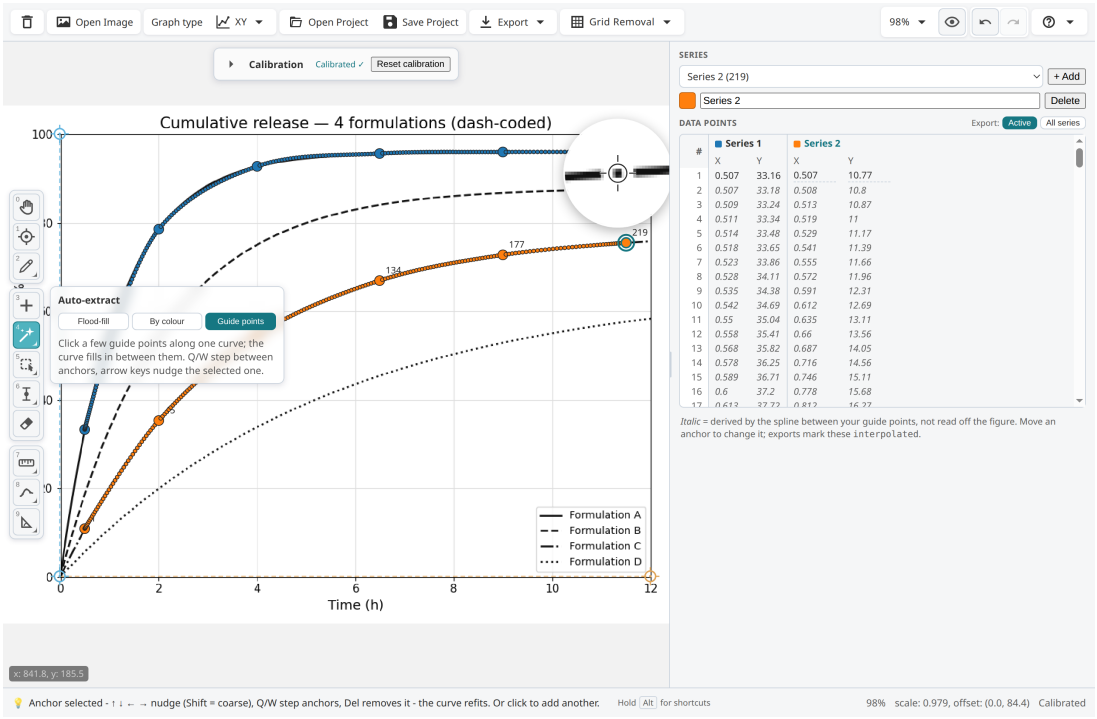
PLOTTRACER

User Guide

Turning figures back into numbers



Version 2.5 ▪ AGPL-3.0 ▪ Katalyst Nord, Stockholm

Contents

Chapter 1

Introduction

1.1 What PlotTracer is

An enormous amount of quantitative knowledge exists only as figures. Stress–strain curves, load–displacement data, fatigue-life plots, statistical distributions, economic time series, dose–response relationships: published in papers, reports, standards, and regulatory filings, locked inside a raster image, unavailable for reanalysis unless someone manually re-extracts the numbers. Plot digitisation is not an exotic edge case — it is a routine part of working with any published or legacy dataset.

PlotTracer is a free, fully offline, cross-platform desktop application for that job. It reads a figure, walks you through calibrating its axes, and lets you trace, correct, and export the underlying data — across eleven chart types, at the figure's own precision, with nothing invented and nothing hidden.

Three principles run through every part of the tool, and are worth knowing before you open your first figure:

- **Auditable.** The code that computes your extracted coordinates is open and inspectable. A black-box cloud service cannot be independently verified, and a proprietary binary cannot be scrutinised.
- **Available everywhere.** Linux, Windows, and macOS, installed from a single binary, working fully offline — including in air-gapped environments.
- **Honest about precision.** The tool records what the figure actually shows and never fabricates precision the source never carried. Where it cannot determine an answer with confidence, it says so, in the interface and in the export, rather than returning a plausible-looking number.

1.2 Who this guide is for

This guide assumes no prior experience with plot digitisers. It is organised so you can either read it start to finish, or jump straight to the chart type or task you need via the table of contents. Chapter 4 is a five-minute worked example; Chapters 5–6 are the full reference.

1.3 Scope of this document

This guide covers the desktop application's workflow, every chart type, analysis tools, and export. It does not cover building PlotTracer from source or its internal architecture — see the project `README` for that.

Chapter 2

Installing PlotTracer

PlotTracer ships as a single installer per platform, built directly from the source repository by continuous integration. There is no account to create and nothing is installed beyond the application itself.

Platform	Package	Notes
Linux	AppImage, `.deb`	`chmod +x` the AppImage, or `dpkg -i` the `.deb`
Windows	NSIS `.exe`	64-bit only
macOS	`.dmg`	Apple Silicon

Linux — AppImage

```
chmod +x plottracer_<version>_x86_64.AppImage
./plottracer_<version>_x86_64.AppImage
```

Linux — .deb (Debian / Ubuntu)

```
sudo dpkg -i plottracer_<version>_amd64.deb
```

The post-install step sets the sandbox permissions Chromium needs, so no manual `chown`/`chmod` is required afterwards.

> The macOS and Windows builds are currently **unsigned**. On first launch:
>
> - **macOS** — Gatekeeper will block the app. Right-click the app and choose *Open* rather than double-clicking it.
> - **Windows** — SmartScreen will warn you. Click *More info* then *Run anyway*.
>
> Code-signing is a planned follow-up; this is expected behaviour for the current release, not a sign anything is wrong.

Installers are attached to each tagged release on the project's GitHub page, and also produced automatically by every CI build if you want the latest in-development version.

Chapter 3

The Interface

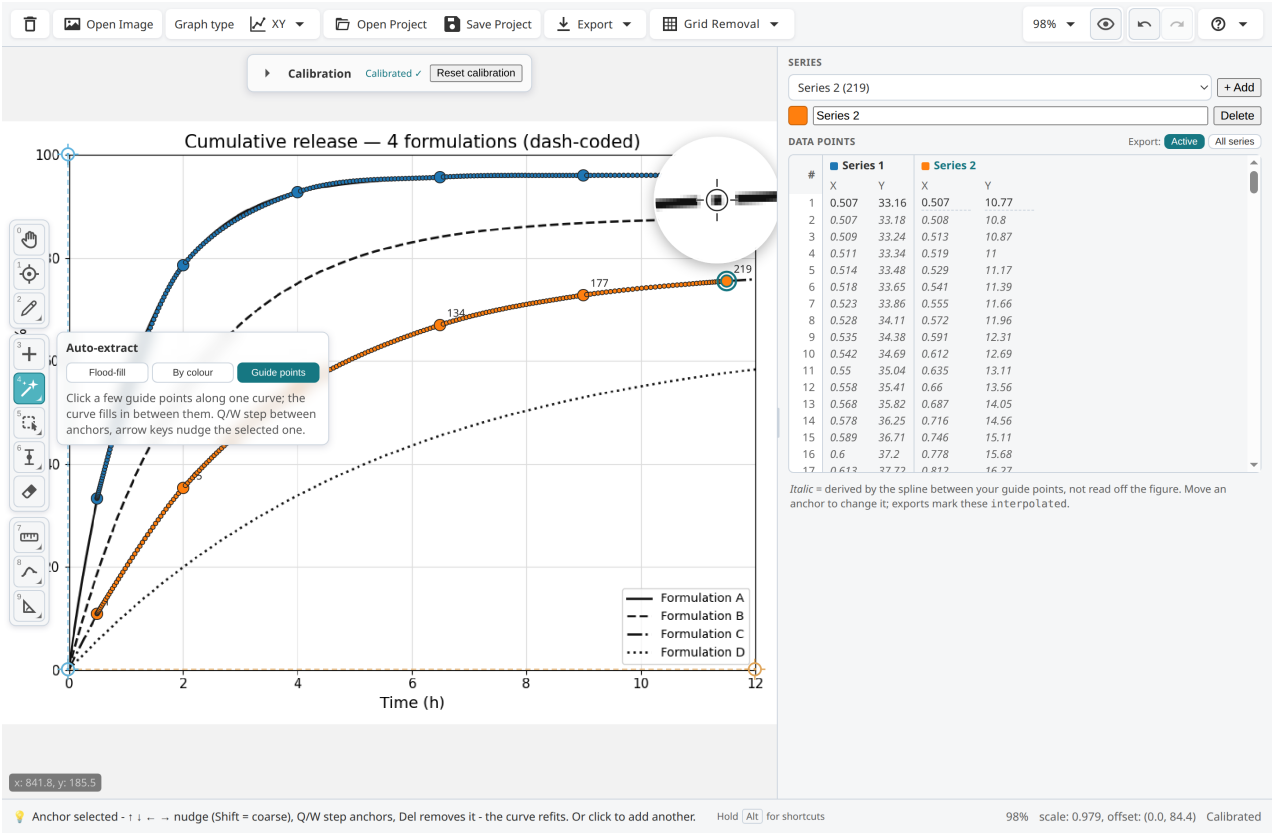


Figure 3.1: The main window: tool rail on the left, document actions along the top, the figure on the canvas, extracted data in the right panel.

