SelectSmart: A Multi-Agent Reinforcement Learning Framework for Gene Panel Selection

Erwin Poussi
Qiu Lab, Stanford University
Stanford, CA, USA
erwinpi@stanford.edu

***Abstract*—Gene panel selection aims to identify a compact subset of genes that preserves the biological structure of single-cell datasets while reducing experimental cost. Classical methods often disagree and fail to generalize across tissues, motivating a unified and adaptive approach.**

We introduce SelectSmart, a reinforcement-learning framework that treats each gene as an agent acting over a learned representation of the current panel. By combining meta-voted candidate genes with an actor–critic architecture and a reward based on clustering fidelity ARI and panel-size regularization, SelectSmart learns directly from data rather than relying on fixed heuristics.

Trained on a 30k-cell kidney dataset and evaluated on an independent CZ Kidney dataset, SelectSmart produces a 500-gene panel that preserves transcriptomic geometry and outperforms classical gene panel selection methods. Visual and quantitative analyses confirm strong generalization.

We conclude with further improvements—notably slow reward computation and scaling challenges—and outline future directions such as GPU-accelerated clustering, richer state representations, and surrogate reward models.

***Index Terms*—Single-cell transcriptomics, gene panel selection, reinforcement learning, clustering, ARI, NMI.**

I. Introduction

GENE panel selection is a fundamental problem in single-cell and spatial transcriptomics. Modern assays measure more than 20,000 genes per cell, but sequencing such a large set is costly, technically challenging, and unnecessary for most biological analyses. Instead, biologists rely on *gene panels*—compact subsets of 200–1000 genes—designed to preserve the structure of the cellular landscape and enable accurate cell-type discrimination. The core challenge is to identify a subset that maintains biological resolution while substantially reducing experimental cost.

Classical selection strategies often disagree sharply. When applied to the same dataset, they frequently produce almost disjoint panels with inconsistent performance across tissues and conditions. This lack of consensus suggests that each method captures only a partial aspect of cellular structure, motivating the need for a principled framework that can *aggregate and refine* heterogeneous signals.

In this work, we introduce **SelectSmart**, a reinforcement-learning (RL) framework for gene panel selection inspired by the multi-agent formulation of RiGPS [9]. We cast the task as a sequential decision process in which each gene acts as an agent choosing whether to be included in the panel based on a learned representation of the current subset. While our long-term goal is to design ∼1000-gene panels for large-scale cancer spatial transcriptomics datasets—often exceeding 300,000 cells [3]—such data introduce substantial computational and stability challenges. As a proof of concept, we therefore evaluate SelectSmart on a kidney cancer dataset of ∼30,000 cells, which provides sufficient biological diversity while enabling stable training and reproducible benchmarking.

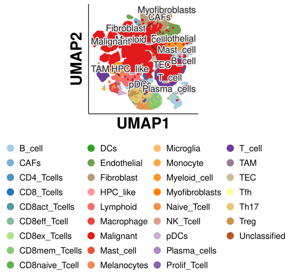

Fig. 1. UMAP projection of cancer single-cell RNA-seq data from [3]. Cells belonging to the same identity form coherent clusters, illustrating the structure that a reduced gene panel should preserve.

Overall, SelectSmart illustrates how reinforcement learning can unify diverse selection criteria within a single, data-driven framework. By learning to integrate signals from multiple baseline methods, it produces compact and biologically meaningful panels and establishes a foundation for future extensions to multi-tissue or pan-cancer gene panel design.

II. Related Work and Goal of Gene Panel Selection

A central requirement in single-cell analysis is to preserve the geometric structure of the full transcriptomic space when reducing gene dimensionality. Figure 1 illustrates this structure on a pancancer cells dataset. Tumor and immune populations form coherent and well-separated clusters when embedded in a low-dimensional space. A successful gene panel must retain this separability even after reducing from approximately 20,000 genes to a compact subset suitable for downstream assays. This constraint underlies both biological interpretability and the practical deployment of spatial transcriptomics technologies.

A variety of computational strategies have been proposed to select informative genes. Highly variable gene (HVG) selection identifies features with elevated dispersion across cells and is widely used to enhance unsupervised clustering [5]. Differential expression (DE) approaches focus on genes enriched within specific cell populations and rely on non-parametric tests such as Wilcoxon [7] to extract putative markers. Random forest–based importance scores provide a supervised alternative by ranking genes according to their contribution to classification performance [6]. More recent methods such as scGeneFit [2] formulate marker selection as a convex optimization problem that preserves label separation in a compressed representation.

