Survival driven deconvolution (deSurv) reveals prognostic and interpretable cancer subtypes

Amber M. Young[a,1,2], **Alisa Yurovsky**[b], **Didong Li**[a], **and Naim U. Rashid**[a,c]

[a]University of North Carolina at Chapel Hill, Biostatistics, Street, City, State, Zip; [b]Stony Brook University, Street, City, State, Zip

This manuscript was compiled on January 19, 2026

Molecular subtyping in cancer is an ongoing problem that relies on the identification of robust and replicable gene signatures. While transcriptomic profiling has revealed recurrent gene expression patterns in various types of cancer, the prognostic value of these signatures is typically evaluated in retrospect. This is due to the reliance on unsupervised learning methods for identifying cell-type-specific signals and clustering patients into molecular subtypes. Here we present a Survival-driven Deconvolution tool (deSurv) that integrates bulk RNA-sequencing data with patient survival information to identify cell-type-enriched gene signatures associated with prognosis. Applying deSurv to various cohorts in pancreatic cancer, we uncover prognostic and biologically interpretable subtypes that reflect the complex interactions between stroma, tumor, and immune cells in the tumor microenvironment. Our approach highlights the value of using patient outcomes during gene signature discovery.

one | two | optional | optional | optional

Molecular subtyping has transformed precision oncology by stratifying patients into biologically and clinically meaningful groups that inform prognosis and guide therapy (1–5). Subtyping relies on the identification of robust biological signals that define subtypes such as transcriptomic signatures. However, the tumor microenvironment (TME) contains mixtures of diverse cell types such as malignant, stromal, immune, and endothelial cells, and disentangling tumor specific signals from this mixture can be challenging. As such, subtyping pipelines typically rely on the deconvolution of bulk transcriptomic data or single-cell analysis to discover distinct cell types and their corresponding signatures. Downstream, the signatures are evaluated for clinical relevance such as overall survival or response to treatment.

Separating discovery from validation can risk overfitting and limits biological and clinical generalizability. Identified cell types may capture dataset-specific noise rather than reproducible biological signals, undermining their utility in downstream analyses or therapeutic targeting (6, 7). Moreover, even when discovered cell types are biologically valid and reproducible, they may not correspond to the cellular programs most relevant for predicting or influencing clinical outcomes (8, 9). Therefore, there is a clear need for integrative methods that jointly uncover biologically meaningful programs while directly incorporating clinical endpoints to ensure prognostic relevance.

However, integrating patient outcomes into the discovery phase is not straightforward with current technology and methodology. Single-cell transcriptomics can resolve programs at the cellular level, but cohort sizes are often too small to support survival analyses. In contrast, large bulk transcriptomic cohorts with clinical annotations are well-suited for outcome modeling (10, 11), yet deconvolution is needed to disentangle overlapping cellular signals. Reference-based deconvolution methods focus on estimating cell-type proportions from predefined signatures, which limits the utility of these methods for discovery of novel programs (12).

Nonnegative matrix factorization (NMF) is widely used in cancer genomics because its nonnegativity constraints produce biologically interpretable, additive molecular programs (13–16). Although recent extensions have incorporated supervision into the factorization, most target regression or classification rather than time-to-event outcomes. Two studies have proposed survival-aware NMF formulations (17, 18), but both integrate the survival objective through the sample-specific loadings rather than the gene-level programs. This design emphasizes prediction accuracy but limits the model's ability to restructure or refine the underlying molecular programs, reducing its value for biological interpretation and subtype discovery, which are core objectives in cancer transcriptomics. In addition, neither study provides a principled approach for hyperparameter selection or model assessment, and convergence properties are only briefly addressed in one manuscript. Both works remain unpublished and unreviewed, leaving their methodological robustness and reproducibility uncertain. These gaps highlight the need for a rigorously formulated, survival-aware deconvolution method that jointly estimates interpretable molecular programs and their prognostic relevance.

Here we present DeSurv, a Survival-supervised Deconvolution framework that integrates non-negative matrix factor-

Significance Statement

Tumor transcriptomes reflect mixtures of malignant and microenvironmental cell populations, making it challenging to identify the molecular programs that truly drive clinical outcomes. Existing deconvolution and matrix factorization methods discover latent transcriptional programs but do not ensure that these programs are prognostic, while supervised extensions optimized for prediction offer limited biological interpretability. We present DeSurv, a survival-supervised deconvolution framework that integrates nonnegative matrix factorization with Cox modeling to jointly learn biologically coherent gene programs and their associations with patient survival. By embedding outcome information directly into the discovery process and performing automatic model selection, DeSurv reveals clinically relevant transcriptional programs that are reproducible across cohorts. This approach advances the statistical foundations of tumor deconvolution and provides a general tool for identifying actionable molecular drivers of disease progression.

Please provide details of author contributions here.

Please declare any conflict of interest here.

[2] To whom correspondence should be addressed. E-mail: ayoung31@live.unc.edu

ization (NMF) with Cox proportional hazards modelling. In contrast to fully unsupervised approaches that evaluate survival associations only after the factorization, and to existing supervised NMF models that link outcomes to the subject-level factor loadings, DeSurv integrates survival information directly into the gene signature matrix. This design ensures that the discovered transcriptional programs are not only biologically interpretable but also intrinsically aligned with patient outcomes. To enhance robustness and reproducibility, DeSurv performs automatic parameter selection via Bayesian optimization, addressing the quintessential challenge of rank determination in matrix factorization.

By coupling latent program discovery with direct survival supervision, DeSurv resolves longstanding challenges in disentangling tumor–microenvironment interactions and aligns molecular heterogeneity with clinical outcomes. This unified approach represents a methodological advance in translational cancer genomics and provides a general framework for deriving actionable insights from high-dimensional transcriptomic data.

Results

Model Overview. We have developed an integrated framework, DeSurv, that couples Nonnegative Matrix Factorization (NMF) with Cox proportional hazards regression to identify latent gene-expression programs associated with patient survival (Figure @ref(fig:fig-schema)). The model takes as input a bulk expression matrix of p genes by n patients (X_{Train}) together with corresponding survival times (y_{Train}) and censoring indicators (δ_{Train}) (Figure @ref(fig:fig-schema)A).

DeSurv optimizes a joint objective combining the NMF reconstruction loss and the Cox model's log-partial likelihood, weighted by a supervision parameter (α) that determines the relative contribution of each term (Figure @ref(fig:fig-schema)B):

$$\begin{aligned} &(1-\alpha)\ \mathcal{L}_{NMF}(X_{Train} \approx WH) \\ &\quad -\alpha\ \mathcal{L}_{Cox}(X^{T}_{Train}W\beta, y_{Train}, \delta_{Train}) \end{aligned} \tag{1}$$

When $\alpha = 0$, the method reduces to standard unsupervised NMF; when $\alpha > 0$, survival information directly guides the learned factors toward prognostic structure.

Within this framework, the product ($X^{T}_{Train}W$) represents patient-level factor scores - the inferred burden of each latent program across subjects. These factor scores serve as covariates in the Cox model, and their regression coefficients (β) indicate whether higher activity of a given program corresponds to improved or reduced survival.

Model training yields gene weights ($\hat{W}$), factor loadings ($\hat{H}$), and Cox coefficients ($\hat{\beta}$) (Figure @ref(fig:fig-schema)C), where the inner dimension (k) specifies the number of latent factors. Genes with high gene weights in one factor and low gene weights in all others define the factor-specific signature genes (Figure @ref(fig:fig-schema)D). By integrating survival supervision into the factorization, DeSurv not only reconstructs the underlying expression structure, preserving biologial interpretability, but also guides latent factors to be prognostically informative. Subsequent analyses can therefore focus on the survival-associated gene programs (Figure @ref(fig:fig-schema)E).

Bayesian optimisation selects the DeSurv hyperparameters (k, α). We now tune DeSurv with Bayesian optimisation (BO) rather than enumerating a dense cross-validation grid. Each BO evaluation fits the model for a single combination of k, α, and penalty weights, measuring the mean validation C-index. A Gaussian-process surrogate models this surface and proposes new evaluations that balance exploration and exploitation, allowing the search to concentrate rapidly on high-performing regions.

