

cyRAVEN

Flow cytometry files in, population frequencies, a UMAP and statistics out.

Every sample is gated separately, and the report is one file.

Version 1.0.0 · 07 November 2026

https://bhagesh-h.github.io/cyRAVEN

Contents

1 About cyRAVEN **2**

Why per-sample gates . . . 2
Two ways to use it . . . 2
How it works: the ten stages . . . 2
Why explore mode is built this way . . . 4

2 About flow cytometry **7**

1. One cell, one row . . . 7
2. Why cells glow . . . 7
3. The cut . . . 7
4. The three directions . . . 8
5. A worked example . . . 8
6. FSC and SSC . . . 8
7. The `-A`, `-H` and `-W` suffixes . . . 8
8. How cyRAVEN uses each channel . . . 9
9. Declare the big populations first . . . 9

3 Get started **10**

Setup . . . 10
Quick start . . . 10
Your own data . . . 10
Inputs: the sample sheet and the config . . . 10
The gating specification . . . 12
Commands and every option . . . 17

4 Advanced **31**

Explore mode: unsupervised discovery . . . 31
Diagnostics, in reading order . . . 34
Every output file . . . 38
Statistics . . . 48
Using cyRAVEN with cyCONDOR . . . 51
Driving it from Claude Code . . . 53

5 Gallery **55**

0. The data, and how to reproduce every figure below . . . 55
1. Before any number . . . 56
2. Placement precision and counting sufficiency . . . 57
3. Are the populations what they were declared to be . . . 59
4. The shared embedding . . . 60
5. Why a threshold did not resolve . . . 65
6. The results . . . 65
7. Is the contrast confounded . . . 67
The two figures this cohort cannot produce . . . 67
Where the clinical figure designs come from . . . 68
What this example does and does not show . . . 68

6 Limitations **69**

Known limitations . . . 69
Scope: what cyRAVEN does not do . . . 70

1 About cyRAVEN

cyRAVEN turns a folder of FCS files into population frequencies, marker expression, a UMAP and group statistics. This page explains what happens to the data at each stage, and why it is done that way.

New to flow cytometry? Read About flow cytometry first. Want to run something now? Go to Get started.

Why per-sample gates

Manual gating varies between operators. When 38 operators gated the same files, the median expanded uncertainty was 3.6%, and experience made no significant difference (Grant and others, 2021, *PDA J Pharm Sci Technol* 75:33). Across 320 clinical samples analysed by six technologists on three harmonised cytometers, the remaining spread followed the operator more than the instrument (Mead and others, 2026, *Cytometry B*, doi:10.1002/cyto.b.70048).

Copying one set of gate coordinates to every sample does not solve this. Staining intensity differs between samples, so a fixed cut is right for some and wrong for others, and the error then follows staining intensity. cyRAVEN places each cut inside its own sample and reports how precisely that cut is determined.

Statistics follow one rule: **the sample is the unit, not the cell.** A tube run for longer gives more events but no more information about the study, so every test first reduces each sample to one value. Several samples from one donor are not independent either; `--paired-column` and `--condition-column` pair them. Anything that cannot be reduced to one value per sample is reported as an effect size without a p-value.

Two ways to use it

	Declared (default)	**Explore** (`--explore`)
You supply	A list of populations and their marker rules	Only the FCS files
You get	Named populations, with uncertainty and statistics	Numbered clusters over every usable channel
It can find	Only what you listed	Populations nobody listed
Cells used	The CD45+ parent gate	Every event, after its own debris and dead-cell screen

Explore writes to `<outdir>/explore/` and changes nothing else, so adding it to a run is safe. `--explore-only` needs no population list at all. The two halves share nothing unless you add `--maybe-learn`; see Explore mode.

How it works: the ten stages

One call to `run_cyraven()` runs every stage. Each stage is also an exported function, so you can run or replace a single step.

Stage	In plain words	Function
1 Read	Open each file and work out which channel is which marker	`read_fcs_resolved()`
2 Transform	Put intensities on an arcsinh scale, which spreads the dim end and compresses the bright end	`derive_cofactor()`
3 Gate	In each sample, cut each marker at the lowest point between its negative and positive peaks	`density_valley()`, `resolve_threshold()`
4 Score	Count the cells that meet each population's rule from your YAML	`score_populations()`
5 Uncertainty	Re-derive every cut on resampled events to see how far it moves	`threshold_uncertainty()`
6 Embed	Build a UMAP with about the same number of cells from each sample	`run_umap()`
7 Test	Compare groups with one value per sample	`stats_group_comparison()`
8 Diagnose	Check whether the gates and populations held up	`stats_threshold_drift()`, `cluster_gate_agreement()`
9 Explain	Describe an unexplained cluster as a gating strategy	`explain_cluster()`
10 Report	Write tables, figures, the run manifest and `report.html`	`write_run_manifest()`

The sections below give the detail for each stage.

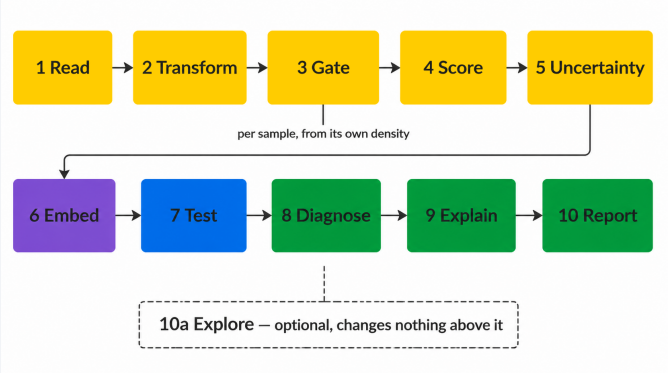


Figure 1.1: The ten pipeline stages in order, with explore mode attached as an optional branch

1 Read

`read_fcs_resolved()` resolves channels to marker symbols from $PnS with fallback to $PnN, and applies the acquisition spillover matrix where the file carries one. Spectral instruments write unmixed data with no matrix, so `maybe_compensate()` detects and reports rather than assuming; applying compensation twice is as damaging as omitting it.

2 Transform

```
set.seed(42)
mk <- c("CD3", "CD4", "CD8", "CD19")
ex <- matrix(abs(rnorm(20000 * 4, 0, 250)),
             ncol = 4,
             dimnames = list(NULL, mk))
cf <- derive_cofactor(
  ex, setNames(1:4, colnames(ex)))
round(as.numeric(cf), 1)
```

```
#> [1] 157.2
```

The arcsinh cofactor governs the width of the quasi-linear region near zero. The conventional value of 5 derives from mass cytometry and over-expands the background band on spectrally unmixed fluorescence data. Each marker is bisected until the interquartile range of its background distribution reaches a target, and the panel adopts the median. `derive_cofactor_pooled()` estimates across samples so that a single weakly stained file does not set the scale on which every threshold in the run is computed.

3 Gate

```
# a bimodal marker
x <- c(rnorm(8000, 0, 0.4),
       rnorm(2000, 4, 0.5))
round(density_valley(x), 2)
```

```
#> [1] 1.52
```

Thresholds are placed at the density minimum within each sample. `density_valley()` returns `NA` for unimodal distributions, after which `resolve_threshold()` uses a declared control if there is one, otherwise a tail gate above the dominant mode, and only then a flagged quantile. Derivation is recorded per threshold in `thresholds_used.csv`.

Per-sample thresholding accommodates staining variation but permits the definition of a population to differ between groups, which `stats_threshold_drift()` exists to detect.

4 Score

```
spec <- default_population_spec()
str(spec[["CD3 pos CD4 pos"]])
```

```
#> List of 2
#>  $ CD3: chr "above"
#>  $ CD4: chr "above"
```

Populations are evaluated as Boolean conjunctions of marker directions read from YAML. Editing the specification changes the analysis without modifying R code.

5 Quantify the uncertainty

```
u <- threshold_uncertainty(
  x_parent, source = "valley", B = 100)
c(u$u_sampling, u$u_method, u$u_combined)
```

Each cut is re-derived from resamples of its own events, and again across the settings that placed it. The two spreads combine in quadrature and propagate to every population reading that marker, alongside the CD45 parent term, which enters all of them because it fixes the denominator.

The reported threshold is unchanged: the perturbation runs on copies. Every entry point saves and restores `.Random.seed`, so the cell selection in the next stage draws from the same stream position whether this runs or not.

6 Embed

```
emb <- run_umap(mat, n_neighbors = 30,
                min_dist = 0.3, seed = 42)
```

`select_umap_features()` excludes scatter, viability, time and the height and width duplicates of area channels; retaining these was the principal cause of uninformative embeddings in the precursor implementation. `plan_subsample()` draws equal cell numbers per sample so that acquisition depth does not determine the manifold.

`ret_model = TRUE` retains the model for projection of later batches through `project_umap()`, which holds coordinates fixed between runs. Model retention alters the embedding; see `?run_umap`.

7 Test

```
ids <- sprintf("S%02d", 1:12)
freq <- data.frame(
  sample_id = rep(ids, each = 2),
  population = rep(c("CD4 T cells",
                     "B cells"), 12),
  pct_of_cd45_pos = c(rbind(
    rnorm(12, 25, 4), rnorm(12, 10, 2))),
  is_control = FALSE, qc_status = "pass")
grp <- setNames(
  rep(c("HC", "Patient"), each = 6),
  ids)

res <- stats_group_comparison(
  freq, grp, reference = "HC")
res[, c("population", "comparison_group",
        "n_reference", "cliffs_delta",
        "p_value")]
```

```
#>    population comparison_group n_reference
#> 2 CD4 T cells          Patient           6
#> 1     B cells          Patient           6
#>   cliffs_delta   p_value
#> 2       -0.167 0.6991342
#> 1        0.000 1.0000000
```

`stats_marker_state()` applies the same treatment to marker expression within a population, testing median intensity and percent positive separately because a shift confined to a subset moves one and not the other.

8 Diagnose

Six outputs constrain interpretation of the preceding stages: threshold drift, gate and counting uncertainty, phenotype concordance, gate against cluster concordance, learned gate geometry, and conformance against a baseline. The Diagnostics covers each in full.

`batch_mixing_report()` quantifies batch structure in the embedding against a permutation null and reports Cramér's *V* between batch and group. Correction through `--correct-batch` is refused above a configurable *V*, where batch and biological effect are not separable.

`stats_threshold_drift()` tests whether marker thresholds separate by group, which would render an abundance difference partly definitional.

`stats_confounding()` reports whether a covariate both differs across groups and associates with the outcome within groups, the two conditions jointly defining a confounder.

`cluster_gate_agreement()` cross-tabulates declared labels against unsupervised clusters, which can reveal a population absent from the specification, or a threshold rejecting events that belong to its population.

`run_gate_uncertainty()` reports how far each threshold moves under resampling and propagates that to every population reading it, so a difference smaller than its own gate's movement is identifiable as such.

9 Explain

`cluster_gate_agreement()` identifies a cluster matching no declared population but cannot characterise it, leaving a cluster index and a cell count.

`explain_cluster()` inverts the gating stage. Given labelled events it selects the two most discriminating markers, fits a convex polygon in that plane, retains the enclosed events and recurses, producing the topology of a manual gating strategy.

```
st <- explain_cluster(marker_matrix,
                      cluster_id == 7)
# marker pair and metrics per level
st$summary
# vertices, on the analysis scale (arcsinh or logicle)
st$levels[[1]]$polygon
```

Convex polygons rather than thresholds: a conjunction of one-dimensional cuts is an axis-aligned rectangle, and an oblique boundary constrained to a rectangle must admit contaminants or exclude genuine events. `derive_singlet_band()` is a hand-written instance of the same geometry.

Metrics are computed on held-out events with the resubstitution value alongside, since eight free half-planes can memorise a few thousand events. The stage is descriptive: it carries no p-value and scored frequencies are identical whether it runs or not.

`--external-labels` supplies the labels from elsewhere, a cyCONDOR clustering for instance, and `gate_transferability()` then refits the strategy with one donor withheld and scores it on that donor. Held-out events come from the same donors and overstate how well a gate travels; the per-donor minimum does not. `--export-gates` writes the result as Gating-ML 2.0 on the scale it was fitted on, and as a vertex table in the units the FCS file stores.

10 Report

Figures, tables, and `write_run_manifest()`, which records R version, package versions, git commit and working-tree state, the full invocation, and every input file with size and modification time. The manifest is written before the expensive stages and rewritten at completion, so an interrupted run records `status: failed`.

10a Explore

`--explore` runs a second, independent analysis over the single cells of every file (scatter and singlet gates only) and every eligible channel, writing to `<outdir>/explore/` and leaving stages 1 to 10 untouched. It has its own quality gate, its own embedding and clustering, and its own self-contained report. `--explore-only` runs it without stages 2 to 10 at all. See Explore mode.

Why explore mode is built this way

The design record for explore mode: what the supervised and unsupervised halves can lend each other, which of those exchanges are implemented, which are refused, and where that leaves cyRAVEN against the tools it sits beside.

Written because “add unsupervised clustering” is the easy half of the problem. The hard half is deciding what the two halves are allowed to tell each other without losing what made each one useful.

1. The two failure modes

Supervised and unsupervised cytometry analysis fail in opposite directions, and neither can detect its own failure.

Supervised analysis cannot find what it was not told to look for. You declare twelve populations, it scores twelve populations. A thirteenth cell type sits in the data unremarked. Worse, a declared label may cover two distinct phenotypes, and their total can stay flat while one rises and the other falls. That is a real pattern that no amount of testing on the declared table can surface, because the table has one column where the biology has two.

Unsupervised analysis cannot tell you what it found. FlowSOM returns cluster 17. Whether cluster 17 is a B cell subset, a doublet artefact, or the same cells as cluster 4 under a different sampling seed is left to a human squinting at a heatmap. And because the clusters are defined on the data, the number of them that reach significance moves with parameters nobody reports. Measured elsewhere in this project, one cyCONDOR run went from **1 significant cluster to 8** by changing only how many cells were drawn.

Running both and reading them side by side does not fix either. What fixes them is deciding which specific facts may cross between them.

2. The exchange, enumerated

Every transfer that is technically possible between the two halves, with the verdict on each.

2.1 Declared -> explore

What could be lent	Verdict	Why
The transform and its estimated cofactor	**Always**	A property of the panel derived from the data, not a product of the declaration. An unsupervised tool assumes a transform; cyRAVEN estimates one. Lending it costs nothing and removes an arbitrary constant
Per-sample thresholds	**Under `--maybe-learn`**	The single largest gain. See section 3
Staining QC verdicts	**Under `--maybe-learn`, as a column**	Recorded, never used to exclude. See section 4
Batch/group confounding verdict	**Under `--maybe-learn`**	Without it explore reintroduces the exact failure the declared path refuses. See section 5
Group labels	**Always**	They come from the sample sheet, not from the analysis. Withholding them would mean explore could not test anything
The parent gate	**Never**	Explore exists to look *outside* it. Applying it would make explore a re-clustering of the declared result
The population definitions	**Never**	That is the thing being checked

2.2 Explore -> declared

What could be lent	Verdict	Why
Populations spanning several clusters	**Under `--maybe-learn`**, as `spec_gaps.csv`	The finding the declared path structurally cannot make about itself
Clusters no population covers	**Under `--maybe-learn`**, same file	Names what the specification missed
Cluster-derived population definitions, adopted automatically	**Never**	Circular. See section 6
Re-gating, re-thresholding, or altering any declared frequency	**Never**	A run whose numbers were changed by a clustering is no longer the run its manifest describes

The asymmetry is deliberate. Explore may **describe** the declared analysis. It may never **alter** it.

3. Per-sample thresholds: the exchange that matters most

A standalone clusterer names a cluster by pooling every cell, computing a median per marker, and putting a colour scale on it. Three things are wrong with that on multi-sample data, and all three are things cyRAVEN has already solved for the declared path:

1. **The pooled median is not a threshold.** It is the middle of the distribution, which for a marker where 80% of cells are positive sits inside the positive population.
2. **It ignores staining variation between samples.** A bright sample and a dim one are compared against one number, which is precisely the systematic bias that per-sample gating exists to remove.
3. **It produces a colour, not a call.** "Reddish for CD19" is not a measurement.

Under `--maybe-learn`, explore reads `thresholds_used.csv` and calls each cell against **its own sample's cut**. A cluster's phenotype becomes:

```
CD19+ HLA-DR+ CD3- CD14-
```

where every + means *this fraction of the cluster's cells were above the threshold derived in their own sample*, recorded in `frac_pos.<marker>` columns. A marker the cluster is mixed for is **omitted** rather than called either way, so the phenotype string admits what it does not know.

That is a claim no standalone clusterer can make, and it is not a clever algorithm. It is the reuse of work the declared path already did.

4. Two things explore can do that the declared path cannot

It keeps the samples staining QC excluded. A sample with no resolvable CD45 mode has no usable parent gate, so every declared percentage from it would be a fraction of an arbitrary slice of the scatter plot. The declared path must drop it. Explore does not depend on that gate, so those samples still contribute, and `staining_qc_verdict` is a *column* rather than a filter. On the KMT2 cohort analysed in this project that is three donors the declared path had to discard and explore can still describe.

It looks outside the parent gate. Anything the CD45 gate excluded is invisible to the declared analysis by construction. Explore embeds it, which is how a population that fails the parent gate can still be seen.

5. Carrying the confounding verdict

The most likely way explore mode could damage this package is by becoming a p-value generator that ignores everything the declared path established.

`explore_cluster_stats.csv` therefore carries `batch_group_cramers_v` and `batch_group_verdict` as columns of the results table itself, not as a note elsewhere. On a cohort where batch and group are confounded, a $q < 0.05$ on a cluster means exactly as little as a $q < 0.05$ on a declared population, and the table says so on the same row.

Explore's statistics are also donor-level, using the same `stats_group_comparison()` the declared path uses: one value per sample, Kruskal-Wallis omnibus, Wilcoxon pairwise, Cliff's delta, Benjamini-Hochberg. Not a reimplementation: the same function, so the two cannot drift apart.

6. Why the draft specification is never adopted automatically

`explore_suggested_spec.yaml` turns clusters into declarable population definitions. The obvious next step, feeding it straight back into the declared pipeline, is refused, and this is a central design decision in the feature.

cyRAVEN's argument is that populations are declared **before** the data are examined, and six diagnostics then try to falsify that declaration. Derive the specification from the same data and the diagnostics become circular:

- **Phenotype concordance** asks whether a population expresses the markers its definition demands. Trivially yes, if the definition was read off those cells.
- **Gate-cluster concordance** compares declared gates against unsupervised clusters. If the gates came from those clusters, it compares a thing to itself.

There is a sharper version. Explore's clustering is **group-blind**. It never sees cohort labels, so deriving a definition from cluster *phenotypes* is a mild form of double-dipping. But `explore_cluster_stats.csv` **does** test by group. A user who reads that table, sees which clusters differ, and declares those as populations has selected on the outcome, and every subsequent p-value is invalid.

The defensible uses, ranked:

Workflow	Verdict
Explore on cohort A -> curate -> declared run on **cohort B**	Clean. The intended loop
Explore -> curate -> declared run on the **same** cohort, reporting frequencies	Acceptable; state that the spec was data-derived
Explore -> read the group statistics -> declare the clusters that differed	**Invalid.** Selection on the outcome

Hence: a YAML file whose header says it is a suggestion, requiring a human edit, with no flag anywhere that adopts it.

7. Choices inherited rather than invented

Three decisions were taken from the unsupervised literature because they are better than what a supervised tool would reach for.

Cluster everything, including scatter and viability. No lineage preference. This is what lets the QC gate find debris (leukocyte marker low), dead cells (viability high) and granulocytes, which no antibody in most panels identifies. The cost is real: some clusters split on scatter rather than lineage. `--explore-markers` overrides it.

Gate on clusters, not events. Cluster coarsely first, then judge *whole clusters* by their median profile, so no per-event cutoff is invented. The units judged are the SOM nodes rather than the merged clusters, because a merge can join debris to the CD45-dim cells beside it. The debris and dead calls come from 2-means on the per-node medians when nothing better is available, and from per-sample thresholds when it is. Both are recorded in the `basis` column of `explore_qc_clusters.csv`.

Refuse a gate that keeps almost nothing. Below `--explore-min-retained` (default 5%), the gate is ignored and the run flagged: at that point the markers are likelier mis-named than the data bad.

One decision goes the other way. **Cells are re-equalised per sample after the gate.** cyCONDOR does not do this, and its faceted panels then differ in density for a reason that is group size rather than biology. That trap was documented while running it during this project. Equalising costs cells and buys panels that can honestly be read side by side.

8. Where this leaves cyRAVEN

Compared with FlowSOM, Phenograph, cyCONDOR, diffcyt and CATALYST:

Capability	cyRAVEN	Also in
Per-sample gate derivation	yes	none of them
Uncertainty carried to every frequency	yes	none of them
Detection limits per population	yes	none of them
Donor-level inference by construction	yes	diffcyt; possible but not enforced in cyCONDOR and CATALYST
Refuses batch correction when confounded	yes	none of them
Unsupervised discovery	yes	all of them
Clusters named from per-sample thresholds	yes	none of them
Cluster turned into gate geometry	yes	none of them
Gating-ML 2.0 export	yes	none of them
Empirical-Bayes moderation across clusters	no	diffcyt
Trajectory inference	no	cyCONDOR
Cluster count found from the data	no	Phenograph, and cyCONDOR through it. FlowSOM and CATALYST are given the count

The gap this feature closes is the "unsupervised discovery" row. The row that remains distinctive is **clusters named from per-sample thresholds**, which requires both halves in one tool and is why bolting FlowSOM onto a supervised pipeline is not the same thing.

What is deliberately not adopted

diffcyt's empirical-Bayes moderation as the primary test (edgeR, limma-voom, GLMM). It borrows strength across clusters via a shared dispersion prior, which pays off at mass-cytometry marker counts with hundreds of clusters. On a declared panel of a dozen populations there is little to borrow, so the rank tests on one value per donor stay primary. The edgeR form, with diffcyt-DA-edgeR's defaults applied to each sample's exclusive-label counts, is available beside them with `--edger`; limma-voom and the GLMM are not implemented.

A data-determined cluster count. Phenograph and consensus methods choose *k* from the data. `--explore-k` is a setting, and `explore_provenance.csv` says so, because a *k* chosen from the data is one more thing the significance count depends on.

Trajectory inference. Diffusion maps and pseudotime need a differentiation continuum the design supports. On cross-sectional human cohorts of 10 to 40 samples it produces a picture, not a measurement.

9. The claim worth testing

If there is a methods paper here, the claim is not "cyRAVEN clusters cells". It is:

> Threshold placement uncertainty, derived per sample and propagated to reported frequencies, predicts the between-operator variance that manual gating produces, and that populations whose propagated uncertainty is large are the ones where published cohorts disagree.

That is falsifiable and, as far as this project's reading goes, untested. It would need the same files gated by several operators, with the propagated uncertainty compared against the observed spread of their results. The inter-laboratory data to do it exists in the literature this package already cites.

The second claim, weaker but easier to support: **a refusal rule is a feature.** A tool that declines to run batch correction above a measured Cramér's *V* between batch and group, and says why, produces fewer results and more defensible ones. That one needs a demonstration rather than a benchmark: a cohort where correction would have produced a clean, wrong answer.

2 About flow cytometry

What an event is, what a cut is, why `SSC-A` sits in a population definition beside two antibodies, and what the `-A` on a channel name means.

Every other chapter assumes this one. Nothing here is specific to cyRAVEN, so skip it if you already work with flow data.

Every other page here assumes you already know what a cut is, why `SSC-A` sits in a population definition next to two antibodies, and what the `-A` on the end of a channel name means. This page assumes none of it.

Read it if you are about to write a `populations:` block and the words `above` and `below` are doing more work than you can follow.

1. One cell, one row

A cytometer does not look at your sample as a whole. It squeezes it into a stream so narrow the cells have to queue up single file, then fires a laser at the stream. Each cell crosses the beam alone, one at a time, thousands per second.

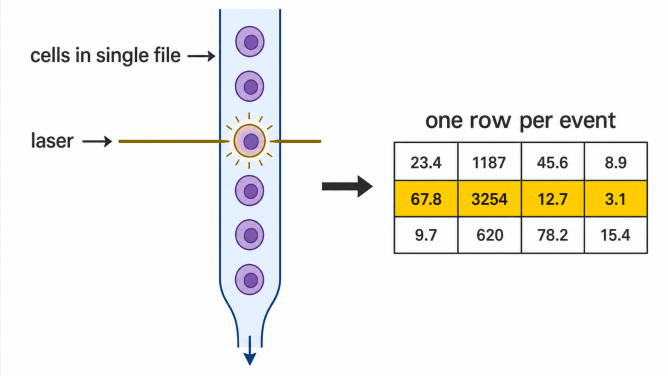


Figure 2.1: Cells queueing single file through a stream, one crossing the laser at a time, each crossing producing one row of numbers

Every crossing is recorded as one row of numbers. That row is one cell. A typical file holds a few hundred thousand rows, and the whole analysis is sorting those rows into groups.

In cytometry the row is called an **event** rather than a cell, and the word is chosen carefully. An event is "something crossed the beam". Usually that is one cell. It can also be two cells stuck together, or a fragment of a dead one. Part of the pipeline exists to throw those out, which is section 7.

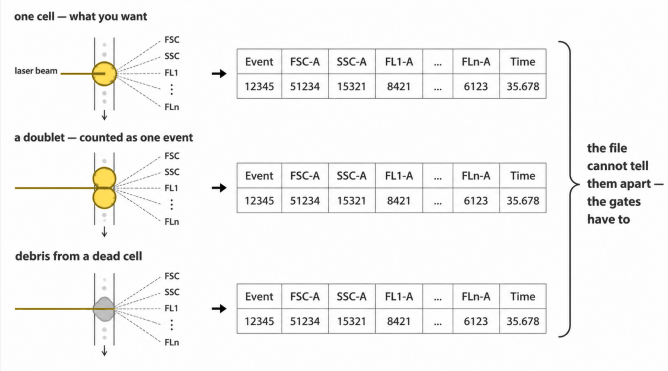


Figure 2.2: A single cell, a doublet and a fragment of debris all cross the beam and all produce one identical-looking row

2. Why cells glow

The numbers in each row come from two places, and it is worth keeping them apart in your head.

Some are physical. The cell blocks and scatters laser light whatever you do to it, so those measurements come free. That is section 5.

The rest come from staining. You add antibodies that stick only to one specific protein, each antibody carrying a fluorescent dye. A cell carrying that protein collects antibodies and glows; a cell without it does not. The protein is called a **marker**, and each marker gets its own channel of measurement.

So a row of numbers is one cell's answer to "how big, how grainy, and how much of each of these proteins are you carrying".

3. The cut

For each marker you get one number per cell: how brightly that cell glowed. Count how many cells sit at each brightness and you usually get two humps.

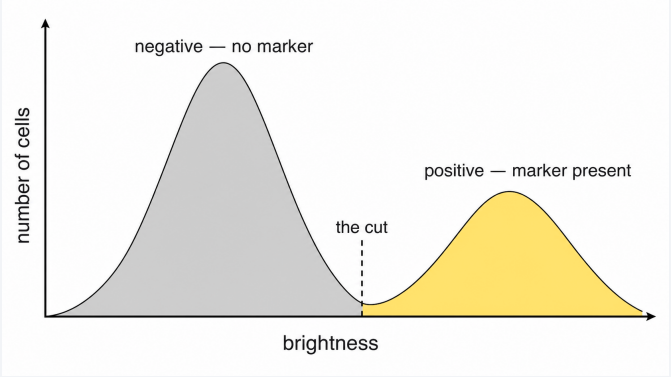


Figure 2.3: A bimodal brightness distribution with the cut placed in the dip between the negative and positive humps

The left hump is cells without the marker, glowing faintly from background. The right hump is cells carrying it. The cut goes in the dip between them, because that is where the fewest cells are, so a small error in placing it misclassifies the fewest cells.

above means right of the cut. `below` means left of it.

Written by hand these are `CD15+` and `CD15-`, said out loud as "CD15 positive" and "CD15 negative".

Why the cut is not a number you type

This is the whole reason cyRAVEN exists, so it is worth a paragraph.

Stain two tubes on different days and both humps shift, because staining strength varies with the reagent lot, the incubation, the instrument settings and the sample itself. The dip shifts with them.

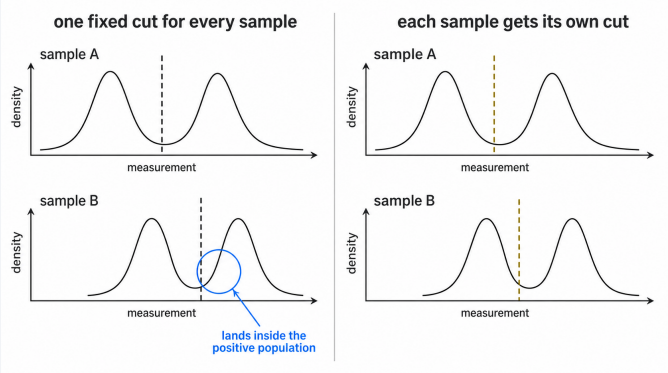


Figure 2.4: One fixed cut applied to two samples lands inside a population in the second; a per-sample cut sits in each sample's own dip

A cut copied from the first tube lands inside a hump in the second and slices a real population in half. cyRAVEN finds the dip separately in every sample, so the cut moves with the data. `above` means "bright for this tube", never "above 1500". The values actually used are written to `thresholds_used.csv`.

When no dip can be found, cyRAVEN does not invent one. It places the cut just beyond the spread of the sample's main peak (a tail gate: above a negative peak, below a positive one), records that in `thresholds_used.csv` as `tailgate`, and draws the line in red in `gating_qc.png` so you can see which cuts to distrust. A fixed percentile, `quantile_fallback`, is the last resort where even a tail gate cannot be placed, because it calls the same share of cells positive whatever the sample holds.

4. The three directions

Direction	Meaning	Written by hand as
`above`	brighter than the cut, the marker is present	CD15+
`below`	dimmer than the cut, the marker is absent	CD15-
`intermediate`	between the cut and a second, higher cut	CD14 dim, or mid

All the markers listed for a population must hold **at the same time**.

`intermediate` needs two cuts

Some markers have three levels, not two. Monocytes are the standard case: CD14 bright, CD14 mid, CD14 dim. One cut cannot split three groups, so `intermediate` uses a second cut above the first, and the cell has to sit between them.

That second cut has to be derivable from the data. If cyRAVEN cannot find a dip separating mid from bright, it does not guess. The population is marked unavailable with the reason attached, rather than reporting a number resting on an invented boundary.

5. A worked example

```
populations:
  Granulocytes:
    CD45: above   # a white blood cell
    SSC-A: above  # grainy inside
    CD15: above   # carries CD15
```

Read as one sentence: count a cell as a granulocyte if it is above the CD45 cut **and** above the SSC-A cut **and** above the CD15 cut.

`SSC-A` is the odd one out, and explaining it is the rest of this page.

6. FSC and SSC

These are the measurements that need no stain. They are laser light bouncing off the cell, so every cell has them whatever you stained for.

FSC, forward scatter, is light that carries on roughly straight ahead past the cell. Bigger cell, more light pushed forward. Read it as **size**.

SSC, side scatter, is light knocked off sideways at about 90 degrees. Smooth interiors deflect little. Interiors packed with granules and folded membranes deflect a lot. Read it as **internal graininess**.

Together they separate the main blood cell types before any antibody is involved:

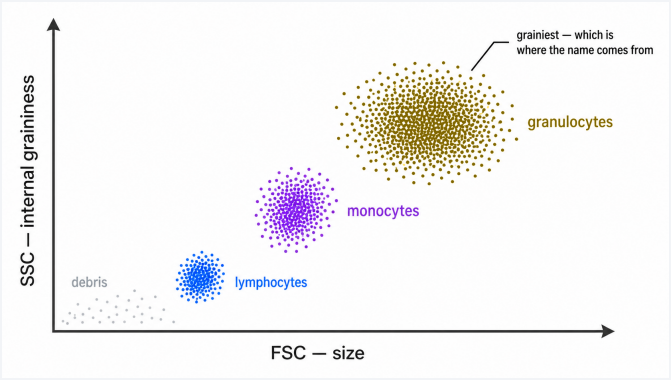


Figure 2.5: Scatter of FSC against SSC with debris, lymphocytes, monocytes and granulocytes occupying separate regions

That is why `SSC-A: above` sits in the granulocyte definition next to two real antibodies. It is not a stain. It is the physical fact that granulocytes are the grainiest white cells, which is where the name comes from, so "grainy" is part of the definition rather than a technicality.

7. The `-A`, `-H` and `-W` suffixes

As a cell crosses the beam the signal climbs, peaks and falls. That trace is a **pulse**, and the instrument records three numbers from it.

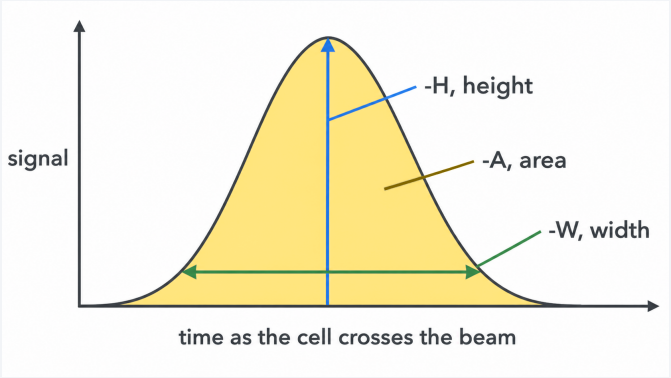


Figure 2.6: One pulse annotated with its area, its height and its width

Height is the peak, **area** is the total underneath, **width** is the duration. The suffix on a channel name says which one you are looking at, so `FSC-A` is forward scatter area and `FSC-H` is forward scatter height.

For a single cell, area and height rise and fall together. For two cells stuck in a clump the pulse lasts twice as long, so the area roughly doubles while the height barely changes. That mismatch is the standard way to spot a clump.

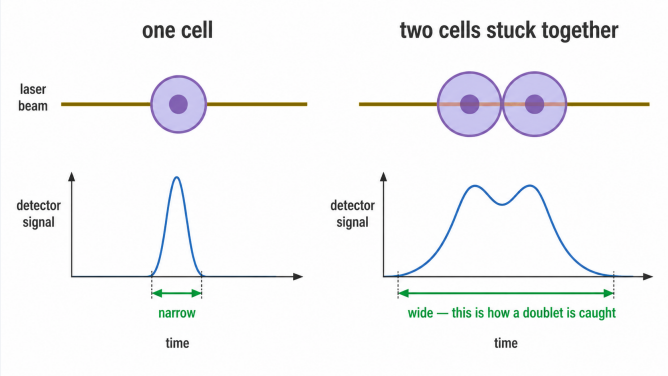


Figure 2.7: One cell gives a narrow pulse; two cells stuck together give a wider one, which is how a doublet is caught

It matters more than it sounds. A clump of a T cell and a B cell stuck together is recorded as one event positive for both CD3 and CD19, which reads as a cell type that does not exist. Left in, clumps manufacture rare populations.

cyRAVEN takes the FSC-H to FSC-A ratio for every cell and keeps those within 3 MAD of that sample's median. Events off that line are doublets and are dropped before anything is counted.

8. How cyRAVEN uses each channel

Channel	Used for
`FSC-A`	Debris removal, at the dip in log10 FSC-A, below which events are too small to be whole cells
`FSC-H` with `FSC-A`	The singlet gate in section 7
`SSC-A`	A phenotype axis you can name in `populations:`, as in the granulocyte example

One deliberate exception. Scatter is excluded from the UMAP feature set. Scatter values span a far wider numeric range than stained markers, so leaving them in makes the map arrange cells mostly by size and graininess and drowns out the markers you actually stained for. Scatter decides what counts as a real single cell, then steps aside.

9. Declare the big populations first

Populations are not nested in the config file, but they share markers, and that is the trap.

Declare `T cells` as `CD3: above`, then subsets like `CD4 T cells` as `CD3: above, CD4: above, CD8: below`. Every subset re-uses the CD3 cut. If that cut is misplaced it is not one wrong number. It is wrong in the parent and in every subset beneath it, all shifted the same way, and all still looking perfectly reasonable in the output table.

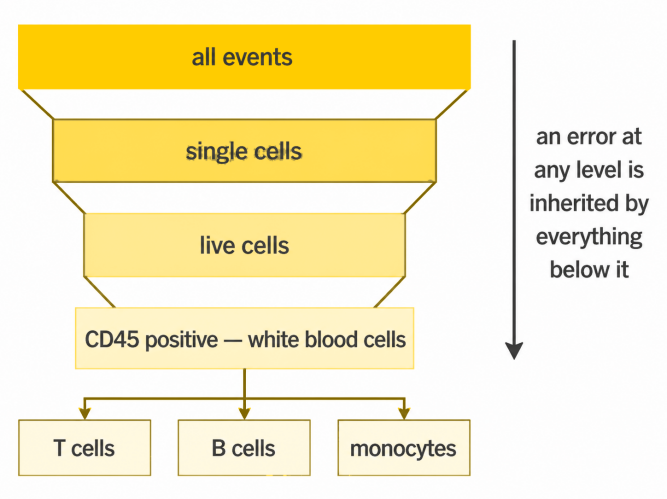


Figure 2.8: A funnel narrowing from all events through single cells and live cells to CD45 positive and then to individual lineages

So score the broad lineage populations first and check they are plausible: granulocytes at roughly half of blood leukocytes rather than 4% or 95%. Once the lineage numbers make sense, the cuts feeding the subsets are trustworthy.

Checking a rare subset first tells you nothing, because at that size you cannot tell a real rare population from a parent cut that drifted.

3 Get started

From nothing to a finished run in four commands, then how to describe your own cohort.

Two habits save the most time:

- **Run `--check` first.** It reads only the file headers and stops, so a misspelled marker costs seconds instead of a whole run.
- **Open `gating_qc.png` before using a number.** It draws each cut on the data it came from. A cut on the side of a peak shows there, and in no table.

Setup

You need Docker.

```
docker pull bhagesh/cyraven:1.0.0
docker tag bhagesh/cyraven:1.0.0 \
  cyraven:1.0.0
```

The image fixes R 4.4.3, a dated CRAN snapshot and Bioconductor 3.20. Gates sit at density minima and the UMAP is stochastic, so other package versions can move both slightly.

To build the image yourself instead, for example after changing the source:

```
git clone \
  https://github.com/bhagesh-h/cyRAVEN
cd cyRAVEN
docker build -t cyraven:1.0.0 \
  -f inst/scripts/Dockerfile .
```

The first build compiles every R dependency and takes tens of minutes.

To install without Docker, see the README.

Quick start

1. Write the demo data. 35 FCS files from five transplant recipients, a sample sheet and a config. They ship inside the image, so nothing is downloaded.

```
docker run --rm -v "$PWD:/w" \
  cyraven:1.0.0 --demo /w/demo
```

2. Check the inputs. It should end with `no problems found`.

```
docker run --rm \
  -v "$PWD/demo:/data:ro" \
  -v "$PWD/results:/results" \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/panel.yaml \
  --group-column cohort \
  --reference-group "GvHD grade 1" \
  --outdir /results \
  --check
```

3. Run it. A few minutes.

```
docker run --rm \
  -v "$PWD/demo:/data:ro" \
  -v "$PWD/results:/results" \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/panel.yaml \
  --group-column cohort \
  --reference-group "GvHD grade 1" \
  --outdir /results
```

4. Open `results/report.html`. Every figure and table is in that one file, in the order to read them.

Paths in the flags are paths inside the container, and `--outdir` must be in a mounted folder or the results vanish when the container exits. On Windows, run these in Git Bash or WSL with `MSYS_NO_PATHCONV=1` in front of each command. In PowerShell, end each continued line with a backtick instead of \, and write `${PWD}` for `$PWD`.

