

A gene-level batch effect quantification method based on condition-aware correction and adaptive thresholding

Wen Li

Anhui University of Finance and Economics

Xiao Chang

chx1augh@aufe.edu.cn

Anhui University of Finance and Economics

Research Article

Keywords: condition-aware Group Technical Effects, gene-level batch effects, adaptive threshold method, highly batch-sensitive genes, single-cell transcriptomics

Posted Date: October 19th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11202430/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: No competing interests reported.

Abstract

Batch effects are one of the core challenges in single-cell transcriptomic data analysis. Existing methods mainly focus on cell-level alignment and integration across batches, with limited attention to how batch effects differ at the gene level, and gene-level tools for quantifying batch effects remain scarce. The Group Technical Effects method enables gene-level quantification of batch effects at the single-gene level, but when applied to data containing multiple biological conditions, it may misclassify condition-driven differentially expressed genes as highly batch-sensitive genes. To mitigate this limitation, we present condition-aware Group Technical Effects. This framework inserts a condition-effect disentanglement step before the original Group Technical Effects computation to separate condition effects from batch effects, produces a ConditionScore for each gene to quantify the degree to which it is driven by condition effects, and uses a multiplicative penalization strategy to correct the original Group Technical Effects score. Meanwhile, we introduce an adaptive threshold method, inspired by the model-selection principle underlying the one-standard-error rule commonly used in Least Absolute Shrinkage and Selection Operator regularization, to provide a data-driven and objective choice for the penalization strength. On a paired interferon-beta stimulation single-cell dataset, our approach substantially reduces the contribution of condition-responsive genes, such as LY6E and IRF7, to the highly batch-sensitive gene list while preserving the rank stability of ribosomal genes. In summary, condition-aware Group Technical Effects provides a correction strategy that can distinguish biological condition effects from technical noise for gene-level batch effect assessment in cross-condition single-cell data.

1.Introduction

Single-cell RNA-sequencing (scRNA-seq) provides a powerful tool for finely characterizing cellular states in disease studies, but technical variation introduced by different experimental batches is often intermingled with the biological differences of interest. Since the widespread adoption of single-cell sequencing, considerable effort has been devoted to removing batch effects to improve the reliability of downstream cross-batch analyses. Many computational methods have been developed for this purpose, ranging from early mutual nearest neighbors (MNN) and Seurat-based canonical correlation analysis to deep generative models such as scVI and scANVI, as well as more recent tools such as scPoli and scDisInFact^[1–5]. These methods correct batch effects by aligning cells in gene expression space or in low-dimensional embeddings, but they focus on cell-level integration with limited attention to how batch effects are distributed at the gene level. Leek et al. pointed out in the context of high-throughput data that batch effects do not affect all genes uniformly but disproportionately affect specific gene subsets and may affect different genes in different ways^[6]. This perspective was long underappreciated in the single-cell field until Zhou et al. proposed Group Technical Effects (GTE), which reexamined batch effects from the viewpoint of individual genes^[7]. GTE quantifies each gene's sensitivity to batch effects by comparing within-group total variation and across-batch variation within the same cell type, revealing pronounced heterogeneity of batch effects in datasets—namely, that a small subset of HBGs accounts

for the majority of total batch effect variance, and that mitochondrial and ribosomal genes consistently show high batch sensitivity across multiple datasets.

Although GTE provides a gene-level perspective for understanding batch effects, it faces an important applicability limitation in cross-condition data. In disease studies and drug-response evaluations, scRNA-seq samples typically come from donors under different disease stages, drug treatments, or genetic backgrounds; these biological conditions themselves are the objects of interest. When expression differences within the same cell type arise both from technical batch fluctuations and from systematic biological responses driven by conditions, condition-responsive genes may acquire high GTE scores because of consistent high expression under a specific condition across batches, and therefore be misclassified as HBGs. The original GTE authors explicitly noted this issue; using an IFN- β stimulated PBMC dataset as an example, they observed that identified HBGs were enriched in IFN- β activation pathways (e.g., ISG15, ISG20, IFIT1), whereas the high expression of these genes actually stems from biological stimulation rather than technical batch effects. The GTE paper also reported another observation worth noting: there is substantial overlap between HBGs and cell-type marker genes, and no gene was consistently identified as a non-HBGs across all 518 datasets examined, suggesting that any gene with biological function can potentially be misclassified as an HBGs. Similar gene-level condition-batch confounding problems exist in cross-condition studies such as drug treatment and disease progression, but to date there is no computational method specifically designed to correct this bias within the GTE framework.

To mitigate this limitation, we present the condition-aware GTE. Its purpose is not to replace GTE, but to introduce a condition-effect disentanglement step before the GTE computation in order to distinguish across-batch-consistent condition differences from across-batch-inconsistent technical fluctuations. We selected scDisInFact as the disentanglement model because it decomposes single-cell data into three independent latent spaces—shared biological factors, condition-specific biological factors, and batch factors—and because its group lasso mechanism outputs contribution weights for each gene to condition classification, providing a natural indicator for quantifying the condition-driven degree of each gene. In implementation, condition-aware GTE first uses scDisInFact to produce a ConditionScore for each gene that reflects the degree to which it is driven by condition effects, then applies a multiplicative penalization strategy to the original GTE so that genes with high ConditionScore are markedly down-weighted. To objectively determine the penalty strength, we propose an adaptive threshold method inspired by the model-selection principle underlying the 1-SE rule used in LASSO, which leverages known condition-responsive genes and known technical-control genes to set a data-driven constraint, and we validate the framework on a paired IFN- β stimulated PBMC dataset.

2. Materials and methods

2.1 Data and preprocessing:

We used the paired IFN- β stimulated human peripheral blood mononuclear cell (PBMC) scRNA-seq dataset from Kang et al. (2018) ^[8]. The dataset contains 29,065 cells from eight patients with systemic lupus erythematosus^[21]. For each donor, PBMCs were divided into an untreated control group(ctrl) and a group stimulated with IFN- β for 6 h(stim). Data were loaded using the Kang18_8vs8() function from the muscDataR package ^[9]. Key metadata variables used in this study include donor ID (ind), stimulation condition (stim), and cell type annotation (cell).

For method development and validation, we randomly sampled 3,000 genes from the full transcriptome with a fixed random seed of 20250515. The sampled gene set showed a similar distribution of mean expression levels compared to the full transcriptome, indicating no systematic bias toward highly expressed genes. Repeating sampling with different seeds produced qualitatively consistent ConditionScore distributions, adaptive threshold selection, and HBGs list correction effects.

2.2 Comparison between normalized data and count data in the GTE model:

We compared GTE values calculated from normalized data and raw count data, and found systematic bias when using count data. GTE values derived from raw counts were significantly elevated, and a small number of genes with extreme count depths dominated the whole GTE distribution. However, after removing these genes, normalized data still exhibited clear batch effects, indicating that the problem lies not in individual genes but in count data itself. This bias may arise because raw counts are strongly affected by sequencing depth and capture efficiency, which can obscure true expression differences^[10–11]. For example, ACTA2 was identified as an HBGs when using count data but as a non-HBGs with normalized data; in some cells ACTA2 had extremely high counts which inflated its GTE. Based on these observations we used normalized expression values for all subsequent analyses. Specifically, normalization was performed with the NormalizeData function from the Seurat R package: UMI counts for each cell were divided by the cell's total UMI, multiplied by 10,000, then log1p-transformed. NA values after normalization were replaced by 0. To ensure stable estimation of within-group variance in GTE, we removed cell types with fewer than 30 cells in any donor–condition combination, a threshold consistent with established guidelines for minimum sample size in single-cell variance estimation^[10].

