疎行列の非負値行列因子分解のための効率的な近似推定法

阿部興 [1] (発表者) ・島村徹平 [1]

2024 年 10 月 25 日

[1] 東京科学大学 総合研究院 難治疾患研究所

動機: 分析対象

Bischoff et al. (2021) [2] の肺がんに関する単一細胞 RNA 発現量データ

- 行（遺伝子）: 33,514
- 列（細胞）: 120,961
- 非ゼロ要素: 239,634,370 (全体の 5%程度)

単一細胞解析の分野では, この研究のデータが特別に大きいわけではない

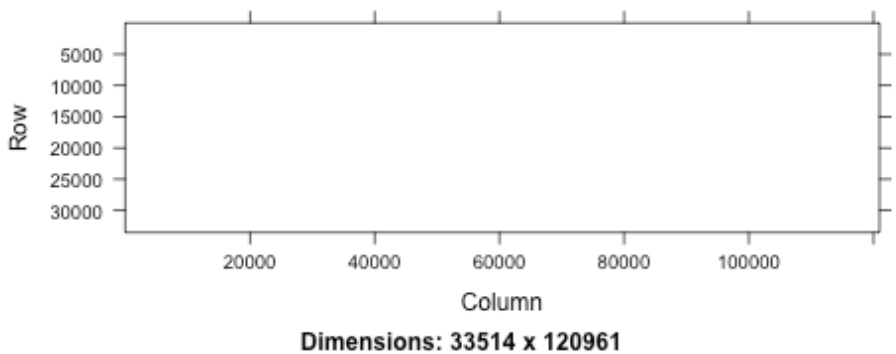
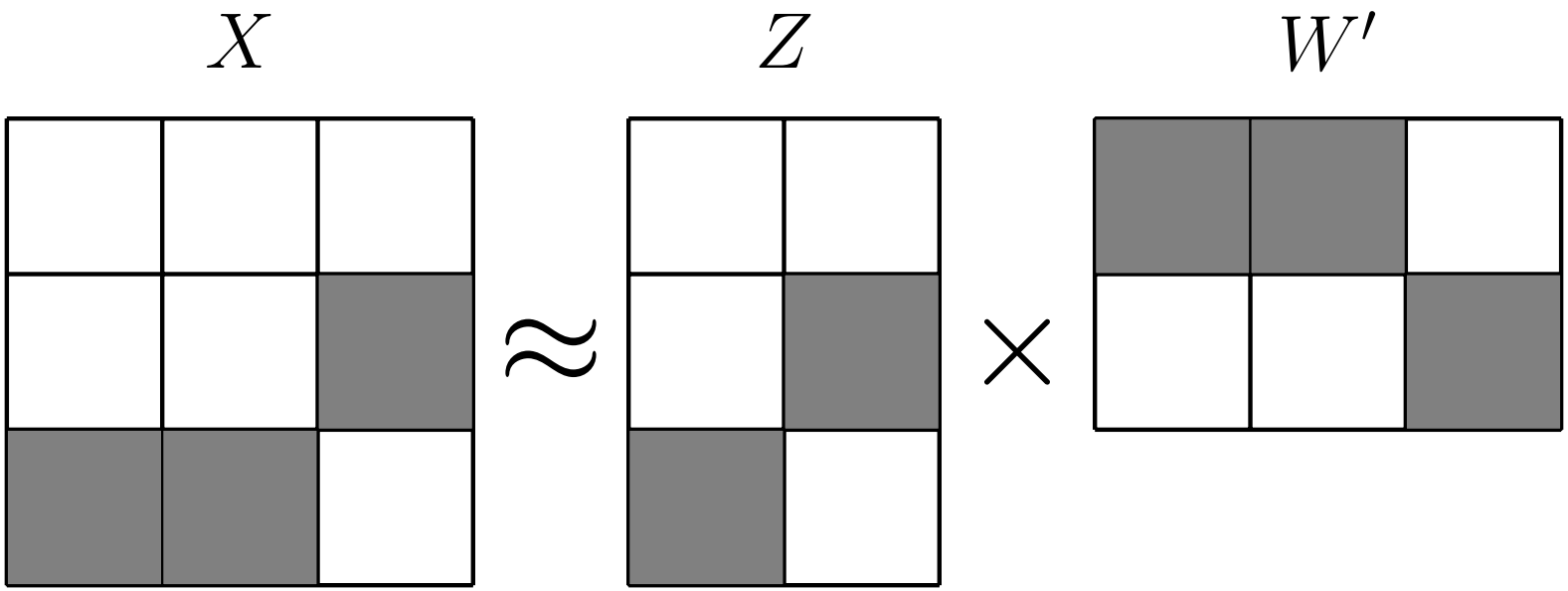


Fig: ゼロ要素を白, 非ゼロ要素を黒としたヒートマップ. 黒はほとんど見えない.

