Signal Demixing with Tensor Factorization

What have I been up to the past 4 years

IAM Retreat | 13 April 2026 | Vancouver, BC

Nicholas Richardson

Department of Mathematics

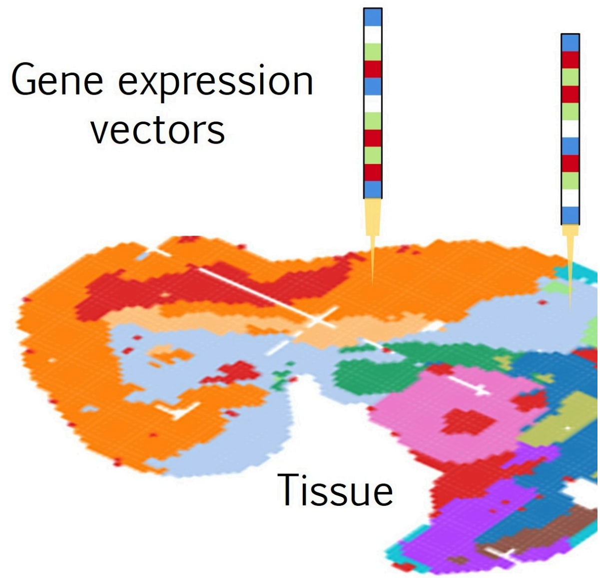
THE UNIVERSITY OF BRITISH COLUMBIA

INSTITUTE OF APPLIED MATHEMATICS
University of British Columbia

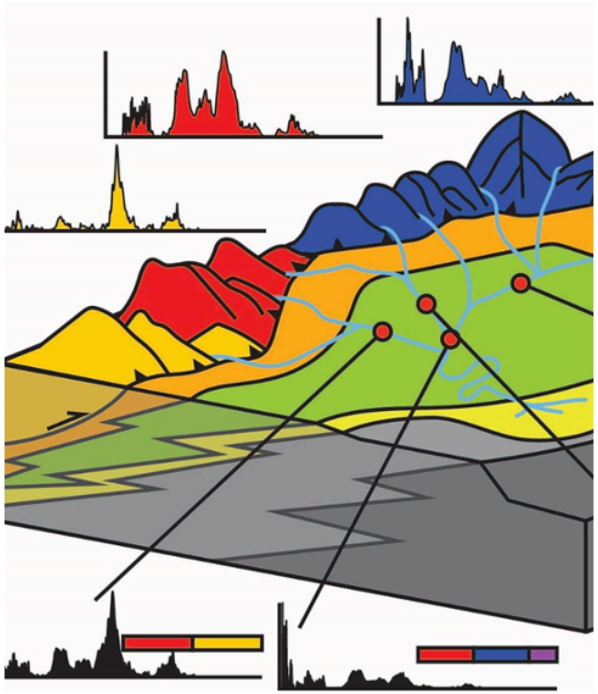
I'm getting mixed signals…

Many real world applications involve mixtures of data

Songs are mixtures of different instruments



Gene counts are influenced by cell types

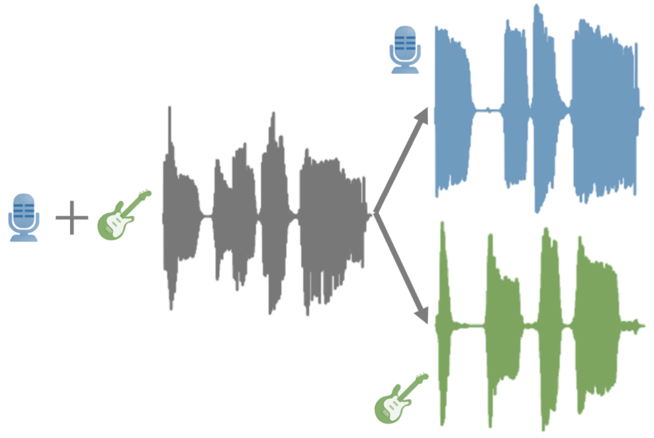


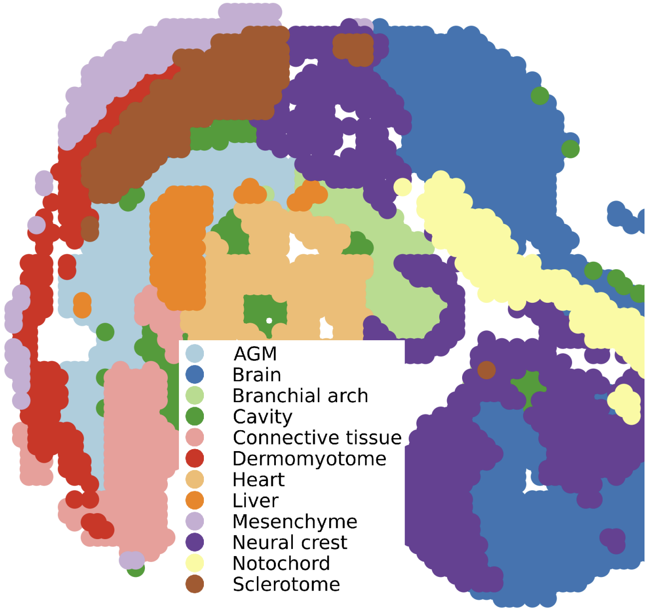
Rocks are mixtures of mineral sources