Although each method captures meaningful biological signals, they often yield divergent gene panels and their performance varies widely across tissues, levels of heterogeneity, and sequencing platforms. These discrepancies indicate that no single criterion provides a comprehensive solution. Gene panel selection therefore benefits from approaches capable of integrating multiple and complementary sources of information.

Our framework builds on this insight. By aggregating signals from classical methods and casting gene selection as a reinforcement-learning problem, we aim to develop a principled and adaptive strategy that learns directly from the structure of the data rather than relying on fixed heuristics. This perspective follows the decentralized multi-agent reinforcement-learning formulation introduced in RiGPS [9], which we extend for label-guided panel refinement in cancer transcriptomics.

III. SelectSmart Approach

SelectSmart formulates gene panel selection as a reinforcement-learning (RL) problem in which genes behave as decision-making agents acting over a compact representation of the current panel. The method combines prior knowledge extracted from classical selection strategies with multi-agent RL, yielding an adaptive and data-driven mechanism for constructing biologically meaningful panels. This section presents the RL formulation, the reward function, the construction of the candidate agent set $\tilde{G}$, and the overall training pipeline.

A. Problem Formulation

We follow the sequential decision-making formalism of [9]. Let G_{all} denote the full set of genes measured in the dataset (typically more than 20,000). Only a small fraction of these genes is biologically informative for distinguishing cell populations, and treating all 20,000 genes as RL agents is computationally impractical. We therefore introduce a reduced candidate set $\tilde{G} \subseteq G_{\text{all}}$, constructed through the meta-voting strategy described in Section III-F. Each gene $g \in \tilde{G}$ is modeled as an agent that must decide whether it should be included in the panel at a given iteration.

At iteration t, the panel is a subset $G_t \subseteq \tilde{G}$. The RL objective is to iteratively refine G_t by selecting the genes that maximize biological separability while maintaining a desired target panel size.

B. State Representation

A central challenge is to represent the evolving panel G_t in a form that is both compact and informative. Instead of using a sparse $|\tilde{G}|$-dimensional indicator vector, SelectSmart encodes the panel through global summary statistics. Let $X_{[:,G_t]}$ denote the expression matrix of the cells restricted to the genes in G_t. We compute a 16-dimensional feature vector

$$S_t = \phi(X_{[:,G_t]}) \in \mathbb{R}^{16} \tag{16}$$

where ϕ aggregates mean, variance, sparsity, quantiles, and additional panel-level statistics (details in Appendix).

A neural encoder E_θ maps this representation into a latent embedding:

$$z_t = E_\theta(S_t) \in \mathbb{R}^{64}. \tag{1}$$

which serves as the shared state input for all agents. This embedding captures the global structure of the current panel and conditions the policy on a compressed but expressive representation of G_t.

C. Actions and Panel Update

Each gene g takes a binary action

$$a_t^{(g)} \in \{0, 1\}. \tag{2}$$

where 1 indicates selection and 0 indicates exclusion. After all agents act independently, the next panel is formed deterministically:

$$G_{t+1} = \{\, g \in \tilde{G} \;:\; a_t^{(g)} = 1 \,\}. \tag{3}$$

This formulation enables parallel, decentralized decision-making while maintaining a coherent global update rule based on the expression patterns captured by the evolving panel.

D. Actor–Critic Architecture

SelectSmart uses a multi-agent actor–critic algorithm. In actor–critic methods, the *actor* learns a policy that maps the state z_t to an action distribution, while the *critic* estimates the value of a state in order to provide low-variance learning signals. This decomposition is quite effective when many agents must coordinate through shared global feedback.

Each gene shares the same state z_t but maintains its own actor and critic networks:

$$\pi_\theta^{(g)}(1 \mid z_t) = \sigma\Big(f_\theta^{(g)}(z_t)\Big), \qquad V_\psi^{(g)}(z_t). \tag{4}$$

Actor–critic is particularly suitable here because:

- The action space is binary and high-dimensional (hundreds to thousands of agents).
- Critics stabilize learning by providing smoother gradient estimates when rewards depend on all agents jointly.
- Multi-agent independence in the actors allows scalable parallel updates.

