SC2: Sparse Coding for Large Scale Single-cell RNA Sequencing Data Analysis

1 Motivations

Single-cell RNA-sequencing (scRNA-seq) technologies enable us measure transcriptomic level of individual cells. In scRNA-seq experiments, scRNA expression data usually comes from different individuals with different phenotypes across different conditions and even species using different technologies [1]. This may lead to severe batch effects and introduction of heterogeneous biological variation in scRNA-seq data, making downstream analysis challenging. Most existing approach are proposed to address technical variations caused by different technologies. For example, statistical approaches, such as Seurat [1] and LIGER [2], were proposed to integrate multiple single cell datasets from different protocols and data modalities. However, few statistical approaches focus on variations from heterogeneous biological variations. scINSIGHT [3] is probably the first one on this issue. Based on nonnegative matrix factorization (NMF), scINSIGHT was proposed to model the variations from heterogeneous biological conditions [3]. However, previous study [4] pointed out that NMF cannot directly infer down-regulations of biological entities in biological data.

We propose a novel statistical approach, SC2, based on additive matrix factorization to model heterogeneous biological variations introduced by different biological variables (e.g., donor, tissue, and disease status) and to decompose cellular variations across single-cell samples into a low-rank latent space simultaneously. SC2 restricts the heterogeneous biological variations in a shared latent space, thus enjoying a better interpretability. This strategy is consistent with our understanding. Specifically, different donors demonstrate heterogeneity in gene expression via some pathways, and different phenotypes of these donors can affect the expression of genes via the same pathways. Meanwhile, we can easily observe that different donors can also be affected differently by the same phenotypes. Thus, to address the concern, SC2 incorporates interactions between different biologically variables.

2 Methods

Here we propose a novel statistical approach, SC2, to decompose cellular variations of scRNA-Seq samples across multiple biological variables (e.g., donor, tissue, and other biological conditions) into low-rank latent spaces to facilitate downstream analysis. In SC2, we integrate matrix factorization, which is utilized to capture variation from the biological variables, and dictionary learning to learn low-rank representations for each cell. In the employment of dictionary learning, we introduce the sparse penalty on latent representations for cells to facilitate cell population discovery and norm constraints on the corresponding gene representations to ensure an equal scaling for each latent components.

2.1 Model specifications

For illustration, let $Z^{N\times M}$ denote the matrix of log-normalized scRNA-Seq expression levels of N samples of M genes. The N samples originate from several biological conditions (e.g., individuals, phenotypes, tissues or disease phases). Here we use donor and phenotype as examples for demonstration. These samples come from N_1 donors with N_2 phenotypes. Thus, the expression level of gene m from sample i, which is obtained from donor j with phenotype t, z_{im}, can be modelled as

$$\hat{z}_{im} \approx d_j^T v_m + p_t^T v_m + s_i^T g_m \tag{1}$$

where d_j, p_t, v_m are vectors of length K_1, s_i, g_m are vectors of length K_2. The donor and phenotype information of sample i is known. In the above equation, the first two terms in the right capture variation from donor and phenotype, which is restricted in a shared low-rank latent space. The last term in the right seeks to decompose scRNA-Seq samples into a different low-rank latent space after controlling variation from donor and phenotype. In SC2, the biological variables depend on needs in applications.

The objective function for Equation 1 is formulated as

$$\begin{aligned}
\mathcal{L}(d,p,v,s,g) =& \frac{1}{2}\sum_{i,m}\left(z_{im} - d_j v_m^T - p_t v_m^T - s_i^T g_m\right)^2 + \\
& \frac{1}{2}\lambda_1\Big(\sum_j ||d_j||_2^2 + \sum_t ||p_t||_2^2 + \sum_m ||v_m||_2^2\Big) + \\
& \lambda_2\Big(\frac{1}{2}(1-\alpha)\sum_i ||s_i||_2^2 + \alpha\sum_i ||s_i||_1\Big) \\
\text{subject to} \quad & \sum_m g_{mk}^2 \le c, \forall k = 1,\ldots,K_2
\end{aligned} \tag{2}$$

In the above equation, g_{mk} is the k-th element of g_m, and c is a constant (usually 1). We introduce the elastic net penalty on the cell representation s_i to encourage sparsity to facilitate cell clustering. Moreover, the norm constrain is introduced to ensure the same scale for each component in decomposing cellular variation.