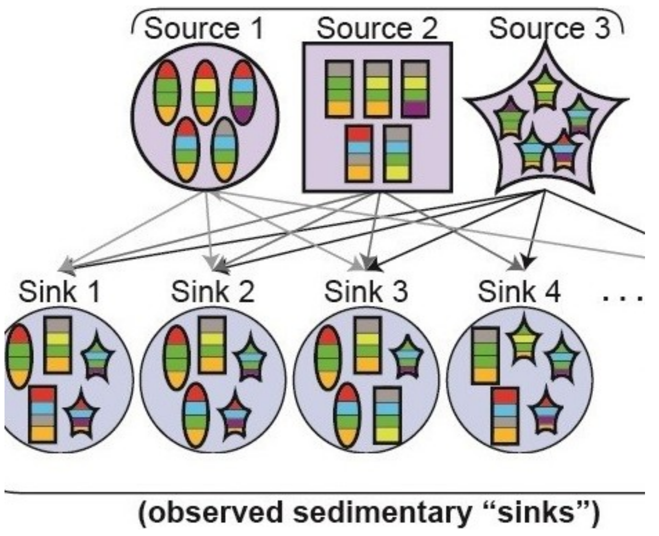
What do we want?

Goal: Given data *without labeled examples*, determine the underlying sources.

Why? For IDing sources, pre-processing, de-noising







Model: Multiple Mixtures

$$\mathbf{y}_1 = a_{11}\mathbf{b}_1 + a_{12}\mathbf{b}_2 + \cdots + a_{1R}\mathbf{b}_R$$

$$\vdots$$

$$\mathbf{y}_I = a_{I1}\mathbf{b}_1 + a_{I2}\mathbf{b}_2 + \cdots + a_{IR}\mathbf{b}_R$$

Package data into a tensor $\mathbf{Y}$ by sampling the mixtures $\{\mathbf{y}_i\}$ or stacking the observations

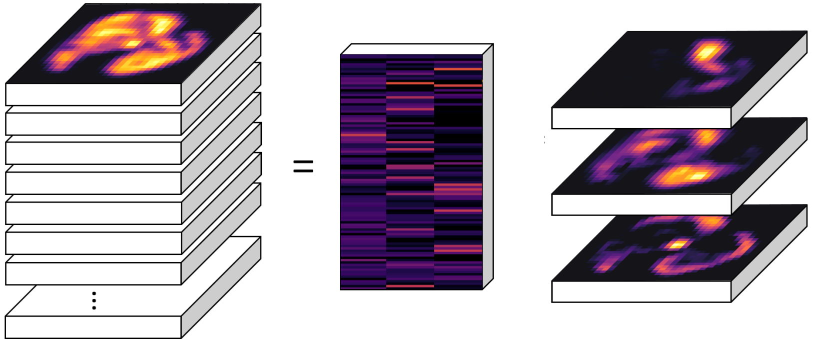


Figure 1: Example data tensor $\mathbf{Y}$ and it's decomposition into $\mathbf{A}$ and $\mathbf{B}$.

