Swiss Institute of Bioinformatics

INTRODUCTION TO SPATIAL TRANSCRIPTOMICS DATA ANALYSIS

Normalization & Scaling

Deepak Tanwar

June 03-05, 2026

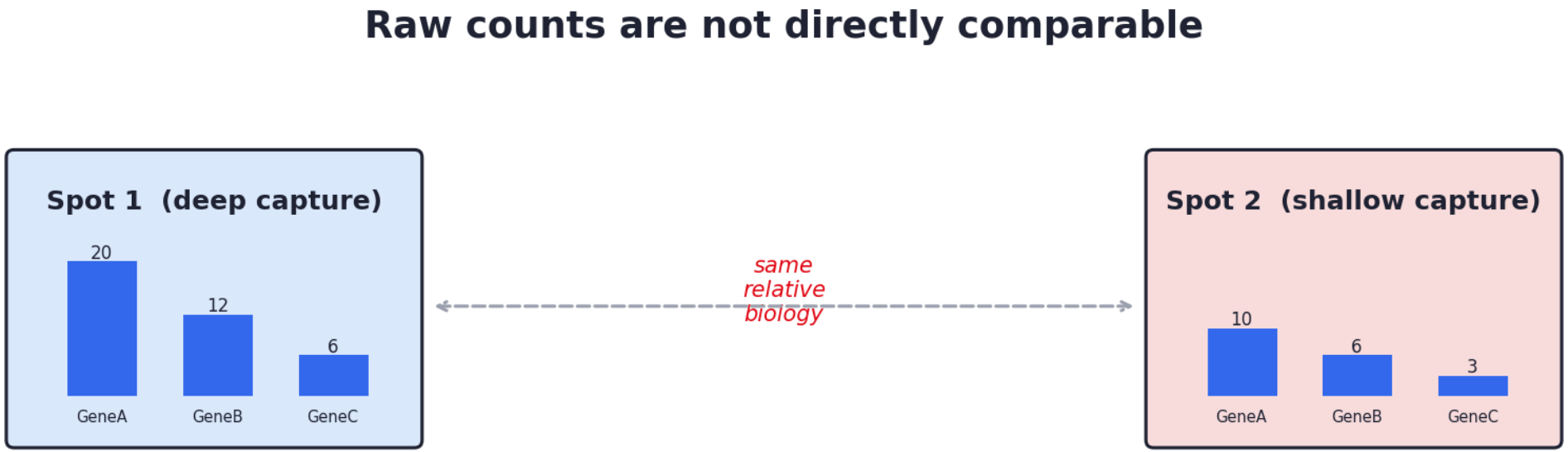
elixir SWITZERLAND

Why do we normalise at all?

Two observations with the **same biology** can show very different total counts.

What technical factors drive that difference?

The problem: counts are not comparable



Totals differ ~2× from capture efficiency / cell size / density — not biology.

Normalisation puts observations on a comparable scale.

Normalisation vs Scaling

Normalisation: makes *observations* comparable

(divide by size factor, then `log1p`)

Scaling: makes *genes* comparable

(centre & rescale each gene; for PCA, clustering, heatmaps)

Always scale **after** normalisation – never on raw counts.

How do you normalize your single-cell data?

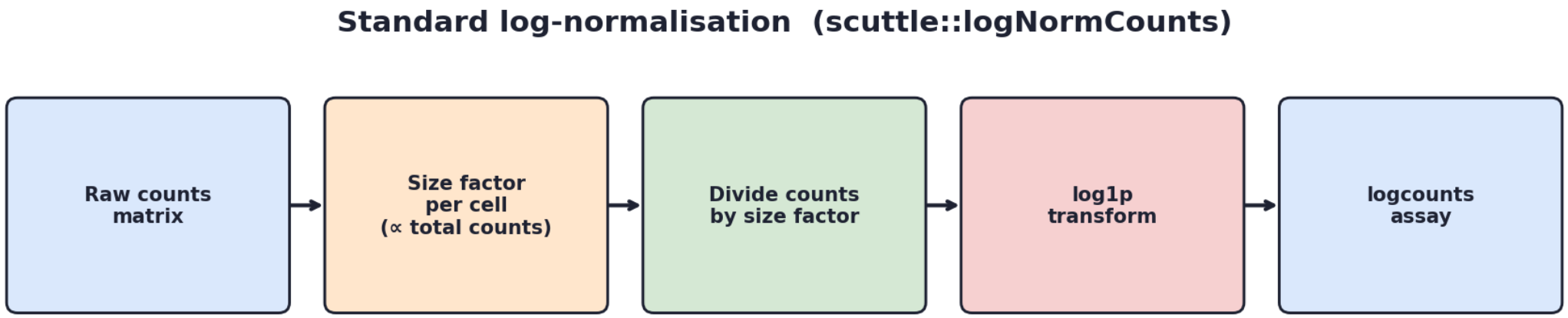
Log-normalization

$$Y_{ij} = log_e((\frac{X_{ij}}{\sigma_i \sum X_{ij}}) + 1)$$

- Simplest and most commonly-used normalization strategy
- Divide all counts for each spot by a spot-specific scaling factor (i.e. size factor)
- Assumes that any cell-specific bias (e.g., in capture or amplification efficiency) affects all genes equally via scaling of the expected mean count for that cell

Standard scRNA-seq normalisation

Log-normalisation by size factors – the scRNA-seq workhorse, also the default for spatial data



Size factors capture technical depth; logcounts stabilise variance for downstream analysis.