2.3 Disentanglement of condition effects and original GTE calculation:

In cross-condition data, the original GTE framework has a known limitation: when expression differences within a cell type contain both technical batch fluctuations and systematic biological responses driven by conditions, condition effects can be incorrectly attributed to batch effects. In Kang et al.'s dataset, IFN- β stimulation induces strong transcriptional responses in multiple cell types, and this biological variation mixes with donor-level technical variation. If one computes GTE directly, IFN response genes that are consistently highly expressed under the stimulated condition will receive high GTE scores due to their cross-batch stable condition-driven differences and be misclassified as HBGs.

To separate the two, we first applied scDisInFact to disentangle condition and batch effects^[12]. This model decomposes gene expression into three independent latent spaces: shared biological factors z_s , condition-specific biological factors z_u , and batch factors b (Fig. 1A). The model training objective minimizes the following composite loss function:

$$\begin{aligned} \mathcal{L} = & -\mathbb{E}_{z_s, z_u, b} [\log P_{\Phi}(x|z_s, z_u, b)] + \lambda_1 \text{kl} \times \text{KL}(Q_{\Theta}(z_s|x, b) \parallel P(z_s)) \\ & + \lambda_2 \text{kl} \times \text{KL}(Q_{\Theta}(z_u|x) \parallel P(z_u|c)) + \lambda_{\text{mmd}} \times (L_{\text{mmd}}(z_s) + L_{\text{mmd}}(z_u)) \\ & + \lambda_{\text{ce}} \times L_{\text{ce}}(y_{\text{cond}}, \hat{y}_{\text{cond}}) + \lambda_{\text{gl}} \times L_{\text{gl}}(W) \quad (1) \end{aligned}$$

where the first term is the negative binomial reconstruction loss, constraining the model to accurately reconstruct the original expression data from the latent variables; the two KL divergence terms constrain z_s to approach the standard normal prior and z_u to approach the condition-specific Gaussian mixture prior; the MMD losses L_{mmd} enforce z_s to be independent of both batch and condition and z_u to be independent of batch, thereby achieving disentanglement; the classification cross-entropy loss L_{ce} forces z_u to retain condition-discriminative information; and the group lasso penalty L_{gl} acts on the first-layer weight matrix of the non-shared encoder to identify genes most contributing to condition classification by promoting group-level sparsity of input features.

After training, the ConditionScore for each gene m is defined as the l_2 -norm of the corresponding column vector in the first-layer weight matrix of the non-shared encoder. Under the group lasso sparsity constraint, this norm directly quantifies each gene's contribution to condition classification: genes with higher weights in the non-shared encoder are those whose expression patterns are most informative for distinguishing between conditions, and the l_2 -norm of their weight vectors thus provides a natural measure of condition-associated variation. In practice, the ConditionScore distribution shows that the vast majority of genes have scores close to zero while a few known IFN-response genes exhibit high scores. This distribution is consistent with the biological fact that IFN- β affects particular immune pathways rather than the whole transcriptome—if ConditionScore did not capture true biological signals, high scores would be randomly distributed rather than significantly enriched in known IFN-response pathways. To confirm that high-scoring genes are driven by condition rather than technical effects, we examined LY6E's expression pattern and found consistent upregulation in stimulated cells across all donors, indicating cross-batch-consistent condition differences rather than irregular batch fluctuations. These validations, detailed further in Results, indicate that ConditionScore effectively captures biological condition signals and can be used for subsequent GTE correction. Detailed model training parameters are provided in the Supplementary Materials.

Meanwhile, we computed each gene's original GTE following the original framework. Specifically, let $X_{M \times N}$ denote the normalized expression matrix. For gene m , its within-group variation $\delta_{g,m}$ and batch-within-group variation $\delta_{b,g,m}$ are defined as:

$$\delta_{g,m} = \sum_{k=1}^K \sum_{n \in gk} (1/N_k) \times (X_{m,n} - (1/N_k) \times \sum_{n \in gk} X_{m,n})^2 \quad (2)$$

$$\delta_{bg,m} = \sum_{l=1}^L \sum_{k=1}^K \sum_{n \in bl, n \in gk} (1/N_k) \times (X_{m,n} - (1/N_{l,k}) \times \sum_{n \in bl, n \in gk} X_{m,n})^2 \quad (3)$$

where $X_{m,n}$ is the normalized expression value of gene m in cell n ; N_k is the number of cells in subgroup k ; and $N_{l,k}$ is the number of cells in subgroup k within batch l . For gene m , GTE is defined as the difference between group variation and batch-group variation:

$$\delta_m = \delta_{g,m} - \delta_{bg,m} \quad (4)$$

where the group variable g is cell type and the batch variable b is donor ID. Variance decomposition interpretation and formula details for GTE follow the original publication.

2.4 Condition-aware GTE correction:

To correct the misclassification of condition-driven genes as HBGs, we introduce a multiplicative penalization strategy. The adjusted GTE for gene m is defined as:

$$\text{GTE}_{\text{adj } m} = \text{GTE}_m \times [\max(0, 1 - \text{ConditionScore}_m)]^k \quad (5)$$

where k is the penalization exponent controlling the correction strength. The statistical rationale of multiplicative penalization is grounded in variance decomposition theory. Conceptually, the total expression variance of a gene can be partitioned as:

$$\text{Var}_{\text{total}} = \text{Var}_{\text{condition}} + \text{Var}_{\text{batch}} + \text{Var}_{\text{noise}} \quad (6)$$

The ConditionScore captures the condition-driven component, so that Under the proposed empirical correction rule, a larger ConditionScore results in stronger down-weighting of the original GTE, including genuine batch effects and random noise^[13]. If ConditionScore approximately represents the fraction of variance explained by condition effects, then $(1 - \text{ConditionScore}_m)$ approximates the proportion not explained by condition. Multiplying the original GTE by the penalty factor reduces the contribution of genes with stronger condition-associated scores, which is conceptually consistent with gene-wise linear regression correction of batch effects. The $\max(0, 1 - \text{ConditionScore}_m)$ ensures non-negativity of the base: when $\text{ConditionScore}_m > 1$ the base becomes zero and the adjusted GTE is set to zero; when $\text{ConditionScore}_m \approx 0$ the adjusted GTE approximates the original GTE. This truncation is biologically justified: When ConditionScore_m exceeds 1, the penalty function sets the adjusted GTE to zero under the current formulation. This is an operational rule of the proposed correction procedure rather than evidence that the gene's expression variation is entirely attributable to the biological condition. With $k = 1$ the down-weighting is linear; with $k > 1$ suppression of high ConditionScore genes grows exponentially. The relationship between adjusted and original GTE under different k is shown in Results.