[2] Bischoff, P., Trinks, A., Obermayer, B. et al. Single-cell RNA sequencing reveals distinct tumor microenvironmental patterns in lung adenocarcinoma. Oncogene 40, 6748 – 6758 (2021). https://doi.org/10.1038/s41388-021-02054-3

非負値行列因子分解（NMF）

- 得られたデータを低次元に射影して圧縮することでパターンを抽出
- 非負制約により解釈がしやすい



$$P(I=i,J=j)=\sum_l \underbrace{P(I=i|L=l)}_{\propto Z}\underbrace{P(J=j|L=l)}_{\propto W}P(L=l)$$

疎行列と NMF

- 観測のゼロ過剰や過分散をモデル化したケース [3]
- 分解で得られる行列を疎にしようとした議論 [4]

本研究: 疎であることを積極的に利用して計算効率を高める

---

[3] e.g. Abe, H., & Yadohisa, H. (2017). A non-negative matrix factorization model based on the zero-inflated Tweedie distribution. Computational Statistics, 32(2), 475-499.
Gouvert, O., Oberlin, T., & Févotte, C. (2020). Negative binomial matrix factorization. IEEE Signal Processing Letters, 27, 815-819.

[4] e.g. Hyunsoo, K. & Haesun, P. (2007). Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis, Bioinformatics, 23 (12), 1495–1502.
Kim, H., & Park, H. (2007). Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. Bioinformatics, 23(12), 1495-1502.

疎行列の形式

(i,j) 成分の値 x_{ij} を $\mathcal{D}_n = (i, j, x_{ij}) = (r_n, c_n, x_n)$ で表す.
$x_{ij} = 0$ となる (i,j) は省略し,$\mathcal{D} = (\mathcal{D}_1, \ldots, \mathcal{D}_{N_1})$ とする.

Tab: A と B は同じ情報を持つ

Tab: A

1	0	2
0	0	2
4	1	0

Tab: B

r_n	c_n	x_n
1	1	1
3	1	4
3	2	1
1	3	2
2	3	2

十分統計量再考

指数型分布族:

$$p(x|\theta) = \exp(T(x)'\eta(\theta))$$

$T(x) = (T_1(x), T_0)'$, $\eta(\theta) = (\eta_1(\theta), \eta_0(\theta))'$ とする.
$T_1(x)$: x に依存する. T_0: x に依存しない.
$T_1(0) = 0$ ならば,

$$\sum_i \log p(x_i|\theta)$$

$$= \left(\sum_{i \in \text{nonzero part}} T_1(x)'\eta_1(\theta_i) \right) + \left(\sum_{j \in \text{all of the data}} T_0\eta_0(\theta_j) \right).$$

ゼロ要素にアクセスせずに対数尤度の評価ができる

ベイズ更新

共役事前分布:

$$\phi(\theta|\xi_1,\xi_0) = \frac{1}{Z(\xi_1,\xi_0)} \exp(\eta_1(\theta)\xi_1 + \eta_0(\theta)\xi_0)$$

事後分布:

$$\begin{aligned}\phi^*(\theta|\xi_1,\xi_0,x) \ &(\propto \phi(\theta|\xi_1,\xi_0)p(x|\theta)) \\ &= \phi(\theta|\xi_1 + T_1(x), \xi_0 + T_0).\end{aligned}$$

例: ポアソン分布

$$\begin{aligned}T_1(x) = (x, \log(x!))', \ T_0 = (1), \\ \eta_1(\theta) = (\log(\theta), -1)', \ \eta_0(\theta) = (-\theta).\end{aligned}$$

共役事前分布 (ガンマ分布):

$$\log(\mathrm{Gamma}(\theta|a,b)) = (\xi_1 - 1)\log(\theta) - \xi_0\theta + \log\left(\underbrace{\xi_0^{\xi_1}/\Gamma(\xi_1)}_{Z(\xi_1,\xi_0)}\right).$$

モデル

ポアソン分布の行列因子分解のための確率モデル[5]:

$$x_{ij} = \sum_{l=1} u_{ijl}, \quad u_{ijl} \sim \mathrm{Pois}\left(z_{il} w_{jl}\right)$$

