Scalable Methods for Nonnegative Matrix Factorizations of Near-separable Tall-and-skinny Matrices

Stanford University
PURDUE UNIVERSITY

Austin Benson, Jason Lee, Bartek Rajwa, David Gleich
{arbenson, jdl17}@stanford.edu
{brajwa, dgleich}@purdue.edu

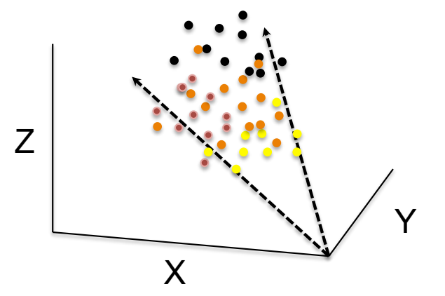
Introduction to (near-separable) NMF

- **NMF Problem**: $X \in \mathbb{R}_+^{m \times n}$ is a matrix with nonnegative entries, and we want to compute a *nonnegative matrix factorization* (NMF) $X = WH$, where $W \in \mathbb{R}_+^{m \times r}$ and $H \in \mathbb{R}_+^{r \times n}$. When $r < m$, this problem is NP-hard.
- A *separable* matrix is one that admits a nonnegative factorization where $W = X(:, \mathcal{K})$, *i.e.* W is just consists of some subset of the columns of X. A *near-separable* matrix is one where $X = X(:, \mathcal{K})H + N$, where N represents noise. The set $\mathcal{K}$ of columns are called *extreme* columns.
- Under the near-separable assumptions, there are efficient algorithms for computing the NMF. The algorithms typically proceed as follows:

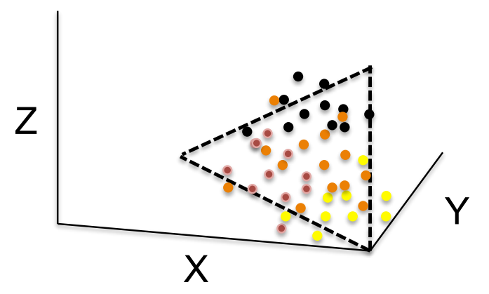
1. Determine the extreme columns, indexed by $\mathcal{K}$, and let $W = X(:, \mathcal{K})$.
2. With W fixed, solve $H = \arg\min_{Y \in \mathbb{R}_+^{r \times n}} \|X - WY\|$.

Our problem: Compute separable NMF when $m \gg n$.

Convex geometry behind NMF algorithms



- **Extreme rays of a cone**: In separable NMF, $X = X(:, \mathcal{K})H$ implies that all columns of X lie in the cone generated by the columns indexed by $\mathcal{K}$. For any $k \in \mathcal{K}$, $\{\alpha X(:, k) \mid \alpha \in \mathbb{R}_+\}$ is an *extreme ray* of this cone. Computing $\mathcal{K}$ is reduced to finding the extreme rays of a cone [1].



- **Extreme points of a convex hull**: If $D_{ii} = \|X(:, i)\|_1$ and X is separable, then $XD^{-1} = (XD^{-1})(:, \mathcal{K})\tilde{H}$. The columns of $\tilde{H}$ have non-negative entries and sum to one, so all columns of XD^{-1} are in the convex hull of the columns indexed by $\mathcal{K}$. Determining $\mathcal{K}$ is reduced to finding the extreme points of a convex hull [2, 3].

Dimension reduction with an orthogonal transformation

Fact: A vector x generates an extreme ray of a cone $\mathcal{C}$ if and only if Mx generates an extreme ray of $M\mathcal{C} = \{Mz \mid z \in \mathcal{C}\}$, where M is nonsingular. Similarly, for any convex set, invertible transformations preserve extreme points.

Our approach: Let $X = U\tilde{\Sigma}V^T$ be the SVD of X, so that U is $m \times m$ orthogonal. Then

$$U^T X = \begin{pmatrix} \Sigma V^T \\ \mathbf{0} \end{pmatrix},$$

where Σ is the top $n \times n$ block of $\tilde{\Sigma}$. The zero rows provide no information about extreme rays or extreme points. Thus, we can restrict ourselves to finding the extreme columns of ΣV^T.

Key idea 1: ΣV^T is $n \times n$, so we have significantly reduced the problem dimension for finding extreme columns of tall-and-skinny matrices ($m \gg n$).