2.5 Adaptive threshold method for selecting k :

The penalization index k directly affects correction behavior: larger k more aggressively suppresses high ConditionScore genes but risks unintentionally penalizing genes driven by technical noise; smaller k may insufficiently correct condition-driven genes. This trade-off is conceptually analogous to model selection in LASSO regularization: both seek to maximize a target while constraining another metric within acceptable limits. Following the widely used 1-SE rule in LASSO^[14], we formalize k selection as a constrained optimization: maximize correction on condition-driven genes while keeping collateral damage to technical-control genes below an acceptable upper bound.

This strategy requires quantifying two key quantities: correction effect on condition-driven genes and collateral damage to technical-control genes. We therefore define two reference gene sets. The positive set, to be suppressed, comprises 18 canonical type-I IFN response genes (MX1, MX2, ISG15, IFIT1, IFIT2, IFIT3, OAS1, OAS2, OAS3, DDX58, SAMD9, HELZ2, RSAD2, IFITM1, IFITM2, IFITM3, STAT1, STAT2) that have been consistently reported as strongly induced by IFN- α/β across multiple studies^[15–16]; their rank shifts serve to measure correction strength. The negative set, to be stabilized, comprises 15 ribosomal protein genes with names starting with RPS or RPL. The original GTE study found such genes to consistently show high batch sensitivity across datasets driven mainly by technical factors like tissue dissociation and library preparation^[17–18]; their rank shifts therefore reflect collateral damage. Ideally, negative set rank shifts should be close to zero.

Acceptable upper bound. At $k = 1$, the 99th percentile of rank shifts for the negative set was 12.9. Rank shift is defined as the difference between adjusted rank and original rank. We chose $k = 1$ as the baseline because under this mild penalization only the very highest ConditionScore genes are substantially suppressed; the negative set's rank shifts then reflect baseline fluctuations introduced by the adjustment procedure rather than systematic bias. We use the 99th percentile rather than the maximum to exclude extreme outliers, consistent with tolerance-interval-style approaches.

We evaluated k values from 1 to 5. For $k = 1, 2, 3$ the negative set's average rank shifts remained within the threshold, indicating acceptable penalization. From $k = 4$ onward, the negative set's shifts exceeded the constraint and their distribution broadened markedly, indicating that over-penalization began to systematically affect ribosomal genes. Among the feasible k values, $k = 3$ maximized the average rank shift of the positive set while keeping the negative set within baseline fluctuation. Hence $k = 3$ was chosen as the optimal penalization index. Note that $k = 4$ and $k = 5$, though exceeding the adaptive threshold, produced even larger positive set shifts; researchers may flexibly choose a higher k depending on their tolerance for collateral damage.

2.6 Identification of HBGs:

HBGs identification follows the binning-cumulative strategy in the original GTE paper^[7]. Genes are sorted by GTE from high to low and divided into 20 equal-sized bins. The cumulative contribution of each bin to the total GTE is computed. The minimal set of top bins whose cumulative contribution first exceeds 95% of the total GTE is selected; genes in these bins are designated HBGs. We performed this procedure on both the original and adjusted GTE values to obtain two HBGs lists and compared them: genes present in

the original list but absent in the adjusted list were labeled "removed HBGs"; genes absent in the original but present in the adjusted list were labeled "new HBGs." Each list contained 1,200 genes; although the total number of genes remained unchanged, their composition differed—7 genes were removed and 7 new genes entered the list.

2.7 Evaluation metrics and statistical analysis:

To quantify the effect of condition-aware correction, we defined two evaluation metrics: the average rank shift of the positive set, which measures the correction strength on known condition-driven genes, and the average rank shift of the negative set, which measures the collateral damage to technical-control genes. A larger average shift in the positive set indicates stronger suppression of condition-driven genes; a smaller average shift in the negative set indicates better specificity of the correction. At $k = 3$, the positive set's average shift was much larger than that of the negative set; a Wilcoxon rank-sum test yielded $P = 0.0147$, indicating a statistically significant difference. This result shows that the correction selectively suppresses condition-driven genes while keeping the impact on technical-control genes within baseline fluctuation.

The proportion of IFN-pathway genes among the removed HBGs contrasted sharply with the genome-wide random expectation, further supporting the specificity of the correction. Gene Ontology (GO) enrichment analysis of the newly added HBGs highlighted protein stabilization and regulation of protein stability as the most significant terms, processes unrelated to IFN signaling. All GO enrichments used the 3,000 randomly sampled genes as the background^[19]; P-values were adjusted by the Benjamini–Hochberg method with a significance threshold of adjusted $P < 0.05$ ^[20].

2.8 Implementation and visualization:

All analyses were performed in R and Python. scDisInFact model training used the scDisInFact Python package; original GTE computation used the GTEs R package. Model training parameters are provided in the Supplementary Materials.

3. Results

3.1 Model overview:

The core problem that condition-aware GTE addresses is that, in cross-condition single-cell data, the original GTE can misclassify condition-driven differentially expressed genes as HBGs. To address this, we introduce a disentanglement learning model that separates condition effects from batch effects prior to computing GTE. We applied scDisInFact to the Kang et al. (2018) paired IFN- β stimulated PBMC dataset, which contains 29,065 cells from 8 donors with each donor's PBMC split into control and IFN- β stimulated groups (Fig. 1A,B). From the full transcriptome we randomly sampled 3,000 genes for analysis and computed a ConditionScore for each gene reflecting the degree to which it was driven by IFN- β stimulation. The ConditionScore distribution was strongly right-skewed (Fig. 1C): most genes

scored near zero while a small number scored substantially higher. LY6E and IRF7 had ConditionScores of 1.38 and 0.84 respectively, falling in the extreme right tail of the distribution. We selected these two genes as representatives because they are canonical ISGs and because their differing scores allow us to examine how different magnitudes of condition effects influence correction. To further validate that high-ConditionScore genes are indeed driven by biological conditions, we examined LY6E expression across conditions and donors and found LY6E to be consistently highly expressed in stimulated cells across all 8 donors and low in control cells (Fig. 1D). This cross-donor consistency supports the interpretation that LY6E expression is primarily associated with the stimulation condition in this dataset that the expression variation of high-ConditionScore genes is dominated by condition effects rather than technical batch effects, which is precisely the signal ConditionScore aims to capture.

3.2 Biological validation of ConditionScore:

The above analysis suggests that ConditionScore can identify genes consistently upregulated under stimulation, but further validation is needed to confirm that top-scoring genes are biologically related to IFN- β response. A heatmap of the top 20 ConditionScore genes shows consistently high expression in stimulated cells, forming a clear condition-response module where known ISGs such as LY6E, ISG20, and IRF7 rank among them (Fig. 2A). We then compared the ConditionScore distribution of the positive set of 18 known ISGs to random genes and found the positive set's median ConditionScore to be significantly higher than that of random genes (Fig. 2B), indicating the model prioritizes known condition-responsive genes. Functionally, GO enrichment on the top 200 ConditionScore genes revealed the most significant terms were response to virus and defense response to virus (Fig. 2C), central downstream programs of type I IFN signaling and consistent with IFN- β stimulation biology. Correspondingly, IRF7, ISG20, and LY6E all show substantial upregulation in stimulated cells relative to controls, with their expression levels in stimulated cells far exceeding those in controls (Fig. 2D). Together, these results indicate that ConditionScore effectively captures biological condition signals and that high-scoring genes are members of the IFN- β -activated transcriptional program. Having obtained a per-gene metric of condition-driven degree, the next question is how to use it to correct raw GTE.

