NOVASYNE

Wearable Workbench

User Guide

Import a session from any of 23 devices, put every signal on one clock, and measure it — with each refusal explained rather than guessed around.

Version 1.0 August 2026

Novasyne · Signal analysis for clinical and translational research

Contents

Before you start

The Wearable Workbench takes a raw export from a consumer or research wearable and turns it into something you can measure. It reads 23 device formats, lands all of them in one internal representation, puts every signal on a single clock, and then lets you label, preprocess, quantify, visualise, interpret and export the session.

The point of the single representation is that the rest of the workbench does not care which device you used. An Empatica E4 and a Fitbit export arrive looking nothing alike and leave looking the same, so a Poincaré plot or an HRV calculation means the same thing whichever produced it.

This guide is task-oriented. Each chapter answers a question you might arrive with — *how do I mark my experimental conditions?*, *how do I get HRV?*, *why will it not draw me an actogram?* Appendix A is the control reference, and Appendix B lists every supported device.

What you need

The application	Running and reachable in a browser. See the README shipped with the package for installation.
Your data	An export from one of the 23 supported devices, or a `.uwf` file the workbench wrote earlier.
An API key	Only for the interpreted report. Everything else runs offline.

> **SET SECRET_KEY BEFORE YOU START USING IT IN EARNEST**
>
> Left unset, the application generates a new signing key every time it starts, so restarting it invalidates the page already open in your browser. The next thing you click reports that the security token is no longer valid and asks you to reload — nothing is lost, but it is avoidable. Set `SECRET_KEY` in `.env` once and restarts stop mattering. The README says how.

The two recordings in this guide

One session cannot demonstrate everything, because two of the charts need days of data and one needs a kind of signal a wrist sensor does not produce.

Recording	Shape	Used for
Empatica E4 session	6 signals, 1 h 35 m	The spine of the guide: import, labels, HRV, EDA, preprocessing, the report, export
PMData p03 (Fitbit)	27 signals, 152 days	The actogram and the daily state composition

WHY THE GUIDE SHOWS TWO REFUSALS BEFORE IT SHOWS THE CHARTS

On the E4 session the actogram and the composition chart both decline, for good reasons that are worth seeing. Chapter 7 shows them declining and Chapter 8 shows the same two charts working on the Fitbit recording. A tool that draws a 24-hour rhythm from 95 minutes would be worse than one that says no.

The workspace

Before a session is imported

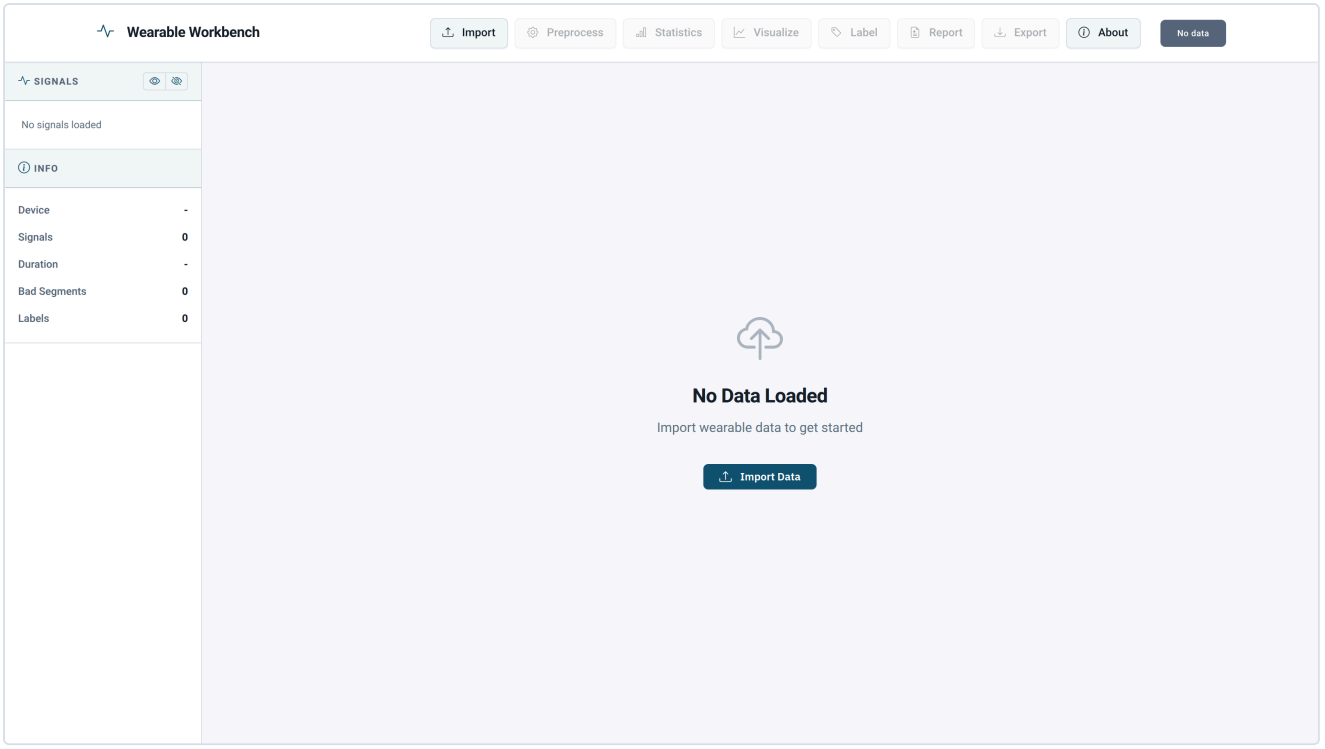

Figure 1 — The workbench on first open.

Everything waits for data. Import and About are the only controls enabled; Preprocess, Statistics, Visualize, Label, Report and Export switch on when a session arrives.

Figure 2 — The toolbar.

The badge at the right end tells you what is loaded — `No data` now, then the signal count once a session is in.

Once a session is loaded

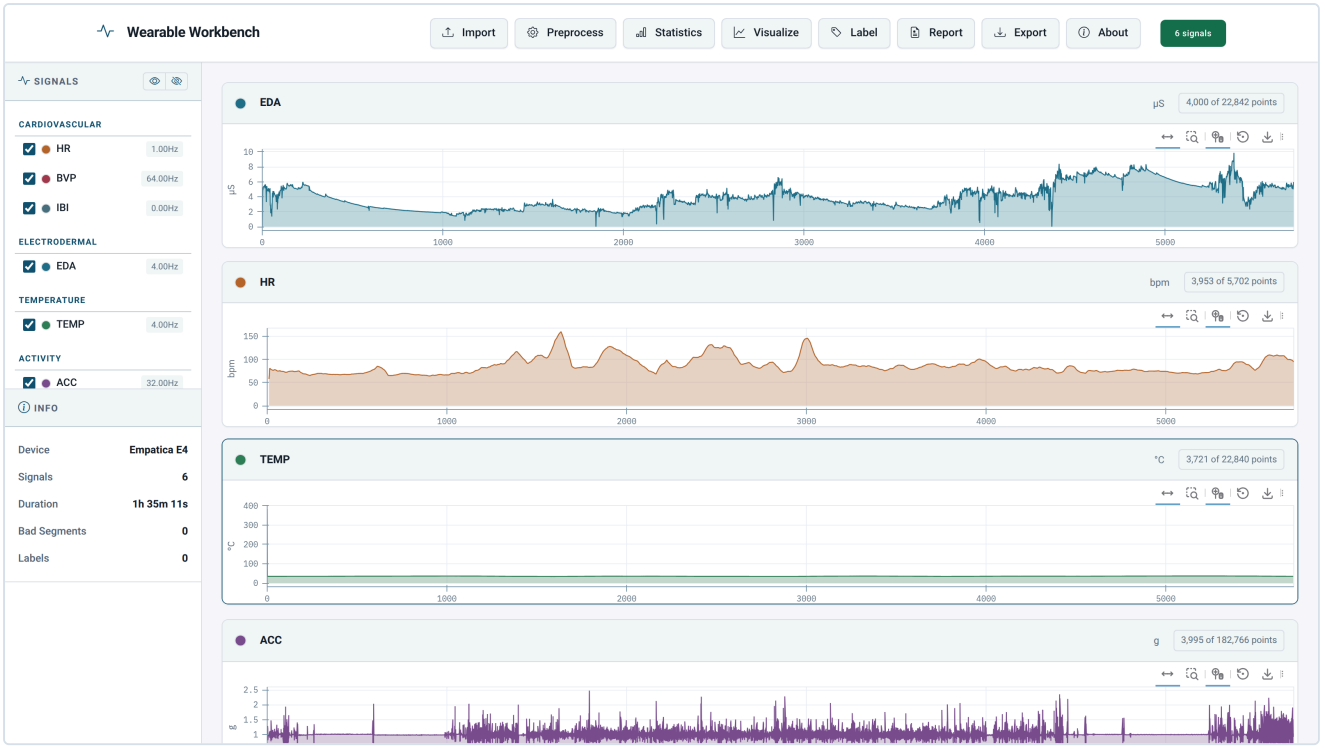

Figure 3 — An Empatica E4 session: six signals, one clock.

Three regions:

The signal list, on the left, groups every signal by domain — cardiovascular, electrodermal, activity, temperature. The checkbox beside each one shows and hides its chart.

The info panel, below it, is the session's summary.

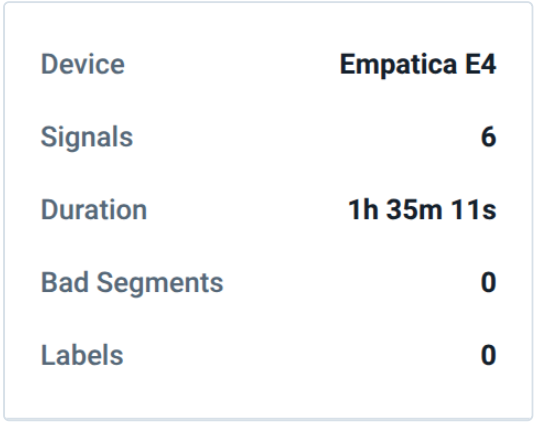

Figure 4 — What the import found.

The charts, filling the rest, are one per visible signal, all sharing one time axis. Zooming one zooms them all, which is the point: an EDA response and the motion that might explain it should be readable against each other.

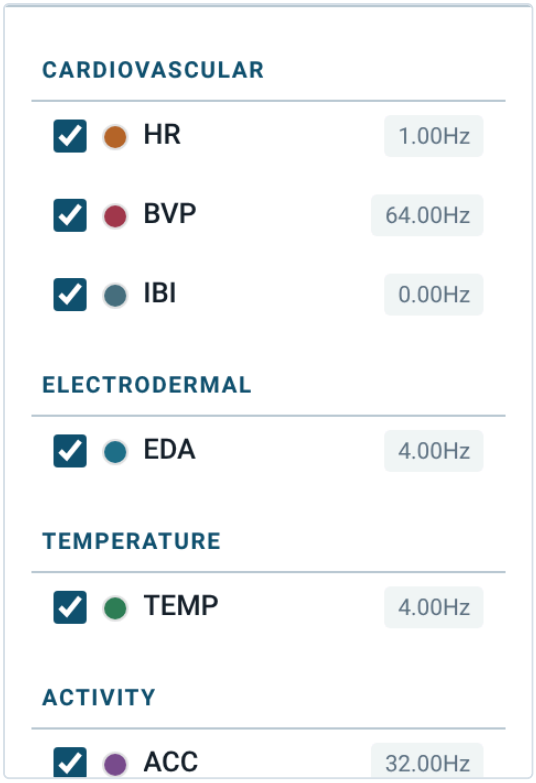

Figure 5 — Every signal, with its sampling rate.