A transition stored in the replay buffer takes the form

$$(z_t,\, a_t^{(g)},\, r_t,\, z_{t+1}). \tag{5}$$

E. Reward Design

A valid gene panel must satisfy two criteria: (i) preserve biological separability of cell populations, and (ii) remain close to a target size.

We quantify separability using the Adjusted Rand Index (ARI) computed after clustering the dataset using a standard single-cell RNA-seq pipeline (PCA, neighborhood graph, Leiden clustering). ARI measures agreement between the clustering obtained from G and the true cell-type annotations.

The reward is

$$r(G) = \alpha \, \mathrm{ARI}(G) + (1-\alpha) \, \mathrm{SizeTerm}(|G|), \tag{6}$$

with $\alpha = 0.9$.

The size term encourages the panel to stay near a target cardinality K_{target}:

$$\mathrm{SizeTerm}(|G|) = \begin{cases} 1, & |G| \le K_{\text{target}}, \\ \left(1 - \dfrac{|G| - K_{\text{target}}}{K_{\max} - K_{\text{target}}}\right)^{\beta} \\ \quad K_{\text{target}} < |G| \le K_{\max}, \\ 0, & |G| > K_{\max}. \end{cases} \tag{7}$$

This smooth term prevents collapsing to overly large or overly small panels and facilitates stable critic learning. We choose $\beta = 1.5$ in this study.

F. Constructing the Candidate Set $\tilde{G}$

We construct the reduced candidate set $\tilde{G}$ by aggregating the outputs of four classical baseline methods, introduced in Section II: highly variable gene (HVG) selection, differential expression, random forest feature importance, and scGeneFit. Each method captures a distinct notion of gene informativeness, and combining them provides a more stable starting point for reinforcement learning.

a) Step 1: Per-method Scoring.: Each method m assigns a score to all genes in G_{all}. Classical pipelines typically form a panel by selecting the top-k ranked genes according to this score. In our setting, we retain the full scoring distribution produced by each method to enable downstream aggregation.

b) Step 2: Extraction of Top-k Genes.: For each method, we extract a size-k panel:

$$T_m = \mathrm{TopK}_m(k). \tag{8}$$

For example, when targeting a 500-gene kidney cancer panel, we set $k = 500$. This yields four candidate sets $\{T_m\}$ reflecting different selection heuristics.

c) Step 3: Method Reliability.: We assess the quality of each T_m by evaluating it with the reward function (Eq. 6):

$$r_m = r(T_m). \tag{9}$$

This quantifies how well each standalone panel preserves biological structure while satisfying panel-size constraints.

d) Step 4: Weight Normalization.: Based on their performance, methods are assigned reliability weights:

$$w_m = \frac{r_m}{\sum_j r_j}. \tag{10}$$

Methods yielding higher-quality panels receive larger weights, ensuring that the aggregation process emphasizes more reliable sources.

e) Step 5: Meta-score and Selection.: We combine the per-method scores and their reliability to form a consensus score. For every gene g appearing in the union of all top-k sets $\bigcup_m T_m$, we compute a weighted meta-score:

$$s_g = \sum_m w_m \, \tilde{s}_{m,g}. \tag{11}$$

where $\tilde{s}_{m,g}$ denotes the z-normalized score assigned to gene g by method m.

Genes with the highest meta-scores are retained to form $\tilde{G}$. In our case we selected the 1200 best candidates, therefore **Selectsmart** consisted of 1200 interacting agents. This weighted aggregation mitigates method-specific biases, filters out noisy genes, and produces a biologically coherent and computationally tractable candidate set for the RL stage.

G. Training Pipeline

Training combines knowledge injection, exploration, and gradient-based optimization.

a) Knowledge Injection.: Before RL begins, transitions corresponding to baseline panels are injected into the replay buffer:

$$(z_0, \, a_f^{(g)}, \, r_f, \, z_f). \tag{12}$$

This stabilizes early learning by anchoring the policy around known informative configurations.

b) Exploration.: SelectSmart uses ε-exploration: with probability ε, actions are sampled uniformly at random; with probability $1-\varepsilon$, the actor outputs are used. This mechanism ensures sufficient exploration early in training and gradually shifts toward exploitation as ε decays.