About this cohort: both groups are transplant recipients. The comparison is GvHD grade 1 against grade 3, not disease against health. The groups are small, one patient against four.

Your own data

Two files besides the FCS folder:

File	Holds
Sample sheet, `samples.csv`	One row per FCS file: sample, patient, group, timepoint, batch
Config, `analysis.yaml`	One entry per population: its marker rule, e.g. `CD3: above`

Start both from a template:

```
docker run --rm -v "$PWD/data:/data" \
  cyraven:1.0.0 \
  --dir /data/fcs --recursive \
  --write-samples /data/samples.csv
```

```
docker run --rm -v "$PWD/data:/data" \
  cyraven:1.0.0 \
  --write-template /data/analysis.yaml
```

Fill in the sheet and edit the `populations:` section of the config. Run with `--check` until it reports no problems, then run without it. The full sequence is in A first run on a new cohort. The rest of this page explains each file and every option.

Inputs: the sample sheet and the config

Anything that varies per sample belongs in the sheet; anything that is one decision for the whole study belongs in the config. A study-wide choice put in the sheet would repeat on every row, and the rows could then disagree about it.

From R, the two templates are `write_config_template("analysis.yaml")` and `system.file("examples", "samples_template.csv", package = "cyRAVEN")`.

1. Build the sheet

Do not write it from scratch. Generate one covering every file in the input directory, then fill in the columns only you know:

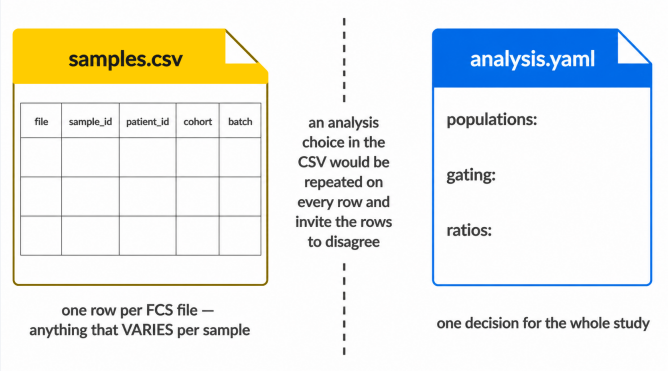


Figure 3.1: The sample sheet holds what varies per sample; the config holds one decision for the whole study

```
docker run --rm \
  -v "$PWD/data:/data" \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --write-samples /data/samples.csv
```

Every acquisition gets a row with its filename and a derived identifier, so the sheet accounts for every file before you touch it. A file the sheet does not cover is a fatal error rather than a guess: inferring which patient a file belongs to from plate order silently mislabels patients, and a mislabelled patient is not visible in any output.

Then check it, which costs seconds because it reads FCS headers and not events:

```
docker run --rm \
  -v "$PWD/data:/data:ro" \
  -v "$PWD/results:/results" \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --outdir /results \
  --check
```

`--check` reports the markers it resolved from the files, the specification entries matching none of them, whether the sheet covers every file, the levels of the group column and their sizes, the number of batches, and the study variables available. It analyses nothing and writes nothing. Run it after every edit to the sheet; a marker-name mismatch found here costs a second, and found during a run costs however long the run had been going.

2. Columns of the sheet

Only `file` is required. Everything else is optional, and a column that is absent disables what depends on it.

2.1 Which file this is

Column	Meaning
`file`	The FCS filename. Matched on basename, so a file moved between subdirectories still matches. Required.
`sample_id`	The identifier used in every output. Derived from the filename when absent.
`well`	Plate well, used as the identifier when `sample_id` is absent.
`patient_id`	The subject. Several rows may share one, which is what makes repeat acquisitions and longitudinal designs work.
`panel`	Forces a panel assignment. Panels are fingerprinted from the channels otherwise.
`timepoint`	Visit or draw.
`is_control`	`TRUE` for a control tube. Controls are read and used as references but excluded from between-group tests. Setting it `FALSE` is not the same as leaving it out: `FALSE` asserts that the file is a biological sample, and the assertion matters. See below.

Say `FALSE` rather than saying nothing An absent `is_control` column means "unknown", not "no controls here". The distinction has teeth.

A control tube supplies the reference distribution for markers that resolve no density minimum, and the cut is then the 99.5th percentile of the control. That is correct when the reference really is unstained. When it is a stained biological sample, the cut lands at the top of a real distribution and the population beneath it all but disappears.

Up to version 1.0.0 a sample that failed staining QC could be promoted to that reference on a sheet that named no controls, on the reasoning that an unlabelled tube might be one. On a cohort of twelve patient samples with no controls at all, that put T cells at 0.034% of CD45+ where the correct figure was over a hundred times higher, with no error and no warning.

cyRAVEN now believes a control only when the sheet declares one, and says so in the log when it declines to promote a sample. Two habits follow from that:

- Write `is_control` on every row, `FALSE` for samples and `TRUE` for controls. `--write-samples` does this for you.
- If most rows of `thresholds_used.csv` read `control_q995` in the `source` column, check which sample supplied the reference before reading any frequency. | `fmo_for` | Comma-separated markers this file is the fluorescence-minus-one control for. | | `control_group` | Confines a control to the samples sharing its value, since a reagent lot changes between batches. |

2.2 Whose it is

These describe the *subject*, so they repeat on every row of that subject.

`patient_id`, `date_of_birth`, `sex`, `age_years`, `height_cm`, `weight_kg`, `infection_focus`, `cohort`, `wbc_per_ul`.

They are recognised by their canonical name and by the spellings common in clinical exports, in English and German, so a sheet headed `Geschlecht` or `Patient ID` resolves without being translated first. Values are coerced the same way whichever route supplies them: decimal commas, dates whose day and month order is resolved from the column as a whole, and sex and clinical vocabulary translated through a dictionary that passes unknown values through and reports them rather than mangling them. Extend the dictionary under `metadata:` in the YAML.

Age is derived from `date_of_birth` when it is missing, against `--reference-date`. Set that to a fixed study date, or ages change as the calendar moves and two runs of the same data disagree.

A subject's rows must agree. This is the one hazard the one-file shape introduces that separate tables did not have: with a row per file, a patient with three acquisitions carries three copies of their sex. If two disagree, the run stops and names every conflict as `subject`

`/ column: value vs value`. There is no defensible way to pick one, and picking silently is how a cohort acquires a patient who is both male and female depending on which tube you look at.

A blank is not a disagreement. It is filled from the rows that have a value.

2.3 What was measured externally

Any column named `count.<Population>` is read as an externally measured absolute count for that population, from bead-based or volumetric counting.

```
file,sample_id,count.Monocytes
HC-01.fcs,HC-01,420
```

A frequency is a proportion of a parent gate, so it moves whenever any other population moves. An absolute concentration does not, which is why these are worth carrying when the laboratory measured them. Blanks are skipped rather than read as zero. Units are `cells/uL` unless `samples: count_unit: cells/mL` says otherwise, in which case values are divided by 1000.

2.4 Everything else

Any other column is kept as a study variable and can be named by `--group-column` or `--batch-column`, or used as a covariate. Acquisition date, treatment arm, instrument, operator and site are all ordinary columns; none of them needs a reserved name. `--check` lists what it found.

3. The config

`populations:` is the only required section. Everything else takes a documented default when omitted, and the default in force is stated in the run log.

Populations, functional blocks and ratios are covered in the the gating specification; thresholds, colours and metadata dictionaries in the Running cyRAVEN. One section belongs here, because it is about the sheet:

```
samples:
  group_column: cohort
  batch_column: acquisition_date
  count_unit: cells/uL
```

Naming the roles here means the two files specify the run on their own, with no further flags. A flag still wins where both are given: the config is the study's standing choice, the flag is this run's.

4. The older three-file format

`--sample-map`, `--patient-table` and `--absolute-counts` still work exactly as before and are not deprecated. Runs made with them remain reproducible.

They cannot be combined with `--samples`. The sheet carries the same facts, and two sources of truth for one fact is what this format removes.

The sheet is a different way to supply the same facts, not a different analysis. The reader splits it into the same three structures the pipeline always consumed and applies the same coercions by calling the same code. A test runs one study both ways and compares every output.

Which to use:

Situation	Use
A new study	The sheet. One key, one file, no joins.
An existing pipeline built on the three files	Leave it. Nothing has changed.
Counts exported by the counting instrument as its own sheet	`--absolute-counts` reads that file's own layout; see the Every output file. Fold it into the sheet only if transcribing it is easy.

5. Worked example

The full sequence for a new cohort, with the two edits that come between the commands, is in A first run on a new cohort under Commands.

6. When it goes wrong

A run that fails writes `report.html` anyway, with the error, the stage it reached, the log leading up to it, what the error means for the data, and the next action. Everything produced before the failure is embedded below the diagnosis, because that partial output is usually where the evidence is.

The errors this format can raise, and what each means:

Message	Cause
`these input files are not in the sample sheet`	The sheet has no row for a file. Add rows, or regenerate with `--write-samples`.
`conflicting values for the same subject`	Two rows of one subject disagree. The message names each conflict.
`duplicate file entries`	Two rows name the same file.
`must contain a 'file' column`	The one required column is missing.
`--samples supersedes ...`	The sheet was combined with a flag it replaces.

Every population reported `UNAVAILABLE`, or an empty frequency table, is almost usually a marker name that does not match `$PnS` exactly. `--check` lists every marker the files carry and names the specification entries matching none of them.

The gating specification

Cell identification in cyRAVEN proceeds in three stages: exclusion of non-cellular and non-viable events, placement of per-sample marker thresholds, and evaluation of a declarative population specification against those thresholds. This article documents each stage and the parameters governing it.

1. Hierarchy

Four sequential gates precede any marker evaluation. Each is derived from the data rather than transferred between samples.

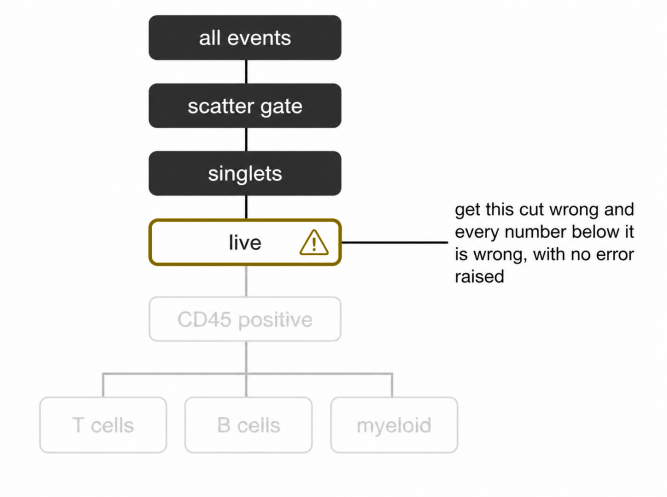


Figure 3.2: A gate tree in which one wrong cut fades every population beneath it

1.1 Scatter. The lower forward-scatter bound is placed at the deepest minimum of the log10 FSC-A density. Sub-cellular debris forms a

distinct low-scatter mode separated from intact cells; the minimum between them is the boundary.

1.2 Singlets. Coincident events traverse the interrogation point over a longer interval than single cells, producing a pulse whose area is inflated relative to its height, so they have a low FSC-H:FSC-A ratio. Events are retained above median − *k*·MAD of that ratio, computed inside the scatter gate, with *k* set by `--singlet-mad-k` (default 3). The gate is one-sided: a high ratio is no sign of a doublet.

1.3 Viability. Amine-reactive viability dyes penetrate only cells whose membrane integrity has been lost. Dead cells are excluded at the density minimum of the dye channel above the live cells' mode, or at a tail gate above that mode where there is none. Panels without a viability marker skip this gate, and the omission is logged. Each sample's live share of single cells is reported in `staining_qc.csv`; `--min-live-pct` fails samples below the assay's minimum. Where the dye also has a minimum below its dominant mode, the share below it is reported too (`pct_live_if_dominant_dead`): the live share if the dominant mode is the dead cells, which the gate cannot tell from a second live mode near zero.

1.4 CD45. Leukocytes are selected on CD45, which is expressed across haematopoietic lineages and absent from erythrocytes and platelets. Where CD45 is not in the panel, all viable events become the parent and a warning is issued, since percent-of-leukocytes and percent-of-viable are not interchangeable denominators.

All subsequent population frequencies are expressed as a percentage of the CD45+ parent.

2. Thresholding

2.1 Rationale

Staining index varies between samples through reagent lot, fluorochrome degradation, time between staining and acquisition, and cellular autofluorescence. A threshold transferred between samples is therefore mis-specified in both directions: conservative in dim samples, permissive in bright ones. The resulting bias is systematic and correlated with acquisition order, which is the configuration most likely to be confounded with study group.

2.2 Implementation

For each marker, within each sample, the density of cells passing the parent gate is evaluated and the threshold placed at the minimum separating the negative from the positive mode. `density_valley()` returns `NA` where the distribution is unimodal. `resolve_threshold()` then attempts an unstained control if one is declared in the sample map, and falls back to a fixed quantile otherwise.

2.3 Interpretation

`thresholds_used.csv` records one row per sample and marker. Two columns determine how far a downstream frequency can be trusted.

`threshold` is the value applied.

`source` is `valley` where a density minimum was resolved and `tailgate` where none existed: the cut then sits 3.5 robust standard deviations above the sample's dominant mode. The spread is the smaller of the MAD of the mirrored half below the mode and the mode's half-width at half maximum on the cut side, so neither a positive shoulder nor a second negative hump (monocytes and lymphocytes on PBMC) widens it. openCyto's `gate_tail` handles the same single-mode case by cutting where the density's slope flattens (Finak et al. 2014); the two agree on a normal negative. The rule assumes that mode is the negative population. A single mode cannot confirm it, since autofluorescence and spillover spread lift negative modes well above zero signal too, so a marker carried by most of its parent, with no density minimum, needs a cut declared under `thresholds:`, a `parent:` in which it separates, or the side declared as openCyto's `side` argument does: `thresholds: CD127: {tail: lower}` puts the tail gate below the dominant mode, taken to be the positive population. `quantile_fallback`, a fixed percentile, remains only where no tail gate can be placed. Either indicates that the marker did not separate positive from negative events in that sample. Frequencies derived from a fallback threshold are not invalid, but they carry no evidence of separation. A marker falling back across the majority of samples is not resolving in that panel, and no gating strategy will recover it.

Each threshold is compared with the median of the same marker's cuts in the other samples of its panel (controls and staining-QC failures are compared, not used as peers): `thresholds_used.csv` carries that `peer_median`, the robust *z* of the deviation and `scale_outlier` (*z* above 3.5). Where any cut is an outlier, `threshold_scale_qc.csv` summarises each panel and marker: the median, the range, and how many cuts are outliers or fallbacks. Flagged thresholds are the first candidates for manual review.

2.4 Precision

`source` records which rule placed a threshold: a density valley, a tail gate where the density showed none, a control tube, a peer median, a manual or configured value, or the quantile fallback. It does not say how well determined the value is, and two cuts both recorded as `valley` can differ by an order of magnitude in that respect: one sitting in a wide, empty gap between well-separated modes, the other on a shallow dip that a slightly different histogram would not have found.

`threshold_uncertainty.csv` supplies the missing quantity. Each cut is re-derived from resamples of the events it was computed on, giving the component that comes from having counted a finite number of cells, and again across the settings `density_valley()` takes, giving the component that comes from the choices this package makes on the analyst's behalf. The two combine in quadrature.

`bootstrap_valley_rate` is the more direct reading of the two. A cut recovered in every resample marks a real boundary; one recovered in half of them was found by histogram noise that happened to clear the relative-depth rule, and a wide interval understates how little is there.

`uncertainty_budget.csv` carries the consequence through to the populations. Each marker a population reads contributes a term, as does the CD45 parent threshold, which enters every population because it fixes the denominator. The Diagnostics covers how to read the result.

2.4a Reference controls

`resolve_threshold()` falls back to a control distribution when no density minimum exists. Which control that is decides what the resulting cut means.

An unstained tube, declared through `is_control`, shows where autofluorescence ends. It cannot show where a marker's background ends in a panel, because a stained sample's negative population sits wider than an unstained one under spillover from every other fluorochrome present.

A fluorescence-minus-one control is the same panel with one reagent omitted, so its distribution in that channel is the negative population under the spreading the real samples experience. Declare one through two optional sample-map columns: `fmo_for`, naming the markers the file controls for, and `control_group`, confining it to the batch it was acquired in. The resulting `source` is `fmo_q995` rather than `control_q995`.

`fmo_agreement.csv` is the reason to supply one rather than merely the means. It reports the distance between the derived cut and its FMO-anchored equivalent in units of that threshold's own uncertainty. Within about one, the two agree to the precision either can claim and the derived cut is corroborated by an independent experiment. Beyond about three they disagree by more than either can explain: a derived cut far above the FMO is discarding real signal, and one far below is calling spillover positive.

2.4b Overriding one sample's cut

A threshold flagged for review previously admitted two responses: accept it, or pin that marker for the whole run through `thresholds:`. The second corrects one tube by applying one number to every sample, which reintroduces the fixed-coordinate bias section 2.1 describes.

A `sample_overrides:` block corrects one sample and one marker:

```
sample_overrides:
  D07:
    CCR7:
      threshold: 2.15
      reason: >-
        valley sat inside the negative
        mode, see gating_qc.png
      set_by: "initials"
```

The resulting `source` is `manual`, distinct from `config`: the first records that a named person moved one cut for a stated reason, the second that the assay declares this cut everywhere. `thresholds_used.csv` gains `override_reason` and `override_by`, and the run manifest lists every override. An entry matching no sample or marker in the cohort is reported rather than silently ignored.

2.5 Sufficiency

Precision of placement and sufficiency of counting are separate guarantees, and a population can have one without the other. A cut through a wide empty gap is well determined however few cells sit beyond it.

`u_counting_pct_points` in `population_frequencies.csv` is what the frequency carries from the number of events behind it, computed as the Wilson half-width at one standard deviation. The ordinary binomial standard error is not used because it evaluates to zero when no events were observed, which asserts certainty about a population that was never seen.

`lod_pct` and `loq_pct` express the conventional twenty and fifty events as percentages of that sample's parent gate, and `detection` states which side of them the population falls. Both are properties of the acquisition rather than of the gating strategy: a population below the limit of quantification is not mis-gated, it is under-sampled, and no threshold placement recovers it.

Because the denominator is the parent-gate events this run saw, `--max-events-per-file` raises both limits in proportion. A population reported below the limit of a subsample may be perfectly well resolved in the full file.

None of this alters a threshold. The value in `thresholds_used.csv` is `density_valley()` at its defaults on the real events, with or without the analysis; the perturbation runs on copies.

2.5 Stability across runs

The scale check is against the other samples of the same run, which identifies one deviant tube among many sound ones. It is blind to a cohort that moved together, because the peer median moves with it.

`--write-baseline` records where an accepted run placed each threshold and how variable it was; `--baseline` measures a later run against that record. This is the check that survives a laser service, a reagent lot change or a year between acquisitions.

3. Specification

3.1 Syntax

Populations are declared in the `populations:` block of the `--config` YAML as conjunctions of marker directions.

```
populations:
  CD4 T cells:
    CD3: above
    CD4: above
    CD8: below
  Central memory CD4:
    CD3: above
    CD4: above
    CD8: below
    CCR7: above
    CD45RA: below
```

An event is assigned to a population when it satisfies every declared direction against that sample's thresholds. The first entry above is the conventional definition of a CD4 T cell, expressed in a form the scoring stage can evaluate.

3.2 Directions

`above` and `below` denote expression and its absence. If the terms in this section are unfamiliar, Flow cytometry for dummies covers events, cuts, scatter and the pulse suffixes from the beginning.

`intermediate` addresses markers with three resolvable levels. CD14 in monocyte subsetting is the canonical case: classical monocytes are CD14++, intermediate monocytes CD14+CD16+, and non-classical monocytes CD14dim. The upper bound of the intermediate interval is derived from a second density minimum within the positive fraction. Where no second minimum exists the population is reported UNAVAILABLE rather than being merged into the bright fraction.

`any_of` holds a disjunction: the population needs any one of its members. A member is one condition, or a named set of conditions that must all hold, so NK cells can be CD56+, or CD16+ without HLA-DR, which keeps CD16+ monocytes out without dropping activated HLA-DR+ NK cells:

```
NK cells:
  CD3: below
  CD14: below
  any_of:
    CD56: above
    CD16 without HLA-DR:
      CD16: above
      HLA-DR: below
```

A member whose markers are not all in the panel is dropped and the rest still gate; with none left the population is UNAVAILABLE.

`parent:` names another population, and the population's own markers are then cut on that parent's events instead of across every CD45+ cell, as a gating template derives each gate on its parent (Finak et al. 2014). A cut across all leukocytes sits wherever the dominant lineages put a valley; the consensus regulatory T cell gate is CD25+CD127lo within CD4+ T cells (Santegoets et al. 2015):

```
Tregs:
  parent: CD4 T cells
  CD25: above
  CD127: below
```

The parent's conditions are inherited, so they are not repeated. The cuts are written to `thresholds_within_parents.csv`; a parent holding fewer than 200 events in a sample gives no cut, and its children are UNAVAILABLE there rather than cut on a handful of cells.

3.3 Construction

1. Enumerate the panel. `thresholds_used.csv` from any run lists every resolved marker; the $PnS keyword of any file in the batch carries the same information.
2. Declare lineage-level populations first and confirm they score plausibly before adding subsets. An error in the CD3 gate propagates to every T cell subset beneath it.
3. Pass the file with `--config`. The same file optionally carries threshold overrides, colour assignments and metadata translations.
4. Read the run log. A population whose markers are not all present is reported UNAVAILABLE and scored as absent. An empty frequency table almost always indicates a nomenclature mismatch between the specification and $PnS rather than a biologically absent population.

`system.file("config", "config_cohorts.yaml", package = "cyRAVEN")` provides the file structure; replace the populations with those of the panel in use.

3.4 Default

The built-in specification describes a myeloid and lymphoid panel comprising CD14, CD16, CD19, CD56 with NKG2D, CD127 with CD25, and

the gamma-delta T cell receptor chains. A T cell panel of CD3, CD4, CD8, CD38, HLA-DR, CCR7 and CD45RA intersects it in three populations of fifteen. It is a worked example, not a default suitable for arbitrary panels.

3.5 Markers read inside a population

A marker can be used to define a population or measured inside one. The two are different questions, and the second is declared in `functional_blocks:`.

```
functional_blocks:
  CD33 on gated myeloid subsets:
    markers: [CD33]
    populations: [Granulocytes, Monocytes]
  activation:
    markers: [CD69, HLA-DR, CD25]
    require: CD3
  exhaustion:
    markers: [LAG-3, TIM-3, PD-1]
    exclude:
      - Classical monocytes
      - Non-classical monocytes
```

Each block reports its markers' median intensity and percent positive within the populations it is scoped to, and `functional_markers_stats.csv` tests them between groups exactly as abundance is tested. Scope is resolved in one of three ways, in this order of precedence: an explicit `populations:` list; `require:`, a marker whose direction must be `above` in a population's definition for that population to be included; or `exclude:`, a list omitted from an otherwise complete set. Expressing scope as a rule means adding a population to `populations:` scores it under the right blocks with no further edit.

`population_marker_mfi.csv` already reports every marker in every population. The point of a block is the scoping, and it has one requirement that matters: a marker must not be read inside a gate its own threshold helped draw. Testing CD15 within a CD15-positive population returns 100 percent in every sample, so the result has zero variance and an undefined p-value, and it measures the definition rather than the biology. Declaring `functional_blocks:` in the config replaces the built-in blocks entirely rather than adding to them; see `?default_functional_blocks` for what those are.

3.6 Ratios between populations

```
ratios:
  gran_lymph:
    label: "Granulocyte:lymphocyte ratio"
    numerator: Granulocytes
    denominator: Lymphocytes
```

A ratio of two declared populations is written to `population_ratios.csv` and tested with the same statistics as any abundance. It is declared rather than computed after the fact because when both populations are percentages of the same parent, the ratio is not recoverable from either frequency on its own once the parent has changed between samples.

There is no default. A ratio hard-coded to population names that a given panel does not contain would be silently meaningless, so the block is opt-in.

3.7 Provenance

The specification is not estimated from the data. A specification fitted to the samples against which it is subsequently tested cannot be falsified by them, and falsification is the function of the threshold drift, phenotype concordance and gate-cluster concordance outputs.

Data-driven proposal is available and kept separate. `--explain-clusters` derives two-marker gate geometry for any cluster the specification does not describe, with performance measured on held-out cells, and writes it to `cluster_gate_proposals.csv`. Promotion of a proposal to a named population is a manual decision, and the following run reads an edited specification.

4. Transformation

4.1 Scale

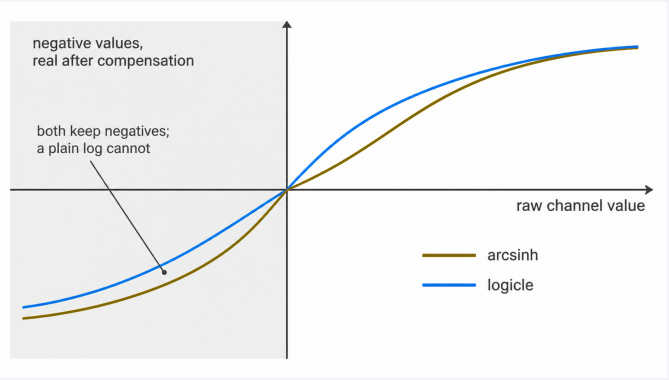


Figure 3.3: Arcsinh and logicle transforms compared, both keeping the negative values a plain log cannot

Fluorescence intensities span four to five decades. Untransformed, the negative population compresses against the axis and mode separation is not resolvable. Every threshold in section 2 is therefore placed on a transformed scale, and the choice of transform determines where it lands.

4.2 Arcsinh

The default. The cofactor governs the width of the quasi-linear region near zero and is estimated per panel by bisection until the interquartile range of the background distribution reaches a target, which puts it near 1.6 standard deviations of a negative population: just wider than the negatives, the condition under which a symmetric negative population stays one mode. This is a heuristic rather than a published criterion. The conventional value of 5 derives from mass cytometry and over-expands the background band on spectrally unmixed fluorescence data, so it is estimated rather than assumed. `derive_cofactor_pooled()` estimates across samples so that a single weakly stained file does not set the scale for the batch, and on the scatter-gated single cells rather than every event, so debris does not set the background (`--cofactor-from-all-events` restores the older estimate).

One cofactor serves every marker by default. Detectors differ manyfold in how wide their negatives are, and a cofactor per channel is the published recommendation for flow data (den Braanker et al. 2021; Azad et al. 2016). `--per-marker-cofactor` gives each marker its own, estimated by the same criterion on that marker's negative population, reconstructed from its values below zero. It is not the default. On a whole-blood cohort scored against manual gating it ranked samples no better (mean Spearman rho 0.64 for both), its worst-sample errors were larger and one more file failed staining QC, and on a PBMC cohort it emptied CD4 T cells in 3 of 30 samples and regulatory T cells and NKT cells in 6 each. Logicle, which fits its width per channel, scored the same as the shared cofactor on the whole-blood cohort and emptied regulatory T cells in 4 of the PBMC samples against 1. A small per-channel cofactor stretches the background differences between cell types, which leaves more markers without a density minimum.

4.3 Logicle

`--transform logicle` implements the automatic logicle rule with linearisation width $w = (m - \log10(t/|r|))/2$, where r is the fifth percentile of the negative values and t the top of the channel's declared range. Both transforms are linear near zero and both represent compensated negative values; arcsinh is defined for every real number and is a Gating-ML scale like logicle (Spidlen et al. 2015). What logicle adds is that the width of its linear region is fitted per channel to that channel's negatives (Parks et al. 2006), which arcsinh does only under `--per-marker-cofactor`.

Parameters are pooled across the panel. Per-file fitting, implemented by some tools, assigns each sample an independent scale and invalidates cross-sample comparison of medians.

4.4 Consequence

The transform is not a display parameter. On a seven-colour T cell panel, lineage-level populations agreed within 0.1% between transforms while CCR7 and CD45RA memory subsets differed by tens of percent.

The mechanism is recorded in `thresholds_used.csv`. Under arcsinh the CCR7 threshold resolved to a density minimum in some samples and to a quantile fallback in others, a difference of approximately 2.5 units on that scale. Under logicle a minimum was resolved in every sample. For dim markers without clear bimodality the transform determines whether a stable threshold exists at all, which makes the `source` column a prerequisite for interpreting any memory subset frequency.

`--transform none` bypasses transformation for pre-transformed input.

5. Compensation

Fluorochrome emission spectra overlap, so signal from one dye is registered in detectors assigned to others. Uncorrected, this produces apparent positivity in channels the cell does not express.

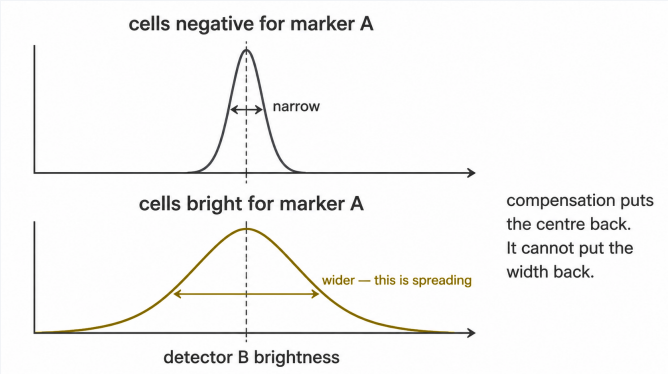


Figure 3.4: Compensation restores the centre of a distribution but cannot restore its width

The spillover matrix is determined at acquisition from single-stain controls and stored in the FCS keyword block. cyRAVEN applies it where present and reports its absence. Spectral instruments write unmixed data with no matrix, so `maybe_compensate()` detects rather than assumes, since applying compensation twice is as damaging as omitting it.

Matrix construction is out of scope. It belongs in acquisition software, where the single-stain controls can be inspected.

5.1 What compensation cannot undo

Subtracting a fluorochrome's expected contribution to another detector is correct on average. What the subtraction cannot remove is the photon-counting variance that came with the contribution, so a channel receiving spillover from a bright neighbour has a negative population that is correctly centred and abnormally wide. Two modes that were separable merge, the valley fills in, and section 2.3 reports a tail gate (`tailgate`) with no indication of why.

`spreading_receivers.csv` supplies the cause. For each ordered channel pair, the spread of the receiver's negative population is compared between cells negative and positive for the source; restricting to the receiver's own negatives is what makes this spreading rather than biology, since co-expression moves the positive cells and not the width of the negatives. `spreading_pairs.csv` holds the full pairwise ranking.

The finding to act on is the join between the two: a marker that both shows no density minimum in most samples and receives substantial spreading is reported as a panel design problem. No gating strategy recovers a cut that spreading has erased.

This is a ranking computed from the samples in hand, not a spillover spreading matrix. The published SSM is derived from single-stain controls, and the values here are not comparable with it; given those controls with `--single-stains`, the package computes it (`spillover_spreading_matrix.csv`).

`spreading_pairs.csv` also screens for compensation errors. Correct compensation leaves the receiver's median where it is as the source brightens, so `trend_rho`, the rank correlation between the two among source-positive cells negative for the receiver, is flagged `possible_compensation_error` at 0.3 or more in size, a conventional cut rather than a validated one. A negative trend on a pair whose receiver is also wider at a given source level is reported as `trend_unresolved` instead: the screen reads only cells below the receiver's cut, and brighter source cells, with the wider receiver distribution, lose more of it to that cut, which pulls the trend negative with compensation correct. Single-stain controls decide such a pair.

Commands and every option

One page for every way to run the pipeline: the commands for each task, in Docker and in R, and every option with the situation it is for.

Part 1 is the commands. Part 2 is how to choose between the options that matter. Part 3 is the exhaustive reference.

Part 1. Commands

1.1 Docker or R

Both call the same `run_cyraven()`. The command-line front end is `system.file("scripts", "cyraven.R", package = "cyRAVEN")`, which is what the container entrypoint executes.

	Docker	R
Use it when	Results will be published, shared or compared against another machine	Developing, exploring, or Docker is unavailable
Reproducible	Yes: R 4.4.3, a dated CRAN snapshot and Bioconductor 3.20 are pinned	Only against your own library
Option syntax	`--group-column cohort`	`group_column = "cohort"`
Paths	Inside the container	On the host

Every option name is the same in both. Replace hyphens with underscores: `--group-column` becomes `group_column =`. Flags that take no value become `TRUE`: `--cluster` becomes `unsupervised = TRUE`. `docker run --rm cyraven:1.0.0 --help` prints the list at the version installed.

1.2 Get the image, once

See Setup. Pull for a released run, build for a modified source tree, and pin the version for anything you intend to publish.

1.3 A first run on a new cohort

Each step is cheap and catches what the next would otherwise waste time on. Every block below is one command, meant to be copied on its own and run before the next is copied. Steps 2 and 4 are edits you make in a text editor, not commands, which is the other reason these are not one block: copying a numbered list of commands and comments into a terminal in one go runs them all before you have looked at anything.

1. Make the folders and put the FCS files in `data/fcs/`.

```
mkdir -p data results
```

2. Write a sheet with a row for every file.

```
docker run --rm \
  -v "$PWD/data:/data" \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --recursive \
  --write-samples /data/samples.csv
```

3. Edit `data/samples.csv` and fill in `patient_id`, `cohort` and any study variable. No command; open it in a spreadsheet or a text editor.

4. Start the config from the shipped template.

```
docker run --rm -v "$PWD/data:/data" \
  cyraven:1.0.0 \
  --write-template /data/analysis.yaml
```

5. Edit `populations:` in `data/analysis.yaml` to match your panel. Again an edit, not a command.

6. Validate, in seconds, analysing nothing.

```
docker run --rm \
  -v "$PWD/data:/data:ro" \
  -v "$PWD/results:/results" \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --recursive \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --outdir /results \
  --check
```

7. Repeat steps 5 and 6 until `--check` reports no problems. Then run it.

```
docker run --rm \
  -v "$PWD/data:/data:ro" \
  -v "$PWD/results:/results" \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --recursive \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --group-column cohort \
  --reference-group "Healthy controls" \
  --outdir /results
```

8. Open `results/report.html`.

The two input files are described in the Inputs.

In R:

```
library(cyRAVEN)

opt <- list(
  dir = "data/fcs", recursive = TRUE,
  outdir = "results/",
  samples = "data/samples.csv",
  config = "data/analysis.yaml",
  group_column = "cohort",
  reference_group = "Healthy controls",
  seed = 42)

# validate
run_cyraven(c(opt, list(check_only = TRUE)))
# run
run_cyraven(opt)
```

1.4 The full analysis

Batch diagnostics and the unsupervised check are opt-in because each adds outputs and runtime.

```
docker run --rm \
  -v "$PWD/data:/data:ro" \
  -v "$PWD/results:/results" \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --recursive \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --group-column cohort \
  --reference-group "Healthy controls" \
  --batch-column acquisition_date \
  --cluster \
  --covariates age,sex \
  --reference-date 2025-01-01 \
  --no-session \
  --outdir /results
```

1.5 A cohort that must fit in memory

Peak memory is set by the events held at once, not by the number of files.

```
docker run --rm \
  -v "$PWD/data:/data:ro" \
  -v "$PWD/results:/results" \
  cyraven:1.0.0 \
  --dir /data \
  --recursive \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --max-events-per-file 300000 \
  --cells-per-sample 6000 \
  --max-cells 150000 \
  --no-session \
  --outdir /results
```

Run one analysis at a time. The stages are already multi-threaded, so nothing is gained by running containers in parallel, and two concurrent runs at these settings will exhaust a workstation. A run killed by the out-of-memory killer exits 137 and writes no report, because SIGKILL runs no handler.

Subsampling raises every detection limit in proportion, because it lowers the parent-gate count each frequency is a fraction of.

1.6 Comparing runs over time

Once, on the run being adopted as the reference.

```
docker run --rm \
  -v "$PWD/data:/data:ro" \
  -v "$PWD/results:/results" \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --outdir /results \
  --write-baseline /results/baseline.rds
```

On every later run.

```
docker run --rm \
  -v "$PWD/data:/data:ro" \
  -v "$PWD/results2:/results" \
  -v "$PWD/results:/prev:ro" \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --outdir /results \
  --baseline /prev/baseline.rds \
  --fail-on-drift
```

`--fail-on-drift` exits non-zero after every output has been written, so it can gate a pipeline without costing the diagnostics that explain the failure.

1.7 The demonstration cohort

See Quick start. Nothing is downloaded: the data ship inside a package cyRAVEN already depends on, so the example is reproducible offline.

1.8 Iterating on the code

Mount a checkout and set `CYRAVEN_SOURCE`; the entrypoint loads that tree through `pkgload::load_all()` instead of the installed copy, so edits take effect without a rebuild.

```
docker run --rm \
  -v "$PWD:/src:ro" \
  -v "$PWD/data:/data:ro" \
  -v "$PWD/results:/results" \
  -e CYRAVEN_SOURCE=/src \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --outdir /results
```

1.9 Paths, mounts and platforms

Every path inside a flag is a path inside the container. `--outdir` must fall within a mounted volume or the output is discarded when the container exits.

Platform	Substitution
Linux, macOS	as written
Windows PowerShell	`${PWD}` for `$PWD`
Git Bash on Windows	prefix `MSYS_NO_PATHCONV=1`, or the shell rewrites `/data` into a Windows path

Mount the data read-only (`:ro`) so a bug cannot alter raw acquisition files.

Part 2. Which option to use when

2.1 The rule behind the defaults

Options fall into three classes, and the class fixes the default.

Additive. Produces a new output and alters nothing existing. The diagnostics are on by default, because an analysis that has to be requested is one that does not get run, and the `--no-*` flags switch them off. Secondary or exploratory analyses (`--cluster`, `--explore`, `--edger`, `--rank-ancova`) and the exports are asked for.

Parametric. Changes how an existing quantity is computed. Defaults are the values the pipeline was validated with; changing one changes results and is a decision to record.

Consequential. Changes numbers a previous run reported, by design. Off by default without exception, and recorded in `run_manifest.txt` when used. They are `--transform`, `--per-marker-cofactor`, `--adaptive-gates`, `--peer-cuts` (and `--auto-fix-gates`, which implies it), `--drop-unstable-events`, `--peacoqc`, `--subsample rare`, `--correct-batch` and `--calibration-beads`.

2.2 Choosing by situation

Each of the following is a complete command. `$D` is the mount pair every run needs, written once here so the recipes stay readable:

```
D="-v $PWD/data:/data:ro \
   -v $PWD/results:/results"
```

or write the two `-v` flags out in full, as in Part 1.

Most thresholds found no density minimum
`thresholds_used.csv` shows `source = tailgate` (or `quantile_fallback`) on most rows. The marker did not separate positive from negative, so the cut carries no evidence of separation. Try the transform first, because for dim markers it decides whether a stable threshold exists at all.

```
docker run --rm $D \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --transform logicle \
  --group-column cohort \
  --reference-group "Healthy controls" \
  --outdir /results
```

If it persists, declare an unstained or fluorescence-minus-one control in the sheet (`is_control`, `fmo_for`) so the cut has a reference rather than a guess. If it still persists, the marker is not resolving in this panel and no gating strategy recovers it; `spreading_receivers.csv` says whether the optics explain it. `--transform` is consequential: it moves every threshold, so say so when reporting.