Figure @ref(fig:fig-bo) summarises the optimisation run. Panel A reports the best observed cross-validated C-index at each k for the supervised and $\alpha = 0$ searches, with error bars denoting the cross-validation standard error across folds. The final configuration of the supervised search converges to $k = 3$ and $\alpha = 0.7$, which provided the highest cross-validated C-index encountered by BO.

For reference we still report standard NMF heuristics (Panels B-D: cophenetic, residuals, and silhouette), which historically guide rank selection when a supervised criterion is unavailable. These diagnostics disagree on the optimal k, underscoring the benefit of the BO-guided DeSurv procedure that explicitly maximises prognostic accuracy.

DeSurv captures prognostic gene signatures. To isolate how supervision shapes the recovered gene programs, we highlight the BO-tuned `n_top` simulations in two contrasting regimes. The R0_easy setting contains a clear survival signal, whereas R00_null contains no prognostic structure by design. Figure @ref(fig:fig-sim-ntop-scenarios) shows that in R0_easy, DeSurv attains higher test C-index and improved precision for lethal genes relative to the $\alpha = 0$ baseline, indicating that supervision concentrates the signatures on true survival drivers. In the null setting, performance compresses toward chance for both methods, suggesting that DeSurv does not manufacture prognostic structure when none is present.

DeSurv improves selection of prognostic gene signatures. To test whether supervision improves the selection of prognostic gene signatures, we used simulations with known lethal factors and tuned the number of top genes per factor (`n_top`) using BO. We compared the supervised DeSurv model to the unsupervised $\alpha = 0$ baseline across simulation regimes. Figure @ref(fig:fig-sim-ntop)A shows the distribution of test C-index values, while Figure @ref(fig:fig-sim-ntop)B reports the precision of recovered prognostic genes (mean across lethal factors). DeSurv consistently yields higher C-index and improved precision, indicating that survival supervision helps focus signatures on truly prognostic genes rather than reconstruction-only structure.

DeSurv provides biologically interpretable gene signatures. For the selected DeSurv model ($k = 3, \alpha = 0.7$), we identified the top 50 factor-specific genes from each latent factor to characterize their underlying biological functions. Panels A-C of Figure @ref(fig:fig-bio) summarize the results of an over-representation analysis (ORA) performed on these genes. The enrichment profiles reveal three distinct biological programs. Factor 1 is strongly enriched for immune-related pathways, including cytokine signaling, T-cell activation, and interferon response. Factor 2 shows enrichment for genes associated with normal pancreatic or exocrine function, suggesting that this component captures residual normal-tissue signal. Factor 3 is enriched for pathways linked to epithelial–mesenchymal transition, cell cycle progression, and KRAS signaling—features

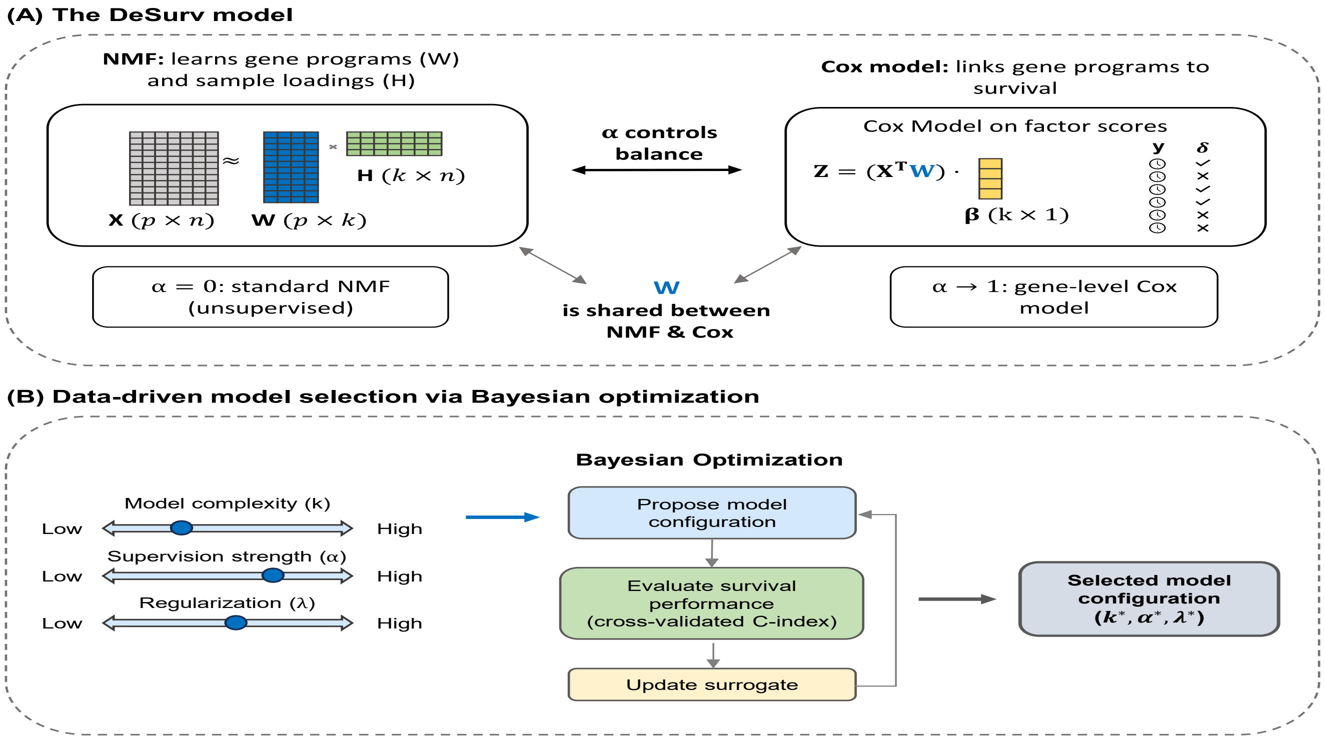
(A) The DeSurv model
NMF: learns gene programs (W) and sample loadings (H)
≈
H (k × n)
X (p × n)
W (p × k)
α controls balance
Cox model: links gene programs to survival
Cox Model on factor scores
Z = (XᵀW) ·
β (k × 1)
y
δ
α = 0: standard NMF (unsupervised)
W is shared between NMF & Cox
α → 1: gene-level Cox model
(B) Data-driven model selection via Bayesian optimization
Model complexity (k)
Supervision strength (α)
Regularization (λ)
Low
High
Bayesian Optimization
Propose model configuration
Evaluate survival performance (cross-validated C-index)
Update surrogate
Selected model configuration (k*, α*, λ*)

Fig. 1. DeSurv overview. Overview of the DeSurv training pipeline. A. The input is a preprocessed bulk gene expression matrix and patient survival outcomes. B. The Desurv model optimizes a joint NMF + Cox loss function. C. The DeSurv model estimates three quantities: Gene weights matrix (W), Subject loadings matrix (H), and regression coefficients from the Cox model (beta). D. We extract the exemplar genes from each signature in the gene weights matrix. E. The exemplar genes from survival associated factors provide prognostic gene signatures.

typically associated with the Basal-like subtype of pancreatic ductal adenocarcinoma (PDAC).

Survival analysis of the factor signature scores on the training data demonstrates clear prognostic stratification. High expression of Factor 1 is associated with longer overall survival (HR = 0.37, p < 2e-16), whereas Factor 3 is linked to significantly shorter survival (HR = 1.43, p = 2e-4). These trends are consistent with known biology: immune-infiltrated tumors generally exhibit improved prognosis, while Basal-like tumors are aggressive and clinically refractory. Factor 2, corresponding to the normal/exocrine signal, shows no significant association with survival (HR = 0.99, p = 0.91), indicating that while this component contributes to reconstructing the expression matrix, it does not carry prognostic information. This behavior illustrates a key property of DeSurv—its joint optimization ensures that the latent factors balance both survival relevance and expression reconstruction, preserving biologically necessary structure even when not directly linked to outcome.

