IHLLS (FINCH_recon) - Software Logic Chart and Description

FINCH_recon - an automatic quality check and reconstruction tool for IHLLS-2L FINCH recordings
FINCH_recon is a single-file, dependency-free MATLAB program that turns the raw output of an IHLLS-2L light-sheet microscope into checked, quantified, publication-ready images. The program addresses a practical bottleneck in FINCH (Fresnel Incoherent Correlation Holography) light-sheet microscopy: the reconstruction step is sensitive to a number of instrument settings and acquisition details, SLM lens focal lengths, camera distance, galvo scan range, detection focus position, and phase-step alignment, and errors in any of these can silently degrade the result. FINCH_recon reads a four-phase recording, diagnoses these conditions from the data itself, reconstructs the volume, and writes a plain-language report that states, per recording, whether the data work and what to fix if they do not.

The program is invoked with a single command, FINCH_recon, from the folder that holds the data folders. It discovers every recording below that folder and processes bead recordings first (they calibrate the SLM lens pair), then time series, then cell recordings (which inherit the measured lens pair). Whether a recording holds beads or cells is read from its own folder and file names and checked against the data, since a dense sample lights up a large part of the frame while beads do not. No configuration is required for a standard dataset; all optics and processing parameters have defaults matching the IHLLS-2L bench and can be overridden through a settings structure or name and value arguments. The program runs in MATLAB without any toolbox and also in Octave.

The processing chart (Fig. 1A) has three stages. **First, data integrity and instrument diagnosis.** The four phase stacks are checked for axial alignment by cross-correlating their per-plane intensity curves; a shift of more than a quarter plane raises a fix request. Camera offset and the up-to-ten-percent gain differences between phase steps are measured and removed. The complex hologram $H = (I_0 - I_{180}) + i(I_{90} - I_{270})$ and its null complement $I_0 - I_{90} + I_{180} - I_{270}$ are formed per plane; the null carries no holographic signal and therefore provides a per-plane noise estimate, so all results are expressed in noise units on a single absolute scale. The two "spot" planes, where one SLM lens alone focuses the light sheet onto the camera, are located from the plain (non-holographic) images; their positions give the galvo plane matched to the lens pair, verify the galvo step, and reveal whether the detection focus lies inside the scan. The lateral drift between adjacent planes distinguishes a galvo scan (sample fixed) from a stage scan (sample moving), which would otherwise invalidate the xz and yz projections.

Second, reconstruction. Beads are refocused individually. Each candidate bead is propagated over 111 signed distances logarithmically spaced from 0.3 to 120 mm, its distance is refined by parabolic interpolation of the focus metric in log distance, and each pixel takes the distance of its dominant bead. This yields the refocused bead image and a measurement of the effective SLM lens pair, from which lens 2 is back-calculated and compared with the nominal setting. The reconstructed amplitude |E| is stored, because it is linear in emitter brightness. The lateral width of each bead comes from a two-dimensional Gaussian fit, on a three times finer grid when the spot is narrower than five pixels, and the beads are paired with those of the LLS image once the orientation, scale and shift between the two cameras have been found, so the two widths are compared bead by bead. The tilt of the bead layer against the detection focal plane gives a galvo step that is independent of the spot planes. Cells are refocused plane by plane at the distance predicted by the two-lens model for that galvo plane, using the lens pair measured on the beads of the same run; this is necessary because dense cell samples give fringes of only one to two percent, too weak for per-object autofocus. Planes whose refocus distance is below 2 mm, next to the spot planes, are left out of the views and the summary numbers, interpolated across in the projections, and skipped by the time-series focus search.

Third, output and reporting (Fig. 1B, 1C and Fig. 2). The program writes the amplitude volume as uint16 TIFF stacks in three forms. finch_volume_planes.tif holds one page per galvo plane. finch_volume.tif is resampled to the voxels of the LLS stack, the LLS pixel in x and y and the same number of z pages as the LLS xz image (the LLS pixel in z when no LLS view is present), so the two stacks can be compared page by page in FIJI. finch_volume_LLS_frame.tif is the same volume turned to the LLS camera orientation. The two cameras see the field rotated by 90 degrees and mirrored; this was measured on cell

recordings against their LLS views, confirmed in FIJI, and is fixed in the program (in FIJI, Rotate 90 Degrees Right followed by Flip Vertically), and a search over the eight image orientations and both z directions by band-passed correlation with the LLS views is kept as a check and reported. The program also writes the plain volume finch_total.tif, the figure panels fig2 to fig7 and views_compare.png (rows IHLLS refocused, IHLLS plain and LLS; columns xy, xz and yz at the LLS pixel), finch_report.txt with every number and how it was found, csv tables of bead positions, widths and pairs, and, at the top level, finch_check.txt with one verdict block per recording followed by the main numbers in plain words. Each check carries one of four status labels, ok, FIX, CHECK or note (Fig. 2A).