At each epoch:

1) Encode state: $z_t = E_\theta(\phi(X_{[:,G_t]}))$.
2) Sample actions via ε-exploration.
3) Construct G_{t+1}.
4) Compute reward and store transitions.

c) Actor–Critic Optimization.: For each gene g, we update in the optimization phase:

- The critic minimizes

$$\mathcal{L}_{\text{critic}} = \Big(V_\psi^{(g)}(z_t) - \Big[r_t + \gamma V_\psi^{(g)}(z_{t+1}) \Big] \Big)^2 . \tag{13}$$

- The actor minimizes

$$\mathcal{L}_{\text{actor}} = -A_t^{(g)} \log \pi_\theta^{(g)}\Big(a_t^{(g)} \mid z_t\Big) - \beta_{\text{ent}} \, \mathcal{H}\Big(\pi_\theta^{(g)}\Big) . \tag{14}$$

- where the advantage is

$$A_t^{(g)} = r_t + \gamma V_\psi^{(g)}(z_{t+1}) - V_\psi^{(g)}(z_t). \tag{15}$$

The entropy term $\mathcal{H}(\pi_\theta^{(g)})$ encourages the policy to remain sufficiently stochastic during training. This prevents premature convergence (e.g., always selecting or ignoring a gene), stabilizes gradients, and promotes exploration of diverse panel configurations before the policy becomes more deterministic later in training.

H. Algorithm

Algorithm 1. SelectSmart Training Procedure

```
Require: Dataset $X$, candidate set $\tilde{G}$, reward $r(\cdot)$
Initialize encoder $E_\theta$, actors $\pi_\theta^{(g)}$, critics $V_\psi^{(g)}$
Initialize replay buffers and initial panel $G_0$
Inject knowledge from baseline panels
for epoch $= 1 \dots T$ do
    Encode state $z_t = E_\theta(\phi(X_{[:,G_t]}))$
    Evaluate reward $R_t = r(G_t)$
    for $n = 1 \dots N_{\text{explore}}$ do
        Sample actions using $\varepsilon$-exploration
        Form $G_{t+1}$ and compute reward $r_t$
        Store transitions in replay buffers
        $G_t \leftarrow G_{t+1}$
    end for
    for $m = 1 \dots N_{\text{optimize}}$ do
        for each gene $g \in \tilde{G}$ do
            Sample mini-batch from replay buffer $\mathcal{B}_g$
            Update critic $V_\psi^{(g)}$ via TD error
            Update actor $\pi_\theta^{(g)}$ via policy gradient
        end for
    end for
    Decay $\varepsilon$
    Save best panel if improved
end for
```

I. Hyperparameters

Parameter	Value
Knowledge injection samples	100
Initial ε	0.5
ε-decay	0.97
Epochs	50
Exploration steps	12 per epoch
Optimization steps	8 per epoch
Discount factor γ	0.99
Reward weight α	0.9
Size penalty β	1.5
Target panel size K_{target}	500
Maximum panel size $K_{\max}$	1000
PCA components	50
k-NN neighbors	15
Leiden resolution	1.0

TABLE I
HYPERPARAMETERS USED IN SELECTSMART.

Component	Architecture
Encoder	MLP 16–128–64–64 (LayerNorm, ReLU, Dropout 0.1)
Actors	MLP 64–128–32–1, sigmoid output
Critics	Same as actors
Optimizers	Adam, lr 10^{-3} (actors/encoder), $5 \cdot 10^{-4}$ (critics)
Replay buffer	2000 transitions per gene, batch size 64

TABLE II
NEURAL NETWORK ARCHITECTURES AND OPTIMIZATION SETTINGS.

J. Scalability

Each training epoch performs N_{explore} exploration steps and N_{opt} optimization steps. The total number of RL transitions is thus on the order of $T(N_{\text{explore}} + N_{\text{opt}})$, so we have linear complexity. The computational bottleneck is reward evaluation, which requires running a full single-cell clustering pipeline (PCA, neighborhood graph, Leiden) which is inherent in any gene panel selection method.

In our experiments on a dataset of $\sim$ 30,000 cells, reward evaluation dominated runtime and was executed on CPU, leading to a total runtime of approximately 8 hours on an H100-equipped workstation. The RL computations themselves remain lightweight, and scaling to larger datasets is primarily limited by the speed of external clustering algorithms rather than by SelectSmart. Further work will involve investigating faster reward computing options.

