

MANUAL

Every knob in AlphaNode, explained in plain language.

What each panel shows, what every setting actually changes, and what each leaderboard column measures — with the shipped default, the range the app allows, and the trap worth knowing before you turn it up. One entry per knob, numbered, so you can jump straight to the one in front of you.

CONTENTS

01 The basics — what you mine, and what the main window shows

1.01	A formula (an "alpha")	4
1.02	A round	4
1.03	TRAIN / VAL / TEST	4
1.04	Fitness — the number the search maximises	5
1.05	► Start node	5
1.06	■ Stop	6
1.07	● stopped / starting / running (the state pill)	6
1.08	⚙ Settings	6
1.09	📅 Sessions	7
1.10	Light / Dark	7
1.11	Activate node / Unlock	8
1.12	ROUNDS · FORMULAS TRIED · ALPHAS FOUND	8
1.13	BEST FITNESS	8
1.14	The live log	9
1.15	SIGNAL API — running services	9
1.16	LEADERBOARD	9
1.17	PORTFOLIO — top-N combined	9
1.18	FORWARD TRACK — paper steps	10

02 Pairs, timeframe and market data

2.01	Which pairs to trade	12
2.02	Timeframe (bar size)	12
2.03	Market data (the automatic download at Start)	13
2.04	Data and state on disk	13

03 The search settings — how hard and how long it looks

3.01	Resources (CPU share)	14
3.02	Population	14
3.03	Generations	14
3.04	Seed (0 = auto)	15
3.05	Pause, sec	15
3.06	Status port	15
3.07	Optimize by win rate	15
3.08	Explore every N-th	16
3.09	Max. rounds (0 = ∞)	16
3.10	Leaderboard size	16
3.11	Warm-start from library	17

04 The engine knobs — simulation, formula shape, breeding, fitness

4.01	Target-vol (ann.)	18
4.02	Fee (turnover)	18
4.03	Max. depth	18
4.04	Max. nodes	19
4.05	Tournament size	19
4.06	Elitism	19
4.07	Random/generation	20
4.08	Crossover share	20
4.09	Complexity penalty	20
4.10	Correlation threshold	20
4.11	Similarity penalty	21
4.12	Hall of Fame size	21

4.13	Robust blocks (0 = legacy)	21
4.14	One-sided book guard	22

05 Date segments, and the honesty discipline

5.01	★ TRAIN start	23
5.02	★ VAL start	23
5.03	★ TEST start	24
5.04	★ TEST end	24
5.05	The window at each bar size	24
5.06	Working with a re-cut	25
5.07	by — how portfolio members are picked	25
5.08	★ The session contract	25

06 The leaderboard, column by column

6.01	★	27
6.02	#	27
6.03	fitness	27
6.04	TEST OOS	27
6.05	maxDD	28
6.06	CAGR	28
6.07	sortino	28
6.08	T ↑	28
6.09	T ↓	29
6.10	T ~	29
6.11	L/S /yr·a	29
6.12	tr/yr·a	29
6.13	win ↑	29
6.14	win ↓	30
6.15	ID	30
6.16	formula	30
6.17	L/S %	30

07 What you can do with a formula

7.01	Double-click a row → the equity window	32
7.02	Passport	32
7.03	PDF report	32
7.04	Serve signal (API)	32
7.05	Download signals (CSV)	33
7.06	Forward track ➕	33
7.07	Chart (Forward Track panel)	33
7.08	★ Favorites	33
7.09	Copy formula / Copy formula + metrics	34
7.10	Right-click a header → the Columns menu	34
7.11	Sorting	34
7.12	Export table (CSV)...	34
7.13	Export full library (CSV)...	35
7.14	► Build portfolio	35
7.15	"families only" switch	35
7.16	Clear all history / Clear node	35

08 The simple screen — the node on autopilot

8.01	The header	37
8.02	► Start node (the run pill)	37
8.03	Search card	37
8.04	Champions card	37
8.05	Portfolio card	38
8.06	The sealed preview (no activation key)	38
8.07	Serving — the autopilot	38

Clear node

39

8.08

Part 01

The basics — what you mine, and what the main window shows

What an alpha is, what one round does, how history is split, and what every card on the main window is telling you. Nothing here is a knob you have to tune.

1.01 A formula (an "alpha")

A short maths expression over the price tables — a real one looks like

`ema:20(ts_delta:14(cs_zscore(cs_demean(add(ts_max:30(volume),ret))))`). The number after a colon is a lookback in bars. It is evaluated once per coin per bar: positive means be long this coin, negative short, bigger magnitude a bigger share of the book. Leaves are the only 12 things it can see (close, open, high, low, volume, ret, vwap, range, body, dvol, logret, funding); branches are 25 operators — arithmetic, shape-benders (`neg/sign/abs/slog/tanh`), lookbacks over 2–200 bars of one coin's own history (`ts_mean`, `ts_std`, `ts_zscore`, `ts_min`, `ts_max`, `ts_delta`, `ts_delay`, `ts_sum`, `ts_roc`, `ema`) and cross-sectional ones that compare all your coins on the same bar (`cs_rank`, `cs_zscore`, `cs_demean`, `cs_scale`). Everything downstream is fixed and identical for every alpha: divide by the coin's recent volatility, normalise to 100% of the book, scale to your target vol, hold through a 10% no-trade band, pay fees on turnover. So the search can only change the SIGN and the RELATIVE SIZE of the numbers across coins and across time.

Watch out. Absolute scale is a no-op — `add(x,x)` is exactly the strategy `x`, two nodes more expensive. And cross-sectional operators need a real cross-section: with a single pair `cs_rank` is a flat 0.5 and `cs_demean` is exactly zero; with two, `cs_demean` is one long/short spread and nothing more, while `cs_rank` only ever emits 0 and 0.5, so on its own it never shorts — these operators start earning their keep at five or six coins.

1.02 A round

default **population 200 · 25 generations · Hall of Fame 15 · pause 5 s · explore every 4th round**

range **population 4–4,000 · generations 1–500 · HoF 1–100 · pause 0–3,600 s**

One complete evolutionary search, start to finish: build 200 candidate formulas, score them all, breed the survivors for 25 generations, fine-tune the lookback windows of the top 5 finishers, then hand back up to 15 champions kept deliberately un-alike. New ones are appended to your library; the node pauses and starts again with a new seed. A round costs at most population × generations evaluations and usually fewer — repeats inside a round are scored once. Rounds are the unit of progress: let it run for tens of them before judging anything.

Watch out. Warm-start only ever sees champions from the CURRENT session, and only the top few ("Leaderboard size", 20) of them. Before you activate the node everything on disk is sealed, so the refine pool starts empty and refine may never run at all.

1.03 TRAIN / VAL / TEST

default a window measured back from today – TRAIN 50% / VAL 20% / TEST 30% of the bar size's budget

History cut into three consecutive ranges. TRAIN is where evolution works. VAL is a second era that breeding never optimises directly — it catches formulas that merely memorised TRAIN. TEST is held out entirely: not the fitness, not the selection, not the seeding ever reads it; its numbers are carried along purely so you can look afterwards. Treat TEST as a report card you read once you have decided what to keep — the moment you pick alphas BECAUSE their TEST is high, it stops being out-of-sample, which is what every $\triangle$ on a TEST-flavoured action means.

Since 3.1 the four dates are not fixed: by default they follow today. Each bar size takes the same shape — a window of 80% of its bar ceiling, cut TRAIN 50% / VAL 20% / TEST 30% and ending on the current day — so on 1d TEST is the last two years, VAL the 16 months before that, TRAIN the 3.3 years before that. The date boxes show the resolved dates, and the note under them says how many of the four still follow today. Type a date into a box and that one is pinned; the others keep sliding.

Watch out. A row in the library was scored on the window that stood on the day it was mined, and its numbers are frozen; the chart, the portfolio and the computed columns use today's window. After a long run the two differ by however many days have passed — usually harmless, but a row's stored TEST Sharpe and a freshly drawn TEST curve are not the same measurement.

1.04 Fitness — the number the search maximises

default min(TRAIN, VAL) Sharpe · complexity 0.010/node · correlation 0.70 · similarity 0.5 · concentration 0.3 under 3 positions · activity floor 10%

The WORSE of two Sharpe ratios: min(TRAIN Sharpe, VAL Sharpe) — a formula is only as good as its weakest era, so curve-fitting one bull market buys nothing. While breeding, the search also subtracts formula size, correlation above 0.70 with a champion already found, and concentration when the book averages fewer than 3 meaningful positions. A formula in the market on under 10% of TRAIN or of VAL bars is discarded unscored. Fitness is the only score not contaminated by TEST — but a high one is no promise: rows scoring above +2 in the shipped library go on to a negative held-out TEST.

Watch out. The fitness you SEE is not the one the search ranks by: the stored number carries only the concentration penalty, while size and diversity steer breeding and are then thrown away. That is why the log's green ★ line reads lower than the round summary for the very same formula.

1.05 ► Start node

Saves your settings, then checks your market-data snapshot: if it is missing, missing one of your pairs, or its newest bar is over 5 days old, a console window opens, downloads exactly your pairs at your timeframe from Binance, and Start resumes by itself. Then it launches the node as a separate process, hands it every setting as an environment variable and polls its status every 1.5 s. A ticker that is not a real Binance USDT-perp is dropped with a warning rather than looping. The node also publishes a live status page on <http://localhost:8787> — bound on all interfaces, so anyone on your LAN who knows the port can read it.

Two checks run before the node is launched. The date segments must be in order, at least 30 bars each and inside the bar size's ceiling. And if this timeframe's library is not empty, the ★ settings must match the ones it was mined under — the session contract (5.08). A mismatch opens a dialog listing every changed field as was → now, with two ways out: revert to the recorded settings, or clear the node and start a fresh library on the new ones. Close the dialog instead and you can keep editing, but the node will not start until one of the two is chosen.

While the node runs, the settings panel is read-only: every field is greyed, the pair chips lose their × and the timeframe box will not open. Stop first, then edit.

Watch out. Only the ★ fields are frozen by the contract. Population, generations, CPU share, seed, pause, the genome and breeding knobs, the penalties and the leaderboard size may all change between runs of the same library — they change how hard the next round looks, not what a score means.

1.06 ■ Stop

Sends the node an interrupt. It notices at the next generation boundary, writes its final status and exits — seconds to a couple of minutes. Stop before anything that rewrites your library: activating, loading a session and clearing history all refuse to run while a miner is appending to the same files. Nothing already written is lost.

Watch out. The tooltip says the current round will finish — it does not. The round in flight is abandoned and its whole Hall of Fame discarded before reaching the library. If the log shows a ★ new best, wait for the log's "✓ round N" line before stopping.

1.07 ● stopped / starting / running (the state pill)

What the node itself reports: grey stopped, green running. There is a "starting" state in the code, but the node writes its status file for the first time only once the search is already under way, so that state never reaches the pill: while the node boots — checking or downloading data, loading your library — the pill still shows whatever the previous run left behind, usually grey stopped. Beside it, a one-line resource summary: CPU budget, cores that works out to, universe and effective target volatility. When nothing runs, those lines are prefixed "last run —" so you cannot mistake stale text for telemetry. In practice you never catch the blue: the node's first write to its status file already says running, so the whole boot window goes by with the pill unchanged — and on the first run of a timeframe that silent stretch can legitimately last a while. What tells you the node is really up meanwhile is the button pair — Start greys out and Stop goes live the moment the process exists.