The Equation 2 can be represented with matrix operation. Let $D^{N_1 \times K_1}, P^{N_2 \times K_1}, V^{K_1 \times M}$ are matrices of latent representations of N_1 donors, N_2 phenotypes, and M genes, respectively, and denote $S^{N \times K_2}, G^{K_2 \times M}$ latent representations for N cells and M genes after controlling variation from donor and phenotype. The matrix representation of Equation 2 can be written as follows:

$$\begin{aligned} \mathcal{L}(D, P, V, S, G) = & \frac{1}{2}\big\|Z - (X_D D + X_P P)V - SG\big\|_F^2 + \\ & \frac{1}{2}\lambda_1(\|D\|_F^2 + \|P\|_F^2 + \|V\|_F^2) + \\ & \lambda_2\big[\frac{1}{2}(1-\alpha)\|S\|_F^2 + \alpha\|S\|_1\big], \\ \text{subject to} \quad & \|G_k\|_2^2 \leq c, \forall k = 1, \ldots, K_2, \end{aligned} \tag{3}$$

where $X_D^{N \times N_1}, X_P^{N \times N_2}$ are indicator matrices, which represent the dummy variables for the samples, G_k is the k-th row of matrix G, and c is a constant, which restricts the scaling of each component of G.

2.2 Model fitting

Alternating block coordinate descent (BCD) was employed to optimize Equation 2. Practically, each time we update a set of independent parameter with all other parameters fixed. For example, when all parameters except v_m are fixed, then our problem becomes a number of linear regression problems with ridge regularization and all v_m can be updated in parallel. In each iteration of BCD, we update each set of parameters of our model sequentially and repeat the process until the stopping criteria meets.

Optimize with the whole data

With the objective function defined by Equation 2 and notations defined in Equation 3, we can easily derive a closed form for updating d_j, p_t, v_m in optimization. First, we have the following update for v_m

$$v_m = \big[W^T W + \lambda_1 \mathbb{I}_{K_1}\big]^{-1} W^T \tilde{Z}_m, \tag{4}$$

where $W = X_D D + X_P P$, $\tilde{Z} = Z - SG$, and $\tilde{Z}_m$ is the m-th column of $\tilde{Z}$. Similarly, the update for d_j is

$$d_j = \left[N_j \sum_m v_m v_m^T + \lambda \mathbb{I}_{K_1}\right]^{-1} \sum_{i \in B_j} \sum_m \hat{z}_{im} v_m, \tag{5}$$

where $\tilde{Z} = Z - SG - X_P PV$, B_j is the set of indices of samples from donor j, and N_j is the number of elements in B_j Likewise, the update for p_t is

$$p_t = \left[N_t \sum_m v_m v_m^T + \lambda \mathbb{I}_{K_1}\right]^{-1} \sum_{i \in B_t} \sum_m \hat{z}_{im} v_m, \tag{6}$$

where $\tilde{Z} = Z - SG - X_D DV$, B_t is the set of indices of samples from donors with phenotype t, and N_t is the number of elements in B_t.

When optimizing the object function defined by Equation 3 with respect to G, the Lagrange dual proposed in the study [5] is employed. The Lagrange dual for our problem is of the following form

$$\mathcal{L}(G, \vec{\psi}) = \frac{1}{2} \operatorname{tr}\left(G^\intercal Q G\right) - \operatorname{tr}\left(WG\right) + \frac{1}{2} \operatorname{tr}(\Psi G G^\intercal - c\Psi) + const.$$

Here $\tilde{Z} = Z - X_D DV - X_P PV$, $Q = S^\intercal S$, $W = \tilde{Z}^\intercal S$, and Ψ is a $K_2 \times K_2$ diagonal matrix with dual variables ψ expanding along its diagonal. By taking the derivative with respect to G, we have

$$G = \left(Q + \Psi\right)^{-1} W^\intercal. \tag{7}$$

Then, by substituting Equation 7 into Equation 2.2, we have the following dual for Equation 2.2

$$\mathcal{D}(\vec{\psi}) = \frac{1}{2} \operatorname{tr}\left((-W\left(Q + \Psi\right)^{-1} W^\intercal - c\Psi)\right) + const. \tag{8}$$