Standard normalisation: pros & cons

Pros

- Simple, fast, well understood
- Stabilises variance via `log1p`

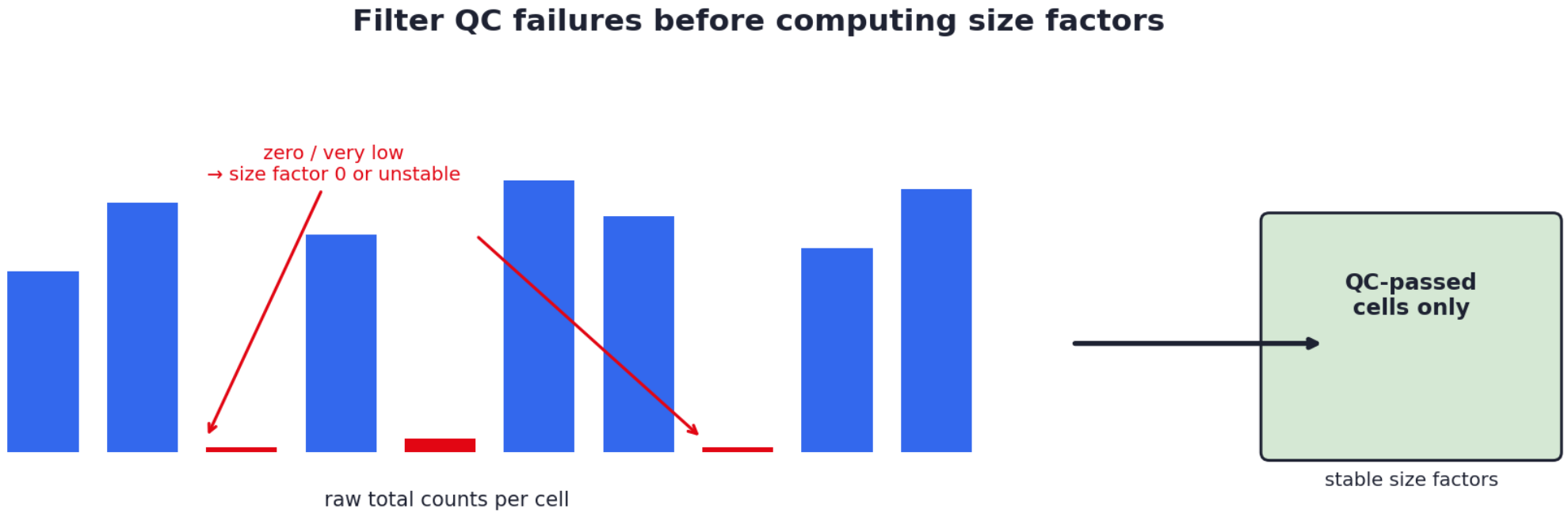
Cons

- Ignores spatial structure entirely

Standard log-normalisation (`logNormCounts`) is the workhorse.