Watch out. The pill mirrors the node's status FILE, not the process. After a hard kill it keeps saying green "running" with nothing behind it — the giveaway is that the Start button is clickable.

1.08 ⚙ Settings default hidden on a fresh install

Shows and hides the left-hand panel holding every knob the engine understands. Hidden on a fresh install, remembered between launches, always exactly as wide as its content (deliberately not draggable). Every field carries a hover tooltip. At the bottom sit "Reset to defaults" and the red "Clear all history". Leave it closed for normal use — the defaults are what the engine was tuned with.

Fields marked ★ — the pairs, the four dates, target vol, fee, the win-rate objective and robust blocks — are locked to the session: the legend at the top of the panel says so, and 5.08 explains why. The whole panel is disabled while the node is mining.

Watch out. Reset keeps two things on purpose: your theme and whether this panel is open. And "Clear all history" deletes the libraries and round history of EVERY timeframe at once, plus the built portfolio, the session contract and the cached return series — settings, favourites and the forward track survive (enrolled strategies keep stepping), your mined alphas do not, and nothing is backed up first. The clear also starts a NEW session id, so forward entries enrolled before and after it are distinguishable.

1.09 📁 Sessions

Your whole workspace as one .tar.gz — everything on the board. Every timeframe's library (so the leaderboard comes back as it was), the round history, the forward track, the built portfolio with its members, your ★ favourites, the run status — the four counters at the top, the live log and the round ticker — and your settings. Two things are deliberately left out: market data, which re-downloads, and your subscription key, so a session travels safely to another machine or another person. The registry of running signal services stays behind too — it is a list of process ids on this machine and means nothing anywhere else. An archive's 6-character ID is the SESSION id — the one in the window header when it was saved, the same shape the leaderboard uses for an alpha. Two saves of one session share it (two photographs of the same work, told apart by date); different sessions never do. It is in the manifest and in the filename, so an archive sitting in a backup folder or arriving by mail says which session it is without being opened — and two archives with the same name, which is the normal case, stay distinguishable. Loading replaces the current workspace — with one deliberate exception: the forward track is MERGED, never replaced. Its steps happen in real time and cannot be recomputed, so a load must not kill what is stepping; the archive's entries join the live list, and where the same id sits on both sides the live entry stays — the archive's copy takes over only when it is the same enrollment (same formulas, pairs, timeframe, target vol, fees, start capital and enrollment date) and has been carried further along. There is deliberately no second, per-save id — an archive simply carries the workspace's own SESSION id (shown in the header): minted at first run, reset by "Clear all history", adopted from the archive on load. Forward entries stamp it at enrollment. Loading refuses to run while the node, a forward step or a portfolio build is working. Double-click a row to see what is inside before you load it.

Watch out. A session is saved mid-run, so its status says the node was running; loading rewrites that one field to "stopped" and dims the line to "last run —", because nothing is actually searching after a load. The counters and log it carries are history and are kept verbatim. The tooltip promises an auto snapshot every time the node stops. There is none, and loading a session takes no checkpoint either — the swap replaces the library, the round history, the portfolio, your ★ favourites and the counters, and the copies it parks in order to undo a failed swap are deleted the moment it succeeds. If you want a way back, save the current session yourself first.

1.10 Light / Dark default follows the operating system on the very first launch

Repaints the palette and rebuilds the window from scratch — that rebuild is what lets the parts which cannot restyle themselves live (the table, the equity images, the chart canvas) come back in the right colours. Your leaderboard, chart and running services are restored immediately after. The flicker is by design. Harmless to flip at any time, including mid-search.

Watch out. "Follow the OS" happens once, at startup: the resolved value is written into your settings and saved by the next Start. There is no system position to go back to, and Reset preserves the theme — the only way back is editing gui_settings.json by hand.

1.11 Activate node / Unlock

Every formula the node mines is sealed before it touches your disk: the library stores encrypted text plus a public 12-character id, and the row shows its stored metrics but not the maths. Activate takes your subscription key once, claims a machine seat, and unlocks every sealed formula in every local library in one pass; from then on the node mines in the open, re-checked at each Start. The button lives in the leaderboard heading on the dashboard and in the header of the simple screen — enter the key on either, and both screens are activated; once a key is accepted the simple header shows a green "● activated" pill beside the brand instead. The key belongs to the MACHINE, not to one install: it is kept in a shared store (~/.config/AlphaNode/licence.json on Linux, the AlphaNode application-support folder on Windows and macOS), every other install on the box — a .deb next to a dev checkout, an AppImage — adopts it at its next launch, claims its own seat and reveals whatever it holds sealed, without a click. An install that already holds a key seeds the store for the rest.

An un-activated node is not blind. A sealed row reads "locked · Sharpe +1.04 · win 54% · PnL +29%" — its held-out TEST Sharpe, win rate and total return, stored at mining time — wherever a formula would be shown. The Portfolio card still builds (a sealed preview, 8.06), the search chart and the tiles work as usual. What needs the plain text stays shut: the computed leaderboard columns, Serve, the signal CSV, the PDF report and the forward track.

Watch out. Activation rewrites the library files, so it refuses while the node is mining — stop, activate, then start. The per-row "Unlock" on the dashboard is session-only: the file on disk stays sealed and the row is locked again after a restart. And a key that has lapsed by the next Start makes the node seal again, silently — the rows it mines then unlock with the next successful activation.

1.12 ROUNDS · FORMULAS TRIED · ALPHAS FOUND

The three tiles, all read from the node's status file rather than your library. ROUNDS is how many complete searches have finished, resumed from the round history across restarts. FORMULAS TRIED is the lifetime total of distinct formulas evaluated — the honest measure of compute spent. ALPHAS FOUND is how many distinct formulas are in the library being mined into. Watch ALPHAS FOUND against ROUNDS to judge productivity: once a library is mature, whole rounds keeping nothing is normal, not a fault.

Watch out. Switching timeframe moves nothing at once: your pick only reaches your settings when they are next saved, and pressing Start saves them. Until then the table below is still reading the previous timeframe's library, and the tiles still show the last run's numbers. FORMULAS TRIED is genuinely cross-timeframe: it comes from one shared status file.

1.13 BEST FITNESS

The fitness of the current number-one champion — the same min(TRAIN, VAL) figure, drawn in the accent colour because it is the number that says whether the search is getting anywhere. It only moves at round boundaries, so a mid-round silence is not a stall. Expect a jump in the first handful of rounds and then a creep; a long flat stretch means the easy structure is exhausted at these settings — widen the universe, raise the complexity limits, or move to the Portfolio card. Shows "—" until the first round finishes, and never shows a TEST number.

Watch out. The objective is stamped into the status file only when a round finishes, so during the first round of a run the tile has no tag to read: a win-rate figure carried over from earlier rounds is drawn Sharpe-style as "+0.56" instead of "56%". From the end of round one it reads as a percentage. The leaderboard's fitness column tags each row from the row itself, so trust the column while that first round runs.

1.14 The live log

Seven colour-coded lines from the node's own event feed (the last 80 events are kept): round starts and finishes in normal ink, ★ new bests in green, forward-track steps in blue, warnings and errors in red. The champion window fine-tuning never reaches this feed — those lines land on the round ticker above the log instead. "► round N: explore" versus "refine" tells you the mode; "✓ round N · 47s · 4,812 formulas tried · +3 champions kept · best fitness +1.81 · held-out TEST −0.26" gives you the cost and the yield of a round in one line. This is the best single place to see whether the node is healthy.

Watch out. While the library is sealed, formula text is stripped out of the log on purpose — the ★ line keeps its numbers but the maths after the em-dash is cut. Nothing is broken; the vault is doing its job.

1.15 SIGNAL API — running services

A card that appears once you are serving something. "Serve" starts a small local web service that recomputes that strategy's live target positions and hands them out as JSON, on the next free port from 8799; each gets a row with its URL, log file, health and a way to shut it down — the status line turns green once it says "serving", so a live API reads at a glance. This is the hand-off to your own execution code — poll the URL, get target weights per asset, trade them however you like. The services are detached: they keep serving after you close AlphaNode, and the app re-adopts them next launch by probing ports 8799-8818 — twice the ten-service cap, so a service pushed further up the range by another program is still found.

Watch out. Because a service outlives the GUI, a stale one from a previous session can still be answering on 8799 with an old strategy. Check the row's label and started-time before blaming the formula.

1.16 LEADERBOARD default every alpha at this timeframe, sorted by fitness, families-only off

The library itself as a table — every formula ever mined at this timeframe, not the node's top 20 — re-read in a background thread whenever the file changes. Click a header to sort, double-click a row for its equity chart, right-click for copy, CSV export and a column picker. The search box filters live by id or formula text. "Families only" collapses the table to one representative per formula shape, the fastest way to see how many genuinely distinct ideas you own. Rows are tinted green only when their held-out TEST Sharpe is at least zero — in a healthy library that is a minority, and that is the honest picture.

Watch out. Clicking "fitness" or "TEST OOS" does more than re-sort — it re-selects WHICH alphas are loaded from the library, and the heading changes to say so. On a sealed library only the numbers stored at mining time survive: the formula cell carries the row's Sharpe, win rate and PnL, while sortino, T↑/T↓/T~, L/S, trades and win↑/win↓ show "—" for every row.

1.17 PORTFOLIO — top-N combined

default top 6, by TEST · 2–20 (a widget default: not stored in settings, untouched by Reset)

Runs N alphas from your library together and plots the combined equity with its metrics above it: Sharpe on TRAIN, VAL and TEST separately, TEST CAGR and max drawdown, against buy-and-hold. This is where mining hundreds of alphas pays off — the combined curve is usually far better than any single member, because their mistakes are uncorrelated. The "by" selector picks the members: TEST Sharpe, fitness, or "combo" (a greedy search for the best COMBINATION, optimised on TRAIN+VAL only). Daily uses the full simulator, one to two minutes; intraday uses the fast approximation and finishes in seconds — the line under the chart always names which engine produced the numbers.

Under the metrics sits the member list: one row per alpha that went in, in pick order, with its ID, its SOLO TEST Sharpe (that member on its own), and the fitness and TEST OOS the leaderboard shows for the same row. Members are equal-weight — none is sized larger than another. Rows the combined Sharpe beat are tinted; that tint IS the diversification gain, and a row without it is a member the mix would be better off without. Double-click one for its own equity chart. Past twelve members the list scrolls instead of growing the card.

On a sealed library the card builds a preview instead of a real book (8.06): the members are picked by the same combination search over the return series the node caches for every champion, mixed equal-weight, and the engine line says so. The member list then shows ids and stored metrics, and no button unlocks.

Watch out. The 2–20 range is enforced on the built portfolio, not on the box: the spinner accepts a typed 150, and the status line then tells you it is building 20. The range is not arbitrary — "combo" caps its search pool at 30, so asking for a combination larger than that would stop searching and hand back the whole pool. Note also that you get the number you asked for only if the library holds that many DISTINCT alphas: near-clones are filtered at 85% similarity, so a 45-row library yields about 39 members however high you set the number. "by TEST" is the dishonest pick and the card says so: the window that chose the members is the window being scored. Prefer fitness or combo.