A rate of `0.00Hz` is not an error. It means the signal is irregular — an event series like inter-beat intervals, or a daily value like a sleep score — and carries a timestamp per sample rather than a fixed cadence.

Importing a session

The Import dialog

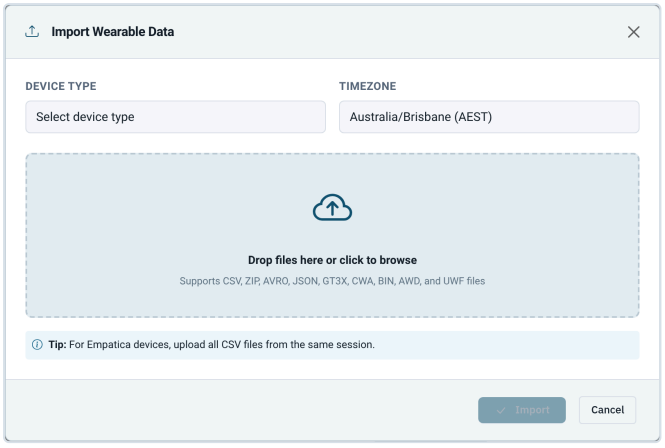

Figure 6 — The Import dialog.

Three things to set, in this order:

Device type. The workbench does not sniff the format. Tell it what produced the file and it applies that device's reader, including the quirks that reader knows about. Appendix B lists all 23.

Timezone. Most devices record local time without saying which zone that was. Whatever you choose here is the zone the session is read in, and it is stated on every chart axis. Get it wrong and every hour-of-day analysis shifts.

The files. Some devices export a single archive; others export a folder. Both are supported — select the archive, or select all the files in the folder at once.

> **THE TIMEZONE IS A CLAIM ABOUT THE DATA, NOT A DISPLAY PREFERENCE**
>
> Apple Health is the only supported device that states its own UTC offset. For everything else the workbench has nothing to go on but what you tell it. Changing it later means re-importing.

A folder-based export

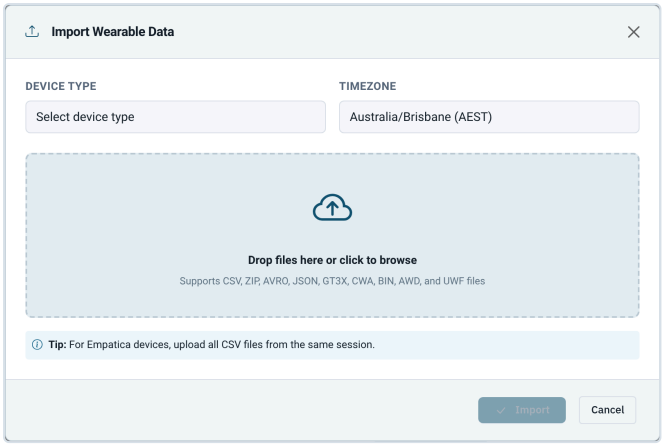

Figure 7 — A Fitbit export is a folder of files, not one file.

Fitbit, movisens and EmbracePlus export directories. Select every file in the folder — the reader works out which is which from their names and contents.

What import reports

Import does not simply succeed. It reports what it found and what it could not use, and those warnings travel with the session: they appear in the info panel, and they are included in the interpreted report later.

> **READ THE WARNINGS, ESPECIALLY THE SKIPPED ONES**
>
> A converter that silently drops a channel costs you that measurement without telling you. Anything the reader could not parse is named, with the reason.

Reading the session

Marking the experimental structure

A recording is rarely uniform. Labels mark what was happening when.

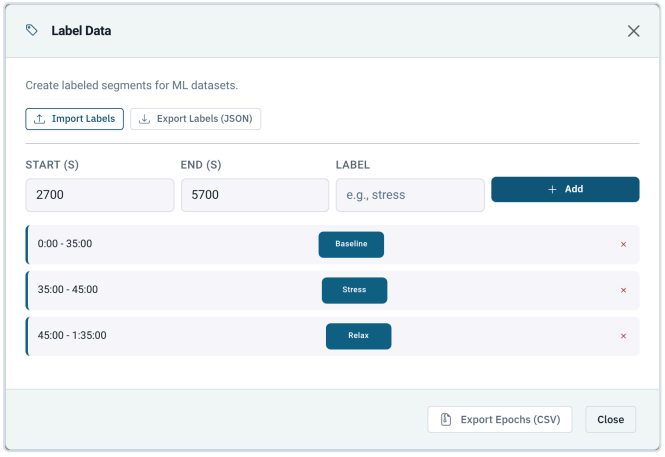

Figure 8 — Three labels over one session.

Give each a start, an end (both in seconds from the start of the recording) and a name. You can also load labels from a file if your protocol produced one.

Figure 9 — The labels overlay every chart at once.

They are not decoration. They overlay every signal on the same clock, so a change at the start of the *Stress* block is visible in EDA, HR and motion simultaneously, and they travel with the file on export.

Measuring the session

The Statistics dialog has three tabs: general descriptives, then two that go deeper into the signals that support it.

Basic statistics

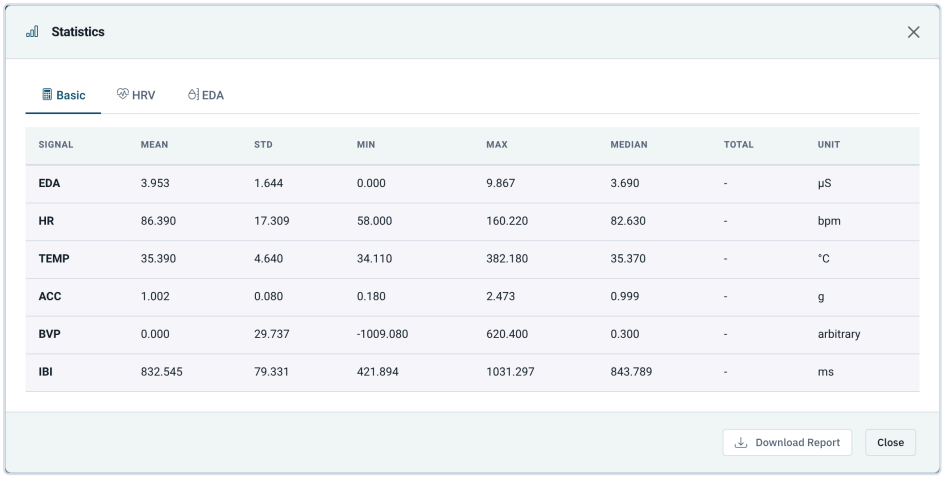

Figure 10 — Summary statistics for every signal.

Mean, median, standard deviation, range and sample count for every signal including the derived ones. Worth scanning before anything else — a maximum that cannot physically occur is the fastest way to spot a corrupt sample.

Heart rate variability

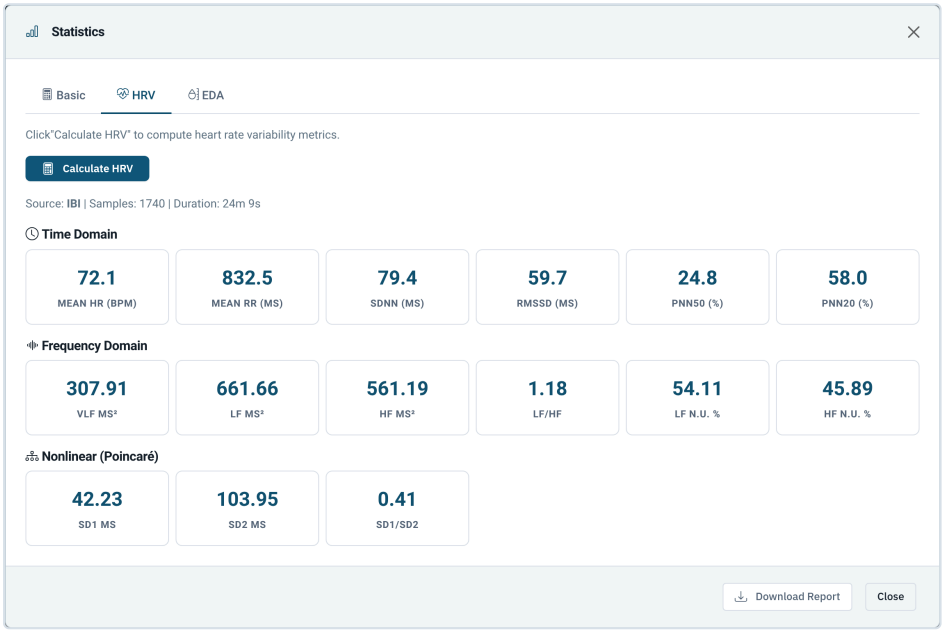

Figure 11 — HRV: time domain, frequency domain, nonlinear.

HRV is computed from beat intervals, and the source is declared above the results — which signal, how many samples, over what duration. That duration is often shorter than the session: an E4 emits an interval only where its beat detection was confident, so 95 minutes of recording produced 24 minutes of usable intervals here.

Time domain — mean HR and RR, SDNN, RMSSD, pNN50, pNN20.

Frequency domain — VLF, LF and HF power, the LF/HF ratio, and the two normalised units.

Nonlinear — SD1, SD2 and their ratio, the Poincaré descriptors.

> **SHORT RECORDINGS DO NOT SUPPORT EVERY METRIC EQUALLY**
>
> RMSSD and SD1 stabilise over a few minutes. The frequency-domain figures, and VLF in particular, need much longer before they mean anything. The duration is printed beside the source for exactly this reason.

Electrodermal activity

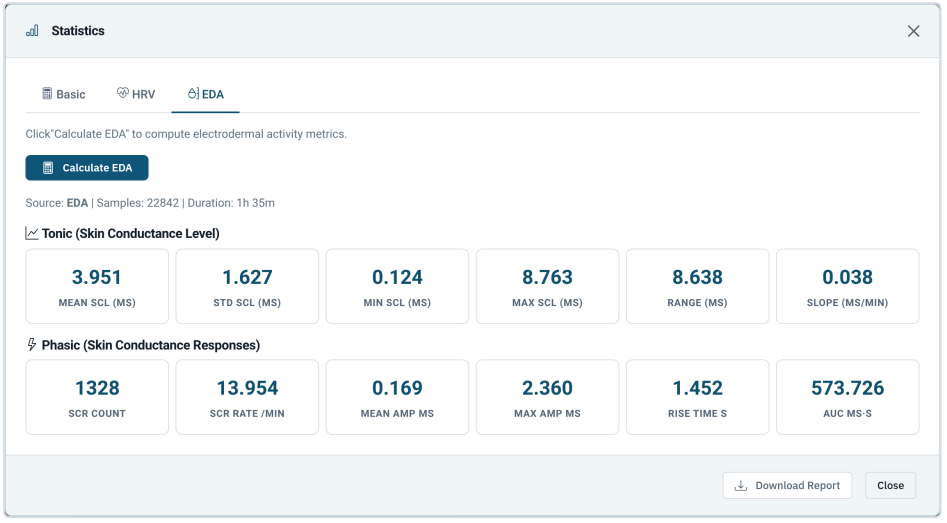

Figure 12 — EDA split into tonic level and phasic responses.