QC-aware normalisation

Normalisation depends on QC: filter before you compute size factors

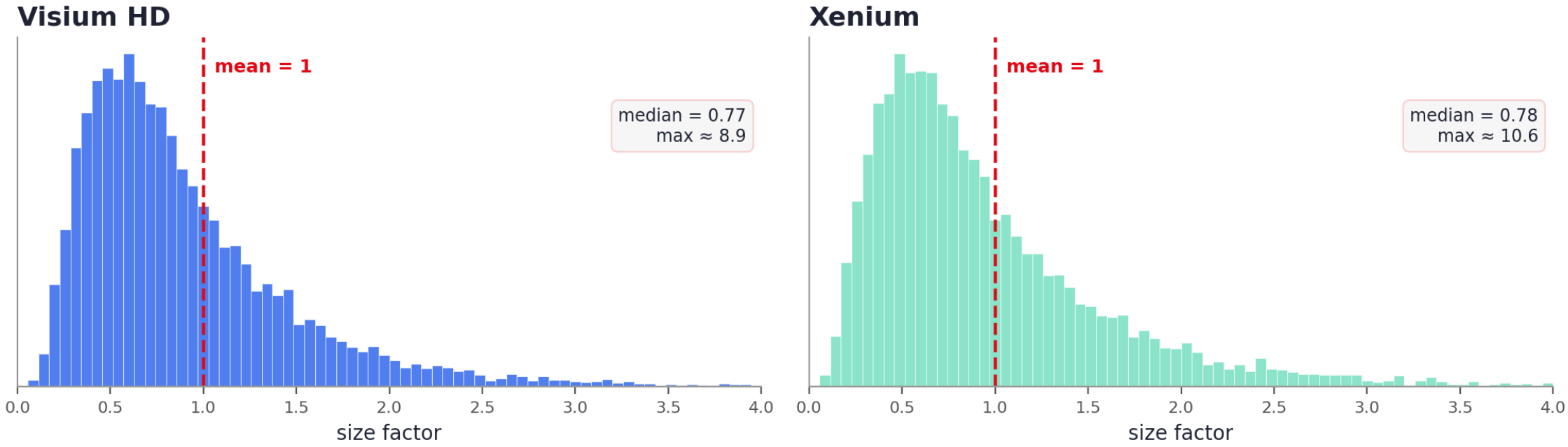


logNormCounts() needs positive totals — zero-count cells must be removed first (esp. Xenium).

What size factors look like

logNormCounts() centres size factors at mean = 1; the spread reflects depth + cell size

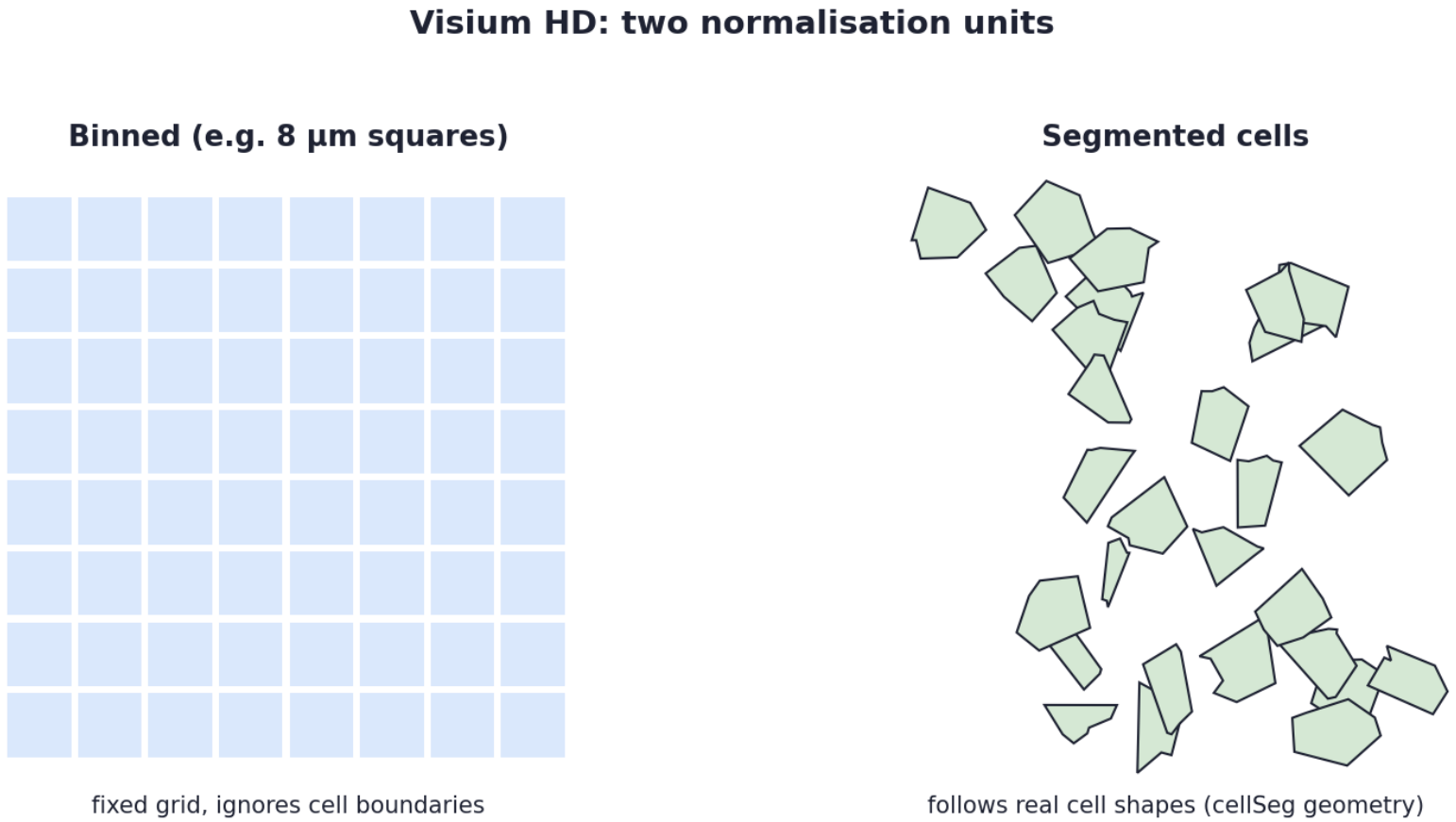
Size factors are right-skewed and mean-centred — a few cells capture far more than average



A long right tail is normal: most cells sit near the median, a few capture far more. Inspect with `summary(sizeFactors(sfe))` and as a spatial plot.