1.18 FORWARD TRACK — paper steps

default starts at \$10,000 · one step per closed bar · checked every 5 minutes

The honest verdict machine. Enrol a portfolio or a single formula and it is frozen — formulas, universe, target volatility, fee — then paper-traded forward from \$10,000, one step per closed bar: fetch the latest closed candles, recompute targets with the real engine, mark to market, rebalance, charge fees. History is append-only; nothing is recomputed backwards. This is the only number in the app that no amount of peeking can contaminate, because the bars did not exist when the formula was mined. Enrol your best candidates early and give them months. The node steps the track every 5 minutes while it mines, and the app re-checks just as often on its own — but both stop when you close the window, because closing AlphaNode terminates the node with it. Leave the app running for the strategies you care about. A missed bar is not lost money — the next step prices the held book against the latest close and accounts for the funding across the whole gap — but it is a rebalance that never happened and a row that never appears, so long gaps leave the track sparse and its Sharpe less trustworthy. The track OUTLIVES sessions: "Clear all history" spares it, and loading a saved session merges that session's entries into the running list instead of replacing it — nothing that is stepping ever stops because you switched workspaces. Each entry records the session it was enrolled from (the SESSION column), so a list accumulated across many sessions still says where every strategy came from.

Watch out. "Delete" is permanent and takes the paper history with it — re-enrolling the same formula starts a fresh \$10,000 track rather than resuming, so a year of forward evidence goes in one click.

Part 02

Pairs, timeframe and market data

The only settings that change the DATA rather than how it is searched. Get them right before touching anything else. AlphaNode downloads its own history from Binance USD-M perps — there is no file to point it at and no Download button.

2.01 Which pairs to trade default BTCUSDT, ETHUSDT, SOLUSDT, XRPUSDT, BNBUSDT range 1–20 pairs

The exact basket the whole app runs on — search, leaderboard statistics, equity charts, CSV, live signals and the forward track use this list and nothing else. Full Binance USDT-perpetual tickers only (BTCUSDT, not BTC): what you type is uppercased and de-duplicated, but nothing appends the USDT for you. Write up to 20 pairs and press Enter once — commas, spaces, tabs and line breaks all separate them, and the whole box turns into chips. Nothing commits while you type: a comma is just a character, so the text stays exactly as you wrote it until Enter (clicking away or pressing Start commits it too, so a pair left in the box is never lost). A paste commits itself, no Enter needed; × removes a chip; clicking a chip's text pulls it back into the box to fix a typo (it returns at the END of the list); Backspace in an empty box pulls the last chip down. Below about 5–6 pairs the cross-sectional operators the search leans on stop meaning anything — that is the practical floor. Going wider buys a broader cross-section and costs download, memory and simulation time in direct proportion, up to the ceiling of 20 — a longer list keeps its first 20 pairs and says in red how many it dropped. The cap is enforced in the engine, not just the panel: editing config.ini or ALPHANODE_UNIVERSE by hand does not get you past it. A recently listed coin is safe to add: its pre-listing bars stay blank and are masked out.

Watch out. A one-pair universe looks like it works and does not — the weight is then always exactly ± 1 , so only the SIGN is traded and every alpha collapses to the same flip. And library rows do not record which basket they were mined on: change the pairs and one row mixes two worlds — fitness, TEST OOS, maxDD and CAGR are the OLD basket's stored numbers while win%, L/S, sortino, the T columns and the chart are re-simulated on the new one.

2.02 Timeframe (bar size) default 1d range 1d | 4h | 1h | 15m

The bar size for the entire pipeline. Every window inside a formula is counted in BARS of this size, so a 20-bar average is 20 days on 1d and 5 hours on 15m; Sharpe is annualised over 365 days, because crypto trades 24/7. Each timeframe is a separate world — its own snapshot file, its own alpha library, its own round history, its own session contract — and alphas from different bar sizes never mix or compete. Start on 1d: longest history, lightest download, fastest generations, and the only snapshot that ships with the app. Drop to 4h or 1h for more bars per year of calendar; intraday history is shorter, the download far heavier, and fees eat a much larger share of each bar's move. The choice takes effect only when settings are written — press Start, or collapse the settings pane. The line under the dropdown states that bar size's ceiling on the whole TRAIN-to-TEST-end window, which is what stops a 15m search from being pointed at six years of history.

Watch out. Picking from the dropdown OVERWRITES all four date fields with that bar size's window measured back from today — including when you re-pick the timeframe you are already on. Hand-pinned dates are thrown away silently. Always pick the bar size first, then pin what you want pinned. The box is locked while the node runs.

2.03 Market data (the automatic download at Start)

Start inspects the snapshot for the active timeframe and downloads first if it is missing, unreadable, missing a pair from your list, or its newest bar is over 5 days old. A console window streams the downloader's live output and Start resumes by itself when it exits cleanly. If Binance's live API is unreachable from where you are, it falls back on its own to the public data archive — the same bars, 10–30 hours behind. The wait scales with bar size: 6 pairs in parallel on 1d, 4 on 4h, 3 on 1h, 2 on 15m — deliberately slow, because more parallelism trips the rate limit. Closing the console cancels safely: the file is written under a temporary name and swapped in only on complete success. History always begins at the timeframe's recommended date or the pair's listing date, whichever is later, so setting TRAIN start earlier buys no extra data. Funding-rate history comes with the candles, which is what makes the funding term mean anything.

Watch out. In the app this is a replacement, not a top-up: it asks for exactly your current pairs and the whole snapshot is overwritten with just those. Adding one pair re-downloads all of them from scratch, and a pair you removed loses its stored history at the next refresh. A headless node does the opposite — it fetches the pairs already in the snapshot plus whatever its universe adds, and deliberately never shrinks the file, so pointing a container at two pairs leaves a forty-pair snapshot whole.

2.04 Data and state on disk

In the installed app everything writable lives in one folder — `~/.local/share/AlphaNode` on Linux, `%APPDATA%\AlphaNode` on Windows, `~/Library/Application Support/AlphaNode` on macOS. It holds the price snapshots, `gui_settings.json`, `exports/`, and `state/` with the mined alphas (`library.jsonl`, `library_4h.jsonl` ...), the round history, `status.json`, the portfolio, the forward track, favourites, sessions, this install's node id, `session_params.json` (the session contract, per timeframe) and `series/` (the return series the node caches for its sealed champions — the top 100 by fitness, a few tens of MB at most). Your alphas are just those `library*.jsonl` text files — back them up, or use Sessions → Save current.... Deleting a `data_*.pickle` is harmless; the next Start downloads it again. Settings are written only when something triggers a save: Start, a theme switch, collapsing the settings pane, and most layout fiddling all rewrite the whole file.

Watch out. Closing the window does NOT save — a pairs or timeframe edit followed straight by quitting is simply lost. And "Clear all history" counts only the active timeframe in its dialog, then deletes every timeframe's library and history, plus `status.json`, the portfolio, the contract, the series cache and the cached equity images. Nothing is backed up automatically.

Part 03

The search settings — how hard and how long it looks

How hard the app hunts, for how long, and how it splits its time between finding new ideas and polishing old ones. All of it is read once, at Start. None of it can touch the honesty of the result — TEST is held out of selection whatever you set here.

3.01 Resources (CPU share) default 50% range 5–95 (slider only)

The share of your processor handed to the search. The bold line above the slider does the arithmetic live — the share times your core count, rounded to whole worker processes, never fewer than one: "50% → 4 of 8 cores". This is the single biggest lever on wall-clock speed. It costs memory (every worker builds its own copy of the price panel, so RAM grows with the worker count), heat, and a machine that feels sluggish. Lower it to about a third to keep working on the same computer. On Linux and macOS the node also asks for background priority; on Windows that call is skipped.

Watch out. The worker pool is created fresh every round and each worker rebuilds the price panel from scratch, so a big core share buys little when Generations is very small — you pay that setup cost per round instead of spreading it over a long one.

3.02 Population default 200 range 4–4,000

How many candidate formulas exist side by side in one generation. Generation 0 is random trees (plus your warm-start champions); every later generation is rebuilt from the previous by selection and breeding. Raise it for a wider sweep of formula space — more building blocks in play, less chance of getting stuck polishing one mediocre idea. Cost is close to linear in time. If you want more total search, raise Population before Generations: breadth is what protects you from settling on the first decent idea the run stumbles into.

Watch out. Each generation is assembled as elites, then random injections, then bred children until the count is reached — nothing trims it back. Any Population below Elitism + Random/generation (16 with the defaults) is silently overridden. A number typed into the box is also not clamped; only the arrows respect the range.

3.03 Generations default 25 range 1–500

How many cycles of scoring, selection and breeding happen inside ONE round. After the last generation comes a short extra pass over the top five: each lookback window is tried at 0.8× and 1.25×, up to three passes, keeping a change only if it improves the same TRAIN/VAL score. Raise it to let a round dig deeper into what it has found; the returns flatten, and you can watch them flatten — if the ★ new-best lines dry up around generation 15, there is little point paying for 40. Lower it for more, shorter rounds: more from-scratch restarts and more variety, at the cost of paying the per-round setup more often. Below about 10, evolution barely has time to combine anything.

Watch out. evolution/config.ini says generations = 30 and the app never uses it. The node always overrides population, generations, seed and cores with the panel's numbers (or 200 / 25 headless). The rest of the file is not dead, though: the node reloads it at the start of every round, and every search setting the panel has no box for comes straight from it — the concentration guard, the two block-fitness knobs, and window_polish, the only switch that turns the champion window polish off.

3.04 Seed (0 = auto) default 0 range 0–999,999

The starting number for the random generator that builds and mutates formulas. The typed seed is only a base: each round runs on that number plus its own round number, and the round counter is restored from this install's own history at Start. The same seed therefore reproduces a run only on a machine with the same number of rounds behind it — a fresh install begins at seed + 1, one with 120 rounds recorded begins at seed + 121. At 0 the app mints a seed once from a random node id in its state folder, so your install walks its own path through formula space and no two installs mine identical libraries. Leave it at 0 — that is what keeps your library yours. Set an explicit number only for repeatability: filing a bug, or A/B-testing one setting with the randomness held still. There is no quality difference between seeds; a "good seed" is survivorship bias.

Watch out. A shared number is a shared search — anyone running the same integer on the same settings and data mines the same formulas. Reproducibility also only holds for explore rounds: refine rounds start from whatever champions are in memory, so two machines with the same seed and different libraries diverge from the first refine onwards.

3.05 Pause, sec default 5 range 0–3,600

How long the node idles between rounds. It sleeps in half-second slices and checks for Stop before each, so Stop still responds during the pause. Raise it on a laptop or a thermally limited machine to let the fans catch up; it is pure lost mining time and has no effect on what is found. Set 0 to chain rounds back to back.

Watch out. This is not a load limiter. During a round the node still uses every core the CPU-share slider gave it, at full tilt — if the machine is too loud, lower Resources, not this.

3.06 Status port default 8787 range 1024–65535

The TCP port for the node's own small web page — a live table of the top formulas, the round log and the forward track at <http://localhost:PORT>, reloading every four seconds, with the raw numbers at [/status.json](#). The desktop window does not use this port at all (it reads the status file from disk), so the page is purely a bonus: a way to check on a mining machine from your phone or a headless server. Leave it alone unless something else owns 8787 or you want two nodes at once.