A population may exist that the specification does not describe
The declared populations are only ever what was declared. Clustering is the one output that can contradict them.

```
docker run --rm $D \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --cluster \
  --explain-clusters \
  --group-column cohort \
  --reference-group "Healthy controls" \
  --outdir /results
```

`--cluster` writes `cluster_gate_agreement_*.csv`: a cluster dominated by no declared label is a population the specification misses, and a label spread across many clusters covers several phenotypes. `--explain-clusters` then learns a two-marker gating strategy for each undescribed cluster and writes `cluster_gate_proposals.csv`. It is descriptive: it proposes gates and never changes a scored one. Promoting a proposal to a named population is a manual edit to the config, and the next run reads it.

```
docker run --rm $D \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --batch-column acquisition_date \
  --group-column cohort \
  --reference-group "Healthy controls" \
  --outdir /results
```

Samples were acquired on several days or instruments This quantifies the batch structure without altering anything: iLISI against a permutation null, per-marker distributional drift, and Cramér's *V* between batch and group.

Read `batch_group_confounding.csv` before going further. If *V* is high, the batches and the groups are the same partition, and correcting the batch removes the finding. cyRAVEN refuses correction above Cohen's large association for the table's size, $0.5/\sqrt{(k-1)}$ with k the smaller of the batch and group counts (0.5 for two groups), for that reason. Where *V* is low and the drift is real, add correction:

```
docker run --rm $D \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --batch-column acquisition_date \
  --correct-batch \
  --group-column cohort \
  --reference-group "Healthy controls" \
  --outdir /results
```

`--correct-batch` is consequential. It applies to the embedding and clustering, not to per-sample thresholds, which are batch-local by construction.

Absolute cell numbers rather than percentages A frequency is a proportion of a parent gate, so it moves whenever any other population moves. Add `count.<Population>` columns to the sheet, in cells per microlitre, and the run tests them alongside the frequencies:

```
file,sample_id,cohort,count.Monocytes
HC-01.fcs,HC-01,Healthy controls,420
```

```
docker run --rm $D \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --group-column cohort \
  --reference-group "Healthy controls" \
  --outdir /results
```

No extra flag: the columns are recognised by their prefix. `absolute_counts_qc.png` draws every supplied count as one tile per sample and population on a log scale, which is how a transcription error (a tile decades off its column) or a name mismatch (a missing row) is caught. Where the counting instrument exports its own sheet and transcription is not worth it, use `--absolute-counts <file>` instead.

The same assay runs repeatedly and must not drift Record where the accepted run placed its thresholds:

```
docker run --rm $D \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --outdir /results \
  --write-baseline /results/baseline.rds
```

Then test every later run against it:

```
docker run --rm $D \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --outdir /results \
  --baseline /results/baseline.rds \
  --fail-on-drift
```

The within-run peer check already finds one deviant tube among its peers. It cannot find a cohort that moved as a whole, because the peer median moves with it; only a baseline from an earlier run can. `--fail-on-drift` exits non-zero after every output has been written, so it can gate a pipeline without costing the diagnostics that explain the failure. Omit it while investigating.

```
docker run --rm $D \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --paired-column patient_id \
  --condition-column timepoint \
  --group-column cohort \
  --reference-group "Healthy controls" \
  --outdir /results
```

A longitudinal or paired design Both flags are required together. Pairing is never inferred: a design where each donor appears twice is indistinguishable, from the data alone, from one where two donors happen to share a label, and guessing wrong changes every p-value.

The same two flags also produce `population_trajectories.png`: one line per patient across the conditions, with the cohort median drawn over them. Add `--clinical-columns survival_28d` and the lines are coloured by outcome, which is the comparison a longitudinal design is usually about. Survivors and non-survivors often differ in the direction each patient moves rather than at any single timepoint, and a box per timepoint averages exactly that away.

```
C=sofa,survival_28d,crp_mg_l,age_years
docker run --rm $D \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --group-column infection_focus \
  --reference-group Pulmonary \
  --clinical-columns "$C" \
  --outdir /results
```

A cohort with clinical scores and outcomes Any sheet column can be named. `sofa` and `crp_mg_l` are numeric and get Spearman's rho, `survival_28d` has two levels and gets Wilcoxon with Cliff's delta, and each effect is written with a percentile bootstrap interval. On a cohort of ten that interval usually spans zero, meaning the effect is not resolved. Read `clinical_variables_correlation.png` first: p-values are adjusted within each variable on the assumption that the variables are separate questions, and two variables correlated with each other are one question asked twice.

One sample's threshold is visibly wrong Where `gating_qc.png` shows the derived cut sitting in a shoulder for one sample only, correct that sample rather than pinning the marker for all of them. In the config:

```
sample_overrides:
  PT-01_v1:
    CD14:
      threshold: 2.4
      set_by: "your initials"
      reason: >-
        bimodal but the minimum fell in
        the shoulder, see gating_qc.png
```

Then run normally. `set_by` and `reason` are recorded in `run_manifest.txt`, so a run touched by hand says so in its provenance rather than only in the table it altered. Setting a fixed threshold for every sample instead would reintroduce the fixed-coordinate bias the pipeline exists to remove.

A rare population is invisible in the embedding Uniform sampling reproduces each sample's composition, so a population at 0.1% contributes 0.1% of the drawn cells and forms no cluster.

```
docker run --rm $D \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --subsample rare \
  --cluster \
  --group-column cohort \
  --reference-group "Healthy controls" \
  --outdir /results
```

Consequential: it changes which cells are embedded, so the UMAP is not comparable with a uniform run, and `cells_umap.csv` gains a `sampling_weight` column. Frequencies are unaffected, because they are computed on all events rather than on the drawn subset.

```
docker run --rm $D \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --cluster \
  --explain-clusters \
  --export-gates \
  --flowjo-export \
  --outdir /results
```

Gates are needed on the instrument or in FlowJo `--export-gates` writes ISAC Gating-ML 2.0 on the scale the gates were fitted on, through its `fasinh` or `logicle` transformation, with each dimension named by the files' `$PnN`; and a vertex table in the linear units the FCS file stores, edges subdivided so the polygon describes the same region the fit did. To apply a file whose gates reference the files' own compensation (`compensation-ref="FCS"`) with flowCore and flowUtils, use flowCore 2.25 or later, or the cyRAVEN image, whose flowCore 2.18.0 carries the one-line fix: releases 2.18 to 2.22 stop on R 4.2 or later when they read a `$SPILLOVER` keyword ("the condition has length > 1"). `--flowjo-export` writes UMAP-annotated FCS: open `_ALL_SAMPLES.fcs`, plot UMAP-1 against UMAP-2, and the populations are there as channels.

```
docker run --rm $D \
  -v "$PWD/beads:/beads:ro" \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --calibration-beads /beads/rainbow.fcs \
  --calibration-values /beads/lot.csv \
  --outdir /results
```

Intensities must be comparable across instruments or over months Consequential, and the most far-reaching of them: conversion happens before any threshold is derived, so every intensity in the run changes units. A channel whose fit falls below `--calibration-min-r2` is left in instrument units rather than converted on a fit that does not hold, and the log says which.

```
docker run --rm $D \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --external-labels /data/labels.csv \
  --export-gates \
  --outdir /results
```

Labels from another tool Learns a gating strategy per supplied label and measures whether it holds on donors it was not fitted to. Held-out events from the same donors overstate how well a gate travels; the per-donor minimum in `gate_transferability.csv` does not. See Interoperability.

The run is slow, or the machine runs out of memory In this order, because the first costs nothing analytically:

Skip the session file: often most of the wall-clock, none of the results.

```
docker run --rm $D \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --no-session \
  --outdir /results
```

Bound the events read per file, then the cells embedded.

```
docker run --rm $D \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --no-session \
  --max-events-per-file 300000 \
  --cells-per-sample 6000 \
  --max-cells 150000 \
  --outdir /results
```

`--no-session` changes no number. `--max-events-per-file` does: it raises every detection limit in proportion, because it lowers the parent-gate count each frequency is a fraction of. Run one analysis at a time; the stages are already multi-threaded.

You suspect the specification is missing something The declared path scores what you declared. It cannot find a population nobody wrote down, and it cannot see outside the parent gate. Explore mode is the complement: every event, every eligible channel, no specification.

Alongside the declared analysis. Writes to results/explore/ and changes NOTHING in the existing output.

```
docker run --rm $D \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --group-column cohort \
  --reference-group "Healthy controls" \
  --cluster \
  --explore \
  --outdir /results
```

...and let the two sides inform each other. The declared run lends explore its per-sample thresholds, so clusters are NAMED rather than guessed at; explore lends the declared run spec_gaps.csv, naming populations that span several clusters and clusters nothing covers.

```
docker run --rm $D \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --group-column cohort \
  --reference-group "Healthy controls" \
  --cluster \
  --explore \
  --maybe-learn \
  --outdir /results
```

Without `--maybe-learn` the two are computed in complete isolation, which is the right default: a check on the specification has to be independent of the specification to be worth anything.

```
# No --config and no --samples:
# a folder of FCS files is the whole input.
docker run --rm $D \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --explore-only \
  --outdir /results
```

A new panel, and no specification written yet Read `results/explore/explore_report.html`, then curate `results/explore/explore_suggested_spec.yaml` into a real specification and run the supervised path with it. Full detail in Explore mode.

2.3 The options that change reported numbers

Turn these on only when you mean to, and say so when you report a result.

Option	What it changes	Read first
`--transform logicle`	Every threshold, and therefore every frequency	`gating_qc.png`
`--per-marker-cofactor`	Every threshold on an arcsinh scale	`gating_qc.png`
`--adaptive-gates`	Thresholds, wherever another rule scores a better gap	`thresholds_used.csv` (`source`), `gating_qc.png`
`--peer-cuts`, `--auto-fix-gates`	The cuts and hierarchy gates they replace or skip	`gate_adjustments.csv`
`--drop-unstable-events`	Every count in an affected file	`acquisition_qc_impact.csv`
`--peacoqc`	Every count in a file PeacoQC flags	`peacoqc_summary.csv`
`--subsample rare`	Which cells are embedded	`cells_umap.csv` gains `sampling_weight`
`--correct-batch`	Marker values in the embedding and clustering	`batch_group_confounding.csv`
`--calibration-beads`	The units every intensity is expressed in	`calibration.csv`

2.4 What not to change without cause

`--cofactor` is estimated per panel for a reason: the conventional value of 5 comes from mass cytometry and over-expands the background band on unmixed fluorescence.

`--batch-max-cramers-v` exists to refuse correction where batch and group are inseparable. Raising it, or passing `--force-batch-correction`, means the resulting difference cannot be attributed to biology.

Pinning a threshold in the config applies it to every sample and reintroduces the fixed-coordinate bias the pipeline exists to remove. Use `sample_overrides:` for one sample instead.

Part 3. Every option

The same names are the element names of the list passed to `run_cyraven()`, with hyphens replaced by underscores.

3.1 Input

Option	Default	Effect
`--files`	none	Comma-separated paths, or a glob in quotes
`--dir`	none	Directory to search for `.fcs`. Alternative to `--files`
`--recursive`	off	Search `--dir` recursively, for cohorts in subdirectories
`--pattern`	`[.]fcs$`	Regex for filenames under `--dir`. Any file containing `.fcs` that fails the pattern is listed as a warning rather than silently skipped
`--exclude`	compensation controls	Regex; matching paths are dropped after discovery. The default removes single-stain setup files, which carry no CD45 population and form a spurious panel group. `--exclude ''` keeps everything
`--max-events-per-file`	0, meaning all	Read at most N events per file, sampled evenly through the acquisition rather than taken from the start. Bounds peak memory, and raises every detection limit in proportion because it lowers the parent-gate count
`--outdir`	`results`	Output directory

3.2 Inputs and metadata

The standard input is one CSV and one YAML. The three separate tables below them still work unchanged and are not deprecated, but cannot be combined with `--samples`. See the Inputs.

Option	Default	Effect
`--samples`	none	The sample sheet: one row per FCS file carrying its identity, its subject's attributes, its study variables and any `count.<Population>` columns. Replaces the three options below
`--config`	built-in spec	The analysis: population specification, `functional_blocks`, `ratios`, threshold and `sample_overrides`, colours, metadata dictionaries, and the `samples:` section naming which sheet column plays which role
`--write-samples`	none	Write a sheet template covering every input file, and exit
`--check`	off	Validate the inputs from FCS headers alone and exit. Reports resolved markers, specification entries matching none of them, sheet coverage, group levels and sizes, batches and study variables. Seconds rather than the full run
`--list-channels`	off	List every acquisition parameter and exit: index, $PnN, $PnS, and the symbol the run resolves it to, which is the name the specification must use. Also reports whether every file carries the same panel. Needs no sheet and no config, because it answers what to put in one
`--ignore-channels`	none	Comma-separated marker names dropped before anything else, including before the panel fingerprint. Names match whole and case-insensitively; a name containing `*` is a glob. See A channel that splits the cohort
`--reference-date`	today	`YYYY-MM-DD` used to derive age from date of birth and to place two-digit years. Set it to a fixed study date so ages do not change between runs
`--group-column`	resolved `cohort`	Column defining the comparison groups. Enables the between-group figure and tests. Also settable as `samples: group_column:` in the config, which the flag overrides
`--reference-group`	first alphabetically	The group every other is tested against. Drawn unfilled and leftmost

Option	Default	Effect
`--write-config`	none	Derive everything, write the resulting YAML, and exit. Shows what the pipeline detected before time is spent. The per-sample cuts are summarised under `thresholds_derived`, which no run reads; a value copied into `thresholds:` is applied to every sample. **Needs `--outdir` as well**: deriving the parameters means reading the files, so the early stages run and write their QC output alongside

The earlier three-file form:

Option	Default	Effect
`--sample-map`	none	CSV linking each filename to a sample, patient, group and control status. Only `file` is required. Also carries `fmo_for` and `control_group`
`--patient-table`	none	Clinical covariates keyed on patient identifier. Supplies `wbc_per_ul` for absolute concentrations
`--absolute-counts`	none	Externally measured absolute counts, wide format. Requires `--sample-map`
`--write-sample-map`	none	Write a sample-map template and exit

A channel that splits the cohort The panel fingerprint is the set of marker names a file resolves. Files sharing a set form one panel, and each panel gets its own cofactor, its own embedding and its own thresholds. That is correct when panels differ, and destructive when they do not.

Spectral unmixing is where it usually goes wrong. Unmixing writes the extracted autofluorescence back as channels, often named `[AF color 1]` and upward, and how many it writes depends on the sample rather than on the panel. They are not stains and nothing is measured with them, but they enter the fingerprint like any other channel, so twelve comparable files can resolve into seven panels of one or two files each. Every downstream quantity is then derived within a panel of one, and nothing is comparable to anything.

`--list-channels` shows the resolved names, and `--check` reports markers that are not present in every file. When the culprit is a nuisance channel, drop it:

```
--ignore-channels '[AF color*'
```

The channel is removed before the fingerprint is taken, so the files collapse back into one panel. Names match whole and case-insensitively, and `*` makes an entry a glob, so `CD16` matches `CD16` and not `CD161`. Separate several with commas.

Only drop a channel that carries no signal you intend to use. A real marker missing from some files is a different problem, and the honest handling of it is the panel split.

3.3 Transformation and gating

Option	Default	Effect
`--transform`	`arcsinh`	`arcsinh`, `logicle` or `none`. **Consequential**: it moves every threshold. For dim markers it decides whether a stable threshold exists at all
`--cofactor`	derived per panel	Fixed arcsinh cofactor. The conventional value of 5 comes from mass cytometry and over-expands the background band on unmixed fluorescence, which is why it is estimated rather than assumed
`--cofactor-from-first-sample`	off	Derive the cofactor from the first file only, as before pooling was introduced
`--per-marker-cofactor`	off	One arcsinh cofactor per marker, from the spread of that marker's negative population, instead of one for the panel. **Consequential**: it moves every cut. See section 4.2
`--cofactor-from-all-events`	off	Estimate the cofactor over every event, debris included, instead of the scatter-gated single cells
`--keep-margin-events`	off	Keep events at the top of a marker or scatter channel's range, capping their values, instead of removing them. Counts are in `margin_events.csv` either way
`--spillover`	none	CSV spillover matrix applied to every file in place of its own `$SPILLOVER`
`--fmo-sets-cut`	off	Let a true FMO set the cut even where the sample's valley is higher, so dim positives between them count as positive (`fmo_q995_valley_lowered`). Off by default because an FMO argues a cut down validly only where its negative is the sample's own, and one tube serves every sample of its panel or `control_group`; a valley below the FMO's edge is raised to it either way. Never applied to unstained or minus-multiple tubes
`--logicle-m`	4.5	Decades on the logicle display scale
`--singlet-mad-k`	3	How far below the FSC-H:FSC-A ridge, in MADs, the singlet gate cuts. One-sided, because doublets have a low ratio. The gate is skipped, with a log line, when the two are not distinct channels
`--viability-marker`	auto-detect	Name the dye where the pattern is ambiguous
`--min-cd45-pct`	5	Staining QC floor, percent CD45+ of live
`--min-live-pct`	unset	The assay's minimum live share of single cells, in percent; a sample below it fails staining QC. Unset, the share is reported in `staining_qc.csv` and not judged, because no published minimum exists. A sample whose dead cells outnumber its live ones reads as mostly live: the viability cut is placed above the dye's tallest mode, which is then the dead cells. `pct_live_if_dominant_dead` beside it gives the live share for that case, from a dye minimum below the dominant mode
`--include-qc-failed`	off	Keep declared samples that failed staining QC. Admits them to every stage, not only the embedding. Their percentages carry no evidence; under this flag `qc_status` reads `pass` and only the `verdict` column in `staining_qc.csv` records which were forced in
`--discover-controls`	off	Let a sample that fails staining QC become the unstained reference for its panel even though the sheet declares no control. See below for why this is off

Option	Default	Effect
`--adaptive-gates`	off	Choose each threshold by sweeping the kernel bandwidth and trying the tail and Otsu rules, instead of one fixed smoothing with a tail-gate fallback. **Consequential**: it moves thresholds. See Placing a threshold when no valley exists
`--peer-cuts`	off	Where a population marker shows a density minimum in most samples of a panel but not in this one, take this sample's cut from the median of those minima; where most samples show none, replace only a quantile fallback or a tail cut far from the others. Each change is a row of `gate_adjustments.csv`, and the original cut stays in `derived_threshold`. **Consequential**: it moves thresholds. Implied by `--auto-fix-gates`
`--panel-optional-markers`	none	Comma-separated markers that do not define the panel. A reagent stained in only some files otherwise splits the cohort into separate panels with separate embeddings. Named here, the files stay one panel; a population needing the marker scores where it exists and is UNAVAILABLE where it does not

Placing a threshold when no valley exists

`density_valley()` answers one question: where is the deepest separation between two modes that counting noise cannot explain. Where a marker shows none it answers `NA`, and the threshold then comes from the tail gate, 3.5 robust standard deviations above the dominant mode and flagged for review; a fixed quantile of the parent remains only where no tail gate can be placed.

Before the tail gate that quantile was the default, and on a spectral panel it was not a rare outcome. On the sepsis cohort this was written against, **65% of every sample-by-marker cut** came back `quantile_fallback`, which means two thirds of the gates were placed by the constant rather than by the data, and the viability gate was skipped in all 20 samples because L/D never resolved.

The markers were not unimodal. The histogram being smoothed (220 bins, a 9-bin moving average) is under-smoothed for these data: L/D showed 32 peaks in one sample and CD45 showed 10, nearly all of them noise. Widen the kernel and CD45 resolves to exactly two modes.

`--adaptive-gates` keeps the same vocabulary as openCyto (Finak and others, 2014) rather than inventing one:

Method	For
mindensity over a bandwidth ladder	a bimodal marker, at whatever smoothing resolves it
tailgate	one real mode with a positive tail. The cut sits a robust spread above the mode, estimated by mirroring the half the tail cannot reach
Otsu	two classes of comparable size

Every candidate is then scored the same way: **how deep a density gap the cut sits in**. That is method-agnostic, needs no per-marker configuration, and measures the property a gate is supposed to have. Where nothing sits in a gap the tail rule is preferred over the quantile, because it is anchored to the sample's own mode and spread rather than declaring a fixed share positive.

Measured on that cohort: quantile fallbacks fell from **65% to under 1%** on properly gated data, and the viability gate placed at 0.1 to 2.4% dead.

It cannot find a population that was not stained. One sample there carries CD3 at 0.3% positive while its CD45 stains normally. No threshold rule recovers T cells from a channel holding none, and the selector reports what is there.

Do not require the parent marker twice

A specification that opens every population with `CD45: above` looks like the textbook leukocyte gate and is a trap.

The hierarchy has already applied CD45: the parent every population is scored inside is `cd45_pos`. Repeating the requirement asks for a second CD45 threshold derived from the cells that passed the first. In PBMC that distribution is all positive and unimodal; in whole blood its one minimum separates the CD45-dim granulocytes from the CD45-bright lymphocytes, and `above` then removes the granulocytes from every population that names it. Either way the second cut describes nothing the specification meant.

When single-mode cuts fell back to the 90th percentile, an `above` comparison kept exactly 10%. On the cohort above, CD45 fell back to the quantile in 17 of 20 samples and the share of parent cells carrying any label came out at 10.0, 9.4, 9.3, 9.8 and 9.5% for the first five. Removing the redundant requirement took classified cells from **21% to 88%**. Nothing else about the specification changed. The tail gate that now replaces the quantile goes above the dominant mode unless the marker is declared `tail: lower`, so on PBMC, where every parent cell is CD45-positive, the repeated requirement now keeps almost nothing.

The rule generalises: do not name a marker in a population if a gate above it has already applied that marker.

3.4 Uncertainty and detection limits

Option	Default	Effect
`--no-uncertainty`	on	Skip the gate placement uncertainty analysis, and with it the detection limits (`lod_pct`, `loq_pct`, `detection`), which are computed in the same pass
`--uncertainty-boot`	100	Bootstrap replicates per threshold. Below about 50 the standard uncertainty is itself noisy
`--uncertainty-max-events`	20000	Events per replicate. The cut is a histogram feature and stops moving well before the whole parent gate is used
`--lod-events`	20	Events below which a population is reported below the limit of detection
`--loq-events`	50	Events below which it is detected but not quantified. Roughly a 14% counting CV, against 22% at the detection limit

3.5 Acquisition and panel quality

Option	Default	Effect
`--no-acquisition-qc`	on	Skip the acquisition stability check
`--acquisition-bins`	40	Equal-width Time intervals per file. More bins resolve a shorter disturbance and make each interval's median noisier

Option	Default	Effect
`--acquisition-mad-k`	5	Robust z above which an interval is flagged
`--drop-unstable-events`	off	**Consequential**: exclude the flagged intervals, which changes every count, threshold and frequency in the affected files. Read `acquisition_qc_impact.csv` first
`--peacoqc`	off	**Consequential**: remove the events PeacoQC (Emmaneel et al. 2022) flags, before any cofactor, threshold or frequency is derived. Its input is prepared as the PeacoQC vignette prescribes (logicle on the fluorescence channels) and its checks are run with `determine_good_cells = "all"`; PeacoQC runs the isolation tree only on 150 or more time bins, and `peacoqc_summary.csv` notes the files where it did not. Needs a Time channel and the PeacoQC package; replaces `--drop-unstable-events` when both are given. `acquisition_qc.csv` still describes the acquisition before removal. Writes `peacoqc_summary.csv`
`--no-spreading`	on	Skip the spillover spreading report
`--spreading-max-samples`	8	Samples contributing to the spreading ranking, which stabilises well before a whole cohort is used

3.6 Embedding

Option	Default	Effect
`--cells-per-sample`	20000	Cells drawn per sample for the embedding
`--max-cells`	200000	Overall ceiling per embedding. The per-sample allowance is reduced proportionally if the total would exceed it
`--no-equalise`	off	Embed every gated cell up to `--cells-per-sample`. By default no sample contributes more than the lower quartile of the per-sample counts, so no sample dominates the map; a sample with fewer cells contributes all it has
`--umap-scale`	`auto`	How each marker is scaled before the UMAP and the clustering: `none`, `range` (0.5th to 99.5th percentile), `robust` (median/MAD) or `zscore`. `auto` uses `none` under `--transform logicle` or `--per-marker-cofactor`, whose scale every channel already shares, and `range` otherwise: median/MAD divides a mostly negative marker by the width of its negatives and inflates its background
`--subsample`	`uniform`	`uniform` reproduces the sample's composition; `rare` weights by inverse local density so a population too small to form a cluster can be recovered. **Consequential**: it changes which cells are embedded and adds `sampling_weight` to `cells_umap.csv`
`--umap-markers`	built-in lineage list	Comma-separated features. Override this where the default list matches fewer than two markers, rather than relying on the all-eligible fallback
`--umap-markers-all`	off	Use every eligible marker instead of preferring lineage markers
`--n-neighbors`	30	UMAP neighbourhood size. Larger values favour global structure
`--min-dist`	0.3	UMAP minimum distance. Smaller values pack clusters more tightly
`--save-umap-model`	none	Persist the trained model so a later batch can be projected into the same embedding
`--umap-model`	none	Project into a saved model instead of embedding fresh. Right for more of the same kind of sample, wrong for a new panel or cell type
`--threads`	0, meaning all cores	UMAP threads

3.7 Clustering and learned gates

Option	Default	Effect
`--cluster`	off	Self-organising map clustering and the cross-check against the specification. One of two outputs, with `--explore`, that can find a population the specification does not describe
`--cluster-k`	12	Metaclusters
`--cluster-grid`	10	SOM grid side; the map has grid2 nodes
`--auto-subcluster-k`	off	Choose the subcluster count per population by mean silhouette. Changes the subcluster lettering, so it is opt-in
`--explain-clusters`	off	Learn a two-marker gating strategy for each cluster the specification does not describe. Requires `--cluster`. Descriptive: it proposes gates and never changes a scored one
`--explain-max-clusters`	4	Ceiling on strategies derived. The largest qualifying clusters are taken and the number skipped is logged
`--explain-max-depth`	4	Maximum gates in a strategy. More levels raise precision and cost recall; the search stops early when another gate does not earn its place

3.8 External labels and gate export

Option	Default	Effect
`--external-labels`	none	CSV of cell labels from another tool. Learns a gating strategy per label and measures whether it holds on donors it was not fitted to
`--external-max-labels`	6	Ceiling on labels gated; the largest are taken and the number skipped is logged
`--transfer-max-donors`	8	Leave-one-donor-out folds. Every fold refits, so cost is one fit per donor per label. The subset is drawn at random rather than taken as the largest donors, because a worst-donor score computed on the best-represented donors is optimistic
`--transfer-max-cells`	20000	Training cells per fold, stratified by label. The held-out donor is always scored in full
`--export-gates`	off	Write learned gates as ISAC Gating-ML 2.0, on the analysis scale through its `fasinh` or `logicle` transformation and named by the files' `$PnN`, and as a vertex table in the linear units the FCS file stores
`--flowjo-export`	off	Also write UMAP-annotated FCS for interactive inspection
`--flowjo-outdir`	`<outdir>/flowjo`	Directory for that export
`--flowjo-no-concat`	off	Skip the concatenated `_ALL_SAMPLES.fcs`
`--flowjo-no-groups`	off	Skip the per-group files

3.9 Statistics

Option	Default	Effect
`--p-adjust-display`	`raw`	Which p-value the figure brackets show, `raw` or `BH`. Both are always written to the CSV
`--no-differential-state`	on	Skip the sample-level differential-state tests
`--no-compositional`	on	Skip the centred log-ratio re-test and its concordance table
`--edger`	off	Add a count-model abundance test beside the rank tests: edgeR's negative binomial GLM on each sample's exclusive-label counts, with diffcyt-DA-edgeR's defaults (populations with 3 or more cells in at least half the samples, no TMM normalisation, dispersion without a trend, a likelihood-ratio test of each group against the reference, BH). Writes `edger_abundance_stats.csv`. Needs the edgeR package
`--no-confounding`	on	Skip the age and sex confounding diagnostic
`--no-group-tests`	on	Keep the grouping but run no between-group test. Figures stay split and coloured by `--group-column` and every per-sample quantity is still reported; the abundance (with its parametric, `--edger` and `--rank-ancova` forms), functional-marker, ratio, absolute-count, marker-state and compositional comparisons are skipped. Diagnostics are unaffected
`--min-group-n`	3	Minimum samples in a group before it is tested. Groups below it appear in `design_feasibility.csv` instead of being tested
`--no-parametric`	on	Skip `parametric_tests.csv` and `posthoc_tests.csv`
`--read-threads`	1	Read this many FCS files at once. Reading is the slowest stage and each file is independent. Peak memory scales with it, and results do not change
`--covariates`	`age,sex`	Patient-table columns screened as confounders
`--clinical-columns`	none	Sheet columns to test as clinical variables in their own right – a severity score, a laboratory value, an outcome flag. Different from `--covariates`: a confounder is screened to decide whether a group difference can be believed, a clinical variable *is* the question. Numeric columns get Spearman, two-level columns Wilcoxon with Cliff's delta, three or more Kruskal-Wallis with epsilon-squared, and p-values are adjusted within each variable. Writes `clinical_association*.csv` and the figures below
`--rank-ancova`	off	Additionally fit an exploratory covariate-adjusted rank ANCOVA. At single-digit n per group the adjustment is usually extrapolation; read `confounding_diagnostics.csv` first
`--paired-column`	none	Column identifying the pairing unit, for example donor. Requires `--condition-column`. Pairing cannot be inferred, so nothing paired runs without it
`--condition-column`	none	Column naming the condition within a pair, for example timepoint
`--no-threshold-drift`	on	Skip the check for thresholds that differ systematically between groups
`--no-heatmaps`	on	Skip the phenotype and cohort composition heatmaps
`--other`	off	Draw the `Other CD45+` catch-all on the UMAP figures. A display choice only: those cells are always gated, counted and tabulated
`--no-marker-group-umaps`	off	Skip `marker_umaps_by_group/`. Written by every run: one pooled UMAP per marker over all samples, plus the same marker split by **every category present**. Turn it off for a panel wide enough that this is a folder you do not want

The pooled panel and the split panels answer different questions. Splitting asks whether a marker sits differently between categories; pooling asks where the marker is at all. The second has no other home: each facet holds only a subset of the cells, and `umap_markers.png` shrinks every marker into one cell of a grid, so no other figure shows a single marker over the whole cohort at full size.

Every category is drawn, not only the one named by `--group-column`. That flag decides what is *tested*; it says nothing about what is worth *seeing*, and a cohort usually carries several categories. Numeric columns are never faceted (one panel per distinct value), nor are identifiers such as `sample_id` and `patient_id`, nor single-level columns. Past four categories the extras are named in the log rather than dropped silently, and `--group-column` moves one to the front of the list.

3.10 Batch structure

Option	Default	Effect
`--batch-column`	`$DATE` where it varies	Column identifying the acquisition batch. Enables iLISI, Cramér's *V*, per-marker distributional drift and the threshold test grouped by batch
`--correct-batch`	off	**Consequential**: align each marker across batches by monotone quantile mapping. Applied to the embedding and clustering, not to per-sample thresholds, which are batch-local by construction
`--batch-max-cramers-v`	Cohen's large	Refuse correction above this association between batch and group. The default is Cohen's large association, w = 0.5: *V* = 0.5 for two batches or groups, 0.35 for three. Above it, removing the batch and removing the finding are the same operation
`--batch-pooled-goal`	off	Align every batch towards all cells pooled, as before. By default each study group is aligned towards the mean of its quantiles over the batches every group was measured in, so a batch holding mostly one group does not pull it towards the other and groups measured in different batches stay on one scale; nothing is corrected where no batch holds every group
`--force-batch-correction`	off	Correct anyway. Recorded in the manifest, and the finding cannot then be attributed to biology
`--batch-method`	`quantile`	How a correction is fitted, never whether one is defensible. `quantile` fits one map per marker over the whole file. `cluster` fits one per marker **per cell type**, which a whole-file map cannot do when a detector shift moves bright and dim populations by different amounts. `cytonorm` is accepted for `cluster`, with a note: it is not the CytoNorm package and uses no anchor samples
`--batch-cluster-k`	10	Cell-type clusters to fit for `--batch-method cluster`

The refusal is evaluated before `--batch-method` is read, so both methods are refused on identical evidence. A better alignment algorithm does not make a confounded design correctable.

`cluster` is fitted in two stages. Clustering the raw matrix would let a large batch shift become the dominant source of variance, so the clusters would be the batches, each holding one batch with nothing to align against. The clustering is therefore fitted on a whole-file-aligned copy and the per-cluster maps on the original values. Measured on a synthetic three-batch shift: fitted the naive way it left a mean between-batch gap of 1.296 against whole-file alignment's 0.003; fitted this way it reaches 0.014 while leaving the true between-cell-type separation at 3.996 against a true 3.994, where whole-file alignment inflates it to 4.078.

Whichever method runs, `batch_correction.csv` records it, including on a refusal.

3.11 Calibration

Option	Default	Effect
`--calibration-beads`	none	**Consequential**: FCS of calibration beads acquired on the same instrument and settings. Converts channel units to the assigned units before any threshold is derived, so every intensity in the run changes
`--calibration-values`	none	CSV of assigned bead values, wide or long. Supplied by the manufacturer for that lot
`--calibration-min-r2`	0.98	Fit quality below which a channel is left in instrument units rather than converted on a fit that does not hold

3.12 Conformance across runs

Option	Default	Effect
`--write-baseline`	none	Record where this run placed each threshold, how variable it was, how often it needed the fallback, and what the populations came out at. Does not end the run
`--baseline`	none	Test this run against a baseline written earlier. Catches drift that moved a whole cohort, which the within-run peer check cannot see
`--fail-on-drift`	off	Exit non-zero when any marker or population fails conformance. Raised after every output has been written

3.13 Reporting and session

Option	Default	Effect
`--no-report`	on	Skip `report.html`, the single self-contained file carrying every figure and table the run produced, in the documented reading order
`--report-images`	`embed`	`link` references the figures beside the report instead of embedding them: a few megabytes that open at once, readable only inside the results directory. Applies to `explore_report.html` too
`--no-miflowcyt`	on	Skip `miflowcyt.md`, the ISAC-structured record of the instrument and the analysis
`--no-session`	on	Skip `session_state.RData`. Worth doing: on a large cohort that file reaches several hundred MB and dominates the runtime, and nothing in the results depends on it
`--keep-exprs`	off	Include raw expression matrices in the saved session. Large
`--seed`	42	RNG seed. Seeded once per run; every stage that consumes draws restores the stream so the embedding cannot shift

`report.html` embeds every figure at full resolution and every table in full, so it can be moved, attached or archived on its own; it references no other file and needs no network. Every section, figure and table is numbered, and a search bar at the top finds them by name or description, and finds tables by the values in their rows (press `/` to start typing). Figures zoom and download at full resolution; tables are searchable, sortable, paged at 10/50/100/all rows and exportable to CSV exactly as filtered. A run that fails writes it too, with the diagnosis and everything produced before the failure.

Its size is the sum of the figures it carries, once each, which on a full run is tens of megabytes; a figure is decoded only when its tab is opened, so the page stays responsive. `--report-images link` makes it a few megabytes instead. A table larger than 8 MB is named with its row count rather than embedded, which in practice means only the per-cell exports; change the limit with `options(cyRAVEN.report_table_max_mb = )`.

3.14 Logging

Progress goes to stderr through `message()`.

```
# or "inform" (default), or "debug"
options(cyRAVEN.verbose = "none")
```

4 Advanced

Five things beyond a first run: finding a population nobody declared, checking whether the specification held, reading every file a run writes, the statistics and their assumptions, and using cyRAVEN alongside other tools.

The order is the one a result is actually interrogated in. Explore mode comes first because it can contradict the specification from outside it. The diagnostics come next, in reading order, since each stage can invalidate every stage after it. Then the output files, then the tests, then interoperability.

Explore mode: unsupervised discovery

Unsupervised discovery, run beside the declared analysis or on its own.

The rest of cyRAVEN starts from a specification: you declare populations, it scores them, and six diagnostics try to break the declaration. That design has one structural limit. **It cannot find a population nobody declared**, and it cannot look outside the parent gate, because both are consequences of the declaration itself.

Explore mode is the complement. It takes the single cells of every file and every eligible channel, embeds and clusters without reference to any specification, and reports what it found. It is what FlowSOM, Phenograph and cyCONDOR do. The cells are drawn from inside the scatter and singlet gates, which need neither the specification nor CD45, because debris and doublets would otherwise form clusters of their own; dead cells and non-leukocytes are judged afterwards by the cluster-level QC gate.

```
--explore
```

Everything it writes lands in `<outdir>/explore/`. **No existing output changes.**

1. Why run it here rather than in a clustering package

An unsupervised tool hands you cluster 17 and a heatmap, and you decide what it is by eye. That inference is done on pooled data, on a colour scale whose limits you chose, with no per-sample calibration behind it.

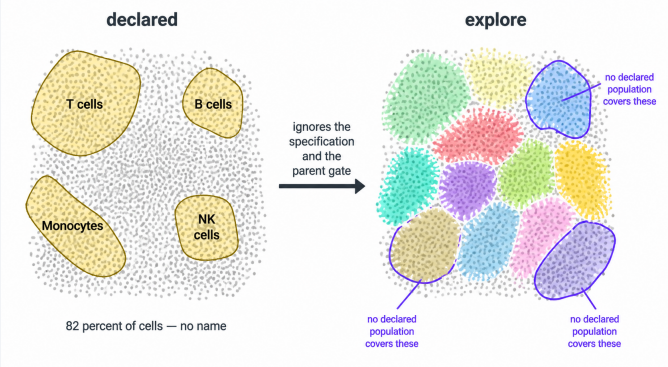


Figure 4.1: The declared analysis names a few regions and leaves most cells unnamed; explore mode partitions everything

With `--maybe-learn`, this module has things a standalone clusterer does not:

From the declared run	What explore does with it
The transform, cofactor **estimated** on this panel	Clusters on the same scale the gates were drawn on, instead of an assumed default
A threshold for every marker **in every sample**	Names each cluster from the fraction of its cells above *that sample's own cut*
Staining QC verdicts	Records them per cell, without excluding. See section 5
The batch/group confounding verdict	Carries it **into the cluster statistics**, so a *q*-value cannot be read without it

The result is a phenotype with a measurement behind it: `CD19+ HLA-DR+ CD3- CD14-`, where every call is a percentage of cells above their own sample's threshold, rather than a colour you interpreted.

Run without `--maybe-learn`, explore is blind: it sees the transform and the group labels and nothing else. It names its clusters against cuts it places on each sample's own cells by the declared run's rule (a valley, else a tail gate above the dominant mode), reading none of the declared run's cuts, and calls a cluster debris when most of its cells sit below its file's CD45 debris reference, the 99.5th percentile of CD45 on the events the scatter gate rejects, as the declared CD45 gate defines CD45-negative. A pooled median, used before for the names, calls half of all cells positive for every marker; a 2-means split of cluster medians, used before for debris, cut through CD45-dim cells on PBMC. Blind is the right default, and section 6 explains why.

2. The three ways to run it

Alongside the declared analysis

```
docker run --rm \
  -v "$PWD/data:/data:ro" \
  -v "$PWD/results:/results" \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/panel.yaml \
  --group-column cohort \
  --reference-group "Healthy controls" \
  --cluster \
  --explore \
  --outdir /results
```

Both analyses run. The declared deliverables are byte-identical to a run without the flag; explore's outputs are in `results/explore/`.

Alongside, and linked

```
  ... --explore --maybe-learn
```

The two inform each other, in both directions. See section 6.