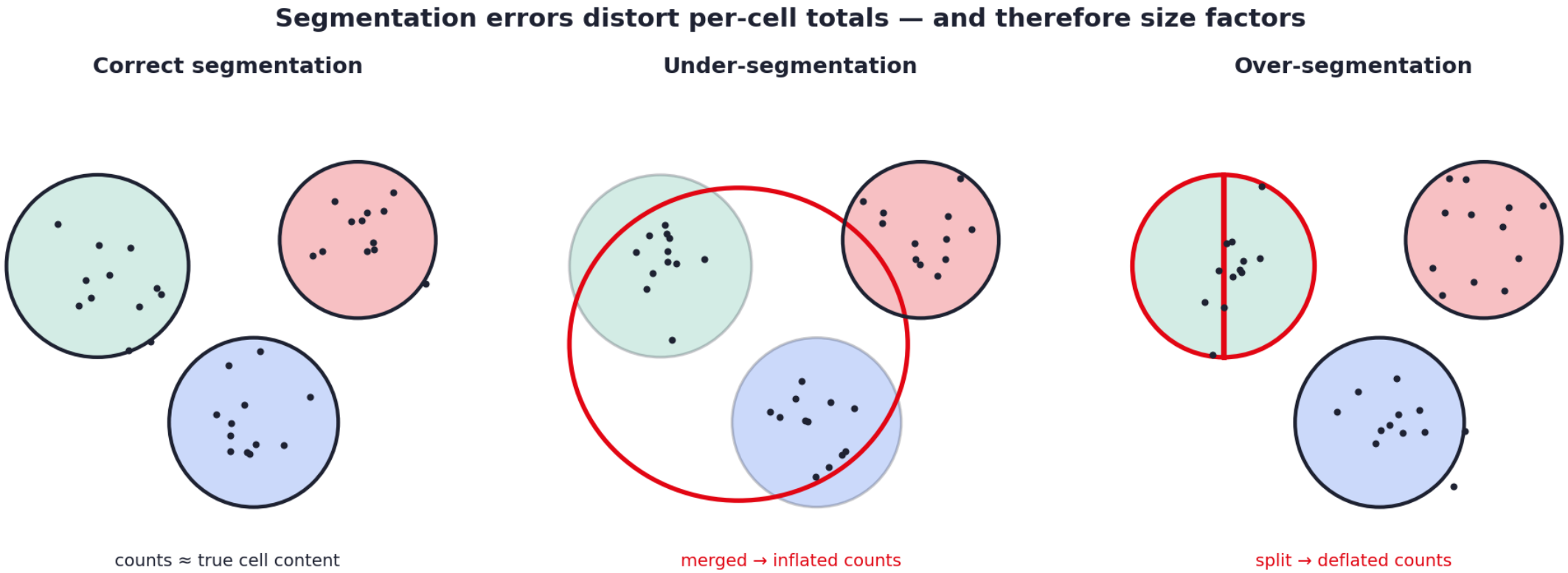
Visium HD: binned vs segmented cells

The exercise uses the segmented-cell object (cellSeg geometry) – counts are per cell, not per square



How segmentation shapes normalisation

Per-cell totals – and thus size factors – are only as good as the segmentation

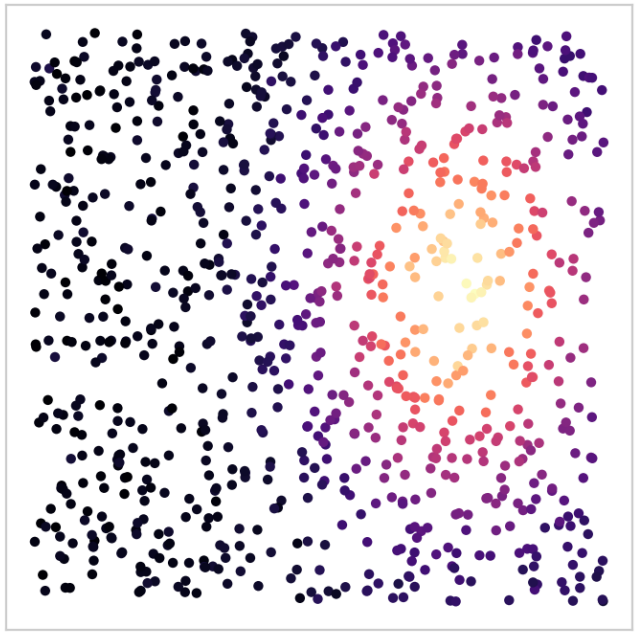


Visualise the effect: before vs after

Plot a known marker in raw vs logcounts space to sanity-check normalisation (Exercise 4B does this for PIGR)

