`	how every residue relates to every other. Makes **no coordinates** — so a big protein shows nothing new at first, and the booth keeps the previous structure up
`DIFFUSION`	the part you watch: noise becomes structure
`CONFIDENCE`	the model scores itself — where the colours come from
`SAVING`	writes the result out as mmCIF

WHAT IT FOLDS

MOLECULE	SIZE	WARM
Trp-cage	20 res	4.2 s
DNA double helix	–	4.4 s
Transfer RNA	–	7.1 s
FKBP12 + binder	107 res	9.6 s
DHFR + methotrexate	187 res	15.4 s
Trypsin + benzamidine	223 res	17.3 s
Albumin the long one	585 res	97 s

Measured on this hardware, warm. Three fold a **bound ligand** *alongside the protein — including DHFR with methotrexate, a chemotherapy drug, shown sitting in the pocket it blocks. Albumin is the long one: at 585 residues it is what a whole, real, drug-carrying protein costs.*

> **Four chips, four proteins — not one protein four times faster.**
>
> A single fold runs on a single chip; that is tt-bio's documented limit, not something this demo works around. Four chips buy four folds at once, and a shorter wait for whatever you pick.

The science is not ours. This booth exists to show other people's work running on Tenstorrent hardware. Folding is **tt-bio** (MIT) by Moritz Thüning; the demo around it is Apache-2.0. Structures are the model's own output, written as mmCIF and drawn here — no reference structure is ever shown in its place.

`github.com/tsingletaryTT/tt-bio-demo`

Operating the booth

For the person **running** it
tt-bio-demo · v0.4.0

STARTING IT

```
# the normal way — double-click
# "tt-bio Protein Folding Demo"

# or, from a terminal:
/opt/tt-bio-demo/scripts/run-demo.sh
/opt/tt-bio-demo/scripts/run-demo.sh --quad
```

`--quad` starts in the 2×2 grid; **Q** toggles it either way at runtime. The launcher starts **both** halves — the compute daemon and the UI — and tears the daemon down when it exits.

STOPPING IT

Ctrl + **Q**, or **Ctrl**+**C** in its terminal. One INFO line about a `tt-smi` sample killed by a signal is **normal** on a clean stop.

Signalling by pid does nothing — the launcher's trap cannot run until the UI returns. Signal the whole process group:

```
kill -INT -$(ps -o pgid= -p <pid> | tr -d ' ')
```

KEYS

? or **F1**	the help card, from any screen
Q	quad view — all four chips at once
T	Tensix activity — live core grids
D	diagnostics — the live protocol log
Esc	close the card, or either rail panel
Ctrl+**G**	easter egg: the Tenstorrent mark, computed on a chip
Ctrl+**F**	leave / return to fullscreen
Ctrl+**Q**	quit
any other key or a tap anywhere	wake the booth and show what it folds

RUNNING IT UNATTENDED

For a booth that must survive all day, let systemd own the compute daemon so it comes back by itself:

```
systemctl --user enable --now tt-bio-demo
systemctl --user status  tt-bio-demo
systemctl --user restart tt-bio-demo
```

It restarts **on failure only** — if you stop it deliberately it stays stopped. The UI is still launched separately, from the desktop entry.

THE RIGHT-HAND RAIL

Pipeline
One row per stage — msa, prep, trunk, diffusion, confidence, saving. The bright row is running now; diffusion owns most of the bar. Clears itself after 20 s of silence, so a dead daemon cannot leave a bar frozen all day.

Chips
Temperature, power and clock per chip, from a `tt-smi` snapshot every two seconds — sampled on the UI's own thread, never through the daemon, so a wedged daemon still leaves the hardware readout live.

Tensix activity – `T`
One animated core grid per chip. Only the chip actually running the fold animates the work, and the header names it.

Diagnostics – `D`
Every protocol event as it arrives, with a plain-language line for each stage. Closes itself after five minutes of no input.

Help card – `?`
Works from any screen without pausing the fold. Closes after a minute.

WHEN A VISITOR TAPS A PROTEIN

It gets folded **next** — never "now". The tap goes to the head of the queue and lands on the next chip to come free. Nothing already running is cancelled, because tearing a fold down mid-device-operation is a known instability source.

Measured: **0.25–5.5 s** with all four chips busy (median ≈ 2 s). And most taps queue nothing at all — with six targets on four chips the protein is usually already folding, so that cell simply takes focus. The booth acknowledges every tap immediately on a one-line notice.

> **Never run `tt-smi -r` on a shared machine.** A chip missing from telemetry may have been deliberately quarantined by the daemon for temperature. Check `tt-smi -s` first.

> **Do the first fold before doors open — about 90 s per target.** Every time on the front of this sheet is measured **warm**. On a machine that has never folded that target, the kernels compile first: measured **94.5 s** against 9.4 s warm. Fold each one you plan to show, once, while nobody is watching.

QUICK FIXES

A cell says **TRUNK** and looks frozen	Working as designed — trunk makes no coordinates. It is holding the last real structure and says so.
A chip missing from the chips panel	Check `tt-smi -s`. It may be quarantined for temperature on purpose.
Panels frozen, protein still moving	Telemetry is sampled separately from the daemon. Restart the daemon; the UI can stay up.
Everything stopped	**Ctrl**+**Q**, then relaunch from the desktop entry.

BEFORE DOORS OPEN

- [ ] **Four chips present.** `tt-smi -s` lists four entries across two `board_id`s. Three means one did not come up.
- [ ] **Fold each target once.** This is the one that gets skipped. It warms the kernel cache so no visitor watches a cold compile.
- [ ] **Help card opens.** Press **?**. It is what a visitor reads when you are talking to someone else.
- [ ] **Weights on disk.** The venue is offline — nothing downloads at the booth. `dpkg -l tt-bio-demo-weights`.
- [ ] **Fullscreen and quad.** **Ctrl**+**F**, then **Q** — or start with `--quad`.
- [ ] **Leave it alone for a minute.** The attract loop should start folding on its own, unprompted.

If something looks wrong: press **D** for the live protocol log — it names the stage and the chip. The UI never displays a stack trace or raw error text by design, so the log is where the detail lives. Install and recovery: **INSTALL.md**. Known gotchas: **docs/followups.md**.

4 chips · 2× p300c · Ubuntu 24.04