Key idea 2: We can also solve for the coefficient matrix H and compute the residual $\|X - X(:, \mathcal{K})H\|$ by only looking at ΣV^T.

Key idea 3: Since U^T is orthogonal, so it is only a rotation or reflection of the data. Therefore, we have preserved the geometry of the problem.

Key idea 4: We do not need to compute the $m \times m$ matrix U, we just need to apply U^T implicitly.

Implementation

- When the matrix is tall-and-skinny, we only need to read the matrix once!
- Reads can be performed in parallel.
- The key component is the TS-SVD algorithm, which computes ΣV^T without storing the matrix U for tall-and-skinny matrices.
- We use Hadoop MapReduce for convenience: `https://github.com/arbenson/mrnmf`.

After computing ΣV^T, we use standard NMF algorithms—XRAY [1] and SPA [2]—to find the extreme columns. We also compare against Gaussian Projections [4], another dimension reduction technique.

Heat transfer simulation data analysis

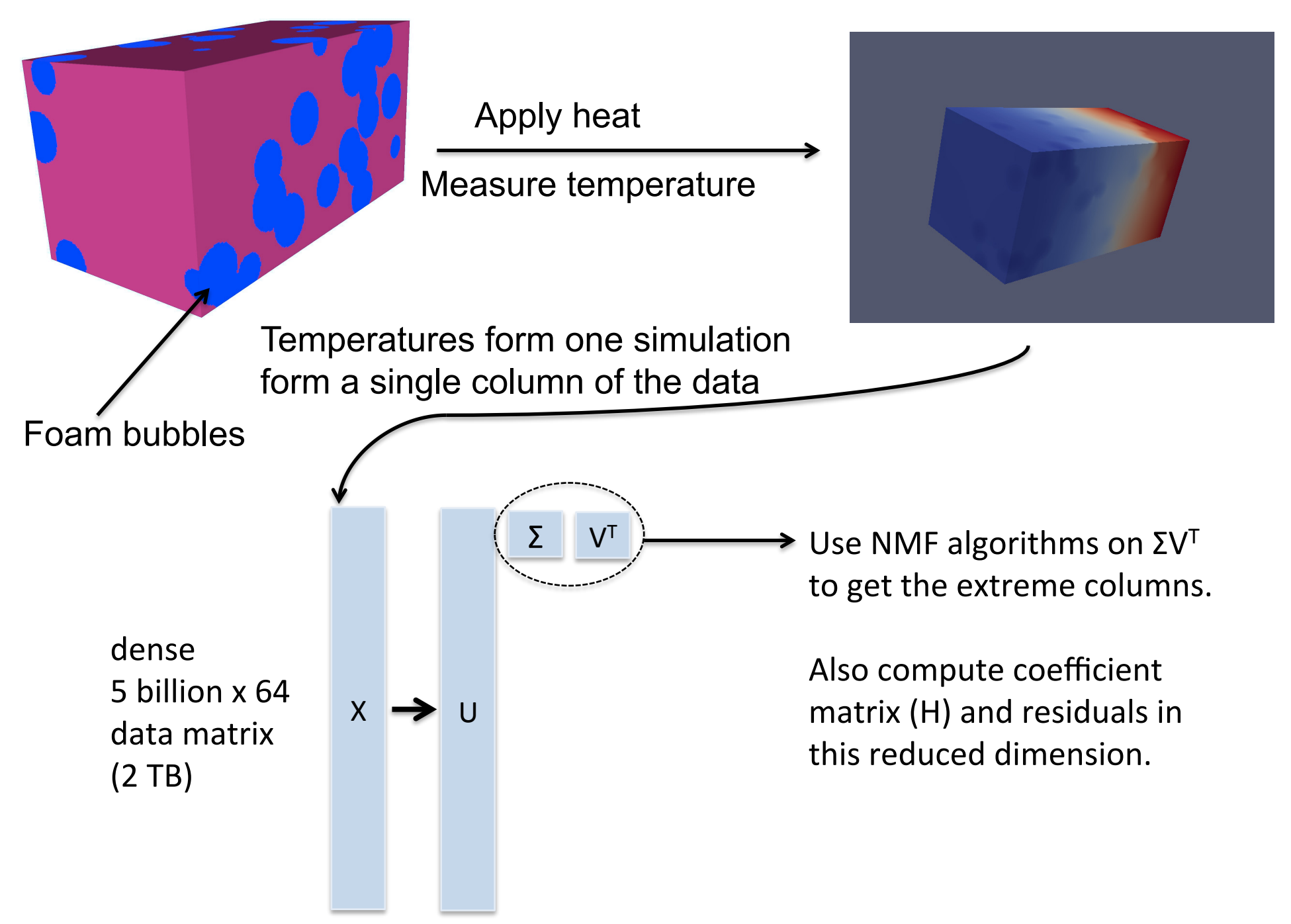

Apply heat
Measure temperature