Time series and tests. A time series is handled by the same file. Recordings with the LLS software names are grouped by stack index, channels 0 to 3 are taken as the phases 0, 180, 270 and 90 and this order is checked against the data, one galvo plane per time point is refocused over 0 to 15 µm from the detection focus, and the beads found in the first time point are followed, each reported as the background-corrected sum of |E| over a 7 by 7 pixel box at its in-focus distance (finch_time_beads.csv, finch_time_t01.tif, finch_time_xy.tif and fig5). The program ships with a selftest mode, which reconstructs a synthetic three-bead hologram and checks the sign convention, the autofocus, the width fit and a two-bead time series, and a lenses mode that converts a target emission wavelength into the focal lengths to enter into the SLM pattern program.

A key design choice is the explicit propagation sign convention, documented in the header: the program propagates with $\exp(+ik_z z)$, under which beads matched to the lens pair appear at a positive distance near $+z_h a/2$, where $a = \frac{1}{2}(z_h/f_1 - 1 + 1 - z_h/f_2)$. The convention is stated in the program header, and finch_report.txt lists the model refocus distance of every galvo plane in it, so the distances in finch_beads.csv are unambiguous. The program is written to be fail-soft: a recording that cannot be processed produces a FAILED block in finch_check.txt with the error message, and the remaining recordings are processed normally.