EDA is decomposed into a slowly-drifting **tonic** level (skin conductance level) and the fast **phasic** responses riding on it (skin conductance responses). The tonic level speaks to arousal over minutes; the phasic peaks are the responses to discrete events.

Visualising the session

The Visualize dialog holds ten chart types. Pick a type, pick a signal where one is needed, and generate.

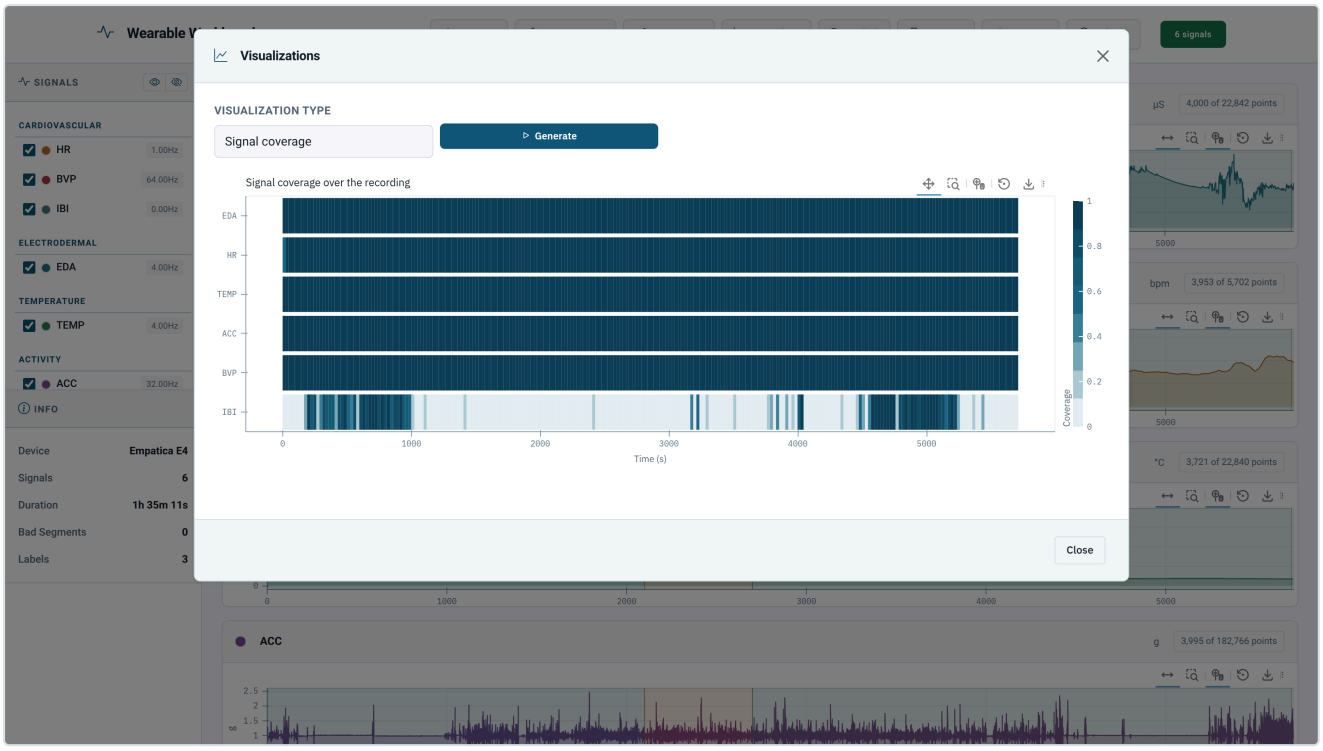

Figure 13 — The Visualize dialog over the loaded session.

Coverage

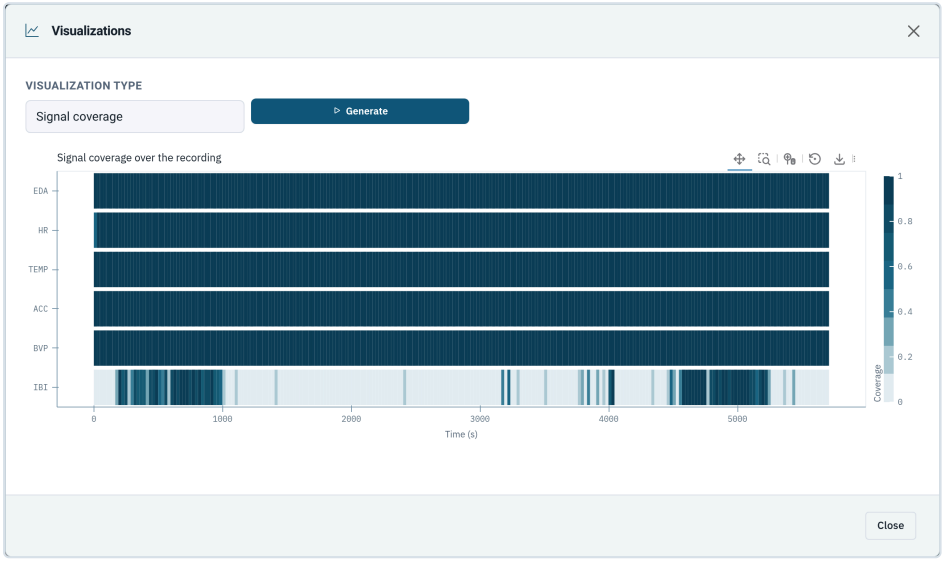

Figure 14 — What the device recorded, and where it did not.

Start here. Coverage shows, per signal, where samples actually exist. Gaps mean the device was off the wrist, the battery died, or the channel was never recorded — and every later number is computed only over what is present.

Correlation

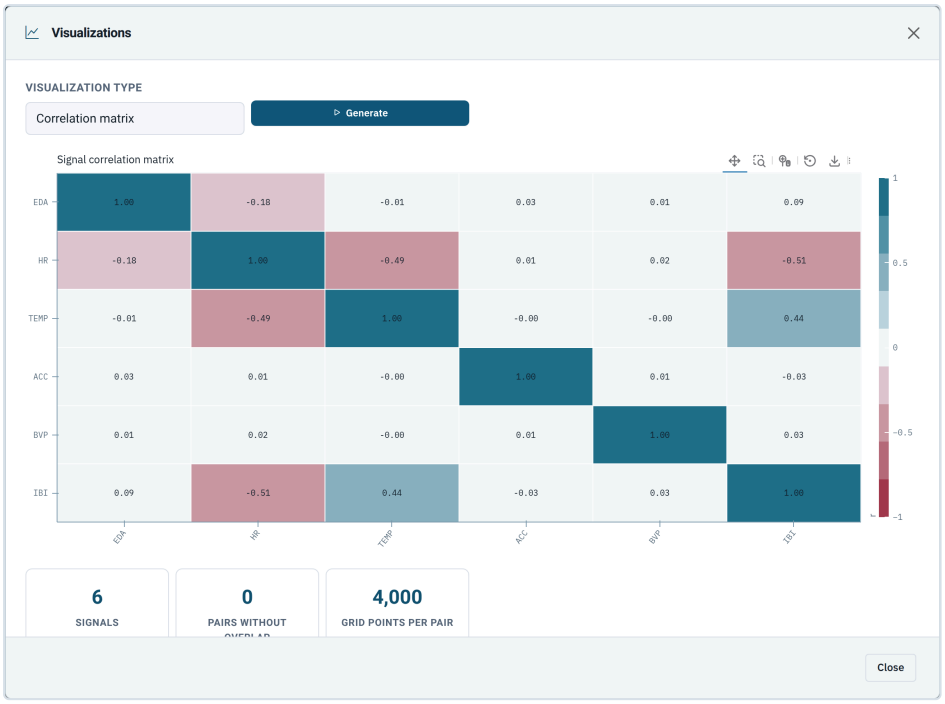

Figure 15 — How the signals move against each other.

Distribution

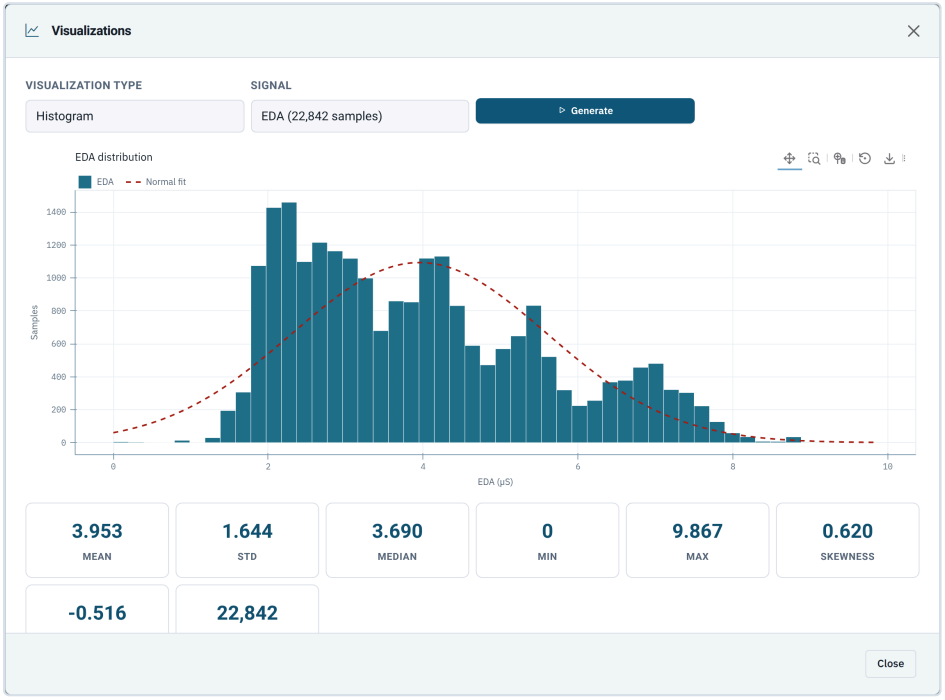

Figure 16 — Any signal as a distribution, with a normal fit.

Poincaré

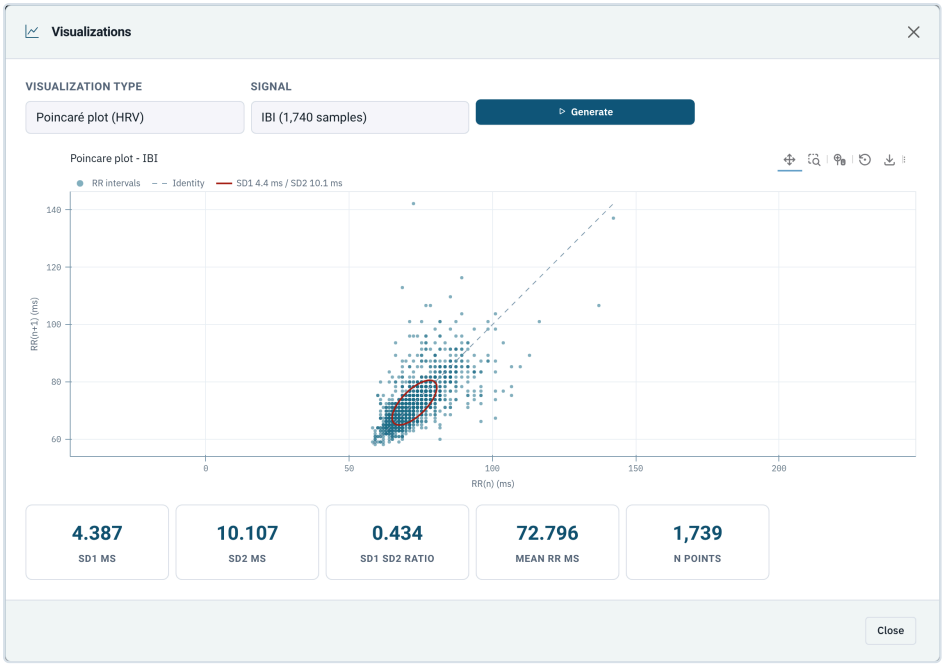

Figure 17 — SD1 against SD2 on the beat intervals.