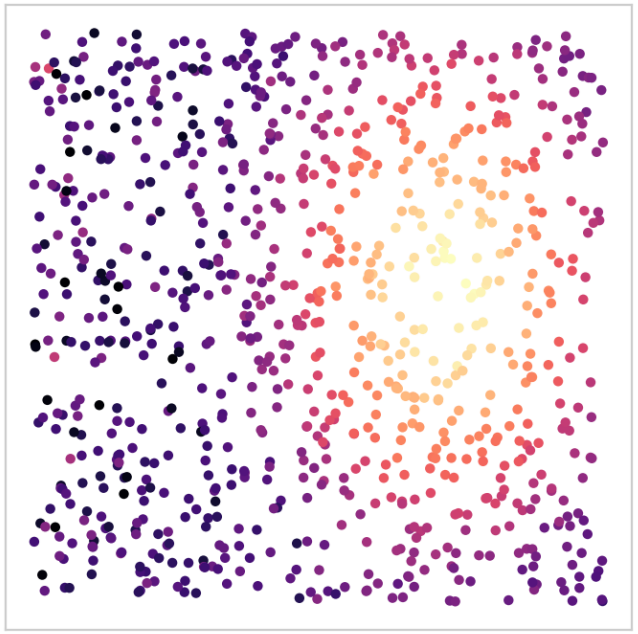
PIGR — raw counts

depth gradient bleeds in (left–right trend)