3.3 Adaptive threshold method for selecting penalization index k :

With ConditionScore available, we next address how to integrate it into GTE adjustment. The correction strength is controlled by a penalization index k : larger k more aggressively suppresses high-ConditionScore genes but may also affect true technical-control genes; smaller k could be insufficient. To choose k objectively, we developed an adaptive threshold method inspired by the LASSO 1-SE rule, formalizing k selection as a constrained optimization: maximize penalization on condition-driven genes while keeping collateral penalty to technical-control genes below an acceptable limit.

Practically, we used 18 known ISGs as the positive set representing genes to be suppressed and ribosomal protein genes as the negative set representing genes that should not be down-weighted.

Ribosomal genes were chosen because the original GTE study found them to be consistently highly batch-sensitive across datasets. At $k = 1$, the 99th percentile of rank shifts for the negative set was 12.9 (Fig. 3A), meaning that under a mild penalty of $k = 1$, 99% of ribosomal genes' rank shifts do not exceed 12.9 positions. We used the 99th percentile rather than the maximum to reduce the influence of extreme outliers. We evaluated k from 1 to 5 and found that the negative set's rank shift distribution remained narrow for $k \leq 3$ but broadened markedly for $k \geq 4$ (Fig. 3B), indicating that excessive penalization begins to systematically perturb ribosomal gene rankings—genes that should remain stable under ideal correction. The negative set's mean rank shift stayed within 12.9 for $k = 1, 2, 3$ but exceeded it for $k = 4$ and 5 (Fig. 3C). Following constrained optimization, we selected $k = 3$ among feasible values because it maximized the positive set's average rank shift (870.2) while keeping negative-set shifts within the baseline fluctuation level; $k = 1$ and $k = 2$ only achieved 534.2 and 692.0 respectively. Consistently, ranks of key IFN genes (LY6E, IRF7, ISG20) monotonically increase with k (Fig. 3D). Note that the ranks of LY6E and ISG20 shifted toward the lower end of the ranking (approximately 3,000) at $k = 1$ and remained stable thereafter because both genes have ConditionScore > 1 , hence were already fully suppressed under mild penalization; IRF7's lower ConditionScore required larger k for substantial rank changes.

3.4 Condition-aware GTE removes misclassified HBGs:

With $k = 3$ selected, we tested whether condition-aware GTE indeed removes IFN-response genes misidentified as HBGs. Compared with the original GTE HBGs list, the adjusted GTE removed 7 genes; LY6E moved from rank 13 to rank 2,986, corresponding to a 2,973-position downward movement in the ranking, ISG20 moved downward by 2,955 positions, and IRF7 moved downward by 1,405 positions (Fig. 4A). These three genes' rank drops were much larger than those of other removed genes such as GPR146, TAF6L, and TMEM41A, because genes with higher ConditionScores are more strongly penalized by multiplicative down-weighting. Among the 7 removed genes, LY6E, IRF7, ISG20, GPR146, TMEM41A, and TAF6L belong to the IFN pathway (Fig. 4C), accounting for the majority of removed genes and far exceeding random expectation, indicating that the correction specifically targets condition-driven genes. The only exception was MTG2, a mitochondrial ribosome-related GTPase whose functional category is closely related to the negative ribosomal control set; MTG2's rank in the original HBGs list was near the boundary, so its removal may reflect natural boundary fluctuation under adjustment.

A scatter plot of original versus adjusted GTE further supports these observations: most genes lie near the diagonal, indicating minimal change in GTE; ISG positive-set genes and removed HBGs lie clearly below the diagonal, with adjusted GTE values penalized to near zero (Fig. 4B), demonstrating selective suppression of high-ConditionScore genes. While 7 genes were removed, 7 new genes entered the HBGs list; among them, USP36 participates in ribosomal RNA processing and functions as a deubiquitinase, and other new HBGs relate to ER–Golgi transport, cell adhesion, and ion channel activity (Fig. 4D). The appearance of ribosome-related genes among newly added HBGs is consistent with the original GTE finding that ribosomal genes are among the most batch-sensitive, indicating that the corrected HBGs list replaces condition-driven genes with genes that are more consistently associated with technical variation under the evaluated dataset and analysis settings.

3.5 Method specificity validation:

The previous results show that condition-aware GTE can remove IFN-response genes from the HBGs list, but equally important is whether the correction causes collateral penalty to true technical-control genes. To answer this, we examined the negative set under different k values. The average rank shift of ribosomal genes remained low when $k \leq 3$, but began to rise significantly for $k \geq 4$ (Fig. 5A), a trend consistent with adaptive threshold choice and indicating that $k = 3$ is just before the inflection point where negative-control shifts accelerate. At $k = 3$, ribosomal genes lie almost exactly on the diagonal in the scatter plot of original versus adjusted ranks (Fig. 5B), with RPL23 and RPL28 showing very small changes, indicating that the adjustment does not erroneously penalize housekeeping genes.

We further compared the rank-shift distributions of the positive and negative sets at $k = 3$ (Fig. 5C). The positive set's shift magnitude is much larger than the negative set's: the positive set's median shift exceeds 300 ranks while the negative set's median is near zero; the difference is statistically significant. This contrast supports the method's ability to distinguish condition-driven genes from technical-control genes with specificity. This asymmetry stems from the multiplicative down-weighting formula: ribosomal genes have ConditionScores close to zero, so their adjusted GTE nearly equals the original GTE and their ranks remain stable. Collectively, these results show that condition-aware GTE achieves its design goal of selectively correcting misidentified condition-driven HBGs while maintaining stable evaluation for technical-control genes.

4. Discussion

In summary, we present condition-aware GTE, a framework that incorporates condition-effect disentanglement into gene-level batch effect evaluation for cross-condition single-cell data. Unlike the original GTE, our method distinguishes biological condition-associated variation from technical batch-driven variation, thereby reducing the risk of misclassifying key condition-responsive genes as HBGs. It is important to note that our condition-aware strategy is not a data integration or batch correction technique; rather, it should be viewed as a diagnostic tool for gene screening before downstream analyses such as dimensionality reduction or visualization. For downstream tasks where preserving original expression levels is vital, the raw data should still be utilized. We also observed that the choice of k reflects different analytical goals: when prioritizing specificity, $k = 3$ is more conservative and prevents over-correction; when maximizing biological discovery, a higher k may reveal subtle cellular heterogeneity previously obscured by condition effects. We anticipate that this framework will provide a more targeted gene-selection strategy for single-cell analyses across multiple biological conditions.

Despite achieving its design goals, several limitations warrant discussion. First, the adaptive threshold method prioritizes controlling false positives in the negative set, which may result in a conservative correction that under-corrects condition-driven genes in some datasets. Second, the definition of the negative control set relies on the assumption that the batch sensitivity of ribosomal genes is driven primarily by technical factors; however, under specific biological contexts, ribosomal expression might

be biologically perturbed, potentially underestimating the threshold and affecting k selection. Third, our validation was restricted to a single dataset representing IFN- β stimulation and leveraged a random gene subset to manage computational costs, its generalizability across broader datasets remains to be fully validated. Future work could extend this framework to multi-condition datasets with more complex experimental designs and evaluate its performance across whole-genome scales.