The classic HRV scatter: each interval against the one after it. The cloud's width across the diagonal is SD1 (short-term variability), its length along the diagonal is SD2 (long-term). A tight cigar means low variability; a fat cloud means high.

Spectrogram

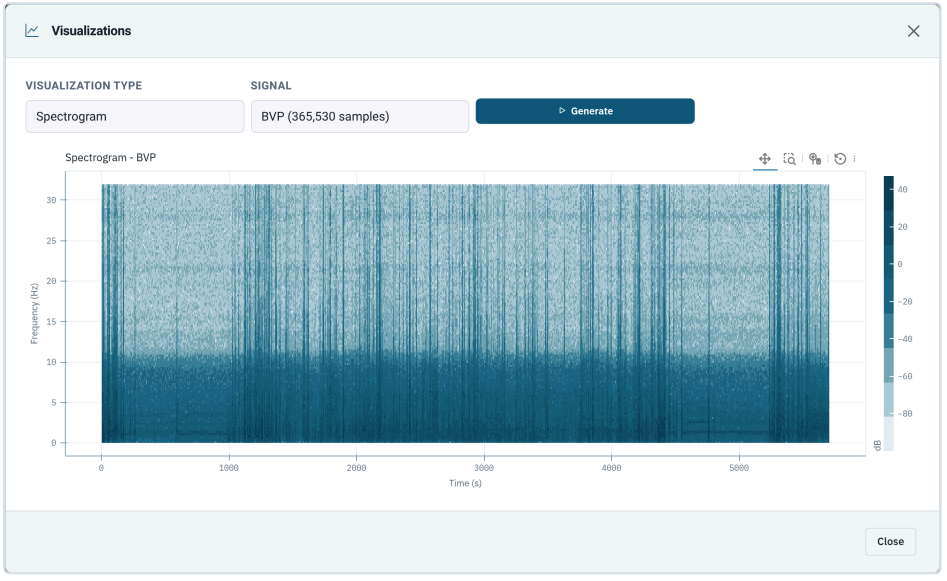

Figure 18 — How the frequency content of a signal moves over time.

Hour-of-day profile

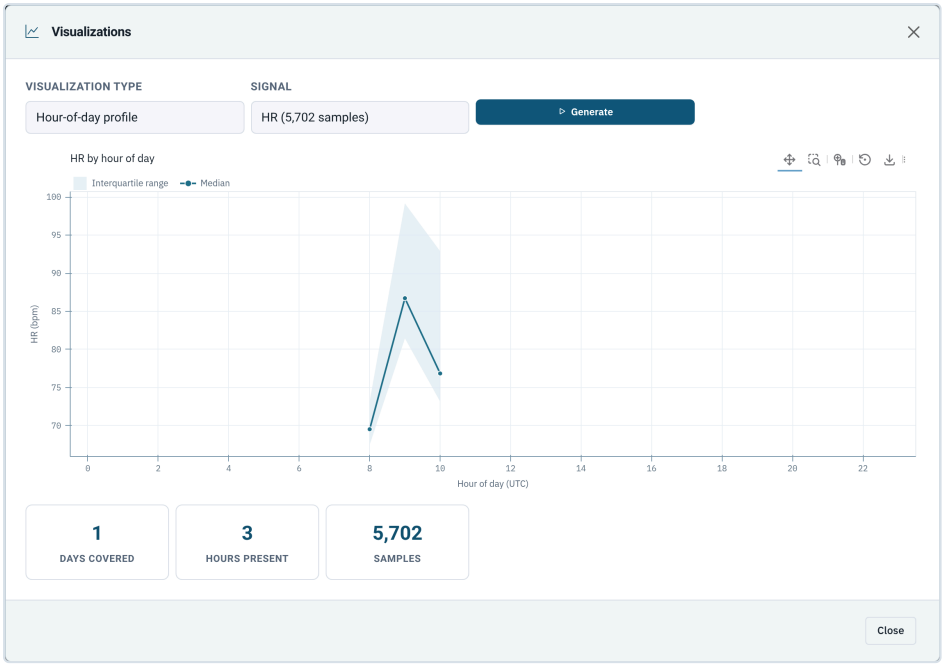

Figure 19 — Hour-of-day profile.

Averages a signal by hour of the local day. On a 95-minute recording this covers only the hours the recording touched — it becomes genuinely useful over days.

Two signals together

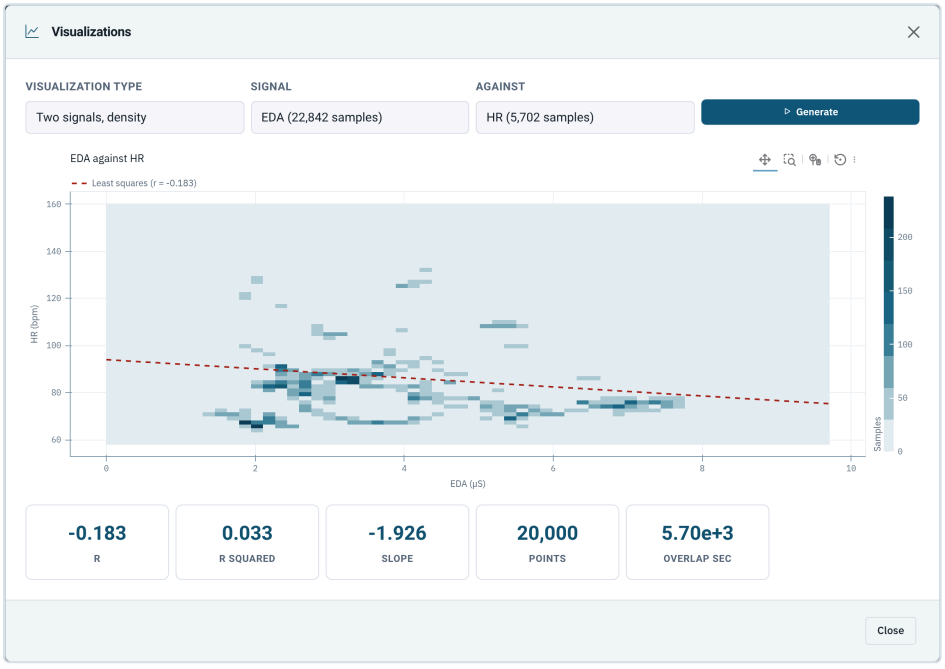

Figure 20 — Two signals against each other, as a density.

EDA decomposition

Figure 21 — EDA split into its tonic and phasic parts.

Two charts this session cannot support

Both of the following are refused on the E4 session, and both refusals are correct.

An actogram needs more than one day

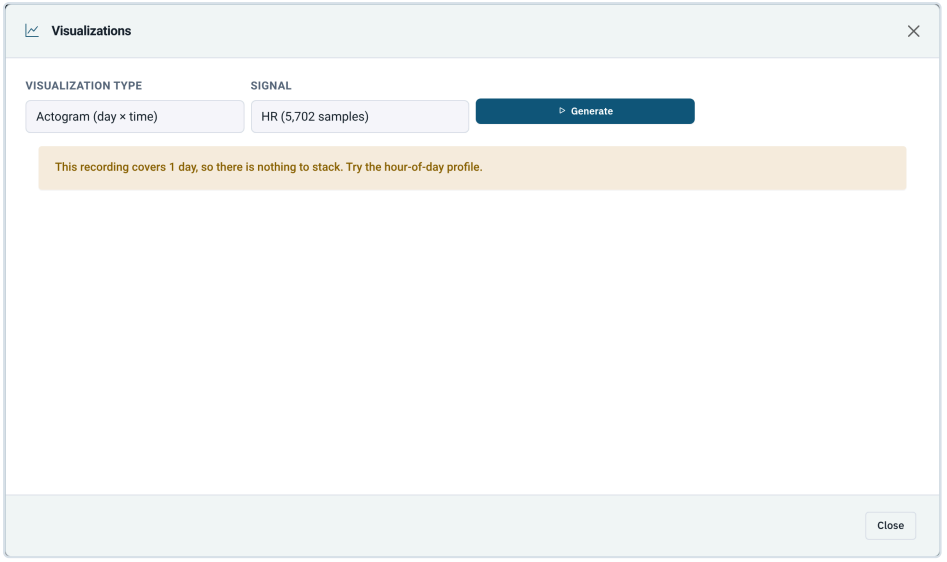

Figure 22 — The actogram declines: this recording covers one day.

An actogram plots one row per day against time of day, so that a rest–activity rhythm becomes visible as a pattern down the page. With a single day there is one row and no rhythm to see.

A composition chart needs states, not measurements

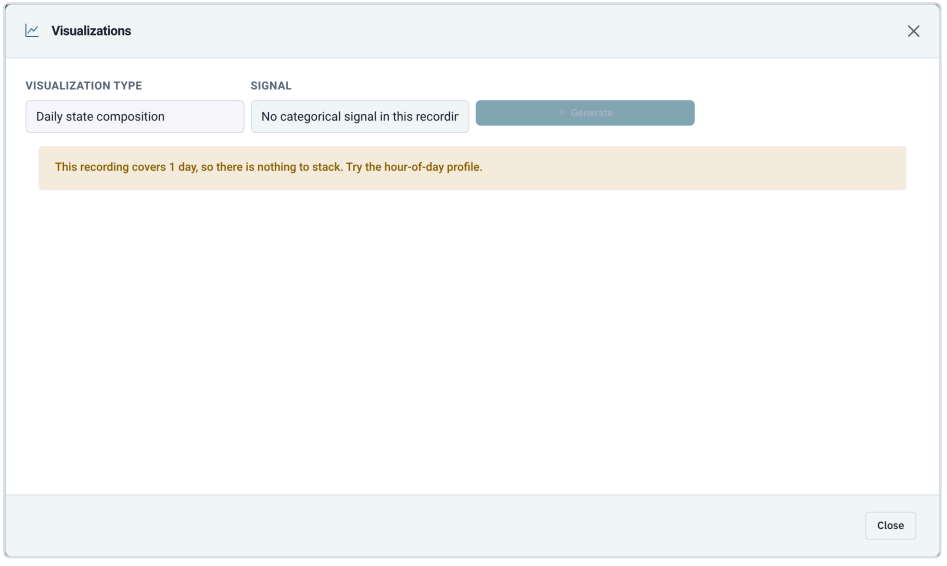

Figure 23 — The composition chart declines: no categorical signal.

This one is refused before you can even ask for it — the signal selector says so and the Generate button stays disabled. A composition chart counts states per day: asleep, awake, sedentary, active. An E4 records continuous measurements and no state signal, so there is nothing to count.

Chapter 8 shows both charts working, on a recording that has what they need.

The same charts on a multi-day recording

Import a Fitbit export and the workspace is unchanged — same panels, same controls, different data.