Model: Tucker-1 Tensor Factorization

- Factorize $\mathbf{Y}$ into a mixing matrix $\mathbf{A}$ times a source tensor B using the Tucker-1 model (Kolda and Bader 2009)
- $\mathbf{Y} = \mathbf{B} \times_1 \mathbf{A}$ with the entry-wise equation
- $\mathbf{Y}[i, j_1, \ldots, j_N] = \sum_{r=1}^{R} \mathbf{A}[i, r] \cdot \mathbf{B}[r, j_1, \ldots, j_N]$

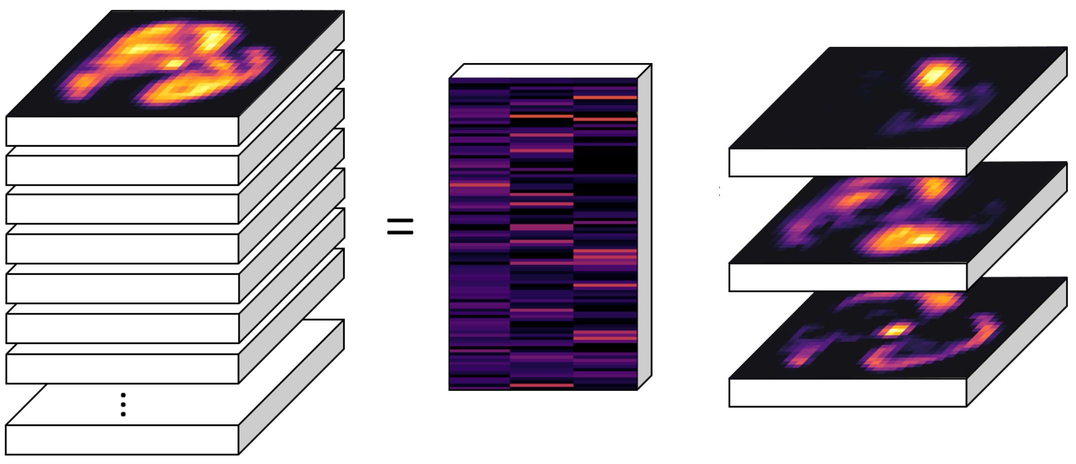


Figure 2: Example data tensor $\mathbf{Y}$ and it's decomposition into $\mathbf{A}$ and $\mathbf{B}$.