Panel D of Figure @ref(fig:fig-bio) further examines the correlation between DeSurv factors and previously published PDAC gene signatures. Factor 1 aligns with signatures from classical tumor, iCAF and restCAF stroma, and immune infiltration, all of which are associated with favorable prognosis. In contrast, Factor 3 correlates with Basal-like tumor, proCAF, and activated stromal programs, which are generally linked to poor outcomes. These overlaps suggest that each factor represents a composite axis of tumor–stroma-immune interaction, capturing biologically coherent programs that align with known PDAC ecosystems but arise de novo from the DeSurv model.

External validation with projected DeSurv factor scores. To evaluate the generalizability of DeSurv-derived prognostic signatures, we projected the DeSurv weights onto multiple independent PDAC cohorts (Dijk, PACA-AU RNA-seq and microarray, Moffitt, and Puleo). We selected the two most prognostic factors using `select_desurv_factors`, rank-transformed expression within each sample over the union of the selected factor gene lists, and computed validation scores as $Z_{val} = X_{val}^T W$ restricted to the top-gene signatures. This yields continuous factor score vectors for each cohort without clustering.

Figure @ref(fig:fig-extval)A shows the ranked expression heatmap for the selected factor gene signatures, annotated with the continuous factor scores and the PurIST, DeCAF, and dataset labels. Figure @ref(fig:fig-extval)B plots the top two factor scores for all samples, colored by PurIST subtype and shaped by DeCAF subtype to show how tumor- and stroma-associated programs align across cohorts. For survival assessment, we discretized each factor score within each cohort at its median and intersected the high/low calls to form four groups; Kaplan–Meier curves for these groups are shown in Figure @ref(fig:fig-extval)C. Because DeSurv was trained only on TCGA-PAAD/CPTAC, these analyses represent an independent validation of prognostic signal transfer to external datasets.

Bladder cancer analysis. We will report a focused bladder cancer analysis here, including a dedicated figure. Placeholder text: summary of bladder cohort preprocessing, DeSurv factor interpretation, and survival stratification results to be added once the bladder figure is finalized.

Discussion

We present DeSurv, a survival-driven deconvolution framework that integrates nonnegative matrix factorization with Cox proportional hazards modeling to uncover latent gene-expression programs associated with patient outcomes. By coupling unsupervised matrix decomposition with direct survival supervision, DeSurv bridges the gap between descriptive molecular subtyping and prognostic modeling. Unlike conventional NMF, which identifies factors that explain transcriptional variance without regard to clinical relevance, DeSurv simultaneously optimizes reconstruction fidelity and survival discrimination, yielding interpretable gene programs that are both biologically coherent and clinically informative.

In pancreatic ductal adenocarcinoma (PDAC), DeSurv identified three major transcriptional programs corresponding to immune/iCAF, normal/exocrine, and basal-like factors. The immune/iCAF factor was associated with prolonged survival and enriched for inflammatory and immune-response pathways, consistent with prior evidence linking tumor-infiltrating immune and inflammatory fibroblast populations to improved outcomes. In contrast, the basal-like factor was strongly associated with poor survival and enriched for epithelial–mesenchymal transition and cell-cycle programs characteristic of aggressive tumor phenotypes. The normal/exocrine factor, while not prognostic, captured background pancreatic tissue signal and contributed to accurate reconstruction, demonstrating that DeSurv can disentangle prognostic sources of variation while preserving biology.

Importantly, DeSurv outperformed unsupervised NMF in cross-validated concordance index, achieving higher predictive accuracy with fewer factors. This suggests that explicitly incorporating survival information regularizes the factorization, producing a more parsimonious representation that focuses on biologically and clinically relevant variation. Moreover, when applied to independent PDAC cohorts—including Dijk, PACA-AU, Moffitt, and Puleo—DeSurv-derived gene signatures stratified patients into three reproducible clusters with clear prognostic differences. These clusters aligned strongly with existing PurIST and DeCAF classifiers, recapitulating the interplay between tumor-intrinsic (Basal versus Classical) and stromal (proCAF versus restCAF/iCAF) subtypes. Notably, survival prediction appeared to depend on the presence or absence of basal and immune/iCAF programs rather than binary subtype identities, indicating that the co-activation or suppression of these programs better captures the biological continuum of PDAC progression.

Beyond PDAC, DeSurv provides a generalizable framework for discovering prognostic transcriptional programs across complex cancers where tumor and stromal signals are interwoven. By integrating survival modeling into matrix factorization, it unifies molecular deconvolution and outcome prediction within a single, interpretable model. This approach complements both single-cell–based tumor–stroma characterization and bulk expression subtyping, offering a scalable route to translate transcriptomic heterogeneity into clinically actionable prognostic insight. Future work will extend DeSurv to multi-omic data and time-varying outcomes, further enhancing its ability to resolve dynamic tumor–microenvironment interactions that shape disease progression and therapy response.

Materials and methods

A. Problem formulation and notation. Let $X \in \mathbb{R}_{\geq 0}^{p \times n}$ denote the nonnegative gene expression matrix. DeSurv approximates $X \approx WH$, where $W \in \mathbb{R}_{\geq 0}^{p \times k}$ contains nonnegative gene programs and $H \in \mathbb{R}_{\geq 0}^{k \times n}$ contains sample-level activations. Additionally, let $y,\ \delta \in \mathbb{R}^n$ represent patient survival times and censoring indicators, respectively. Survival outcomes are modeled through a cox proportional hazards model with covariates $Z = W^\top X$.

B. The DeSurv Model. DeSurv integrates Nonnegative Matrix Factorization (NMF) with penalized Cox regression to identify gene programs associated with patient survival.

The joint objective is

$$\mathcal{L}(W, H, \beta) = (1-\alpha)\,\mathcal{L}_{\mathrm{NMF}}(W, H) - \alpha\,\mathcal{L}_{\mathrm{Cox}}(W, \beta), \qquad [2]$$

where $\mathcal{L}_{\mathrm{NMF}}(W, H)$ is the NMF reconstruction error and $\mathcal{L}_{\mathrm{Cox}}(W, \beta)$ is the elastic-net penalized partial log-likelihood. Optimization proceeds by alternating updates for H, W, and β, using multiplicative rules for H (13), projected gradients for W, and coordinate descent for β. Although non-convex, these updates are shown to converge to a stationary point under mild conditions (SI Appendix). Complete derivations and algorithmic details are provided in the SI Appendix.

C. Hyperparameter selection and cross-validation. Hyperparameters $(k, \alpha, \lambda_H, \lambda, \xi)$ were selected by maximizing the cross-validated C-index using Bayesian optimization. Each fold was trained using multiple random initializations, and fold-level performance was defined as the average C-index across initializations. For stability, we used a consensus-based initialization for the final model, aggregating multiple DeSurv runs into a gene–gene co-occurrence matrix and constructing an initialization W_0 from the resulting clusters (SI Appendix). Before validation, each column of W was truncated to its BO-selected number of top genes (details in SI Appendix), denoted $\tilde{W}$. External validation was performed by first projecting new datasets onto the learned programs via $Z = \tilde{W}^\top X_{\mathrm{new}}$ and evaluating survival associations using C-index and log-rank statistics. To evaluate the quality of the DeSurv derived gene signatures for subtyping, the new datasets X_new were clustered on genes in $\tilde{W}$, and survival differences were analyzed for the derived clusters. Further training, validation, and runtime details appear in the SI Appendix.

D. Simulation studies. Simulation studies were conducted to assess recovery of prognostic latent structure and survival prediction. Gene expression data were generated from a nonnegative factor model $X = WH$, where gene loadings W comprised three gene classes: marker genes, background genes, and noise genes. Marker genes were simulated to load strongly on a single factor and weakly on others, background genes to load strongly across all factors, and noise genes to have uniformly low loadings; each class was generated from a distinct gamma distribution. Sample-level factor activities H were generated from a gamma distribution.