IV. RESULTS AND DISCUSSION

A. SelectSmart Training Dynamics

During training, the policy evolves through the balance between exploration and exploitation. Figure 2 summarizes four complementary diagnostics collected over 50 epochs (corresponding to $50 \times (12 + 8) = 1000$ iterations):

- **Reward at epoch start** (leftmost): evaluation of the reward associated with the panel produced by the policy at the beginning of each epoch. This reflects the *current* quality of the policy.
- **Exploration schedule** (middle–left): decay of the exploration rate ε, which controls how often actors choose random actions instead of following the learned policy. High ε at the beginning promotes wide exploration of the panel space, while its gradual decay encourages convergence toward more stable choices.
- **Panel size evolution** (middle–right): size of the gene panel $|G_t|$ selected at each epoch. Despite fluctuations inherent to exploration, the panel remains stably centered around the target size.
- **Exploration-phase rewards** (top–right): mean reward collected during exploratory rollouts at each epoch, with a 95% confidence envelope. This shows how the policy behaves not only in greedy mode but also under noise and sampling variability.

At the end of training, SelectSmart returns the best panel ever discovered during search. We observe a clear upward trend in reward quality, and the highest-performing panel emerges near epoch 42, indicating successful convergence of the learning process.

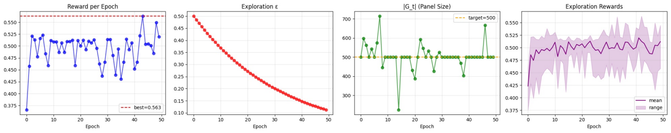

Fig. 2. **SelectSmart training diagnostics.** Reward progression, exploration decay, panel-size stability, and exploration-phase reward distribution over 50 epochs.

B. Visualization of the Final Gene Panel on the Training Dataset

To verify that the learned panel preserves biological structure, we compare in Figure 3 the UMAP embedding of the full transcriptome of 20,000 genes(left) with the UMAP embedding produced using only the 500-gene RL panel (right).

Both embeddings show well-separated cell-type clusters, including malignant, lymphoid, myeloid, endothelial, and fibroblast populations. Importantly, the RL panel preserves the global geometry of the manifold despite using only 500 genes instead of tens of thousands. Preserving the malignant–immune–stromal separation is essential for downstream tumor microenvironment analyses, and the RL panel maintains these boundaries despite a 40× reduction in dimensionality.

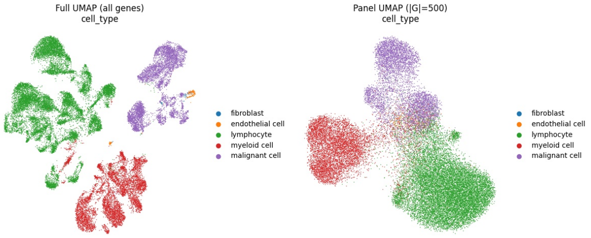

Fig. 3. **UMAP embeddings of the training dataset.** Left: full gene set. Right: 500-gene RL-derived panel. Cell-type separability is largely preserved.

C. Benchmarking and Panel Evaluation

To assess the generalization capacity of SelectSmart, we benchmark the RL-designed panel on an *independent* kidney dataset from the Chan Zuckerberg CellxGene collection (CZ CellxGene dataset).

.

We evaluate three widely used clustering-quality metrics:

- **Adjusted Rand Index (ARI)** — measures how well Leiden-derived clusters match the true annotated cell types; ARI = 1 indicates perfect agreement.
- **Normalized Mutual Information (NMI)** — quantifies the mutual information between predicted clusters and true labels; NMI is robust to class imbalance.
- **Silhouette Index (SI)** — measures cluster compactness relative to separation. Higher SI indicates tighter clusters, although in some biological datasets loose, elongated clusters are expected and even desirable.

Since real-world annotation is typically obtained by (i) performing unsupervised Leiden clustering, (ii) examining marker-gene expression for each cluster, and (iii) assigning a biological identity, ARI and NMI provide a faithful measure of how well a gene panel preserves biologically meaningful structure.