Application: Spatial Transcriptomics

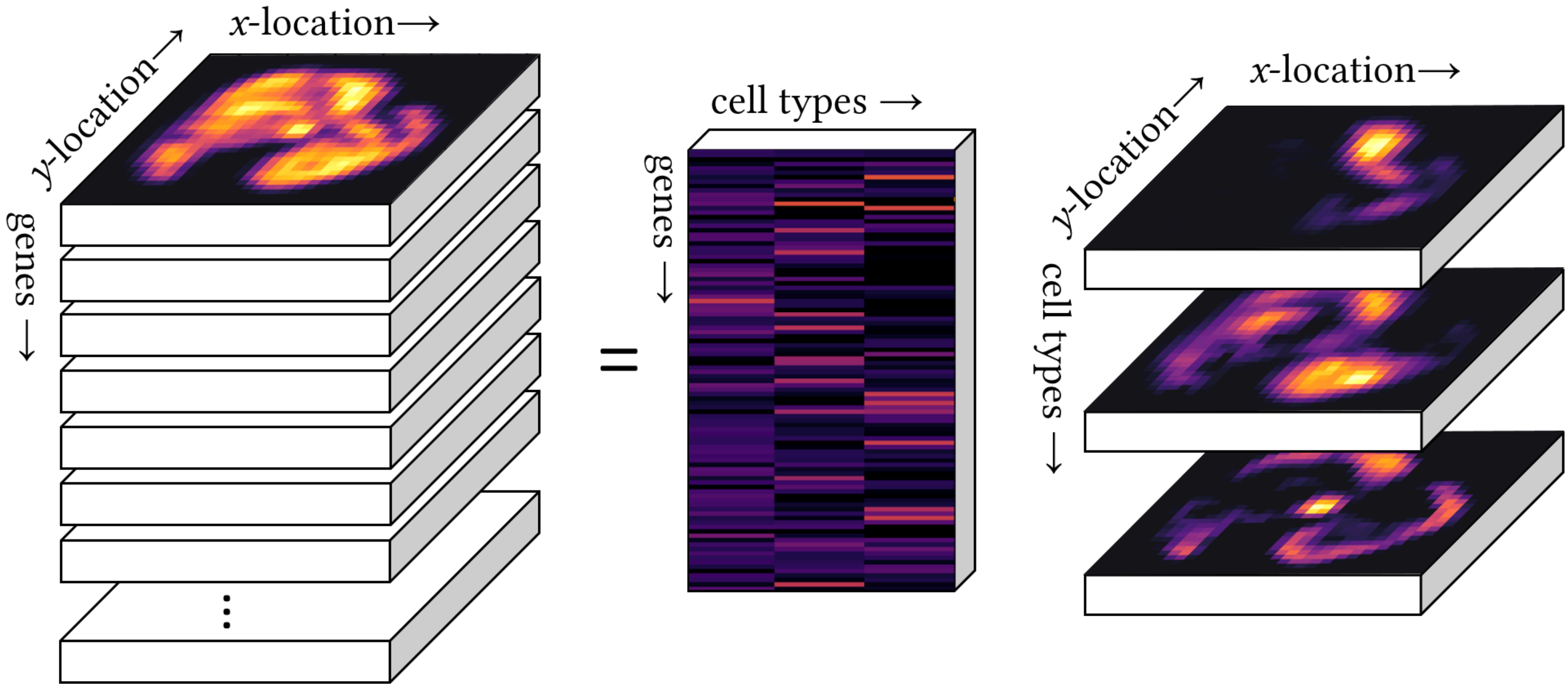


Figure 3: Spatial transcriptomics factorization model. Spatial distribution of many genes can be decomposed into few cell types. We uncover the gene expression and spatial distribution of these cell types, and can label distinct regions accordingly.