Survival times were generated from an exponential proportional hazards model in which risk depended on marker gene structure through $X^T\tilde{W}$, where $\tilde{W}$ retained marker gene loadings for their corresponding factors and was zero otherwise; censoring times were generated independently from an exponential distribution. Each dataset was analyzed using DeSurv, standard NMF followed by Cox regression on inferred factors, and penalized Cox regression (CoxNet), with all methods tuned using the same cross-validated concordance index. Performance was summarized across repeated simulation replicates.

E. Evaluation metrics.

F. Real-world datasets. We analyzed publicly available RNA-seq and microarray cohorts of pancreatic ductal adenocarcinoma (PDAC) and bladder cancer with corresponding overall survival outcomes. Gene expression matrices were converted to TPM, log-transformed, and filtered to remove low-expression genes. Survival times and censoring indicators were taken from the associated clinical annotations. Of the seven PDAC cohorts we considered, two were used for training (TCGA and CPTAC) and the rest were used for external validation (Dijk, Moffitt, PACA, Puleo). The bladder cohort was split into training and validation cohorts via a 70/30 split. To harmonize differences in scale across cohorts, filtered gene expression data was within-subject rank transformed before model training. More details about the datasets can be found in the SI Appendix.

G. Simulations, Benchmarking, and Availability. Code and processed data used in this study are available at [repository link].

ACKNOWLEDGMENTS. Please include your acknowledgments here, set in a single paragraph. Please do not include any acknowledgments in the Supporting Information, or anywhere else in the manuscript.

H. Model details. Let $X \in \mathbb{R}_{\geq 0}^{p\times n}$ denote the nonnegative gene expression matrix with p genes and n subjects. DeSurv approximates $X \approx WH$ with $W \in \mathbb{R}_{\geq 0}^{p\times k}$ and $H \in \mathbb{R}_{\geq 0}^{k\times n}$, and links the shared program matrix W to survival via an elastic-net–penalized Cox model.

Let $y \in \mathbb{R}_{\geq 0}^{n}$ be the observed survival times, $\delta \in \{0,1\}^n$ the event indicators, and $Z = X^\top W \in \mathbb{R}^{n\times k}$ the sample-wise program loadings. We write $Z_i^\top$ for the ith row of Z, and $n_{\text{event}} = \sum_{i=1}^{n} \delta_i$. For convenience we recall the DeSurv loss from Equation~2 in the main text and rewrite it as

$$\mathcal{L}(W,H,\beta) = \\ (1-\alpha)\left(\frac{1}{2np}\|X-WH\|_F^2 + \frac{\lambda_H}{2nk}\|H\|_F^2\right) \\ -\alpha\left(\frac{2}{n_{event}}\ell(W,\beta) - \lambda\{\xi\|\beta\|_1 + \tfrac{1-\xi}{2}\|\beta\|_2^2\}\right)$$

where $\|\cdot\|_F$ is the Frobenius norm and $\beta \in \mathbb{R}^k$ are the Cox coefficients. The Cox log-partial likelihood $\ell(W,\beta)$ is

$$\ell(W,\beta) = \sum_{i=1}^{n} \delta_i \left[Z_i^\top \beta - \log\left(\sum_{j:y_j\geq y_i} \exp(Z_j^\top \beta) \right) \right]. \qquad [3]$$

Hyperparameters satisfy $\alpha \in [0,1)$, $\lambda_H \geq 0$, $\lambda \geq 0$, and $\xi \in [0,1]$. The constants $1/(2np)$, $2/n_{\text{event}}$, and $1/(2nk)$ are chosen for numerical convenience and do not affect the minimizer.

I. Optimization Algorithm.

I.1. Block coordinate descent scheme. Algorithm~1 summarizes the block coordinate descent (BCD) scheme used to minimize $\mathcal{L}(W,H,\beta)$. At each outer iteration, we update H, then W, then β while holding the other blocks fixed.

Algorithm 1 DeSurv block coordinate descent

```
Input: $X \in \mathbb{R}_{\geq 0}^{p\times n}$, survival times $y \in \mathbb{R}_{\geq 0}^{n}$, event indicators
    $\delta \in \{0,1\}^n$, tolerance $tol$, maximum iterations $maxit$,
    supervision parameter $\alpha$, rank $k$, penalties $\lambda_H$, $\lambda$, and $\xi$
Output: Fitted DeSurv parameters $(W, H, \beta)$
Initialize $W^{(0)}, H^{(0)}$ with positive entries (e.g., $W^{(0)}, H^{(0)} \sim \text{Uniform}(0, \max X)$)
Initialize $\beta^{(0)}$ (e.g., $\beta^{(0)} = \mathbf{1}$)
$loss = \mathcal{L}(W^{(0)}, H^{(0)}, \beta^{(0)})$
$eps = \infty$, $t = 0$
while $eps \geq tol$ and $t < maxit$ do
    (H-update)
        Update $H^{(t+1)}$ from $H^{(t)}$ using the multiplicative rule in Eq. 4,
        holding $W^{(t)}$ and $\beta^{(t)}$ fixed.
    (W-update)
        Update $W^{(t+1)}$ from $W^{(t)}$ using the hybrid multiplicative rule in Eq. 8,
        holding $H^{(t+1)}$ and $\beta^{(t)}$ fixed, and perform backtracking to ensure
        $\mathcal{L}$ does not increase.
    ($\beta$-update)
        Update $\beta^{(t+1)}$ from $\beta^{(t)}$ using a Newton-like step for the Cox loss
        (Eq. 9), holding $W^{(t+1)}$ and $H^{(t+1)}$ fixed.
    $lossNew = \mathcal{L}(W^{(t+1)}, H^{(t+1)}, \beta^{(t+1)})$
    $eps = |lossNew - loss|/|loss|$
    $loss = lossNew$
    $t = t + 1$
return $\hat{W} = W^{(t+1)}, \hat{H} = H^{(t+1)}, \hat{\beta} = \beta^{(t+1)}$
```

I.2. Update for H. Conditional on W, the loss $\mathcal{L}(W,H,\beta)$ reduces to a strictly convex quadratic function of H. We adopt the standard NMF multiplicative update with ℓ_2 penalty:

$$H \leftarrow \max\left(H \odot \frac{W^\top X}{W^\top WH + \lambda_H H + \varepsilon_H}, \varepsilon_H \right), \qquad [4]$$

where $\odot$ denotes elementwise multiplication, all divisions are elementwise, and $\varepsilon_H > 0$ is a small floor to prevent entries from becoming exactly zero. This update is equivalent to a majorization–minimization step and guarantees a nonincreasing reconstruction term conditional on W (19, 20). The ε_H-floor ensures that iterates remain in the interior of the nonnegative orthant, which simplifies the convergence analysis.

I.3. Update for W. For W, we construct a hybrid multiplicative update that balances the contributions of the NMF and Cox gradients. Let

$$\nabla_W \mathcal{L}_{\text{NMF}}(W,H) = \frac{1}{np}(WH - X)H^\top,$$

and let

$$\nabla_W \mathcal{L}_{\text{Cox}}(W,\beta) = \frac{2}{n_{event}} \nabla_W \ell(W,\beta).$$

where $\nabla_W \ell(W,\beta)$ denotes the Cox gradient with respect to W. A closed-form expression for $\nabla_W \ell$ is

$$\nabla_W \ell(W,\beta) = \sum_{i=1}^{n} \delta_i \left(x_i - \frac{\sum_{j:y_j\geq y_i} x_j \exp(x_j^\top W\beta)}{\sum_{j:y_j\geq y_i} \exp(x_j^\top W\beta)} \right) \beta^\top, \qquad [5]$$

where x_i denotes the ith column of X.

We define the gradient-balancing factor

$$\delta^{(t)} = \frac{\left\|(1-\alpha)\nabla_W \mathcal{L}_{\mathrm{NMF}}(W^{(t)}, H^{(t+1)})\right\|_F^2}{\left\|\alpha \nabla_W \mathcal{L}_{Cox}(W^{(t)}, \beta^{(t)})\right\|_F^2}, \qquad [6]$$

and clip $\delta^{(t)}$ to avoid extreme values: $\delta^{(t)} \leftarrow \min(\delta^{(t)}, \delta_{\max})$ with $\delta_{\max} = 10^6$ in all experiments.