The gradient ∇ and Hessian H of the above dual with respect to $\vec{\psi}$ can be derived as follows:

$$\begin{aligned} \nabla_i = \frac{\partial \mathcal{D}(\vec{\psi})}{\partial \psi_i} &= \frac{1}{2} \| W (Q + \Psi)^{-1} e_i \|^2 - \frac{1}{2} c. \\ H_{ij} = \frac{\partial^2 \mathcal{D}(\vec{\psi})}{\partial \psi_i \partial \psi_j} &= -\left((Q + \Psi)^{-1} W^\intercal W (Q + \Psi)^{-1}\right)_{i,j} \left((Q + \Psi)^{-1}\right)_{i,j}. \end{aligned}$$

The Newton's method is used to optimize Equation 8 with respect to Ψ. Thus, the update for Ψ at iteration t can be written as

$$\Psi^{(t)} = \Psi^{(t-1)} - (H^{(t-1)})^{-1} \nabla^{(t-1)}, \tag{9}$$

where $\Psi^{(t-1)}, H^{(t-1)}, \nabla^{(t-1)}$ are the diagonal matrix of ψ, gradient, and Hessian matrix at iteration $t-1$. In practice, we alternatively compute the updates for G, Ψ with Equation 7 and 9, respectively,

until the sum of squared difference in Ψ between two consecutive iterations less than a predefined threshold (10^{-4} is used in our studies). Further details on the derivation of Lagrange dual are provided in $A.1$ in the Appendices.

When optimizing Equation 3 with respect to s_i, the i-th row of S, our objective function with respect to s_i can be simplified to

$$\mathcal{L}(s_i) = \frac{1}{2}\|\tilde{Z}_i - G^T s_i\|_2^2 + \frac{1}{2}\lambda(1-\alpha)\|s_i\|_2^2 + \lambda\alpha|s_i|_1. \tag{10}$$

Here $\tilde{Z} = Z - X_D DV - X_P PV$, and $\tilde{Z}_i$ is the vector of the i-th row of $\tilde{Z}$. Random coordinate descent (RCD) with strong rules is proposed in Algorithm 1 in Study [6] to solve the problem.

Optimize with the batch strategy

When the scale of data is huge, optimizing SC2 with the whole data is memory demanding. To relieve this issue, we propose a batch strategy to optimize SC2. In practice, we split the whole data into several batches and optimize our object function with one batch each time to lower the memory consumption.

The key to perform batch optimization of SC2 is to come up with a *surrogate* that asymptotically converges to the same solution defined by Equation 3. As inspired by a previous study [7], we come up with the following *surrogate* for our objective

$$\begin{aligned}\ell(V,G) =& \frac{1}{k}\sum_{j}^{k}\frac{1}{2}\big\|Z_j - (X_{D_j}D_j + X_{P_j}P_j)V - S_jG\big\|_F^2 + \\ & \frac{1}{k}\sum_{j}^{k}\frac{1}{2}\lambda_1(\|D_j\|_F^2 + \|P_j\|_F^2) + \frac{1}{2}\lambda_1\|V\|_F^2 + \\ & \frac{1}{k}\sum_{j}^{k}\lambda_2\big[\frac{1}{2}(1-\alpha)\|S_j\|_F^2 + \alpha\|S_j\|_1\big],\end{aligned} \tag{11}$$

Here k is the number of batches, and P_j, D_j, S_j are obtained with previous batches. Algorithm 1 is proposed to optimize the above *surrogate*.

In Algorithm 1, we note that A_t, B_t, E_t, F_t carry all information from the past iteration. In particularly, these matrices can carry "old" information for the same batch in different iterations. Actually, this kind of information is outdated. Mairal et al. suggested that one can accelerate convergence by removing old information for the same batch from these matrices [7]. Specifically,

Algorithm 1: Batch SC2

Data: $Z \in \mathbb{R}^{N\times M}$, $V_0 \in \mathbb{R}^{K_1\times M}$, $V_0 \in \mathbb{R}^{K_1\times M}$(random initiation), $G_0 \in \mathbb{R}^{K_2\times M}$ (random initiation), $X_D^{N\times N_1}$, $X_P^{N\times N_2}$, T (maximum of iterations), K (number of batches)