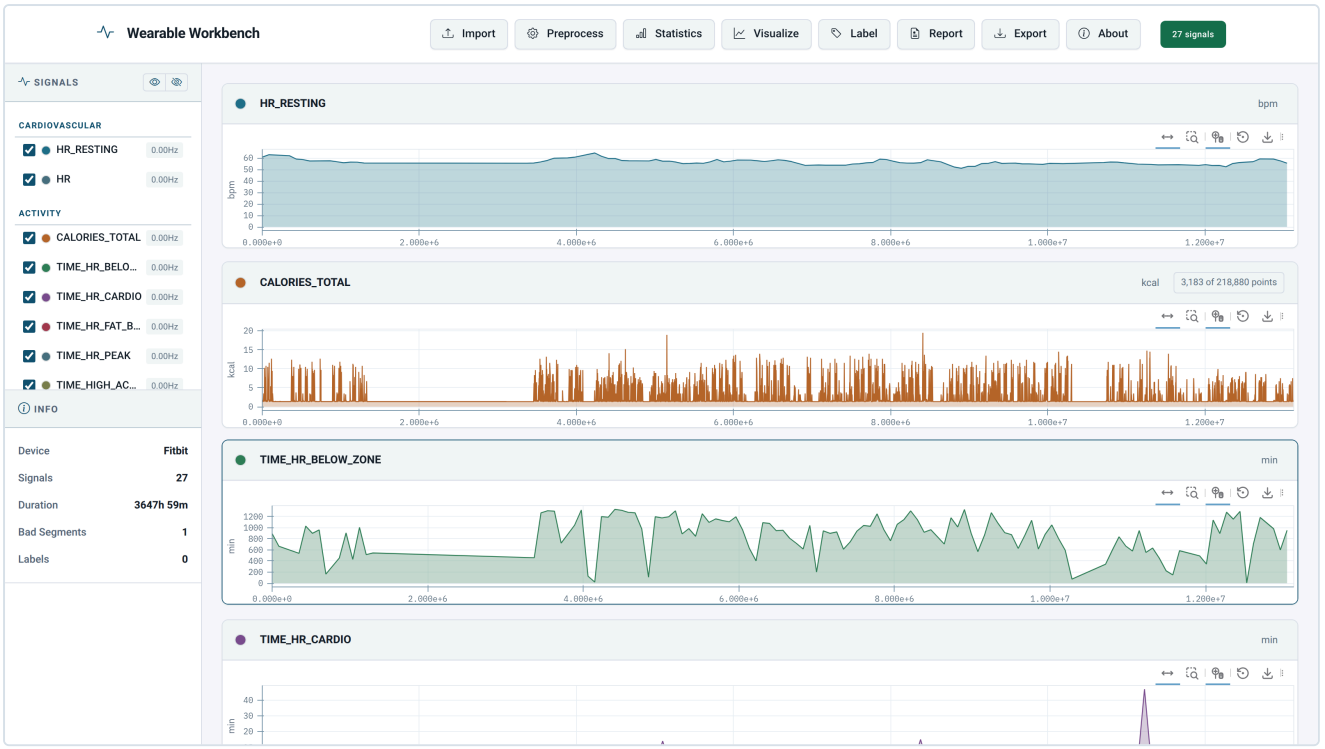

Figure 24 — The same workspace, a different device.

Device	**Fitbit**
Signals	**27**
Duration	**3647h 59m**
Bad Segments	**1**
Labels	**0**

Figure 25 — Five months instead of 95 minutes.

The actogram

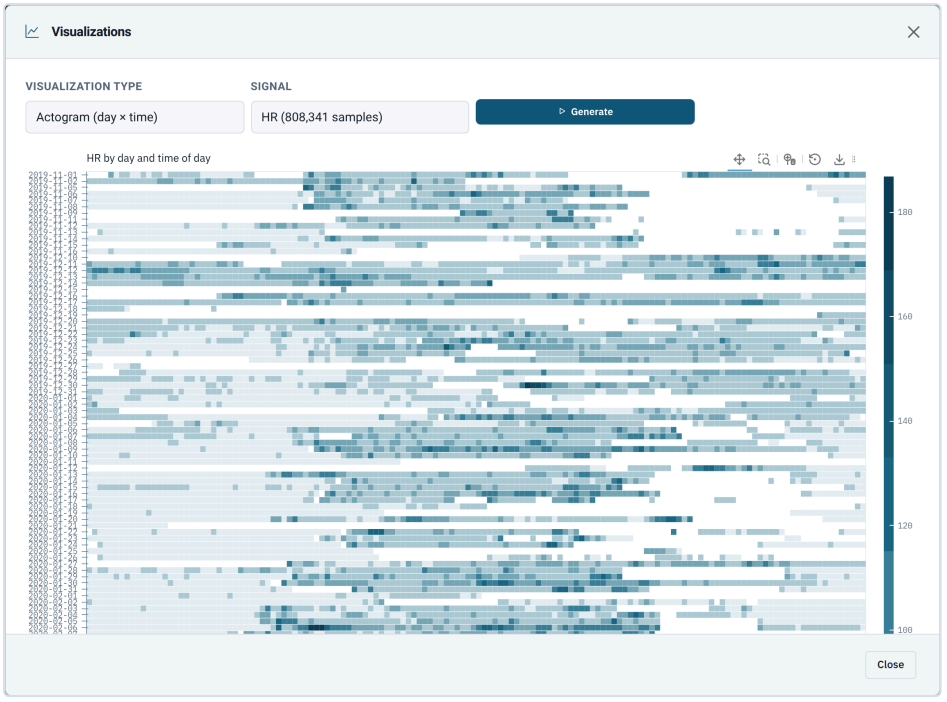

Figure 26 — An actogram over 119 days.

Each row is a day, each column a ten-minute bin of the local clock, and colour is the mean of the signal in that bin. The dark band running down the right and the pale block through the middle of each night are the rest–activity rhythm this chart exists to show.

Daily state composition

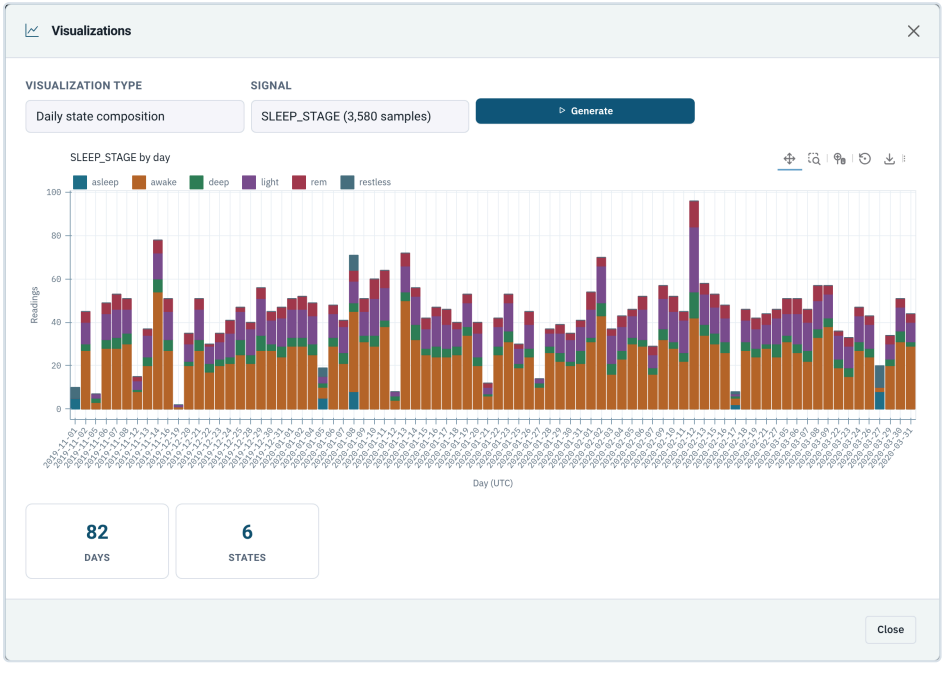

Figure 27 — Daily state composition, from a categorical signal.

This is the chart the E4 could not support. The Fitbit export carries a sleep stage signal — a state per reading rather than a number — so each day becomes a stacked bar of how many readings fell in each state.

And a refusal at the other end

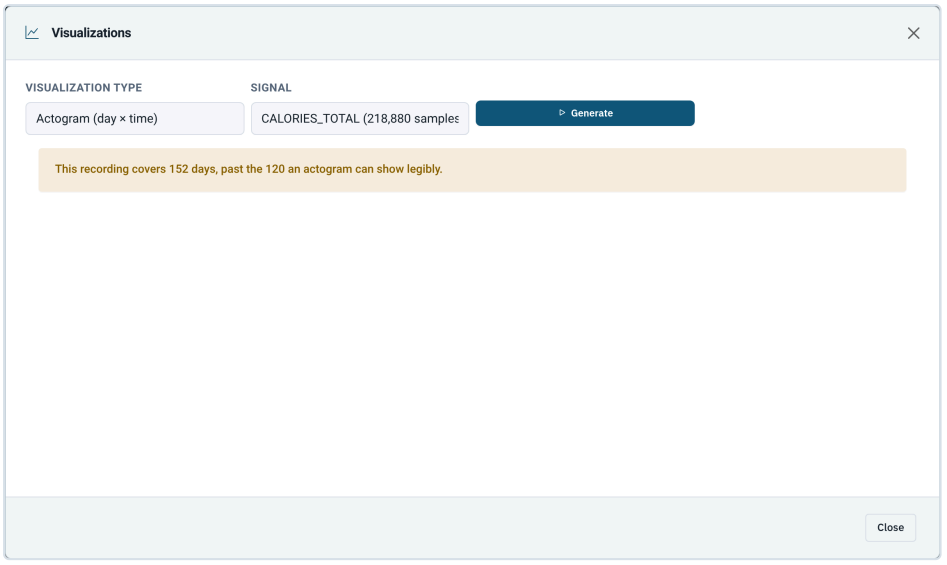

Figure 28 — Refused again, for the opposite reason.

The same chart declines a 152-day signal: past about 120 days the rows are thinner than a pixel and the picture stops being readable. Choosing a signal that covers fewer days, or cropping the session, gets you a chart you can actually read.

Preprocessing

The Preprocessing dialog has six tabs. Auto Process is the one to start with; the rest are manual tools for when it is not enough.