The multiplicative factor for W is then

$$R^{(t)} = \frac{\frac{(1-\alpha)}{np} X H^{(t+1)\top} + \frac{2\alpha}{n_{\mathrm{event}}} \delta^{(t)} \nabla_W \ell(W^{(t)}, \beta^{(t)})}{\frac{(1-\alpha)}{np} W^{(t)} H^{(t+1)} H^{(t+1)\top} + \varepsilon_W}, \qquad [7]$$

and the proposed update is

$$W^{(t+1)} = \max\Big(W^{(t)} \odot R^{(t)}, \varepsilon_W\Big), \qquad [8]$$

with a small floor $\varepsilon_W > 0$. When $\alpha = 0$, this reduces to the standard multiplicative update for NMF (19); for $\alpha > 0$, the Cox gradient perturbs the update in a direction that decreases the supervised loss.

Because the combined multiplicative step does not, by itself, guarantee a nonincrease in the full loss $\mathcal{L}$, we embed it in a backtracking line search.

Algorithm 2 W update with backtracking

```
Input: Value of $W$ and the previous iteration $t$: $W^{(t)}$, the multiplicative update term at the current iteration $R^{(t+1)}$, the max number of backtracking updates $max\_bt$, the backtracking parameter $\rho \subset (0,1)$
Output: $W^{(t+1)}$
$\theta = 1$
b = 1
$flag\_accept = FALSE$
while $b \leq max\_bt$ do
    $W_{cand}^{(t+1)} = W^{(t)} \odot [(1-\theta) + \theta R^{(t+1)}]$
    (Column normalization)
        Compute $D = \mathrm{diag}\big(\|W_{(:,1)cand}^{(t+1)}\|_2, \dots, \|W_{(:,k)cand}^{(t+1)}\|_2\big)$
        and set $W_{cand}^{(t+1)} \leftarrow W_{cand}^{(t+1)} D^{-1}$, $H_{cand}^{(t+1)} \leftarrow DH^{(t+1)}$, and $\beta_{cand}^{(t)} \leftarrow D\beta^{(t)}$
    if $\mathcal{L}(W_{cand}^{(t+1)}, H^{(t+1)}, \beta^{(t)}) \leq \mathcal{L}(W^{(t)}, H^{(t+1)}, \beta^{(t)})$ then
        $W^{(t+1)} = W_{cand}^{(t+1)}$
        $H^{(t+1)} = H_{cand}^{(t+1)}$
        $\beta^{(t)} = \beta_{cand}^{(t)}$
        $flag\_accept = TRUE$
        break
    $\theta = \theta * \rho$
    $b = b + 1$
if $flag\_accept = FALSE$ then
    $W^{(t+1)} = W^{(t)}$
```

The column normalization preserves both WH and $W\beta$, and therefore leaves the loss $\mathcal{L}$ invariant up to numerical error. Backtracking guarantees that the accepted W update does not increase $\mathcal{L}$.

I.4. Update for β**.** Conditional on (W, H), the loss in β reduces to a convex elastic-net–penalized Cox problem:

$$\min_{\beta \in \mathbb{R}^k} \left\{ -\frac{2\alpha}{n_{\mathrm{event}}} \ell(W, \beta) + \lambda \left(\xi \|\beta\|_1 + \frac{1-\xi}{2} \|\beta\|_2^2 \right) \right\}.$$

We solve this subproblem by cyclic coordinate descent following (21). Writing $\ell(\beta) = \ell(W, \beta)$ and $\tilde{\eta}_i = Z_i^\top \beta$, the update for coordinate r has the closed form

$$\hat{\beta}_r = \frac{S\left(\frac{1}{n} \sum_{i=1}^n w(\tilde{\eta})_i z_{i,r} \left[v(\tilde{\eta})_i - \sum_{j \neq r} z_{i,j} \beta_j \right], \lambda\xi \right)}{\frac{1}{n} \sum_{i=1}^n w(\tilde{\eta})_i z_{i,r}^2 + \lambda(1-\xi)}, \qquad [9]$$

where $S(a, \tau) = \mathrm{sign}(a) \max(|a| - \tau, 0)$ is the soft-thresholding operator, and $w(\tilde{\eta})$, $v(\tilde{\eta})$ are standard local quadratic approximations of the Cox log-partial likelihood (21). We iterate coordinate updates until the subproblem converges to numerical tolerance, yielding a minimizer in β for the current W.

I.5. Normalization of W. After each accepted W update, we normalize the columns of W as $W \leftarrow WD^{-1}$, and adjust H and β as

$$H \leftarrow DH, \qquad \beta \leftarrow D\beta,$$

where D is the diagonal matrix of column ℓ_2-norms of W. This preserves the reconstruction WH and the linear predictor $W\beta$, leaving both the NMF and Cox terms in the loss unchanged. Normalization prevents degeneracy in the scale-nonidentifiable factorization and keeps columns of W comparable in magnitude and interpretable as gene programs.

I.6. Derivation of W update from projected coordinate descent. The W update can be derived directly from the projected coordinate descent update with a specific step size.

Recall that the overall loss function is

$$\mathcal{L}(W, H, \beta) = \frac{(1-\alpha)}{2np} ||X - WH||_F^2 - \frac{2\alpha}{n_{event}} \ell(W, \beta) + \lambda(\xi ||\beta||_1 + \frac{(1-\xi)}{2}, \qquad [10]$$

where $\ell(W, \beta)$ is the log-partial likelihood for the Cox model. Let $\nabla_W \ell$ represent the derivative of $\ell(W, \beta)$ with respect to W. Then the derivative of the overall loss with respect to W is

$$\frac{\partial L}{\partial W} = \frac{(1-\alpha)}{np} \Big(WHH^T - XH^T\Big) - \frac{2\alpha}{n_{event}} \nabla_W \ell \qquad [11]$$

Then the gradient descent update rule at iteration t is

$$\begin{aligned} W^{(t)} &= W^{(t-1)} - \gamma \left(\frac{\partial L}{\partial W} \right) \\ &= W^{(t-1)} - \gamma \left(\frac{(1-\alpha)}{np} \Big(WHH^T - XH^T\Big) - \frac{2\alpha}{n_{event}} \nabla_W \ell \right) \end{aligned} \qquad [12]$$

$$[13]$$

Let the step size γ be defined as in CITE:

$$\gamma = \frac{np}{(1-\alpha)} \frac{W}{WHH^T} \qquad [14]$$

Then the update becomes

$$\begin{aligned} W^{(t)} &= W^{(t-1)} - \frac{np}{(1-\alpha)} \frac{W}{WHH^T} \left(\frac{(1-\alpha)}{np} \Big(WHH^T - XH^T\Big) - \frac{2\alpha}{n_{event}} \nabla_W \ell \right) \\ &= \frac{W}{WHH^T} XH^T + \frac{2\alpha np}{n_{event}(1-\alpha)} \nabla_W \ell \\ &= W \odot \frac{XH^T + \frac{2\alpha np}{n_{event}(1-\alpha)} \nabla_W \ell}{WHH^T} \\ &= W \odot \frac{\frac{(1-\alpha)}{np} XH^T + \frac{2\alpha}{n_{event}} \nabla_W \ell}{\frac{(1-\alpha)}{np} WHH^T} \end{aligned} \qquad [15]$$

Finally, projected coordinate descent projects the W update in to the positive space

$$W^{(t)} = \max \left(W \odot \frac{\frac{(1-\alpha)}{np} XH^T + \frac{2\alpha}{n_{event}} \nabla_W \ell}{\frac{(1-\alpha)}{np} WHH^T}, 0 \right) \qquad [16]$$

Since $W \geq 0$ this is equivalent to

$$W^{(t)} = W \odot \max \left(\frac{\frac{(1-\alpha)}{np} XH^T + \frac{2\alpha}{n_{event}} \nabla_W \ell}{\frac{(1-\alpha)}{np} WHH^T}, 0 \right) \qquad [17]$$

which matches the multiplicative form used in the software implementation.