Result: V, G

Divide $Z \in \mathbb{R}^{N\times M}$ into K batches;

for $t \leftarrow 0$ **to** $T-1$ **do**

  if *$t == 0$* **then**

    $D_0 \leftarrow 0, P_0 \leftarrow 0, S_0 \leftarrow 0, A_0 \leftarrow 0, B_0 \leftarrow 0, E_0 \leftarrow 0, F_0 \leftarrow 0$;

  else

    $D_0 \leftarrow D_K, P_0 \leftarrow P_K, S_0 \leftarrow S_K, A_0 \leftarrow A_K, B_0 \leftarrow B_K, E_0 \leftarrow E_K, F_0 \leftarrow F_K$;

  end

  for $k \leftarrow 1$ **to** K **do**

    $\tilde{Z}_k \leftarrow Z_k - S_{k-1}G_{k-1}$;

    Compute with a closed form similar to Equation 5

$$D_k \triangleq \operatorname{argmin}_D \left\|\tilde{Z}_k - X_D^k D V_{k-1} - X_P^k P_{k-1} V_{k-1}\right\|_F^2 + \lambda_1 \|D\|_F^2.$$

    Compute with a closed form similar to Equation 6

$$P_k \triangleq \operatorname{argmin}_P \left\|\tilde{Z}_k - X_D^k D_k V_{k-1} - X_{P_k} P V_{k-1}\right\|_F^2 + \lambda_1 \|P\|_F^2.$$

    $A_k \leftarrow A_{k-1} + (X_D^k D_k + X_P^k P_k)^\intercal (X_D^k D_k + X_P^k P_k)$, $B_k \leftarrow B_{k-1} + \tilde{Z}_k^\intercal (X_D^k D_k + X_P^k P_k)$;

    Compute with a closed form similar to Equation 4

$$\begin{aligned} V_k \triangleq & \operatorname{argmin}_V \frac{1}{k} \sum_{j=1}^{k} \left\|\tilde{Z}_j - (X_D^j D_j + X_P^j P_j) V\right\|_F^2 + \lambda_1 \|V\|_F^2 \\ = & \operatorname{argmin}_V \frac{1}{k} \operatorname{tr}(V^\intercal A_k V) - \frac{2}{k} \operatorname{tr}(B_k V) + \lambda_1 \|V\|_F^2. \end{aligned}$$

    $\tilde{Z}_k \leftarrow Z_k - (X_D^k D_k + X_P^k P_k) V_k$;

    Compute with RCD with strong rules

$$S_k \triangleq \frac{1}{2} \operatorname{argmin}_S \left\|\tilde{Z}_k - S G_{k-1}\right\|_F^2 + \lambda_2 \left[\frac{1}{2}(1-\alpha)\|S_i\|_F^2 + \alpha \|S_i\|_1\right].$$

    $E_k \leftarrow E_{k-1} + S_k^\intercal S_k$, $F_k \leftarrow F_{k-1} + \tilde{Z}_k^\intercal S_k$;

    Compute with Lagrange dual

$$\begin{aligned} G_k \triangleq & \operatorname{argmin}_G \frac{1}{k} \sum_{j=1}^{k} \frac{1}{2} \left\|\tilde{Z}_j - S_j G\right\|_F^2 \quad \text{s.t.} \ \|G_i\|_2^2 \le c, \forall i = 1, \ldots, K_2 \\ = & \operatorname{argmin}_G \frac{1}{k} \sum_{j=1}^{k} \frac{1}{2} \operatorname{tr}\left(G^\intercal E_j G\right) - \operatorname{tr}\left(F_j G\right) \quad \text{s.t.} \ \|G_i\|_2^2 \le c, \forall i = 1, \ldots, K_2. \end{aligned}$$

  end

end

owning to the design of our algorithm, we use the following equations to exploit this idea

$$\begin{aligned} A_k \leftarrow & A_{k-1} - (X_D^k D_k' + X_P^k P_k')^\intercal (X_D^k D_k' + X_P^k P_k') \\ B_k \leftarrow & B_{k-1} - \tilde{Z_k'}^\intercal (X_D^k D_k' + X_P^k P_k') \\ E_k \leftarrow & E_{k-1} - S_k'^\intercal S_k' \\ F_k \leftarrow & F_{k-1} - \tilde{Z_k'}^\intercal S_k'. \end{aligned} \tag{12}$$

Here D_k', P_k', S_k' are the corresponding matrices from the previous iteration $t-1$. With slight abuse of notation, $\tilde{Z'}_k$ in the second and fourth line are computed with equations in lines 9 and 14 in Algorithm 1, respectively.