Watch out. The page listens on ALL network interfaces — anyone on your network who can reach the port can open it, and while sealed formulas show as locked, the timings, fitness and held-out TEST numbers are in the clear. If the port is taken the node keeps mining silently: no page, no dialog, nothing in the log. Keep clear of 8799 and the fifty ports above it — that is where live-signal services live.

3.07 Optimize by win rate default off

Switches what the search maximises. Unticked it maximises Sharpe — return per unit of wobble. Ticked it maximises the per-bar win rate: out of the bars where the strategy actually held a position, the share that closed in the black (flat bars count as neither). Either way it is the WORSE of TRAIN and VAL, so a formula must hold up on both. The score used for selection is deliberately lower than the raw percentage: the share is pulled toward 50% by the square root of how often the formula was active (62% of 40 bars is weaker evidence than 55% of 700), then one binomial standard error is subtracted with a small prior, so a perfect 5-of-5 reads as luck. Under 30 active bars in TRAIN or VAL, the formula leaves the round.

Complexity and similarity penalties are automatically scaled down tenfold here, because win rate lives near 0.5 while Sharpe spans a range several times wider. Use it when you want an equity curve that feels steady day to day, or as a second opinion when every Sharpe champion looks like the same trend-following family. Takes effect at the next Start.

Watch out. Win rate is blind to SIZE: 65% small green bars against 35% catastrophic red ones is a losing strategy this objective happily ranks first — judge anything it mines by TEST Sharpe and maxDD. In this build the switch also fails to reach the node's own ranking of the library, so its top list, the :8787 page and the fitness tile stay Sharpe-ordered, and refine rounds re-seed from Sharpe formulas. Mine win-rate alphas into a fresh library, or a timeframe whose library is still empty.

★ Part of the session contract (5.08): once this timeframe's library holds a row, the value is frozen until you clear the node — Start refuses a changed one.

3.08 Explore every N-th default 4 range 1–100

The mix between hunting and polishing. Every round whose number divides evenly by N explores — a completely fresh random population, looking for families you have never seen. Every other round refines: the population starts from the champions on the node's top list. With 4 the pattern is refine, refine, refine, explore... Each round announces which it is in the live log. Lower N means a broader, more varied library but a best-fitness that creeps; higher N climbs the headline faster but converges onto one or two ideas — and a library of eight variations on one formula is one strategy, not eight. If the top of your leaderboard has been the same shape for hours, lower it; if you have hundreds of scattered mediocre alphas, raise it.

Watch out. N = 1 quietly turns refinement off completely — every round number divides by 1, so everything explores and the warm-start switch stops doing anything. The only visible symptom is that every round in the log says "explore".

3.09 Max. rounds (0 = ∞) default 0 range 0–999,999

Stops the node automatically after this many rounds; 0 means it never stops on its own. Use a finite number for a bounded experiment or an overnight budget you do not want to babysit. It is a count of rounds, not of time — the same number can mean twenty minutes or twenty hours depending on population, generations, pairs and cores.

Watch out. The count is lifetime, not per session: the node restores its round number from your history, so with 120 rounds recorded and Max rounds 50 it starts, sees it is already past, and stops without mining anything. Set it above the round count on the dashboard.

3.10 Leaderboard size default 20 range 1–200

How many champions the running node keeps in its own top list — what the :8787 page shows, what the best-fitness tile is drawn from, and, most importantly, the exact pool a refine round warm-starts from. Read it as "how wide is my warm start". Small (5–10) makes refine hammer on a handful of formulas: fast convergence, narrow results. Large (50+) gives a broad, varied starting pool. It also decides how much of an existing library the node carries at all: on start it keeps only this many rows in memory and forgets the rest, so nothing below the cut can seed a refine round again. On a node you have not activated, nothing above the cut can either — those rows come back from disk sealed, with no formula text — and the warm-start pool holds only what the current session has mined. Keep it comfortably below Population — an order of magnitude smaller is sane.

Watch out. It does not limit your library and never deletes anything; the app's table reads the library file, not this list. And the table's own height has nothing to do with this number — you drag its bottom edge to resize it.

3.11 Warm-start from library default on

The master switch for refine rounds. On, a non-explore round begins with your top-list champions dropped into generation 0 unchanged and the rest of the population filled randomly around them. Off, every round is a from-scratch exploration and Explore every N-th stops meaning anything. Leave it on for normal use — this is what turns a series of independent searches into something that compounds. Turn it off deliberately when you want breadth rather than a peak.

Watch out. A champion with more operations than the current Max. nodes is silently skipped when the round seeds itself — no message anywhere — so lowering Max. nodes after mining large formulas quietly converts your warm start back into a random start.

Part 04

The engine knobs — simulation, formula shape, breeding, fitness

How the simulator trades a formula, how big a formula may get, how the next batch is bred, and how candidates are scored. Every value here is read at Start and overrides config.ini. The boxes are spinners: only the arrows respect the limits — a typed number is passed through unchecked.

4.01 Target-vol (ann.) default 0.25 range 0.01–3.00

The annual volatility the simulated book is steered to. The engine keeps a running estimate of how much the book is swinging and scales every position so that swing lands on this number — this is the leverage knob, solved for rather than typed. Because the score is a Sharpe ratio, it barely changes WHICH formulas win: measured on one signal at 0.10 / 0.25 / 0.60 / 1.50, realised volatility came out 0.10 / 0.26 / 0.63 / 1.58 (the targeting works) while Sharpe stayed +0.65 / +0.69 / +0.67 / +0.63. What it changes is the risk you would actually run: worst drawdowns in the same test were –20% / –44% / –76% / –99.7%. Set it to the risk you are willing to hold, not as a tuning dial.

Watch out. There is no margin model and no liquidation, so the high end is arithmetic rather than a tradeable setting. Across those same four targets the CAGR of the identical formula went 6.4% → 15.6% → 25.1% → –21%: at 1.50 the Sharpe was still fine and the compounded result was a loss, purely from the size of the swings.

★ Part of the session contract (5.08): once this timeframe's library holds a row, the value is frozen until you clear the node — Start refuses a changed one.

4.02 Fee (turnover) default 0.001 (10 bp) range 0.0000–0.0500

What trading costs, as a fraction of the notional that changes hands, charged each time the book moves — so a round trip costs about twice this. It is charged on the CHANGE in a position, not the position: a formula that sits still pays nothing. This is the only brake the search has on churn; there is no separate turnover penalty anywhere in the fitness. Set it at or above your real all-in cost, fee plus expected slippage, and the search will refuse to hand you signals that only work for free.

Watch out. Setting it to 0 does not merely flatter results, it changes what wins. Measured: a fast-turning signal scores +0.57 with fees off and –0.76 at the default, while a slow one only drops +1.13 → +0.69. A whole family of high-churn formulas becomes "profitable" at zero, and those are what the search then hands you. (A hard-coded 10% no-trade band already absorbs small rebalances, so your real cost is below turnover × fee.)

★ Part of the session contract (5.08): once this timeframe's library holds a row, the value is frozen until you clear the node — Start refuses a changed one.

4.03 Max. depth default 6 range 2–12

How many operators may be stacked on top of one another before a branch must end in a raw feature. This binds harder than it looks: of the 25 operators, 19 take a single argument and only 6 take two, so formulas grow as long narrow chains rather than wide bushes. In a real library of 270 champions only

about one node in ten was a two-argument operator and roughly a third of the champions sat exactly at the depth ceiling — depth is usually what caps a formula, not Max. nodes. Drop it to 3–4 for formulas you can read at a glance; raise it to 8 to allow longer chains of transformations.

Watch out. Depth is enforced by pruning, and pruning replaces an over-deep branch with a random raw feature — a child that breaks the limit is not rejected and retried, it is damaged and evaluated anyway. Warm-start champions are checked for size but never for depth, so lowering this after mining deep formulas means every child bred from them gets random leaves grafted in.

4.04 Max. nodes default 22 range 3–80

The total count of everything in the formula, operators and raw features alike; `ts_mean:30(close)` is 2 nodes. At the 22-node default a formula is typically 16–17 operations over 3–5 features. This is the size limiter and the number the complexity penalty is charged against. Small (10–15) gives simple, readable, harder-to-overfit formulas and quick convergence; large (40–80) gives expressive combinations and many more ways to memorise TRAIN. If you raise it, raise Complexity penalty too, or you are paying formulas to be complicated.

Watch out. The limit leaks exactly where it matters: crossover and mutation children over the limit are rejected and retried, but the fresh random injections are added with no size check at all. Measured across 20 seeds, at the default this is negligible — at Max. nodes 8, roughly 4 of the 10 injections per generation are oversize, and they can be selected as parents.

4.05 Tournament size default 5 range 2–50

How a parent is chosen: draw this many candidates at random and keep the best. The single dial for selection pressure. Low (2–3) makes mating nearly random — diverse, exploratory, slow to improve. High (15–50 against 200) lets the top few per cent dominate at once: fitness climbs for two or three generations, then the population becomes copies of one lineage. Watch the per-generation "evaluated N (uniq M)" line: if the unique count stops growing, pressure is too high for your population.

Watch out. Draws are made with replacement, so one formula can fill several slots of the same tournament. At 50 out of 200 the winner comes from the best five per cent of the population about nine times in ten, while the single best individual is actually drawn in roughly one tournament in five — pressure high enough to converge the population fast, not literally "always breed the best". The fitness compared here is the penalised one — a formula can lose for being large or for resembling a champion, rather than for trading worse.

4.06 Elitism default 6 range 0–50

How many of the best formulas are copied into the next generation completely untouched. Without it, the best thing you have found can be destroyed by the very mutation meant to improve it. 6 out of 200 (3%) is sane. At 0, progress can go backwards; above roughly 10% of the population the same handful keeps re-winning tournaments and the search stalls. Elites cost nothing to evaluate, but every elite slot is one not spent breeding something new.

Watch out. Elites are picked by the penalised fitness of that moment, similarity against the Hall of Fame included — so a formula can be an elite in one generation and drop out in the next without changing at all, simply because a champion resembling it entered the Hall of Fame in between.

4.07 Random/generation default 10 range 0–200

How many brand-new random formulas are dropped into each generation. Crossover recombines what the population already holds, while mutation reaches beyond it a piece at a time: one mutation in three grafts a freshly generated branch, whose operators and features are drawn from the full lists, and a point mutation swaps in another operator of the same kind. What injection adds is whole new formulas rather than variations on the ones already there, so it is the knob to reach for when the pool has narrowed onto one family. 10 out of 200 keeps a trickle of novelty. Raise it (20–40) when the unique-formula count flattens early. Every injection is a full evaluation and nearly all are junk, so it is a direct CPU tax.

Watch out. Elitism + Random/generation is not capped by Population: 50 and 200 against a population of 200 silently produces 250 individuals, costing 25% more CPU than you asked for. And when those two together already reach Population, no crossover or mutation happens at all — the run degenerates into pure random search.