5. Conclusions

We developed condition-aware GTE to reduce the misclassification of condition-responsive genes as highly batch-sensitive genes in cross-condition single-cell RNA-sequencing data. The framework combines condition-effect disentanglement, gene-level scoring, and adaptive penalization of the original GTE. In the evaluated IFN- β -stimulated PBMC dataset, the method selectively reduced the rankings of condition-responsive genes while preserving the rankings of ribosomal technical controls. However, the current results represent a proof of concept based on a single dataset and should be further validated across diverse conditions, tissues, and experimental designs.

Abbreviations

GTE

Group Technical Effects

HBGs

highly batch-sensitive genes

PBMC

peripheral blood mononuclear cell

IFN

interferon

ISG

interferon-stimulated gene

scRNA-seq

single-cell RNA sequencing

GO

Gene Ontology

MMD

maximum mean discrepancy

KL

Kullback-Leibler

LASSO

least absolute shrinkage and selection operator.

Declarations

Ethics approval and consent to participate:

This study used publicly available, de-identified single-cell RNA sequencing data from the GSE96583 dataset^[21]. The original study reported that informed consent was obtained from all participants whose samples were sequenced. No new human participants were recruited or sampled for the present study, and no additional human-subject research was conducted.

Author contributions:

Wen Li: Conceptualization, Methodology, Software, Formal analysis, Data curation, Visualization, Writing – original draft, Writing – review & editing.

Xiao Chang: Supervision, Writing – review & editing.

Abbreviations

GTE: Group Technical Effects; HBGs: highly batch-sensitive genes; PBMC: peripheral blood mononuclear cell; IFN: interferon; ISG: interferon-stimulated gene; scRNA-seq: single-cell RNA sequencing; GO: Gene Ontology; MMD: maximum mean discrepancy; KL: Kullback-Leibler; LASSO: least absolute shrinkage and selection operator.

Data Availability:

The Kang et al. PBMC single-cell RNA sequencing dataset used in this study is publicly available through the Gene Expression Omnibus under accession number GSE96583^[21]. The dataset can also be accessed through the Kang18_8vs8() function in the muscData R package^[9]. The 3,000 randomly sampled gene subset and corresponding metadata generated for this study are available at GitHub (<https://github.com/a-556-a/-git>).

Code Availability:

All analysis code used in this study is available at GitHub (<https://github.com/a-556-a/-git>).

Declaration of generative AI and AI-assisted technologies in the manuscript

preparation process:

During the preparation of this manuscript, the author used DeepSeek for language editing, translation, and manuscript formatting. After using this tool, the author reviewed and edited the content as necessary and take full responsibility for the content of the published article. DeepSeek was not used for data analysis or figure generation.

Declaration of competing interests:

The authors declare that they have no competing interests.

Funding:

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

1. Haghverdi L, Lun ATL, Morgan MD, Marioni JC. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat Biotechnol.* 2018;36:421–7. <https://doi.org/10.1038/nbt.4091>.
2. Butler A, Hoffman P, Smibert P, Papalexi E, Satija R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat Biotechnol.* 2018;36:411–20. <https://doi.org/10.1038/nbt.4096>.
3. Lopez R, Regier J, Cole MB, Jordan MI, Yosef N. Deep generative modeling for single-cell transcriptomics. *Nat Methods.* 2018;15:1053–8. <https://doi.org/10.1038/s41592-018-0229-2>.
4. Xu C, Lopez R, Mehlman E, Regier J, Jordan MI, Yosef N. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Mol Syst Biol.* 2021;17. <https://doi.org/10.15252/msb.20209620>.
5. De Donno C, Hediye-Zadeh S, Moinfar AA, Wagenstetter M, Zappia L, Lotfollahi M, et al. Population-level integration of single-cell datasets enables multi-scale analysis across samples. *Nat Methods.* 2023;20:1706–15. <https://doi.org/10.1038/s41592-023-02035-2>.
6. Leek JT, Scharpf RB, Corrada Bravo H, Simcha D, Langmead B, Johnson WE, et al. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat Rev Genet.* 2010;11:733–9. <https://doi.org/10.1038/nrg2825>.
7. Zhou Y, Sheng Q, Wang G, Xu L, Jin S. Quantifying batch effects for individual genes in single-cell data. *Nat Comput Sci.* 2025;5:612–20. <https://doi.org/10.1038/s43588-025-00824-7>.
8. Kang HM, Subramaniam M, Targ S, Nguyen M, Maliskova L, McCarthy E, et al. Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. *Nat Biotechnol.* 2018;36:89–94. <https://doi.org/10.1038/nbt.4042>.
9. Crowell H. muscData: multi-sample multi-group scRNA-seq data, R package version 1.1.2. Bioconductor; 2020. <https://bioconductor.org/packages/muscData/>. Accessed 29 Sep 2026.
10. Vallejos CA, Risso D, Scialdone A, Dudoit S, Marioni JC. Normalizing single-cell RNA sequencing data: challenges and opportunities. *Nat Methods.* 2017;14:565–71. <https://doi.org/10.1038/nmeth.4292>.
11. Cole MB, Risso D, Wagner A, DeTomaso D, Ngai J, Purdom E, et al. Performance assessment and selection of normalization procedures for single-cell RNA-seq. *Cell Syst.* 2019;8:315–e288. <https://doi.org/10.1016/j.cels.2019.03.010>.
12. Zhang Z, Zhao X, Qiu P, Zhang X. scDisInFact: disentangled learning for integration and prediction of multi-batch multi-condition single-cell RNA-sequencing data. *Nat Commun.* 2024;15:1187.

<https://doi.org/10.1038/s41467-024-45227-w>.

13. Hoffman GE, Schadt EE. variancePartition: interpreting drivers of variation in complex gene expression studies. *BMC Bioinformatics*. 2016;17:483. <https://doi.org/10.1186/s12859-016-1323-z>.
14. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. *J Stat Softw*. 2010;33:1–22. <https://doi.org/10.18637/jss.v033.i01>.
15. Schneider WM, Chevillotte MD, Rice CM. Interferon-stimulated genes: a complex web of host defenses. *Annu Rev Immunol*. 2014;32:513–45. <https://doi.org/10.1146/annurev-immunol-032713-120231>.
16. Schoggins JW, Wilson SJ, Panis M, Murphy MY, Jones CT, Bieniasz P, et al. A diverse range of gene products are effectors of the type I interferon antiviral response. *Nature*. 2011;472:481–5. <https://doi.org/10.1038/nature09907>.
17. Denisenko E, Guo BB, Jones M, Hou R, de Kock L, Lassmann T, et al. Systematic assessment of tissue dissociation and storage biases in single-cell and single-nucleus RNA-seq workflows. *Genome Biol*. 2020;21:130. <https://doi.org/10.1186/s13059-020-02048-6>.
18. van den Brink SC, Sage F, Vértessy Á, Spanjaard B, Peterson-Maduro J, Baron CS, et al. Single-cell sequencing reveals dissociation-induced gene expression in tissue subpopulations. *Nat Methods*. 2017;14:935–6. <https://doi.org/10.1038/nmeth.4437>.
19. Gene Ontology Consortium. The Gene Ontology resource: enriching a GOld mine. *Nucleic Acids Res*. 2021;49–34. <https://doi.org/10.1093/nar/gkaa1113>.
20. Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J R Stat Soc B*. 1995;57:289–300. <https://doi.org/10.1111/j.2517-6161.1995.tb02031.x>.
21. Gene Expression Omnibus. GSE96583: single-cell RNA-seq of human peripheral blood mononuclear cells stimulated with interferon beta. <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE96583>. Accessed 29 Sep 2026.