Foam bubbles
Temperatures form one simulation form a single column of the data
dense
5 billion x 64
data matrix
(2 TB)
X
U
Σ
V^T
Use NMF algorithms on ΣV^T to get the extreme columns.
Also compute coefficient matrix (H) and residuals in this reduced dimension.

Figure 1: Overview of the heat transfer simulation data analysis pipeline. Our work enables us to compute nonnegative matrix factorizations on the massive simulation data in a scalable matter.

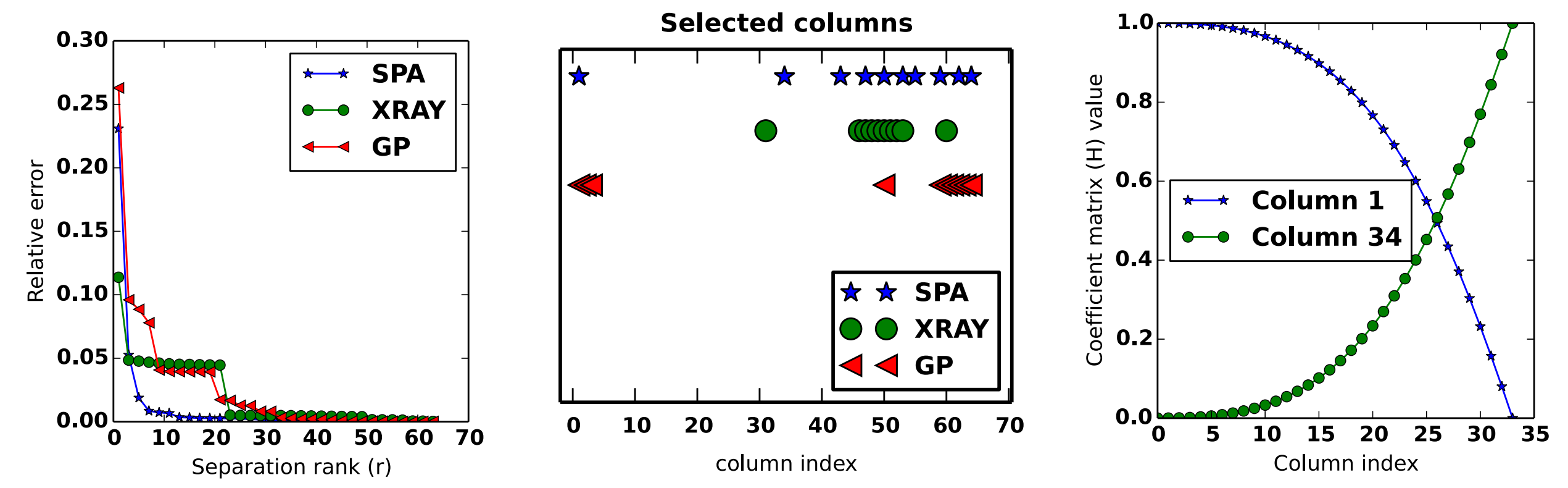


Figure 2: (Left) Relative error in the separable factorization as a function of separation rank (r) for the heat transfer simulation data. Our dimension reduction technique lets us test all values of r quickly. (Middle) The first 10 extreme columns selected by SPA, XRAY, and GP. (Right) Values of $H(\mathcal{K}^{-1}(1), j)$ and $H(\mathcal{K}^{-1}(34), j)$ computed by SPA for $j = 2, \ldots, 33$, where $\mathcal{K}^{-1}(1)$ and $\mathcal{K}^{-1}(34)$ are the indices of the extreme columns 1 and 34 in W ($X = WH$).