J. Convergence proof. We show that, under mild regularity conditions, the block coordinate descent (BCD) Algorithm~S1 converges to a stationary point of the DeSurv loss function

$$\mathcal{L}(W, H, \beta).$$

For clarity, we first analyze the algorithm *without* the column normalization of W inside the loop, and then argue in Section~J.1 that the normalization step preserves stationarity of limit points.

Throughout, let $\theta = (W, H, \beta)$ and denote $\mathcal{L}(\theta) = \mathcal{L}(W, H, \beta)$. The feasible set is

$$\Theta = \{(W, H, \beta) : W \in \mathbb{R}^{p\times k}_{\geq 0},\ H \in \mathbb{R}^{k\times n}_{\geq 0},\ \beta \in \mathbb{R}^k\}.$$

Assumption 1 (Regularity and parameter space). We assume:

(i) The data $X \in \mathbb{R}^{p\times n}_{\geq 0}$, $y \in \mathbb{R}^n$, and $\delta \in \{0, 1\}^n$ are fixed and bounded.

(ii) The hyperparameters satisfy $\lambda_H > 0$, $\lambda > 0$, $\alpha \in [0, 1)$, and $\xi \in [0, 1]$.

(iii) The initial iterate $\theta^{(0)} = (W^{(0)}, H^{(0)}, \beta^{(0)})$ lies in Θ and satisfies $W^{(0)}, H^{(0)} > 0$ elementwise.

Assumption 2 (Block updates). At each outer iteration t, the three block updates in Algorithm~S1 satisfy:

(i) *H-update.* For fixed $(W^{(t)}, \beta^{(t)})$, the update $H^{(t)} \mapsto H^{(t+1)}$ is given by the multiplicative rule

$$H \leftarrow H \odot \frac{W^\top X}{W^\top WH + \lambda_H H},$$

applied at $(W^{(t)}, H^{(t)})$. This rule preserves nonnegativity and yields a nonincreasing value of the reconstruction term conditional on $W^{(t)}$ and $\beta^{(t)}$; see e.g. [@seung2001algorithms; @pascualmontano2006nonsmooth; @lin2007convergence].

(ii) *W-update.* For fixed $(H^{(t+1)}, \beta^{(t)})$, the update $W^{(t)} \mapsto W^{(t+1)}$ is the hybrid multiplicative step followed by backtracking as in Algorithm S2, producing $W^{(t+1)}$ such that

$$\mathcal{L}(W^{(t+1)}, H^{(t+1)}, \beta^{(t)}) \leq \mathcal{L}(W^{(t)}, H^{(t+1)}, \beta^{(t)}).$$

If no candidate satisfies the Armijo-type condition within the allowed number of backtracking steps, the algorithm sets $W^{(t+1)} := W^{(t)}$.

(iii) *β-update.* For fixed $(W^{(t+1)}, H^{(t+1)})$, the update $\beta^{(t)} \mapsto \beta^{(t+1)}$ is obtained by running the coordinate descent method of [@simon2011regularization] on the convex elastic-net Cox subproblem until convergence. Thus

$$\beta^{(t+1)} \in \arg\min_{\beta\in\mathbb{R}^k} \mathcal{L}(W^{(t+1)}, H^{(t+1)}, \beta),$$

and

$$\mathcal{L}(W^{(t+1)}, H^{(t+1)}, \beta^{(t+1)}) \leq \mathcal{L}(W^{(t+1)}, H^{(t+1)}, \beta^{(t)}).$$

Lemma 1 (Continuity and bounded level sets). Under Assumption 1, $\mathcal{L} : \Theta \to \mathbb{R}$ is continuous, and the initial sublevel set

$$\mathcal{S}_0 := \{\theta \in \Theta : \mathcal{L}(\theta) \leq \mathcal{L}(\theta^{(0)})\}$$

is nonempty, closed, and bounded.

Proof. The NMF term $\|X - WH\|_F^2$ is a polynomial in the entries of W and H, hence is continuously differentiable. The penalty $\|H\|_F^2$ is continuous as well. The Cox partial log-likelihood is a smooth function of the linear predictor $\eta_i = x_i^\top W\beta$, which is linear in (W, β), so this term is continuously differentiable in (W, β). The ℓ_2 penalty on β is smooth, and the ℓ_1 penalty is convex and lower semicontinuous. Therefore $\mathcal{L}$ is continuous on Θ.

The constraints $W, H \geq 0$ define closed convex cones, and $\beta \in \mathbb{R}^k$ is unconstrained. Because $\lambda_H > 0$ and $\lambda > 0$, the terms $\frac{\lambda_H}{nk}\|H\|_F^2$ and $\lambda\frac{1-\xi}{2}\|\beta\|_2^2$ dominate the objective as $\|H\|_F \to \infty$ or $\|\beta\|_2 \to \infty$, respectively. Thus the sublevel set $\mathcal{S}_0$ is bounded in (H, β). Since $W \geq 0$ and X is fixed, the reconstruction term and Cox term being bounded prevent W from diverging while $\mathcal{L}$ remains below $\mathcal{L}(\theta^{(0)})$, so W is bounded on $\mathcal{S}_0$. Because Θ is closed and $\mathcal{L}$ is continuous, $\mathcal{S}_0$ is closed. Nonemptiness follows from $\theta^{(0)} \in \mathcal{S}_0$. □

Lemma 2 (Monotone descent and existence of limit points). Under Assumptions 1 and 2, Algorithm~S1 (without W normalization) generates a sequence $\{\theta^{(t)}\}_{t\geq 0} \subset \mathcal{S}_0$ such that

$$\mathcal{L}(\theta^{(t+1)}) \leq \mathcal{L}(\theta^{(t)}) \quad \forall t,$$

and $\{\mathcal{L}(\theta^{(t)})\}$ converges to a finite limit $\mathcal{L}^*$. Moreover, $\{\theta^{(t)}\}$ is bounded and therefore admits at least one limit point.

Proof. By Assumption 2(i),

$$\mathcal{L}(W^{(t)}, H^{(t+1)}, \beta^{(t)}) \leq \mathcal{L}(W^{(t)}, H^{(t)}, \beta^{(t)}).$$

By Assumption 2(ii),

$$\mathcal{L}(W^{(t+1)}, H^{(t+1)}, \beta^{(t)}) \leq \mathcal{L}(W^{(t)}, H^{(t+1)}, \beta^{(t)}).$$

By Assumption 2(iii),

$$\mathcal{L}(W^{(t+1)}, H^{(t+1)}, \beta^{(t+1)}) \leq \mathcal{L}(W^{(t+1)}, H^{(t+1)}, \beta^{(t)}).$$

Combining these inequalities gives

$$\mathcal{L}(\theta^{(t+1)}) \leq \mathcal{L}(\theta^{(t)}) \quad \forall t,$$

so $\{\mathcal{L}(\theta^{(t)})\}$ is monotonically nonincreasing. Since $\mathcal{L}$ is bounded below on Θ, the sequence converges to some finite $\mathcal{L}^*$.

Each iterate lies in $\mathcal{S}_0$ by construction, and $\mathcal{S}_0$ is bounded by Lemma 1. Therefore $\{\theta^{(t)}\}$ is bounded and has at least one limit point by the Bolzano–Weierstrass theorem. □

Lemma 3 (Blockwise optimality of limit points). Let $\theta^* = (W^*, H^*, \beta^*)$ be any limit point of $\{\theta^{(t)}\}$ under Assumptions~1–2. Then:

(i) H^* satisfies the KKT conditions for

$$\min_{H\geq 0} \mathcal{L}(W^*, H, \beta^*).$$

(ii) W^* satisfies the KKT conditions for

$$\min_{W\geq 0} \mathcal{L}(W, H^*, \beta^*).$$

(iii) β^* is the unique minimizer of

$$\min_{\beta\in\mathbb{R}^k} \mathcal{L}(W^*, H^*, \beta),$$

and satisfies the KKT conditions for the elastic-net Cox subproblem.