Region	What lives there
Left rail	The tools, each numbered with its keyboard shortcut: Add points, Auto-extract, Select, Eraser, Measure, and the rest.
Top bar	Document actions — Open Image, Save/Open Project, Export — and the Calibration card for the current graph type.
Canvas	The figure itself. Everything else floats above it; nothing here is destructive except the image edits you explicitly commit.
Right panel	Your extracted data: one table per series, or one row per category/axis for Bar and Spider charts. This is what leaves the application when you export.

3.1 Mouse and keyboard

Input	Action
Left button	The active tool (place / calibrate / trace / measure …)
`Ctrl`+Left, or middle button	Pan
Scroll wheel	Zoom
Right-click	Quick context menu (delete point, edit value, fit to view, …)
`Enter`	Accept the current step (apply crop, finish area, run calibration)
`Esc`	Back out of the current step / clear the selection
`Del` / `Backspace`	Delete the active point or measurement
`0`–`9`	Switch tools (mirrors the rail)
Arrow keys	Nudge the selected point/handle (`Shift` = coarse)
`Q` / `W`	Step to the previous / next point
`Ctrl+Z` / `Ctrl+Shift+Z`	Undo / redo

Chapter 4

Quick Start: Your First Figure

This walkthrough uses a sample bundled with every install (`samples/xy-stress-strain.png`), so you can follow along without hunting for your own figure first. It should take about five minutes.

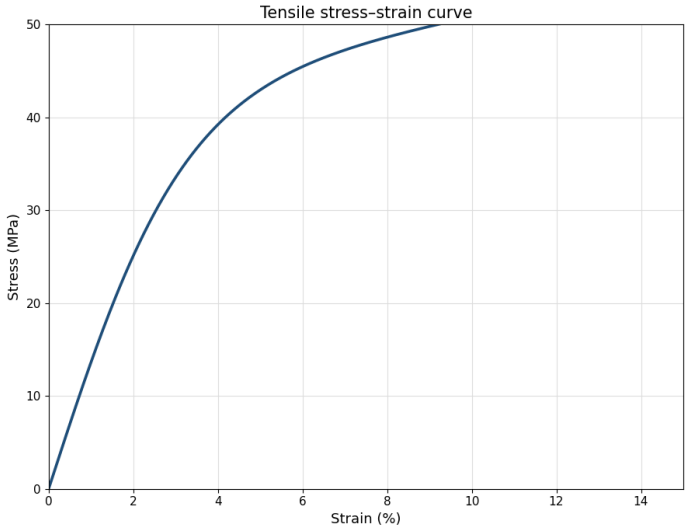


Figure 4.1: The sample figure used in this walkthrough: a single stress–strain curve, strain (%) on the X axis from 0–15, stress (MPa) on the Y axis from 0–50.

1. **Open the figure.** *Open Image* on the top bar, or drag the PNG onto the canvas.
2. **Choose the graph type.** Pick **XY** from the card picker in the top bar.
3. **Calibrate.** Open the Calibration card and place the four reference points it asks for:

- Click the tick mark at **Strain = 0**, then the tick mark at **Strain = 15** — these are your two X references.
- Click the tick mark at **Stress = 0**, then the tick mark at **Stress = 50** — your two Y references.

Press **Calibrate**. The card should read **Calibrated ✓**.

4. **Capture the figure.** Press *Capture figure* on the calibration card to freeze the framed image as your working copy.
5. **Trace the curve.** Open *Auto-extract* (tool 4) and choose **By colour**. Use the pipette to sample the curve's own colour, then run the extraction. A live preview shows exactly which

pixels it will take before you commit — check the preview follows the curve, not a legend or axis line, then accept.

6 **Sanity-check the result.** The traced curve should start near the origin and level off toward roughly **54 MPa** as strain approaches 15% — a useful check that the calibration and trace both landed correctly.

7 **Export.** *Export* on the top bar, choose a format (CSV is the simplest for a first look), and either save it or copy it straight to the clipboard.

Every sample figure under `samples/` ships with a matching `.truth.json` file recording the figure's real calibration and data — useful for checking your own extraction technique against a known answer while you're learning the tool.

That is the entire loop the rest of this guide elaborates on: **calibrate** → **capture** → **trace** → **correct** → **export.** Chapter 5 walks through each stage in depth, and Chapter 6 covers every graph type PlotTracer supports.

Chapter 5

The Core Workflow

5.1 Opening a figure

Open Image (top bar), drag an image onto the canvas, or paste from the clipboard (`Ctrl+V`). Supported formats: PNG, JPG, GIF, BMP, WebP, SVG, TIFF (including multi-page), and PDF (multi-page — you choose which page). Zoom with the scroll wheel (`Cmd/Ctrl`+scroll on a trackpad); pan with the middle mouse button or `Space`+drag; fit the view with `Ctrl+0`.

5.2 Choosing a graph type and calibrating

Pick the graph type from the card picker in the top bar — each type shows its own icon, so a bar chart and a histogram are told apart by shape rather than by reading two similar names. Chapter 6 covers calibration for every type individually; the general pattern is the same throughout the application: place the reference points the card asks for, then press **Calibrate**. The card confirms with **Calibrated** ✓.

> **Check calibration** overlays the computed axis box back onto the image, so you can confirm the mapping is right *before* you trace a single point.

5.3 Capturing the figure

Press **Capture figure** on the calibration card. This freezes the framed figure as the working image of record — what you see is what you captured — so the data always traces back to a stable source. With a PDF, the captured figure and its source page are remembered inside the project file.

5.4 Preparing the image

The image tool (rail 2) opens the **Image** card. Rotate and flip in 90 degree steps, **Straighten** a scan by up to 15 degrees with the slider (or let the app propose the angle from an axis you have already marked), and **Crop…** to a dragged rectangle. Calibration handles, data points and measurements all move with the image, so a calibrated value is unchanged by a rotate, a flip or a crop.

Mask an area… paints a dragged rectangle out with the paper colour measured around it. It is the answer to a legend drawn inside the plot box: a legend's swatches are the series' own ink at roughly the series' own size, so no colour filter and no plot-area rule can tell them from the bars they describe, and a trace picks them up as data every time. Masking removes them once, for every mechanism at

once — the colour trace, the bar detect, the scatter detect, the flood fill and the label reader all stop seeing them together. Unlike a crop the image keeps its size and nothing moves; `Ctrl+Z` takes it back.

5.5 Tracing

Several methods are available, chosen to fit the figure in front of you:

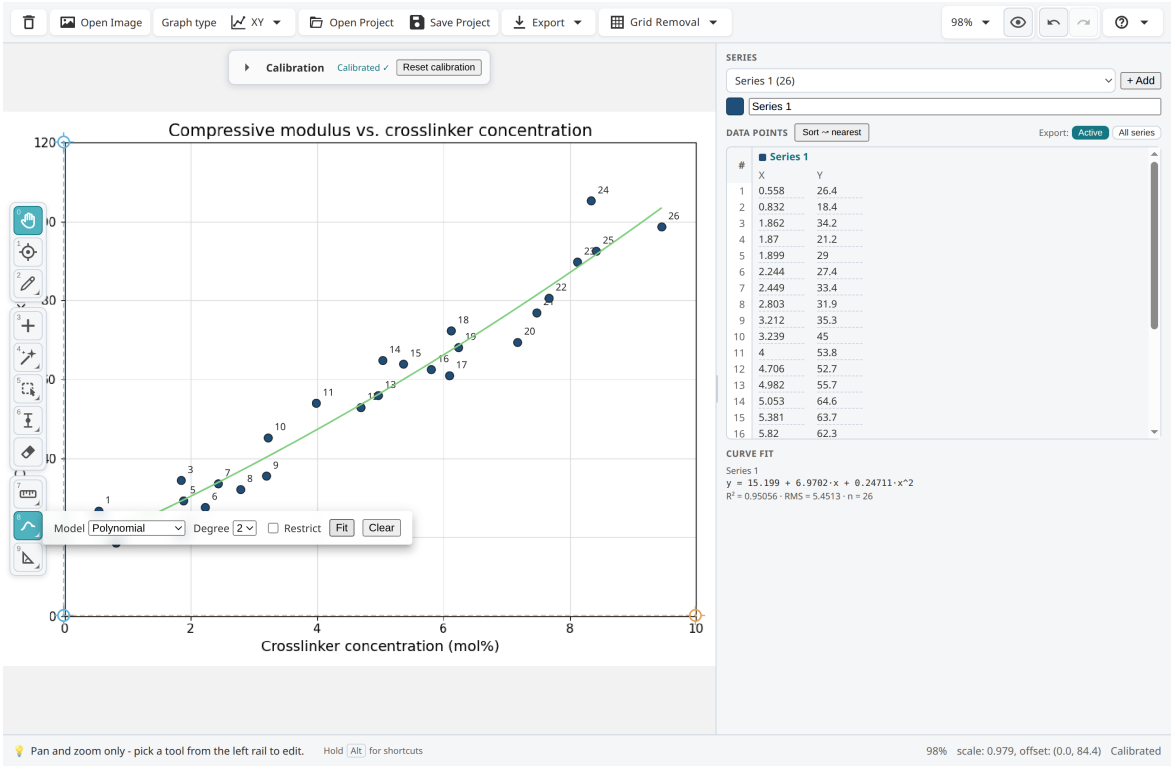
- **Add points** (tool 3) — click along the curve by hand. A zoom loupe follows the cursor for precision. This is the reliable default and works on any figure.
- **Auto-extract** (tool 4) — the wand tool, with three mechanisms:
  - *Flood-fill* — click one point on a solid curve; it traces the connected line.
  - *By colour* — pick the series' colour (type a hex value, or take it from the figure with the pipette) and it extracts every matching pixel in one pass. Good for dashed or marker curves. A live preview shows exactly which pixels will be taken; *Restrict to a box* limits the search to a region so a same-coloured legend swatch is ignored.
  - *Guide points* — for monochrome, dashed, or overlapping curves that colour tracing cannot separate: click a few guide points along one line and a centripetal spline fills the rest. The guide points are your record; the fill is marked as derived (see the **role** column, §9).
- For **scatter** data, set Auto-extract ▷ By colour's shape to *Scattered points* — one point per marker instead of one per pixel column.