4.08 Crossover share default 0.60 range 0.00–1.00

Of the children bred each generation, the fraction made by crossover — cutting a branch out of one parent and grafting it into a copy of another. The rest are mutations of a single parent, in one of three equally likely kinds: replace a branch with a fresh random one; swap an operator for another of the same kind keeping its horizon (`ts_mean:30` → `ema:30`); or nudge a lookback window, usually a small jitter and about one time in five a jump to the coarse starting grid. Crossover recombines whole working ideas; mutation does the fine work and is the only thing that tunes lookback windows.

Watch out. At 1.00 window tuning is off entirely — the nudge lives only inside mutation. The only fine tuning left is the polish pass after evolution ends, which touches just the top 5 champions.

4.09 Complexity penalty default 0.010 per node range 0.000–1.000

Subtracted in proportion to node count: $\text{score} - \text{penalty} \times \text{nodes}$. At the default the whole spread from the smallest formula to a full 22-node one is about 0.2 Sharpe — meaningful, not decisive. At 0.05 a full-size formula gives up 1.1 Sharpe and the search collapses toward stumps. Raise it when champions come out unreadable and their TEST lags their fitness (the classic overfit signature); lower it toward 0.005 when you have raised Max. nodes and want the room used.

Watch out. This steers selection, not the output: the Hall of Fame is sorted and deduplicated on the RAW score, so a bloated formula with the best raw score still tops your library however high you set this. Under "Optimize by win rate" it is multiplied by 0.1 internally, so the same typed number is ten times gentler.

4.10 Correlation threshold default 0.70 range 0.00–1.00

How alike two formulas' results must be before the app calls them the same alpha — the correlation of their per-bar TRAIN+VAL returns, ignoring sign. It does two jobs: it triggers the similarity penalty during selection, and it is the deduplication rule for the Hall of Fame. Lower it (0.5–0.6) when your library fills with near-identical variants of one idea; raise it (0.8–0.9) when too few distinct families come back and you would rather keep close cousins than nothing. Because sign is ignored, a formula and its exact inverse count as duplicates.

Watch out. Both ends are broken, in opposite ways. At 0.00 every candidate counts as a duplicate of its nearest champion — fed 500 unrelated candidates the list ended with exactly one entry (measured). At 1.00 dedup can never fire and the Hall of Fame degenerates into fifteen twins of the same formula.

4.11 Similarity penalty default 0.5 range 0.0–2.0

How hard a candidate is punished for resembling a champion already in the Hall of Fame: $\text{penalty} \times (\text{correlation} - \text{threshold})$, against the most-similar champion. It exists to stop the population converging on one idea. Raise it when every round returns variations of the same alpha; lower it to 0 when you are deliberately refining one family. Costs no CPU.

Watch out. At the defaults this is close to cosmetic: the largest subtraction possible is $0.5 \times (1 - 0.70) = 0.15$ Sharpe, noise next to a champion Sharpe of 1.5–2.5. To make diversity actually cost something, lower Correlation threshold as well. Setting it to 0 does not switch deduplication off — that is a separate mechanism on the threshold.

4.12 Hall of Fame size default 15 range 1–100

How many champions a round produces — the formulas written into your library when it finishes. Not a simple top-N: a new formula correlating above the threshold with an existing champion replaces it if it scores better and is discarded if it does not, so the list stays diverse rather than merely high-scoring. Raise it when rounds keep reporting "none kept" and you want the near-misses; lower it (5–8) when the library fills with entries you never open. Every formula is correlated against every champion twice per generation, so $15 \rightarrow 100$ makes that bookkeeping about seven times heavier.

Watch out. Raising it does not get you more of your best work: the polish pass that fine-tunes champion lookbacks is hard-coded to the top 5 whatever this says, and champions already in your library are skipped when the round writes out — so a larger Hall of Fame mostly adds weaker, newer entries.

4.13 Robust blocks (0 = legacy) default 0 range 0–12

An alternative way of scoring. At 0, fitness is $\min(\text{TRAIN}, \text{VAL})$ Sharpe. At 2 or more, the app merges TRAIN and VAL into one span, cuts it into that many equal time slices, shrinks each slice's Sharpe by its own margin of error, and scores the formula on the near-worst slice. The intent is that a formula must work across regimes. The honest answer: the app's own config.ini records a retrospective A/B on an 861-alpha daily library that found no improvement — rank correlation with held-out TEST was +0.40 legacy, +0.40 for the best 3-block variant, +0.26 for 5 blocks. $\min(\text{TRAIN}, \text{VAL})$ already IS a two-block worst-regime rule, split at the bull/bear boundary. Treat it as an opt-in experiment; if you try it, use 3.

Watch out. Turning it on changes what "fitness" means everywhere and nothing in the interface says so. Block fitness routinely runs deep in the negative (the same three formulas: $-0.28 / -0.32 / -0.89$ legacy versus $-0.86 / -1.92 / -1.26$ at 3 blocks), the leaderboard sorts on that raw number without knowing which rule produced it, so block-scored champions land under every legacy row and never seed a refine round. The box always opens at 0 — on a fresh install, and on an upgrade from a version that had no such setting — so nothing puts the app on block fitness behind your back; only a number you type and save can.

★ Part of the session contract (5.08): once this timeframe's library holds a row, the value is frozen until you clear the node — Start refuses a changed one.

4.14 One-sided book guard

config.ini only · default on, 0.5 past 75/25

A guard that docks the fitness of formulas whose book leans hard to one side. It measures net over gross across TRAIN and VAL — never TEST — which runs from -1 (always short) through 0 (balanced) to $+1$ (always long), and subtracts nothing until the tilt passes `max_net`, then subtracts proportionally more, reaching `net_penalty` at a fully one-sided book. The reason it exists: in crypto a long-only basket is not an alpha, it is the market with extra steps, and during a rising TRAIN and VAL it scores well for reasons that have nothing to do with the formula — so the search will mine one given the chance. The default cap of `0.5` is deliberately generous: a genuine cross-sectional alpha measures around `0.05`, so only genuinely lopsided books are charged. Lives in the `[fitness]` section of `evolution/config.ini`, which the node re-reads at the start of every round, or as `ALPHANODE_FIT_NET_PENALTY` and `ALPHANODE_FIT_MAX_NET`.

Watch out. This changes what gets MINED, not how anything is displayed: rows already in your library keep the fitness they were mined under, so a library that spans a change of this setting mixes two populations. The tilt it charges for is shown per row by the `L/S %` column (6.18), but that column measures TEST while the guard measures TRAIN and VAL, so the two can differ on a formula whose direction drifted. Setting `net_penalty` to `0` turns the guard off entirely and is the honest way to A/B it.

Part 05

Date segments, and the honesty discipline

Four dates define the three stretches, and by default all four follow today. The search may stare at TRAIN and VAL as long as it likes; TEST is walled off and only ever reported — never used to pick, rank, seed or fine-tune. That single rule is the difference between a number you found and a number you fitted — and the session contract at the end of this part is what keeps a library measured on one rule.

5.01 ★ TRAIN start default follows today: the whole window back – 2,400 days on 1d

range YYYY-MM-DD · checked as you type and again at Start

The first bar of the window the app builds. Every feature a formula reads sits on a grid that starts here, so nothing earlier reaches the signal or the prices the simulation walks. The one exception is the volatility estimate that scales each pair's position: it is measured on the raw downloaded history rather than on that grid, so the moment you move TRAIN start later than the date your snapshot was fetched from, it reaches one vol window — 30 bars, 2,880 at 15m — back past this date. Earlier means more history and more market moods to prove a formula across, at the price of speed: every candidate is simulated over the whole TRAIN-start-to-TEST-end span on every evaluation (TEST included — simulated, just never scored), so doubling the span roughly doubles the cost of a round. There is a warm-up tax at this edge: a formula may look back 200 bars, rolling operators start halfway through their window, the first five bars are never tradable, and when the download starts here too the volatility estimate has nothing to measure for its first 30 bars (2,880 at 15m), so every pair is sized off the same 0.005 floor until that window fills. Leave a couple of hundred bars of runway before the part of TRAIN you care about.

Left alone, the box shows today minus the bar size's whole window (5.05) and advances by one day every day. Type a date and it is pinned there. This field decides the length of the whole window, so it is the one the per-timeframe ceiling argues with: TRAIN start to TEST end may not exceed 3,000 bars on 1d, 16,000 on 4h, 40,000 on 1h or 96,000 on 15m — roughly 8.2, 7.3, 4.6 and 2.7 years. The sliding window uses 80% of that; the cost of a round is linear in bars, and the same four calendar years is 1,460 daily bars but 140,000 fifteen-minute ones. Exceed it and a red line appears under the fields saying how many days to move TRAIN start, and Start refuses until you do.

Watch out. The download never asks for your TRAIN start — it fetches from the timeframe's own history start. Pin TRAIN start earlier and nothing warns you: the missing stretch is filled with a frozen price, every formula sits flat through it, and because flat bars stay in the scoring window every Sharpe is dragged down. If under 10% of TRAIN is left trading, the activity filter rejects every candidate and the round finds nothing.

5.02 ★ VAL start default follows today: 50% of the window back – 1,200 days on 1d

range strictly between TRAIN start and TEST start – enforced

The wall between TRAIN and VAL, and the most powerful anti-overfitting lever in the app — a date, not a number. An alpha's score is the LOWER of its TRAIN and VAL Sharpe, so a formula that made all its money in one kind of market is judged by the kind where it did not. The sliding default puts it halfway through the window, so TRAIN and VAL+TEST are the same length; pin it where the market's character genuinely changes if you know better. Move it later and TRAIN gets richer while VAL thins into noise; earlier and VAL

quietly becomes the dominant judge. Keep VAL long enough to mean something — the engine's own error bar puts about ± 1.2 around a one-year VAL measured at Sharpe 1.0, ± 0.87 at two years, ± 0.61 at four.

Watch out. Setting "Robust blocks" to 2 or more silently retires this date from the score: the engine re-cuts the whole TRAIN-to-TEST span into equal slices and the VAL boundary plays no part in the number any more. It still decides validity and the VAL row still shows on the champion card, so nothing looks different while the number you are optimising has changed meaning.

5.03 ★ TEST start default follows today: 30% of the window back – 720 days on 1d

range **strictly between VAL start and TEST end – enforced**

The wall. Everything from here to TEST end is held out: it never enters the score, never ranks the shortlist, never seeds a warm start, never guides window fine-tuning. Treat its length as a budget you are spending — push it later and the search gets more data while your verdict gets noisier; pull it earlier and you buy a long, trustworthy TEST at the cost of a thinner search. The sliding 1d window gives roughly 1,200 TRAIN, 480 VAL and 720 TEST bars — two years of TEST ending today. For a daily search keep TEST to at least 1.5–2 years of calendar. Pin it (type a date) when you want the wall to stay put while the rest of the window keeps sliding.

Watch out. A TEST under 30 bars is refused outright, not degraded quietly: the note under the date fields counts the bars as you type and says that a segment that short has nothing to measure in it, and Start stops with a Date segments error until you widen the window. Above that floor the degrading is quiet: under 5 traded bars the TEST cell just shows "—"; the direction columns each need 30 bars in their bucket and blank out first; and sorting by TEST OOS drops every row with no TEST number, so alphas appear to vanish. Because this date is part of the contract, re-cutting it on a non-empty library is refused at Start — the dialog offers to revert or to clear the node, which is the honest choice.