Figures

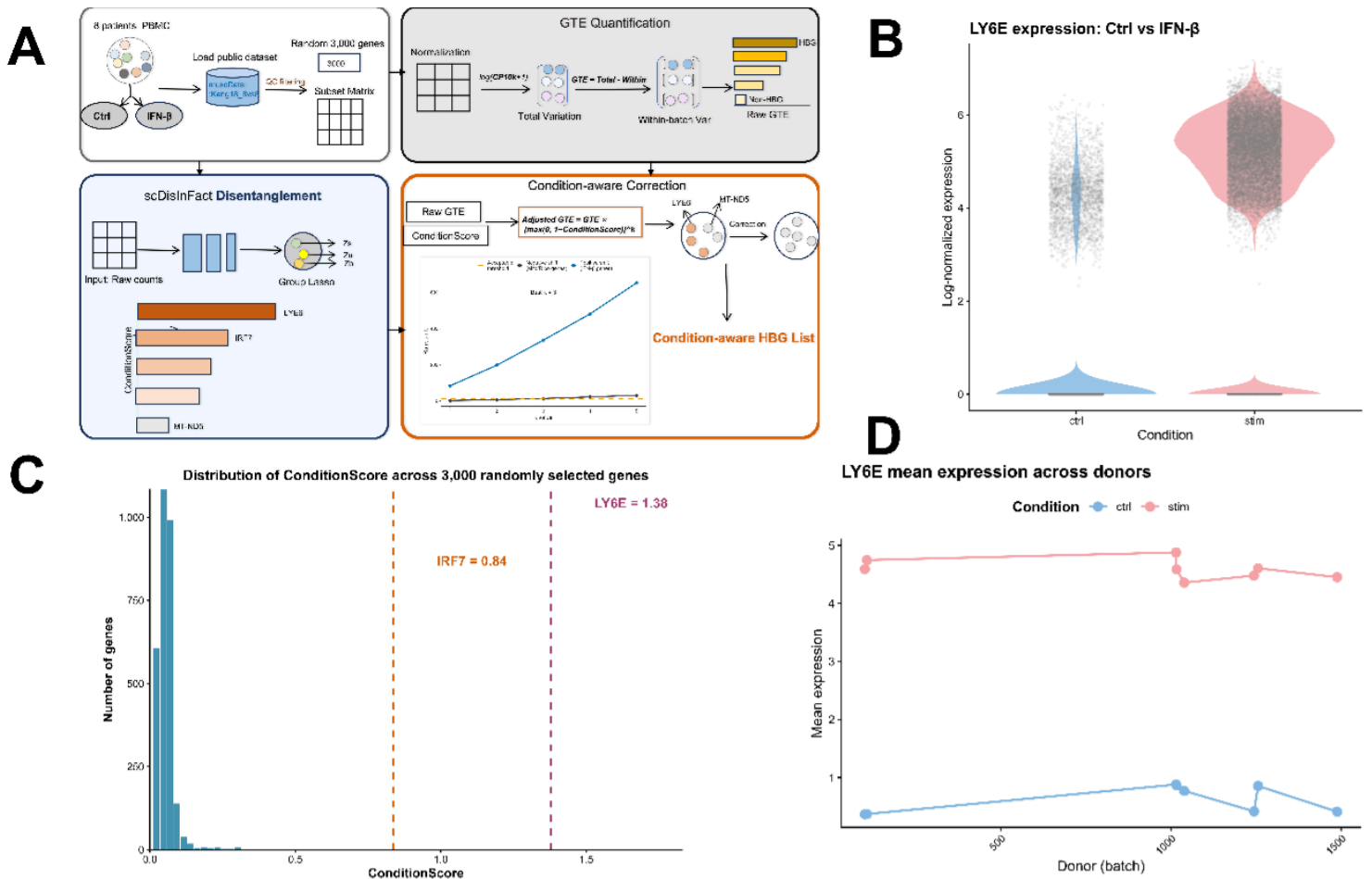


Figure 1

Condition-aware GTE: A, workflow of condition-aware GTE. B, schematic of the paired experimental design. C, distribution of ConditionScores for 3,000 random genes, dashed lines marking IRF7 (0.84) and LY6E (1.38). D, LY6E expression in control and stimulated conditions.