> Every automatic method shows you what it captured *before* you commit, so you always know what you're accepting rather than trusting a black box.

5.6 Extracting by hand

Reach for this whenever the figure argues with you, and on anything where being right matters more than being quick. It works on every chart type and it never guesses.

1. Choose **Add points** (tool 3). A zoom loupe follows the cursor, so you are placing the point against magnified pixels rather than against your estimate of where they are.
2. Click along the curve. Each click is one reading, recorded where you put it.
3. Adjust without re-clicking. The arrow keys nudge the selected point, Shift makes the step coarse, Q and W step the selection along the points you have placed, and Del removes one. The tips bar at the bottom of the window always says which of these apply right now.
4. Drag any point to move it. The number in the table follows, because the table reads the point back through the axes rather than storing what you clicked.
5. Use **Select** (tool 5) to marquee or lasso a group, then nudge or delete them together.



#	X	Y
1	0.558	26.4
2	0.832	18.4
3	1.862	34.2
4	1.87	21.2
5	1.899	29
6	2.244	27.4
7	2.449	33.4
8	2.803	31.9
9	3.212	35.3
10	3.239	45
11	4	53.8
12	4.706	52.7
13	4.982	55.7
14	5.053	64.6
15	5.381	63.7
16	5.82	62.3

Figure 5.1: Scatter data traced by Auto-extract, with a curve fit overlaid. The fit is drawn on top of the traced points but exported as its own block — see §8.1.

A value you type into the table is treated the same way as a click: it moves the datum through the axes rather than overwriting the number, and it is marked in `[square brackets]` so a reader can see which readings came from your eye rather than from the pixels. That is not a lesser reading — there may be something in the colour, a hatch or a printed label that you can see and the sampler cannot.

5.7 Extracting automatically

Automatic extraction is a *proposal* you inspect before accepting, never a result that appears in your table unannounced.

1. Calibrate first. Nothing can be extracted into numbers until the axes exist.
2. Choose **Auto-extract** (tool 4) and pick the mechanism that fits the figure. The card offers the ones this chart type can use:
   - **Flood-fill** — for a solid, unbroken line. Click the curve and it follows the connected ink.
   - **By colour** — for dashed curves, markers, or several series drawn in different colours. Set Colour by typing a hex value or by taking it off the figure with the pipette, which is the reliable way: the colour you see on screen has been through anti-aliasing and compression.
   - **Guide points** — for monochrome or overlapping curves that colour cannot separate. Click a few points along one line and a spline fills between them. Your guide points are the record;

the fill is marked derived.

3. Set Shape to match what is drawn: Curve (line) takes one reading per pixel column, Scattered points takes one per marker. A scatter read as a curve returns a ragged line through the middle of the cloud.

4. Tune with Tolerance — how far from the chosen colour a pixel may be and still count — and with the minimum marker size, which discards specks. A legend swatch in the same colour is excluded by restricting the search to a box around the plot area.

5. **Read the preview.** It shows exactly which pixels will be taken, before any of them become data. This is the step that matters: an automatic method that looks plausible and is wrong costs more than one that visibly fails.

6. Accept it, then correct by hand. The two are meant to be used together — extract the bulk, then fix the handful the figure made difficult.

A curve fit is drawn only where it was fitted, never extrapolated across a gap, because a drawn curve is read as an answer.

5.8 Correcting

- **Select** (tool 5) — click a point to select it, or drag a box to select a range. `Del` removes the selection; arrow keys nudge selected points (`Shift` = coarse); `Esc` clears. It never selects calibration handles by accident.
- Drag any point to reposition it; drag a calibration handle to re-calibrate live.
- Edit a value directly in the right-panel table (XY and Spider/Radar). Typing a number moves the point to match it — on a spider chart it slides along that axis's own ray, so the marker and the number can never disagree.
- **Erase one reading** with the eraser (rail) or right-click ▷ Delete point.
- **Grid Removal** (top bar) pipettes a colour and wipes every pixel within a tolerance of it — useful when gridlines run through a curve you are tracing.

> Grid Removal knows nothing about charts, only about colour. On a spider chart this also strips the web while leaving the spokes, *provided the figure draws them in different shades*. Plenty of radar charts draw rings and rays in the same grey — there, Grid Removal takes both. Always check the rays survived before you trace, and undo if they didn't.

Undo/redo (`Ctrl+Z` / `Ctrl+Shift+Z`) covers everything, including image edits.

5.9 Reading labels off the figure

The one thing the app still asks you to type is a name, and a name is a transcription either way — so it can be read off the figure instead. Everything runs on your machine: the recogniser and its training

data are bundled, so reading a label needs no network and no figure ever leaves the computer.

1. Mark the category axis first. The ticks are what the words are matched against; without them there is nothing to file a name under.
2. Press Read labels from the figure on the Categories card. A banner appears at the bottom of the canvas saying the reader is armed, with a Cancel beside it.
3. Drag a box round the row of labels beneath that axis. It may touch the axis line — tick marks caught inside the box are discarded rather than read as names, and so is anything the box reaches that reads far worse than the labels beside it, such as the bottom of a bar. A generous box is fine.
4. A card lists what came back: each name beside the strip it was read from, with a confidence. **Nothing has reached your categories yet.**
5. Correct anything misread, then press Apply names. A row you clear is left alone rather than blanked, which is how you decline one reading without losing the rest.

Rotated labels are handled. The card states the angle they were read at — measured by reading the band at several angles and keeping the best — and a slider lets you say otherwise in one-degree steps, then read again. If the labels sit at an angle the sweep did not find, that is the control for it.

What arrives is always a *proposal*, and it looks like one until you accept it. A name is a transcription, and a transcription you did not check is a guess wearing a number's clothes.

5.10 Multiple series and figures

+ **Add** in the right panel starts another series on the same calibration — each named and colour-coded, captured side by side. **Extract another graph** re-enters a multi-page source (a paper's PDF, say) as a fresh figure; flip between figures with the arrows beside the calibration card. Each figure keeps its own image, calibration, series, and graph type within the one project.

Chapter 6

Chart Types, One by One

Eleven graph types over eight axis systems. This chapter covers calibration and capture for each; corrections, analysis, and export are common to all of them and covered in their own chapters.

6.1 XY (linear, log, date)

The default calibration: two X references and two Y references, each either linear, log (base-10 or natural), or a date/time scale — chosen independently per axis on the calibration card. See Chapter 4 for a full worked example.

6.2 Line (categorical X)

Behaves like XY for tracing and correcting, but the X axis carries category names rather than numeric ticks. Each point carries its own **Category** label, typed from the figure's own axis text — nothing is inferred from pixel position.

6.3 Bar and column

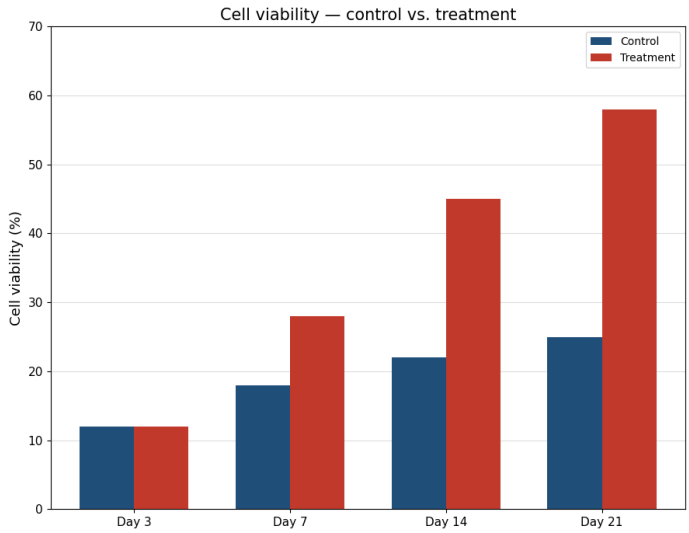


Figure 6.1: A grouped bar chart — one of the sample figures, showing the kind of source PlotTracer's bar handling is built around.