One click per signal type

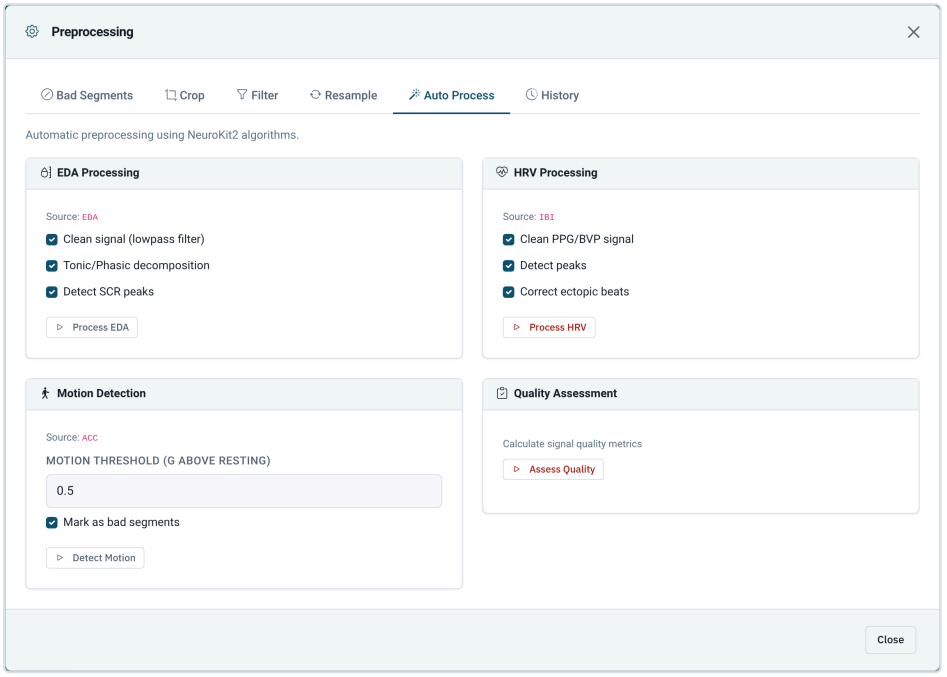

Figure 29 — One-click preprocessing, one policy per signal type.

Four cards, each naming the signal it will work from. A card whose source signal is missing is greyed out with the reason, rather than offering a button that fails.

HRV processing cleans the pulse signal, detects peaks and corrects ectopic beats.

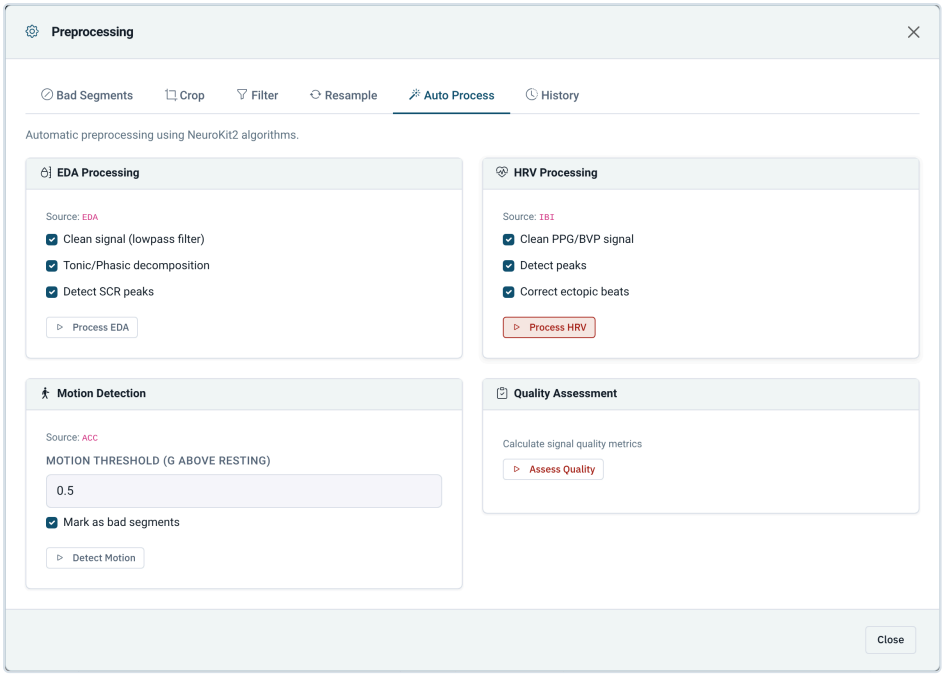

Figure 30 — HRV: artifact detection and interpolation over the intervals.

EDA processing lowpass-filters the signal, then decomposes it into tonic and phasic components — adding both as new signals.

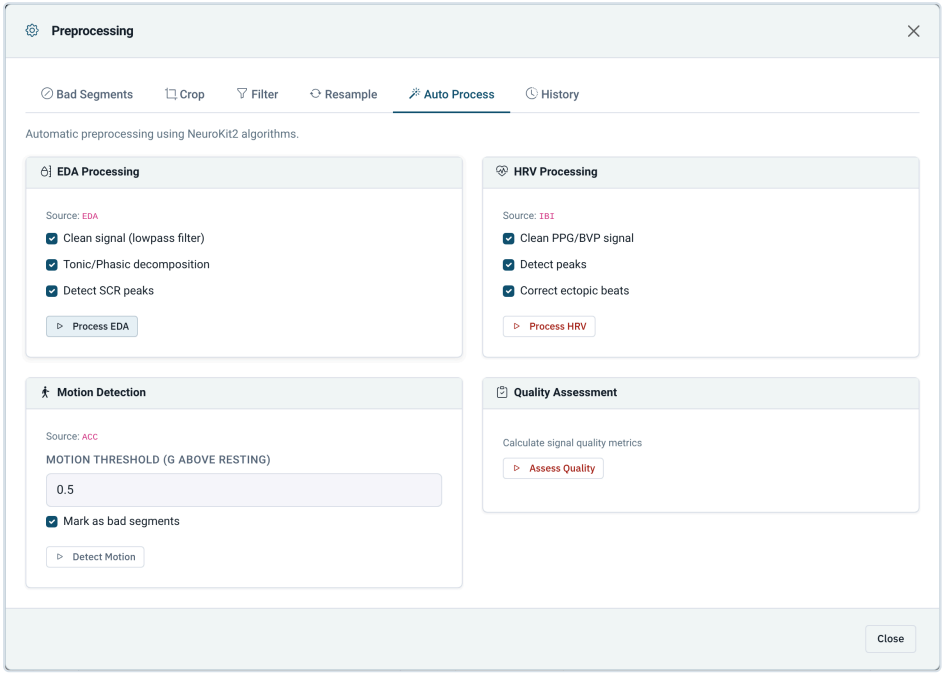

Figure 31 — EDA: lowpass filter, then tonic and phasic decomposition.

Motion detection takes the magnitude from the three accelerometer axes and marks stretches above a threshold, optionally as bad segments.

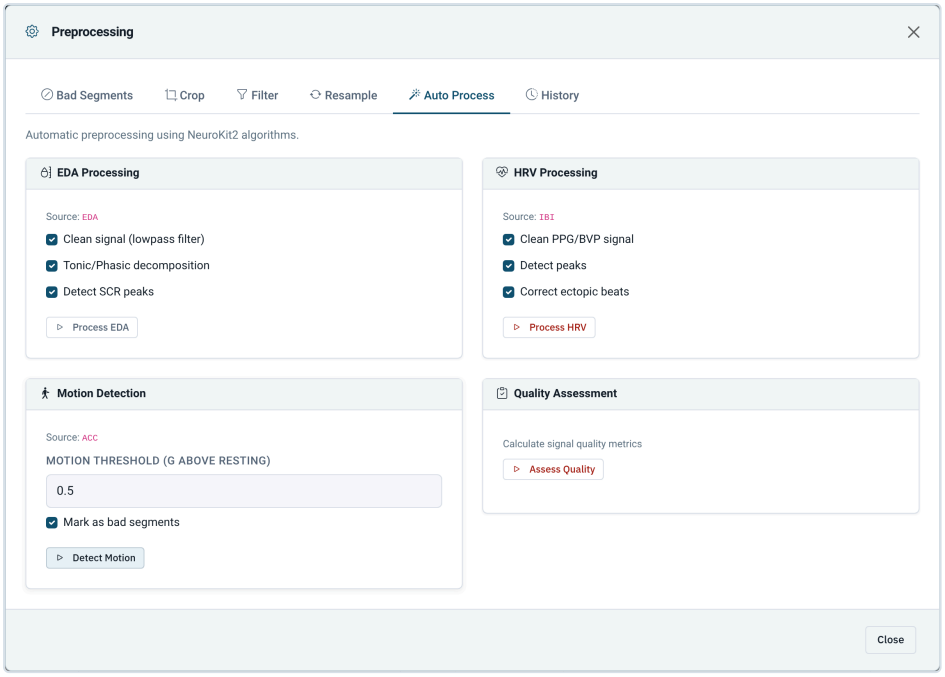

Figure 32 — Motion: magnitude from the three axes, against a threshold.

Quality assessment scores every signal.

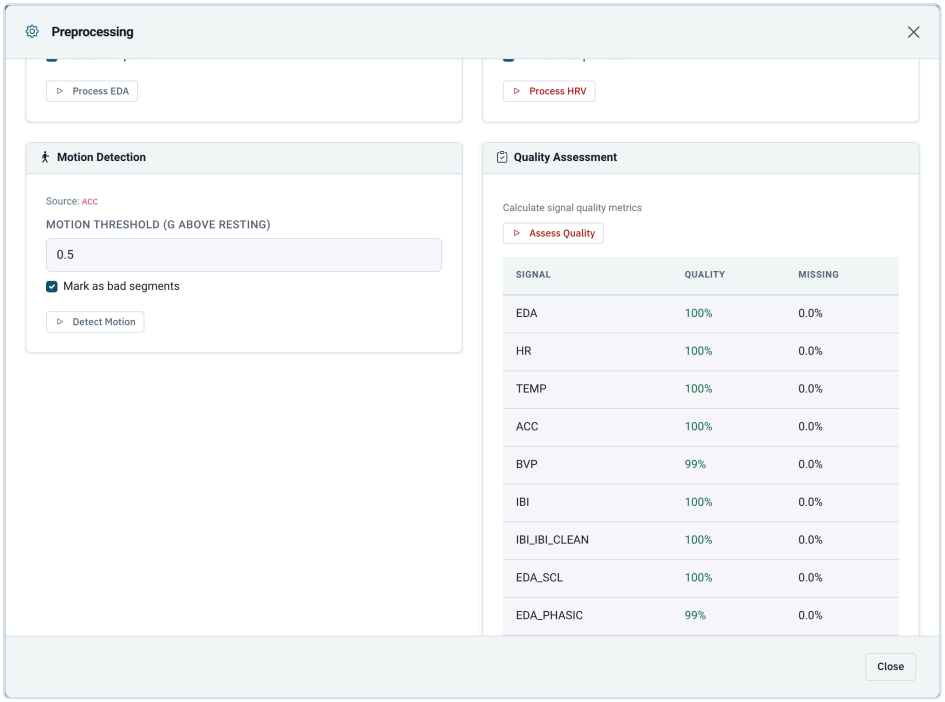

Figure 33 — Quality: a score and a missing percentage per signal.

SIGNAL	QUALITY	MISSING
EDA	100%	0.0%
HR	100%	0.0%
TEMP	100%	0.0%
ACC	100%	0.0%
BVP	99%	0.0%
IBI	100%	0.0%
IBI_IBI_CLEAN	100%	0.0%
EDA_SCL	100%	0.0%
EDA_PHASIC	99%	0.0%

Figure 34 — What the quality assessment reports.

Note that the derived signals appear here too — `IBI_IBI_CLEAN`, `EDA_SCL` and `EDA_PHASIC` were created by the steps above and are assessed like any other.