5.04 ★ TEST end default today

range **strictly after TEST start – enforced; still not clamped to the data you actually hold**

The last bar the app loads, for everything — search, leaderboard numbers, portfolio build, CSV export. By default it IS today: the box shows the current date and moves with the calendar, so the fresh bars Start keeps downloading are simulated the day after they close. Pin it (type a date) only to reproduce an older run.

Watch out. Setting it past the end of your data costs Sharpe silently: the last close is carried forward, those bars drop out of eligibility, every strategy returns exactly zero across them, and TEST Sharpe is multiplied by roughly the square root of the share of real bars. That is now the pinned case only — the default never runs ahead of a snapshot that Start refreshes.

5.05 The window at each bar size

Every bar size takes the same shape: a window of 80% of its ceiling on bars, ending today, cut TRAIN 50% / VAL 20% / TEST 30%. 1d — 2,400 days (6.6 years): about 1,200 / 480 / 720 bars. 4h — 2,133 days (5.8 years): 6,400 / 2,560 / 3,840 bars. 1h — 1,333 days (3.7 years): 16,000 / 6,400 / 9,600 bars. 15m — 800 days (2.2 years): 38,400 / 15,360 / 23,040 bars. The ceilings themselves are 3,000 / 16,000 / 40,000 / 96,000 bars, so a window you pin by hand has 20% of headroom before Start refuses it. The pattern is

deliberate: finer bars trade calendar span for bar count. Switching the timeframe selector refills all four dates from this rule, and a settings file still holding an older version's fixed defaults is moved onto the sliding window on launch (a window you typed yourself is never touched).

Watch out. The extra bars do NOT buy a more believable verdict — the error bar on a Sharpe depends on calendar YEARS, not on how finely you sliced them. At Sharpe 1.0 the TEST windows carry roughly ± 0.7 (1d), ± 0.9 (4h), ± 1.2 (1h) and ± 1.6 (15m). A 15m TEST holds 23,000 bars and still cannot tell a Sharpe of 1 from a Sharpe of 0. Read intraday TEST as a sanity check, not as evidence, and lean much harder on the forward track there.

5.06 Working with a re-cut

Once you want a different TEST start or TEST end, the honest options are to clear and re-mine on the new cut, or to treat the surviving library as historical and never compare its numbers against new rows — and since 3.1 the app enforces the first: a library mined under one cut refuses to be mined into under another (5.08). In practice: mine under one cut, read TEST once, enrol whatever you liked into the forward track — an enrolled entry freezes its own dates and settings, so it keeps accumulating genuinely unseen bars whatever you do to the fields afterwards — then re-cut and clear. Save a named session first: it snapshots the settings file and the contract too, so both the old library and the old cut can come back.

Watch out. A sliding window is not a re-cut: "auto" equals "auto" tomorrow, so a library mined on the default dates keeps accepting rows as the calendar advances, each scored on the window of its own day. Two rows both reading "+1.42" may have been measured on windows a season apart. Pin the four dates when that matters — before the first row is mined, because pinning afterwards is a contract change.

5.07 by — how portfolio members are picked

default TEST on the dashboard · combo, auto size, on the simple screen

range TEST | fitness | combo

Combining alphas is a selection step like any other, and this is where the discipline is easiest to break by accident. "fitness" picks members by the worse of TRAIN and VAL, so TEST never enters the choice and the combined TEST curve is genuinely out-of-sample — you can quote it. "combo" is better still: instead of the N best individuals it searches for the best COMBINATION of N, taking a pool of the top alphas by fitness ($4 \times N$, never under 12 or over 30), simulating the pool once, then running a greedy-then-swap search that maximises the mix's Sharpe on TRAIN+VAL only. Because the goal is the mix's Sharpe rather than any member's, it naturally hunts members that do not move together. The simple screen always builds this way, and lets the search choose the SIZE as well: the pool is the top 30, and the mix of 2 to 10 that scores best on TRAIN+VAL is kept.

Watch out. The dashboard selector opens on TEST — the one option of the three that breaks the discipline. What is still real in a TEST-picked portfolio is the diversification GAIN: combined beating every member is a structural fact about blending uncorrelated signals. The LEVEL of the curve is not.

5.08 ★ The session contract

default recorded at the first Start that mines into an empty library · per timeframe

A library is only comparable with itself if every row was scored under the same rules. Nine settings decide what a score MEANS — the pairs, the four dates, target vol, fee, the win-rate objective and robust blocks — and they are marked ★ in the settings panel. The first Start that mines into an empty library

records them (state/session_params.json, one snapshot per timeframe). Every later Start compares the panel against that snapshot: a match starts the node, a difference opens a dialog listing each changed field as was → now with two buttons — "Revert to session settings" puts the recorded values back into the panel, "Clear node & start fresh" wipes the library so the new rules get a library of their own. Closing the dialog leaves you editing, with the node not started.

While the node runs, nothing in the panel can be edited at all — the fields grey out, the pair chips lose their ✕, the timeframe box is locked — so a mid-run change can never slip into the next round.

Everything that is not ★ (population, generations, resources, seed, the breeding knobs, the penalties, the leaderboard size, warm-start) is free to change between runs: it changes how hard the search looks, not what a number in the table means.

The date sentinels are compared as sentinels: a window that follows today is the same contract tomorrow, so the sliding defaults never trip it. Pinning a date that was sliding — or unpinning one — is a change.

Watch out. A library from before 3.1 has no snapshot; the first Start after the upgrade records whatever the panel holds at that moment, without checking it against the rows already there. If you changed settings since those rows were mined, clear first. The contract is also what Sessions restore: loading a session brings its settings file, so a saved library comes back with the rules it was mined under.

Part 06

The leaderboard, column by column

Four columns — fitness, TEST OOS, maxDD, CAGR — are stored numbers written when the formula was found. The rest are recomputed on demand by a background helper that re-simulates the formula on TEST only, using your pairs, target vol, fee and dates as they stand when it runs — but the answer is then cached under the formula text alone, so a row already measured keeps its old numbers when you change target vol, fee, pairs or the dates; only a restart, a timeframe switch or a fresh data download clears the cache and measures everything again. A dot (·) means "still computing"; an em dash (—) means "no honest number here".

6.01 ★ default always shown

A star means the formula is in your Favorites. Clicking anywhere in the cell toggles it — no dialog. Use it as your shortlist while scrolling a library of hundreds.

Watch out. The star set is keyed by the 6-char ID with no timeframe in the key, so the same formula text starred on 1d also shows starred on 4h. Sealed rows cannot be starred at all.

6.02 # default always shown

The row's position in the current sort — 1 is the top row of this sort, not a permanent rank. A place-marker for talking about rows, never a quality score, and not sortable.

6.03 fitness default always shown · the default sort

The score the search ranked by: the WORSE of the TRAIN and VAL Sharpe. The only column that is honest for picking — TEST never touched it. As a rough rule on daily bars, above roughly +1.5 is worth opening; well above +2.5 is usually too good and worth suspecting rather than celebrating.

Watch out. The tooltip does not mention that a concentration guard is on by default and docks this number when the book averages under 3 meaningful positions, nor that the value shown is BEFORE the size and similarity penalties the engine uses to pick tournament winners. A row mined under "Optimize by win rate" shows a percentage here and a Sharpe-mined row a signed number, whichever way the switch is set today, and clicking this header sorts rows mined under your ACTIVE objective to the top as a block — a 0.57 win rate and a +1.8 Sharpe are not on the same ladder.

6.04 TEST OOS default always shown

The Sharpe on the held-out slice — the honest out-of-sample verdict on a single formula. A dash means too little to measure: fewer than 5 TEST bars where the strategy actually made or lost anything. Read it, do not select on it. Expect it to sit well below fitness; a formula whose TEST is CLOSE to its fitness is the rare good sign, and a wide gap is the classic overfitting tell.

Watch out. Clicking this header does far more than re-sort: the whole library is re-read and re-ranked by held-out TEST, every row with no TEST number is dropped, and the heading changes to "cherry-picked ▲". That is peeking by construction. Click "fitness" to get the honest population back.

6.05 maxDD default shown by default

Worst peak-to-trough fall of the TEST equity curve, as a negative percentage. The pain metric — this is what decides whether you could actually hold the thing. Two formulas with the same Sharpe and drawdowns of -15% versus -45% are completely different products. It moves with your target-vol setting: halve the target and the drawdown comes down roughly in step.

Watch out. Unlike the recomputed cells, this prefers the value STORED when the alpha was mined, so changing target vol, fee, pairs or dates leaves maxDD and CAGR describing the OLD settings. The recomputed cells go stale as well: they are cached under the formula TEXT alone, with no vol, fee, universe or dates in the key, and that cache is emptied in only three places — when the app starts, when a data download finishes, and when the active library file changes because you switched timeframe. Writing settings does not empty it, so a row that has already been measured keeps the pairs, vol, fee and dates that were in force at the moment it was measured. Restart the app to re-measure the whole table under the settings you have now.

6.06 CAGR default shown by default

Compound annual growth rate on TEST. A dash means the equity ended at or below zero, so no growth rate exists. Pair it with maxDD, never read it alone: $\text{CAGR} \div |\text{maxDD}|$ is a quick sense of return per unit of pain. Like maxDD it moves with your target-vol dial, so it says as much about your risk setting as about the alpha's edge.

Watch out. Same stale-value trap as maxDD — taken from the library row first.

6.07 sortino default hidden — right-click a header to switch it on

Like Sharpe, but only losing bars count toward the wobble in the denominator; upside volatility is free. Look at it when a Sharpe seems unfairly low: a big sortino-to-Sharpe gap means the volatility Sharpe punished was mostly good volatility. If it is barely above Sharpe, the returns are symmetric and Sharpe was telling the truth.

Watch out. A dash here can mean the opposite of failure — with no losing bars at all the downside deviation is zero and the code returns nothing rather than infinity. In practice that almost always means an alpha that traded a handful of bars; cross-check $\text{tr/yr}\cdot\text{a}$.

6.08 T ↑ default shown by default · the header carries the bucket's bar count

TEST Sharpe measured only on bars where the market as a whole was trending UP. The regime is decided by the market, not the formula: an equal-weight basket of your universe has its drift measured over the trailing ~ 30 calendar days, and if that drift is statistically confident ($t \geq 1.28$) the bar takes the drift's sign. This is the regime question — does the alpha only make money when everything rises, in which case you have bought beta, not alpha? T ↑ and T ↓ both meaningfully positive is genuinely direction-neutral and much more valuable.

Watch out. The label is deliberately lagged one bar; without that lag a plain market-clone showed roughly $+1.3$ of pure T ↑-minus-T ↓ Sharpe out of thin air. Two different causes of a dash: under 30 TEST bars in the bucket (hits every row at once, since the split is the market's calendar) or the formula sat flat through the whole bucket (per-row).

6.09 T ↓ default shown by default

The same measurement on confidently DOWN-trending bars. The most informative of the three for crypto: most formulas quietly make their money long, and a positive T ↓ is the evidence that the alpha survives a bear leg. If you plan to run this through a drawdown, read this column first.

6.10 T ~ default shown by default

TEST Sharpe on FLAT bars — where the basket's drift was not confident either way. Usually the largest of the three buckets. This is where mean-reversion and relative-value formulas earn their keep and where trend-followers bleed. Strong in T ~ and weak in T ↑/T ↓ is a chop specialist: useful, but it underperforms exactly when the market is most exciting.