Bars are captured as a drag-box. With Add points (tool 3) on a Bar chart, press at one corner

of a bar and release at the opposite corner: a bar's value is its *extent*, not a point on it, so the two dragged corners *are* the measurement — nothing is averaged or centroided away. Plain, grouped, stacked, and floating bars all work this way, and a bar drawn below the baseline reads negative. A plain click places one corner and leaves the bar half-captured — its row shows a dash until the other corner is placed.

Auto-extract ▷ **By colour** also works on Bar charts, for the same reason: a bar blob's own bounding box *is* its two ends.

The data table is **one row per category and one column per series**, so a grouped figure shows every series at once instead of hiding all but the active one. Type each category's name once, in its row, from the figure's own tick labels; every series bound to that category updates immediately, and the export carries it in a **Category** column.

> Names are never invented. An unnamed bar reads as a **dash**, not as "Bar 1" — a placeholder in an export's Category column is indistinguishable from a name someone actually transcribed from the figure. On a grouped chart you only need to type each category's name once: a bar added to the next series takes the name of the nearest already-named bar along the category axis, so it lands in the right row however you click and whatever you skip. Where the application genuinely cannot tell which category you meant, the cell is left blank rather than guessed.

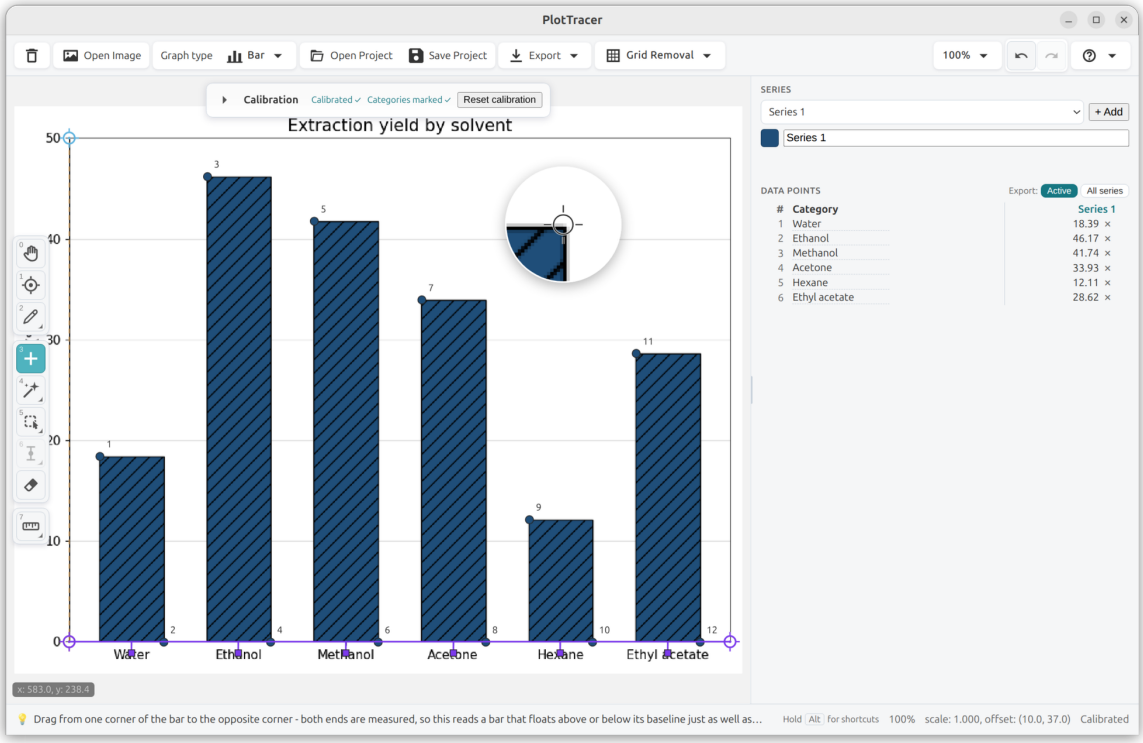


Figure 6.2: A hatched bar, which breaks a colour trace into dozens of fragments. The category axis is measured off the bars themselves, and the fragments are rejoined into the bars the figure actually draws.

6.4 Histogram

Captured the same way as a Bar chart (§6.3) — drag-box by hand, or Auto-extract ▷ By colour — with one difference in what is recorded: a histogram bar's two dragged corners are stored as genuine **bin edges and a height**, not just a centre and a value. This matters for any downstream statistics that depend on bin width, which a centre-only reading would silently discard.

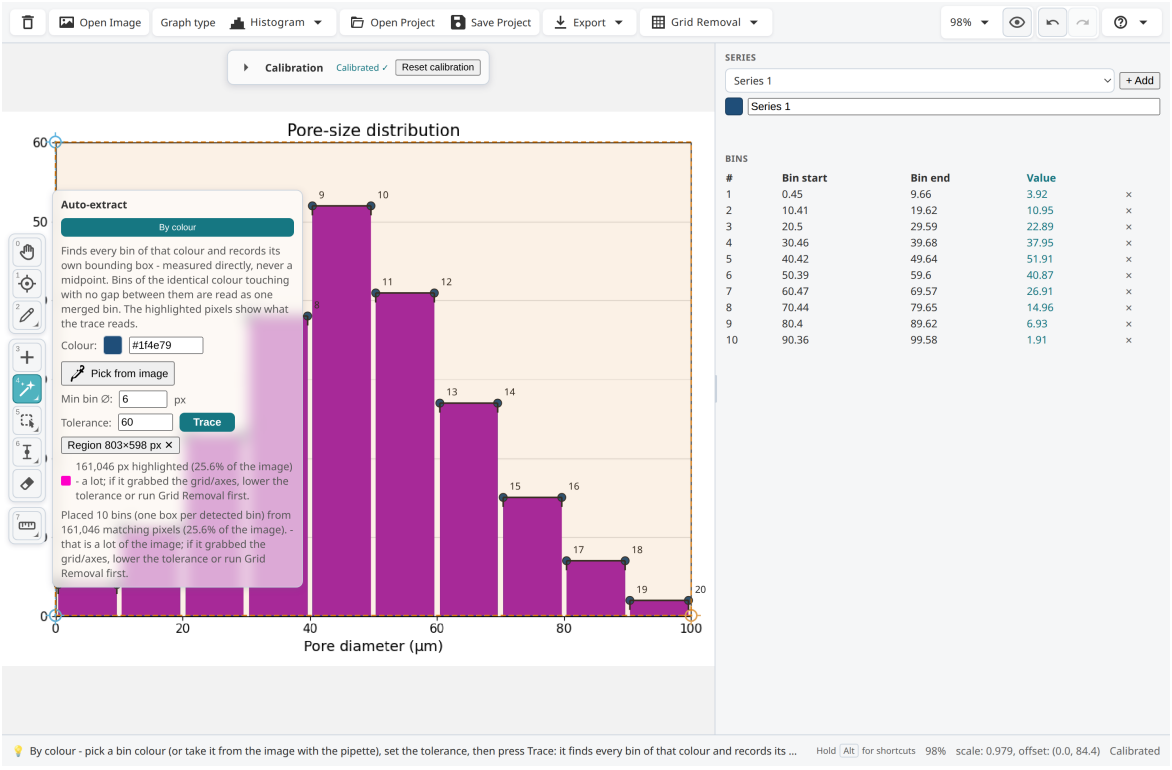


Figure 6.3: A histogram traced in the application, showing bin edges captured alongside height.

6.5 Span chart

A bar that floats clear of the axis — a temperature range, a tornado chart, a diverging figure — is a *span*, not a bar. Both of its ends are measurements, where a bar has one measurement and a reference line, so it is its own type rather than a mode of Bar. Capture it with the same drag-box; the two ends are reported as `Min` and `Max`, sorted by value rather than by which corner you pressed first.

Each end can carry its own error bar, which is what the same picture usually means in a scientific figure.

6.6 Stacked bar

A stacked bar chart is its own type because its segments are *connected*. A segment's bottom is never a number of its own: it **is** the top of the segment below it, and the lowest segment sits on the baseline. The edge the figure draws once is measured once.

That makes the reported `Value` a **height** rather than a position on the axis, which is the difference from a bar. A segment running from 2 to 5 reads 3, with its `Base` of 2 shown beside it, derived from the segment underneath. Type a number into `Value` and it means *this segment is that tall*: the segments above move up and keep their own values, because their bases moved with them. `Base` cannot be edited, because there is nothing there to change that is not the segment below's top.

Dragging a segment's top on the figure does exactly the same thing — the stack above it rides with it, keeping its own heights. The other corner of the box is there to measure the bar's width, so moving it changes no value.

Which way a column stacks comes from the figure rather than a setting. The segment nearest the baseline is the first, and each one after it is the next edge outward, so a stack drawn downward from zero, or one with segments on both sides, needs nothing switched on.

A stacked chart connects to the baseline. A column floating clear of it is a different figure, which PlotTracer does not read yet.

6.7 Candlestick

Four marks per period, clicked from the bottom of the figure upward: the low, the lower body edge, the upper body edge, the high. The prompts name *places* rather than open and close, because on a falling candle the open is the top edge — the word would point your hand at the wrong mark half the time.