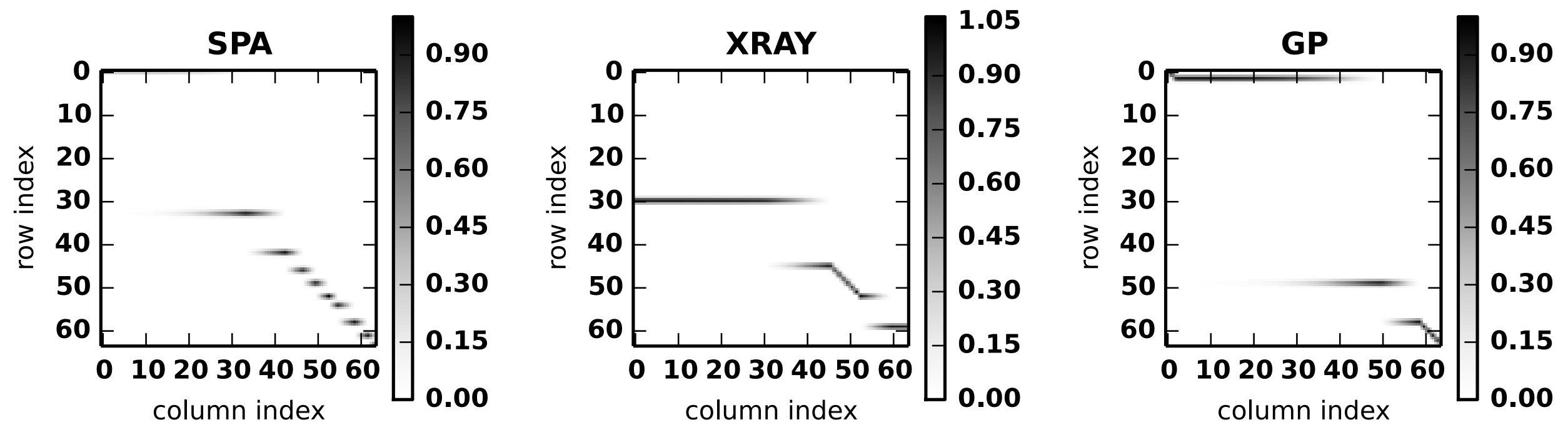


Figure 3: Coefficient matrix H for SPA, XRAY, and GP for the heat transfer simulation data when $r = 10$. In all cases, the non-extreme columns are conic combinations of two of the selected columns, i.e., each column in H has at most two non-zero values.

Computational details

- **Residual**: Given $\mathcal{K}$, how do we compute H in $X \approx X(:, \mathcal{K})H$? Choosing the Frobenius norm error results in a set of n NNLS problems:

$$H(:, i) = \arg\min_{y \in \mathbb{R}_+^r} \|X(:, \mathcal{K})y - X(:, i)\|_2^2 = \|\Sigma V^T(:, \mathcal{K})y - \Sigma V^T(:, i)\|_2^2,$$

as the 2-norm is unitarily invariant ($X = QR$). Thus, we can solve the NNLS problem with matrices of size $n \times n$. This is a major advantage because a challenge with NMF is finding the correct size of $|\mathcal{K}|$.

- **Column normalization**: Some algorithms require column normalization of X. If D is the diagonal matrix of column norms, then

$$X = QR \rightarrow XD^{-1} = Q(RD^{-1}).$$

The matrix $\hat{R} = RD^{-1}$ is upper triangular, so $Q\hat{R}$ is the thin QR factorization of the column-normalized data. With $X = QR$ and $R = U_R \Sigma V^T$, we have the decomposition $X = (QU_R)\Sigma V^T$. Therefore, we simultaneously compute D and ΣV^T in just one pass over the data.