Filtering

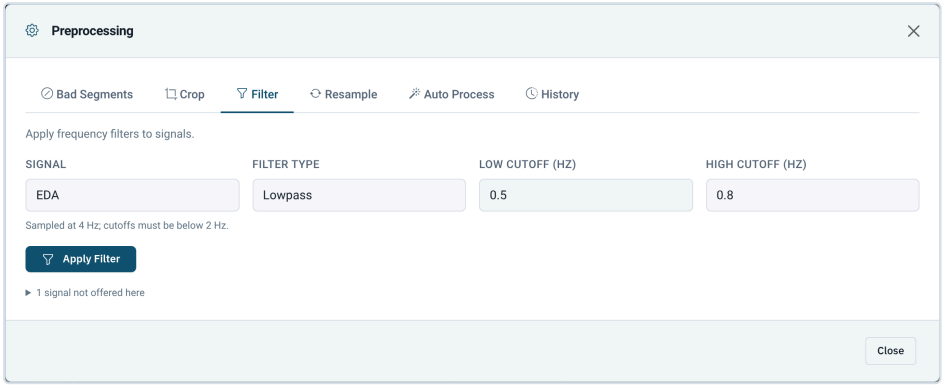

Figure 35 — Filtering, with the refusals stated up front.

Lowpass, highpass and bandpass, per signal. The cutoffs are in Hz and the hint under them gives the valid band for the signal you picked — a filter cutoff above half the sampling rate is meaningless, and the workbench will not let you set one.

Figure 36 — Signals that cannot be filtered, and why.

Irregular signals cannot be filtered at all: a digital filter assumes an even sample spacing that an event series does not have. They are listed with the reason rather than silently omitted from the dropdown.

Resampling

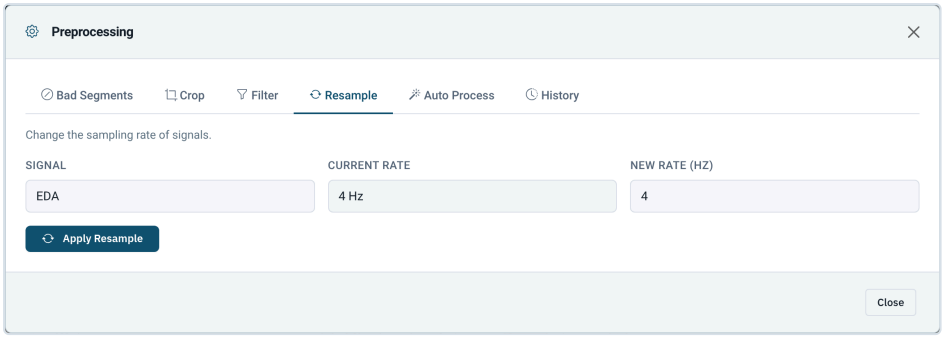

Figure 37 — Resampling a signal to a new rate.

The current rate is shown so you can see what you are changing from. Downsampling loses detail permanently in the working copy; the original file on disk is untouched.

Cropping

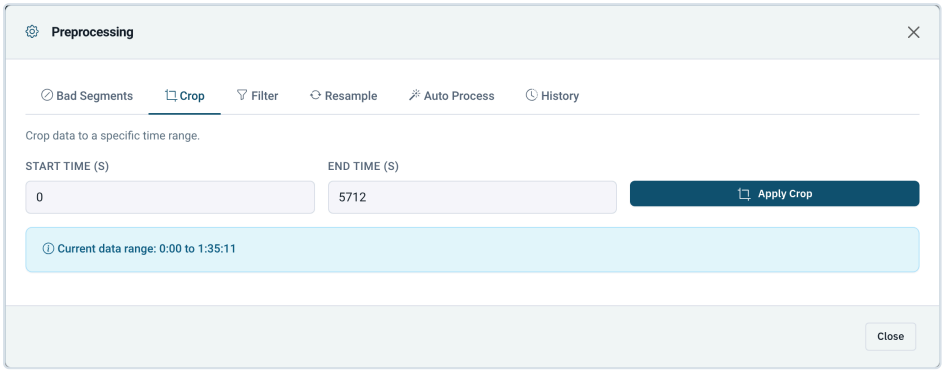

Figure 38 — Cropping the session to a window.

Bad segments

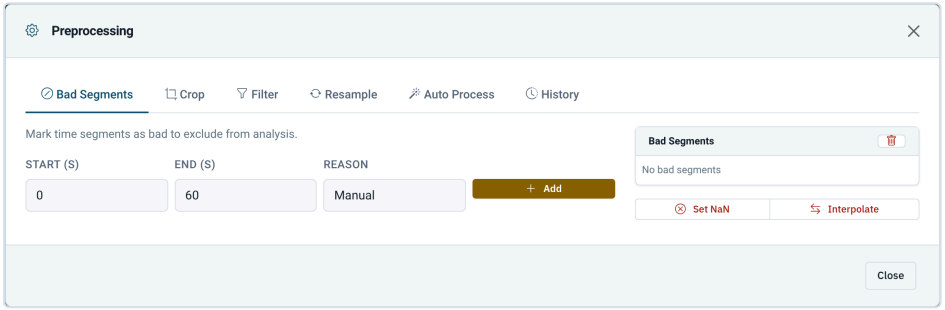

Figure 39 — Marking a stretch bad rather than deleting it.

A bad segment marks a span as untrustworthy with a reason — motion, noise, missing data, a sensor issue — without deleting anything. Statistics can then exclude those spans, and on export you can choose whether they come out as NaN or as the original values. This is the honest alternative to quietly deleting data you did not like.

The history

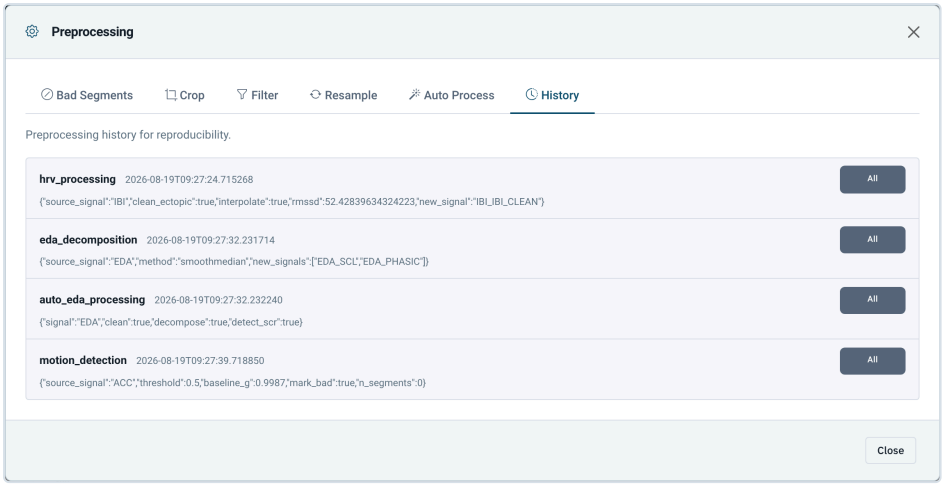

Figure 40 — Every step recorded, and it travels with the file.

Every operation is recorded with its full parameters — the source signal, the method, the thresholds, the signals it created. This is what makes a session reproducible, and it is written into the `.uwf` file on export, so a colleague opening your file can see exactly what you did to it.

The interpreted report

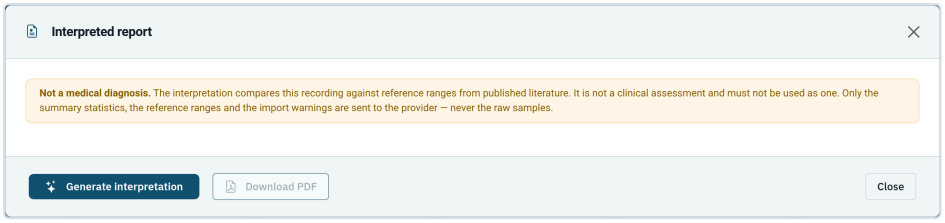

Figure 41 — What the report will send, stated before it sends it.

The report compares the session's summary statistics against reference ranges from published literature and writes up what it finds.

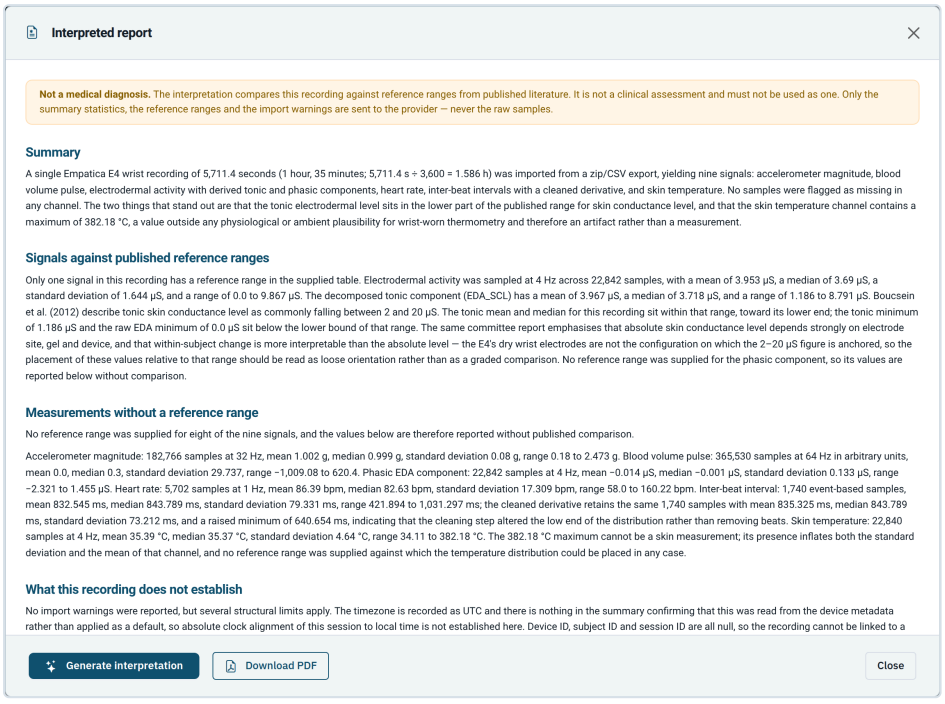

Figure 42 — Measurements placed against the literature, with the citations named.

Three things to notice about the output.

It names its sources. Where a reference range exists it is cited, and where one does not the report says so and declines to compare — the "Measurements without a reference range" section exists precisely so that uncited numbers are not mixed in with cited ones.

It reads the data critically. In this session it flagged a maximum skin temperature of 382 °C against a median of 35 °C, and said plainly that this cannot be a wrist temperature and marks a corrupt sample. That is the kind of thing summary statistics contain and nobody reads.

It states its limits. The banner at the top is not boilerplate: this is not a clinical assessment, and the interpretation of a single session with no repeat measurement is weaker than the same numbers with a within-subject comparison. The report says so itself.

WHAT LEAVES THE MACHINE

Only the summary statistics, the reference ranges and the import warnings are sent to the provider. Raw samples never are. If your governance does not permit even that, leave the API key unset and the feature stays off.

Exporting

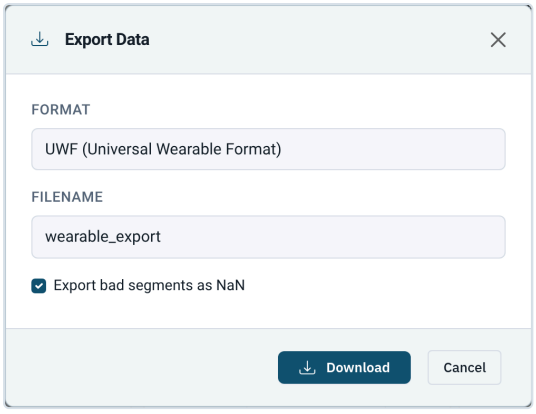

Figure 43 — The Export dialog opens on UWF; the Format list holds two more.