Standalone, with no specification at all

```
docker run --rm \
  -v "$PWD/data:/data:ro" \
  -v "$PWD/results:/results" \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --explore-only \
  --outdir /results
```

No `--config`, no `--samples`, no populations. Just a folder of FCS files. This is the mode for a panel you have not written a specification for yet, and `explore_suggested_spec.yaml` is where you get a first draft of one.

3. What it clusters on

Every eligible channel, including scatter and viability. No lineage preference, no exclusions. That is the unsupervised default and it is deliberate: it is what lets the QC gate find debris (CD45 low), dead cells (viability high) and granulocytes (side scatter high, identified by no antibody in most panels).

The cost: some clusters will split on scatter rather than on lineage. If that is not what you want:

```
# cluster on these markers only
--explore-markers CD3,CD4,CD8,CD19,CD56
# or drop the usual suspects
--explore-exclude FSC-A,SSC-A,LiveDead
```

4. The quality gate

cyRAVEN's declared path starts by gating CD45-positive leukocytes. Explore mode cannot: a declared parent gate is exactly what it is meant to work without.

So it uses the cluster-level gate the unsupervised family uses. Cluster coarsely first, then judge **whole clusters** by their marker profile, so no per-event cutoff is invented. The units judged are the SOM nodes (100 on the default 10 × 10 grid), not the merged clusters: a merge can join debris to CD45-dim cells that border it, such as low-density granulocytes in PBMC, and whether it does depends on the seed. Node by node, the same rule keeps those cells at every seed.

Call	Rule
`debris`	Leukocyte marker low across the cluster
`dead`	Viability dye high across the cluster
`saturated`	Top percentile in most channels at once. No real cell type is, so these are aggregates and doublets

Saturation is decided **first**: an aggregate is bright in the leukocyte marker too, so calling it debris would mislabel it.

Under `--maybe-learn` the debris and dead calls are made against each sample's own thresholds. Without it they come from 2-means on the per-node medians, which is what a standalone tool has. Either way `explore_qc_clusters.csv` records the call **and the basis for it** per node, with the merged cluster each node belongs to.

If the gate would keep less than `--explore-min-retained` (default 5%), it is ignored and the run flagged: at that point the markers are more likely mis-named than the data bad.

```
# embed every event, debris included
--no-explore-qc
# keep the cluster marked dead
--explore-keep-dead
```

5. Two things it does that the declared path cannot

It keeps the samples staining QC excluded. A sample with no resolvable CD45 mode has no usable parent gate, so the declared path must drop it: every percentage would be a fraction of an arbitrary slice. Explore does not depend on that gate, so those samples still contribute, and `staining_qc_verdict` is a column rather than a filter. On a cohort where three samples failed, that is three donors you can still say something about.

It looks outside the parent gate. Anything the CD45 gate excluded is invisible to the declared analysis by construction. Explore embeds it.

6. `--maybe-learn`, and why it is off by default

Without the flag the two analyses are computed in complete isolation. That is not caution for its own sake:

- An explore run that used the declared thresholds **is no longer a blind unsupervised run**. If you want to compare cyRAVEN's clusters against another tool's, or check whether the specification is any good, the check has to be independent of the thing being checked.
- A declared run carrying a diagnostic derived from clusters **is no longer the run its manifest describes**.

With the flag, both directions open:

Declared to explore. Per-sample thresholds name the clusters, QC verdicts are recorded, the confounding verdict is attached to the statistics, and the bridge tables against the declared populations are written.

Explore to declared. One file appears in the run directory: `spec_gaps.csv`, with two kinds of row.

`issue`	Meaning
`population spans several clusters`	The declared label lumps distinct phenotypes. Its total can be flat while a subset inside it moves
`cluster no declared population covers`	Something in the blood nobody declared

The first row type is the one worth the flag. A declared population whose total does not differ between groups, while a cluster inside it does, is invisible to any amount of testing on the declared table, and it is a real pattern, not a hypothetical.

7. What it writes

All in `<outdir>/explore/`.

File	Content
`explore_report.html`	**Start here.** Self-contained, every figure and table embedded
`explore_cluster_profile.csv`	Per cluster: size, phenotype string, fraction positive and median per channel
`explore_qc_clusters.csv`	The gate: call and basis per coarse cluster
`explore_cluster_abundance.csv`	Per donor per cluster, with counting uncertainty and LOD/LOQ
`explore_cluster_stats.csv`	Group tests, donor-level, with the confounding verdict attached
`explore_cells.csv`	One row per cell: sample, event index, cluster, UMAP coordinates
`explore_vs_populations.csv`	Cross-tab against the declared labels
`explore_findings.csv`	Clusters no declared population covers
`explore_population_split.csv`	Declared labels spanning several clusters
`explore_suggested_spec.yaml`	A **draft** specification, for curation
`explore_provenance.csv`	Every choice made, and on what basis

Figures: `explore_umap_clusters.png`, `explore_cluster_heatmap.png`, `explore_umap_by_group.png`, `explore_umap_markers.png`, and `explore_marker_umaps_by_group/`, one panel per group per marker, on the shared embedding.

The report is separate from `report.html` rather than a section inside it, for the reason in section 8: `report.html` is a declared deliverable and `--explore` must not change it.

What each cluster corresponds to, and what it is

Two questions, two answers, and they can disagree.

Correspondence is measured by which cells overlap. `explore_cluster_identity.png` puts clusters in rows and declared populations in columns, grouped by the population each cluster best matches. The match is scored by the **F1 statistic**, the harmonic mean of precision (how much of the cluster is that population) and recall (how much of that population is in the cluster), rather than by the largest overlap. The distinction matters: a cluster of 500 cells holding 300 CD4 T cells has CD4 as its plurality even when those 300 are a twentieth of the CD4 T cells in the run. The label would describe the cluster; the cluster would not describe the label. F1 can only be high when both directions hold. The method follows Weber & Robinson, *Cytometry A* 2016;89:1084-1096, who match clusters to manually gated populations this way.

Every cluster has a best match, so the call is qualified rather than asserted: `confident` at F1 ≥ 0.5, `partial`, or `no clear match`, all written to `explore_cluster_identity.csv` with the precision and recall behind them.

The catch-all is ranked apart, because it is the complement of the specification rather than a cell type. Letting it compete would name most clusters "Other" on a specification with modest coverage, which is true but not informative. A cluster at or above 70% catch-all is reported as **undescribed**, which is a gap in the specification rather than a property of the cluster. A cluster half or more catch-all is never called confident, however clean its F1 against the remainder.

Identity is read from the cluster's own marker profile. `explore_cluster_subsets.csv` matches each cluster's positivity fractions against the immune subsets the panel can express (activated, exhausted and homing T cell subsets, NK CD56 bright/dim, HLA-DR low and CD38+ monocytes, and so on) and takes the most specific definition the profile satisfies. The base lineages are candidates too, so a cluster of plain CD4 T cells is named rather than left blank. The `margin` column is the smallest distance from the positivity cut across the requirements that were checked: a cluster 51% positive for a marker satisfies the same requirement as one 99% positive, and only the margin separates them.

This is written whether or not a specification was scored, so an explore-only run still gets it.

The subsets a panel can define are a property of the **panel**, not of immunology. Only definitions whose every marker is present are proposed, and the canonical groups that cannot be reached are named in the log with the markers they would need: naive/central-memory/effector-memory without CD45RA and CCR7, the three monocyte subsets without CD16, senescent T cells without CD28.

The subsets are **not** added to the declared specification. Appending them there re-scores every cell, changes every frequency and legend, and asks a circular question of a subset named after the marker that defines it: a CD69+ subset is 100% CD69-positive by construction. The annotation attaches to the cluster, never to the cell, and no declared output changes.

The heatmap is fractions, not medians

`explore_cluster_heatmap.png` plots the **fraction of each cluster positive** for each marker, not the median expression. A median is a number on a transformed scale whose apparent meaning depends on the colour limits you chose. A fraction positive is "this share of the cluster is above its own sample's cut", which is the same quantity a person reads a gate for.

8. The isolation guarantee

`--explore` must not alter a single byte of the declared deliverables. That is tested rather than asserted: the demonstration cohort is run with and without the flag and every top-level output compared byte-for-byte, excluding only `run_manifest.txt` and `report.html`, which embed a timestamp.

The two new tables in section 9 are the one intended change, and they appear on **every** run, with or without explore.

`spec_gaps.csv` is the only file explore is ever allowed to add to the declared output, and only under `--maybe-learn`.

9. The statistics catalogue

Two files now appear in the normal output on every run, and they exist because a reader arriving from an immunophenotyping paper expects t-tests and ANOVA and finds rank tests instead.

`statistical_methods.csv` lists every commonly reported method: Student's and Welch's *t*, one-way and two-way ANOVA with Tukey, repeated measures, Mann-Whitney, Kruskal-Wallis with Dunn, chi-squared, Bonferroni, and the cytometry-specific diffcyt family (edgeR, limma-voom, GLMM), and for each one says whether this run computed it and **why**. The edgeR form runs with `--edger`.

`normality_tests.csv` carries the evidence: a Shapiro-Wilk test per population per group, and a Brown-Forsythe test for equal variances across groups. Those are the two assumptions a *t*-test or ANOVA rests on.

What it will usually show, and the reason the `interpretation` column exists: Shapiro-Wilk on 4 to 10 donors has almost no power, so a non-significant result is **not** evidence of normality. That is why the rank test is used, and the table says so rather than leaving a reader to conclude "p > 0.05, so a t-test would have been fine".

What does not happen: running every test and reporting all the p-values. At these group sizes that is p-hacking with extra steps.

10. Options

Option	Default	Effect
`--explore`	off	Run explore alongside the declared analysis
`--explore-only`	off	Run **only** explore; no specification needed
`--maybe-learn`	off	Let the two analyses inform each other, both ways
`--explore-k`	20	Metaclusters
`--explore-qc-k`	20	Merged clusters of the coarse QC clustering; the gate judges their SOM nodes
`--explore-grid`	10	SOM grid side
`--explore-neighbors`	30	UMAP `n_neighbors` for the explore embedding
`--explore-min-dist`	0.1	UMAP `min_dist` for the explore embedding, separate from `--min-dist` (0.3)
`--explore-cells-per-sample`	6000	Cells drawn per sample before the gate
`--explore-max-cells`	150000	Ceiling across all samples
`--explore-markers`	every channel	Cluster on these only
`--explore-exclude`	none	Drop these channels
`--no-explore-qc`	off	Embed every single cell, dead and CD45-negative cells included
`--explore-keep-dead`	off	Keep the dead cluster
`--no-explore-equalise`	off	Keep every gated cell instead of equalising per sample
`--explore-min-retained`	0.05	Ignore the gate below this retention and flag the run
`--no-explore-spec`	off	Skip the draft specification

11. What explore mode is not

It is a **hypothesis generator**. A cluster is not a population until it has been declared, thresholded per sample, and checked.

The intended path is a loop: explore finds something to you curate `explore_suggested_spec.yaml` into a real specification to the supervised path scores it with per-sample thresholds, propagated uncertainty and the six specification checks to and if it survives, `--explain-clusters` and `--export-gates` turn it into gate geometry a sorter can be driven with.

Discovery is the beginning of that loop, not the end of it.

Diagnostics, in reading order

Population labels in cyRAVEN derive from a specification declared before the data were examined. That specification is a hypothesis about how the panel behaves across the batch, and it can fail in ways that a frequency table cannot express. The outputs below exist to detect those failures.

Sections 1 to 3 are sufficient to reject most compromised runs and should be read first. `report.html` presents all of them in this order, embedded in one self-contained file.

0. Before the run: `--check`

Every check below is applied to a run that has already happened. Some failures are knowable before it starts, from the FCS headers alone: a file the sample sheet has no row for, a specification naming markers the panel does not contain, a group column that does not exist or has one level, a subject whose rows disagree about that subject.

```
docker run --rm \
  -v "$PWD/data:/data:ro" \
  -v "$PWD/results:/results" \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/analysis.yaml \
  --outdir /results \
  --check
```

It reads keyword blocks rather than events, so it costs seconds regardless of cohort size, and writes nothing. A marker name that does not match `$PnS` exactly is the most common cause of an empty frequency table, and it is the first thing `--check` reports.

Run it after every edit to either input file.

1. Gate inspection

`recon_diagnostics.png` and `gating_qc.png` are written unconditionally and show, per sample, the scatter gate boundary, the singlet band, and every marker threshold superimposed on the density from which it was derived.

A threshold placed on a distribution shoulder rather than a minimum is visible here and is not detectable in any downstream table.

1a. Acquisition stability

Every number reported for a sample is derived from that sample's pooled events, which is correct only if the instrument behaved the same way throughout the acquisition. A partial clog, a bubble or a drift in laser power makes a file two instruments over its run, and one threshold then suits neither half.

`acquisition_qc.csv` bins the Time channel into equal-width intervals and tracks two quantities across them: the event rate, which a clog lowers and a bubble spikes, and each channel's median, which catches a shift the rate does not show. Bins are equal in time rather than in count, because a bin holding a fixed number of events cannot reveal that the rate changed. `acquisition_qc.png` draws the rate per sample with the flagged intervals marked, and `acquisition_qc_bins.csv` holds the per-interval detail.

Nothing is removed. `acquisition_qc_impact.csv` states how far each population would move if the flagged intervals were excluded, and that is the number the decision rests on: a file with a visible anomaly that moves no population by more than its own gate uncertainty does not need re-acquiring. Compare `pct_delta_if_cleaned` against `u_pct_points` for the same population before concluding anything. `--drop-unstable-events` performs the exclusion and is recorded in the manifest.

A file carrying no Time channel is reported as such rather than as stable.

1b. Why a threshold did not resolve

Section 1 shows *that* a marker had no density minimum and got a tail gate. `spreading_receivers.csv` and `spreading_pairs.csv` are what distinguish the two reasons it can have.

Compensation removes the mean contribution of one fluorochrome to another detector. It cannot remove the photon-counting variance that came with it, so a receiver channel's negative population is wider wherever a spilling source is bright. A widened negative fills the density valley a threshold would sit in, and no gating strategy recovers a minimum that the optics erased.

`spreading_pairs.csv` reports, for every ordered source and receiver pair, the ratio of the receiver's negative spread when the source is bright to its spread when the source is dim. `spreading_receivers.csv` ranks receivers by their worst source and pairs that against how often the marker had no density minimum (`fallback_rate`, tail gates and quantiles together).

Read the two columns together. A marker with a high fallback rate and substantial spreading is failing for a panel design reason, and the fix is a different fluorochrome assignment rather than a different threshold method. A marker with a high fallback rate and no spreading is failing for some other reason: too few positive events, a titration problem, or a continuous distribution. Reporting the fallback rate alone cannot tell those apart.

2. Staining QC

`staining_qc.csv` records a per-sample verdict. A sample with no resolvable CD45+ mode contains no usable parent gate; it is excluded and the exclusion reported, rather than contributing a spurious frequency to a group mean. Samples declared as controls through the `is_control` column of the sample map are excluded from testing without being recorded as failures.

`--include-qc-failed` overrides exclusion for inspection.

3. Phenotype concordance

`population_marker_heatmap.png` displays marker intensity against declared populations, z-scored across the run.

In clustering-first analysis this figure assigns identity. Here identity is already declared, so the figure serves the inverse function: a column labelled CD4 T cells that does not show elevated CD4 and depressed CD8 falsifies the gate that produced it. The figure outlines the markers each population's specification requires to be positive, reading the same `populations:` block used for scoring, so the audit cannot diverge from the definitions being audited.

`cohort_composition_heatmap.png` normalises each group to a common notional cell count before partitioning by population, which prevents unequal sample numbers or acquisition depth from generating the pattern.

4. Threshold drift

Per-sample thresholding introduces a specific hazard. If thresholds for a given marker differ systematically between study groups, the populations defined by that marker are not identically specified across the comparison, and any abundance difference is partly definitional.

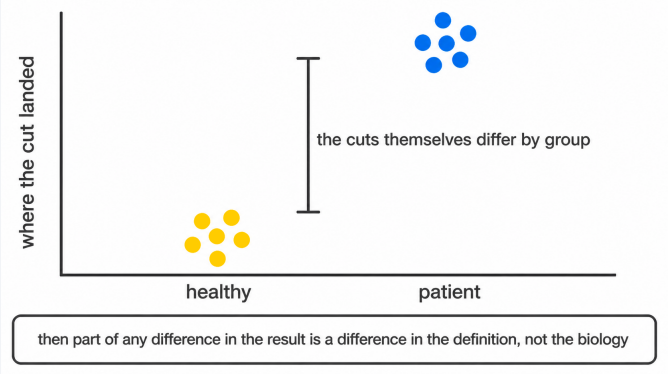


Figure 4.2: Cuts that differ systematically between study groups, making part of any difference definitional

`threshold_drift_stats.csv` tests each marker's per-sample thresholds against group membership and flags those that separate. A flagged marker invalidates naive interpretation of every population depending on it.

No clustering-based method has an equivalent output, because none carries a prior definition capable of drifting.

5. Gate uncertainty

Threshold drift asks whether a cut sits in a different place for one group than another. This asks a prior question: how well is it determined at all.

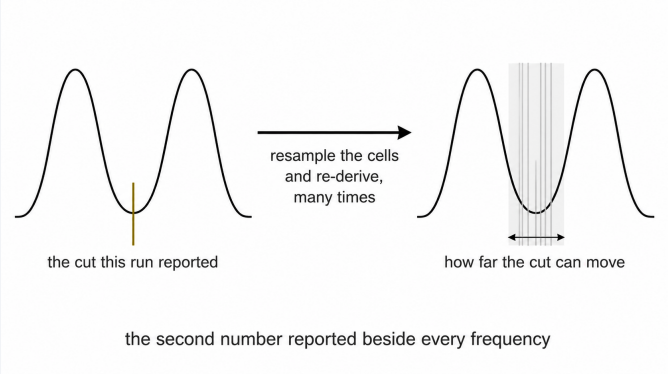


Figure 4.3: Resampling the cells and re-deriving the cut many times shows how far it can move

Each cut is re-derived from resamples of the events it was computed on, and again across the settings that placed it, and the two spreads combine in quadrature following the GUM convention. `threshold_uncertainty.csv` reports both components. `uncertainty_budget.csv` propagates them to each population, with the CD45 parent gate as a term in every one, since it fixes the denominator all frequencies are expressed against.

Read `bootstrap_valley_rate` before `u_combined`. A cut recovered in every resample is a population boundary. One recovered in half of them is histogram noise that happened to clear the depth rule, and the two are written to `thresholds_used.csv` in exactly the same way.

`frequency_uncertainty.png` places the two quantities side by side: the spread between samples, and the bar within each. Where the bars are as wide as the scatter, the variation on display is the cut moving rather than the biology, and `difference_over_gate_u` in the group comparison states the same thing for a tested difference. Below one, the groups differ by less than the distance the threshold itself travels under resampling.

The published figures for manual gating are the comparison worth making. Operator studies following the same convention report expanded uncertainty rising from around 12% on a three-gate strategy to around 16% on a five-gate one, with the first gate contributing most of it (Grant et al. 2021, *Methods Protoc* 4:24).

5a. Detection limits

The preceding section asks how well the cut is determined. This asks whether enough cells were counted for the answer to mean anything, which is a separate question with a separate failure mode: a cut through a wide empty gap is well determined however few events lie beyond it.

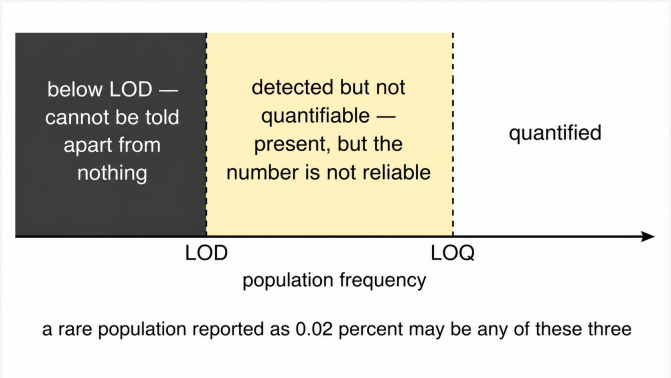


Figure 4.4: Three zones of a frequency axis: below detection, detected but not quantifiable, and quantified

`u_counting_pct_points` in `population_frequencies.csv` is the uncertainty the frequency carries from its own event count, and `u_total_pct_points` combines it with the gate term. `difference_over_total_u` in the group comparison applies the same reading as `difference_over_gate_u` against both components and is the stricter of the two; the two separate exactly where a well-placed cut sits in front of too few cells.

`detection` classifies each value against the conventional twenty and fifty events, expressed as `lod_pct` and `loq_pct` of that sample's parent gate.

`detection_limits.png` counts how many samples clear each limit, per population. Three readings:

A population quantified in every sample is measurable in this cohort.

A population mostly below the limit of quantification is not, and the gating strategy is not the reason. The numerator is small because few cells were acquired, so the remedies are a longer acquisition or a higher `--max-events-per-file`, not a different threshold.

A population split across the limit is the case to be careful with, because a group difference in it can be produced entirely by which samples happened to clear the limit.

Both limits are computed against the parent-gate events this run saw, so subsampling raises them proportionally. That is deliberate: it reports the resolution of the analysis that was actually performed rather than of the files on disk.

6. Cluster concordance

`--cluster` computes a self-organising map clustering of the same cells without reference to the specification, then cross-tabulates. Four configurations are diagnostic.

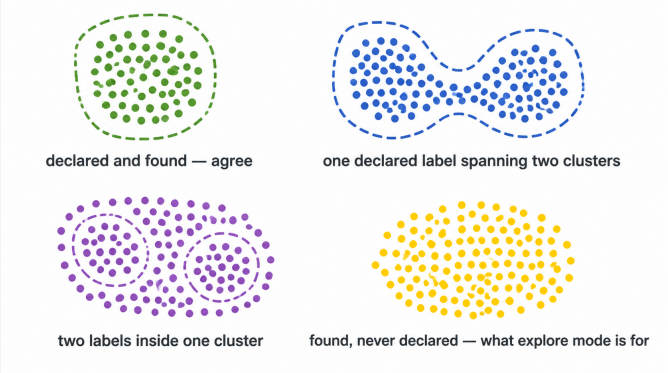


Figure 4.5: Declared gates against found clusters: agreement, splitting, merging, and a cluster nothing declared covers

This check clusters WITHIN the parent gate, on the cells the specification already selected. `--explore` is the wider version: it clusters the single cells of the file over every eligible channel, so it can also find a population the CD45 or viability gate excluded. Under `--maybe-learn` it writes `spec_gaps.csv`, which names declared populations that turn out to span several clusters. See Explore mode.

6.1 Concordant. A cluster composed predominantly of one declared label. The specification and the data agree.

6.2 Undescribed population. A cluster dominated by the unassigned label. A phenotypically coherent population exists that the specification does not describe. Supervised analysis cannot produce this finding, since it has no mechanism for detecting an entity it never declared.

6.3 Under-specified label. One declared label distributed across several clusters. The definition is coarser than the phenotypic structure. Where the constituent subsets respond in opposite directions, the aggregate frequency reports no change.

6.4 Misplaced threshold. A label containing substantially fewer cells than the cluster it dominates. If CD4 T cells score 0.3% while a cluster comprising 20% of events is CD4-bright and unassigned, the population is present and the Boolean rule is rejecting it. This is the diagnosis a frequency table structurally cannot deliver.

Output: `cluster_gate_agreement_clusters.csv`, `cluster_gate_agreement_populations.csv`.

7. Gate transferability

A gate fitted to one cohort is worth having only if it selects the same population in the next one. `--external-labels` takes a cell labelling produced elsewhere, a cyCONDOR clustering for instance, learns a strategy for it, and then withholds one donor at a time: the whole strategy is refitted on the remaining donors and scored on the donor that was held back.

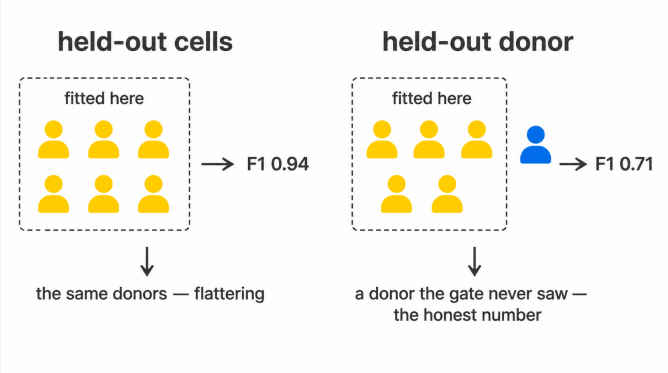


Figure 4.6: Held-out cells score higher than a held-out donor, which is the honest number

This is a different measurement from the held-out metric in section 8. Cells reserved from a fit come from the same donors, acquired in the same tubes on the same day, so they share every source of between-donor variation the gate will meet in use. A strategy can score an excellent held-out F1 on cells and still fail on the next patient.

`gate_transferability_summary.csv` reports the minimum, median, maximum and IQR across donors. Read the minimum. A gate scoring 0.9 on every donor and a gate scoring 1.0 on nine donors and 0.1 on one share a median and are not the same gate.

The join between the label file and this run is on sample and event index rather than row position, because two tools subsample independently and a positional join relabels every cell without raising anything. `join_external_labels()` reports the matched fraction and declines below a hundred cells.

Output: `external_label_gates.csv`, `gate_transferability.csv`, `gate_transferability_summary.csv`, and with `--export-gates`, a Gating-ML 2.0 document in the linear units the FCS file stores.

8. Gate geometry

A cluster index and a cell count are not actionable at the instrument. `--explain-clusters` converts an undescribed cluster into executable gate geometry.

For each qualifying cluster, cyRAVEN selects the two most discriminating markers, fits a convex polygon in that plane, retains the enclosed events, and recurses. The output is the topology of a manual gating strategy and can be reproduced by hand.

Convex polygons rather than rectangles: a conjunction of one-dimensional thresholds is an axis-aligned rectangle, and CD4 against CD8, CD14 against CD16, and FSC-A against FSC-H all separate along oblique boundaries. A rectangle imposed on an oblique boundary must either admit contaminating events or exclude genuine ones. `derive_singlet_band()` is a hand-written instance of the same geometry.

Every reported metric is computed on events held out of the fit, with the resubstitution value printed alongside. Eight free half-planes can memorise a few thousand events, so the difference between the two quantifies overfitting.

The stage is descriptive. A proposed gate localises events in marker space without asserting that they constitute a biological population. It carries no p-value, and scored frequencies and test results are identical whether it runs or not.

Output: `cluster_gate_proposals.csv`, `cluster_gate_polygons.csv`, `cluster_gate_strategy_<k>.png`.

9. Covariates

A variable confounds a comparison only when it both differs between groups and associates with the outcome within the groups. Either condition alone is inert, and flagging either alone would raise an alarm on any study with unequal age distributions.

`confounding_diagnostics.csv` reports both conditions using tests estimable at small *n*, and assigns a `confounder_risk` verdict. The outcome association is a partial Spearman correlation with the group held fixed (a rank *F* test for a categorical covariate): tested across the pooled groups, a real group difference makes any unbalanced covariate track the outcome by itself.

Adjustment is opt-in and kept separate. At single-digit *n* per group, a model carrying group with age and sex expends most residual degrees of freedom on nuisance terms. Where age is strongly associated with group, as in a syndrome-against-adult-control design, the parameters are not separable at any sample size: there are no young controls from which the age effect can be estimated independently. The adjusted estimate extrapolates beyond the observed covariate range and reports a confidence interval that does not reflect this.

`--rank-ancova` fits a rank-based ANCOVA where residual degrees of freedom permit, labelling every row `EXPLORATORY`. Where they do not, the output records `NOT FITTED` with the reason rather than a value that would appear comparable to the others in the folder.

9a. Clinical variables, which are the opposite question

`--covariates` names variables screened as *nuisances*: the question is whether a group difference can be believed in spite of them. `--clinical-columns` names variables that *are* the question: a severity score, a laboratory value, an outcome flag. Each is associated with every population and every marker. The same column would be a different analysis under each flag, which is why they are two flags.

One diagnostic belongs here rather than with the results. `clinical_variables_correlation.png` reports the clinical variables against each other, and is read before the association heatmap. p-values there are adjusted within each variable on the grounds that each variable is a separate question; in a cohort where the sickest patients are also the ones who died, SOFA and 28-day survival describe one gradient, and a population associated with both is one finding counted twice. No correction repairs that, and nothing else in the run would show it.

Read the effect and its bootstrap interval before the asterisk. At the sample sizes these designs have, an interval spanning zero is the usual outcome and is the informative part; `underpowered` marks every test run on fewer than ten samples, where a null result carries almost no information.

10. Batch structure

`--batch-column` names the variable identifying acquisition batch. Absent that, the `$DATE` keyword is used where it varies.

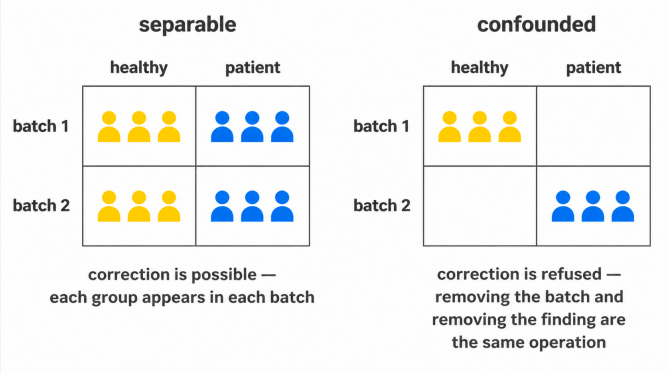


Figure 4.7: A separable design has every group in every batch; a confounded one does not, and correction is refused

Two quantities are reported and both are required.

10.1 Magnitude. `batch_mixing_stats.csv` reports iLISI, the metric against which Harmony is evaluated, together with a permutation null. The null is necessary because the attainable score depends on the number and relative sizes of the batches, so an absolute iLISI is uninterpretable. Permuting batch labels and recomputing yields the score this dataset would produce under no batch structure.

10.2 Separability. `batch_group_confounding.csv` reports Cramér's *V* between batch and study group, and between batch and timepoint where the sheet has one. Where patients and controls were acquired in distinct periods, *V* approaches unity and batch is not distinguishable from the comparison of interest.

10.3 Which channel. The two quantities above are properties of the shared embedding. Neither identifies the marker responsible, and a flagged embedding does not correspond to any action at the bench.

`marker_batch_drift.csv` compares each marker's distribution between batches by Earth Mover's distance, the L1 distance between their empirical quantile functions. It is reported in analysis units as `emd_max`, and divided by the marker's own pooled MAD as `emd_over_mad`, which is the comparable form: a marker displaced by half its own spread is flagged regardless of the scale it sits on. `worst_pair` names the two batches responsible.

`threshold_batch_drift.csv` is the test in section 4 with batch substituted for study group. The two answer different questions and both are needed. A threshold is one number per sample, so it registers only drift that moves the cut; a marker can change its spread, grow a tail, or lose the separation between its modes while the density minimum between them stays exactly where it was. The distributional comparison sees that, and the threshold comparison cannot.

A marker flagged in either is a statement about the assay rather than about the donors, unless batch and study group coincide, which is what 10.2 establishes and should be read first. Where *V* is high, neither table separates a reagent lot from the biology.

Correction proceeds only when both readings permit it. `--correct-batch` aligns each marker across batches by monotone quantile mapping; monotonicity guarantees that within-batch cell ordering is preserved, so location and scale are adjusted without introducing structure. Above `--batch-max-cramers-v` correction is refused, since at that level of confounding removing the batch effect and removing the biological effect are the same operation. The default threshold is Cohen's large association, w = 0.5, which is $V = 0.5$ for a two-level table and smaller for larger ones (0.35 for three levels). `--force-batch-correction` overrides, and the override is recorded in the run manifest.

`--batch-method` selects how the alignment is fitted, and is read only after the refusal has been evaluated, so both methods are refused on identical evidence. A better alignment algorithm does not make a confounded design correctable.

`quantile`, the default, fits one map per marker over the whole file. That is correct only when the batch effect is the same for every cell, which it often is not: a shift in a detector moves a bright population and a dim one by different amounts, so one map fitted to the pooled distribution over-corrects one and under-corrects the other. A per-file map also moves with the biology, so it can remove the difference it was meant to preserve.

`cluster` fits one map per marker per cell type, which is the published remedy and the method CytoNorm implements. Two stages, and the second is not optional: clustering the raw matrix lets a large batch shift become the dominant source of variance, so the clusters turn out to be the batches, each holding one batch with nothing to align against. The clustering is fitted on a whole-file-aligned copy and the per-cluster maps on the original values. Measured on a synthetic three-batch shift, the naive form left a mean between-batch gap of 1.296 against whole-file alignment's 0.003; fitted correctly it reaches 0.014 while leaving the true between-cell-type separation at 3.996 against a true 3.994, where whole-file alignment inflates it to 4.078. `cytonorm` is accepted for `cluster`, with a note in the log: it is not the CytoNorm package, which learns its maps from anchor samples.

The goal each batch is aligned towards. Quantile alignment maps each batch onto a goal distribution. A goal pooled over every cell weights the batches by their cell counts and blends the study groups, so where one batch holds mostly patients and another mostly controls, aligning both to it pulls the groups together. Each study group is therefore aligned on its own, towards the mean of its batches' quantiles, which is CytoNorm's goal (Van Gassen et al. 2020) tailored to the design as CytoNorm 2.0 recommends (Quintelier et al. 2025). On a planted two-batch, two-group example the pooled goal shrank a 2-unit group difference to 1.2; the per-group goal kept 2.0 and removed the batch shift. `--batch-pooled-goal` restores the pooled form.

The clustering reuses the same seeded, stream-safe routine the unsupervised cross-check uses, so no dependency is added and a run reproduces. `batch_correction.csv` records which method ran, what it was judged on, and why, including when correction was refused.

Correction is applied to the shared embedding, the clustering and the gate-cluster concordance. It is not applied to per-sample frequencies, marker medians or the differential tests, which derive from per-sample thresholds and are batch-local by construction. The quantity a batch effect distorts is the single embedding computed across all samples.

11. Conformance

Every check above is internal to one run. This one is not.

The scale check (`scale_outlier` in `thresholds_used.csv`, summarised in `threshold_scale_qc.csv`) compares each threshold against the other samples of the same panel, which identifies a single deviant tube. It cannot identify a cohort that moved as a whole,

because the leave-one-out peer median moves with it. After a laser service, a reagent lot change or six months of drift, every sample can agree with its peers and disagree with the assay as it was validated.

`--write-baseline` records where an accepted run placed each threshold, how variable it was, how often it needed the quantile fallback, and what the populations came out at. `--baseline` measures a later run against that record and writes `specification_conformance.csv` with a verdict per marker and per population.

The baseline holds summaries and the specification text, no event-level or patient data, so it belongs in version control beside the config it describes.

A failure is not a statement that the run is bad. It says the two runs no longer place their cuts in the same place, so their frequencies are not the same measurement and should not be pooled until someone has looked. `--fail-on-drift` turns that verdict into an exit code for scheduled runs, raised after every output has been written.

Cuts are compared in instrument units: each marker's median cut is carried back through this run's transform and forward through the baseline's, because the cofactor and logicle widths are re-derived from every run's data and a changed ruler would otherwise read as drift. Two results are withdrawn rather than reported: a cut whose transform cannot be rebuilt on both sides when the transforms differ, and a redefined population, which is a different population; both read `not comparable`, which is not a small drift.

12. Provenance

`run_manifest.txt` records the R version, platform, the version of every package loaded at run time, the git commit and working-tree state where the code was a checkout, the full invocation, and every option in force. It is written before the expensive stages and rewritten at completion, so an interrupted run leaves `status: failed` rather than an absent or misleading record.

`miflowcyt.md` restates the same run against the ISAC reporting checklist, which several journals check at submission. Its instrument section is read from the FCS keyword block: cytometer, serial number, acquisition software, date, operator and event counts per file, and per panel a detector table giving $PnN, $PnS, voltage, range and whether amplification was linear or logarithmic. Its data-analysis section is read from the run, so it cannot disagree with what was computed.

The two sections a run cannot establish, why the experiment was done and what the cells and reagents were, are marked `TO BE COMPLETED`. An omitted section reads as one that did not apply. Individual keywords follow the same rule and are reported as `not recorded in the FCS file` rather than dropped, since a missing $PnV or $SPILLOVER is usually the explanation for something further down.

`--no-miflowcyt` skips it.

Every output file

Reference for every file a run writes, its columns, and the flag that produces it. Worked examples of the principal figures, rendered from a run on public data, are in the Worked example; this article is the exhaustive list rather than a second copy of them.

1. Quality control

File	Content
`recon_diagnostics.png`	Gate boundaries per sample
`gating_qc.png`	Thresholds superimposed on the densities they derive from
`staining_qc.csv`	Per-sample verdict and exclusion reason, the CD45+ share of live cells and the live share of single cells (`pct_live_of_singlets`, empty where no viability gate was applied); `pct_live_if_dominant_dead` is the share below a dye minimum lying under the dominant mode, the live share if that mode is the dead cells (or empty), for a mostly dead sample the gate reads as mostly live; a second live mode near zero, common on whole blood, also produces one, so read it with the dye's plot
`margin_events.csv`	Per file: margin events removed (or capped), events below a channel's lower limit by flowAI's rule (counted, not removed: in compensated data they are mostly cells bright in a spilling channel, and removing them would bias those cells' frequency), channels linearised from $PnE, and the compensation applied
`thresholds_used.csv`	Threshold, derivation, cofactor and outlier status per sample and marker
`thresholds_within_parents.csv`	Cuts a population with `parent:` derives on its parent's events, beside the CD45+ cut
`threshold_scale_qc.csv`	Per panel and marker: median, range, and how many cuts are scale outliers or fallbacks. Written when any cut is an outlier; the per-sample robust *z* is in `thresholds_used.csv`
`acquisition_qc.csv`, `.png`	Per-sample acquisition stability, and the flagged intervals
`acquisition_qc_bins.csv`	Per-interval rate and channel departure
`acquisition_qc_impact.csv`	How far each population would move if the flagged intervals were excluded
`peacoqc_summary.csv`	Per file, the events PeacoQC removed, the share each of its checks flagged, and a note where the isolation tree was not run (fewer than 150 time bins), written by `--peacoqc`
`fmo_agreement.csv`	Derived cut against its FMO-anchored equivalent, in units of the cut's own uncertainty
`controls_refused.csv`	Channels where a declared control could not anchor a cut, and why
`spreading_pairs.csv`	Per channel pair, how much wider the receiver's negatives become, overall and at a given source level, and `trend_rho`, a screen for compensation or unmixing error (`trend_unresolved` where spreading alone could produce the trend)
`spillover_spreading_matrix.csv`	The published spillover spreading matrix, written by `--single-stains`
`spreading_receivers.csv`	Per marker, total spreading received and whether it explains a fallback
`calibration.csv`	Bead fit per channel, written by `--calibration-beads`

`thresholds_used.csv` gains `override_reason` and `override_by` only when the config declares a `sample_overrides:` block, so a run that overrides nothing writes the table it always wrote. `source` is `manual` for an overridden cut, `fmo_q995` for one anchored to a fluorescence-minus-one control, `fmm_q995` for a tube that leaves out several reagents at once, and `control_q995` for an unstained tube. Those three bare names mean the data indicated no cut (no density minimum: the tail gate or quantile it would otherwise have used is in `derived_source`) and the control supplied one; the `_valley_rejected` suffix means the data did indicate a cut and the control put the boundary higher, and `fmo_q995_valley_lowered`, under `--fmo-sets-cut` only, that a true FMO put it lower. Either way the row carries `derived_threshold` and `derived_source`, the cut that would have been used without the control, so the two stay comparable.