Which of the two body edges is the open is the period's direction, and that is read off the figure rather than clicked: each body's colour is sampled and the candles are grouped into the two appearances the figure draws. A card at the bottom of the canvas says what was found, and one tick swaps the convention if the figure uses the opposite one. A figure drawn in a single colour has no direction to read, so every candle is reported rising.

6.8 Heatmap

A heatmap is a matrix: two coordinate axes and a colour axis, and the colour *is* the value. Both position axes carry bands rather than points, and the colour key is calibrated like any other axis — two ends of the strip, then labelled ticks along it.

The grid is offered, never asserted: dividers detected off the figure are drawn differently from ones generated by a declared count, because a grid we generated is a proposal and a grid we measured is a reading. Each cell's value can be corrected by hand, and an edit moves the cell along the colour key rather than overwriting the number — your eye may read a hatched or labelled cell better than the sampler does.

Each position axis is independently **Values** or **Categories**, and all four combinations are published — gene × sample, treatment × time, field × field. The choice is not cosmetic. A named axis has no numbers printed on it, so the band positions come from the count you declare plus the two corners you click, which is why it also asks whether the figure's marks sit *between* the bands or *under* them; answer wrongly and every boundary moves by half a band. A value axis is asked neither question: its marks are coordinates that fall where they fall, its cell boundaries are in the ink, and your two clicks are taken as the extent exactly as typed. The convention row greys out there, so you can see the question exists and see that this axis is not being asked it.

Names for the columns and rows are read off the figure with **Read column names** and **Read row names**, the same reader the bar family uses: drag a box round the row of labels, check what came back, and apply. A name can also be typed directly on the band, by double-clicking it in the Cells table. Only a named axis takes names — on a value axis the coordinate already identifies the cell, and a name would be invented rather than read.

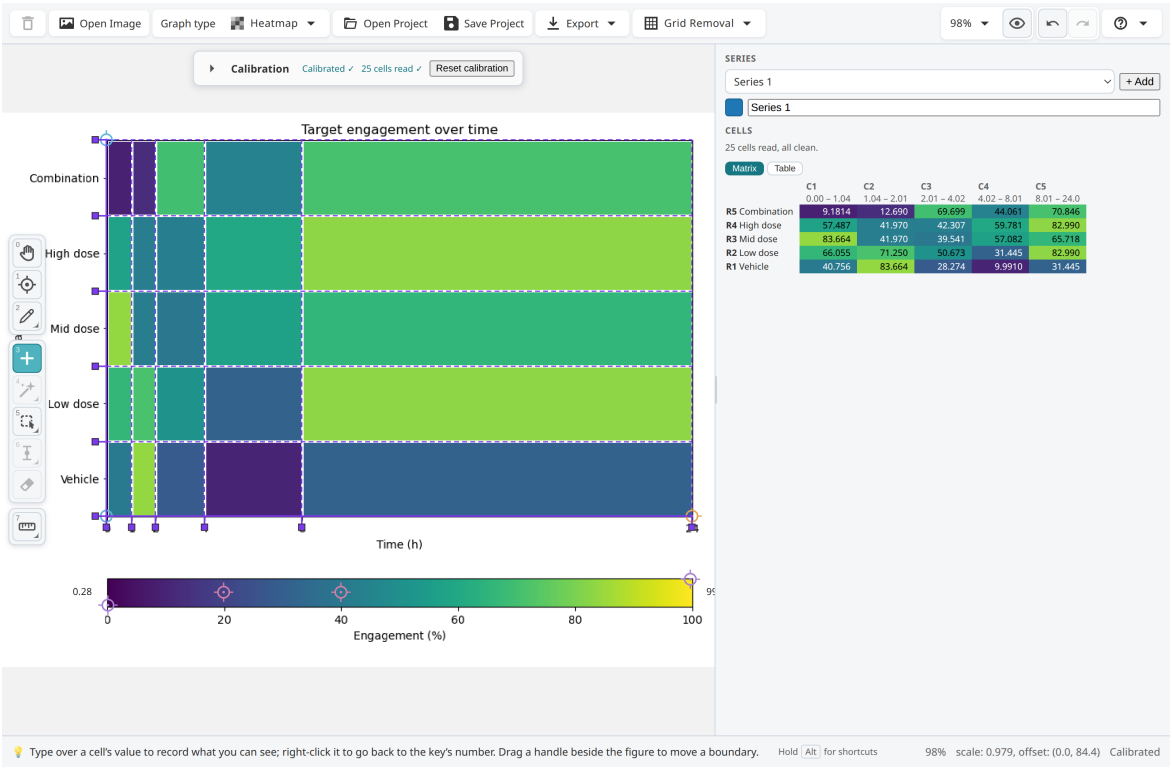


Figure 6.4: A heatmap: two banded axes and a calibrated colour key, with the matrix read cell by cell.

6.9 Box plot

Box and whisker figures are placed by hand with the loupe rather than auto-extracted: a box's five letter-values (whisker ends, quartiles, median) are not a bounding box in the way a bar or histogram bin is, so Auto-extract ▷ By colour is disabled for this type by design — returning a plausible-looking bounding box here would silently misread the figure. Quartiles and whiskers are captured per category and exported side by side, named the same way as Bar categories.

6.10 Pie and donut

Pie and donut charts are calibrated from the **outline**, not the centre. Click three or more points around the rim: three fit a circle exactly, five or more fit an ellipse and report how well the rim really is one. The centre is worked out from the fit, not clicked directly — a donut has no centre to click.

Two numbers are prefilled with the answer for an ordinary pie, and worth setting deliberately:

- **Total** (default `100`) — what the whole circle is worth. Leave it and slices read as percentages;

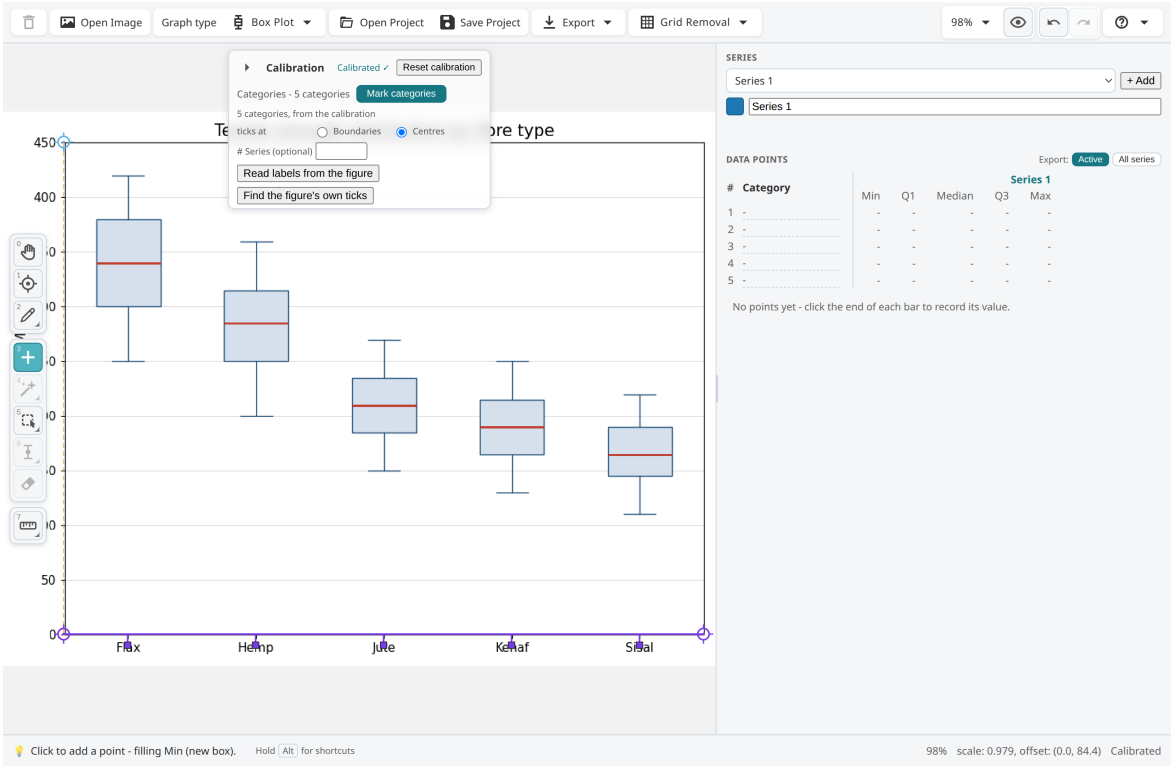


Figure 6.5: Box-and-whisker data captured per category.

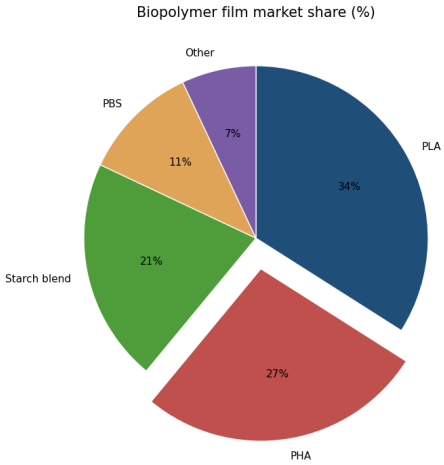
type the figure's own printed total (the number in a donut's hole, say) and they read in its units.