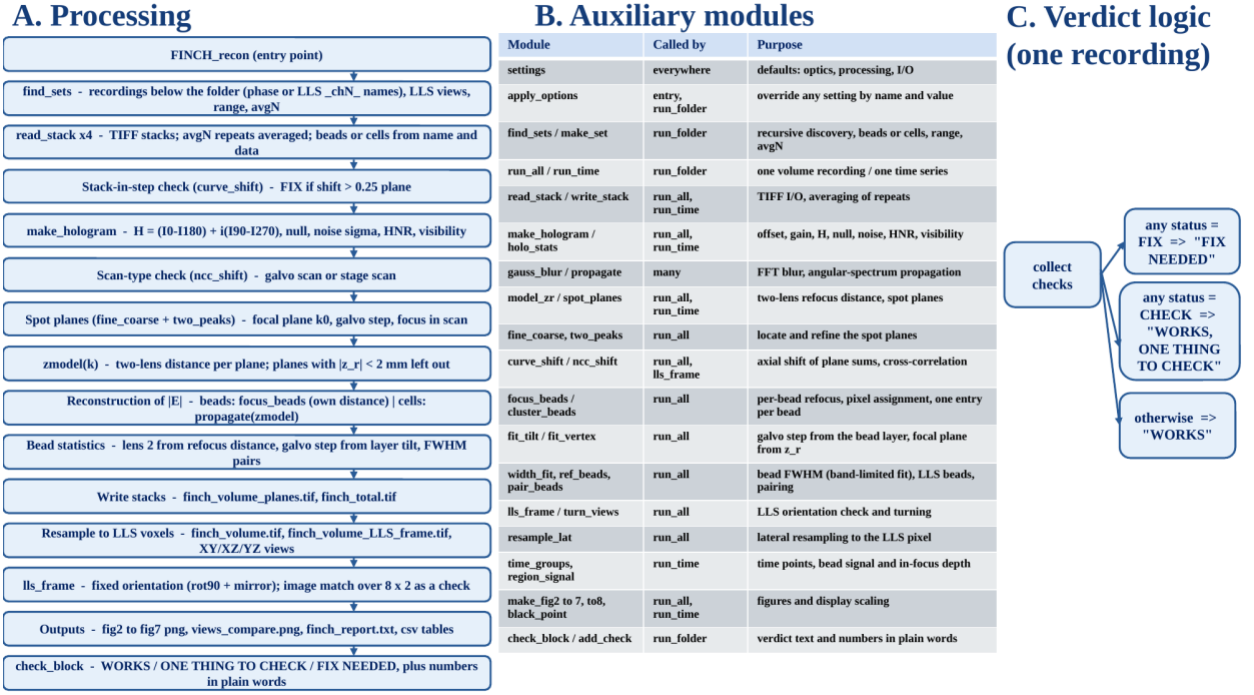


Figure 1. Logic chart of FINCH_recon (version of 14 October 2026). A) Processing of one volume recording, from file discovery to the verdict. B) The modules and what each one does. C) Verdict logic for one recording.

A. Check inventory

Check	Condition that triggers it	Status
four stacks in step	|z shift| > 0.25 plane	FIX
sample fixed, galvo scan	lateral motion >= max(4, 0.4 stage_px) px per plane	CHECK
focus inside the scan	spot plane at the edge, or k0 outside the central 40 %	FIX
focus inside the scan	two spot planes not found	CHECK
galvo range / galvo step	spot planes vs given range differ > 25 %; step from the bead layer tilt	CHECK / ok
holograms (beads)	no bead above 30 sigma^2, or none within 3 to 20 mm	FIX
holograms (cells)	hologram power < 10 x noise (15 to 30 px band)	FIX
lens pair on the SLM	refocus distance vs model differ > 15 %	FIX
beads or cells; lines up with LLS; bead widths	name and data disagree; no reference or image match disagrees with the fixed orientation; no pairs found	note / CHECK
time series: phase order, holograms, bead traces	data disagree with ch0..ch3 = 0, 180, 270, 90; refocus gain < 1.5; no bead kept	CHECK / FIX

B. External interfaces

INPUTS
Four TIFF phase stacks per volume: phase0/90/180/270 in the names,
or LLS software names with _ch0_ to _ch3_ = phases 0, 180, 270, 90
Optional LLS views _xy, _xz, _yz: orientation check, z pages, bead pairs
Time series: LLS software names, four channels per stack index

CALLS
FINCH_recon run everything below the current folder
FINCH_recon(folder) run below a given folder
FINCH_recon(folder, 'f2_mm', 298) override any setting
FINCH_recon('lenses', 570) focal lengths for another wavelength
FINCH_recon('selftest') synthetic test, prints PASS or FAIL
P = FINCH_recon('defaults'); FINCH_recon(P) struct interface

OUTPUTS
finch_check.txt (verdicts and numbers in plain words), finch_report.txt, csv tables
finch_volume.tif (LLS voxels), finch_volume_LLS_frame.tif,
finch_volume_planes.tif,
finch_total.tif, fig2 to fig7 png, views_compare.png
time series: finch_time_t01.tif, finch_time_xy.tif, finch_time_beads.csv, fig5

DEPENDENCIES: MATLAB base only (FFT, TIFF I/O, fminsearch). No toolbox.
Runs in Octave 8.

C. Key design choices

Sign convention, explicit and stated in the report:
propagate with exp(+i kz z)
beads at the focal plane appear at a POSITIVE distance near
+zh a / 2
a = (1/2) [(zh/f1 - 1) + (1 - zh/f2)]

Noise referenced to a single scale:
null = I0 - I90 + I180 - I270 carries no hologram
results in units of the noise sigma of H
widths and traces use the amplitude |E|, linear in brightness

Beads or cells:
beads: per-object autofocus over 111 signed distances, 0.3 to
120 mm
cells: per-plane refocus at the two-lens model distance
the choice follows the folder name and is checked on the data

Planes next to the spot planes (|z_r| < 2 mm):
left out of the views and the time-series focus search

Fail-soft:
a failing recording gives a FAILED block with the line; the
rest still run

LLS alignment:
fixed camera orientation, rot90 + mirror
(FIJI: Rotate 90 Degrees Right, then Flip Vertically)
image match over 8 orientations x 2 z directions kept as a
check
volume resampled to the LLS voxels, z pages from the LLS xz
image

Figure 2. A) The checks, the condition that triggers each one and its status label. B) Inputs, calls and outputs. C) Design choices, the sign convention, the noise units, beads against cells, the planes left out, fail-soft behaviour and the LLS orientation.