Only the FMO is a full reference. An unstained tube and a minus-

multiple tube both lack the spreading the stained panel puts into a channel, so their negatives are narrower than the sample's and the cuts they anchor are permissive on any channel that receives spreading (Roederer 2001; Maecker & Trotter 2006). `spreading_receivers.csv` names those channels; they are the ones worth an FMO.

What a control has to be before it is used

Two things go wrong with a control-anchored cut, so each declared control channel is checked before it is trusted.

A far quantile finds the edge of the negative population only when the control contains nothing but that population. A tube carrying roughly one percent of bright events puts the 99.5th percentile inside that tail instead. On the minus-multiple data this check was written against, one channel's control sat at 1.0 at the 95th percentile and 5.6 at the 99.5th: a cut there was above the full stain's own 99th percentile, so nothing in any sample would have been positive. The run also places the cut by the negative mode and the spread of the half below it, which a bright tail cannot move, and takes whichever of the two is lower. For a clean control that is the quantile, so nothing changes.

A minus control must also be dimmer in the channel it leaves out. When it is brighter, the tube, the sample sheet or the unmixing is wrong, and anchoring to it raises the cut and deletes a real population. Comparing the two negative modes in units of the sample's own spread catches that. A refused control is listed in `controls_refused.csv` with its reason, and the threshold comes from the sample's own data as it would have without any control.

Keep one consequence in mind when reading `fmo_agreement.csv`. A cut the control supplied has no uncertainty of its own, because the events behind it are in a different tube, so those rows carry `distance` but leave `distance_in_u` empty. The verdict scaled by uncertainty is therefore available for the rows where the sample's own cut was the one applied, which are also the rows where the two methods were free to disagree.

Read `acquisition_qc_impact.csv` against `u_pct_points` for the same population before acting on an acquisition flag. `--drop-unstable-events` performs the exclusion; without it nothing is removed.

These constrain the interpretation of every subsequent file. A run whose gates are misplaced produces internally consistent statistics that are unusable.

2. Uncertainty

File	Content
`threshold_uncertainty.csv`	Sampling and method components per sample and marker, their quadrature sum, the valley's relative depth, and the fraction of resamples that found it
`uncertainty_budget.csv`	Contribution of each threshold to each population's uncertainty, per sample
`frequency_uncertainty.png`	Per-sample frequencies with their uncertainty, one row per population
`uncertainty_budget.png`	The same budget as a stacked median contribution
`detection_limits.png`	Samples per population, split by whether the event count clears each limit

`bootstrap_valley_rate` is the column to read first. A threshold found in every resample is a population boundary; one found in half of them is histogram noise that happened to clear the depth rule, and the two are indistinguishable in `thresholds_used.csv`.

`u_combined` is `NA` rather than zero where it could not be computed, which happens for a threshold taken from a separate control tube. `u_n_terms_missing` on the frequency table counts how many of a population's markers contributed nothing, so a small total that is small only because most terms are absent is distinguishable from a tight one.

Two uncertainties are reported per frequency and they answer different questions. `u_pct_points` is placement: how far the number moves when the cuts behind it move. `u_counting_pct_points` is sufficiency: what it carries from the number of events counted, as the Wilson half-width at one standard deviation. `u_total_pct_points` is their quadrature sum. The two are alike for an abundant population and diverge for a rare one, where a cut through a wide gap is well determined and the count behind it is not.

`lod_pct` and `loq_pct` place the conventional twenty and fifty events on the percentage scale of that sample's parent gate, and `detection` says which side of them the value falls. Both are set by what was acquired rather than by the gating strategy, so `--max-events-per-file` raises them in proportion; `--lod-events` and `--loq-events` change the conventions themselves.

Counting is not a row in `uncertainty_budget.csv`. That table answers which threshold a population's uncertainty comes from, and counting is not a threshold.

Written by default. `--no-uncertainty` skips it and restores the previous output exactly.

3. Conformance

Written by `--baseline`.

File	Content
`specification_conformance.csv`	Per marker: this run's threshold against the baseline's, scaled by the baseline's own spread, with a verdict
`specification_conformance_populations.csv`	The same for population frequencies
`specification_changes.csv`	Populations added, removed or redefined since the baseline

Three verdicts. `pass` is within tolerance, `qualify` is a marker worth looking at before pooling the two runs, `fail` says the cuts are no longer in the same place and the frequencies are not the same measurement.

`not comparable` is a fourth value and means the comparison was withdrawn rather than made: the transform differs from the baseline's, so thresholds are on different scales, or the population was redefined, so it is a different population and a drift statistic would be a category error.

`--write-baseline` writes the reference itself. It contains summaries and the specification text, no event-level or patient data, so it can be version-controlled beside the config it describes.

4. Abundance

File	Content
`population_frequencies.csv`	Percentage of parent per population per sample
`population_marker_mfi.csv`	Per sample, population and marker: the median intensity on the analysis scale (`median_asinh`, which keeps its name under `--transform logicle` and holds the logicle value there), the same median in instrument units (`mfi`), and the percent positive
`population_mfi_summary.csv`	Median fluorescence intensity per population and marker: the median over donors of each sample's median, in instrument units with its interquartile range, overall and per study group, with the median percent positive
`population_composition.csv`	Each population's share of the embedded cells, its defining markers, its spread across donors, its share within each study group (`pct_<group>`, summing to 100 down each group), the same cells as a share of all embedded cells (`pct_of_embedded_<group>`, summing across a row to the population's share), and the percent of it positive for each marker (`pct_pos_<marker>`)
`population_composition_by_timepoint.csv`	The same table for each timepoint, in the pooled table's row order
`functional_markers.csv`	The same restricted to functional marker blocks declared in the config
`population_ratios.csv`	Derived ratios where declared
`gate_counts.csv`	Event counts at each level of the gate hierarchy
`population_frequencies.png`	Pooled composition across samples
`population_marker_heatmap.png`	Marker intensity against declared populations, z-scored
`population_mfi_heatmap.png`	MFI per population and marker, the value printed in each tile, colour scaled per marker; a bar per row gives the population's share
`population_mfi_dotplot.png`	Dot size is the percent positive, colour the scaled MFI
`population_mfi_heatmap_by_group.png`	The MFI heatmap per study group, on one colour scale
`umap_composition.png`	The shared UMAP with a legend of each population's defining markers, share of the embedded cells and that share split by study group with a total row, and a grid of the cells of each population positive for each marker. Every number is a percent of all embedded cells, so a row's group columns sum to its share and no marker tile exceeds it
`umap_composition_<timepoint>.png`	The same figure for each visit, on the pooled map's axes and in its row order

`count` is an event count and scales with acquisition duration. It is not an abundance. Report `pct_of_cd45_pos` unless absolute counts were supplied through `--absolute-counts`.

MFI is a median of medians, in instrument units. Each sample's median comes first and the cohort value is the median of those, so donors are the replicates and the donor with the most events does not set the value. The median is taken on the analysis scale and inverted through the run's transform; a median commutes with a monotone transform, so `mfi` is the median on the instrument's linear scale exactly. Compare `mfi` between runs only when the instrument settings match: it is in the instrument's units, which a voltage change rescales. The heatmap colours each marker separately, between its 1st and 99th percentile across populations, so the colour says "bright for this marker" and the printed number says how bright.

Composition shares sum to 100; frequencies do not. In `population_composition.csv` and `umap_composition.png` each embedded cell has one label, its most specific population, so "CD3 pos" holds only the CD3+ cells no deeper population claimed. `population_frequencies.csv` scores overlapping gates, where "CD3 pos" includes the CD4 and CD8 T cells. The shares are of the embedded cells, which are drawn per sample; `pct_median_per_sample` is the per-donor view. Under `--subsample rare` the shares are weighted back to the true composition.

5. Embedding

File	Content
`cells_umap.csv`	Per-cell coordinates with sample, population and panel
`umap_overview.png`	Embedding by population and by sample
`umap_markers.png`	One panel per marker
`umap_density.png`	Cell density in embedding space
`umap_density_by_group.png`	The same, faceted by study group
`umap_overview_by_group.png`	Combined embedding with one column per group
`marker_umaps_by_group/`	Full-size UMAPs per marker, in its own folder. `umap_CD3.png` pools every sample, and `umap_CD3_by_<category>.png` is written beside it for **every** category present, not only the `--group-column` one. Numeric columns, identifiers and single-level columns are not faceted; past four categories the extras are named in the log
`umap_multigraph_overlay.png`	Per-compartment marker distributions against the reference group
`umap.model`, `umap.model.meta.rds`	Persisted model, written by `--save-umap-model`

A persisted model permits projection of subsequent batches into the same coordinate space, holding cluster positions fixed between runs. Projection is refused where the new data lack markers the model used, since zero-filling would produce coordinates that are geometrically valid and biologically meaningless.

6. Inference

File	Content
`group_comparison_stats.csv`	Abundance between groups, per population
`group_differences.png`	Every population on one pair of axes: the effect against the evidence. The shortlist – read this to decide which panel below to read carefully
`group_comparison.png`	The same as per-sample distributions, one panel per population
`marker_state_stats.csv`	Differential state per population and marker
`marker_state.png`	The same as per-sample distributions
`functional_markers_stats.csv`	Functional blocks between groups
`population_ratios_stats.csv`	Declared ratios between groups
`compositional_clr_stats.csv`	Abundance tests on centred log-ratios
`compositional_concordance.csv`	Classification of each result against both parameterisations
`edger_abundance_stats.csv`	edgeR count-model test of each population's cell counts, each group against the reference, written by `--edger`
`paired_comparison_stats.csv`	Paired designs, written by `--paired-column`
`covariate_adjusted_stats.csv`	Rank ANCOVA, written by `--rank-ancova`
`subcluster_marker_shifts.csv`	Pooled-event marker shifts per compartment
`design_feasibility.csv`	Which group comparisons can be made, and why the rest cannot
`parametric_tests.csv`	The t-test or ANOVA equivalent, with its assumptions recorded
`posthoc_tests.csv`	Pairwise comparisons by three methods, for three or more groups

Every file above except the last three uses one value per sample. `subcluster_marker_shifts.csv` operates on pooled events, carries effect sizes without p-values, and answers a different question from the tests above it.

`group_differences.png`

One point per population: **Cliff's delta** on the x-axis, -log10 p on the y. A fold change of medians is the conventional x-axis and is the wrong one here, it is unbounded, and at single-digit group sizes it is decided by whichever sample sits at the median, so a population whose reference median is near zero produces a fold change of 40 that means nothing. Cliff's delta is bounded [-1, 1], is the effect size the rank test corresponds to, and reads directly: 0.5 means the comparison group was higher in three quarters of the cross-sample pairs.

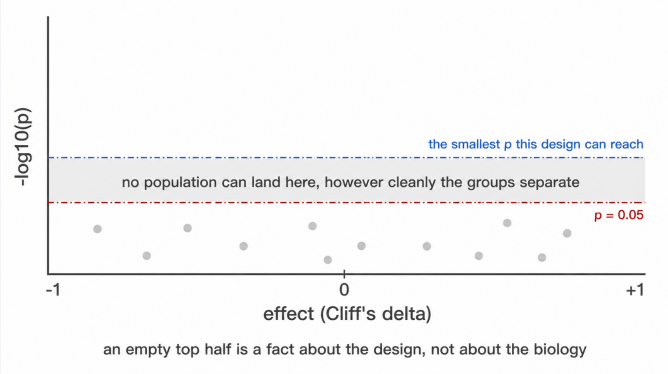


Figure 4.8: A volcano plot whose attainable p floor sits above the 0.05 line, so no population can reach significance

Two horizontal lines. The dashed red one is p = 0.05. The dot-dash blue one is the **smallest p this design can produce at all**: a Wilcoxon rank-sum test has `choose(n1 + n2, n1)` equally likely rank arrangements under the null, so the most extreme possible separation gives a two-sided p of `2 / choose(n1 + n2, n1)` and nothing below it is reachable. At 4 against 5 that floor is 0.016, and after correcting across a dozen populations no population can reach 0.05 however cleanly the groups separate. When the blue line sits *above* the red one, an empty upper region of the figure says nothing about biology, it is a property of the design, and the figure says so rather than leaving the reader to work it out.

The floor is drawn per comparison, not once for the figure: each comparison group has its own size and so its own minimum, and 3 against 6 stops at 0.024 where 6 against 6 reaches 0.0022. With three or more groups the figure facets by comparison and each panel carries its own line.

`design_feasibility.csv`

Read before any of the tests. One row per group, carrying `n_samples`, `n_donors`, `will_be_tested`, `valid_unpaired` and a `reason`.

Two failures it is there to catch. A group smaller than `--min-group-n` is skipped, and a skipped comparison looks exactly like a comparison that ran and found nothing, because both produce no row. And a donor contributing to more than one group, which happens whenever the group column is a timepoint, makes an unpaired test treat repeated measures on one person as independent observations. That second one is the dangerous case, because the test runs and its output is well formed. `will_be_tested` says what the run did; `valid_unpaired` says whether it should have.

`parametric_tests.csv` and `posthoc_tests.csv`

The rank tests remain the primary result. These hold the parametric equivalents most immunology papers report, computed on the logit of each percentage (on log concentrations where the abundance is cells/µL), which stabilises the variance of proportion data. The `transform` column records which.

Two groups give Welch's t-test with Student's alongside it and Cohen's *d*. Three or more give Welch's ANOVA with the classical one-way alongside it and eta squared. `posthoc_tests.csv` then carries Games-Howell, Tukey HSD and Dunn for every pair. Their `p_value` differs in kind: Tukey HSD and Games-Howell refer to the studentized range and are already adjusted over the population's pairs, Dunn's is the unadjusted pairwise p (`p_value_adjusted_over_pairs` says which). `p_adjusted_BH` applies Benjamini-Hochberg within each test across every population and pair.

Every row records `shapiro_p` and `brown_forsythe_p` with the derived `residuals_normal`, `equal_variance` and `assumptions_met`. Where `assumptions_met` is FALSE the rank test is the defensible one, and the `recommended` column says which test to read. Normality is tested on the within-group residuals, not the pooled values, because pooled values are bimodal whenever the groups differ.

Disable with `--no-parametric`.

7. Diagnostics

File	Content
`threshold_drift_stats.csv`, `threshold_drift.png`	Whether thresholds separate by group
`confounding_diagnostics.csv`	Covariate imbalance, outcome association, and verdict
`batch_mixing_stats.csv`	iLISI against a permutation null
`batch_group_confounding.csv`	Cramér's *V* and correction verdict
`batch_diagnostic.png`	Observed against null mixing
`populations_by_batch.png`	Every population's abundance per acquisition batch, one box per batch and a point per sample. `batch_diagnostic.png` asks whether the *embedding* separates by batch; this asks whether the *reported numbers* do, which is the question that decides whether a group difference might be an acquisition difference

File	Content
`clinical_variables_correlation.png`	The clinical variables against **each other**. Read this first, see below
`clinical_association.csv`	One row per population × variable: test, n, the effect with its bootstrap interval, raw and BH-adjusted p, the levels or range tested, and an `underpowered` flag
`clinical_association.png`	Heatmap of the signed effect, populations against variables, with `*` on the tiles that survive correction
`clinical_landscape.png`	The whole cohort as one picture: one column per sample ordered by the first numeric variable, every clinical variable as a strip above, every population as a z-scored row below
`clinical_effects_<variable>.png`	Every population's effect against that variable, ordered, with a 95% percentile bootstrap interval on each
`clinical_<variable>.png`	The data behind one column of the heatmap: a scatter against a numeric variable, a box plot against a categorical one, one point per sample and one panel per population. Where a categorical variable's labels are too long to fit under the panels, such as organism names, each category gets a colour and is named once in a legend, with its sample count
`clinical_association_markers.csv`, `clinical_association_markers.png`	The same against each population's median marker intensity, one test per population and marker (the pairs in `functional_markers.csv`)
`population_trajectories.png`	Written by `--paired-column` with `--condition-column`, and coloured by the first two-level clinical column when there is one: per-patient lines across the conditions with the median over them

The test follows the variable's type rather than being chosen: **Spearman's rho** for a numeric column, **Wilcoxon rank-sum with Cliff's delta** for a two-level column, **Kruskal-Wallis with epsilon-squared** for more levels. Rank methods throughout, for the reason they are used everywhere else here, clinical scores are ordinal by construction, laboratory values are skewed with outliers that are real rather than erroneous, and cohorts are small.

Epsilon-squared is `H / (n - 1)`, the proportion of rank variance the grouping accounts for. It is **unsigned**: a variable with three levels has a magnitude but no single direction, which is why the `signed` column exists and why those tiles stay grey on the heatmap rather than being coloured as though they pointed somewhere.

The effect, its interval, and why they come before the p-value

`ci_low` and `ci_high` are a 95% percentile bootstrap interval on the effect, 2000 resamples of the patient-value pairs for Spearman, stratified within each arm for Cliff's delta, from a fixed seed so the same table always yields the same interval. `clinical_effects_<variable>.png` draws them ordered by effect.

On ten patients most intervals span zero, and that is the finding: a rho of 0.61 whose interval runs -0.1 to 0.9 is a lead worth powering a follow-up on, not a result. The interval is itself approximate at this size, below about nine observations the bootstrap's effective resample is smaller than the nominal one and coverage falls short of 95%, so read a wide interval as *this cohort does not constrain the effect* rather than as a precise range. It is suppressed entirely below six samples rather than quoted at a width nobody should trust.

Clinical variables, written by `--clinical-columns`

A severity score, a laboratory value or an outcome flag is neither the study group nor a confounder. A confounder is screened to decide whether a group difference can be believed; a clinical variable is the question. Name any sheet column with `--clinical-columns`, or `samples: clinical_columns:` in the config , and it is associated with every population and every marker.

Read the variables against each other first

Benjamini-Hochberg is applied within each variable, which treats the variables as separate questions. That holds only when they carry different information. A cohort where the sickest patients are also the ones who died has **one** gradient and two columns describing it, and an association found against both is one finding reported twice. `clinical_variables_correlation.png` is where that is visible: circle area is the absolute Spearman coefficient, fill is its sign, and the number is printed where the unadjusted p-value is below 0.05. Two-level variables are included coded 0/1, for which Spearman is the point-biserial correlation of the ranks (not the rank-biserial: two perfectly separated groups of five give 0.87, not 1); variables with three or more unordered levels are excluded and named in the caption, because there is no ordering to correlate and coding them 1/2/3 would invent one.

Benjamini-Hochberg is applied **within each variable**, across the populations tested against it, because each variable is its own question asked of every population. Pooling across variables would penalise a well-powered variable for the company it keeps.

Two things this is not. It is not **survival analysis**: a 28-day flag is tested as the two-group comparison it is, and no time-to-event model is fitted because the sheet carries no follow-up time. And it is not a **claim about cells**: every test runs on one value per sample, so the replicates are subjects. Correlating a score against tens of thousands of events would treat one deeply acquired patient as tens of thousands of observations.

Read the effect before the asterisk. On a small cohort these tests detect only very large effects, so a null result says little, and the `underpowered` column marks every test run on fewer than ten samples. | `marker_batch_drift.csv` | Earth Mover's distance between batches per marker, in analysis units and scaled by the

marker's own MAD | | `threshold_batch_drift.csv` | The threshold test above, grouped by batch instead of by study group | | `batch_correction.csv` | Whether a correction ran, which method fitted it, the Cramér's *V* it was judged on, and the reason. Written whenever `--correct-batch` is set, including on a refusal |

The last two are written by `--batch-column` and answer the question the first three cannot: which channel moved. iLISI is a property of the embedding, so a flagged run names nothing actionable, whereas a flagged marker names a reagent lot or a detector.

Read `emd_over_mad` rather than `emd_max`: the raw distance is in the units of whichever marker it describes, and dividing by that marker's own spread is what makes two of them comparable. `worst_pair` names the batches responsible.

Both tables are needed. A threshold is one number per sample and registers only drift that moves the cut, while a marker can change its spread or lose the separation between its modes with the density minimum between them unmoved.

File	Content
`run_manifest.txt`	Versions, commit, invocation, options

8. Clustering

Written under `--cluster` and `--explain-clusters`.

File	Content
`unsupervised_clusters.csv`, `.png`	Cluster assignment per cell
`cluster_gate_agreement_clusters.csv`	Dominant label and purity per cluster
`cluster_gate_agreement_populations.csv`	Recovery of each declared population
`subcluster_k_selection.csv`	Silhouette-selected k, from `--auto-subcluster-k`
`cluster_gate_proposals.csv`	Learned gate geometry with held-out metrics
`cluster_gate_polygons.csv`	Polygon vertices in transformed units
`cluster_gate_strategy_<k>.png`	One figure per explained cluster

9. External labels

Written by `--external-labels`, and by `--export-gates` for the last two.

File	Content
`external_label_gates.csv`	Learned strategy per supplied label, with held-out metrics at each depth
`external_label_polygons.csv`	Polygon vertices on the analysis scale
`gate_transferability.csv`	Precision, recall and F1 on each donor, from a strategy refitted without that donor
`gate_transferability_summary.csv`	Minimum, median, maximum and IQR of F1 across donors
`external_label_strategy_<label>.png`	One figure per label
`*.gatingml.xml`	ISAC Gating-ML 2.0, linear units, levels chained parent to child
`*_polygons_linear.csv`	The same vertices as a table, for redrawing by hand

Read `f1_min` from the summary, not `f1_median`. A gate scoring 0.9 on every donor and a gate scoring 1.0 on nine donors and 0.1 on one have the same median and are not the same gate.

The Gating-ML vertices are subdivided along each edge. An edge that is straight on the analysis scale is a curve in the linear units the FCS file stores, so a polygon built from the corners alone would describe a different region from the one that was fitted and validated. Dimensions are named by marker symbol, which is what cyRAVEN resolves from $PnS.

10. Auxiliary

File	Content
`run_manifest.txt`	R and package versions, git commit, invocation, options, run status
`miflowcyt.md`	The same run against the ISAC reporting checklist
`report.html`	Every output above embedded in one self-contained file, in the order this documentation says to read them. Written for failed runs too, with a diagnosis
`patient_metadata_english.csv`	Patient table after column mapping and value translation
`absolute_counts.csv`, `absolute_counts.png`	Measured cells per microlitre per sample and population, written by `--absolute-counts`
`absolute_counts_raw.csv`	The supplied workbook flattened to CSV exactly as read, before any header or population-name interpretation. Open this first when a population from `--absolute-counts` looks wrong: it separates a reading problem from a mapping one
`absolute_counts_stats.csv`	Between-group tests on the measured counts, on `cells_per_ul`. Written when the run performs group tests, so it needs `--group-column` and is absent under `--no-group-tests`
`absolute_counts_qc.png`	The supplied counts, one log-scale tile per sample and population, controls and QC failures labelled, same flag. Written unconditionally whenever counts are supplied, and deliberately before the group-comparison figure, because a unit or mapping error has to be visible before the more inviting figure is read as a finding
`populations_unavailable.csv`	One row per sample and population the panel cannot score, with the reason. Written only when there is at least one
`flowjo/`	UMAP-annotated FCS, written by `--flowjo-export`

File	Content
config_derived.yaml	Derived parameters, written by --write-config
sample_map_template.csv	Written by --write-sample-map
gate_adjustments.csv	One row per hierarchy gate --auto-fix-gates skipped and per population cut it replaced, with threshold_not_applied, pct_of_parent_kept and the reason. Empty without the flag

Gate adjustments

Written only under --auto-fix-gates, and the one table to read before any frequency from a run that used it, because a skipped gate changes every count beneath it.

Column	Meaning
sample_id, gate, marker	Which cut, in which sample
threshold_source	Why it was eligible. quantile_fallback means no density minimum was found, so the cut came from a percentile
threshold_not_applied	The number that would have been used. Kept so the decision is auditable rather than merely reported
action	skipped, with the parent carried through unchanged; for a population cut, replaced by and the new value
pct_of_parent_kept	What the gate retains after the decision. 100 when it was skipped
reason	The rule that fired, in words

Two rules put a gate here. A threshold that came from quantile_fallback has its retention fixed by the fallback constant rather than by the data, so it keeps the same share of the parent whatever the sample holds. And a gate retaining less than 5% of its parent is misplaced whatever its source says, because a viability or CD45 gate that discards nineteen cells in twenty is not measuring what it claims to.

Population markers are never skipped: a quantile fallback on CD19 is a poor threshold, but "no threshold" does not mean "every cell is a B cell". They borrow instead (--peer-cuts, which --auto-fix-gates implies). Every sample of a panel is on one pooled transform, so where most of them found a density minimum for the marker, the median of those minima is a measured cut on this sample's scale too, and a sample without one – a quantile_fallback or a tail-rule cut – takes it. Where most samples found none, as for a rare marker, the tail rule stands, and only a fallback or a tail cut more than 3.5 robust SDs from the other samples' cuts takes their median. The row in thresholds_used.csv then has source peer_median, keeps needs_review, and holds the original cut in derived_threshold. The same applies to the functional-block markers, whose positive fractions use these cuts. Cuts from a control, a config entry or a manual override are never replaced.

10d. Externally measured cell counts

Written by --total-counts. A share is constrained to sum to 100, so a frequency table cannot distinguish one population expanding from every other contracting. A measured yield per acquisition is what separates the two.

File	Content
total_counts.csv	The yields as parsed and matched: patient_raw, timepoint, total_cells, sample_id. One row per acquisition that matched
total_counts_raw.csv	The sheet flattened exactly as read, before any header or block interpretation. Open this first when a number looks wrong: it separates a reading problem from a matching one
total_counts_qc.png	**Read this before anything derived from it.** The yields on a log axis. Everything downstream inherits their error, and a yield in the wrong unit lands decades off the median here while staying invisible in the derived table
absolute_vs_share.png	The two measures side by side per population. Where a global depletion is visible as the thing a share cannot express
absolute_vs_share_by_timepoint.png	The same, once per visit, under --split-by-timepoint
explore_total_counts_qc.png, explore_absolute_vs_share.png	The twins of those two inside explore/
explore_cluster_stats_absolute.csv	Inside explore/: the donor-level cluster tests on cells per cluster, kept apart from the share tests in explore_cluster_stats.csv
explore_cluster_count_concordance.csv	Inside explore/: per cluster, whether the two agree. absolute only is a compartment that changed size with its composition unchanged; share only is a redistribution

population_frequencies.csv gains total_cells, cells_absolute and count_basis; where the sheet carries blood_volume_ml it gains cells_per_ml as well, which is the column to read across timepoints because the draws differ in volume.

count_basis records the route on every row that carries one. Treat cells_absolute as **dual-platform**: one measurement from this run multiplied by one from an instrument it never saw. Published inter-laboratory CVs for that route are roughly 20-33%, against 10-16% for single-platform bead counting, so a difference smaller than that is unresolved. Shares are never overwritten.

10e. Clinical variables

Written by --clinical-columns. One figure per variable, plus the effects across populations.

File	Content
clinical_<variable>.png	One figure per variable named to the flag, showing the association with every population
clinical_effects_<variable>.png	The effect sizes for that variable with their bootstrap intervals, populations ordered by effect

File	Content
clinical_association.csv	Every variable against every population: the test chosen from the column's type, estimate, effect, ci_low, ci_high, p_value, p_adj_BH, and underpowered
clinical_association_markers.csv, .png	The same against marker expression within each population rather than abundance
clinical_variables_correlation.png	**Read this first.** p-values are adjusted within each variable on the assumption that the variables are separate questions, and this is where you see whether they are
clinical_landscape.png	Every clinical variable and population on one grid, columns ordered by timepoint then study group
clinical_timepoint.png, clinical_infection_focus.png	A categorical clinical column gets the same treatment as a numeric one, with the test following the type

n_patients and repeated_measures on every row say what the test actually had: a variable that is a property of the donor is tested at the donor level even when several acquisitions carry it, because the unit of replication is the donor.

10f. The figure set per visit

Written by --split-by-timepoint into by_timepoint/<level>/, plus the paired views that need every visit at once and so stay in the run directory.

File	Content
by_timepoint/<level>/	The whole figure set again, restricted to that visit
timepoint_trajectories_pct_of_cd45_pos.png	One line per patient across visits, on share
timepoint_trajectories_cells_absolute.png	The same on absolute cell number, with --total-counts
timepoint_trajectories_cells_per_ml.png	The same per millilitre, where blood_volume_ml is supplied. This is the one to read across visits, because the draws differ in volume
timepoint_subset_balance.png	Subset composition by visit
timepoint_marker_intensity.png	Marker expression by visit
functional_markers_by_timepoint.png, population_ratios_by_timepoint.png	Per-visit twins of the pooled figures. The ratio figure shows only the ratios the config declares
population_ratios_by_cell_type_by_timepoint.png	Every population as a share of CD45+, one panel per cell type, visits on the x axis. No test is drawn, because the visits are the same patients
umap_markers_<level>.png	The marker grid restricted to one visit, on the same embedding

The embedding, the thresholds and the statistics are **not** recomputed per visit: only the rows drawn are restricted, so a position on one visit's UMAP is the same position on another's and a threshold is the same cut.

10g. What explore writes back

File	Content
explore_cluster_identity.csv, .png	Which declared population each cluster corresponds to, scored by F1 with precision and recall beside it, plus pct_catch_all and a call
explore_cluster_subsets.csv	The immune subset each cluster's own marker profile matches, with the markers that decided it and the margin over the runner-up
explore_cluster_median_heatmap.png	Scaled median expression per cluster, the direct equivalent of what a tool with no gating step produces
spec_gaps.csv	Under --maybe-learn: what the specification does not cover, one row per issue with subject, detail and action
suggested_config_next_run.yaml	The declared specification plus a draft entry per uncovered cluster, in runnable config format. A draft: the names are placeholders and nothing in the run that writes it uses it

The `--absolute-counts` input

A frequency is a proportion of a parent gate, so it moves when any other population moves. An absolute concentration does not, which is why bead-based or volumetric counts are worth carrying through the run when the laboratory measured them.

The flag takes a CSV or Excel sheet laid out as the counting instrument exports it: the first row is a header, the first column identifies the sample, and every further column whose header is non-empty and whose body is numeric is read as a population. Blank rows are skipped, so cohort separators do not need removing.

```
sample,Granulocytes,Monocytes,Lymphocytes
HC-13;2.fcs,3810,420,1650
HC-14;1.fcs,2990,395,1880
```

Units come from any header cell that is not itself a population column. A label containing microlitre, `uL` or `µL` is read as cells per microlitre; one containing `mL` or millilitre is converted by dividing by 1000. When no unit label is present the values are assumed to be cells per microlitre already and the log says so; verify that before trusting the figure.

Each row is matched to a `sample_id` by `patient_id` first and then by acquisition filename, both insensitive to case, surrounding whitespace, a `.fcs` extension and a trailing `copy`. The filename match is a suffix match with a delimiter required immediately before it, because export sheets routinely omit the batch prefix the acquisition filename carries. A key matching more than one file is treated as no match rather than guessed at, and every unmatched row is named in the log. `--absolute-counts` needs `--sample-map`, since that is where the join key lives.

`miflowcyt.md` completes the two checklist sections a run can establish. The instrument section comes from the FCS keyword block: cytometer, serial, acquisition software, date, operator and event counts per file, then one detector table per panel with `$PnN`, `$PnS`, voltage, range and amplification mode. The data-analysis section comes from the run, so the compensation statement, the transform and its parameters, the gate hierarchy, the population specification and the tally of how the thresholds were obtained all describe what was actually computed.

Experiment intent and specimen biology are marked `TO BE COMPLETED`, and a keyword the file does not carry is written as `not recorded in the FCS file` rather than dropped. Both follow the same rule: an absent line reads as a line that did not apply, and neither of those is true here.

`report.html`

The reading order made navigable, and the one file worth sending to someone else. Each section states what it reports and how to read the outputs under it, and a run that excluded samples says so above every result.

It is **self-contained**. Every figure is embedded at full resolution and every table in full, so it can be moved, attached to an email or archived on its own; it references no other file, loads no font or script from a network, and works from a `file://` path. A report whose images live beside it becomes a page of broken icons the moment it is moved, and a result that cannot survive being moved is not a record.

Feature	Behaviour
Numbering	Every section is numbered in reading order, and every figure and table within it (4.1, 4.2, …), on its tab, caption and contents entry. Numbers are assigned after the sections a run did not write are dropped, so they have no gaps
Search	A bar pinned to the top finds sections, figures and tables by title, name, description and column name, and tables by the values in their rows; a row match opens the table already filtered. / or Ctrl+K focuses it, arrow keys move, Enter opens
Sections	Collapsible, with a sidebar indexing every section, figure and table, and expand/collapse-all
Sidebar	Drag its right edge to resize, or focus it and use the arrow keys. The width is remembered for next time
Back to top	A button at the bottom right once you are a screen down, since the report runs to tens of screens
Figures	Uniform display box whatever the native aspect ratio, with a one-line description underneath; click to zoom, with keyboard +/- and Escape; download the original PNG at full resolution. A figure is decoded only when its tab is shown
Small screens	Below 900px the contents list becomes a drawer behind the button in the search bar, and figures and table controls fit the width
Tables	Searchable across all columns, sortable by clicking a heading, shown 10, 50, 100 or all rows, scrollable with a sticky header
Export	Any table exports to CSV exactly as filtered and sorted, so what you export is what you were looking at
Completeness	Every figure and table the run wrote appears; anything no section names is collected under “Further outputs” rather than omitted

Its size is the sum of the figures it carries, once each: tens of megabytes on a full run (59 MB for one 24-sample cohort with 108 figures, which was 108 MB while each figure was stored twice). Opening it costs less than its size suggests, because no figure is decoded until it is shown. `--report-images link` writes a report of a few megabytes that references the PNGs beside it instead of carrying them; it opens at once but shows its figures only while it stays in the results directory, and its tables are still embedded. A table larger than 8 MB is named with its row count instead of being embedded, which in practice means only the per-cell exports; the limit is `options(cyRAVEN.report_table_max_mb = )`.

A run that fails writes it too. The diagnosis comes first: the error, the stage the run reached, the log leading up to it, what the message means for the data rather than for R, and the next action. Everything produced before the failure is embedded below, because that partial output is usually where the evidence is. A `gating_qc.png` written before the failure often shows the cause directly.

`--no-miflowcyt` and `--no-report` skip those two. `--write-config`, `--write-samples` and `--write-sample-map` terminate after writing and cannot be combined with a full run, and `--check` validates the inputs and exits without writing anything.

The FlowJo export writes one pooled file, one per sample and one per group. Open `_ALL_SAMPLES.fcs`, plot UMAP-1 against UMAP-2, and decode the Population, SampleID and CohortID parameters using `population_codes.csv`.

10a. The statistics catalogue

Written on every run, grouped or not.

File	Content
`statistical_methods.csv`	Every commonly reported method in the immunophenotyping and cytometry literature, whether this run computed it, and why
`normality_tests.csv`	Shapiro-Wilk per population per group, and Brown-Forsythe for equal variances across groups

These exist because a reader arriving from a paper that reports *t*-tests and ANOVA finds rank tests here instead. `statistical_methods.csv` names Student’s and Welch’s *t*, one-way and two-way ANOVA with Tukey, repeated measures, Mann-Whitney, Kruskal-Wallis with Dunn, chi-squared, Bonferroni and the diffcyt family, and states what was done about each.

`normality_tests.csv` is the evidence for that choice. Read its `interpretation` column rather than `shapiro_p` alone: Shapiro-Wilk on 4 to 10 donors has almost no power, so a non-significant result is **not** evidence of normality, and the column says so. That is why the rank test is used.

What does not happen is running every test and reporting all the p-values. At these group sizes, choosing among six p-values per population is p-hacking with extra steps.

10b. `explore/`: unsupervised discovery

Written only under `--explore` or `--explore-only`, and only into this subdirectory. Nothing outside it changes.

File	Content
`explore_report.html`	Self-contained report, separate from `report.html`, numbered and searchable in the same way
`explore_cluster_profile.csv`	Per cluster: size, phenotype string, fraction positive and median per channel
`explore_cluster_mfi.csv`	MFI per cluster and marker: median over donors of each sample's cluster median, in instrument units with its interquartile range, overall and per group
`explore_cluster_composition.csv`	Each cluster's share of the cells after the QC gate, with its phenotype, matched subset, spread across donors and share per group
`explore_umap_composition.png`	The explore embedding with a legend of each cluster's subset, phenotype and share
`explore_cluster_mfi_heatmap.png`, `explore_cluster_mfi_dotplot.png`	The cluster MFI as a value-printed heatmap and as a dot plot, as for declared populations
`explore_qc_clusters.csv`	The cluster-level gate: `call` and the `basis` for it
`explore_cluster_abundance.csv`	Per donor per cluster, with counting uncertainty and LOD/LOQ
`explore_cluster_stats.csv`	Donor-level group tests, carrying the batch/group confounding verdict
`explore_cells.csv`	One row per cell: sample, event index, cluster, UMAP coordinates
`explore_vs_populations.csv`	Cross-tab against the declared labels
`explore_cluster_identity.csv`	Per cluster: the declared population it best corresponds to, the precision and recall behind that match and their harmonic mean (F1), the runner-up, and the share of the cluster that is the catch-all. A cluster at or above 70% catch-all is reported as undescribed
`explore_cluster_subsets.csv`	Per cluster: the immune subset its own marker profile matches, the requirements checked, and the margin – the smallest distance from the positivity cut across them. Written whether or not a specification was scored
`explore_findings.csv`	Clusters no declared population covers
`explore_population_split.csv`	Declared labels spanning several clusters
`explore_suggested_spec.yaml`	A draft specification, for curation
`explore_provenance.csv`	Every choice explore made, and on what basis

Figures: `explore_cluster_identity.png`, `explore_umap_clusters.png`, `explore_cluster_heatmap.png`, `explore_cluster_median_heatmap.png`, `explore_umap_by_group.png`, `explore_umap_markers.png`, and `explore_marker_umaps_by_group/`.

`explore_cluster_identity.png` is the one to read first: clusters as rows, declared populations as columns, grouped by the population each cluster best matches, with the subset its marker profile matches in the row label. The match is scored by F1 rather than by the largest overlap, because a cluster holding 300 of a run's 6,000 CD4 T cells has CD4 as its plurality while accounting for almost none of that population. See Explore mode.

The last four tables appear only when a specification was scored **and** `--maybe-learn` is set. Without that flag the two analyses are computed in isolation and neither sees the other; see Explore mode.

`spec_gaps.csv` is the one file explore may add to the run directory itself, and only under `--maybe-learn`. It names declared populations spanning several clusters and clusters no population covers.

10c. `by_timepoint/`: the figure set per visit

Written under `--split-by-timepoint`, when the sample sheet carries a `timepoint` column with two or more levels of at least two samples each. One subdirectory per level, each holding the same figures the pooled run writes, drawn over that visit's samples alone.

File	Content
`by_timepoint/<level>/population_frequencies.png`	Abundance, that visit only
`by_timepoint/<level>/population_marker_heatmap.png`	Phenotype check, that visit only
`by_timepoint/<level>/marker_state.png`	Positivity per population, that visit only
`by_timepoint/<level>/umap_overview.png`	The shared embedding, that visit's cells
`by_timepoint/<level>/umap_markers.png`	Marker grid on the shared embedding
`by_timepoint/<level>/umap_density.png`	Density per sample
`by_timepoint/<level>/functional_markers.png`	Functional blocks, that visit only
`by_timepoint/<level>/population_ratios.png`	Declared ratios, that visit only
`by_timepoint/<level>/absolute_vs_share.png`	Written when an external total was supplied
`by_timepoint/<level>/frequency_uncertainty.png`, `detection_limits.png`	Counting uncertainty, that visit only

The embedding, the thresholds and the statistics are not recomputed per visit. Only the rows drawn are restricted. A position on one visit's UMAP is therefore the same position on another's, and a threshold is the same cut. A difference between two of these directories is a difference in the cells, not in the method. Re-running the pipeline per visit would instead produce embeddings computed from different cells, between which no position is comparable, and re-derive the thresholds, making "CD4-positive" a different predicate at each visit.