Proof. Let $\{t_j\}$ be a subsequence such that $\theta^{(t_j)} \to \theta^* = (W^*, H^*, \beta^*)$. By continuity of $\mathcal{L}$, $\mathcal{L}(\theta^{(t_j)}) \to \mathcal{L}^*$.

(i) Conditional on (W, β), the H-subproblem is

$$\min_{H\geq 0} \frac{(1-\alpha)}{np}(\|X - WH\|_F^2 + \frac{\lambda_H}{nk}||H||_F^2) + \text{const in } H,$$

a strictly convex quadratic program. The multiplicative update for H is known to be a descent method whose fixed points coincide with the KKT points of this constrained subproblem (19, 22). Since the full objective $\mathcal{L}(\theta^{(t)})$ converges, the per-iteration decrease in $\mathcal{L}$ due to the H-update must vanish along $\{t_j\}$. If H^* were not KKT for the H-subproblem given (W^*, β^*), there would exist a feasible descent direction for H at (W^*, β^*), implying that for sufficiently large j the H-update would still produce a strict decrease in $\mathcal{L}$, contradicting convergence. Hence H^* satisfies the KKT conditions.

(ii) Conditional on (H, β), the W-subproblem is differentiable and convex in W (sum of a quadratic term and a negative Cox partial log-likelihood in a linear predictor). The hybrid multiplicative step with backtracking can be viewed as a projected gradient-like update onto the nonnegative orthant. Under mild conditions on the search direction and step sizes, any limit point of such a projected gradient scheme is a KKT point for the constrained subproblem; see, for example, (23, 24). If W^* did not satisfy the KKT conditions, there would be a feasible descent direction at (W^*, H^*, β^*) and a sufficiently small step that reduces $\mathcal{L}$, which the line search would eventually accept. This would contradict the fact that $\mathcal{L}(\theta^{(t_j)}) \to \mathcal{L}^*$. Therefore W^* is KKT for the W-subproblem.

(iii) For fixed (W, H), the β-subproblem is the convex elastic-net penalized Cox objective. The coordinate descent algorithm converges to the unique minimizer. By Assumption 2(iii),$\beta^{(t+1)}$ is taken to be this minimizer at each iteration. Passing to the limit along $\{t_j\}$ shows that β^* is the unique minimizer of the β-block given (W^*, H^*) and hence satisfies its KKT conditions. □

Theorem 1 (Convergence to a stationary point). Under Assumptions 1 and 2, the sequence $\{\theta^{(t)}\}$ produced by Algorithm~S1 (without W normalization) satisfies:

(a) The objective values $\{\mathcal{L}(\theta^{(t)})\}$ are monotonically nonincreasing and converge to a finite limit $\mathcal{L}^*$.

(b) Every limit point $\theta^* = (W^*, H^*, \beta^*)$ of $\{\theta^{(t)}\}$ is a stationary point of $\mathcal{L}$ on Θ, i.e. it satisfies the first-order KKT conditions for
$$\min_{\theta\in\Theta} \mathcal{L}(\theta).$$

Proof. Part (a) is Lemma 2. For part (b), Lemma 3 shows that any limit point θ^* is blockwise optimal: each block is optimal (in the KKT sense) when the others are held fixed.

The DeSurv loss can be written as
$$\mathcal{L}(\theta) = f(W, H, \beta) + g(\beta) + I_{\{W\geq 0\}}(W) + I_{\{H\geq 0\}}(H),$$
where f is continuously differentiable, $g(\beta) = \lambda\xi\|\beta\|_1$ is a separable convex (possibly nondifferentiable) penalty, and I_C denotes the indicator of a closed convex set C. This is the smooth+nonsmooth composite form considered in block coordinate descent analyses such as (23). In this setting, any point that is optimal with respect to each block individually (a block coordinatewise minimizer) is a stationary point for the full problem: equivalently,
$$0 \in \nabla_{W,H} f(W^*, H^*, \beta^*) + \partial I_{\{W\geq 0\}}(W^*) + \partial I_{\{H\geq 0\}}(H^*),$$
$$0 \in \nabla_\beta f(W^*, H^*, \beta^*) + \partial g(\beta^*),$$
which are precisely the KKT conditions for $\min_{\theta\in\Theta} \mathcal{L}(\theta)$. Hence every limit point of Algorithm~S1 is stationary. □

J.1. Effect of column normalization of W.

The above analysis omits the column-normalization step for W used in Algorithm~S2. We briefly argue that this normalization does not affect stationarity of limit points.

Let D be a diagonal matrix with strictly positive diagonal entries. The transformation
$$(W, H, \beta) \mapsto (W', H', \beta') = (WD^{-1},\ DH,\ D\beta)$$
preserves both the product WH and the linear predictor $W\beta$, and hence leaves $\mathcal{L}$ invariant. The column-normalization step is exactly such a transformation, with D chosen from the column norms of W.

Let $\{\theta^{(t)}\}$ be the sequence generated by the algorithm without normalization, and let $\{\tilde{\theta}^{(t)}\}$ be the sequence with normalization. For each t there exists a diagonal $D^{(t)}$ with strictly positive entries such that
$$\tilde{\theta}^{(t)} = (W^{(t)}D^{(t),-1},\ D^{(t)}H^{(t)},\ D^{(t)}\beta^{(t)}),$$
and $\mathcal{L}(\tilde{\theta}^{(t)}) = \mathcal{L}(\theta^{(t)})$. Thus limit points of $\{\tilde{\theta}^{(t)}\}$ are obtained from limit points of $\{\theta^{(t)}\}$ by such invertible diagonal scalings.

Since the KKT conditions are expressed in terms of the gradients with respect to WH and $W\beta$, and these quantities are invariant under the above scaling, stationarity is preserved under the transformation. Therefore Theorem~1 implies that every limit point of the *normalized* algorithm is also a stationary point of $\mathcal{L}$ (up to this scaling equivalence), which establishes convergence of the implementation used in practice.

J.2. Remark (Coxnet implementation).

Our implementation updates the β block using a Coxnet-style coordinate descent that relies on an approximate Hessian. Consequently, the formal optimality proof is established for an idealized version of the β update that assumes exact second-order information. Nonetheless, in empirical applications the approximate update is numerically stable and yields a monotone decrease in the objective, consistent with the behavior predicted by the idealized analysis.

K. Supervision on W versus H.

Classical nonnegative matrix factorization (NMF) is non-identifiable: for any invertible matrix R with nonnegative entries, the factorization (W, H) can be transformed to $(WR, R^{-1}H)$ without changing the reconstruction error, yielding multiple valid solutions (25–27). Although normalizing columns of W or rows of H resolves the trivial scaling ambiguity, nonnegative mixing transformations persist, and empirical studies consistently show that the sample-loading matrix H is more sensitive to initialization and local minima than the program matrix W (28–30).

In DeSurv, survival supervision enters through the projection $Z = X^\top W$ in the partial log-likelihood, so that the gradient of the DeSurv loss with respect to the programs is
$$\nabla\mathcal{L} = (1-\alpha)\nabla_W \mathcal{L}_{\text{Cox}}(W, H) - \alpha\nabla_W \mathcal{L}_{\text{Cox}}(W, \beta),$$
and therefore the update rule for W as in Equation~8 relies on both the reconstruction term and the supervision term. As such, the gene-program definitions are directly shaped by their association with survival. This mechanism amplifies genes aligned with the hazard gradient and suppresses those that dilute prognostic structure, ensuring that the learned programs encode survival-relevant biological variation.

By contrast, if supervision were applied to the sample loadings H, the model could reduce the Cox loss by redistributing patient-specific coefficients while leaving the gene-level programs W nearly unchanged, a behavior documented in multiple supervised variants of NMF and supervised topic models, where supervision applied only to the coefficient matrix modifies H but not the basis W (31–33).

Supervising W also provides a practical advantage for **portability**: because W defines gene-level programs, new samples can be embedded by the closed-form projection $Z_{\text{new}} = X_{\text{new}}^\top W$. In contrast, supervising H would not yield such a mapping; instead, obtaining sample loadings for external datasets would require solving a nonnegative least-squares (NNLS) problem for each new sample, introducing optimization noise and reducing reproducibility.