We compared SelectSmart to four classical methods (HVG, differential expression, random forest, scGeneFit) over five non-overlapping batches of the test dataset. Figure 4 reports the distribution of all metrics.

SelectSmart consistently achieves the best ARI and NMI, indicating superior preservation of cell-type identity. The Silhouette Index is lower for RL, which is consistent with our observations from the visualization: kidney cell clusters naturally exhibit elongated or overlapping geometries. In this setting, a lower SI is not detrimental and can even be more biologically realistic. Overall Selectsmart considerably dominates baselines methods which is encouraging for further study.

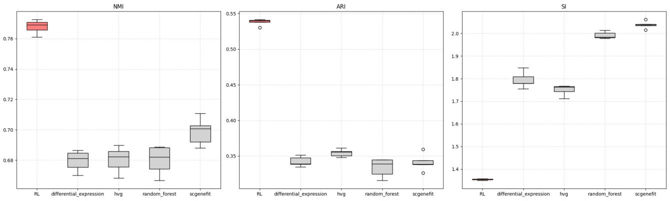

Fig. 4. **Benchmarking of the RL panel vs. baseline methods** on independent kidney datasets using NMI, ARI, and Silhouette Index. SelectSmart achieves consistently better clustering agreement with ground truth.

D. Limitations and Future Work

Several improvements could further strengthen SelectSmart and make it suitable for large-scale, production-level gene panel design:

- **Richer state representations**: the current state summarizes the panel using global aggregated statistics. More expressive encodings—such as transformer-based embeddings, graph neural networks, or contrastive representations—may increase policy sensitivity to subtle gene–gene interactions.
- **Scaling to multi-cancer datasets**: extending SelectSmart to pan-cancer datasets (e.g., breast, skin, kidney, lung) would enable construction of universal 1000-gene oncology panels. Achieving this requires an efficient strategy to downsample extremely large datasets while preserving full cellular diversity and producing kernel-manageable data.
- **GPU-accelerated reward computation**: the current bottleneck is CPU-based clustering in Scanpy [8]. Replacing this with GPU-accelerated frameworks such as RAPIDS SingleCell [1] could drastically reduce reward evaluation time and allow SelectSmart to scale to much larger datasets.
- **Reward surrogates**: SelectSmart remains slowed by reward computation, which relies on repeated single-cell clustering. Learning a differentiable surrogate model of ARI/NMI could significantly accelerate training by avoiding expensive clustering operations during most iterations.

Code Availability

All code used for this project, including the final training pipeline and benchmarking scripts, is available at:
https://github.com/Rwin2/pantheon-gene-panel-selection.git/gene
(Main notebook: `pantheon_gene_panel_selection/pipelines/final_project.ipynb`)

V. Conclusion

SelectSmart demonstrates strong potential as a next-generation gene-panel selection strategy. By combining the strengths of classical methods through meta-voting and refining them with a multi-agent reinforcement-learning framework, it constructs biologically meaningful panels that generalize well across datasets and contexts.

Its ability to incorporate prior knowledge, adapt to diverse data distributions, and explore a vast combinatorial space of possible panels makes it a powerful alternative to fixed analytical pipelines. Furthermore, its performance across multiple metrics highlights its robustness and versatility.

Nevertheless, several challenges remain. In particular, reward computation is currently a bottleneck due to its reliance on CPU-based clustering pipelines, and scaling the method to very large multi-cancer datasets will require faster evaluation strategies and more expressive state representations. These limitations point toward natural future directions, including GPU-accelerated reward evaluation, surrogate reward models, richer panel encodings, and extensions to pan-cancer panel design.

SelectSmart thus represents a promising step toward automated, data-driven panel design for large-scale single-cell and spatial transcriptomics, with broad applications in oncology, immunology, and integrative multi-omics.

VI. Contributions

I, Erwin Poussi, was responsible for designing the method, implementing the code, conducting experiments, and writing the report.

VII. Acknowledgement

I thank Prof. Mykel Kochenderfer for foundational training in reinforcement learning [4], and the Qiu Lab for the opportunity to apply RL to real biological problems.

Appendix: State Statistics

Let $X \in \mathbb{R}^{n_{\text{cells}} \times |G_t|}$ denote the *cell-by-gene expression matrix* restricted to the current panel G_t, where each row corresponds to a cell and each column corresponds to a selected gene. Thus, $X_{i,g}$ represents the expression level of gene g in cell i.