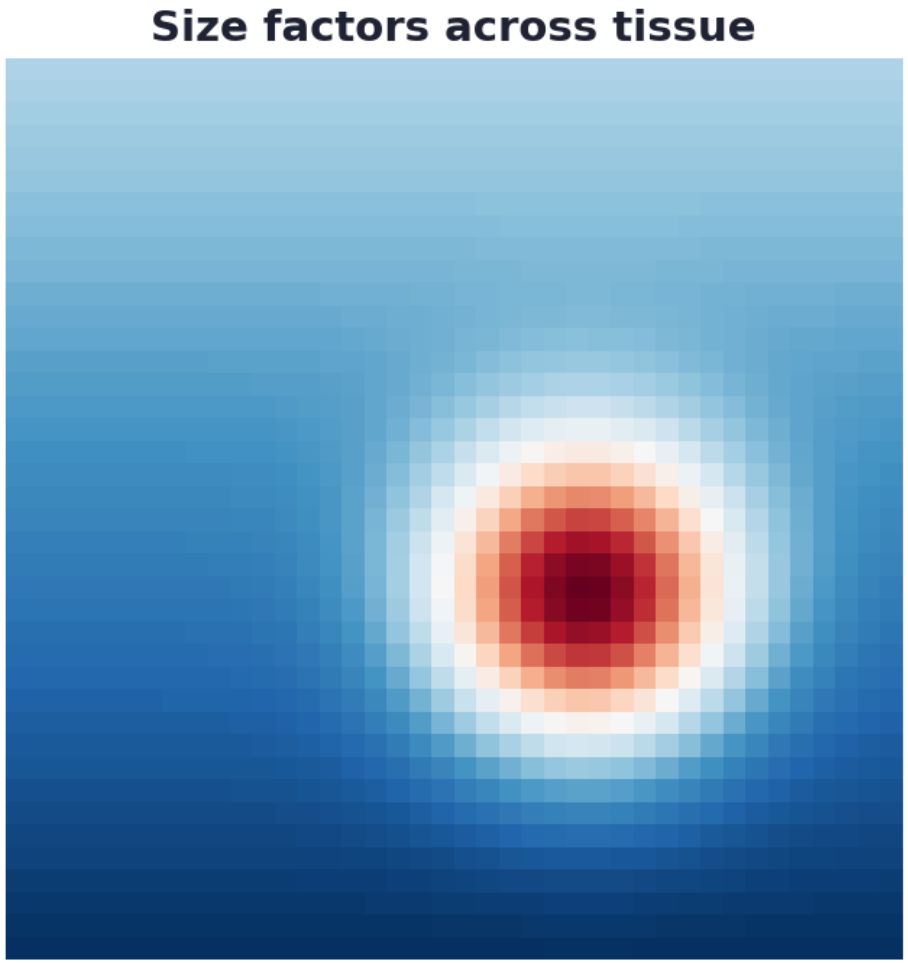


PIGR — logcounts

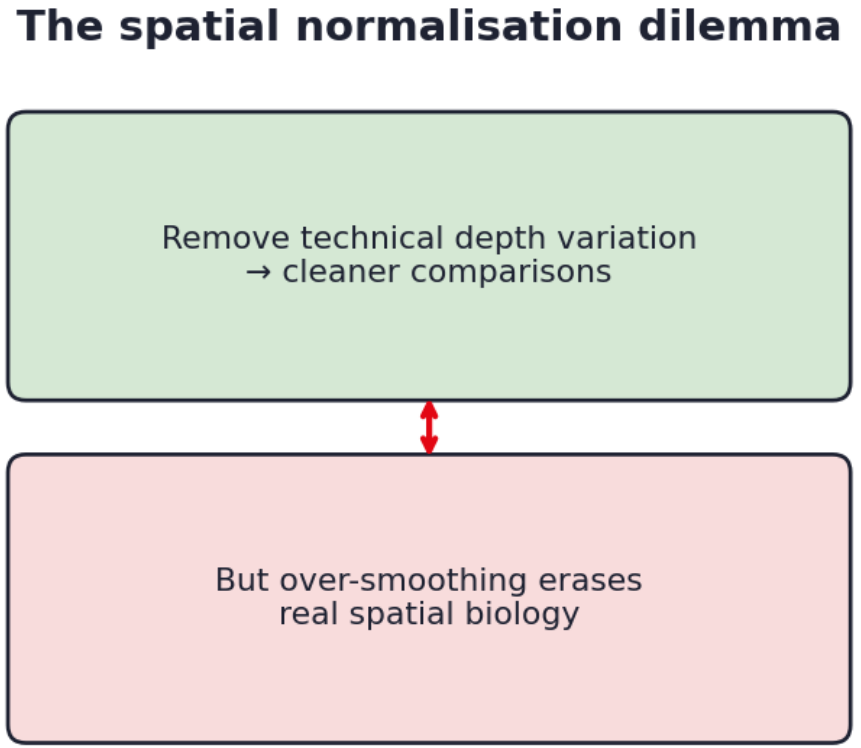
gland stands out; gradient reduced



Spatially-aware normalisation



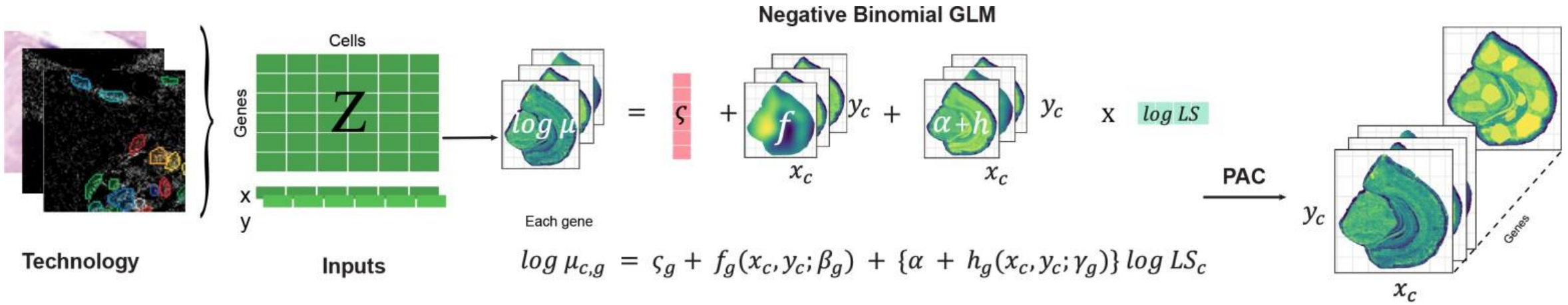
Spatial pattern: part technical, part real biology



Methods (e.g. SpaNorm) try to balance the two

Discussion: OSTA – Orchestrating Spatially-Resolved Transcriptomics Analysis (lmweber.org/OSTA)

SpaNorm



Models library size and biology jointly with a gene-wise NB GLM, then removes only the library-size part (PAC / Pearson residuals).

On Bioconductor: `SpaNorm()` · `SpaNormPCA()` · `SpaNormSVG()`

Salim*, Bhuva*,..., Davis (2025) Genome Biol 26:109 · doi:10.1186/s13059-025-03565-y

Spatial normalisation: methods & challenges

Methods

- `SpaNorm` — models spatial dependence of library size
- Spatial smoothing / neighbour pooling
- Region- or domain-wise size factors

Challenges

- Biology and technical depth are confounded in space
- Smoothing can erase real signal
- Assumptions are tissue-dependent (homogeneous brain vs structured colon)

Salim et al., SpaNorm; see OSTA normalization chapter for the trade-offs.

How about any other method?

How about any other method?

Any method from single-cell field?

Can we use sctransform?