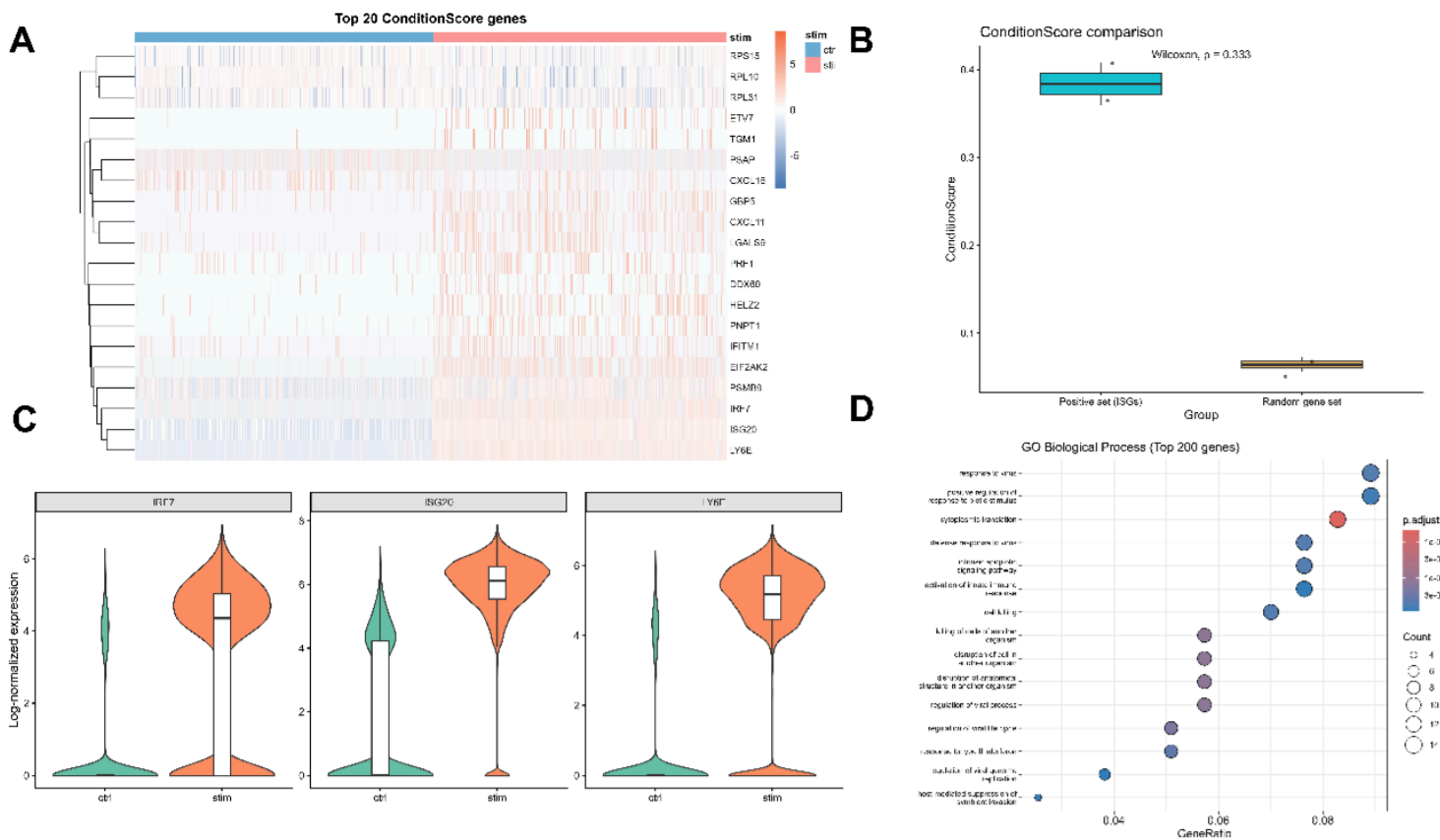


Figure 2

Biological validation of ConditionScore. A, Heatmap of the top 20 ConditionScore genes. B, Comparison of ConditionScore distributions between the positive ISG set and random genes. C, GO enrichment analysis of the top 200 ConditionScore genes. D, Expression of IRF7, ISG20, and LY6E in control and stimulated cells.

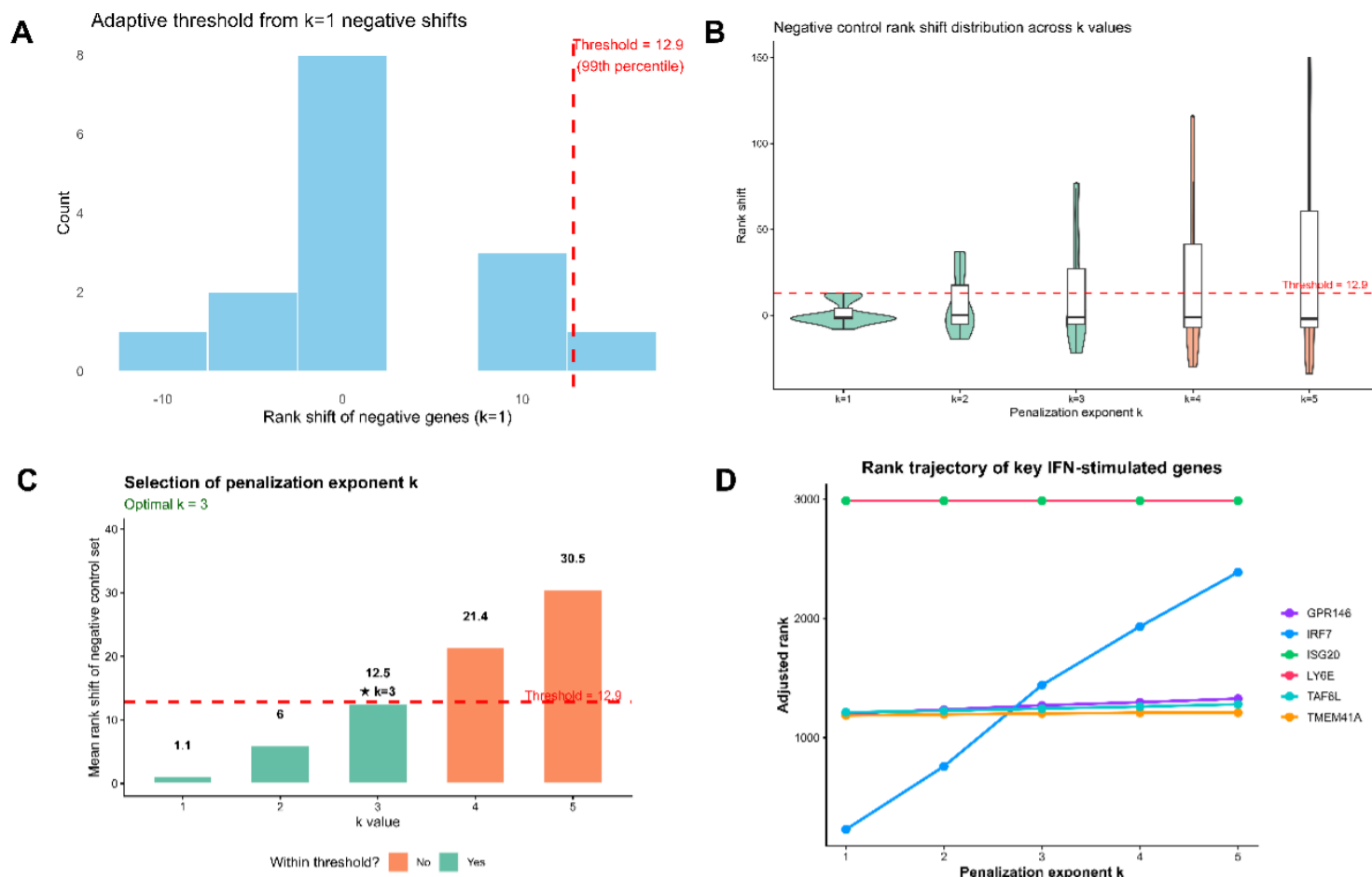


Figure 3

Adaptive threshold selection of k . A, Distribution of negative-control gene rank shifts at $k = 1$ and the 99th percentile threshold (12.9). B, Distributions of negative-control rank shifts across k values. C, Summary of average negative-control rank shifts by k . D, Rank trajectories of key IFN genes across k values.