Three formats, and the difference between them is not the file extension but **what one row means**.

Format	One row is	For
UWF	— (a whole session in one file)	Archiving, or handing the session to a colleague
CSV	one raw sample	Analysis in MATLAB, R or Python
Analytics Workbench	one **observation** — a recording, a window, or a label	The Analytics Discovery Workbench

UWF is the workbench's own format: every signal, the labels, the bad segments and the full preprocessing history in one file. It re-imports here with everything intact, which makes it the right choice for archiving a processed session or handing it to a colleague.

CSV writes one file per signal — timestamp and value, every sample. The history does not survive this route, because a column of samples has nowhere to put it, so keep a UWF alongside if the provenance matters.

Export bad segments as NaN decides whether spans you marked bad come out as missing values or as the original samples. Choose NaN when the file is going straight into an analysis that should ignore them; choose the original values when the recipient should judge for themselves.

Exporting to the Analytics Discovery Workbench

The third format is not a different container for the same data. CSV gives you the signal; this gives you **derived measures** — one tidy table where rows are observations and columns are variables, which is the shape the Analytics Workbench reads.

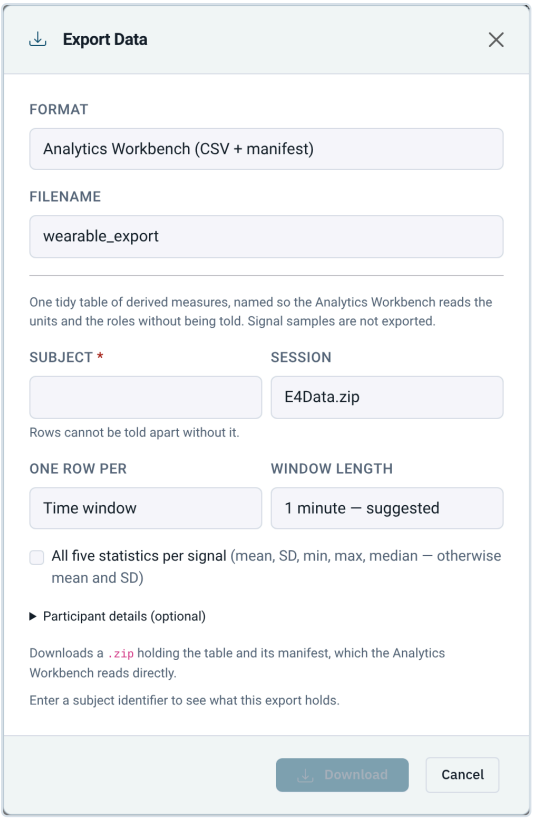

Figure 44 — The export is refused until a subject identifies the rows.

A subject identifier is required, and the Download button stays disabled until you give one. This is not bureaucracy. The moment this table is loaded beside another recording — which is the entire reason for exporting it — rows with no subject cannot be told apart, and no test can be run on them. Where the device export carried a subject id, as Dexcom and Apple Health do, the field is filled in for you; otherwise you type it.

Session defaults to the file the recording came from. Give it the name of the visit if you are building a cohort, so the same subject's two sessions do not collide.

One row per what

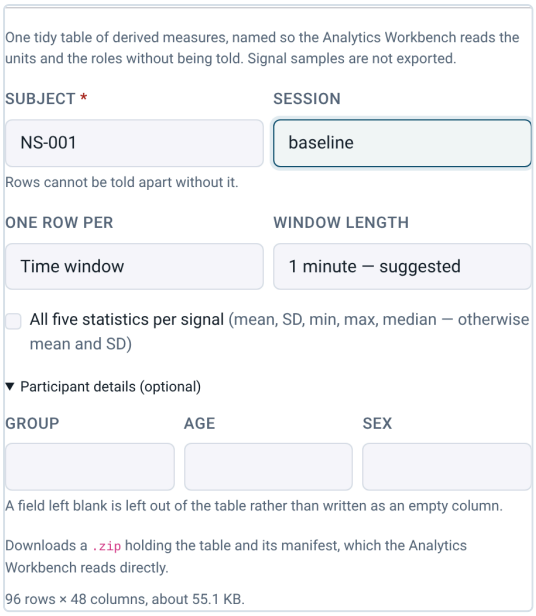

Figure 45 — One row per window, with the shape of the table stated before it is written.

Granularity	Rows	Use it for
Whole recording	one	One row per subject per visit — the cohort table
Time window	one per window	Change within a session; a day-long window over a multi-day recording
Label	one per label	A within-session condition contrast

Window length offers one minute to one day, and the dialog proposes a length from the recording's own duration — the aim is a few hundred rows, not a row per second. A window holding no samples at all is dropped rather than written as a row of blanks, and the count is reported.

The **Label** option is available only when the labels you have imported fall inside the recording. If they do not, it greys out and says so; see *Marking the experimental structure* for why a timezone mismatch is the usual cause.

What comes out

Every signal is exported, not just HRV and EDA. What each one contributes depends on what kind of quantity it is:

The signal is	Columns per signal
An instantaneous measure (BVP, EDA, TEMP)	Mean, SD, Min, Max, Median, Valid%
A per-interval total (steps, calories)	Total, and a per-day rate
A category (sleep stage, activity intensity)	The share of the window spent in each state

so the column set differs by device. An E4 summary is 48 columns; an Oura export is over 200. On top of those come the HRV and EDA blocks, and the quality columns — coverage, bad-segment share, preprocessing steps — which arrive as covariates rather than as measurements.

All five statistics per signal widens Mean and SD to the full five. Leave it off for a windowed export unless you need them: it more than doubles the column count, and a per-day rate is only written when the recording is long enough for one to mean anything.

Before you commit, the dialog states **rows × columns and the size** the file will be.

> **THE MANIFEST IS NOT OPTIONAL**
>
> The download is a `.zip` holding the table and a `_analytics.json` beside it. The manifest carries the units, the column roles and this session's preprocessing. Without it the Analytics Workbench has to infer a role from each column name, and the ones it infers wrongly do not error — they quietly stop counting as measurements. There is no checkbox because there is no case for leaving it out.

Appendix A — Control reference

Toolbar

Control	Effect
Import	Opens the import dialog; also replaces the current session
Preprocess	Filtering, resampling, cropping, bad segments, auto processing, history
Statistics	Basic, HRV and EDA statistics
Visualize	The ten chart types
Label	Add, edit and load experiment labels
Report	The interpreted report
Export	UWF, per-signal CSV, or a table for the Analytics Workbench
About	Version and build information

Import

Control	Effect
Device type	Which reader to apply. Required — the format is not sniffed
Timezone	The zone the recording is read in
File picker	One archive, or every file of a folder export

Statistics

Tab	Reports
Basic	Mean, median, SD, range and count per signal
HRV	Time domain, frequency domain and nonlinear metrics, with the source declared
EDA	Tonic level and phasic response measures

Visualize

Type	Needs	Shows
Signal coverage	—	Where samples exist, per signal
Correlation matrix	Two or more signals	Association between signals
Two signals, density	Two signals	One signal against another
Hour-of-day profile	One signal	Mean by hour of local day
Actogram	One signal, 2–120 days	Day × time-of-day heatmap
Daily state composition	A categorical signal	States counted per day
Histogram	One signal	Distribution with a normal fit
Poincaré plot	Beat intervals	SD1 against SD2
Spectrogram	One regular signal	Frequency content over time
EDA decomposition	An EDA signal	Tonic and phasic components

Preprocess

Tab	Effect
Auto Process	HRV, EDA, motion and quality, one click each
Filter	Lowpass, highpass or bandpass on a regular signal
Resample	Change a signal's sampling rate
Crop	Restrict the session to a time window
Bad Segments	Mark a span untrustworthy, with a reason
History	Every step and its parameters

Export

Control	Effect
Format	UWF, per-signal CSV, or a table for the Analytics Workbench
Filename	The name of the download; the extension follows the format
Export bad segments as NaN	Bad spans come out as missing rather than as samples

Shown only for an analytics export:

Control	Effect
Subject	**Required.** Identifies the rows once the table is loaded beside another
Session	Names the visit; defaults to the source file
One row per	Whole recording, time window, or label
Window length	One minute to one day; a length is proposed from the duration
All five statistics per signal	Mean, SD, Min, Max, Median rather than Mean and SD
Participant details	Group, age and sex; a blank field is left out rather than written empty

Appendix B — Supported devices

Twenty-three devices plus the workbench's own format. Choosing the right one matters: each reader knows its device's quirks, and the wrong reader will either fail or import the wrong thing.

Device	Formats
ActiGraph	`.gt3x`, `.csv`
Philips Actiwatch	`.awd`
Apple Health / Apple Watch	`.xml`
Axivity	`.cwa`
Wahoo (ELEMNT)	`.fit`, `.zip`
COROS	`.fit`, `.zip`
Suunto	`.fit`, `.zip`
GENEActiv	`.bin`
Empatica E4	`.zip`, `.csv`
Empatica EmbracePlus	`.zip`, `.csv`, `.avro`
Fitabase (Fitbit research export)	`.csv`
Fitbit	`.json`, `.csv`
Dexcom (Clarity)	`.csv`
FreeStyle Libre (LibreView)	`.csv`
Levels	`.csv`
Nutrisense	`.csv`
Garmin (CARP export)	`.json`, `.zip`
Garmin Connect	`.fit`, `.zip`
MMASH (Polar H7 + ActiGraph)	`.csv`
movisens (UNISENS)	`.xml`, `.bin`, `.csv`
Oura Ring	`.csv`, `.json`
Shimmer3	`.csv`
Withings Health Mate	`.csv`
Universal Wearable Format	`.uwf`, `.json`

> **TWO READERS SHARE THE `.BIN` EXTENSION**
>
> ActiGraph and GENEActiv both produce `.bin` files and they are unrelated formats. The extension does not identify the device, which is why the device type is a choice you make rather than something guessed from the filename.

Appendix C — What the workbench refuses to do

A refusal is working as designed. These are the ones you are most likely to meet.

An actogram on a single day. It needs at least two, so that one row can be compared with the next.

An actogram on more than about 120 days. Rows thinner than a pixel are not a chart. Crop the session or pick a signal covering fewer days.

A composition chart with no categorical signal. It counts states per day. A device that records only continuous measurements has no states to count, and the chart is disabled rather than offered and then failed.

A filter on an irregular signal. Digital filtering assumes even sample spacing. Event series such as inter-beat intervals do not have it, and are listed as not filterable with that reason.

A filter cutoff above the Nyquist frequency. The valid band for the chosen signal is shown beneath the cutoff boxes.

An auto-processing card whose source signal is absent. The card is greyed out and names what is missing, rather than offering a button that will fail.

HRV from a session with too few intervals. The source, sample count and duration are printed above every HRV result so you can judge whether the frequency-domain figures are supportable.

Appendix D — Recordings used in this guide

Recording	Provenance
`E4Data.zip`	An Empatica E4 wrist session, 1 h 35 m, six signals (ACC, BVP, EDA, HR, IBI, TEMP), shipped in `SampleData/E4Zip/`. The three experiment labels in this guide were added by hand to demonstrate labelling.
PMData p03	The Simula **PMData** dataset — a Fitbit export folder covering roughly five months for one participant, shipped in `SampleData/PMData/`. Used only for the actogram and the composition chart.

Neither is patient data. If you reuse PMData beyond testing, check its licence and attribution requirements first.