2.3 Initialization, hyperparameter tuning, and the stopping criteria

In SC2, all latent variables (D, P, V, S, G) are initiated from normal distribution $N(0, 0.001)$. For the initial start of s_i in solving subproblems defined by Equation 10, we consider the solution from ridge regression or the solution for s_i from the previous iteration, depending on which one leads to a lower loss in the objective defined by Equation 10. Following the subproblem defined by Equation 10, the warm start with ridge solution is as follows

$$s_i = [GG^T + \lambda(1-\alpha)\mathbb{I}_{K_2}]^{-1} G\tilde{Z}_i.$$

For model selection in SC2, grid search is utilized to select hyperparameters $\lambda_1, \lambda_2, \alpha, K_1, K_2$. In practice, when the number of observations is huge (e.g., $\geq$ 500,000), we randomly and evenly draw a small proportion (e.g., 0.1 or even less) of scRNA-Seq samples as dataset for model selection. Then, we randomly draw 10% elements from the matrix of the dataset drawn for model selection as testset and select the set of hyperparameters that performs the best in terms of root-mean-square error (RMSE) on the testset. For each set of candidate hyperparameters, we run alternating BCD a number of times (e.g., 20) and choose the one with the best performance on the testset.

In practice, we find that SC2 always chooses α equal 1 in model selection, that is, SC2 favors Lasso penalty in applications. Thus, to simplify the parameter tuning for SC2, we set α equal 1. Meanwhile, we also consider that our two hyperparameters K_1 and λ_1 are kind of redundant, since one can increase K_1 and λ_1 simultaneously without changing the model complexity. Therefore, we consider $K_1 = K_2$ in model selection.

The detailed procedure for model selection is as follows. First, we set λ_1, λ_2 to a small number (e.g., 0.1) to avoid singularity in matrix inverse and α is fixed to 1 and choose the ranks of latent

representations $K_1 = K_2$ from sequences from 5 to 30 with step size 2. Then, after choosing the ranks of latent representations, we define a broad parameter grid for λ_1, λ_2 and perform a grid search to tuning hyperparameters. We may also refine the parameter grid based on the performance of our parameter sets on the testset. Finally, we select the parameters λ_1, λ_2 with the best performance on the testset and run SC2 with the selected parameters until stopping criteria meets.

In our study, the loss in the i-th iteration $\ell_i(\cdot)$ is calculated every 10 iterations to reduce computational burden. Then, the stopping criteria is defined as

$$\frac{|\ell_i(\cdot) - \ell_{i-10}(\cdot)|}{\ell_{i-10}(\cdot)} < \sigma, \tag{13}$$

where σ is a predefined threshold and set to 10^{-8} in our experiment.

References

[1] A. Butler, P. Hoffman, P. Smibert, E. Papalexi, and R. Satija, "Integrating single-cell transcriptomic data across different conditions, technologies, and species," *Nature biotechnology*, vol. 36, no. 5, pp. 411–420, 2018.

[2] J. Liu, C. Gao, J. Sodicoff, V. Kozareva, E. Z. Macosko, and J. D. Welch, "Jointly defining cell types from multiple single-cell datasets using liger," *Nature protocols*, vol. 15, no. 11, pp. 3632–3662, 2020.

[3] K. Qian, S. Fu, H. Li, and W. V. Li, "Scinsight for interpreting single-cell gene expression from biologically heterogeneous data," *Genome biology*, vol. 23, no. 1, pp. 1–23, 2022.

[4] G. L. Stein-O'Brien, R. Arora, A. C. Culhane, A. V. Favorov, L. X. Garmire, C. S. Greene, L. A. Goff, Y. Li, A. Ngom, M. F. Ochs, *et al.*, "Enter the matrix: Factorization uncovers knowledge from omics," *Trends in Genetics*, vol. 34, no. 10, pp. 790–805, 2018.