Together, these considerations motivate supervising through W, which shapes the gene programs toward prognostic directions, stabilizes solutions across initializations, and yields biologically interpretable signatures that generalize across cohorts.

L. Cross-validation procedure.

Algorithm~S3 describes the cross validation procedure with F folds and R initializations. We define the c-index using comparable pairs $\mathcal{P} = \{(i, j) : y_i < y_j,\ \delta_i = 1\}$ and linear predictors $\hat{\eta}_i$:
$$\hat{c} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \left[\nVdash(\hat{\eta}_i > \hat{\eta}_j) + \tfrac{1}{2}\nVdash(\hat{\eta}_i = \hat{\eta}_j)\right]. \qquad [18]$$

Algorithm 3 Cross-validation for DeSurv pipeline

```
Input:
    Number of folds $F$
    Number of initializations $R$
    Observed data $(X, y, \delta)$
    Hyperparameters $k$, $\alpha$, $\lambda$, $\xi$, and $\lambda_H$
Output: The cross-validated c-index
Divide subjects into $F$ folds.
for $f = 1$ to $F$ do
    Split data into training and validation $(X_{(-f)}, y_{(-f)}, \delta_{(-f)})$, and $(X_{(f)}, y_{(f)}, \delta_{(f)})$ sets
    for $r = 1$ to $R$ do
        set.seed($r$)
        Apply Algorithm 1 with inputs $(X_{(-f)}, y_{(-f)}, \delta_{(-f)})$, and $(k, \alpha, \lambda, \xi, \lambda_H)$
        Obtain $\hat{W}$ and $\hat{\beta}$ as output from Algorithm 1
        Compute the estimated linear predictor: $\hat{\eta} = X_{(f)}^T \hat{W} \hat{\beta}$
        Compute c-index, denoted $\hat{c}_{(f)r}$, according to Equation 18
    end loop over $r$
    Compute average c-index across initializations: $\hat{c}_{(f)} = \frac{1}{r} \sum_{r=1}^{R} \hat{c}_{(f)r}$
end loop over $f$
Compute final cross-validated c-index: $\hat{c} = \frac{1}{F} \sum_{f=1}^{F} \hat{c}_{(f)}$
return Final estimate $\hat{c}$
```

M. Bayesian Optimization.

N. PDAC Datasets.

O. Bladder Datasets.

P. Consensus Clustering.

Q. Survival Analysis.

Versioning

```
## DeSurv package version: 1.0.1
## DeSurv git branch: 20260107bugfix
## DeSurv git commit: fea641a96e4743315a35b9b6addcc1a1ff53da9d
## Paper git branch: main
## Paper git commit: d086ac2a82f21a98c4772d5056f2746b736be63a
```

1. Pareja F, et al. (2016) Triple-negative breast cancer: The importance of molecular and histologic subtyping, and recognition of low-grade variants. *NPJ breast cancer* 2(1):1–11.

2. Dienstmann R, Salazar R, Tabernero J (2018) Molecular subtypes and the evolution of treatment decisions in metastatic colorectal cancer. *Am Soc Clin Oncol Educ Book* 38(38):231–8.

3. Zhou X, et al. (2021) Clinical impact of molecular subtyping of pancreatic cancer. *Frontiers in cell and developmental biology* 9:743908.

4. Seiler R, et al. (2017) Impact of molecular subtypes in muscle-invasive bladder cancer on predicting response and survival after neoadjuvant chemotherapy. *European urology* 72(4):544–554.

5. Prat A, et al. (2015) Clinical implications of the intrinsic molecular subtypes of breast cancer. *The Breast* 24:S26–S35.

6. Ou F, Michiels S, Shyr Y, Adjei AA, Oberg AL (2021) Biomarker discovery and validation: Statistical considerations. *Journal of Thoracic Oncology* 16(Suppl 15):S539–S547.

7. Planey Catherine R, Gevaert O (2016) CoINcIDE: A framework for discovery of patient subtypes across multiple datasets. *Genome Medicine* 8(1):27.

8. Prat A, Pineda E, Adamo B, et al. (2014) Molecular features and survival outcomes of the intrinsic subtypes in the international breast cancer study group trial 10-93. *Journal of the National Cancer Institute* 106(8):dju152.

9. Ellrott K, et al. (2025) Classification of non-TCGA cancer samples to TCGA molecular subtypes using compact feature sets. *Cancer cell* 43(2):195–212.

10. Tomczak K, Czerwińska P, Wiznerowicz M (2015) Review the cancer genome atlas (TCGA): An immeasurable source of knowledge. *Contemporary Oncology/Współczesna Onkologia* 2015(1):68–77.

11. Zhang J, et al. (2019) The international cancer genome consortium data portal. *Nature biotechnology* 37(4):367–369.

12. Nguyen H, Nguyen H, Tran D, Draghici S, Nguyen T (2024) Fourteen years of cellular deconvolution: Methodology, applications, technical evaluation and outstanding challenges. *Nucleic Acids Research* 52(9):4761–4783.

13. Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. *nature* 401(6755):788–791.

14. Bailey P, Chang DK, et al. (2016) Genomic analyses identify molecular subtypes of pancreatic cancer. *Nature* 531(7592):47–52.

15. Moffitt RA, et al. (2015) Virtual microdissection identifies distinct tumor-and stroma-specific subtypes of pancreatic ductal adenocarcinoma. *Nature genetics* 47(10):1168–1178.

16. Peng XL, Moffitt RA, Torphy RJ, Volmar KE, Yeh JJ (2019) De novo compartment deconvolution and weight estimation of tumor samples using DECODER. *Nature communications* 10(1):4729.

17. Le Goff V, et al. (2025) SurvNMF: Non-negative matrix factorization supervised for survival data analysis. PhD thesis (Institut Pasteur Paris; CEA).

18. Huang Z, Salama P, Shao W, Zhang J, Huang K (2020) Low-rank reorganization via proportional hazards non-negative matrix factorization unveils survival associated gene clusters. *arXiv preprint arXiv:200803776.*

19. Seung D, Lee L (2001) Algorithms for non-negative matrix factorization. *Advances in neural information processing systems* 13(556-562):35.

20. Pascual-Montano A, Carazo JM, Kochi K, Lehmann D, Pascual-Marqui RD (2006) Nonsmooth nonnegative matrix factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(3):403–415.

21. Simon N, Friedman JH, Hastie T, Tibshirani R (2011) Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of statistical software* 39:1–13.

22. Lin C-J (2007) On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *IEEE Transactions on Neural Networks* 18(6):1589–1596.

23. Tseng P (2001) Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications* 109(3):475–494.

24. Grippo L, Sciandrone M (2000) On the convergence of the block nonlinear gauss–seidel method under convex constraints. *Operations research letters* 26(3):127–136.

25. Donoho D, Stodden V (2004) When does non-negative matrix factorization give a correct decomposition into parts? *Advances in Neural Information Processing Systems.*

26. Laurberg H, Christensen MG, Plumbley MD, Hansen LK, Jensen SH (2008) Theorems on the uniqueness of nonnegative matrix factorization. *Neurocomputing* 71(1-3):606–616.

27. Gillis N (2014) The why and how of nonnegative matrix factorization. *Regularization, Optimization, Kernels, and Support Vector Machines* (Chapman; Hall/CRC), pp 257–291.

28. Brunet J-P, Tamayo P, Golub TR, Mesirov JP (2004) Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences* 101(12):4164–4169.

29. Kim H, Park H (2007) Sparse non-negative matrix factorization for clustering. *Journal of Scientific Computing* 36:205–222.

30. Gillis N, Glineur F (2012) Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization. *Neural Computation* 24(4):1085–1105.

31. Cai D, He X, Han J (2011) Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp 1548–1560.

32. Wang K, Yang P, Yang Y, Sun L (2014) A novel supervised nonnegative matrix factorization algorithm for gene expression classification. *Computational Biology and Chemistry* 53:189–197.

33. Blei DM, McAuliffe JD (2007) Supervised topic models. *Advances in Neural Information Processing Systems.*