事前分布:

$$z_{il} \sim \mathrm{Gamma}(a, b), \quad w_{jl} \sim \mathrm{Gamma}(a, b).$$

Note: x_{ij} は次と同値

$$x_{ij} \mid z, w \sim \mathrm{Pois}\left(\sum_{l} z_{il} w_{lj}\right).$$

[5]Cemgil, A. T.(2009). Bayesian Inference for Nonnegative Matrix Factorization Models, Computational Intelligence and Neuroscience, 785152, 17 https://doi.org/10.1155/2009/785152

モデルの対数尤度

$$\ell(Z, W) = \left\{ \sum_{n=1}^{N_1} u_{nl} \log(z_{r_n l} \cdot w_{c_n l}) - \log(u_{nl}!) \right\} + \left\{ \sum_{i=1}^{R} \sum_{j=1}^{M} \{ - z_{il} \cdot w_{jl} \} \right\}.$$

第 1 項: ゼロ要素にアクセスしていないことに注意
第 2 項: Z, W を所与としたときデータの値に依存しない

参考: ガンマ分布

$$\log(\mathrm{Gamma}(z|a, b)) = \underline{(a-1)\log(z)} - \underline{bz} + \log\left(b^a/\Gamma(a)\right).$$

擬似コード: 形状パラメータの更新

1: Function updateA($\mathcal{D}$, Z, W, a)
2: $\alpha^z_{il} \leftarrow a$; $\alpha^w_{jl} \leftarrow a$ $(i = 1, \ldots, R,\ j = 1, \ldots, C)$
3: **for** $n \in \{1, \ldots, N_1\}$ **do**
4: **for** $l \in \{1, \ldots, L\}$ **do**
5: $$U_{nl} \leftarrow \frac{x_n \exp(E_q[\log z_{r_n l}] + E_q[\log w_{c_n l}])}{\sum_{l=1}^{L} \exp(E_q[\log z_{r_n l}] + E_q[\log w_{c_n l}])}$$
6: $\alpha^z_{r_n l} \leftarrow \alpha^z_{r_n l} + U_{nl}$
7: $\alpha^w_{c_n l} \leftarrow \alpha^w_{c_n l} + U_{nl}$
8: **end for**
9: **end for**
10: Return α^z_{il}, α^w_{jl} $(i = 1, \ldots, R,\ j = 1, \ldots, C)$
11: EndFunction

3 行目: ゼロ要素にアクセスしていないことに注意
5 行目: 和で条件付けたポアソン分布は多項分布

擬似コード: レートパラメータの更新

1: Function updateB(Z, W, b)
2: **for** $l \in 1, \ldots, L$ **do**
3: $\quad \beta_l^z \leftarrow b + (\sum_j E_q[w_{jl}])$
4: $\quad \beta_l^w \leftarrow b + (\sum_i E_q[z_{il}])$
5: **end for**
6: Return β_l^z, β_l^w $(l = 1, \ldots, L)$
7: EndFunction

1 行目: $\mathcal{D}$ に依存しないことに注意

- `updateA` と `updateB` を収束するまで繰り返す

数値例: 設定

データ: 非ゼロ要素については「ポアソン乱数 $+1$」としてランダム行列を作成

- 非ゼロ要素の割合: 0.1, 0.2, 0.3, 0.4, 0.5
- 行 R: 100, 1000, 5000
- 列 C : 2000
- 分解のランク L: 2, 5, 10
- アルゴリズムのイテレーション: 100 回（固定）

各 10 回繰り返した

計算量は,

- 通常の NMF のアルゴリズム [6]: $R \cdot C \cdot L$ に比例
- 提案法: $R \cdot C \cdot (\text{非ゼロ要素の割合}) \cdot L$ に比例

[6] Gaujoux, R., Seoighe, C. A (2010). A flexible R package for nonnegative matrix factorization. BMC Bioinformatics 11, 367.