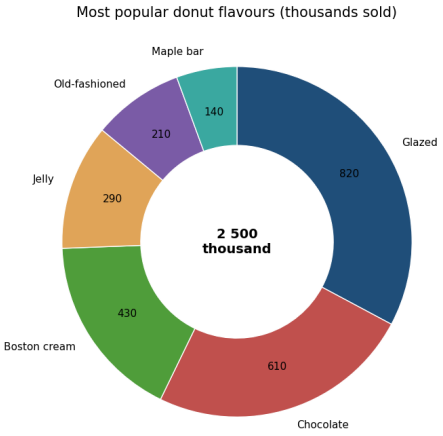
- **Sweep** (default 360) — how much of a full turn the chart actually draws. A half-pie or gauge chart has a smaller sweep, and getting this wrong silently halves every value — so it is measured, not assumed.

Tick **Tilted / 3D pie** for a chart drawn in perspective. The top face of a tilted pie is a complete ellipse — the extrusion hides none of it — so it is fully recoverable. This matters: read as if flat, a tilted 7% slice can read as 13%, and the slices *still sum to 100*, so nothing about the error looks wrong on the page.

Capture one click per boundary. Slices share their edges, so the click that closes one sector opens the next — a ten-slice pie is ten clicks, not twenty. Clicks near the rim are tidied onto it without changing the reading, because a slice's value comes from its *angle*, and moving a point along its own radius does not change that angle — which is also why a donut needs no special handling: click any ring, at any radius, and one calibration reads them all. On returning to the first boundary, the application offers to close the ring; nothing closes it automatically, because only the figure knows whether it should (a half-pie should not).

Exploded slices — a wedge pulled out from the rest — use the *Exploded slice* button at the bottom-right of the canvas. It asks for the slice's own tip, then its two edges, and measures the reading about that tip rather than the pie's centre. This is worth several points of share on a real figure, and is invisible without correcting for it, because the chart still sums to 100 either way.





6.11 Polar

> This guide's coverage of Polar-specific calibration steps (radial and angular reference points, angle convention, clockwise/counter-clockwise handling) has not yet been verified against the running application in the same detail as the other chart types. The screenshot above is confirmed accurate; a full step-by-step description should be added once checked against the current build.

6.12 Spider / radar

The number of axes on a spider chart belongs to the figure, not to the tool. Click the **centre** and give the value every axis starts from (0, unless the chart says otherwise). Then, for each spoke: click one point of known value on it and type that value and the axis's name — one click supplies both the ray's direction and its scale. Use **+ Add axis** for as many spokes as the chart draws; going clockwise keeps you in step, but nothing enforces an order.

> Each axis keeps its **own range**. A chart with tensile strength on one spoke and a cost index on the next reads correctly on both, with no rescaling between them.

Capturing steps round the axes automatically: the live one is drawn in magenta, the tip bar names it, and your click is recorded where it crosses that ray — points sit *on* the axis, so what you see is the number recorded. The data table reads one row per axis and one column per series.

Auto-extract ▷ By colour is the only automatic mechanism offered on a spider chart, and it does a different job than elsewhere: it walks each calibrated ray outward and records the value where the series' colour crosses it. Where a ray crosses the colour more than once — a grid ring in a similar ink, a second series, a filled polygon's far edge — it records **nothing** on that axis and names which one in the status message, because the crossing it should read is exactly what is in doubt. Those axes stay empty and the capture cursor lands on them next, so the refusals become your worklist rather than a plausible wrong number. A reading placed by hand is never overwritten by a later automatic pass.

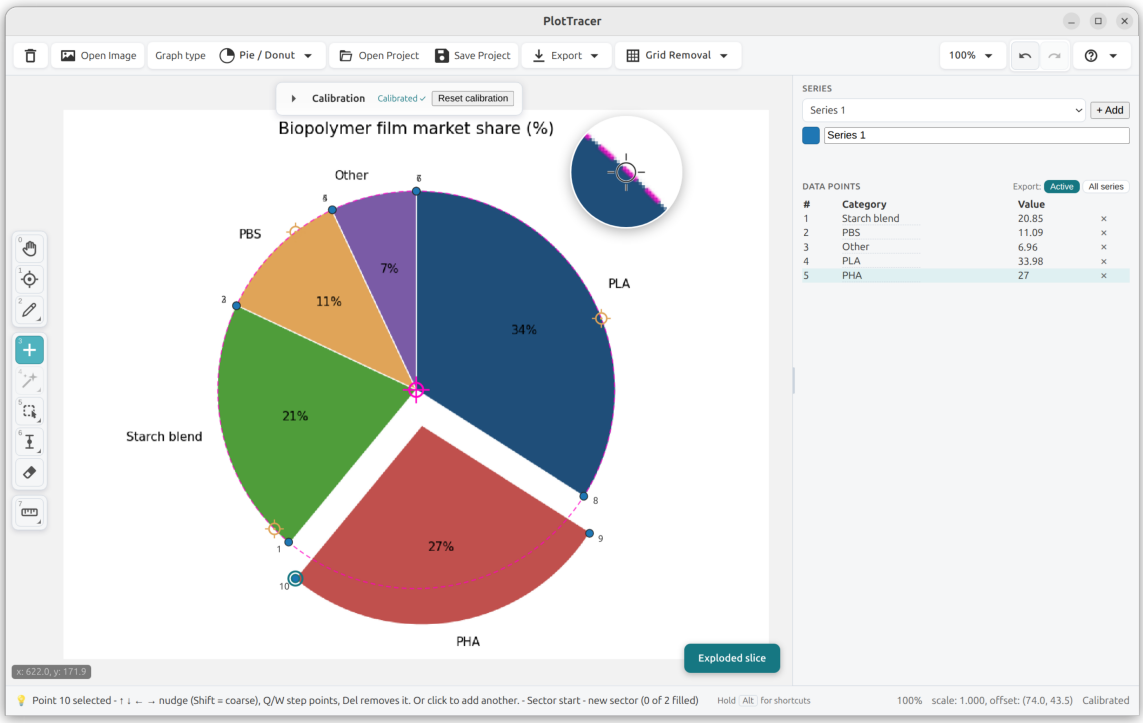


Figure 6.6: The same kind of figure in the app: the centre fitted from the outline rather than assumed, an exploded slice measured about its own apex, and the sectors reported as they were read rather than tidied to a round 100%.

Figure 6.7: Sample pie and donut source figures — an exploded wedge (left) and a donut (right).

Every cell in the data table is live: click a value to type it, click a point's cell to select that point on the canvas, click an empty cell (`--`) to aim the next capture at that axis, or click an axis name to type or edit it. An axis name is optional — an illegible label is still a real axis, and an unnamed one exports positionally as `Axis 3`.

6.13 Ternary

Calibrate a ternary diagram by clicking its three corners, going round the triangle, and naming each one as the figure does — `Sand`, `Silt`, `Clay`. The names are optional and become the export columns; without them the components read `A`, `B` and `C`.

The order does not matter. Slot 1 is whatever corner you clicked first, and every click order reads correctly, because the reading is the clicked triangle's own barycentric coordinate and the handedness cancels. There is nothing to configure: what tells the three components apart is the name you gave each corner, not a setting.

A point is read as its position within that triangle, so the three components sum to the range you chose (0–100 or 0–1) by construction rather than by arithmetic that happens to agree. All three are stored, not two plus a derived third.

Three corners on one line have no area, and every reading would divide by it, so that calibration is

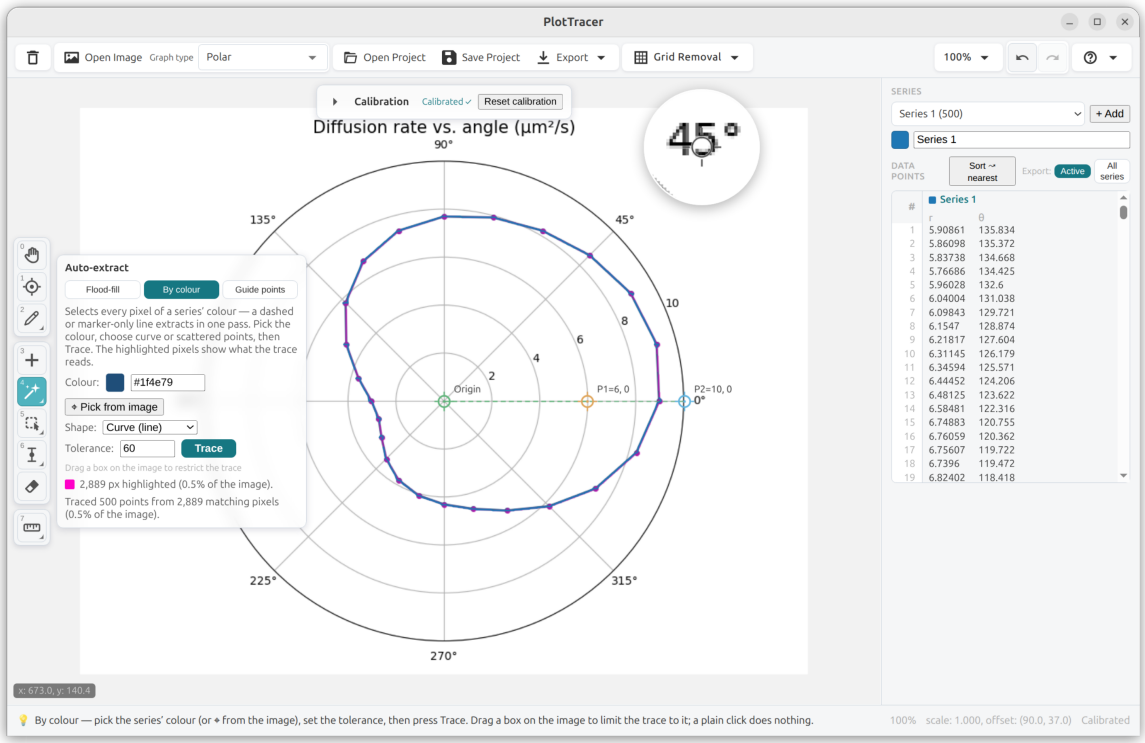


Figure 6.8: A polar plot captured in the application.

refused at the click that makes it rather than eight steps later.