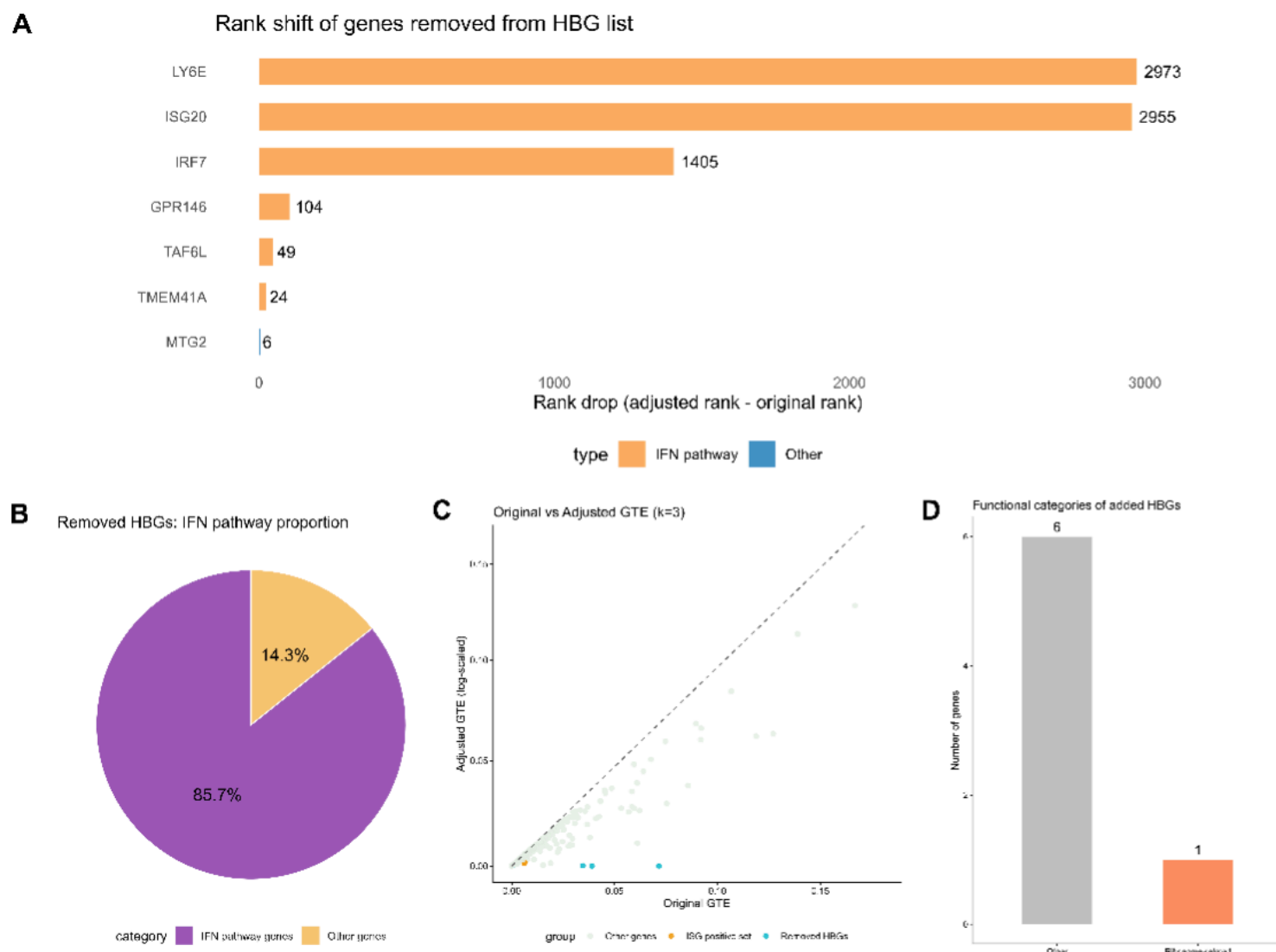


Figure 4

Condition-aware GTE removes misidentified HBGs. A, Waterfall plot of rank drops for removed genes. B, Scatter plot of original versus adjusted GTE. C, Proportion of IFN-pathway genes among removed HBGs. D, Functional classification of newly added HBGs.

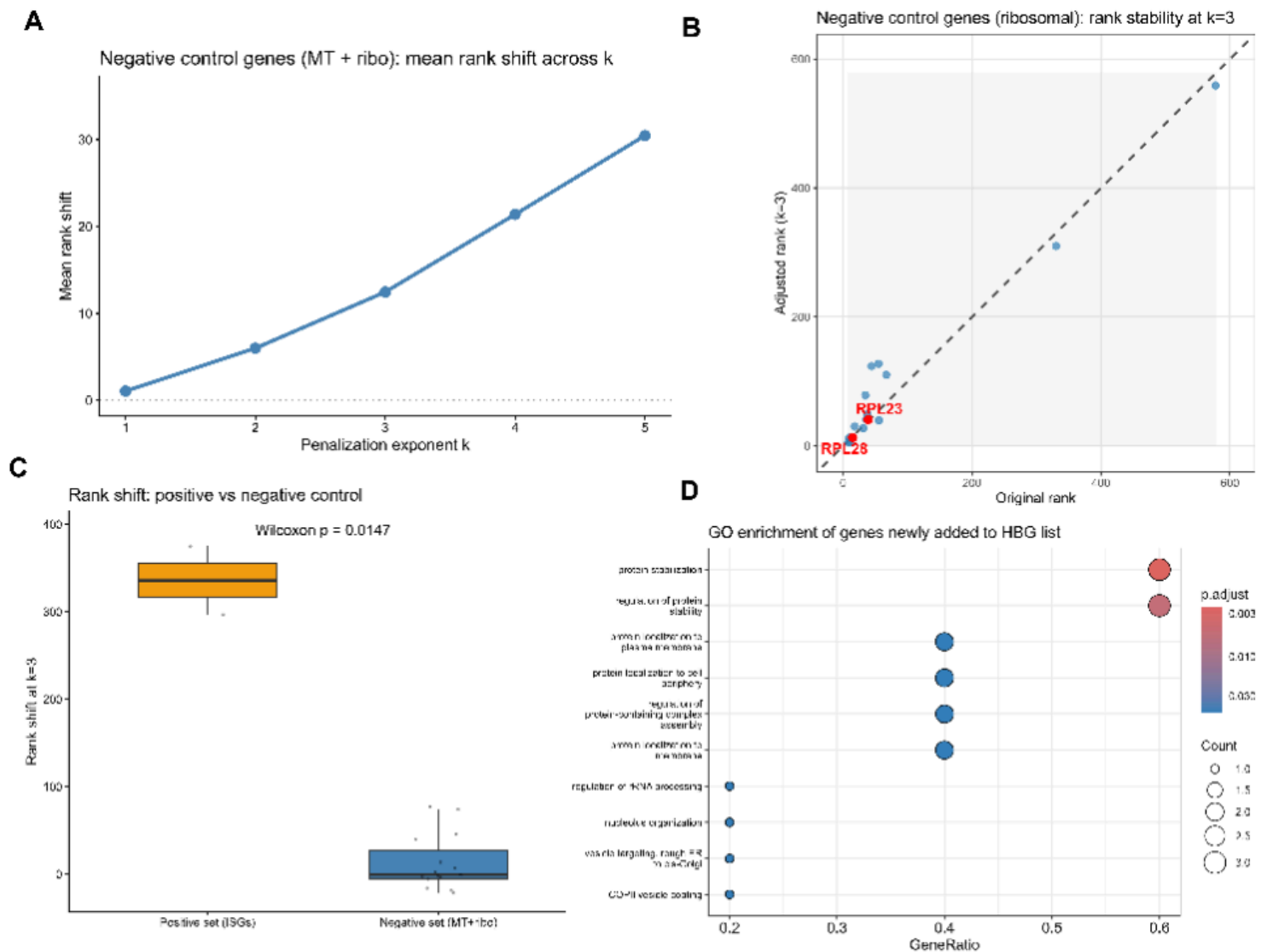


Figure 5

Specificity validation: stability of technical-control genes. A, Average rank shifts of negative-control genes across k values. B, Scatter plot of original versus adjusted ranks for negative-control genes at k = 3, with RPL23 and RPL28 highlighted in red. C, Comparison of rank shifts between positive and negative sets at k = 3. D, GO enrichment analysis of newly added HBGs.