[5] H. Lee, A. Battle, R. Raina, and A. Ng, "Efficient sparse coding algorithms," *Advances in neural information processing systems*, vol. 19, 2006.

[6] K. ZHAO, S. Huang, L. Cuichan, P. C. Sham, S. Hon-Cheong, and Z. Lin, "Insider: Interpretable sparse matrix decomposition for bulk RNA expression data analysis," *bioRxiv*, 2022.

[7] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 689–696.

A Appendices

A.1 Lagrange dual for learning bases

Here we derive the updates of bases B in the problem defined by the following:

$$\text{minimize} \qquad \|Z - SG\|_F^2 + \lambda\|S\|_1 \qquad \text{subject to } \|G_i\|_2^2 \leq 1, \forall i = 1, \ldots, k,$$

where Z is the matrix to be approximated, S is the sparse coding, and G is the matrix for the bases. In the equation, G_i is the i-th row of G.

To solve the problem, we consider the following Lagrangian:

$$\begin{aligned} \mathcal{L}(G, \vec{\lambda}) &= \frac{1}{2}\operatorname{tr}\big((Z - SG)^\intercal(Z - SG)\big) + \frac{1}{2}\sum_k \lambda_k(\sum_j G_{kj}^2 - 1) \\ &= \frac{1}{2}\operatorname{tr}\big((Z - SG)^\intercal(Z - SG)\big) + \frac{1}{2}\operatorname{tr}(\Lambda G G^\intercal - \Lambda), \end{aligned} \tag{14}$$

where each $\lambda_k \geq 0$ is a dual variable and $\Lambda = \operatorname{diag}(\vec{\lambda})$. By taking derivative with respect to G, we obtain

$$\frac{\partial \mathcal{L}(G, \vec{\lambda})}{\partial G} = -Z^\intercal S + S^\intercal S G + \Lambda G = 0.$$

Here $F = Z^\intercal S$ and $E = S^\intercal S$. With these notations, we have

$$G = \big(E + \Lambda\big)^{-1} F^\intercal. \tag{15}$$

By replacing Equation 15 into the Lagrangian 14, we can further derive the Lagrange dual for our problem:

$$\begin{aligned} \mathcal{D}(\vec{\lambda}) =& \min_G \mathcal{L}(G, \vec{\lambda}) = \frac{1}{2}\operatorname{tr}\big((Z^\intercal Z - 2FG + G^\intercal E G + \Lambda G G^\intercal - \Lambda)\big) \\ =& \frac{1}{2}\operatorname{tr}\big((Z^\intercal Z - 2F\big(E + \Lambda\big)^{-1}F^\intercal + F\big(E + \Lambda\big)^{-1}E\big(E + \Lambda\big)^{-1}F^\intercal + \\ & F\big(E + \Lambda\big)^{-1}\Lambda\big(E + \Lambda\big)^{-1}F^\intercal - \Lambda)\big) \\ =& \frac{1}{2}\operatorname{tr}\big((Z^\intercal Z - 2F\big(E + \Lambda\big)^{-1}F^\intercal + F\big(E + \Lambda\big)^{-1}F^\intercal - \Lambda)\big) \\ =& \frac{1}{2}\operatorname{tr}\big((Z^\intercal Z - F\big(E + \Lambda\big)^{-1}F^\intercal - \Lambda)\big), \end{aligned}$$

which is formulated as

$$\mathcal{D}(\vec{\lambda}) = \frac{1}{2}\operatorname{tr}\big((Z^\intercal Z - F\big(E + \Lambda\big)^{-1}F^\intercal - \Lambda)\big). \tag{16}$$

The gradient and Hessian of $\mathcal{D}(\vec{\lambda})$ are computed as follows:

$$\frac{\partial \mathcal{D}(\vec{\lambda})}{\partial \lambda_i} = \frac{1}{2}\|F(E+\Lambda)^{-1}e_i\|^2 - \frac{1}{2}. \tag{17}$$

$$\frac{\partial^2 \mathcal{D}(\vec{\lambda})}{\partial \lambda_i \partial \lambda_j} = -\big((E+\Lambda)^{-1}F^\intercal F(E+\Lambda)^{-1}\big)_{i,j}\big((E+\Lambda)^{-1}\big)_{i,j}. \tag{18}$$

The Newton's method is used to optimize the above problem.