Flow cytometry data analysis

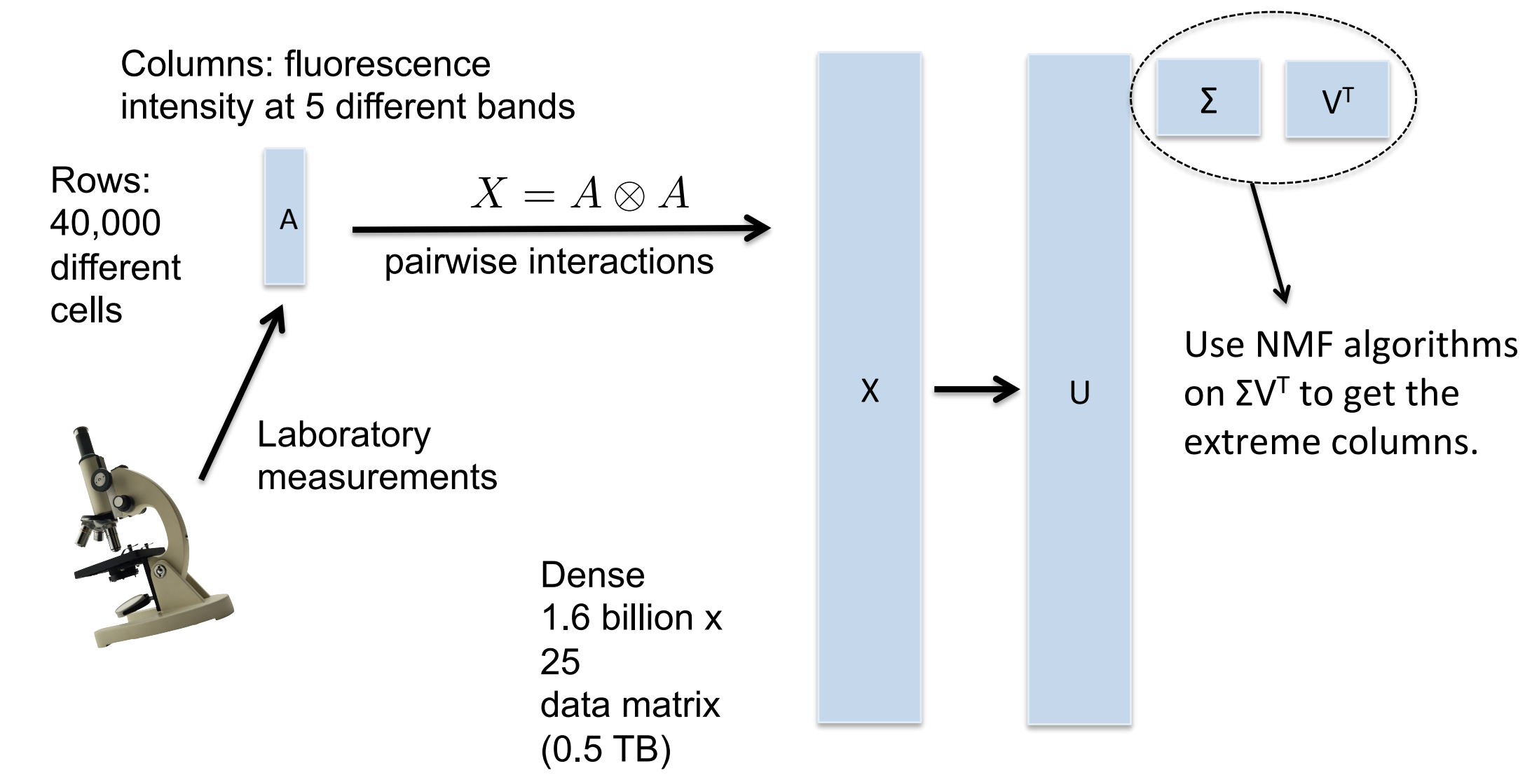

Columns: fluorescence intensity at 5 different bands
Rows: 40,000 different cells
A
$X = A \otimes A$
pairwise interactions
Laboratory measurements
Dense
1.6 billion x
25
data matrix
(0.5 TB)
X
U
Σ
V^T
Use NMF algorithms on ΣV^T to get the extreme columns.

Figure 4: Overview of the flow cytometry data analysis pipeline.