6.11 L/S /yr·a default hidden — right-click a header to switch it on

Long entries and short entries, per asset, per year, on TEST — "3.2/1.8" is roughly three new longs and two new shorts per coin per year. An entry is the moment the target weight crosses into long or short from flat or the opposite side. The alpha's directional personality: 4.0/0.1 is a long-only formula wearing a long/short costume and will die in a bear market whatever its TEST Sharpe says; near 2.5/2.5 is genuinely two-sided. Normalised per asset, so a 5-coin universe compares directly with a 30-coin one.

Watch out. Flipping straight from long to short counts as ONE entry, and holding for eight months counts as zero further entries — this measures decision frequency, not how much notional you push around.

6.12 tr/yr·a default hidden — right-click a header to switch it on

Trades per asset per year on TEST: the two L/S numbers added together. The activity dial. Low single digits means a slow position-holder where fees barely matter; a few hundred means an alpha whose profitability lives or dies on your fee setting — re-check TEST Sharpe with a realistic fee before believing it.

Watch out. It counts INTENDED entries, not fills: the simulator leaves a position alone whenever the required change is under 10% of the new target. Honest about how often the formula changes its mind, an overstatement of how often you would touch the exchange.

6.13 win ↑ default shown by default

The share of the formula's UP calls that came true. Every (bar, asset) cell where the formula held a long position at the previous close is a call that this asset rises over the next bar — exactly the bet the simulator books PnL on. The cell counts as right when the price rose, wrong when it fell, and as nothing when it did not move. Unlike T ↑, which slices the equity curve by what the MARKET was doing, this judges what the FORMULA said, asset by asset. Fees and position size do not enter — this is pure directional skill on the long book, which is why a formula can show a healthy win ↑ and still lose money: it calls direction well but its rare misses are large.

Watch out. Two evidence floors: 30 long calls on TEST AND at least 5 where the price actually moved. A dash beside busy L entry counts means the assets it held barely moved. Note what the percentage divides by: only the calls where the price MOVED, not every held cell. On a normal run that is thousands of cells and a 52% is a real edge, but the floors let a formula through on as few as five judged cells — read this column beside L / S and tr/yr·a, and treat a high number from a barely-trading formula as the coin flip it is.

6.14 win ↓ default shown by default

The same precision on the formula's DOWN calls — cells where it held a short position, right when the price fell over the next bar. Same two evidence floors as win ↑. Read the pair together: win ↑ 56% against win ↓ 48% says the short book is guesswork riding along with a genuine long edge — a long-bias machine whatever its entry counts suggest. A formula that never shorts shows a dash here: it made no down calls, so there is nothing to be accurate about.

6.15 ID default shown by default

A stable 6-character fingerprint of the formula (the first six of its md5), sitting immediately after # at the left edge of the table. The same formula always gets the same ID. This is the handle used everywhere else — the forward track, favourites, the search box, both CSV exports.

Watch out. The two exports do not print it the same way: "Export table" writes the 6 characters you see, "Export full library" writes 12, so a plain text match between the files fails unless you trim.

6.16 formula default always shown

The alpha itself, in full. The column is sized to the widest formula rather than truncated, so long ones bring out the horizontal scrollbar. Length is itself information: the search subtracts a penalty per node, so a short formula that still made the library beat that handicap and is usually the more robust one. Click a cell to make the text selectable, or use Copy formula.

Watch out. Absolute SCALE is meaningless — a formula and the same formula times 100 are literally the same strategy, so do not read big constants as "aggressive". A locked row shows "locked" here and dashes in all nine recomputed cells, while fitness, TEST OOS, maxDD and CAGR still show real numbers, because those were stored before it was sealed.

6.17 L/S % default shown by default

How the book SITS, as a long/short split of its exposure on TEST — 50/50 is market neutral, 100/0 never shorts, 0/100 never goes long. It is time-averaged and inverse-vol weighted, meaning it uses the same weights the simulator trades: a long in a quiet coin carries more dollars than a long in a wild one, so the split reflects money at risk rather than a count of positions. Read it as the honesty check on every other number in the row. A cross-sectional alpha earns its living from the SPREAD between coins and should sit near 50/50; the further a row drifts from that, the more of its result is simply the market, and in a rising market that flatters TEST OOS, CAGR and win% all at once. This is different from L/S /yr·a, which counts how many positions were OPENED on each side — a formula can open equal numbers of longs and shorts and still hold a lopsided book.

Watch out. A wrapper is often the culprit rather than the idea: a running maximum over a rate of change is almost always positive, so `ts_max(ts_roc(close))` sits around 84/16 however sound the underlying signal is. Wrapping the same signal in `cs_demean` pulls it back to about 50/50. The search charges for this past 75/25 — see 4.14 — but the column shows the tilt whether or not the guard is on, and shows it on TEST while the guard measures TRAIN and VAL.

Part 07

What you can do with a formula

Everything reachable from a leaderboard row: the equity window and the five buttons inside it, the table's own menus and exports, and the two buttons that can destroy work.

7.01 Double-click a row → the equity window

Growth of \$1 on a log scale, shaded into TRAIN | VAL | TEST, with an equal-weight buy-and-hold basket of your universe alongside; all curves net of fees. Stretches where market volatility was above its own one-year median are tinted, and a lower panel tracks the unrealised profit of open positions. The header repeats Sharpe / CAGR / drawdown per segment. You are looking for a curve that keeps its SHAPE across the TRAIN→VAL→TEST boundaries rather than one that made its money in a single lucky stretch. It is a live matplotlib canvas — wheel to zoom, drag to pan.

Watch out. A double-click landing inside the ★ cell never opens the chart: each of its two clicks toggles the star, so nothing appears to happen. If a double-click does nothing, you hit the first column.

7.02 Passport

Builds an explanation of the formula rather than a measurement of it: the expression as a labelled tree, plain-English reading steps, the signal plotted against one asset's price, an ablation showing which inputs it really depends on, its holding period and long/short balance, and which classic archetype its returns most resemble. No network, no AI — a fixed translation of the operators. Use it when a formula looks good and you want to know WHY before trusting it: an alpha you can describe in a sentence is one you can reason about when it stops working.

Watch out. An explanation is not evidence. A convincing story does not make the TEST number more real — the ordering is always TEST first, forward track second, story third.

7.03 PDF report

A four-page analytics dashboard written where you choose: headline numbers with equity and drawdown; exposure, long/short balance and turnover over time; the structure of the weights plus a monthly-return calendar; and a TRAIN/VAL/TEST breakdown with attribution and written conclusions. Every page is stamped with the date, the time and the 6-char ID. This is the artefact to keep or hand to someone else — the app's state can be cleared, a dated PDF cannot. The Portfolio card has its own button that does the same for a combined book.

7.04 Serve signal (API)

Starts a small local web service that keeps this formula's live target positions available as JSON: it fetches fresh closed candles from Binance for your universe, recomputes the weights and serves the latest bar at /signal, with /health reporting its age. It claims the next free port from 8799 and appears in the SIGNAL API card. At most 10 services run at once — each is a whole engine process re-simulating on every refresh — and the eleventh is refused until you free a port; rows whose process already exited do not occupy a slot. This is the bridge to an execution bot — point your code at `http://127.0.0.1:<port>/signal`. It uses the same target vol and fee as the leaderboard, so what it serves matches what you were looking at; it refreshes every 15 minutes on 1d and 4h, every 5 on 1h and 15m.

Watch out. On daily bars the response carries "leverage" separately and the weights are the relative book — multiply to get real sizes. On intraday the weight already includes leverage and the served leverage is a flat 1.0. It is weight $\times$ leverage $\times$ equity that stays identical between the two, not the weights.

7.05 Download signals (CSV)

The full position history: one row per asset per bar where it held a meaningful position, labelled with its segment, the ticker, LONG or SHORT, the signed weight as a number and a percentage, and that asset's OHLCV on the bar. A dialog then shows what the strategy would be holding on the very last bar — the quick "what do I buy right now" answer. This is how you take the strategy out of AlphaNode.

Watch out. These weights are the normalised target book (absolute values sum to 1 across a bar) taken BEFORE volatility targeting. They are relative allocations, not position sizes — size directly from this column and you are running at whatever leverage you happened to pick.

7.06 Forward track +

Enrols this alpha: the strategy is frozen exactly as it stands — formula, universe, target vol, fee, timeframe — and from then on paper-traded once per closed bar on freshly downloaded Binance data from a clean \$10,000. Do this the moment a formula survives TEST and looks explainable. Everything else in the app is measured on data that already existed; this is the only clock that runs forward. Give it weeks before reading anything into the number.

Watch out. Enrolling the same formula twice is refused rather than starting a second track. Delete removes the strategy AND its whole paper history permanently. Steps only happen while something is running: the app checks every five minutes while it is open, and a mining node on the same state folder — headless or in Docker — steps the track itself, on every timeframe, so it takes both being down for the track to stall. Nothing is backfilled either: the next step jumps straight to the newest closed bar and the frozen book rides through the gap, which costs you most on 4h/1h/15m. Fees also weigh proportionally more on short bars, which is what the enrol dialog warns you about there.

7.07 Chart (Forward Track panel)

The forward equity curve of an enrolled strategy: one point per live paper step since enrolment, with the starting capital as a dashed line. This is the only curve in the app that was never fitted to anything — every point was computed on data that did not exist when the formula was found. It is short and noisy for weeks, and that is fine; the value is that it cannot lie to you.

Watch out. History is append-only and nothing is recomputed backwards: if the app was closed for a week, the gap stays a gap in the line.

7.08 ★ Favorites

The button in the leaderboard heading opens your starred list: date added, ID, the timeframe it came from, fitness, TEST OOS and the formula. Double-click for the equity chart. The starred formulas live in their own file outside the library, so they survive "Clear all history": the whole row is copied at star time, which is what keeps it readable after the library row is gone. They belong to the SESSION, though — saving a session saves its stars, and loading one replaces the stars on screen with that session's. A star

points at a formula in a particular library; inheriting it into a workspace mined on another basket, another cut or another timeframe said nothing true. Star anything in this session you would be upset to lose, and save the session to keep it.

Watch out. Loading a session archive saved by an older build leaves you with no stars at all — those archives carry no favorites file, and a load replaces every session-owned file wholesale rather than merging. A favourite is also a snapshot: the whole row is copied at the moment you star it and never refreshed. Change target vol, fees or dates and the leaderboard's numbers move while the Favorites window's do not.

7.09 Copy formula / Copy formula + metrics

Right-click a row for both. The plain copy (also Ctrl+C) puts the bare expression on the clipboard — use it when pasting into code. The +metrics version adds a line with the fitness value, a note if it was mined by win rate, and the TRAIN, VAL and TEST Sharpe ratios — use it when recording a candidate in notes, since that one line stays readable months later.

7.10 Right-click a header → the Columns menu

default on: ID, maxDD, CAGR, T↑, T↓, T~, L/S %, win↑, win↓ · off: sortino, L/S /yr·a, tr/yr·a

A checklist of the twelve optional columns; your choice is saved and reused next launch. ★, #, fitness, TEST OOS and formula are always present. Turn L/S /yr·a and tr/yr·a on the moment you start caring whether an alpha is genuinely two-sided or how much its fee assumption matters. The plain win% column was retired in 3.1: the share of profitable bars sat at 47-52% for every row and told you nothing — win↑ and win↓, the precision of the formula's calls by side, are the columns that discriminate.