Bhuva *et al. Genome Biology* (2024) 25:99
https://doi.org/10.1186/s13059-024-03241-7

Genome Biology

SHORT REPORT **Open Access**

Library size confounds biology in spatial transcriptomics data

Dharmesh D. Bhuva[1,2,3*], Chin Wee Tan[2,3,4], Agus Salim[2,5], Claire Marceaux[3,6], Marie A. Pickering[7], Jinjin Chen[2,3], Malvika Kharbanda[1,2,3], Xinyi Jin[2,3], Ning Liu[1,2,3], Kristen Feher[1,2,3], Givanna Putri[2,3], Wayne D. Tilley[7], Theresa E. Hickey[7], Marie-Liesse Asselin-Labat[3,6], Belinda Phipson[2,3†] and Melissa J. Davis[1,2,3,4,8†]

- Tested the effects of normalization on spatial domain identification

Though sctransform removes library size effects effectively, their confounding with biology results in removal of biological effects as well.

See also for imaging-based SRT…

Atta *et al. Genome Biology* (2024) 25:153
https://doi.org/10.1186/s13059-024-03303-w

Genome Biology

RESEARCH **Open Access**

Gene count normalization in single-cell imaging-based spatially resolved transcriptomics

Lyla Atta[1,2], Kalen Clifton[1,2], Manjari Anant[2,3], Gohta Aihara[1,2] and Jean Fan[1,2*]

What is sctransform?

The idea

Model each gene's counts with a **regularised negative binomial** regression on sequencing depth.

Returns **Pearson residuals** as the normalised + variance-stabilised values.

Why people use it

- Avoids the arbitrary `log1p` pseudocount
- Removes depth-variance relationship
- Often sharpens HVG selection & PCA

Implemented in `Seurat::SCTransform()`

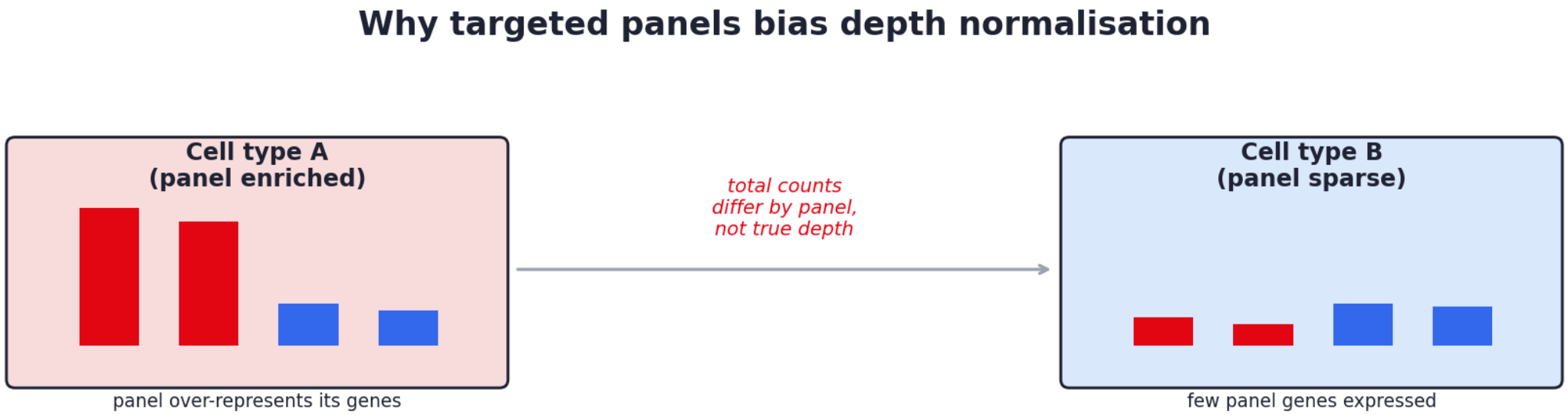
Controversy: sctransform & friends

`sctransform` models counts with a regularised negative binomial instead of `log1p`.

Debated points:

- Does the variance-stabilising step over-correct?
- Parametric assumptions may not hold for spatial data
- Reproducibility across versions (v1 vs v2)

Controversy: biased Xenium panels



Depth-based size factors absorb panel composition → region/cell-type-specific distortion

of normalised values, affecting DE, fold-changes and SVG calls.

Atta, Clifton, Anant, Aihara & Fan (2024). Gene count normalization in single-cell imaging-based SRT. Genome Biology 25:153 · doi:10.1186/s13059-024-03303-w

Controversy: normalise, then integrate?