6.14 Map

Map calibration is scale-bar based (per the project README), but the exact workflow — how a scale bar is placed, whether coordinate reference systems or simple relative distances are supported — has not yet been verified in enough detail for a reliable step-by-step here. Confirm against the running application before publishing this section.

6.15 Circular chart recorder

This chart type reads traces from analogue circular recorders (a spiral or concentric-ring time axis with a radial value axis). The calibration workflow has not yet been verified against the running application; write this section once confirmed.

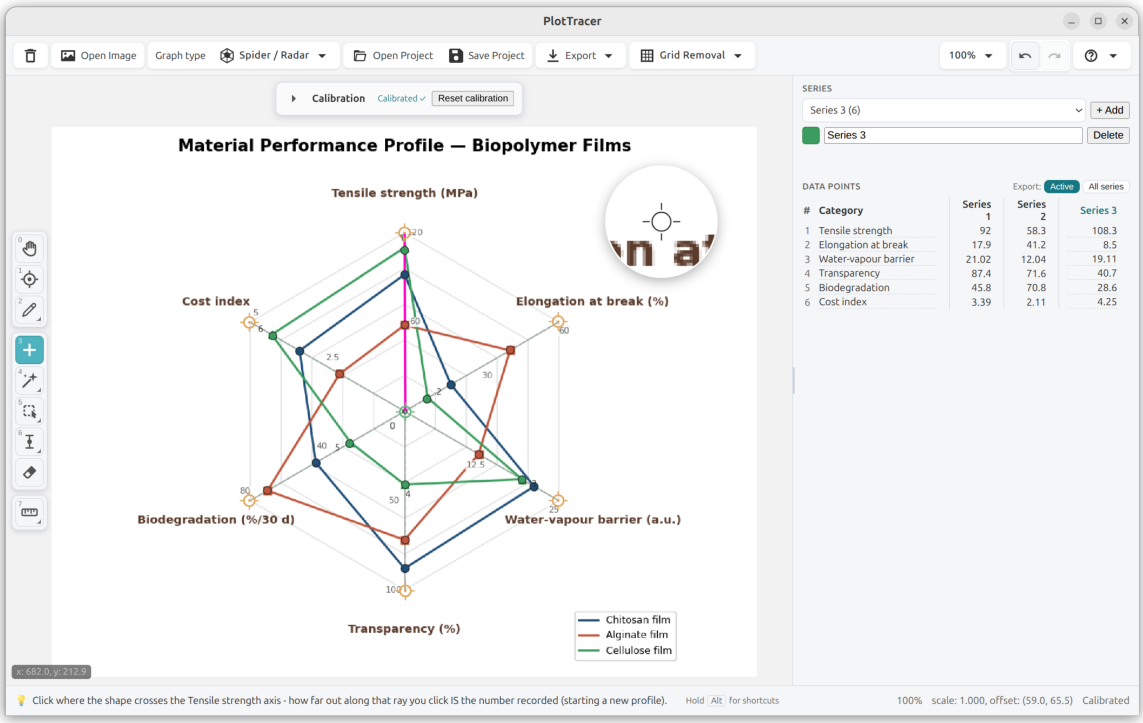


Figure 6.9: A six-axis spider chart, each spoke calibrated on its own scale.

6.16 Error bars

Error bars are **not** a graph type of their own — they are a rail tool (hotkey `6`), captured on top of whichever series they belong to, on any chart type that carries them. Each cap is placed where the figure actually draws it, asymmetric caps included, rather than assumed to mirror around the point.

> The exact click sequence for placing a cap (single click per cap vs. drag, how the application distinguishes a "mirrored" cap the user places once from two independently measured caps) is described in outline on the marketing site but not yet written up step-by-step here. Confirm the precise interaction against the running application before publishing this section in full.

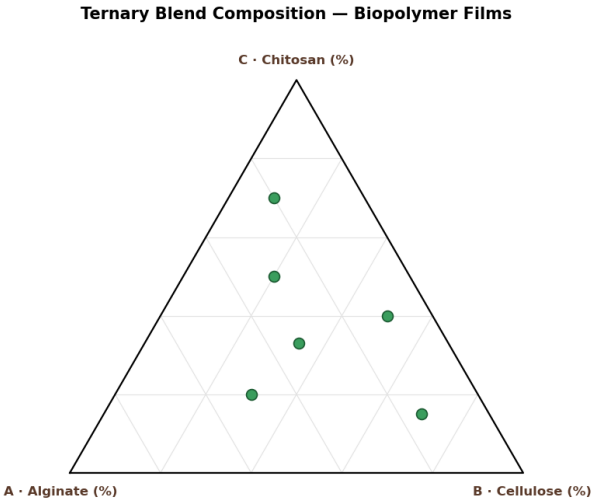


Figure 6.10: A ternary composition diagram — one of the bundled sample figures.

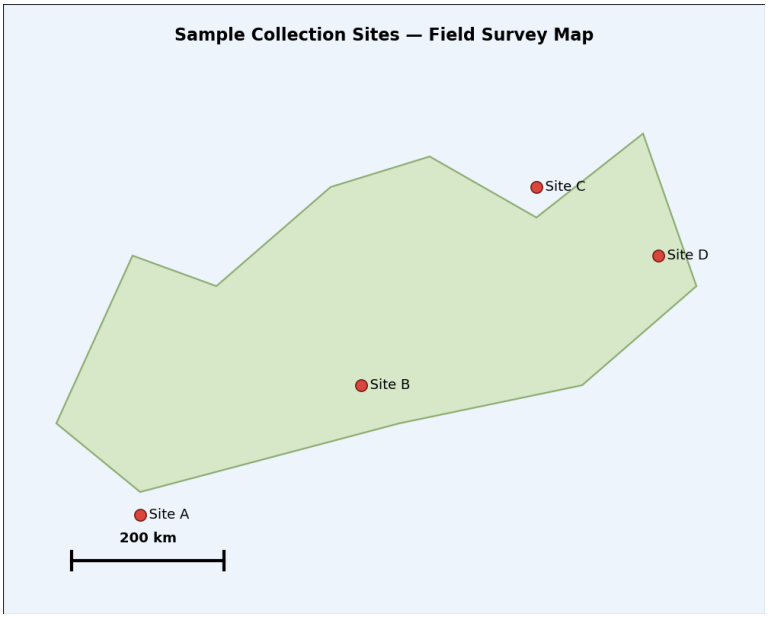


Figure 6.11: A sample map figure with a scale bar, for calibrating spatial coordinates.

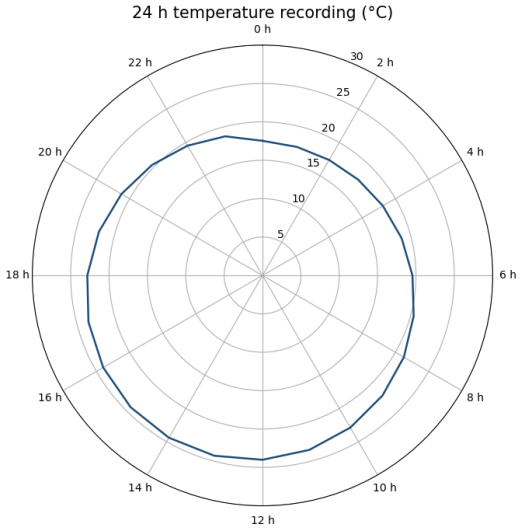


Figure 6.12: A circular chart recorder trace — the kind of figure produced by older analogue instrumentation (temperature, pressure, humidity loggers).