At each RL step, the encoder receives a fixed-size summary vector $S \in \mathbb{R}^{16}$ computed from X. These statistics are independent of the number of genes and capture global distributional structure relevant for panel evaluation.

We compute

$$S = \big[\mu_{\text{glob}},\ \sigma_{\text{glob}},\ \text{sparsity},\ q_{10},\ q_{25},\ q_{50},\ q_{75},\ q_{90},\\ \mu_\mu,\ \sigma_\mu,\ \mu_\sigma,\ \sigma_\sigma,\ s_{\text{size}},\ |G_t|,\ \max(\mu_g),\ \max(\sigma_g)\big]. \tag{16}$$

where:

- $\mu_{\text{glob}} = \frac{1}{n_{\text{cells}}|G_t|}\sum_{i,g} X_{i,g}$
- $\sigma_{\text{glob}} = \mathrm{Std}(X)$
- sparsity $= \frac{1}{n_{\text{cells}}|G_t|}\sum_{i,g} \mathbf{1}[X_{i,g} = 0]$
- $q_p = \text{quantile}(X, p)$ for $p \in \{10\%, 25\%, 50\%, 75\%, 90\%\}$
- $\mu_\mu = \mathrm{Mean}_g(\mathrm{Mean}_i(X_{i,g}))$
- $\sigma_\mu = \mathrm{Std}_g(\mathrm{Mean}_i(X_{i,g}))$
- $\mu_\sigma = \mathrm{Mean}_g(\mathrm{Std}_i(X_{i,g}))$
- $\sigma_\sigma = \mathrm{Std}_g(\mathrm{Std}_i(X_{i,g}))$
- $s_{\text{size}} = |G_t|/|\tilde{G}|$ (normalized size feature)
- $|G_t|$ = number of selected genes
- $\max(\mu_g) = \max_g \mathrm{Mean}_i(X_{i,g})$
- $\max(\sigma_g) = \max_g \mathrm{Std}_i(X_{i,g})$

References

[1] Noah M. Daniels, Yeon Park, Hung Nguyen, Sienna Fynch, Megan Zinter, Mohsin Zubair, and Chris Nolet. Accelerating single-cell data analysis with rapids. *Nature Computational Science*, 3:794–797, 2023. doi: 10.1038/s43588-023-00542-3.

[2] Bianca Dumitrascu, Soledad Villar, Dustin G. Mixon, and Barbara E. Engelhardt. Optimal marker gene selection for cell type discrimination in single cell analyses. *Nature Communications*, 9:1–14, 2018.

[3] Mahnoor N. Gondal, Marcin Cieslik, and Arul M. Chinnaiyan. Integrated cancer cell-specific single-cell rna-seq datasets of immune checkpoint blockade-treated patients. *bioRxiv [Preprint]. 2024 Apr 3:2024.01.17.576110. doi: 10.1101/2024.01.17.576110.*, 2024.

[4] Mykel J. Kochenderfer, Tim A. Wheeler, and Kyle H. Wray. *Algorithms for Decision Making*. MIT Press, 2022.

[5] Yijun Li, Stefan Stanojevic, Bing He, Zheng Jing, Qianhui Huang, Jian Kang, and Lana X. Garmire. Adding highly variable genes to spatially variable genes can improve cell type clustering performance in spatial transcriptomics data. *Genome Biology*, 23:1–16, 2022.

[6] W. G. Touw, J. R. Bayjanov, L. Overmars, L. Backus, J. Boekhorst, M. Wels, and S. A. van Hijum. Robustness of random forest-based gene selection methods. *BMC Bioinformatics*, 15(1):8, 2014. doi: 10.1186/1471-2105-15-8. URL https://doi.org/10.1186/1471-2105-15-8.

[7] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1:80–83, 1945.

[8] Fabian A Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1):15, 2018. doi: 10.1186/s13059-017-1382-0.

[9] Meng Xiao, Weiliang Zhang, Xiaohan Huang, Hengshu Zhu, Min Wu, Xiaoli Li, and Yuanchun Zhou. Knowledge-guided gene panel selection for label-free single-cell rna-seq data: A reinforcement learning perspective. *arXiv:2501.04718v2 [q-bio.GN] 11 Sep 2025*, 2025.