Application: Spatial Transcriptomics

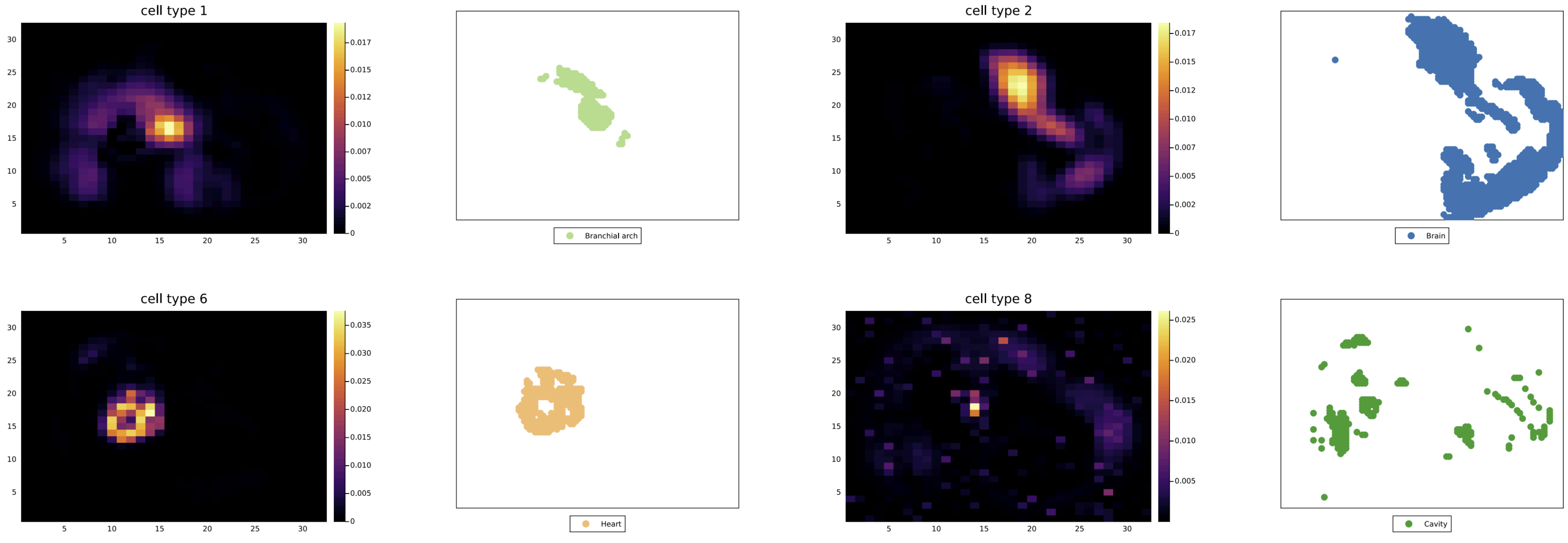


Figure 4: Learn cell types and their spatial presence (heatmaps) matched with known cell types (Branchial Arch, Brain, Heart, and Cavity).