No test is annotated on these figures. Splitting the cohort by visit and again by study group leaves too few samples per comparison, and a bracket computed on the pooled cohort drawn beside one visit's data would attribute a cohort-level result to a subset of it.

The paired views (`timepoint_trajectories_*.png`, `timepoint_subset_balance.png`, `timepoint_marker_intensity.png`) stay in the run directory and are written whether or not the flag is set. They need every visit at once, so splitting them by visit is what destroys them.

11. Failure handling

Analyses beyond the core pipeline execute after the primary outputs are written and are individually wrapped. A failure logs a warning naming the analysis and leaves all other output intact. An addition capable of destroying the result it augments is not an addition.

Statistics

```
library(cyRAVEN)
```

Every test here trades assumptions against power. This article states which trade was made at each point, and the sample size at which it should be revisited.

1. Aggregation

1.1 Pseudoreplication

The event count in an FCS file is determined by acquisition duration, not by study design. Treating events as independent observations sets the degrees of freedom from an operational parameter: sixteen donors contributing one million events each yield a nominal *n* of 1.6 × 107.

Two consequences follow. Statistical power becomes a function of instrument time, so extending acquisition on one tube increases apparent significance. More seriously, a single donor acquired to greater depth contributes disproportionate weight, and inter-individual variation in that donor is indistinguishable from a group effect.

1.2 Implementation

Every test aggregates to one value per sample per population before estimation, following Weber et al. (2019, *Commun Biol* 2:183). The degrees of freedom reflect the number of donors.

```
ids <- sprintf("S%02d", 1:12)
mfi <- expand.grid(
  sample_id = ids,
  population = "CD4 T cells",
  marker = "HLA-DR",
  stringsAsFactors = FALSE)
mfi$n_cells <- 500
mfi$median_asinh <- rnorm(12, 2, 0.3) +
  rep(c(0, 0.8), each = 6)
mfi$pct_positive <- pmin(100, pmax(0,
  rnorm(12, 30, 5) +
    rep(c(0, 10), each = 6)))
mfi$is_control <- FALSE
mfi$qc_status <- "pass"
grp <- setNames(
  rep(c("HC", "Patient"), each = 6), ids)

res <- stats_marker_state(
  mfi, grp, reference = "HC")
res[, c("marker", "measure", "n_reference",
        "n_comparison", "delta_median",
        "p_value")]
```

```
#>   marker      measure n_reference
#> 1 HLA-DR median_asinh           6
#> 2 HLA-DR pct_positive           6
#>   n_comparison delta_median     p_value
#> 1            6     1.025903 0.002164502
#> 2            6    13.592904 0.008658009
```

`n_reference` is 6, the number of control donors, not 3,000, the number of events they contributed.

Event-level quantities remain available through `stats_subcluster_shifts()`, which reports effect sizes without p-values. They are hypothesis-generating.

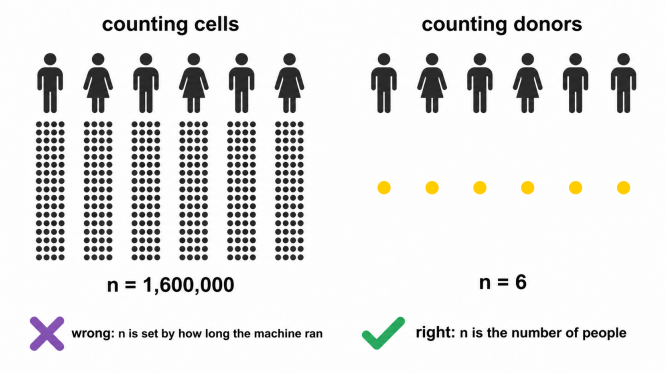


Figure 4.9: Counting cells sets n from how long the machine ran; counting donors sets it from the number of people

2. Test selection

Group comparisons use Wilcoxon rank-sum and Kruskal-Wallis, which assume independence and ordinal comparability.

The principal alternative, implemented in diffcyt, is limma's moderated *t*, which stabilises variance estimates through an empirical Bayes prior shared across markers. At mass cytometry marker counts of 30 to 40 and designed sample sizes, this is the more powerful choice.

At a dozen markers and single-digit donors per group the prior is estimated from too few markers to stabilise, and the normality assumption becomes load-bearing at precisely the sample size where it cannot be assessed.

Cost. Reduced power under normality, and no direct mechanism for covariate adjustment.

Revisit at. Approximately fifteen samples per group, where the prior becomes informative and mixed models can accommodate repeated measures.

2a. The parametric tests, and why they are reported separately

Most immunology papers report a *t*-test or an ANOVA, and a reviewer may ask for one. Declining to compute it does not remove it from the paper; it moves the calculation somewhere with no record of whether its assumptions held.

So `parametric_tests.csv` carries the parametric equivalent of every comparison, and `group_comparison_stats.csv` is unchanged. Two files, two answers, and the columns that decide between them sit on the parametric row.

The transform. Frequencies are proportions. They are bounded, their variance depends on their mean, and near 0 or 100 they are strongly skewed, which is exactly where rare populations live. The parametric tests therefore run on the empirical logit, log((p + e) / (1 − p + e)) with e the smallest non-zero proportion so that 0% and 100% stay finite. It is symmetric about 50% and stabilises the variance near 0 and 100%, where the arcsine square root, the older remedy, does not. Group means are reported back on the percentage scale, because a difference in logits means nothing to a reader.

Which test. Two groups give Welch's *t*-test, which does not assume equal variances, with Student's *t* beside it. Three or more give Welch's ANOVA with the classical one-way beside it. Welch's is the default in both cases because equal variance is an assumption, not a property, and the cost of Welch's when variances happen to be equal is small.

The assumptions, on every row. `brown_forsythe_p` tests equal variance, using deviations from the median rather than the mean so it does not itself assume normality. `shapiro_p` tests normality of the **within-group residuals**, not the pooled values, which are bimodal whenever the groups differ and would fail the test for the wrong reason. `assumptions_met` combines them and `recommended` names the test to read.

Post-hoc. With three or more groups, `posthoc_tests.csv` carries all three standard families: Games-Howell, which uses Welch's standard error and pair-specific degrees of freedom and so pairs with Welch's ANOVA; Tukey HSD, which assumes equal variances and pairs with the classical ANOVA; and Dunn, the rank equivalent that follows Kruskal-Wallis. All three are written, adjusted within each family, and the assumption columns say which to read. Choosing after seeing the p-values is the thing this arrangement is designed to make visible.

What is still not here. Mixed models and GLMMs are the right tool for repeated measures and are not implemented; `--paired-column` with `--condition-column` covers the two-timepoint case and nothing wider. Beta regression, which models the bounded support directly rather than transforming it, is the better answer at larger sample sizes. Both become worth the dependency at roughly fifteen donors per group.

3. Compositionality

3.1 Constraint

Population percentages within a sample are constrained to sum to 100 and cannot vary independently. A genuine granulocyte expansion mechanically depresses every lymphocyte percentage. Those depressions test significant while absolute lymphocyte counts per microlitre remain unchanged.

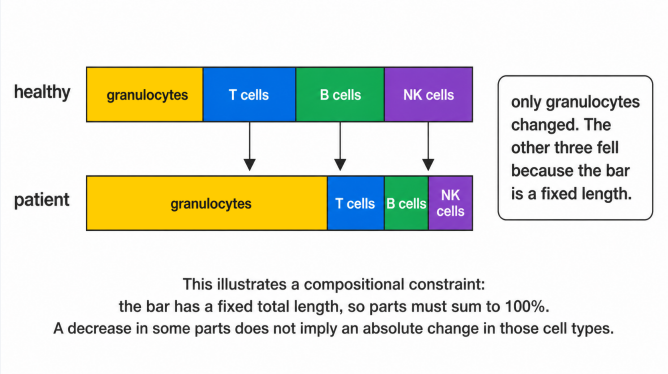


Figure 4.10: Two stacked bars of equal length: only granulocytes changed, yet every other segment shrank because the bar is a fixed length

3.2 Log-ratio

`clr_frequencies()` applies the centred log-ratio, expressing each population relative to the geometric mean of the sample composition rather than to a fixed total.

```
share <- function() {
  v <- c(60, 25, 15) * exp(rnorm(3, 0, .1))
  100 * v / sum(v)
}
freq <- data.frame(
  sample_id = rep(sprintf("S%02d", 1:6),
                  each = 3),
  population = rep(c("Granulocytes",
                     "CD4 T cells",
                     "B cells"), 6),
  pct_of_cd45_pos =
    as.vector(replicate(6, share())),
  is_control = FALSE, qc_status = "pass")
cl <- clr_frequencies(freq)
head(cl[, c("sample_id", "population",
            "pct_of_cd45_pos", "clr")], 3)
```

```
#>   sample_id   population pct_of_cd45_pos
#> 1       S01 Granulocytes        61.69888
#> 2       S01  CD4 T cells        24.02756
#> 3       S01      B cells        14.27356
#>          clr
#> 1  0.8023069
#> 2 -0.1407573
#> 3 -0.6615496
```

3.3 Concordance

Both parameterisations are tested and `compositional_concordance()` classifies each result.

`robust_to_composition` survives both and supports interpretation as an independent change.

`raw_only__possible_composition_artefact` is significant on percentages and not on log-ratios, which is the configuration produced by section 3.1.

`clr_only__was_masked_by_composition` is the converse, an independent change obscured by the constraint.

3.4 Limit

The log-ratio does not recover absolute abundance. Proportional expansion of every population leaves the composition invariant and is undetectable in frequency data by construction. Discriminating expansion from relative expansion requires cells per microlitre, so `abundance_measure()` uses absolute counts wherever the patient table supplies a leukocyte count.

4. Covariates

Age and sex are diagnosed rather than adjusted.

At the sample sizes cytometry cohorts reach, commonly under ten donors per group, a model estimating age and sex effects alongside the group term consumes the residual degrees of freedom required for the comparison of interest. Where the covariate is strongly associated with group, the parameters are not separable at any sample size, because the design contains no observations at the covariate levels required to estimate the effects independently.

`stats_confounding()` therefore reports the two conditions that jointly define a confounder: differential distribution across groups, and association with the outcome within groups (Rothman, Greenland & Lash 2008), a partial Spearman correlation with group held fixed. `stats_rank_ancova()` fits the adjusted model on request, labels every row `EXPLORATORY`, and records `NOT FITTED` with the reason where residual degrees of freedom are insufficient.

4a. Clinical variables, which are a different question

A confounder is screened to decide whether a group difference can be believed. A clinical variable, a severity score, a laboratory value, an outcome flag, is the question itself, and `stats_clinical_association()` treats it that way.

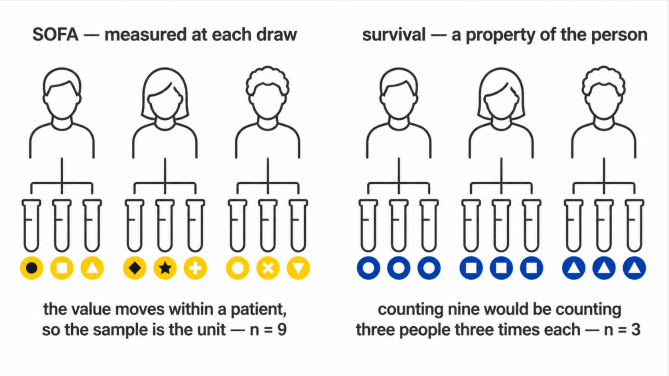


Figure 4.11: A score measured at each draw is per sample; a property of the person is per patient

The test follows the variable's type rather than being chosen:

Variable	Test	Effect reported	Signed
Numeric (SOFA, CRP, lactate, BMI)	Spearman's rho	rho, with a bootstrap interval	yes
Two levels (28-day survival)	Wilcoxon rank-sum	Cliff's delta, with a bootstrap interval	yes
Three or more levels (infection focus)	Kruskal-Wallis	epsilon-squared	no

Epsilon-squared is `H / (n - 1)`, the proportion of rank variance the grouping accounts for, bounded 0 to 1. It is unsigned, because a variable with three unordered levels has a magnitude and no direction, and that is carried explicitly in a `signed` column rather than left for the reader to infer from a sign that happens to be positive. Before it was added, a multi-level variable produced a p-value and nothing else: the figure could say a test had run and not whether it had found anything.

Rank methods throughout, for the reason they are used everywhere else in this package. Clinical scores are ordinal by construction, so a Pearson correlation asserts an interval scale the data does not have. Laboratory values are skewed and carry outliers that are real rather than erroneous, a creatinine of 335 is a patient, not a typing error, and a product-moment correlation on nine points with one such value is a statement about that one patient.

Multiplicity is corrected **within each variable**, across the populations tested against it. Each variable is one question asked of every population, and that is the family; pooling every variable into one correction would penalise a well-powered variable for the company it keeps. The choice is recorded here because it is a judgement rather than a default.

It also has a stated assumption, which is that the variables are separate questions. They frequently are not: in a cohort where the sickest patients are also the ones who died, SOFA and 28-day survival describe one gradient, and a population associated with both is one finding counted twice. `fig_clinical_correlogram()` exists to make that visible before the heatmap is read, because no correction can repair it and nothing else in the run would show it. Two-level variables appear there coded 0/1, for which Spearman is the point-biserial correlation of the ranks; variables with three or more unordered levels are excluded rather than coded 1/2/3, which would invent an ordering.

4b. Effect intervals on a small cohort

Every signed effect above is written with a 95% percentile bootstrap interval: 2000 resamples of the patient-value pairs for Spearman, stratified within each arm for Cliff's delta, from a fixed seed so a given table always yields the same interval and the RNG stream is restored afterwards so nothing downstream is perturbed.

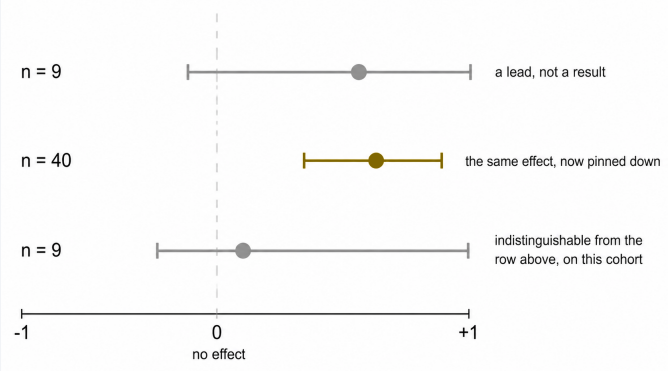


Figure 4.12: The same effect estimate with a wide interval at n = 9 and a narrow one at n = 40

The interval is the part of the answer that carries how little a small cohort constrains the effect. On ten patients a rho of 0.61 with an interval from -0.1 to 0.9 and a rho of 0.05 can be the same underlying quantity, and the point estimate alone cannot say so; reporting the effect and its interval beside the p-value, rather than the p-value alone, is the standard recommendation for exactly this situation.

Its own limits are stated on the figure that draws it. A percentile bootstrap at this sample size is approximate: below about nine observations the effective resample is smaller than the nominal one and coverage falls short of 95%. So a wide interval should be read as *this cohort does not constrain the effect* rather than as a calibrated range, and no interval is written at all below six samples, a number quoted at a width nobody should trust is worse than a blank.

The same reasoning is why `group_differences.png` draws the smallest p-value the design can reach. A Wilcoxon rank-sum test has `choose(n1 + n2, n1)` equally likely rank arrangements under the null, so the most extreme separation possible gives a two-sided p of `2 / choose(n1 + n2, n1)`. At 4 against 5 that is 0.016, and after correcting across a dozen populations nothing can reach 0.05 however cleanly the groups separate. Where that floor sits above the 0.05 line, an empty region of the figure is a fact about the design and not about the biology.

Two boundaries. This is **association, not survival analysis**: a 28-day flag is tested as the two-group comparison it is, and no time-to-event model is fitted, because the sample sheet carries no follow-up time and a proportional-hazards fit on a binary column with no time is a category error rather than an approximation. And every test runs on **one value per sample**, so the replicates are subjects; correlating a score against pooled events would treat one deeply acquired patient as tens of thousands of independent observations, which is the same failure `abundance_measure()` exists to prevent.

The `underpowered` column marks every test run on fewer than ten samples. At that size a rank test detects only very large effects, so a null result carries almost no information and should not be reported as evidence of no association.

5. Batch effects

Correction methods displace cells until batch labels mix. This is valid when batch is independent of the study design.

Clinical cohorts rarely satisfy that condition: recruitment and acquisition are typically sequential, so batch and group approach collinearity. Displacing cells until batches mix then also removes the biological contrast, and the algorithm cannot distinguish the two contributions.

`batch_mixing_report()` returns both required quantities before any correction: iLISI against a permutation null, establishing whether batch structure exists, and Cramér's *V*, establishing whether it is separable from group. A high *V* means no setting makes correction safe, which is the most informative result the diagnostic can return.

`--correct-batch` performs the correction and is refused above a configurable threshold.

6. Multiplicity

Benjamini-Hochberg correction is applied within each test family, with raw and adjusted p-values both reported.

Differential state adjusts within each measure rather than across both. Median intensity and percent positive are two summaries of the same events and are strongly correlated; pooling them would inflate the family with non-independent hypotheses and over-penalise every result.

7. Effect sizes

Every p-value is reported with an effect size: Cliff's delta across samples, or matched-pairs rank-biserial for paired designs. Clinical associations add a bootstrap interval on the effect as well; see §4b for why, and for why the interval is suppressed rather than narrowed below six samples.

Differences rather than ratios are reported for transformed intensities. Arcsinh and logicle scales admit negative values, on which a ratio is undefined in interpretation and changes sign without a corresponding change in biology.

8. Measurement resolution

An effect size states how large a difference is relative to the spread between donors. It does not state whether the instrument and the gating strategy could resolve a difference of that size at all.

`difference_over_gate_u` supplies the second quantity: the observed difference in medians divided by the typical within-sample uncertainty on the frequency, itself propagated from where the thresholds were placed. Below one, the groups differ by less than the distance the cut travels under resampling of the cells that determined it.

This is a screen, not a test. The uncertainty is partly shared across the run, since every sample passes through one panel, one transform and one placement rule, so it cancels in part when a difference is taken and the ratio is conservative. A ratio below one is a reason to open `threshold_uncertainty.csv` before interpreting the comparison, not grounds to discard it.

`difference_over_total_u` repeats the comparison against the gate term and the counting term together. The two ratios coincide for an abundant population and separate for a rare one, where the threshold can be well placed and the frequency still rest on too few events to support the difference being tested. Where they disagree, the smaller is the one to act on.

The order to read remains: adjusted p, then effect size, then these. A result that survives all of them is one where the groups differ, the difference is large relative to donor variation, and it is larger than both the distance the gate can move and the noise in the count.

8a. Detection limits

An effect size and a p-value can both be computed from a population of nine events. Neither reports that fact.

`detection` in `population_frequencies.csv` classifies each value against the conventional limits of detection and quantification, twenty and fifty events against the parent gate. A population below the limit of quantification in most samples should not be carried into a group comparison at all: the test will run, and its result is a statement about acquisition depth.

This is a screen on the measurement rather than a correction to the test. No p-value is adjusted by it, and the tests behave as they did before.

9. Summary

Question	Function	Unit
Did population abundance change?	`stats_group_comparison()`	sample
Does it survive the compositional constraint?	`clr_frequencies()`, `compositional_concordance()`	sample
Did marker expression change within a population?	`stats_marker_state()`	sample
Is the contrast confounded by age or sex?	`stats_confounding()`	sample
Is the contrast confounded by batch?	`batch_mixing_report()`	cell, descriptive
Is the contrast an artefact of the gate?	`stats_threshold_drift()`, `cluster_gate_agreement()`	sample and cell
Is the difference bigger than the gate's own uncertainty?	`run_gate_uncertainty()`, `annotate_gate_uncertainty()`	sample
Were enough cells counted for the frequency to mean anything?	`counting_uncertainty()`	sample
Which markers shift within a subcluster?	`stats_subcluster_shifts()`	cell, no p-value

10. Sources for the tests implemented here

Where each method came from, and what it is for.

Proportions. Population frequencies are bounded and their variance depends on their mean, so a test assuming constant variance is wrong in a way that gets worse as the population gets rarer. The logit is used here: Warton and Hui (2011, *Ecology* 92:3) argue the arcsine square root should not be used for proportions, and for flow cytometry results the logit gives symmetric distributions and homogeneous variances near 0 and 100% (Seiffert et al. 2024, *Front Mol Biosci* 11:1335174).

Unequal variance. Welch's t-test and Welch's ANOVA do not assume equal variances, so they stay valid when group sizes and spreads differ, which they almost always do in an immunophenotyping cohort. Games-Howell is the post-hoc test that matches: it uses Welch's standard error and pair-specific degrees of freedom rather than one pooled estimate (rstatix).

Rank post-hoc. Dunn's test compares mean ranks computed over the whole dataset, which is what makes it the correct follow-up to Kruskal-Wallis. A set of pairwise Wilcoxon tests re-ranks within each pair and answers a different question.

Clinical associations: the unit of analysis. A cohort sampled at several timepoints carries two kinds of clinical column. One is constant within a patient , 28-day survival, BMI, age, and asks a *between-subject* question; the other varies within a patient, a severity score at each draw, and asks a *within-subject* one. Treating repeated samples from one patient as independent observations is pseudoreplication: it inflates the apparent sample size, narrows the interval and makes significance more likely without a single extra patient having been recruited, and it is common enough in the life sciences to be one of the standard threats to reproducibility (*The problem of pseudoreplication in neuroscientific studies*, BMC Neuroscience; *Statistics Done Wrong*, on pseudoreplication). Collapsing to one value per subject is the first remedy named in that literature and is what `clin_variable_unit()` triggers for a patient-constant column.

For a column that varies within a patient, collapsing would delete the variation being asked about, so the per-sample test is kept and `n_patients` and `repeated_measures` are reported beside it. The dedicated method there is a repeated-measures correlation, which estimates the common within-individual slope without averaging (Bland & Altman 1995; Bakdash & Marusich, *Frontiers in Psychology* 2017, implemented in `rmcorr`). It is named rather than implemented, for the same reason as the mixed models below.

Effect sizes with intervals, not p-values alone. Reporting the estimated effect with a confidence interval, rather than a p-value on its own, is what the reporting guidelines for prognostic marker studies ask for, and the surveys behind those guidelines found 96% of papers reporting p-values against 42% reporting an interval (REMARK explanation and elaboration, *BMC Medicine* 2012). `clinical_association.csv` carries `ci_low` and `ci_high` on every signed effect for that reason, and `fig_clinical_forest()` draws them.

Rank effect sizes. Cliff's delta is the non-parametric two-group effect size for ordinal or non-normal data, and percentile bootstrap intervals are the standard way to bound it (*Cliff's Delta Calculator*, Universitas Psychologica). Epsilon-squared, `H / (n - 1)`, is the corresponding effect size for Kruskal-Wallis, following Tomczak & Tomczak's recommendation as implemented in `rstatix::kruskal_effsize()`; it is unsigned because a variable with three unordered levels has no direction. The bootstrap's own limit at these sample sizes, an effective resample smaller than the nominal one, and coverage falling short of 95% below about nine observations, is documented rather than ignored (*Statistical uncertainty analysis for small-sample, high log-variance data*), which is why no interval is written below six.

What is not implemented. Generalised linear mixed models are the right tool when donors contribute several samples, and CytoGLMM implements them for cytometry specifically (Seiler et al.). Beta regression models bounded support directly instead of transforming it (npj Systems Biology, 2025). Both need more donors per group than the cohorts this package is aimed at, and both add a dependency whose failure modes are harder to explain than the tests above. `--paired-column` with `--condition-column` covers the two-timepoint case; anything wider belongs in one of those packages.

Using cyRAVEN with cyCONDOR

cyRAVEN applies a declared specification and tests whether it held. cyCONDOR clusters events and assigns identity afterwards. The

two approaches carry non-overlapping failure modes, which makes combined application informative for a substantial class of study designs.

1. Rationale

Supervised quantification is bounded by its specification. A population absent from the declaration cannot be recovered, because no gate exists to fail.

Unsupervised partitioning carries the complementary limitation. It returns clusters for any input and holds no prior claim capable of being contradicted, so it cannot identify a mis-specified threshold. Its own parameters, cluster count, feature set and transform, are selected after inspecting the data, which relocates analyst discretion rather than removing it.

Applied together, each constrains the other. The principle is established: DAFi converts a manual gating strategy into constraints on a recursive clustering and recovers rare populations that neither approach resolves independently (Lee et al., 2018).

2. Selection

Objective	Tool	Basis
Abundance of a declared population between groups	cyRAVEN	Populations defined a priori, donor-level inference
Verification that gates transferred across samples	cyRAVEN	Threshold drift, phenotype concordance, gate-cluster concordance
Detection of an undeclared population	Both	cyRAVEN identifies the discordance, cyCONDOR characterises the population
Conversion of a cluster into executable gate geometry	cyRAVEN	`--explain-clusters`, or `--external-labels` for a labelling cyCONDOR produced
Whether that gate works on the next donor	cyRAVEN	Leave-one-donor-out refitting
A gate file the instrument or Cytobank can read	cyRAVEN	`--export-gates`, Gating-ML 2.0 in linear units
How precisely a reported frequency is determined	cyRAVEN	Threshold resampling propagated to the population
Differentiation trajectory	cyCONDOR	Diffusion maps and Slingshot
Label transfer to an independent cohort	cyCONDOR	Trained classifiers
Ingestion of existing FlowJo gates	cyCONDOR	`.wsp` workspace parsing
Data-determined cluster count	cyCONDOR	Phenograph
Unsupervised discovery without leaving cyRAVEN	cyRAVEN	`--explore`, over every eligible channel, with clusters named from per-sample thresholds. See Explore mode
Whether batch correction is admissible	cyRAVEN	Cramér's *V* refusal rule

3. Applications

3.1 Abundance change

A declared population differs in frequency between groups. Four outputs determine whether the difference is interpretable, read in order.

`threshold_drift_stats.csv`. A flagged marker means the per-sample thresholds separate by group, so the population is not identically defined across the comparison and the difference is partly definitional.

`compositional_concordance.csv`. Percentages are constrained to sum to 100. A result classified `raw_only__possible_composition_artefact` is consistent with expansion elsewhere in the composition rather than change in the population tested.

`cluster_gate_agreement_populations.csv`. A declared population distributed across several clusters is phenotypically heterogeneous, and the aggregate frequency averages subsets that may respond in opposite directions.

`confounding_diagnostics.csv`. Whether a covariate both differs between groups and associates with the outcome.

cyCONDOR contributes nothing here unless the third output indicates heterogeneity, in which case the constituent subsets require characterisation.

3.2 Undeclared population

Order matters in this analysis.

Run cyRAVEN with `--cluster --explain-clusters`. Gate-cluster concordance identifies a cluster dominated by the unassigned label, which establishes that the specification is incomplete. `cluster_gate_proposals.csv` returns two-marker gate geometry selecting those events, with performance measured on held-out cells, which is sufficient to reproduce the gate at the instrument.

Characterise in cyCONDOR. `plot_marker_ridgeplot`, `plot_marker_violinplot` and `plot_marker_dotplot` resolve the full intensity distribution rather than a median, and `plot_marker_group_HM` compares it across groups. Where the population may occupy a differentiation axis, `runDM` and `runPseudotime` establish whether a trajectory is supported.

Return to cyRAVEN. Add the population to the `populations:` block using the markers the proposal identified and re-run. It now carries a per-sample frequency and a donor-level test.

The final step is not optional. A population identified by clustering and tested by clustering on the same events was selected because it separated in those samples. Once declared, it is tested honestly in the next cohort.

3.3 A cyCONDOR cluster made executable

The reverse direction, and the one that closes the loop. cyCONDOR finds a population; the finding lives in a vector of cluster assignments and cannot be sorted on, drawn at the instrument, or applied to next year's files without clustering them again.

Export the assignment with the sample and the event index, then hand it over.

```
fs <- condor@clustering$FlowSOM
meta <- fs$metacluster
cl <- data.frame(
  sample  = condor@anno$sample_id,
  event   = condor@anno$event_index,
  cluster = paste0("k", meta))
write.csv(cl, "labels.csv",
          row.names = FALSE)
```

```
--external-labels labels.csv --export-gates
```

cyRAVEN fits a sequence of two-marker polygon gates to each label, then refits the whole sequence with one donor withheld and scores it on that donor, once per donor. `gate_transferability_summary.csv` carries the spread of those scores. Read `f1_min`: a gate that works on nine donors and fails on the tenth has the same median as one that works on all ten.

`--export-gates` writes the surviving strategy as Gating-ML 2.0 in the linear units the FCS file stores, which Cytobank and FlowRepository read directly and FlowJo reads through an ACS archive.

The join is on sample and event index, never row position. cyCONDOR subsamples at `prep_fcd(max_cell = )` and cyRAVEN takes its own cap for the embedding, so the two tables are not row-aligned and a positional join would relabel every cell without raising anything.

Hypergate established cluster-to-gate conversion, fitting one hyperrectangle per cluster (Becht et al., 2018). What is added here is the polygon hierarchy, which is the shape a sorter is actually driven in, the per-donor score distribution, and the gate file.

3.4 Multi-site or longitudinal designs

Analyst variance is the limiting factor. Thirty-eight operators gating the same files carried a median expanded uncertainty of 3.6% under a gauge repeatability and reproducibility design (Grant et al., 2021), and across 320 routine clinical samples the residual dispersion was structured primarily by operator identity rather than by instrument configuration (Mead et al., 2026). Reagent standardisation alone is insufficient: EuroFlow reached between-centre coefficients of variation below 7% only after standardising the analysis (Kalina et al., 2012).

Use cyRAVEN as the primary analysis. A declared specification with per-sample thresholds removes the analyst from the step at which that variance enters. Version-control the config alongside the results and retain `run_manifest.txt`.

Write a baseline from the first accepted run and pass `--baseline` on every later one. The within-run threshold check compares each sample against its peers, which finds a bad tube and cannot find a site or a season that moved as a whole, since the peer median moves with it. On a scheduled pipeline, `--fail-on-drift` makes the verdict an exit code.

Use cyCONDOR as a periodic audit. Cluster the accumulated data without reference to the specification and confirm that reported populations still resolve as coherent clusters. Populations that begin to split or merge indicate acquisition drift requiring the specification to be revisited.

4. Execution

Both tools read the same FCS files. cyCONDOR requires an annotation table whose first column names the files, which is the information a cyRAVEN sample map already carries, so one file serves both.

```
run_cyraven(list(
  dir = "data/fcs/",
  outdir = "results_cyraven/",
  sample_map = "data/sample_map.csv",
  config = "data/panel.yaml",
  group_column = "cohort",
  reference_group = "Healthy controls",
  unsupervised = TRUE,
  explain_clusters = TRUE
))

anno <- read.csv("data/sample_map.csv")
at <- "results_condor/anno_table.csv"
write.csv(anno, at, row.names = FALSE)

condor <- cyCONDOR::prep_fcd(
  data_path = "data/fcs/",
  max_cell = 30000, useCSV = FALSE,
  transformation = "auto_logi",
  remove_param = c("Time", "InFile"),
  anno_table = at,
  filename_col = "file", seed = 42)
condor <- cyCONDOR::runPCA(
  condor, data_slot = "orig", seed = 42)
condor <- cyCONDOR::runUMAP(
  condor, input_type = "pca",
  data_slot = "orig", seed = 42)
condor <- cyCONDOR::runFlowSOM(
  condor, input_type = "pca",
  data_slot = "orig",
  nClusters = 8, seed = 42)
```

The `benchmark/` directory of the cyRAVEN repository executes both on a public dataset and writes the comparison.

5. Embeddings

The two UMAPs are not comparable, for four reasons measured on the benchmark dataset and given in order of magnitude.

Event set. cyCONDOR embeds all 235,259 events. cyRAVEN embeds 40,000 events surviving scatter, singlet, viability and CD45 gating. A fraction of the cyCONDOR manifold is debris.

Feature set. cyCONDOR includes FSC-A, FSC-H, FSC-W and SSC-A among eleven features. Scatter spans decades and dominates the principal components on which the embedding is computed, so much of the resulting structure reflects size and granularity. cyRAVEN uses scatter for gating and excludes it from the feature set, embedding four markers.

Input space. cyCONDOR embeds principal component scores; cyRAVEN embeds the scaled marker matrix, which alters the neighbour graph before construction.

Transform. Auto-logicle per file against arcsinh pooled per panel places identical raw intensities at different coordinates.

Compare the tables, not the figures.

6. Resolution

Cluster count is an analytical choice with consequences. On the benchmark data FlowSOM requested at 8 returns 8 clusters; Phenograph, which determines its own, returns 24 on the same events. Both are correct implementations. Neither corresponds to the number of populations present.

A result that appears at one resolution and disappears at another is a property of the setting. Confirm persistence across resolutions before interpretation.

FlowSOM is a defensible default: it ranked at or near the top across every dataset in the reference comparison of clustering methods, at runtimes permitting routine use as a cross-check (Weber and Robinson, 2016).

7. References

- Lee et al., 2018. DAFi: gating strategy as constraint on recursive clustering. *Cytometry A*.
- Briefings in Bioinformatics, 2025. Systematic comparison of manual gating, unsupervised clustering and auto-gating across 23 tools.
- Weber and Robinson, 2016. Clustering method comparison. *Cytometry A* 89:1084.
- Kalina et al., 2012. EuroFlow standardisation. *Leukemia* 26:1986.
- Grant et al., 2021. Operator subjectivity as measurement uncertainty: 38 participants, median expanded uncertainty 3.6%. *PDA J Pharm Sci Technol* 75:33.
- Mead et al., 2026. Interpretive contribution to analytical variability, 320 clinical samples and six technologists. *Cytometry B Clin Cytom*.
- Becht et al., 2018. Hypergate: reverse-engineering a gating strategy from a cluster. *Bioinformatics* 35:301.
- Spidlen et al., 2015. ISAC Gating-ML 2.0. *Cytometry A* 87:683.
- Grant et al., 2021. Uncertainty analysis applied to manual gating; a diagrammatical protocol cut between-participant variation by 57%. *Methods Protoc* 4:24.

Driving it from Claude Code

The repository ships a skill for Claude Code at `.claude/skills/cyraven/`. Installed, it gives the assistant the parts of this documentation that determine whether an answer is correct: Docker as the execution path, the gate hierarchy and specification syntax, and the order in which the outputs must be read.

It is documentation, not a wrapper. Nothing about the pipeline changes, and every command works without it.

The skill executes through Docker by default and does not fall back to a local R installation unless the user states that Docker is unavailable, for the reproducibility reasons given in the README.

1. What it covers

File	Content
`SKILL.md`	What cyRAVEN does, the Docker execution commands, required inputs, the flags that matter, and the analytical constraints in section 4
`references/docker.md`	Build and run in detail, path and mount semantics, Windows and PowerShell, resource tuning, iterating with `CYRAVEN_SOURCE`, container-specific failures
`references/gating.md`	The four-gate hierarchy, per-sample thresholding and the `source` column, writing the population YAML, three-level markers, arcsinh against logicle, compensation
`references/interpretation.md`	Every output file and the order to read them in, with the columns that decide whether a result stands
`references/troubleshooting.md`	Failure modes with their actual causes

Only `SKILL.md` is read when the skill activates. The reference files are opened when the task needs them, so the description above is also the routing table.

2. Installing

For every project

Copy the directory into the personal skills folder:

```
cp -r .claude/skills/cyraven \
  ~/.claude/skills/
```

```
Copy-Item -Recurse `
  .claude\skills\cyraven `
  $HOME\.claude\skills\
```

Confirm with `/skills`, or ask Claude to invoke `/cyraven`.

For this repository only

Nothing to do. A checkout already contains `.claude/skills/cyraven/`, and Claude Code discovers it when the session is rooted in the repository.

Without a copy

Point Claude at the raw SKILL.md by pasting its link with:

```
Read this SKILL.md and follow it:
<paste the link here>
```

3. Using it

The skill activates on its own when a request matches its description. Force it with `/cyraven`.

Requests it is built for:

```
Run cyRAVEN on the FCS files in ./data.
The panel is CD3, CD4, CD8, CD14, CD16,
CD19, CD56, HLA-DR.

Set up the demonstration dataset and
show me what the output looks like.

Write a population spec for this panel
and explain each definition.

My frequency table came out empty. Why?

Read results/ and tell me which findings
survive.

Is the difference in CD8 T cells between
the two cohorts real?
```

The first builds and invokes the container without being asked to, because that is the skill's default execution path.

The last is the case worth having the skill for. It has a documented answer that is not a p-value: adjusted p, then Cliff's delta, then `difference_over_gate_u`, then `difference_over_total_u`, then a check that the marker is not flagged in `threshold_drift_stats.csv`. The order is set out in Statistics §8, and the skill states it so the assistant does not stop at the first significant result.

4. Analytical constraints

The skill carries five positions the tool takes. They are recorded because overriding any of them produces output that appears correct and is not. Each is documented at length elsewhere and summarised there only to the extent needed to apply it.

Constraint	Treated in full
Thresholds are not transferred between samples	Gating §2.1
`count` is an event count, not a cell number	Output §4
Placement precision does not imply counting sufficiency	Gating §2.5
A specification is a hypothesis, not a validated result	Diagnostics
Gates are inspected before any derived number is quoted	Diagnostics §1

5. Contributing through it

`SKILL.md` also carries the conventions this package is maintained under, so a change proposed with the skill loaded arrives in the right shape:

- Additions are additive. Every file the previous version wrote is still written, with the same name, the same columns in the same order, and the same values.
- Additivity is verified by running both versions on one cohort and comparing byte for byte, not by asserting it.
- Anything that only adds an output is on by default. Anything that changes an existing number is opt-in.
- Any new entry point that consumes random draws saves and restores `.Random.seed`. `run_cyraven()` seeds once and the UMAP cell selection draws from that stream, so a step that spends draws silently redraws every embedding in the run.

6. Keeping it accurate

The skill quotes flag defaults, output filenames and column names. Those were checked against the source when it was written and will drift if the package changes without it.

When adding a flag or an output, update `SKILL.md` and the reference file that mentions the area. The filenames are verifiable mechanically:

```
grep -rohE \
  '"[a-z0-9_]+\.(csv|png|txt|rds)"' \
  R/pipeline.R | sort -u
```

Cross-check that list against `references/interpretation.md`.

5 Gallery

Every figure a run produces, shown as unmodified output, with what each one measures and what this particular run shows.

The data is the public GvHD cohort that ships with flowCore, so every figure here can be reproduced from a clean checkout using the commands in section 0. The two figures the package can draw but this cohort cannot produce are at the end, with the reason.

0. The data, and how to reproduce every figure below

0.1 What the data is

The graft-versus-host disease dataset distributed with flowCore (Brinkman et al., 2007, *Biol Blood Marrow Transplant* 13:691; Artistic-2.0): peripheral blood from five allogeneic transplant recipients, drawn at up to seven successive visits and stained with a four-colour myeloid panel.

Samples	35
Patients	5
Acquisition batches	7 successive visits
Groups	GvHD grade 1 (7 samples) against grade 3 (28)
Panel	CD15, CD45, CD14, CD33, with forward and side scatter and a Time channel
Events per file	2,205 to 66,105, median 12,666

Nothing is downloaded. The data ship inside a package cyRAVEN already depends on, so this example is reproducible offline and cannot break when a repository moves or a certificate expires. `inst/scripts/demo_data.R` writes them out as FCS together with the two input files, making two preparation changes and no others: `$PnS` is reduced from `CD15 FITC` to `CD15`, because cyRAVEN matches marker symbols exactly; and the single area channel `FL2-A` is dropped, leaving a consistently height-only file. Event values are written as flowCore ships them.