数値例: 結果

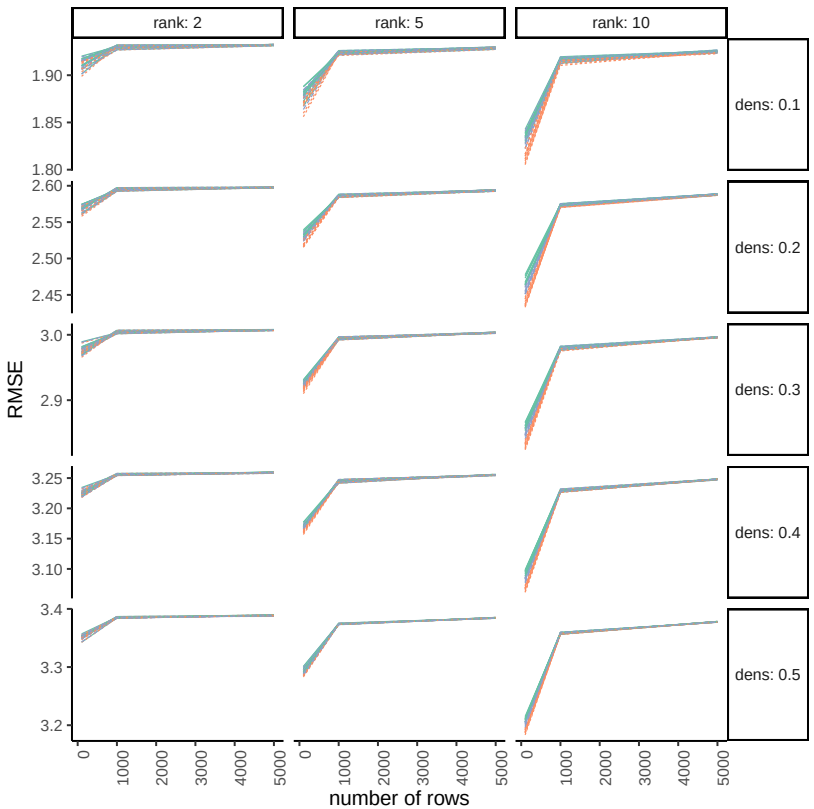


Fig: Root mean squared error: X と推定された ZW の平均 2 乗誤差

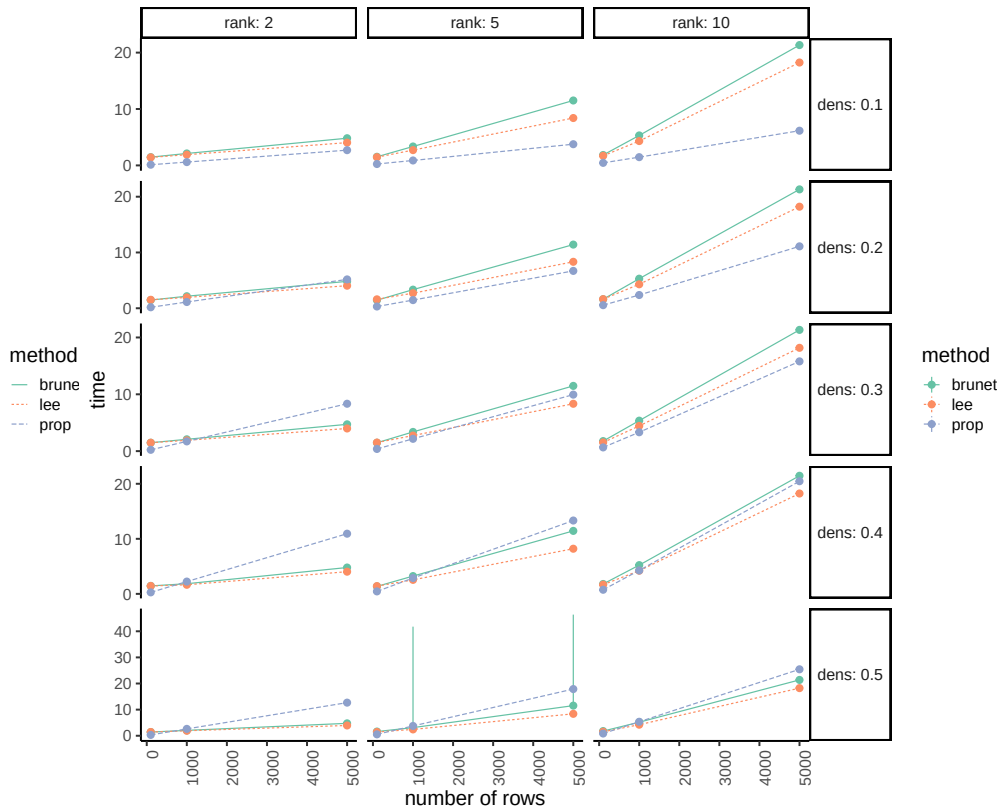


Fig: 計算時間（秒）：点は中央値, エラーバーは 80%区間

確率的変分ベイズ法

方針: 毎回すべてのデータセットを使うのでなく, リサンプルして更新を繰り返す[7]