Application: Sediment Analysis

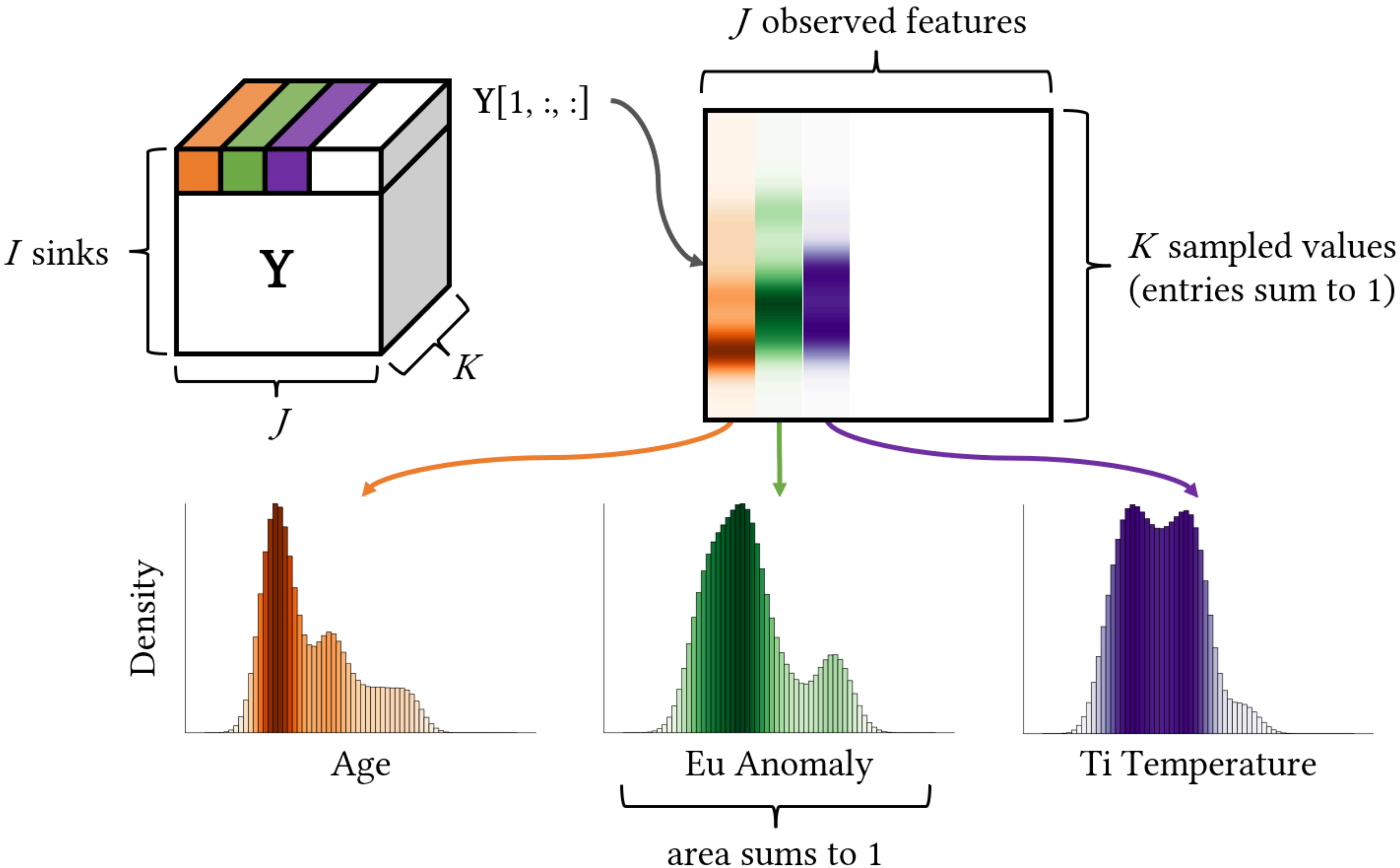


Figure 5: Input data tensor $\mathbf{Y}$. Each depth fibre $\mathbf{Y}[i, j, :]$ is a discretized probability density for a different geological feature [Graham *et al.* 2025].

Application: Musical Instrument Separation

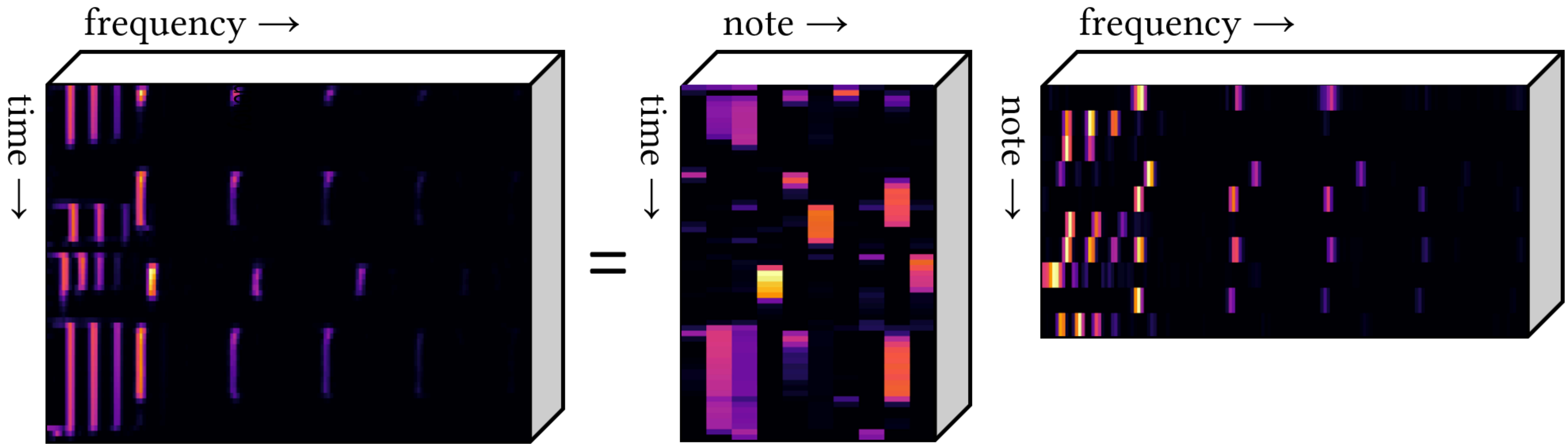


Figure 6: Audio source separation model. The short-time Fourier transform of a mixture can be separated into harmonically distinct notes. These can be grouped by their spectral similarity to recover instrument sources.

Computing Factorizations:

Constrained Optimization

Minimize the error between the model $\mathbf{B} \times_1 \mathbf{A}$ and the data $\mathbf{Y}$:

$$\min_{\mathbf{A},\mathbf{B}} \frac{1}{2} \|\mathbf{B} \times_1 \mathbf{A} - \mathbf{Y}\|_F^2 \quad \text{s.t} \quad \mathbf{A} \in \mathcal{C}_A, \ \mathbf{B} \in \mathcal{C}_B.$$

Implementation in Julia [Richardson *et al.* 2025]:

```julia
using BlockTensorFactorization
Y = load_data()

options = (rank=3, model=Tucker1, objective=L2(), constraints=nonnegative!)
decomposition, stats, kwargs = factorize(Y; options...)

(B, A) = factors(decomposition)
```