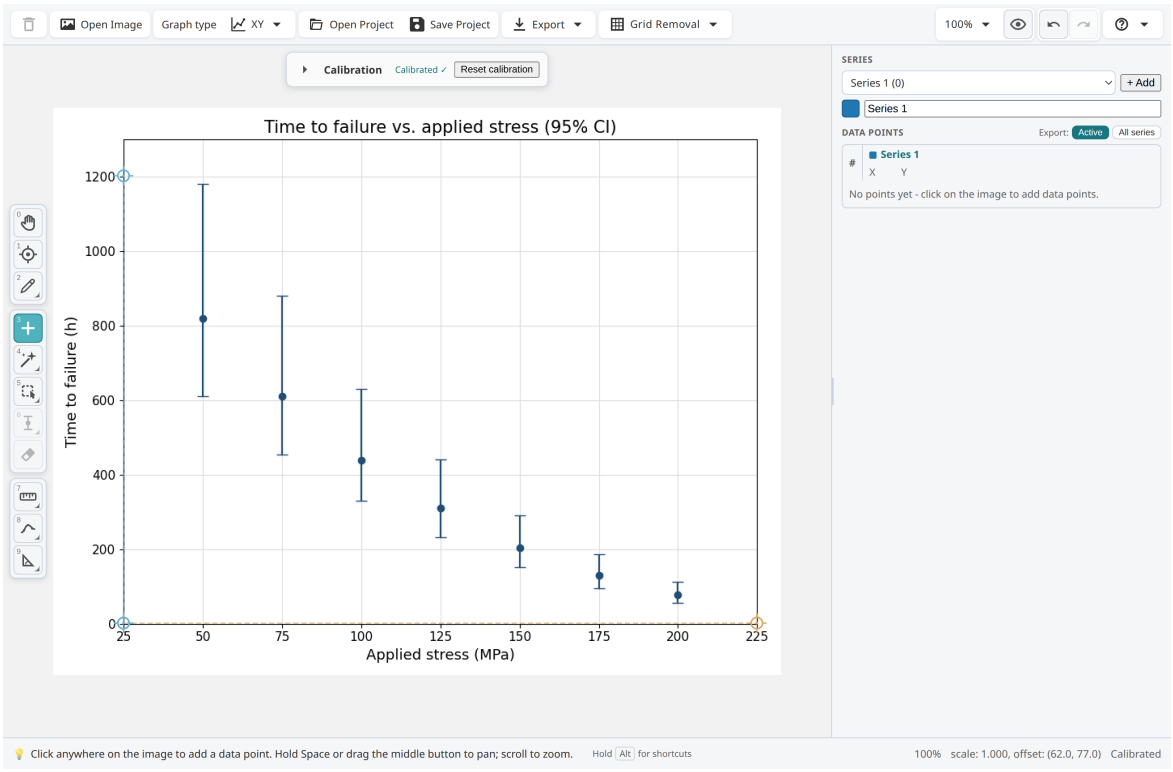


Figure 6.13: Error bars captured on top of a data series, asymmetric caps included.

Chapter 7

Measuring

The ruler tool (hotkey 7) opens a Measure card with four modes: **distance**, **angle**, **area**, and **slope** — read either in the chart's own calibrated units, or in a scale you set from any known length on the image. Measurements are kept as a separate collection from your traced series data, so they never mix into an export of the figure's actual points.

Chapter 8

Analysis

8.1 Curve fitting

Open **Curve fit** from the tool rail to fit a model through a traced series.

Model	Form
Polynomial	$y = a_0 + a_1 x + a_2 x^2 + \dots$ (degree is chosen)
Exponential	$y = a \cdot e^{bx}$
Power	$y = a \cdot x^b$
Logarithmic	$y = a + b \ln x$
Gaussian	$y = a \cdot e^{-(x-b)^2/2c^2}$
Logistic	$y = a/(1 + e^{-b(x-c)})$

The card reports the equation, its coefficients, R^2, and RMS, and draws the fitted curve over the figure. **Degree** applies only to the polynomial and disappears for the other models. Some models cannot take all data — a logarithmic fit needs every x above zero, a power law the same, an exponential every y above zero — and PlotTracer states which requirement is unmet rather than returning a number it cannot stand behind. Use **Restrict** to fit over a chosen x range.

A fit that did not settle says so. The nonlinear models are solved iteratively (Levenberg–Marquardt), and a solver that runs out of iterations still returns *something*. When that happens, the card reports the curve as where the solver stopped rather than as a result, and the exported `Curve fit` block carries `settled = no` in its own column, so the caveat survives the hand-off to whoever opens the file later. A polynomial is solved directly and has nothing to converge, so it reports `n/a`.

The fit is always exported as a **separate block** from the traced record — the points you placed are never overwritten by the model drawn through them.

8.2 Geometry and statistics

Arc length, enclosed area, and curvature are available alongside curve fitting, computed directly from a traced series or a fitted model.

8.3 Check calibration

An overlay that draws the calibrated axis box back onto the figure, so you can confirm the mapping is correct at any point — not only right after calibrating.

Chapter 9

Exporting Your Data

9.1 Formats

Export on the top bar offers CSV, TSV, JSON, OpenDocument (`.ods`), Excel (`.xlsx`), LaTeX, MATLAB, Python, R, or a PNG of the annotated figure. Any text format can also be copied straight to the clipboard. Choose **Active** (the current series) or **All series**.

Values are rounded to the figure's real resolution — never padded with false precision, never collapsed to zero — with a full-precision option available when every digit is wanted. Fitted curves, geometry, and measurements always export as their own blocks, kept separate from the recorded points.

> The format menu states up front what each format *will not* carry, before you pick one. No data format carries the figure image, the axis calibration, or the source document — save a **project** to keep those. Individual formats name their own limits: MATLAB becomes a cell array once any cell holds text; the flat text formats put every block into one stream.

9.2 The role column

If a series was traced with **Guide points**, its export gains a **role** column: `anchor` for a point placed by eye, `interpolated` for one the spline filled in between anchors, and **blank** for an ordinary point clicked with Add points. Keep only the anchor rows if you want strictly what a human placed on the figure. The column appears only on series traced this way — an ordinary trace exports exactly as before.

9.3 Project files

Save Project writes a self-contained `.zip` holding every image, calibration, series, measurement, and the original source PDF, so the whole extraction reopens exactly and any number traces back to its figure.

9.4 Reading other tools' projects

Open Project also reads projects written by other digitisers — currently WebPlotDigitizer `.tar` archives, Engauge Digitizer `.dig` files, and StarryDigitizer `.zip` projects. There is no separate command for any of them: the format is recognised from the file's own bytes, never its extension, so a project opens the same way whatever wrote it, and a file no filter recognises is refused with the list of formats that do work.

This is a **one-way** road. PlotTracer reads those formats but does not write them — a third of what it records (spider spokes and their per-axis scales, point roles, box-plot tuples, measurement blocks) has nowhere to go in any of them, so offering an export back would promise a round trip it could not honour.

Chapter 10

Troubleshooting

A spider chart's spokes disappeared after Grid Removal.
Grid Removal strips every pixel within tolerance of the sampled colour, and knows nothing about the chart type. If the figure draws rings and spokes in the same shade of grey, both are removed. Undo (`Ctrl+Z`) and either raise the tolerance selectively or trace the spokes before removing the grid.

Some spider axes are left blank after Auto-extract.
This is by design — where a ray crosses the traced colour more than once, the application cannot tell which crossing is the real reading, so it records nothing and names the axis in the status message rather than guess. Place that one reading by hand with the loupe.

A curve fit looks wrong / the card says the fit did not settle.
Nonlinear models (exponential, power, logarithmic, Gaussian, logistic) are solved iteratively and can run out of iterations on noisy or poorly-conditioned data. Try restricting the fit to a narrower x range, or confirm the traced points themselves are clean before refitting.

macOS says the app is damaged / from an unidentified developer.
The current builds are unsigned. Right-click the app and choose *Open* rather than double-clicking; this only needs doing once per install.

Windows SmartScreen blocked the installer.
Click *More info*, then *Run anyway*. Same cause as above — an unsigned binary, not a corrupted download.

An exported bar/category is blank instead of named.
This is deliberate: PlotTracer never invents a name like "Bar 1" for an unlabelled bar, because a placeholder in an export is indistinguishable from a name someone actually transcribed from the figure. Type the category name yourself from the figure's tick labels.

A project written by another digitiser won't open.
Only WebPlotDigitizer, Engauge Digitizer, and StarryDigitizer project files are currently recognised (§9.4), and recognition is based on the file's own bytes rather than its extension. If the file genuinely comes from one of those tools but still isn't recognised, it may be a version or variant the reader doesn't yet cover.

Chapter 11

About This Project

11.1 License

PlotTracer is distributed under the **GNU Affero General Public License v3.0**, inherited from WebPlotDigitizer upstream and preserved in all distributions.

11.2 Attribution

PlotTracer's calibration engine began as a TypeScript port of **WebPlotDigitizer** by Ankit Rohatgi (AGPL-3.0); the application itself — interface, interaction model, workflow — is a ground-up rebuild. Several algorithms (flood-fill curve tracing, grid-line removal, curve fitting, geometry/statistics) are clean-room reimplementations of ideas from **Engauge Digitizer** (Mark Mitchell, Jason Nicholson; GPL-2.0), written from the algorithm descriptions rather than translated from source, to remain license-compatible. PlotTracer also reads **StarryDigitizer** projects (MATO Tomoya; MIT) — the file format only.

11.3 No telemetry

PlotTracer makes no network calls of its own. Your figures, your data, and your project files never leave your machine unless you choose to share them.

11.4 Getting help

Issues and pull requests are welcome on the project repository: `https://github.com/katalystnord/plottracer`.