Watch out. Column ORDER is fixed and follows the menu, not the order you tick things in — so switching sortino on inserts it mid-table rather than appending it at the right. ID is the one exception: it is pinned immediately after #, because it names the row instead of scoring it. Hiding a column only hides it: sorting still uses the hidden value, and "Export table" writes every leaderboard column whatever is on screen (win% included — it lives on in the export). The full-library export is the exception — it never carries the recomputed ones.

7.11 Sorting default opens on fitness, descending – the sort is not remembered between launches

Clicking a header sorts by it, best first; clicking again flips. Everything except ★, # and ID is sortable, formula included (alphabetically). Only fitness and TEST OOS change WHICH formulas the table contains; every other column simply reorders what is already there.

Watch out. Sorting by any recomputed column forces the app to compute that statistic for EVERY row in the table, not just the visible ones — on a large library that is a long job with a stalled-looking table. And rows with no value are treated as minus-infinity, so ascending sort puts all the dashes at the TOP.

7.12 Export table (CSV)...

Writes exactly the rows currently in the table, in the current order, with twenty-two columns — every leaderboard metric plus the TRAIN and VAL Sharpe and the objective the row was mined under — regardless of what is hidden on screen. The export for a specific working set: filter with the search box, sort how you want, then export those rows with all their analysis columns.

Watch out. The recomputed columns come from an in-memory cache holding only rows that have actually been on screen, filled roughly a screenful at a time. Export a 600-row table without scrolling and everything past the first screenful comes out blank in those cells. Scroll first, or sort by one of them to force the whole set.

7.13 Export full library (CSV)...

The CSV button in the heading: every alpha ever mined into this timeframe's library — no family dedup, no TEST filter, no on-screen subset — sorted by fitness, in nineteen columns including the round it was found in, its timestamp and the stored Sharpe / drawdown / CAGR / active-bar count for each of TRAIN, VAL and TEST. The archive export. It reads the library file directly, so it works while the node is mining.

Watch out. It carries NONE of the recomputed columns — no sortino, no regime splits, no trade counts, no win rates. And it sorts strictly by the raw fitness number, so in a mixed library every win-rate row (a value under 1) sinks below every Sharpe row regardless of quality.

7.14 ► Build portfolio default top 6, by TEST · neither is saved – both reset every launch

Runs N alphas together as one combined book and shows the combined curve against buy-and-hold. Once a build finishes the member list fills in under the metrics — which alphas went in and how each does alone — and four buttons unlock: CSV (combined signals), Serve (one API for the whole book), PDF, and Track (enrol the whole portfolio into the forward track). Prefer "fitness" or "combo" over the default — see the entry on the by selector. On a sealed library the build is a preview (8.06) and none of the four buttons unlocks.

Watch out. Because neither setting is persisted, a habit of building by fitness has to be re-established every session. The member list joins fitness and TEST OOS from the library by formula text, so a member whose library row is no longer there keeps its place in the list with dashes in those two columns, and double-clicking it cannot chart it. "Clear all history" is not one of those cases: it deletes the built portfolio along with the library, and the card comes back empty.

7.15 "families only" switch default off – every alpha shown

Collapses the table to the best representative per family, where two formulas are the same family if their text is 80% or more similar. Turn it on when the search has converged and the top of your table is fifty cosmetic rewrites of one expression; turn it off to see the whole population or to search by ID. Both sets are cached in the background, so the switch is instant — and unlike the sort order, this choice is remembered across restarts.

Watch out. Families mode is capped twice: it stops at 20 distinct families and only ever examines the top 500 rows while collecting them. A genuinely novel formula at rank 600 is invisible here — it is a view of the head of the list, not a summary of the library.

7.16 Clear all history / Clear node

The red button in the settings panel, and "Clear node" in the simple header — the same action. Deletes the mined alpha libraries, the round history, the status file, the built portfolio, the session contract, the cached return series and the cached equity images, after showing a count of what will go. Your search settings, your starred favourites and the forward track survive — enrolled strategies keep stepping

through the clear. The clear also starts a **NEW** session id (in the header), so forward entries enrolled before and after it are distinguishable. It refuses to run while the node is mining. This is also the way out of a contract conflict: with the library gone, the ★ fields may be set to anything.

Watch out. It clears **EVERY** timeframe, not just the one you are looking at, while the counts in the dialog are read from the **CURRENT** timeframe only — so a dialog saying "40 found alphas" can be about to delete several thousand. This is the single most destructive button in the app. The dialog does say the deletion is irreversible and tells you to save a session first — take it up before you confirm, because nothing else offers a way back.

Part 08

The simple screen — the node on autopilot

The screen a fresh install opens on: pairs, bar size, one button. It runs the same node, the same library and the same portfolio builder as the dashboard, with the choices made for you — and it is where an un-activated node shows what it can and cannot do. "⚙️ Advanced" in the header switches to the full dashboard at any time; whatever is running keeps running.

8.01 The header

default a fresh install opens here; a settings file from an older version opens the dashboard

Left to right: the brand, then a green "● activated" pill once the node has a subscription key and a green "serving :8800 :8801" pill while any signal service is up; the pair chips — the first six of your universe, "+N" for the rest, and "edit" which unfolds the pairs card (up to 20 pairs, the same chip editor as the dashboard); the bar-size box; then "Activate" (only while no key is known — an activation done on the dashboard, or on any other install on this machine, hides it), "Clear node", "⚙️ Advanced", the theme picker and the run pill. Everything else — population, dates, fees — is the dashboard's settings panel; the simple screen reads the same settings file and never changes them.

Watch out. The pair chips and the bar-size box are ★ contract fields like their dashboard twins: locked while the node runs, and a change after a stop is met by the same conflict dialog at the next Start.

8.02 ► Start node (the run pill)

One button, three jobs. Idle, it reads "Start node" and does exactly what the dashboard's Start does — checks the dates, checks the contract, downloads a missing or stale snapshot, launches the node. During a download it reads "Cancel download" and a click aborts the fetch. While mining it shows "round N" over the best fitness so far and the round's expected time left, with a ring that fills with the round's REAL progress from the node's status — when the node has no estimate yet the ring stays empty rather than faking one — and a click is Stop.

Watch out. Stop abandons the round in flight exactly as on the dashboard (1.06). Wait for the event line to show "✓ round N" if the round found something you want kept.

8.03 Search card

The state line ("Search · round 12 · refining best", or "stopped — the report below is final"), three tiles — the best fitness, alphas found, the generation in progress — and the chart: the solid line is the best fitness after each round, the dashed line the best held-out TEST found so far, a running maximum over the whole library by the round that found each row, so its last point agrees with the top of the Champions card sorted by test. Under the chart runs the node's latest event — a new best, a finished round, a warning.

Watch out. Fitness and TEST are different measurements on different years: a rising solid line with a flat dashed one is the search learning TRAIN and VAL without finding anything that holds out of sample. That is the normal shape of a maturing library, not a fault — and the honest signal to widen the universe or move on.

8.04 Champions card

default 8 rows · by fitness

The top of the library, formula by formula, with its fitness and its held-out TEST Sharpe. Click the "by fitness" label to switch to "by test": fitness order is the node's own top list, test order ranks the WHOLE library by TEST — the same population the dashboard shows when its TEST OOS header is clicked — so a strong-TEST formula sitting below the fitness cut still appears. TEST cells are green from +1.00, red below zero. On an un-activated node each row reads "locked · Sharpe +1.04 · win 54% · PnL +29%" — the row's stored TEST numbers — instead of the maths.

Watch out. "by test" is a look, not a selection: nothing here feeds the search or the portfolio. The moment you carry a formula somewhere BECAUSE it topped this list, you have picked on TEST.

8.05 Portfolio card default combo · auto size 2-10 · rebuilt after every finished round

The autopilot's portfolio. As soon as the library holds two alphas, every finished round triggers a build, and one more build runs after Stop — that last one is the report the card calls final. The selection is always combo (5.07) with the size chosen by the search: from a pool of the top 30 by fitness, the mix of 2 to 10 members whose equal-weight Sharpe is highest on TRAIN+VAL — TEST never enters, so the headline reads clean. The line shows TEST CAGR, max drawdown and buy-and-hold's Sharpe for comparison; the chart is the combined equity over TRAIN, VAL and TEST with the segment bands, the basket in grey behind it. Three buttons: Signal API (serve the whole book), Report PDF and Forward track.

Watch out. Never two builds at once: a round that finishes while a build is running is simply skipped until the next one. And the card builds on the DASHBOARD's state too — the two screens share one portfolio file, so a build you start there replaces the autopilot's.

8.06 The sealed preview (no activation key)

Without a key the node cannot hand the builder a formula, so it hands it the next best thing: for every champion it mines it also caches that champion's realised return series (state/series/, the top 100 by fitness). A return series carries no forward signal — it ends at the last closed bar and never says what to hold next — so nothing tradeable leaves the vault, yet it is exactly what the combination search needs. The card then runs the same search over the cached series and mixes the members equal-weight: the same objective the search scored, so the pick and the report agree. The card says "sealed preview", the note under it says why, and Signal API, Report PDF and Forward track stay disabled — the document holds members, metrics and equity, and no formulas, weights or positions.

Activate (the header button) and the next build uses the real engine on the revealed formulas; the numbers move a little, because the real book applies a second volatility-targeting layer over the mix.

Watch out. Rows mined before 3.1.2 have no cached series and do not enter the preview until activation; a library with fewer than two cached champions says so in the note instead of building. And a preview is not a book: the equal-weight mix of the members' own vol-targeted returns is what the search optimises, not what Serve would trade.

8.07 Serving — the autopilot default after Stop only · up to 3 services

When the node stops and the final portfolio is built, the autopilot fills up to three services: "auto_portfolio" — the whole final book — and "auto_top1" / "auto_top2", the two best formulas by held-out TEST from the library, a near-clone of the top not counting as a second champion. Each service is the same local JSON API the dashboard's Serve starts (7.04), and its row on the card shows the port, the

label, its health, the live book of positions and a "× Free port" button. A service is replaced only when its content changed — the same members serve on through a restart — and an auto_top slot whose champion left the top is stopped.

Watch out. Nothing is served mid-run, whatever the round number: a running search keeps the services it found at the last stop. Services YOU raised — any other label, on the dashboard or from an older session — are never touched, and at the three-service cap the autopilot backs off with a note asking you to free a port rather than freeing one itself. Services outlive the app: "× Free port" on a row stops that one, "× Free all ports" in the card's heading stops every one (after a confirm) — the same actions as the dashboard's SIGNAL API card, and the autopilot re-serves only after the next Stop. On a sealed library nothing is servable, so the card stays empty — the portfolio note explains.

8.08 Clear node

The same action as the dashboard's Clear all history (7.16), with the same confirmation: every timeframe's library and history, the status, the portfolio, the contract and the series cache go; settings, favourites and the forward track stay. The cards reset on the spot — an empty chart, an empty Champions list, "no portfolio yet" — so nothing stale outlives the files it came from.

Watch out. It refuses while the node is mining. And it is the only way to change a ★ setting once a library exists — which is by design, not a limitation.