Algorithm:

Block Projected Gradient Descent

Cyclically update factors with descent updates (Xu *et al.* 2023)

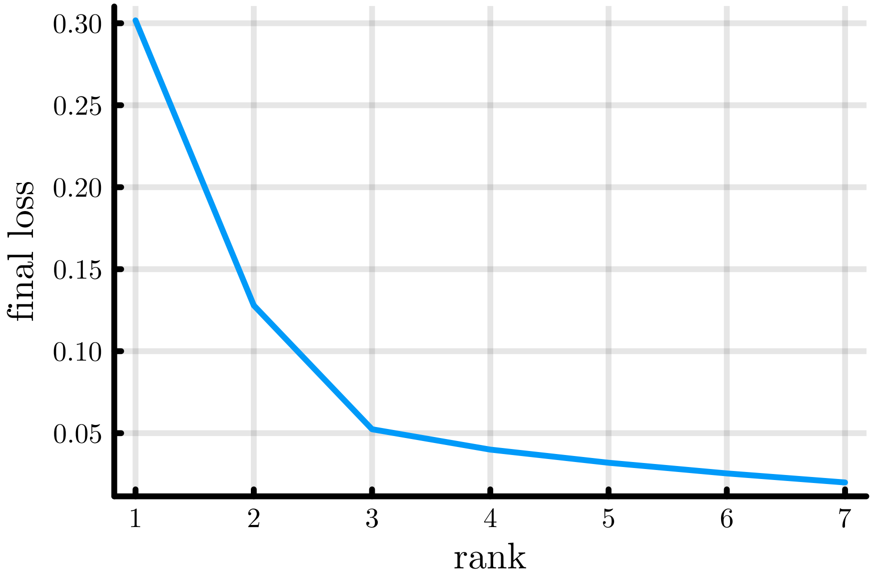
$$\mathbf{A}^{t+1} = P_{\mathcal{C}_A}\left(\mathbf{A}^t - \alpha \nabla_A \ell(\mathbf{A}^t, \mathbf{B}^t)\right).$$

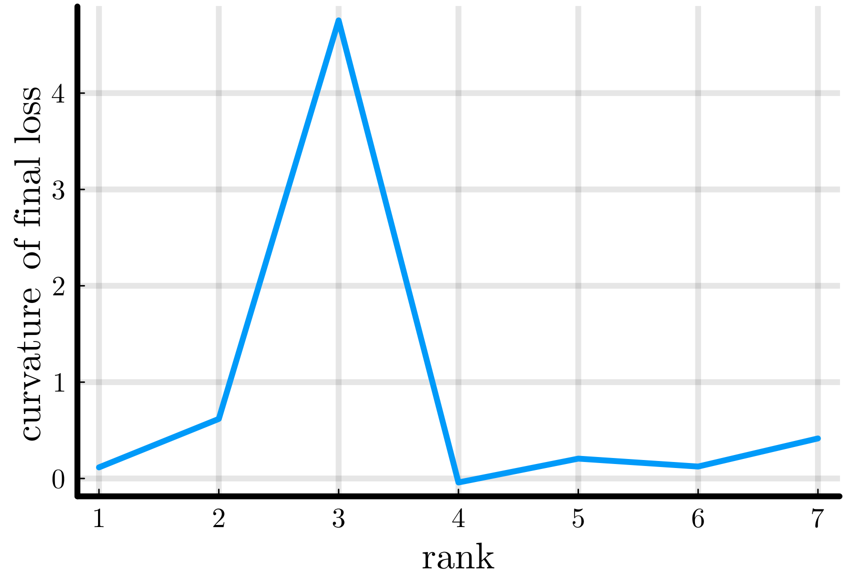
Converges to a *block-wise* minimum and stationary point:

$$\ell(\mathbf{A}^*, \mathbf{B}^*) \leq \min_{\mathbf{A}\in\mathcal{C}_A, \mathbf{B}\in\mathcal{C}_B} \{\ell(\mathbf{A}^*, \mathbf{B}), \ell(\mathbf{A}, \mathbf{B}^*)\}.$$