Multi-sample studies stack two corrections:

- Per-sample normalisation (size factors)
- Batch integration (Harmony, scVI, MNN…)

Debated:

- Integration can erase real spatial/biological differences between samples
- Order and choice of method change results
- Especially fraught with targeted panels

What to use when

Visium HD (Exercise 4A)

- Binned cells, broad capture
- `logNormCounts()` is a sound default
- Check size factors for spatial structure before trusting them

Xenium (Exercise 4B)

- Segmented cells, targeted panel
- Remove **zero-count** cells first
- Normalise **separately** from Visium
- Interpret within the panel – not as whole-transcriptome

Practical recommendations

Defaults that work

- Start with `logNormCounts()`
- QC-filter (incl. zero-count) first
- Scale only after normalisation
- Always plot a marker before/after
- Inspect size factors in space

Heads-up: logNormCounts() is being superseded by `scrapper::normalizeRnaCounts.se()`

Be cautious about

- Targeted panels (Xenium): interpret within-panel, not whole-transcriptome
- Heavy spatial smoothing on structured tissue
- Treating segmentation as ground truth
- Over-tuning sctransform versions

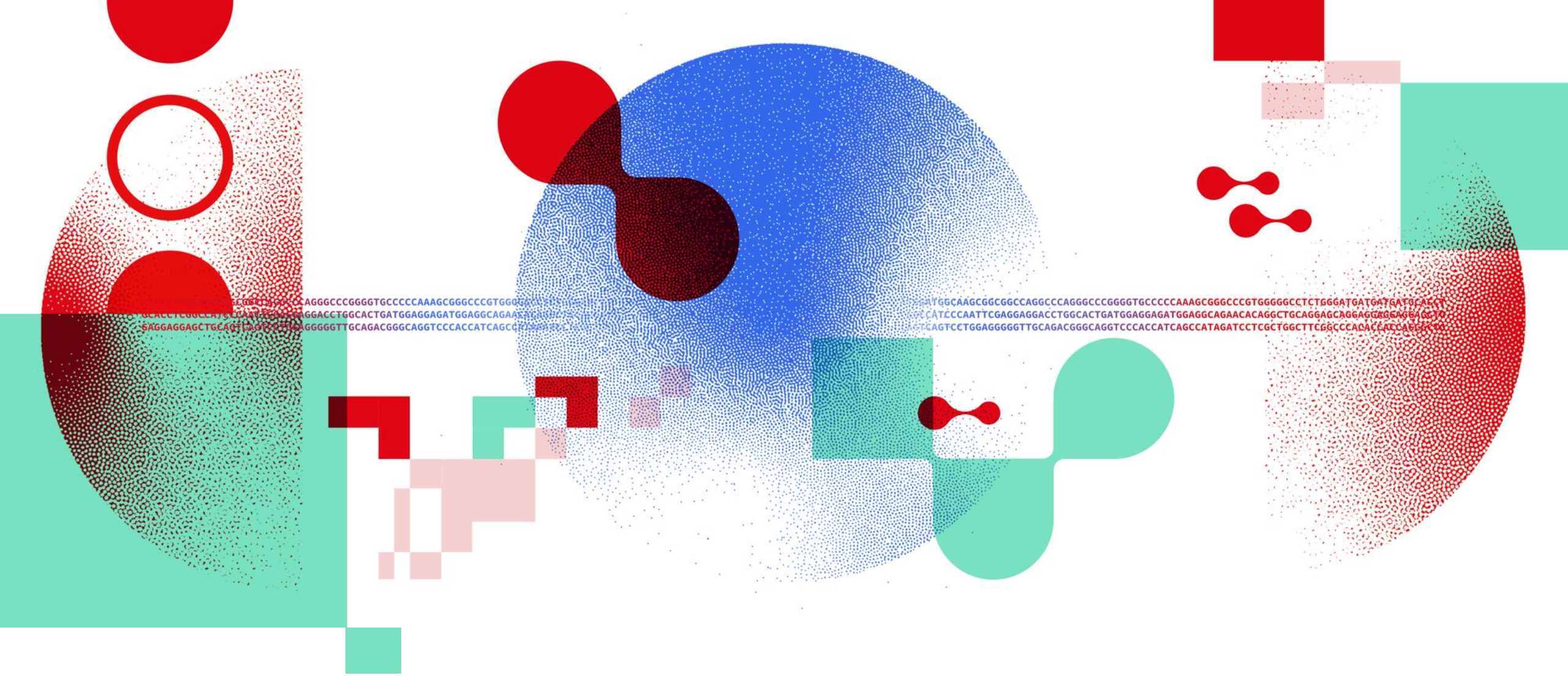
Exercise 4

4A `day2-1a_normalization_visium_segmented.qmd`

4B `day2-1b_normalization_xenium.qmd`

You will:

- Add `logcounts` with `logNormCounts()`
- Inspect size factors & their spatial pattern
- Scale marker genes after normalisation
- Compare Visium HD vs Xenium normalisation



Thank you

DATA SCIENTISTS FOR LIFE

sib.swiss

elixir SWITZERLAND

SIB