サンプル 1 つあたりの尤度:

$$\ell_n(Z, W) \approx E_{(i,j)}[\ell_n(Z, W)]$$

(i, j) が独立な離散一様分布とすると,

$$\ell_n(Z, W) \approx \{u_{nl} \log(z_{r_n l} \cdot w_{c_n l}) - \log(u_{nl}!)\} + \frac{1}{RC} \left\{ \left(\sum_{i=1}^{R} z_{il} \right) \cdot \left(\sum_{j=1}^{C} w_{jl} \right) \right\}.$$

- サイズ S のミニバッチを取るとき第 2 項は S 倍

[7] Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. Journal of Machine Learning Research.

擬似コード: 確率的変分ベイズ法

レートパラメータを更新する関数のみ変更:

Function updateB_s(Z, W, b, S_k/RC)
for $l \in 1, \ldots, L$ **do**
$\tilde{\beta}_l^z \leftarrow b + (S_k/RC)(\sum_j E_q[w_{jl}])$
$\tilde{\beta}_l^w \leftarrow b + (S_k/RC)(\sum_i E_q[z_{il}])$
end for
return $\tilde{\beta}_l^z$, $\tilde{\beta}_l^w$ $(l = 1, \ldots, L)$
EndFunction

$\mathcal{D}$ をサイズ S_k のミニバッチ $\mathcal{D}^{(k)}$ に分割 $(k = 1, \ldots, K)$

- $\mathcal{D}^{(k)}$ に対して updateA と updateB_s により $\tilde{\theta}$ を得る
- $\theta \leftarrow (1 - \eta_t)\theta + \eta_t \tilde{\theta}$

変分パラメータ $(\alpha_{il}^z, \beta_l^z, \alpha_{jl}^w, \beta_l^w)$ をまとめて θ と書いた.

学習率: $\eta_t = (N_S/N_1) \cdot (t + \tau)^{-\kappa}$, ($\tau \geq 0$, $\kappa \in [0.5, 1]$)

データ分析: Bischoff et al. (2021)

肺がんに関する単一細胞 RNA 発現量データ（再掲）

- 行（遺伝子）: 33, 514
- 列（細胞）: 120,961
- 非ゼロ要素: 239,634,370 (全体の 5%程度)

```
R> Mat <- Matrix::readMM("path")
R> res <- NMF::nmf(as.matrix(Mat), rank = 2)
```

```
Error in h(simpleError(msg, call)) :
  error in evaluating the argument 'x' in selecting a method for function 'nmf':
vector memory limit of 24.0 Gb reached, see mem.maxVSize()
In addition: Warning message:
In asMethod(object) :
  sparse->dense coercion: allocating vector of size 30.2 GiB
```