Estimating Rank

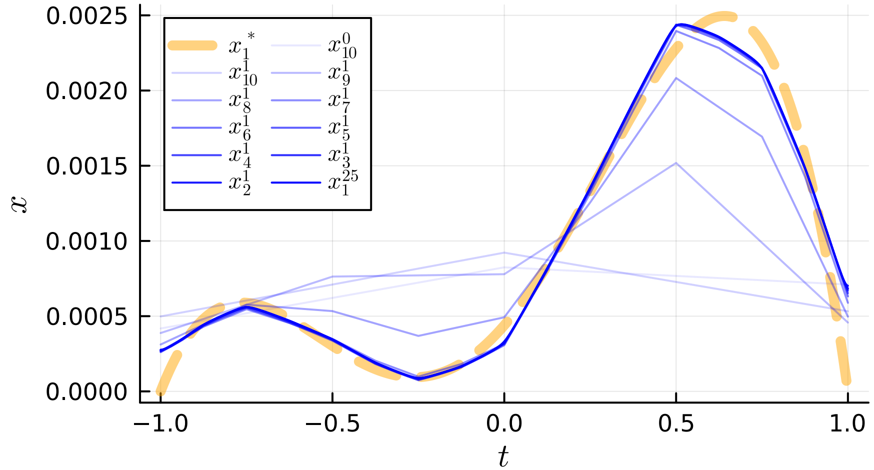
- Usually R needs to be known in advance, but can be estimated
- Let $f(r) = \|\mathbf{X}_r^* - \mathbf{Y}\|_F / \|\mathbf{Y}\|_F$ be final relative error with a rank r factorization X_r^*
- Pick the r at the maximum curvature (Satopaa *et al.* 2011)





Multi-Scaled Decomposition

Optimize over multiple scales for faster convergence.



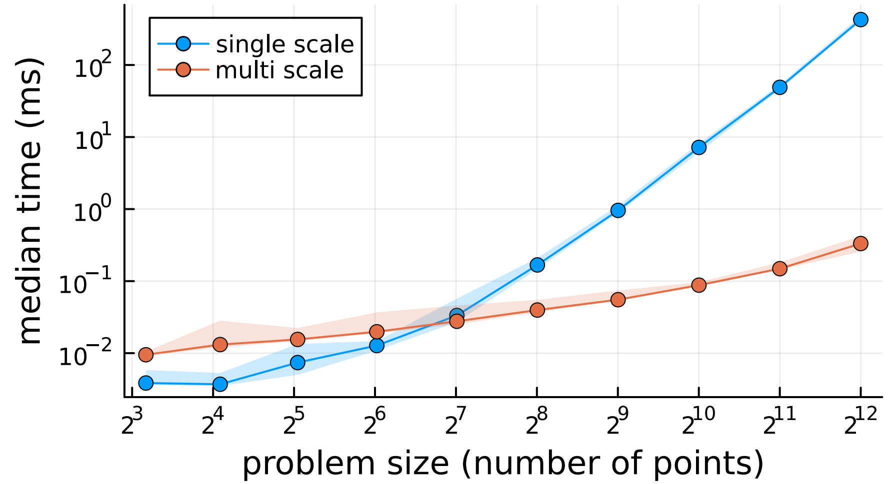


Figure 7: (Left) Optimize over cheaper, coarse discretization with fewer points and refine. (Right) Performance scales better with problem size [Richardson *et al.* 2025].

Contact & References

Website: https://njericha.github.io
Email: njericha@math.ubc.ca

[1] N. Graham, N. Richardson, M. P. Friedlander, and J. Saylor, "Tracing Sedimentary Origins in Multivariate Geochronology via Constrained Tensor Factorization," *Mathematical Geosciences*, Feb. 2025.

[2] N. Richardson, N. Marusenko, and M. P. Friedlander, "BlockTensorFactorization.jl," *GitHub*. https://github.com/MPF-Optimization-Laboratory/BlockTensorFactorization.jl; GitHub, 2025.

[3] N. J. E. Richardson, N. Marusenko, and M. P. Friedlander, "Multiple Scale Methods For Optimization Of Discretized Continuous Functions." arXiv, Dec. 2025.