0.2 The exact sequence

Five commands, one block each. Copy and run them one at a time; together they reproduce every figure on this page.

1. Get the image. This is the image these figures were produced with, so pulling is the shorter path to the same numbers.

```
docker pull bhagesh/cyraven:1.0.0
```

```
docker tag bhagesh/cyraven:1.0.0 \
  cyraven:1.0.0
```

To build it from source instead, see Commands and every option; the build context is the repository root.

2. Make the folders.

```
mkdir -p demo results
```

3. Write the cohort, its sample sheet and its config.

```
docker run --rm -v "$PWD:/w" \
  cyraven:1.0.0 --demo /w/demo
```

4. Validate the inputs, in seconds, analysing nothing.

```
docker run --rm \
  -v "$PWD/demo:/data:ro" \
  -v "$PWD/results:/results" \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/panel.yaml \
  --group-column cohort \
  --reference-group "GvHD grade 1" \
  --batch-column visit \
  --outdir /results \
  --check
```

5. Run it.

```
docker run --rm \
  -v "$PWD/demo:/data:ro" \
  -v "$PWD/results:/results" \
  cyraven:1.0.0 \
  --dir /data/fcs \
  --samples /data/samples.csv \
  --config /data/panel.yaml \
  --group-column cohort \
  --reference-group "GvHD grade 1" \
  --batch-column visit \
  --cluster \
  --outdir /results
```

On Windows PowerShell substitute `${PWD}` for `$PWD`; on Git Bash prefix the `docker run` lines with `MSYS_NO_PATHCONV=1`, or the shell rewrites `/data` into a Windows path.

Step 4 should report 35 files, four resolved markers, that every marker the specification names is present, that the sheet covers every file, and the group sizes. Step 5 writes 29 figures and 33 tables into `results/`, plus `report.html`, which carries all of them embedded in one file, numbered, with a search bar at the top.

0.3 What each flag is doing

Flag	Effect on this page
`--group-column cohort`	Enables section 6. Without it there is no between-group comparison and `group_comparison.png` is not written
`--reference-group "GvHD grade 1"`	The group the other is tested against, drawn unfilled and leftmost
`--batch-column visit`	Enables section 7. Without it `batch_diagnostic.png` and the drift table are absent
`--cluster`	Enables `unsupervised_clusters.png` and the gate-cluster concordance tables in section 4

The defaults supply everything else: gate uncertainty, detection limits, acquisition QC and the spreading report are all on unless switched off, because an analysis that has to be requested is one that does not get run.

What the colours mean, and why they are those colours

Study, cohort and timepoint groups take **green, red, purple, amber, blue, pink** in that order, with green always the reference group. The assignment is fixed, so a colour means the same group in every panel of every figure in a run.

The order is chosen for separation rather than aesthetics. Groups are usually few, and with few categories the eye needs hues that are far apart: red, orange and yellow are neighbours on the colour wheel, are

the hardest triple to tell apart at bar-chart size, and collapse almost completely under red-green colour vision deficiency. Each colour here sits roughly a sixth of the wheel from the last.

Two groups are never given the same colour. Past six the run generates further hues rather than starting the list again, because a repeated colour reads as one group split across two bars.

The same reasoning applies to small non-study category sets, where the picks are spread across the population palette instead of taken in order.

Two more things about how the figures are drawn

Backgrounds are opaque white. Figures used to be written with a transparent ground, 11 of 20 in one run. That is invisible inside `report.html`, which is white, and breaks everywhere else: expand a figure in a dark viewer, or drop it on a dark slide, and the black axis text sits on whatever is behind it.

Legends sit at the top left on the large multi-panel figures - `functional_markers.png`, `population_frequencies.png`, `population_ratios.png`, `group_comparison.png`, `marker_state.png` and `umap_multigraph_overlay.png`. A collected guide is otherwise centred vertically on the right, which on a figure several thousand pixels tall is a long way from the panels it explains. The single-panel UMAP figures keep theirs on the right, where it already sits beside what it describes.

0.4 Why your figures should match

The container pins R 4.4.3, a dated CRAN snapshot and Bioconductor 3.20, and the run is seeded once at 42 with every stage restoring the RNG stream. Those two things together are what make the figures on this page reproducible rather than merely similar. Run outside the container and the embedding will differ, because `uwot` is stochastic across its own versions and across the BLAS beneath it.

Every option, and the situation each is for, is in Running cyRAVEN. The two input files are described in Inputs.

0.5 Four properties of this cohort that govern how the figures read

The contrast is disease severity, not disease against health. Both groups are transplant recipients; the comparison is GvHD grade 1 against grade 3. There is no healthy arm.

The groups are unbalanced and confounded with donor. Seven samples are grade 1 and 28 are grade 3, and grade is a property of the patient rather than of the visit, so between-group differences are partly between-donor differences. Section 7 is where that shows.

The acquisition recorded pulse height and no area. cyRAVEN falls back to height and says so. Height understates a bright wide event, so thresholds here are not interchangeable with those from an area acquisition of the same panel. The singlet gate is skipped, since it needs FSC-H and FSC-A to be distinct.

Eight of 35 samples fail staining QC and are excluded from every test. Twenty- seven remain.

Which file each figure comes from, and which flag produces it, is in the Every output file. The order to read them in is in the Diagnostics and is reproduced by `report.html`.

1. Before any number

recon_diagnostics.png

The scatter gate per sample, drawn on the distribution it was derived from.

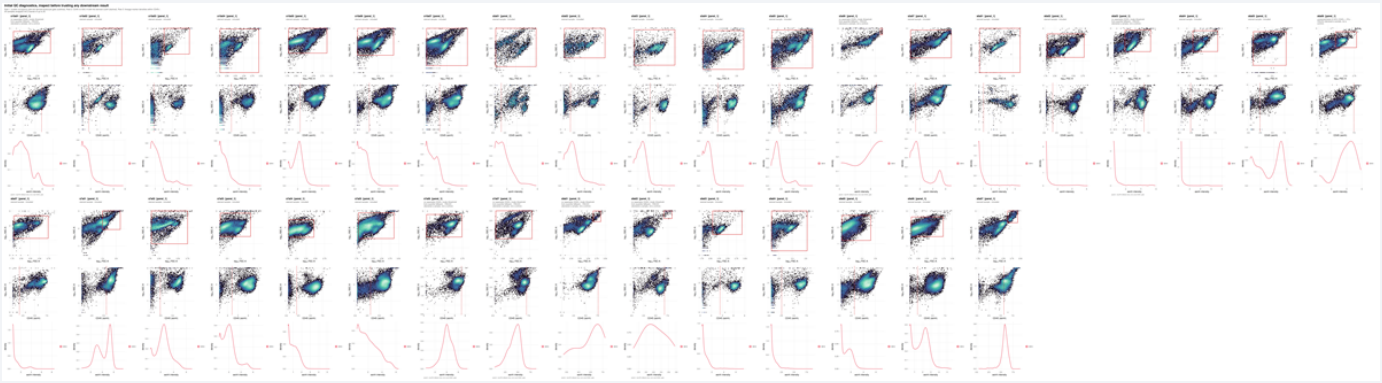


Figure 5.1: Forward-scatter density per sample with the derived scatter gate boundary marked

What to look for. The scatter boundary should sit in the trough between the debris mode and the intact-cell mode.

This run. The log carries `no FSC-A density valley found; using the 60th percentile`. On this instrument the forward-scatter distribution has no clean debris shoulder, so the boundary is a quantile rather than a minimum. That is a fallback, and it is the reason the figure has to be read before anything derived from it: every population below sits inside a parent gate whose lower edge was placed by convention rather than by the data.

gating_qc.png

Every marker threshold superimposed on the density it was derived from.

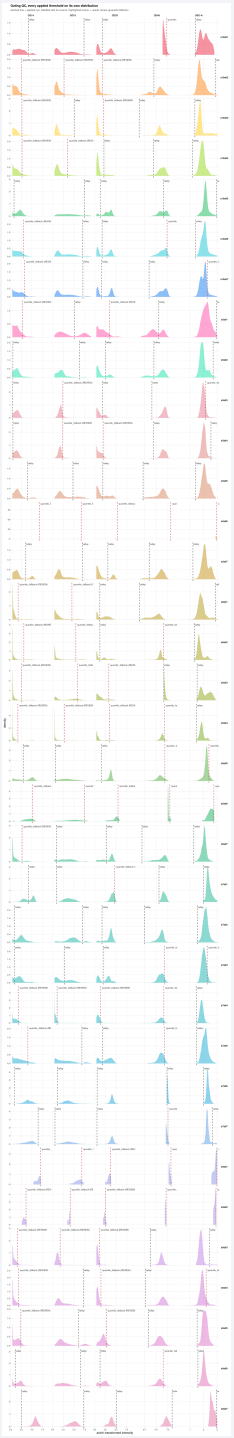


Figure 5.2: Per-sample marker densities with each derived threshold drawn on the distribution it came from

What to look for. A cut in the trough between two modes is well placed. A cut on the flank of a single mode is a quantile fallback and carries no evidence of separation.

This run. Of 175 sample-and-marker thresholds, **103 resolved a density minimum and 72 fell back to a quantile**. The fallbacks are not spread evenly: `spreading_receivers.csv` in section 5 shows which markers they concentrate in and why.

2. Placement precision and counting sufficiency

frequency_uncertainty.png

Each population's per-sample frequency with the standard uncertainty propagated from the thresholds behind it.

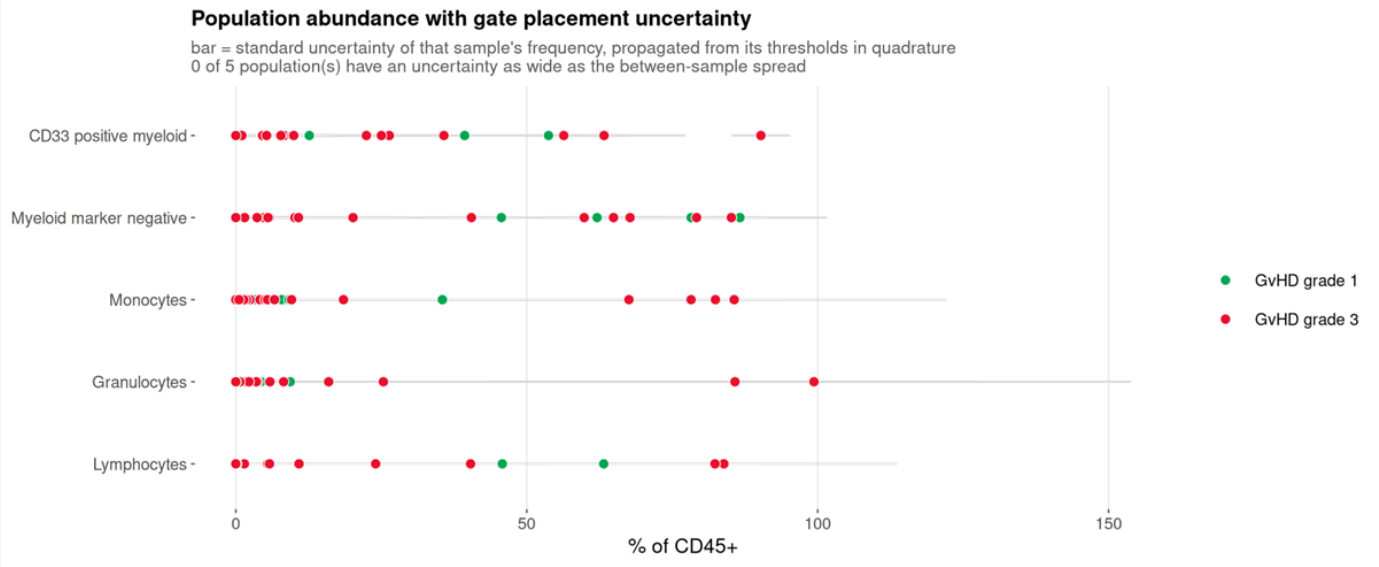


Figure 5.3: Population frequency per sample with gate placement uncertainty as an interval

What to look for. Whether the bar within a sample is narrower than the scatter between samples. Where it is not, the variation on display is the cut moving.

This run. None of the five populations has a gate uncertainty as wide as its between-sample spread. The donors differ from each other by more than the cuts move, which is the configuration in which a between-group comparison is worth attempting at all.

uncertainty_budget.png

Which threshold each population's uncertainty comes from.

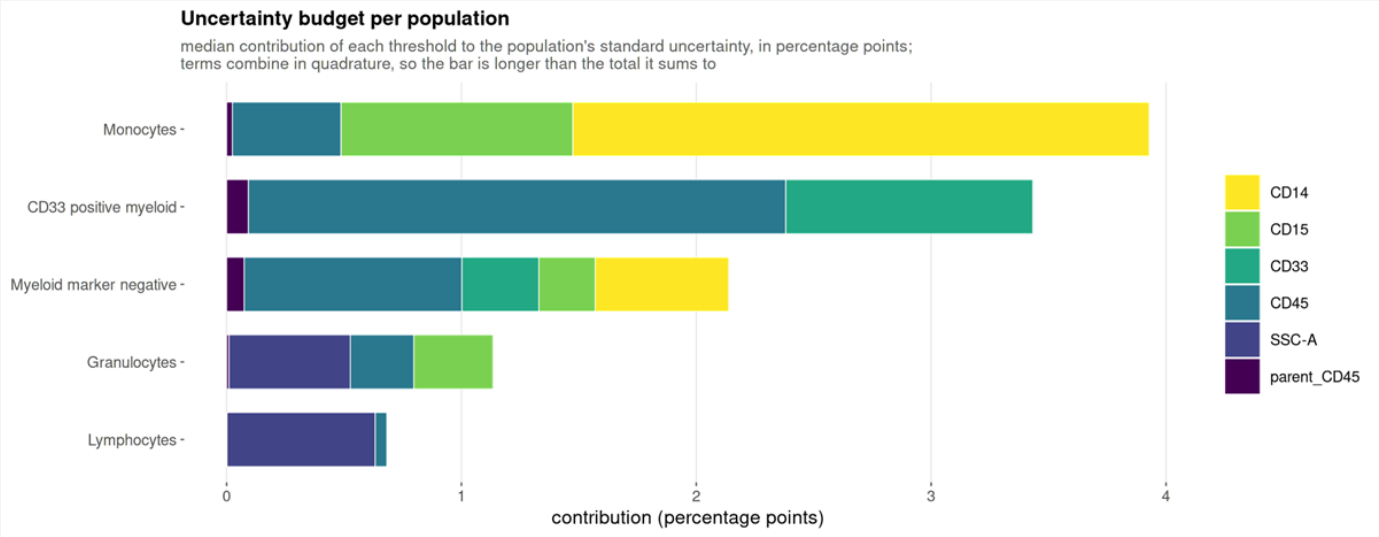


Figure 5.4: Stacked contribution of each marker threshold to every population's uncertainty

What to look for. Which gate to fix. A population dominated by one marker has one threshold worth inspecting.

detection_limits.png

How many samples each population clears the limits of detection and quantification in.

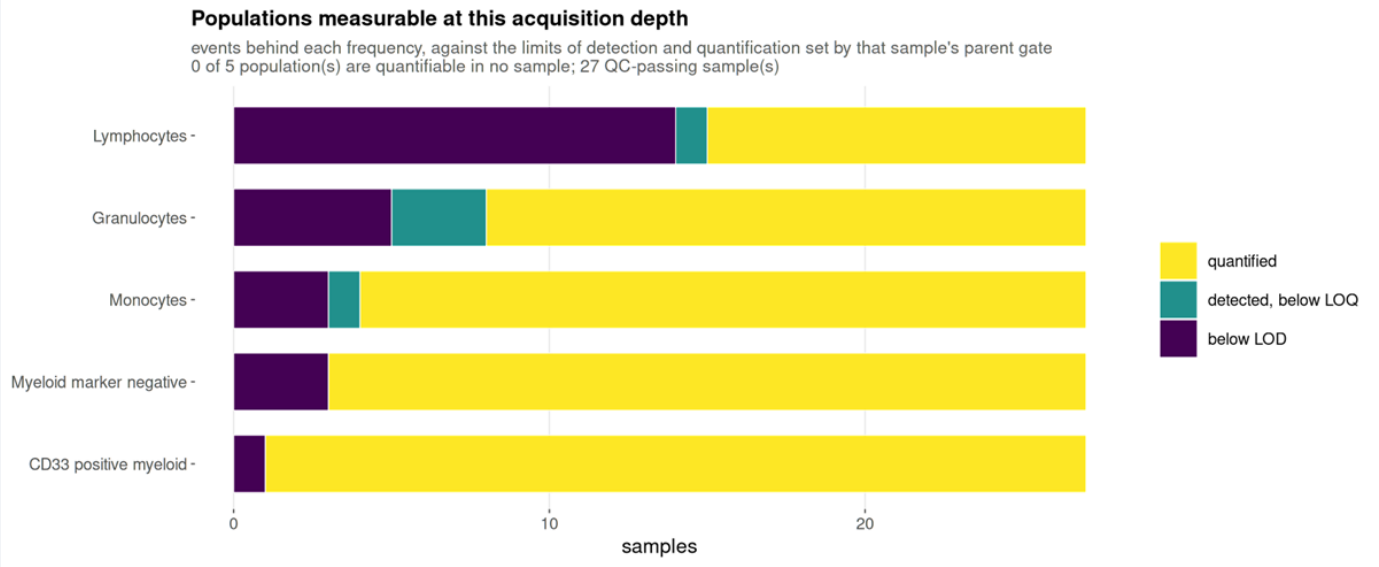


Figure 5.5: Populations counted by whether their event count clears the limits of detection and quantification

What to look for. A population mostly below the limit of quantification cannot be recovered by re-gating: the numerator is small for want of acquired cells.

This run. Of 135 population-and-sample values, **104 are quantified, 5 are detected but below the limit of quantification, and 26 are below the limit of detection**. The below-limit values are concentrated in the smaller populations and in the samples with the fewest events, which range from 2,205 upward. Those 26 values should not carry a group comparison on their own.

3. Are the populations what they were declared to be

population_marker_heatmap.png

Measured marker intensity against the declared populations, z-scored.

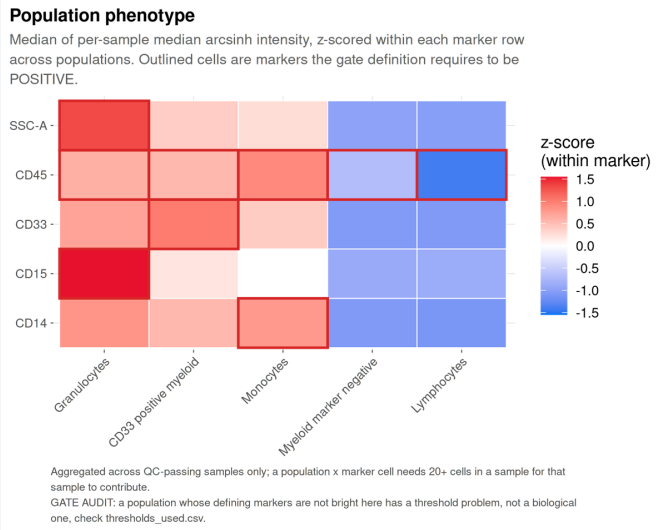


Figure 5.6: Heatmap of z-scored marker intensity across the declared populations

What to look for. Identity is declared here before the data are examined, so this figure serves the inverse function of its equivalent in clustering-first analysis: a population that does not show the markers its definition requires falsifies the gate that produced it.

population_mfi_heatmap.png

Median fluorescence intensity per population and marker, in the instrument's own units: each sample's median first, then the median over samples. The number in each tile is the intensity; the colour is the same value on the analysis scale, rescaled per marker, so it reads as "bright for this marker". The bar beside each row is the population's median share of CD45+ cells.



Figure 5.7: Heatmap of median fluorescence intensity per population and marker, with the value printed in each tile and a bar of each population's share

What to look for. Read a row for a population's phenotype and a column for which population is brightest on a marker. Compare the printed values between runs only at matched instrument settings, because they are in the instrument's units.

This run. Lymphocytes sit at 696 on CD45 and at 3 or less on CD14, CD33 and CD15, which is what their definition requires. Granulocytes have the highest side scatter (315, against 81 for lymphocytes) and read 1.0k on CD15. The CD45 column reads 1.0k for every myeloid population because these files record intensity on a 0 to 1,023 scale and more than half of those events sit at the top of it: the populations are bright, but how bright is beyond what the files can say.

population_mfi_dotplot.png

The same medians as colour, with the percent of each population above its own sample's threshold as the size of the dot.

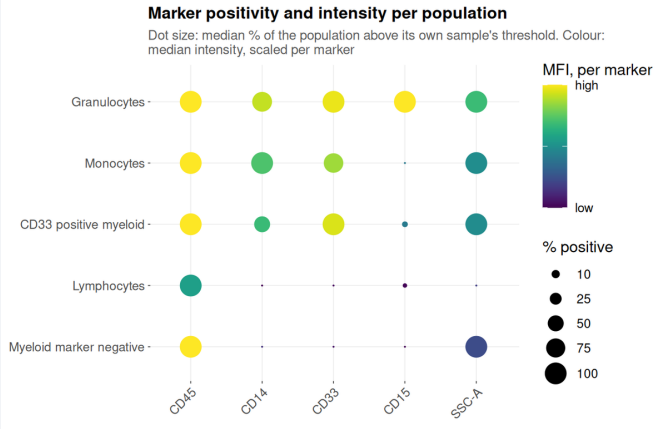


Figure 5.8: Dot plot of percent positive as dot size and scaled median intensity as colour, per population and marker

What to look for. Size and colour disagreeing is the informative case. A large pale dot is a population that is uniformly dim on the marker; a small dark one is a bright minority inside it.

This run. Monocytes are CD14-positive at a median of 100% and CD15-positive at 0%, while granulocytes are the reverse on CD15. The CD33-positive myeloid population is CD14-positive at about 50%: half of it is monocytes, which the specification allows because it defines that population by CD45 and CD33 alone.

population_frequencies.png

The pooled composition across samples.

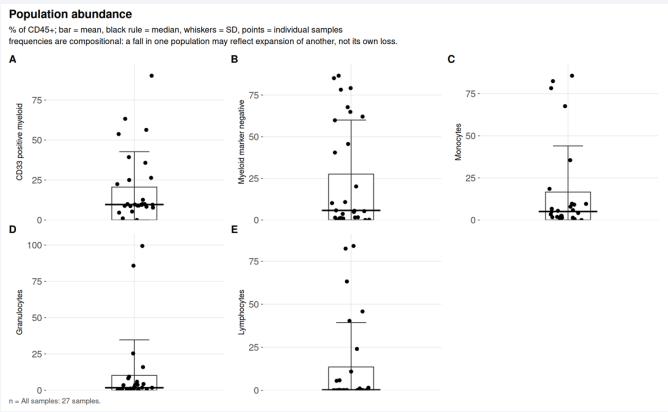


Figure 5.9: Pooled population composition across all samples

cohort_composition_heatmap.png

Each group's share of every population, after normalising the groups to a common notional cell count.

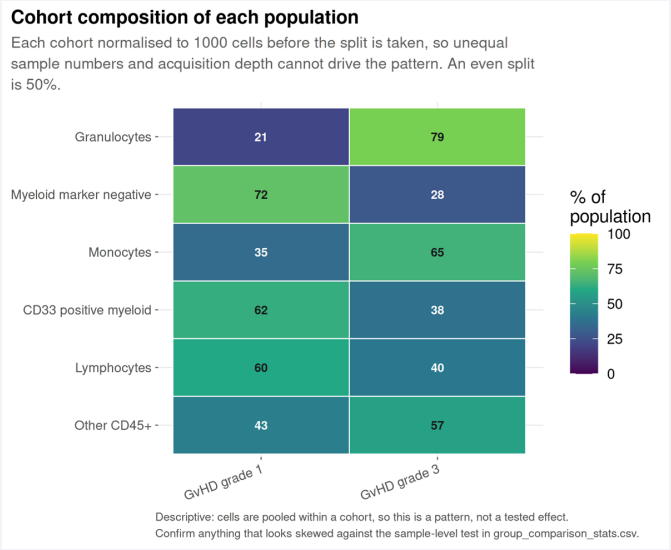


Figure 5.10: Heatmap of each group's share of every population after normalising to a common cell count

What to look for. An even split is 50%. Normalising first is what stops the 7-against-28 sample imbalance from generating the pattern.

4. The shared embedding

umap_composition.png

The shared embedding coloured by population, with a legend that says what each colour is: the population's name, its defining markers from the specification, its share of the embedded cells, and its share within each study group. To the right, on the same rows, the percent of each population above its own sample's threshold for every marker the panel thresholds. A sheet with a `timepoint` column also gets one copy per visit, `umap_composition_<timepoint>.png`, on the same axes and in the same row order.

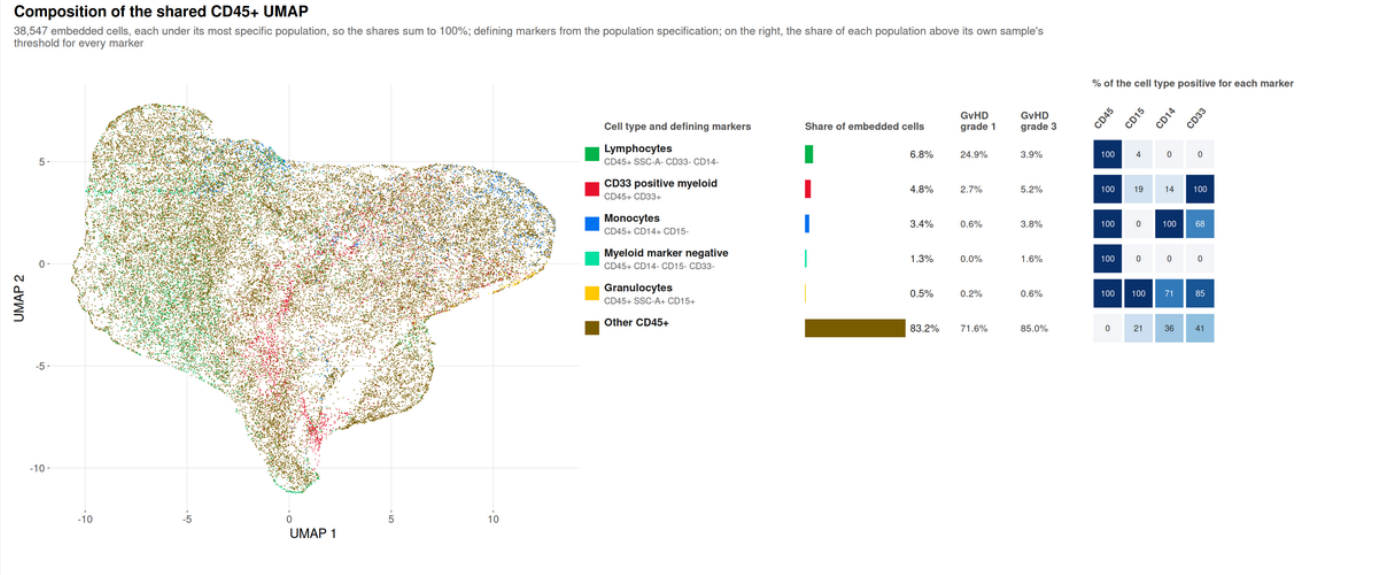


Figure 5.11: Shared UMAP coloured by population beside a legend of each population's defining markers, share of embedded cells and share per GvHD grade

What to look for. Each embedded cell carries one label, its most specific population, so the shares sum to 100. They are therefore not the frequencies in `population_frequencies.csv`, where the gates overlap.

This run. Most of the embedding, 83%, is "Other CD45+": the specification declares five populations, and on a four-colour myeloid panel most CD45+ events match none of them. Lymphocytes are 24.9% of the grade 1 cells and 3.9% of the grade 3 cells, but `population_composition.csv` gives their median share per sample as 0: only 5 of the 22 embedded samples hold any. Read that column before quoting a pooled share, and the per-donor test in `group_comparison.png` before reading the difference as one between grades. In the marker grid, monocytes are 100% CD14-positive and 68% CD33-positive, while the CD33-positive myeloid row is only 14% CD14-positive: each cell carries its most specific label, so the CD14-positive CD33-positive cells are counted as monocytes, and the row holds what is left. The dot plot in section 3, which scores the overlapping gates, puts the whole CD33-positive population at about 50%.

umap_overview.png

One UMAP per marker panel, computed across all samples.

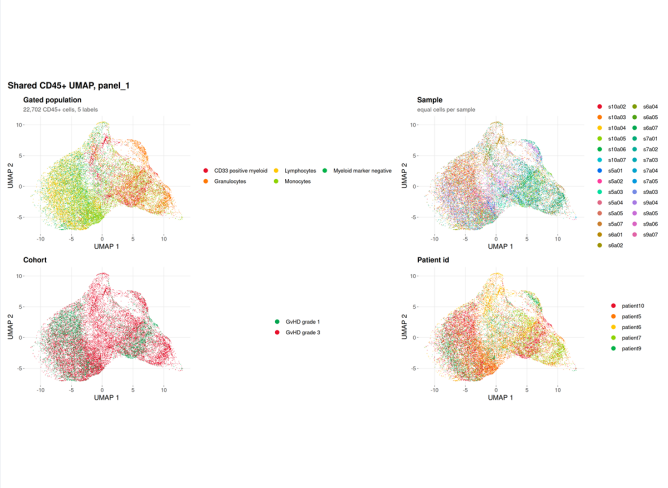


Figure 5.12: Shared UMAP embedding coloured by population, sample and group

This run. The panel resolves four markers, and the log records `fewer than 2 preferred lineage markers present; falling back to all 4 eligible markers`. The embedding is therefore driven by CD15, CD45, CD14 and CD33 rather than by a curated lineage set, which is the correct behaviour for a four-colour panel and a limitation of the panel rather than of the run.

umap_overview_by_group.png

The same embedding split by GvHD grade.

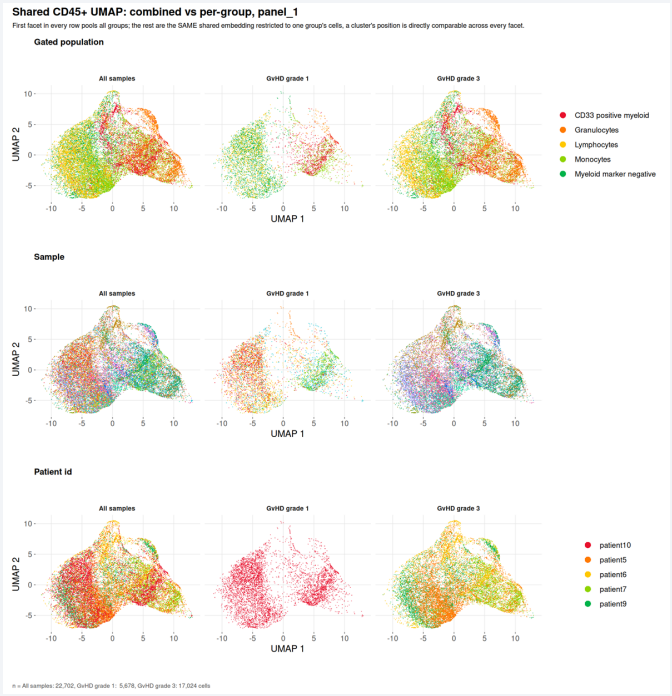


Figure 5.13: Shared UMAP embedding split into one panel per GvHD grade

umap_markers.png

The embedding coloured by each marker's intensity in turn.

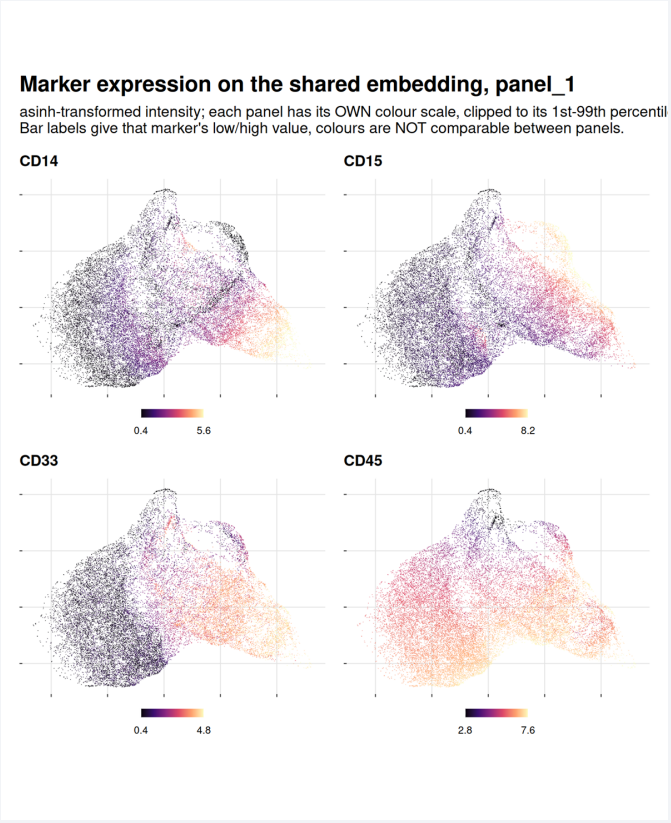


Figure 5.14: Shared UMAP embedding coloured by each marker's intensity in turn

marker_umaps_by_group/

A folder rather than a single file, written by every run: the same markers as above, one per file at full size instead of shrunk into a grid cell.

Two kinds of file per marker.

File	What it shows
`umap_CD3.png`	Every sample pooled
`umap_CD3_by_<category>.png`	The same marker split by a category from the sample sheet

A split file is written for every category present, not only the one named by `--group-column`. That flag decides what is tested; it says nothing about what is worth looking at, and a cohort usually carries several categories at once. A run grouped by timepoint still gets its infection-focus and sex splits.

The pooled file is kept alongside them because it answers a different question. Splitting asks whether a marker sits differently between categories; pooling asks where the marker is at all, and no split figure can show that, because each facet holds only a subset of the cells.

Within one file the colour scale is shared across the facets and clipped to the 1st and 99th percentile of the cells drawn, so a colour difference between two panels is a real intensity difference. It is not shared between files: markers differ by orders of magnitude in brightness, and one scale would flatten every dim one.

Not faceted: numeric columns, which would give one panel per distinct value; identifiers such as `sample_id` and `patient_id`, which are not categories; and single-level columns, since one facet is not a comparison. Past four categories the run draws four and names the rest in the log. `--no-marker-group-umaps` skips the folder.

Both kinds of file carry the same panel border and grey title bar. The pooled one is a single panel faceted by a constant, "All samples", so it comes out of the same code path as the split ones: a section where one figure is framed and the next is not reads as two kinds of figure rather than one kind drawn two ways.

umap_density.png and umap_density_by_group.png

Where cells are concentrated, overall and by group.

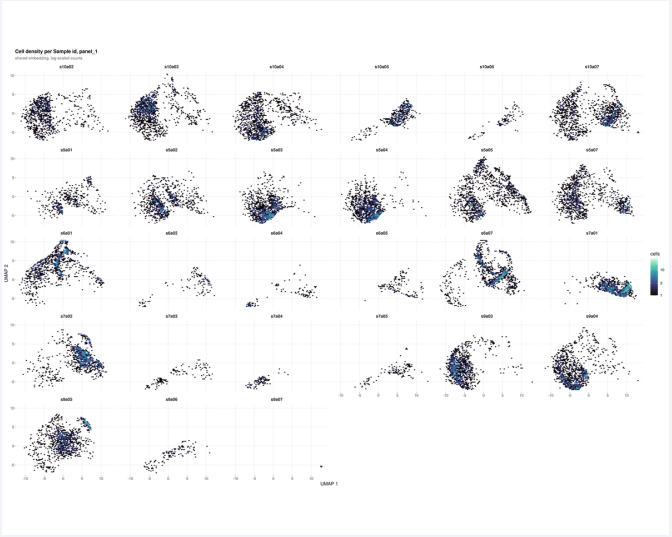


Figure 5.15: Cell density across the shared UMAP embedding

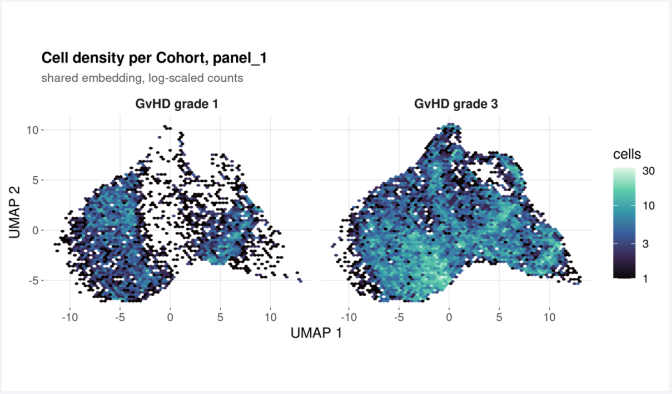


Figure 5.16: Cell density across the shared UMAP embedding, split by GvHD grade

umap_multigraph_overlay.png

Subclusters within each declared population, drawn on the shared embedding.

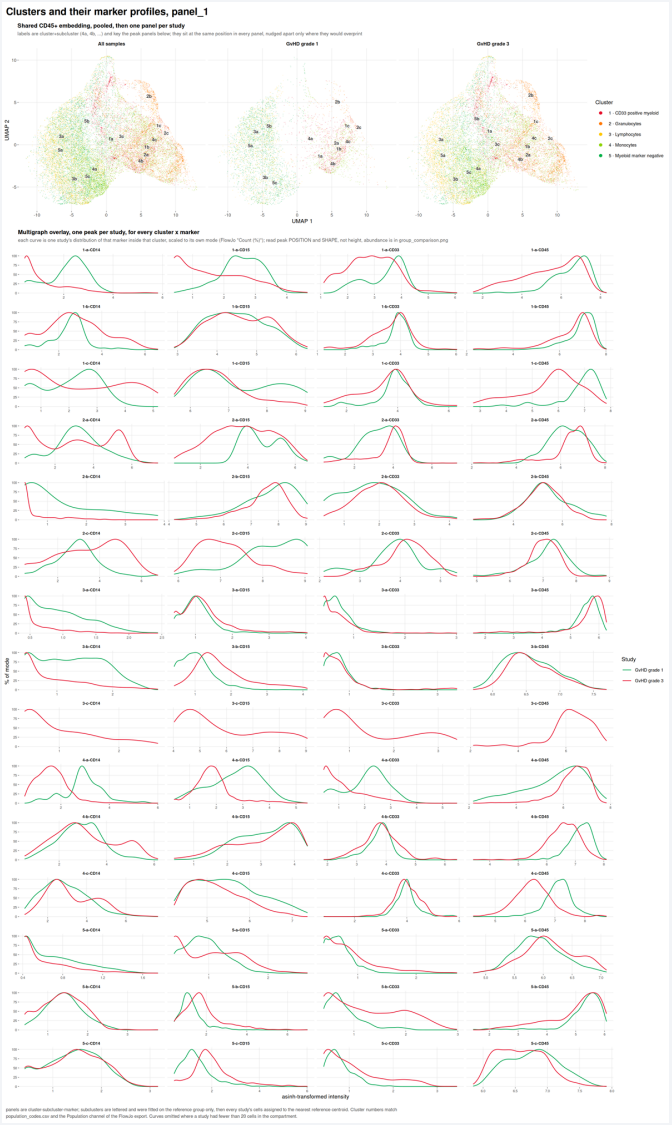


Figure 5.17: Subclusters within each declared population drawn on the shared embedding

unsupervised_clusters.png

A self-organising map clustering of the same cells, computed without reference to the specification.