すべてをメモリ上に展開するのは無理がある

細胞（列）の特徴量 W をプロット

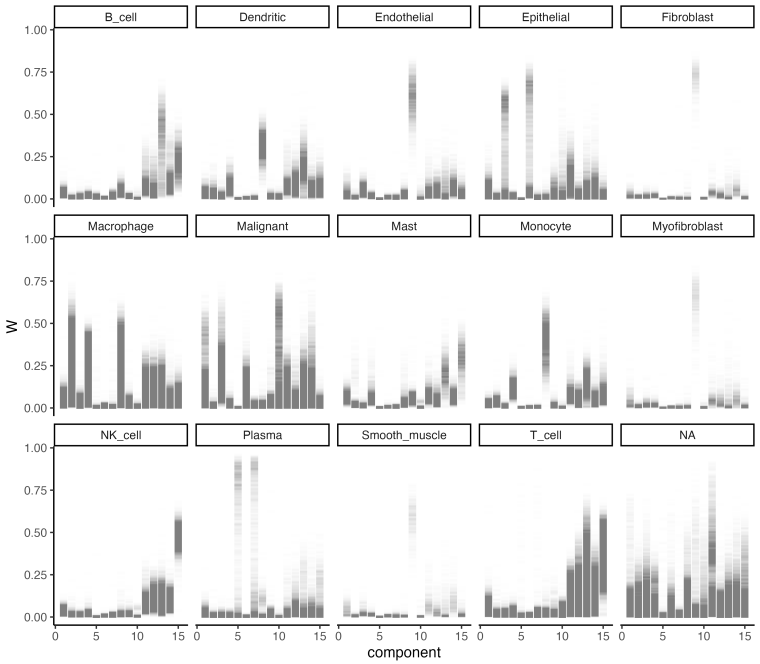


Fig: NMF: ミニバッチサイズ 10^8, $\tau = 1$, $\kappa = 0.9$ とした

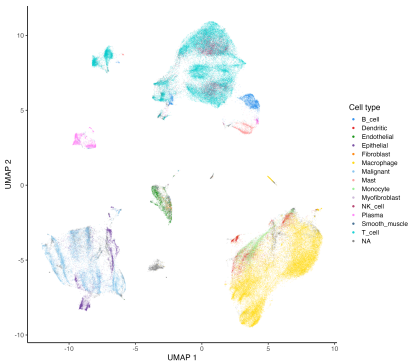


Fig: UMAP: NMF で 15 次元にしたものをさらに 2 次元に.

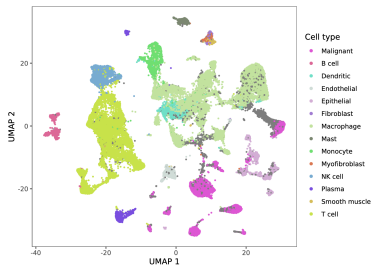


Fig: UMAP: PCA で 15 次元にしたものをさらに 2 次元に. Bischoff et al. (2021) より.

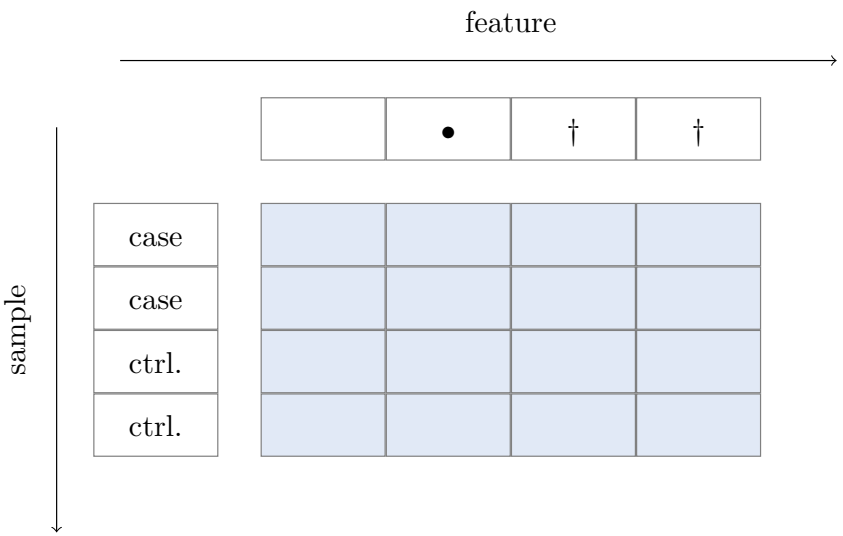
UMAP は大規模なデータに対しても高速な実装があるため単一細胞解析でよく使われるが解釈が難しい [8]

[8] Chari, T., & Pachter, L. (2023). The specious art of single-cell genomics. PLoS Comput Biol. 17;19(8):e1011288. doi: 10.1371/journal.pcbi.1011288.

拡張: Blessing of dimensionality へ向けて

Abe & Shimamura (2023) のモデル[9] を次のように表記すると本報告との関係がわかる:

$$y_{ij} = \sum_{l=1} u_{nl}, \quad u_{nl} \sim \mathrm{Pois}\left(\sum_{l} \prod_{d=1}^{D} v^{(d)}_{x_{nd},l}\right)$$



sample	feature	annotation		
1	1		case	
2	1		case	
⋮	⋮	⋮		
4	1		crl	
1	2	•	case	
⋮	⋮	⋮		
4	2	•	ctrl.	
1	3	†	case	
⋮	⋮	⋮		
4	4	†	ctrl.	

context | y

[9] Abe, K. & Shimamura, T. (2023). UNMF: a unified nonnegative matrix factorization for multi-dimensional omics data, Briefings in Bioinformatics, 24(5), bbad253, https://doi.org/10.1093/bib/bbad253

まとめと議論

- 疎行列に適した非負値行列因子分解のアルゴリズムを提案した
- 疎であるほど計算量やメモリ効率の点で有利
- わずかな変更で応用の可能性（一般化線形モデル, 混合分布, ...）

本報告の実装: `https://github.com/abikoushi/VBsNMF`