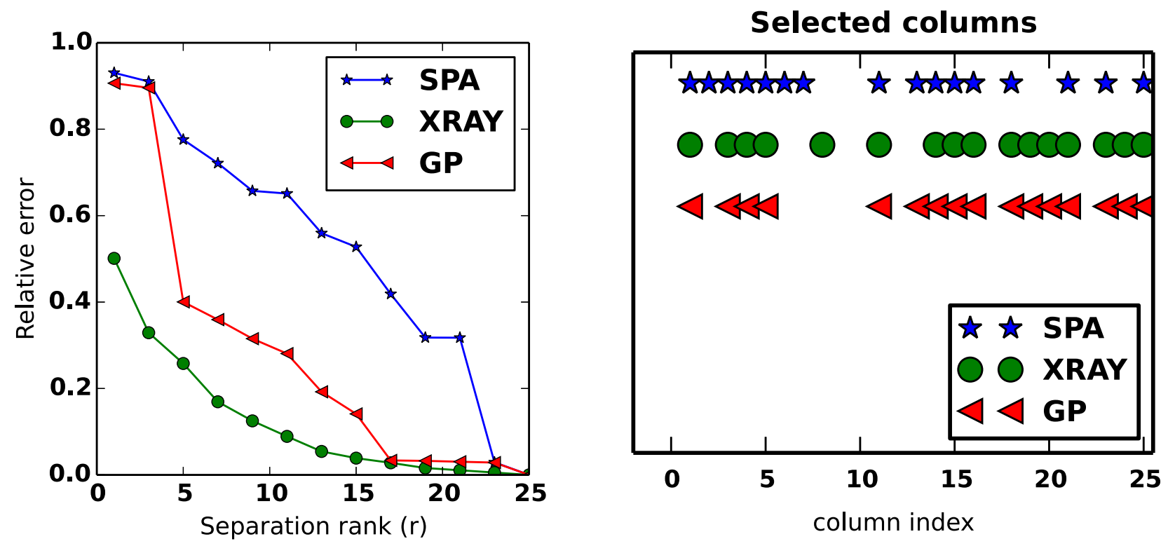


Figure 5: (Left) Relative error in the separable factorization as a function of nonnegative rank (r) for the flow cytometry data. (Right) The first 16 extreme columns selected by SPA, XRAY, and GP.

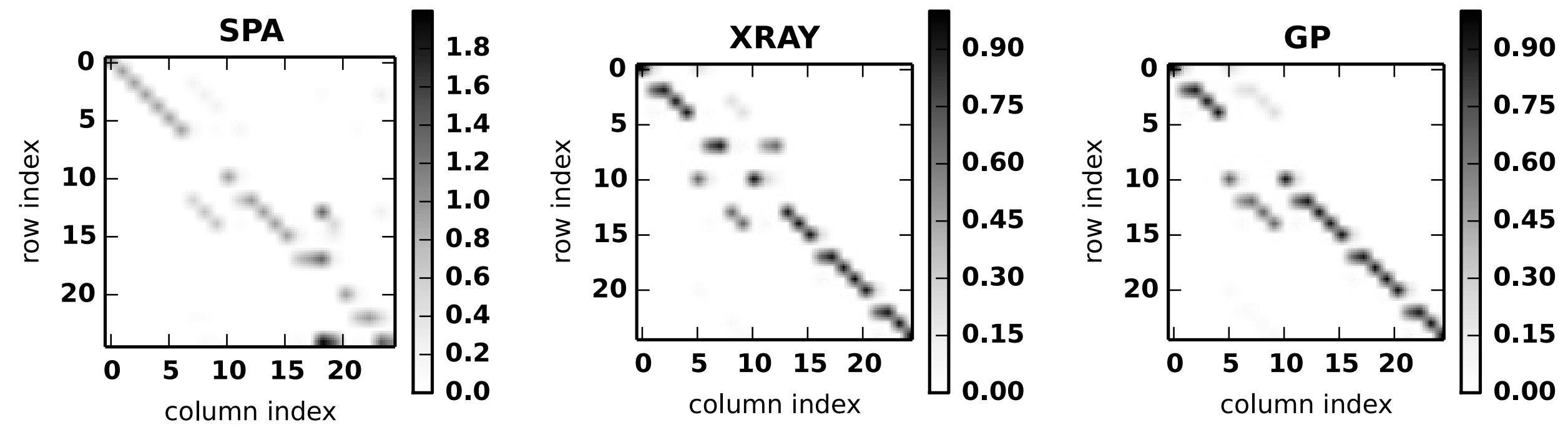


Figure 6: Coefficient matrix H for SPA, XRAY, and GP for the flow cytometry data when $r = 16$. The coefficients tend to be clustered near the diagonal. This is quite different from the coefficients for the heat transfer simulation data.

References

[1] A. Kumar *et al*. Fast conical hull algorithms for near-separable non-negative matrix factorization. In *ICML*, 2013.

[2] N. Gillis and S. Vavasis. Fast and robust recursive algorithms for separable nonnegative matrix factorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PP(99):1–1, 2013.

[3] V. Bittorf, B. Recht, C. Re, and J. A. Tropp. Factoring nonnegative matrices with linear programs. In *NIPS*, 2012.

[4] A. Damle and Y. Sun. Random projections for non-negative matrix factorization. *arXiv:1405.4275*, 2014.