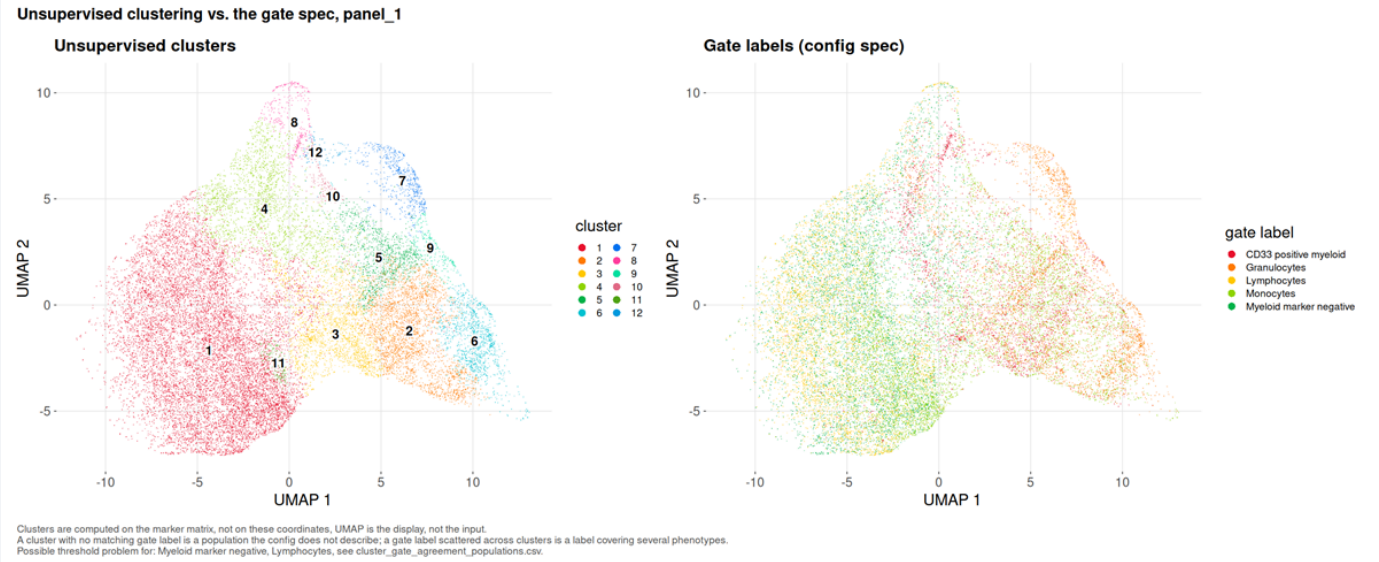


Figure 5.18: Self-organising map metaclusters on the shared embedding, with their declared-label composition

What to look for. A cluster dominated by the unassigned label is a population the specification does not describe. A declared population holding far fewer cells than the cluster it dominates is a misplaced threshold.

This run. Twelve metaclusters over 22,702 cells, and no undescribed cluster: the specification accounts for every phenotype the data contain. Two findings follow that a frequency table cannot deliver.

Two populations are flagged **suspect threshold**. Lymphocytes and Myeloid marker negative both have their dominant cluster in cluster 1, and both sit at an effective cluster count near one (1.30 and 1.54), meaning each is a single phenotype. But cluster 1 holds 11,724 cells and is 36.9% Lymphocytes against 35.8% Myeloid marker negative. The two definitions, CD45+ SSC-A low CD33- CD14- and CD45+ CD14- CD15- CD33-, select largely the same cells in this panel. They are not independent readouts, and section 6 tests them as though they were.

Three populations are flagged **fragmented**: Granulocytes spans an effective 6.13 clusters, CD33 positive myeloid 5.91, Monocytes 3.39. Each label covers several distinct phenotypes that a four-colour panel cannot separate.

5. Why a threshold did not resolve

acquisition_qc.png

Events recorded in each equal-width slice of the acquisition, with the flagged slices marked.

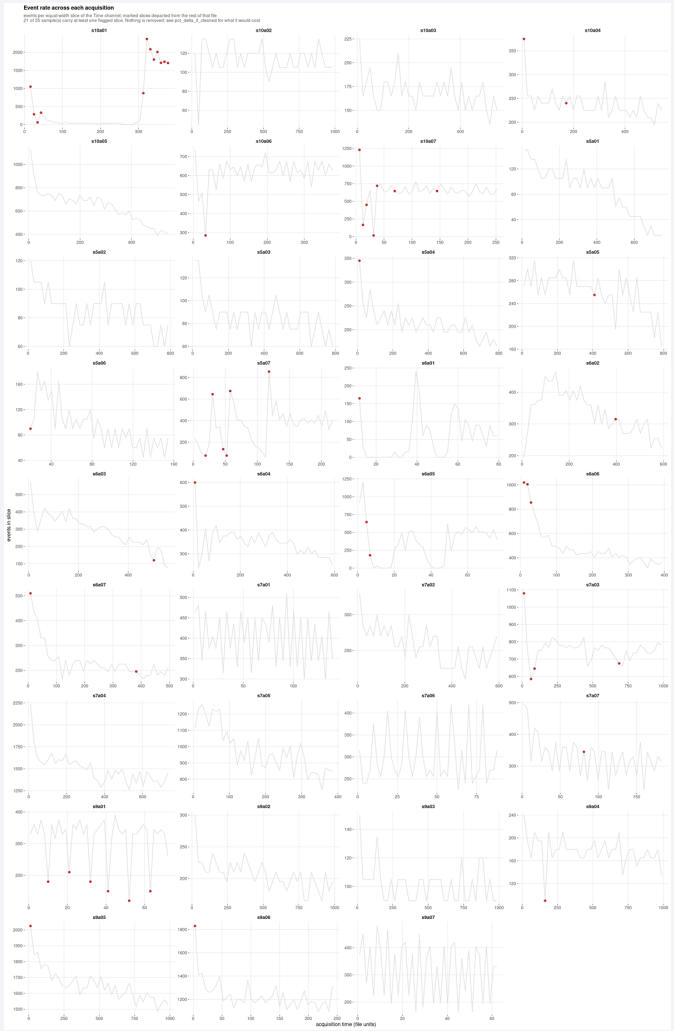


Figure 5.19: Event rate across each acquisition with the flagged intervals marked

What to look for. A flat line is a steady acquisition. A sustained trough is a partial clog, a spike is usually a bubble, a step is a settings change.

This run, and its main finding. Only **14 of 35 samples are stable**; 11 are minor and **10 are unstable**. Excluding every flagged slice would move the largest population by **5.749 percentage points**.

That number is what makes the flag actionable. It is an order of magnitude larger than the gate uncertainty on the same populations, so unlike a cosmetic instability this one would materially change the reported frequencies. On this cohort the honest reading is that acquisition instability, not gate placement, is the dominant technical term, and the files carrying it deserve inspection before any frequency from them is published. Nothing is removed unless `--drop-unstable-events` is given.

Spillover spreading

`spreading_receivers.csv` and `spreading_pairs.csv` have no figure; they are ranked tables. They belong here because they answer the question section 1 left open.

Receiver	Worst source	Widening	Fallback rate
CD15	CD33	3.06x	0.40
CD45	CD15	1.51x	0.40
SSC-A	CD15	1.38x	0.26
CD14	CD45	1.10x	0.57
CD33	CD14	0.99x	0.43

CD15's negative population is three times wider when CD33 is bright, and CD15 falls back to a quantile in 40% of samples. Spreading fills in the valley a threshold would sit in, so those cuts are unresolved for an optical reason that no gating strategy recovers.

CD14 is the instructive counter-case: it has the **highest** fallback rate at 57% and receives almost no spreading. Its failure to resolve has a different cause, and the table is what separates the two.

6. The results

group_comparison.png

Per-population abundance between groups, as per-sample distributions.

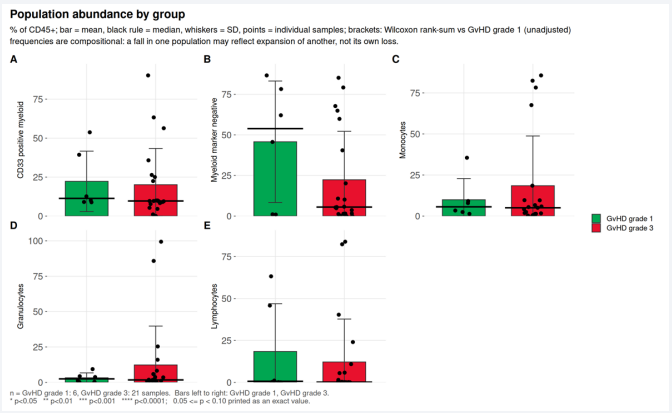


Figure 5.20: Per-population abundance between GvHD grades as per-sample distributions

What to look for. Adjusted p, then Cliff's delta, then `difference_over_gate_u`, then `difference_over_total_u`.

This run. Of five populations, **none reaches raw $p < 0.05$** and none survives correction. The largest effect, Myeloid marker negative, has a Cliff's delta of -0.254 and $p = 0.37$.

That population is worth dwelling on, because it is exactly the case the four-way reading order exists for. Its `difference_over_gate_u` is **11.29**: the difference between group medians is eleven times the distance the gate itself moves, so it is not an artefact of threshold placement. And it is still not significant, because with seven samples in one group and 28 in the other, drawn from five donors, the between-donor variation swamps it. A run that stopped at "the difference is much larger than the gate uncertainty" would have called this a finding. The donor-level test is what prevents that.

marker_state.png

Differential state per population and marker, on per-sample values.

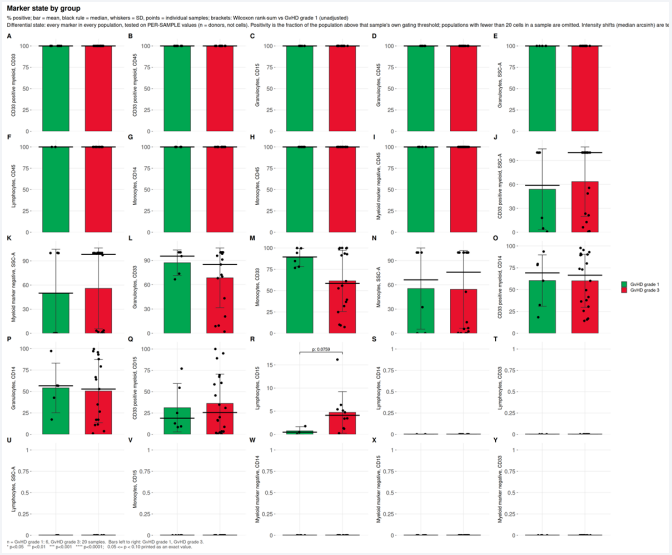


Figure 5.21: Percent positive per population and marker, compared between GvHD grades

This run. 50 tests, none reaching raw $p < 0.05$.

functional_markers.png

Marker intensity read within an already-defined population, rather than used to define one, scoped by the `functional_blocks:` section of the configuration.

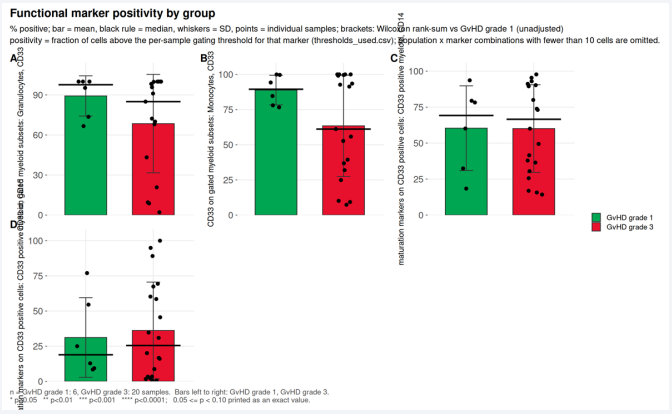


Figure 5.22: Marker intensity within scoped populations, compared between GvHD grades

What to look for. Whether the marker is one of the definers of the population it is being read in. Testing CD15 inside a CD15-positive gate returns 100% in every sample: zero variance, an undefined p-value, and a result that measures the definition rather than the biology. Scoping exists to prevent that, and the demo configuration splits its two blocks precisely so no marker is reported inside a gate its own threshold helped draw.

This run. Four tests across two blocks. CD33 on gated granulocytes has the largest effect, Cliff's delta -0.333 at $p = 0.24$. CD14 and CD15 within CD33 positive cells both return a Cliff's delta of exactly 0.000 at $p = 1.0$, which is a real null and not a degenerate one: the values span 14.2% to 97.7% and 0.4% to 100% respectively, and the two grade groups interleave exactly.

population_ratios.png

Ratios of declared populations, from the `ratios:` section, tested with the same statistics as any abundance.

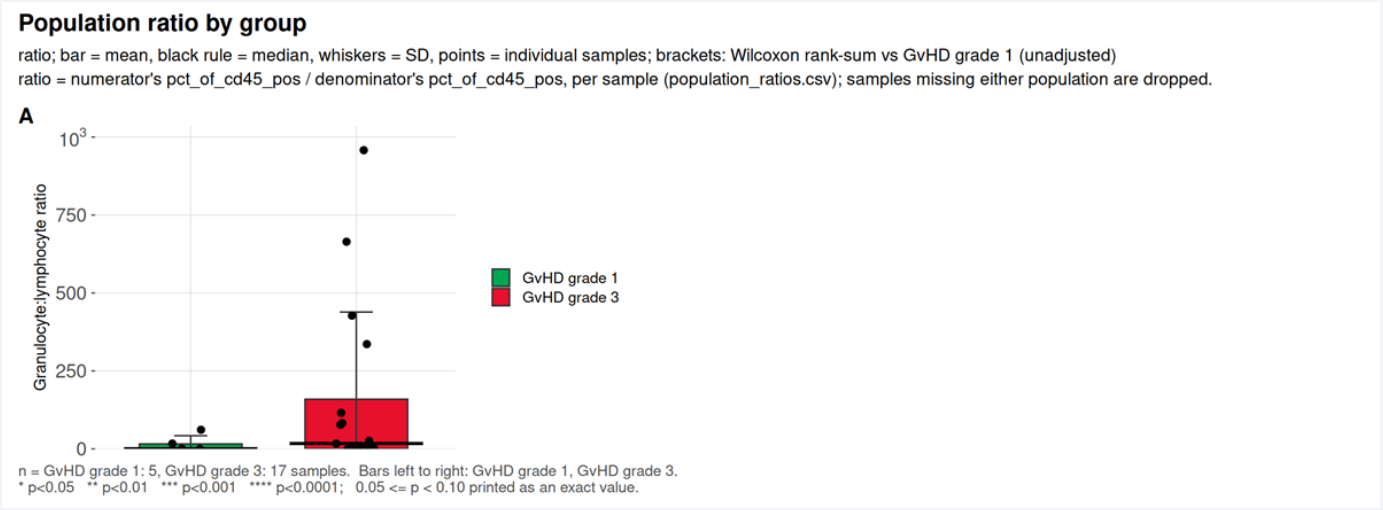


Figure 5.23: Declared population ratios compared between GvHD grades

What to look for. A ratio is not recoverable from either frequency alone when both are percentages of the same parent, which is why it is declared rather than derived after the fact.

This run. One ratio, granulocyte to lymphocyte, a conventional inflammatory index. Cliff's delta 0.271 at $p = 0.39$.

7. Is the contrast confounded

threshold_drift.png

Whether each marker's per-sample thresholds separate by study group.

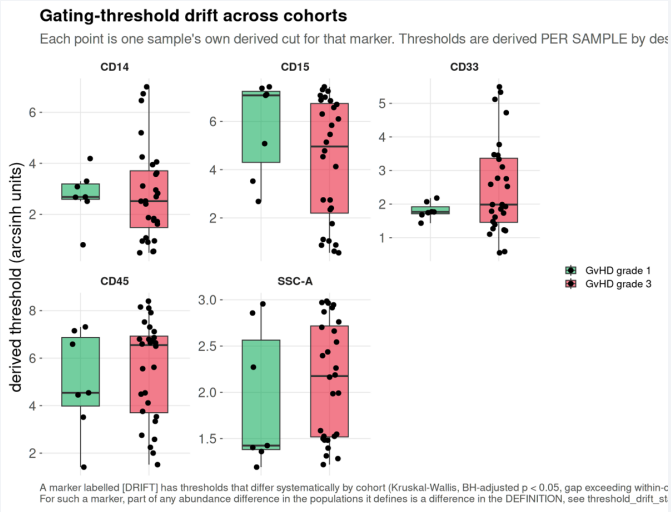


Figure 5.24: Per-sample thresholds per marker, checked for separation by study group

This run. Five markers tested, none flagged. The thresholds do not track GvHD grade, so no part of a between-group difference here is definitional.

batch_diagnostic.png

Batch structure in the shared embedding, against a permutation null.

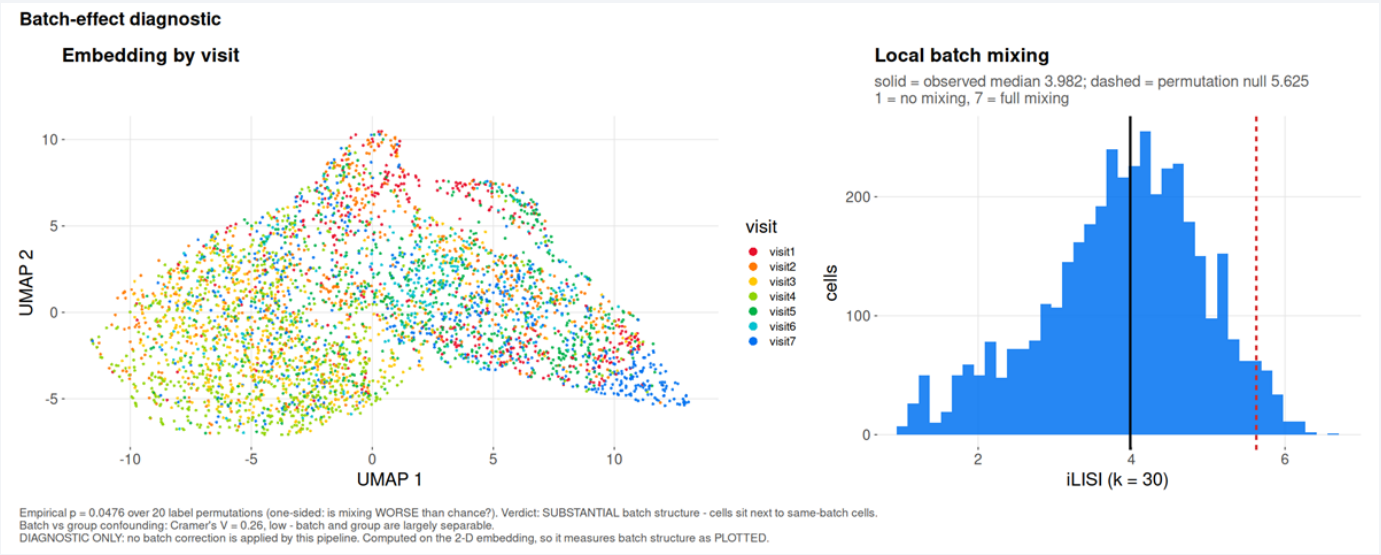


Figure 5.25: Batch mixing in the shared embedding against a permutation null

This run. iLISI **3.982 against a null of 5.625**, p = 0.048: substantial batch structure, meaning cells sit next to cells from the same visit more often than chance allows.

`marker_batch_drift.csv` names the channels responsible. **All four markers differ between visits by at least 1.1 times their own spread**, the worst being CD14 at 1.24 MAD between visit 4 and visit 7.

The question that decides what may be done about it is separability, and `batch_group_confounding.csv` answers it: Cramér's *V* between visit and grade is **0.26**, reported as *low, batch and group are largely separable*. Correction is therefore permitted here. Had the grades been acquired in distinct periods, *V* would approach one and `--correct-batch` would refuse, because removing the batch and removing the finding would be the same operation.

The two figures this cohort cannot produce

That is everything the GvHD cohort supports. The package writes two more, and both need an input no public FCS archive carries.

`absolute_counts.png` and `absolute_counts_qc.png` are written by `--absolute-counts`, which takes a table of measured cells per microlitre per sample and population, from bead-based or volumetric counting on the instrument. A frequency is a proportion of a parent gate, so it moves when any other population moves; an absolute count does not. The QC figure draws every supplied count as one log-scale tile per sample and population, which is how a transcription error (a tile decades off its column) or a sample-name mismatch (a missing row) is caught.

They are absent here because the cohort has no volumetric measurement, and a count invented to fill the gap would be indistinguishable in the rendered figure from a measured one. The input format is documented in the Every output file.

Where the clinical figure designs come from

The clinical family, `clinical_variables_correlation.png`, `clinical_association.png`, `clinical_effects_<variable>.png`, `clinical_landscape.png` and `population_trajectories.png`, is not drawn from this cohort, which carries no severity score. What each one is modelled on:

Figure	Convention it follows
`group_differences.png`	The volcano plot as used in cytometry: effect on x, -log10 p on y, so the largest shifts and the strongest evidence are read in one glance (Teiko, *identifying immune differences with volcano plots*). The x-axis is Cliff's delta rather than a fold change, because a fold change of medians is unbounded and at single-digit group sizes is decided by whichever sample sits at the median
`clinical_effects_<variable>.png`	Effect with its interval rather than a p-value alone, the central recommendation of the prognostic-marker reporting guidelines (REMARK explanation and elaboration), with percentile bootstrap intervals as used for Cliff's delta (*Cliff's Delta Calculator*)
`clinical_variables_correlation.png`	The correlogram convention: circle area for the absolute coefficient, fill for its sign, the number printed only where it reaches significance (*Correlation-based network generation, visualization and analysis*, BioMed Research International)
`clinical_landscape.png`	The annotated-matrix layout of immunoprofiling papers: a z-scored molecular matrix with clinical annotation bars above it, columns ordered by severity (*An immune dysfunction score for stratification of patients with acute infection*; *Multi-omics resolves a sharp disease-state shift between mild and moderate COVID-19*, Cell)
`population_trajectories.png`	Per-patient trajectories split by outcome, the figure that separates survivors from non-survivors in longitudinal sepsis cohorts, where the two differ in direction of travel rather than at any single timepoint (*Longitudinal immune profiling in sepsis*, medRxiv)

Two departures from those conventions, both deliberate. The volcano draws the smallest p-value the design can reach, which no published volcano does and which at these group sizes is often above 0.05. And a Kruskal-Wallis tile stays grey with an unsigned epsilon-squared rather than being coloured on the signed scale, because a three-level variable has a magnitude and no direction.

What this example does and does not show

It shows a pipeline behaving correctly on real, imperfect clinical data: a scatter gate that had to fall back, 72 thresholds that could not resolve with an optical explanation for most of them, acquisition instability large enough to matter, two declared populations that the unsupervised check shows are largely the same cells, real batch structure that is nonetheless separable from the contrast, and a difference eleven times the gate uncertainty that is still not a finding once donors rather than cells are the replicates.

It does not show sensitivity. The group contrast is unbalanced, confounded with donor, and null at this sample size. Nothing here establishes that cyRAVEN detects a real biological effect; it establishes what the diagnostics report when the data are difficult.

6 Limitations

Every caveat in one place, and the methods left out, with the reason for each.

Read this before quoting a number from a run. Nothing here is hypothetical: each limitation is something the pipeline can do to a real result, and each exclusion is a decision that has been argued rather than an oversight.

Known limitations

Every caveat below exists somewhere else in this repository already, in a code comment, a NEWS entry or a function's documentation. Scattered like that, a new user has no way to see the aggregate risk before deciding whether to trust a result. This page is the single place that collects them.

The same page also records what was considered and **not** added, and why. That half exists so the same arguments do not get relitigated every six months.

Part 1. Limitations

A control used to be believed too easily

`staining_verdict()` read an **absent** `is_control` column as "might be a control", because only `is_control = FALSE` positively asserts that a file is a biological sample. On a sheet without that column, any sample whose CD45 gate found no density minimum was promoted to the unstained reference for its whole panel, and every threshold there became the 99.5th percentile of that one sample. Where it was really a stained sample, every population collapsed, with no error raised and no warning printed.

How to check whether a result you already have was affected. Open `thresholds_used.csv`. If `source` reads `control_q995` on most markers, those thresholds came from a reference. Find out which sample supplied it before reading a single frequency. A cohort that declares no control should show `valley`, `tailgate` and `quantile_fallback` only.

Fixed. `--discover-controls` restores the old behaviour for anyone who needs it.

The version number will not tell you whether you are affected, so check the file. A `1.0.0` image was published before this fix and replaced afterwards under the same tag, so two pulls of `bhagesh/cyraven:1.0.0` can differ. The `source` column in `thresholds_used.csv` is the reliable test, and it costs a few seconds.

Automated gates have not been benchmarked against independent expert gating

`learn_convex_gate()` and `explain_cluster()` are validated on held-out cells, and, where donor count allows, leave-one-donor-out. Both of those are internal: the only cohort they have been measured on is the demonstration one. Nothing here establishes that the defaults generalise to a different panel, a different instrument, or a different laboratory's staining.

Two consequences. The hardcoded constants in `density_valley()`, `min_rel_depth` among them, may have been tuned to one cohort without anyone intending to tune them. And a learned gate is descriptive: it carries no p-value and is not evidence that the population is real.

Read the minimum, not the median. On a 21-donor cohort, three clusters gave median F1 of 0.987, 0.731 and 0.189 with worst-donor F1 of 0.953, 0.111 and 0.008. Cells held out from a fit come from the same donors, acquired in the same tubes on the same day, so they flatter the model. Only the per-donor minimum tells you whether a gate survives contact with the next patient.

No mixed-effects differential abundance

The tests are rank-based: Wilcoxon rank-sum, Kruskal-Wallis, Wilcoxon signed-rank and Friedman when paired, with Benjamini-Hochberg within families. Parametric equivalents sit beside them with their assumptions attached.

None of those can model donor and batch as **crossed random effects**. They can stratify or pair, which is not the same thing. A GLMM with a count offset is the standard remedy, and `statistical_methods.csv` names it in the catalogue without computing it. That gap is deliberate for now rather than overlooked, and it is recorded in the backlog.

A single convex gate can misrepresent a multimodal population

`learn_convex_gate()` fits one convex polygon per population. Nothing currently checks whether the label being fitted is a single coherent region in the chosen marker pair before fitting it. A hull drawn around two disjoint clusters of the same nominal population reports a large, low-precision polygon that "explains" the label without describing the geometry.

If a population is known to be phenotypically heterogeneous, check `spec_gaps.csv` from explore mode, which reports how many effective clusters a declared label spans.

FMO disagreement is computed but not enforced

`fmo_agreement.csv` reports the distance between a derived cut and its FMO-anchored equivalent in units of that threshold's own uncertainty. Beyond about three units, one of them is wrong. Nothing acts on that: the file is written and the run proceeds. An unattended pipeline will not stop on it, unlike `--fail-on-drift` for baseline conformance.

Spectral panels can fragment a cohort silently

The panel fingerprint is the set of resolved marker names, so a channel present in some files and absent in others makes them separate panels, each with its own cofactor, thresholds and embedding. Spectral unmixing writes extracted autofluorescence back as extra channels and how many appear varies per acquisition, so twelve comparable files can become seven panels of one to three files each.

Nothing fails when this happens. Every file loads and every table is written. `--list-channels` now names the offending channels and prints the exact `--ignore-channels` pattern that merges the cohort, so run it before a long run.

There is no unmixing-specific QC. `spreading.R` diagnoses conventional spillover spreading well, but poor reference spectra and autofluorescence misattribution are distinct failure modes it does not capture.

Everything is documented on one cohort

The README, the worked example and the container smoke test all run on the same demonstration cohort: one panel, one biological question. Behaviour on a larger marker panel, on spectral channels, or on a design with three or more groups is exercised by tests but not by a documented worked example.

Batch correction changes intensities, not conclusions, and can be wrong

`--correct-batch` aligns marker distributions and refuses above a Cramér's *V* threshold between batch and group, because at that overlap removing the batch and removing the finding are the same operation. `--batch-method cluster` fits one map per cell type rather than one per file, which is the better choice when a detector shift moves bright and dim populations by different amounts.

Two things to know. The clustering it fits is derived from the data, so a cohort with batch-specific biology will have that biology partly absorbed into the correction. And no correction is applied to the gate hierarchy, which is derived per sample and is already batch-local; correction is descriptive of intensity only.

Part 2. Deliberately not included

Nothing below is missing by accident. Each was considered against what the package already does and rejected for the stated reason. If circumstances change, the reason is what has to change first.

Packages

Package	Overlaps	Why not
AutoSpill	compensation	Installs only from GitHub, which breaks the dated-snapshot guarantee the Docker image rests on. For spectral data, unmixing has already happened on the instrument
CytoNorm	batch normalisation	Same GitHub-only problem, and its API is file-based: `QuantileNorm.train()` takes FCS paths and `QuantileNorm.normalize()` writes new FCS to disk, while correction here happens on an in-memory matrix already read, transformed and gated. Its *method*, one map per cell type, is implemented natively as `--batch-method cluster`
fdaNorm, **gaussNorm**	normalisation	No longer supported, and `fdaNorm` is gone from recent flowStats versions
flowTrans	transformation	The cofactor is derived per panel from the data, so there is nothing to fit. The package also carries a multi-year unfixed biexponential bug
ggcyto, **flowViz**	plotting	One figure system and one theme already. A second plotting idiom costs consistency and adds nothing a reader sees
Rtsne	dimensionality reduction	UMAP only, deliberately. Two embeddings of the same cells invite comparison between two things that are not comparable
diffcyt, **CATALYST**	differential abundance	They test clusters *defined on the subsample*, so the subsample changes what is tested. That is the failure the donor-level design exists to avoid. Note this rejects the paradigm, not the mathematics: diffcyt-DA-edgeR on our own per-population counts is available with `--edger`; a GLMM remains open
FastPG, **Rphenograph**	clustering	FlowSOM is a cross-check here, not the primary method, so clustering speed is not the binding constraint
openCyto	gating	CSV templates applied uniformly across samples. The premise of this package is that one gate copied across samples is the problem
flowAI, **flowClean**	acquisition QC	They remove anomalous events. Acquisition QC here reports what removing them *would* change and deletes nothing unless asked. **PeacoQC** fills that role instead, opt-in with `--peacoqc`
PICAFlow, **CytoPipeline**, **CytoExploreR**	whole workflow	Alternatives to this package, not components of it
CytoML, **flowWorkspace**	FlowJo import, Gating-ML	**Not rejected, deferred.** Reading a `.wsp` into `--external-labels` would let a hand-drawn strategy be scored leave-one-donor-out. Held back on dependency weight: it pulls `flowWorkspace` and `ncdfFlow`, which compile C++. Both now come into the image with PeacoQC (`cytolib` and `RProtoBufLib` already came with flowCore), so that reason is weaker than it was

Capabilities

Capability	Why not
Trajectory inference / pseudotime	Ordering cells along a trajectory assumes the population is a continuum that was sampled densely enough to reconstruct. A gated cytometry panel with a dozen markers usually is not, and a trajectory will still be returned
Batch correction without a confounding check	Available in several other tools. Where batch and group overlap far enough, correcting removes the finding, so the check comes first and the refusal is the feature
Automatic removal of anomalous events	The impact of removal is reported first. `--drop-unstable-events` (the in-house detector) or `--peacoqc` (PeacoQC) performs it only when asked
A p-value on anything that cannot be collapsed to one value per sample	Testing over events treats one deeply acquired donor as many independent observations. Those quantities get an effect size and no p-value
Trained classifiers for label transfer to a new cohort	A classifier transfers a labelling; it does not transfer the reason for it. What moves between cohorts here is a gating strategy, which is inspectable, exportable to the instrument, and can be scored per withheld donor. A model that cannot be read is a worse artefact even when it is more accurate
Cross-panel benchmarking against public expert gates	Wanted, and blocked on a rule that is not worth breaking: third-party FCS does not go into this repository. It needs an external harness, not a vignette
A GUI	The point of the package is that a run is one command with a recorded invocation. A GUI makes the choices unrecorded again, which is the problem it was built to solve

Where to look when something seems wrong

Symptom	File to open first
Frequencies near zero across the board	`thresholds_used.csv`, `source` column
A population is missing entirely	`staining_qc.csv`, then `gating_qc.png`
More panels than expected	`--list-channels`, which now names the fix
A comparison you expected is absent	`design_feasibility.csv`
A difference that looks too good	`threshold_drift_stats.csv`, then `difference_over_gate_u`
Correction ran or did not	`batch_correction.csv`

Scope: what cyRAVEN does not do

Nine methods available in comparable toolboxes are absent from cyRAVEN. Each section states the omission, the reasoning, and the condition under which it should be revisited. Where a method is required now, the Using cyRAVEN with cyCONDOR covers combined use with cyCONDOR.

1. Harmony and CytoNorm

Batch correction is implemented. `--correct-batch` performs monotone quantile alignment and refuses above a configurable Cramér's *V* between batch and group.

These two implementations specifically are not wrapped. Both introduce substantial dependency weight, and neither declines to execute when batch and biological effect are collinear. That refusal is the behaviour cyRAVEN treats as the purpose of the feature.

Revisit when a study design requires reference-based normalisation across instruments rather than distributional alignment within a batch.

2. Phenograph

FlowSOM addresses the same requirement at lower cost at these event counts. Phenograph determines cluster count from the data rather than accepting it as a parameter, which is informative, but requires `Rphenograph` or `Rphenoannoy`. Neither is distributed through CRAN or Bioconductor, and a GitHub-only dependency is incompatible with the reproducibility guarantee the container provides.

Revisit when cluster count is the quantity under investigation rather than a setting.

`--explore-k` is likewise a setting rather than a finding, and `explore_provenance.csv` records it as one, because a cluster count chosen from the data is one more thing the number of significant results depends on.

3. diffcyt

The rank tests address the same hypotheses without distributional assumptions, which is the correct trade at single-digit *n* per group.

The edgeR form is available opt-in with `--edger`. It is applied to the gated populations' counts per sample, not to clusters defined on a subsample, so it avoids the objection in the table above, and it is written beside the rank tests, never in place of them. limma-voom and the GLMM are not implemented.

Revisit at approximately fifteen samples per group, where the empirical Bayes prior becomes informative and mixed models can accommodate repeated measures.

4. Per-marker cofactors

`derive_cofactor()` computes per-marker candidates and returns them in `attr(cf, "per_marker")`. Application is not implemented.

Applying them alters the arcsinh scale per channel, displacing every threshold, marker median and embedding distance. Validation requires re-derivation and re-inspection of every gating QC figure in the affected dataset.

Revisit when gating QC demonstrates channels whose background distributions differ in scale. The data required for that judgement is already collected.

5. Trajectory inference

Pseudotime methods assume a continuous differentiation process sampled across its extent. Peripheral blood immunophenotyping panels predominantly measure terminally differentiated populations, where a fitted trajectory reflects the fitting procedure rather than a developmental process.

Revisit when the panel resolves a developmental axis, for example thymic emigrants or plasmablast maturation. cyCONDOR provides diffusion maps and Slingshot.

6. Label transfer

Classifier-based cell type assignment is standard elsewhere. Here the training labels originate from a gating specification, so a classifier trained on them reproduces that specification including its errors.

Unsupervised cluster concordance is the stronger check for the same purpose, since it does not consult the labels.

Revisit when an externally annotated reference dataset is available for training.

7. FlowJo ingestion

Gates travel outward. `--export-gates` writes learned strategies as ISAC Gating-ML 2.0, and `--flowjo-export` writes FCS with embedding coordinates as additional parameters, so a result can be opened in FlowJo, Cytobank or FlowRepository.

The inverse is absent: cyRAVEN does not read gates from a `.wsp` workspace, which requires `CytoML` and `flowWorkspace`. The export needs neither, because a Gating-ML document is written directly, whereas parsing a proprietary workspace is not something to reimplement.

Revisit when manual FlowJo gates are required as input rather than output.

8. Cell-level testing

Absent for the reason in section 1 of the Statistics: event counts are set by acquisition duration, so a test over pooled events derives its degrees of freedom from instrument time. Toolboxes providing a cell-level Wilcoxon distribute it with a warning against its use.

The cell-level view is available descriptively in `subcluster_marker_shifts.csv`, which reports effect sizes without p-values.

9. Compensation matrices

cyRAVEN applies the spillover matrix stored in the FCS keyword block and reports its absence. It does not compute one from single-stain controls, which belongs in acquisition software where the controls can be inspected.

10. The 2025-2026 methods survey

A survey of cytometry method releases and papers to August 2026, recorded here so the same candidates are not re-evaluated from scratch. Adopting nothing is a decision too, and this is where it is written down.

Worth adopting, ranked.

Method	What it adds that cyRAVEN lacks	Licence
staRgate	Gates a marker against a *unimodal background* population rather than a density minimum. That is the case cyRAVEN gives a tail gate, a fixed 3.5 robust standard deviations above the dominant mode and flagged for review, and on a 29-marker panel across 266 subpopulations it matched manual gating at 0.97 concordance	MIT
CompensAID	Flags, in the stained sample, marker pairs whose source-positive population is skewed in the receiver, the mark of a wrong reference control: it segments that population and flags the pair when the last segment's Secondary Stain Index is below -1, a rule validated against specialists. cyRAVEN's `trend_rho` screens the same region with an unvalidated cut	GPL-3
CytoScan	Between-file anomaly detection, including novelty against a locked reference set. cyRAVEN's acquisition QC is within-file, so a whole tube that is unlike its cohort passes	preprint
CytoMDS	Attributes a batch separation back to the channels that caused it. cyRAVEN's EMD says a batch differs; this says which markers made it differ	GPL-3
Rajwa & Roederer 2025	Effective rank, information efficiency and the Cramer-Rao lower bound matrix, all computable from the spillover matrix already parsed. Not software – three formulas and a vocabulary that makes the existing QC legible to a reviewer	paper

Considered and declined, with the reason.

`sccomp` tests differential *variability* and can transfer priors from an earlier cohort, neither of which the CLR re-test does, but it pulls Stan into the image for a question users have not asked. `propeller` borrows variance across populations, which is the right instinct at n = 5-10, and is worth offering beside the rank tests rather than instead

of them, because it assumes a variance structure the rank tests do not. CytoBatchFlagR is CC BY-NC-ND, which forbids redistribution in a packaged pipeline, and its MSI/frequency/EMD triad largely duplicates what is here. flowMagic reaches 65% accuracy on rare populations, which is where automated gating has to earn its keep, and a trained model is not auditable the way a declared threshold is. GPCT is a Python transformer answering a different question, sample-level outcome prediction rather than population quantification.

What did not change, which is why nothing here chases it. MIFlowCyt has had no revision. Gating-ML is still 2.0 and FCS still 3.2, so the export target is stable. The Cossarizza EJI guidelines are still the third edition. EuroFlow's 2025 updates are wet-lab only. `SOULCAP` intends to standardise population naming against Cell Ontology identifiers but has shipped no ontology, so the whole of the forward-compatible work is an optional `cl_id:` field per population.

And on small cohorts specifically: there is no consensus to follow. No ISAC position paper, no journal guidance, no methods paper aimed at the n < 20 immunophenotyping case. The field is quietly importing `propeller` and `sccomp` from single-cell RNA sequencing. The approach here (rank tests, Cliff's delta, Benjamini-Hochberg within a family, bootstrap intervals, and the attainable-p floor drawn on the figure) is behind no published standard, because there is not one.