# Designing Cell-Type-Specific Regulatory DNA with Annealed Diffusion Beam Search

**Anonymous authors**
Paper under double-blind review

## Abstract

Designing regulatory DNA with cell-type-specific activity is broadly relevant for cell engineering and gene therapy. An ideal AI-designed sequence should drive high expression in desired cell types, have minimal activity in undesired cell types, and adhere to the genome's natural regulatory grammar. Diffusion models have become powerful priors for DNA sequence generation. Inference-time search guided by predictive oracles can steer these models toward high specificity. However, activity scoring requires complete DNA sequences, which must be derived heuristically from partially masked intermediates during search. Positionwise maximum a posteriori (MAP) estimation provides an efficient heuristic for completing these intermediates and computing lookahead scores. Here, we report that these scores are unreliable early in denoising and their reliability improves nonuniformly as sequence context accumulates. Based on this observation, we developed Progress-Annealed Candidate Selection (PACS), a training-free method that schedules how strongly MAP scores influence selection within diffusion beam search. PACS standardizes scores within each parent's candidate pool and increases selection strength with the expected fraction of revealed nucleotides, allowing broader exploration early and a stronger preference for high-scoring sequences later. We provide a local KL-regularized interpretation of the selector, characterize its limiting cases, and establish large-pool consistency under stated assumptions. Across three human cell lines, PACS improves sequence fidelity by 12–16% relative to greedy selection while maintaining high predicted specificity. On an eight-cell-type immune benchmark, PACS outperforms most of the evaluated reinforcement learning, inference-time alignment, and iterative optimization methods.

## 1 Introduction

Precise control of gene expression is fundamental for cell engineering and targeted gene therapy (Dunbar et al., 2018; Kitada et al., 2018; Yang et al., 2025). A central challenge is designing synthetic regulatory DNA sequences, such as enhancers and promoters, that drive high transcriptional activity in desired cell types while minimizing activity in a potentially large number of undesired cell types. This requires optimizing discrete nucleotide sequences across a vast combinatorial space governed by complex regulatory grammar. The combination of two computational advances makes this optimization possible. First, sequence-to-activity neural networks act as oracles and predict chromatin accessibility and expression profiles across hundreds of cell types from sequence alone (Avsec et al., 2021; Linder et al., 2025; Avsec et al., 2026). Second, masked discrete diffusion models learn the distribution of natural regulatory DNA, providing a generative prior that encodes higher-order sequence constraints (Sahoo et al., 2024; Sarkar et al., 2024). Together, these two components enable a design paradigm in which a predictive oracle steers a learned generative prior toward cell-type-specific activity while preserving the genome's natural regulatory grammar.

Inference-time search realizes this paradigm by allocating test-time compute to steer a frozen diffusion model toward sequences with high predicted specificity (Li et al., 2024; Tang et al., 2024; Awasthi et al., 2026). Search algorithms such as beam search expand multiple candidate continuations at each denoising step and select which to carry forward based on their



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

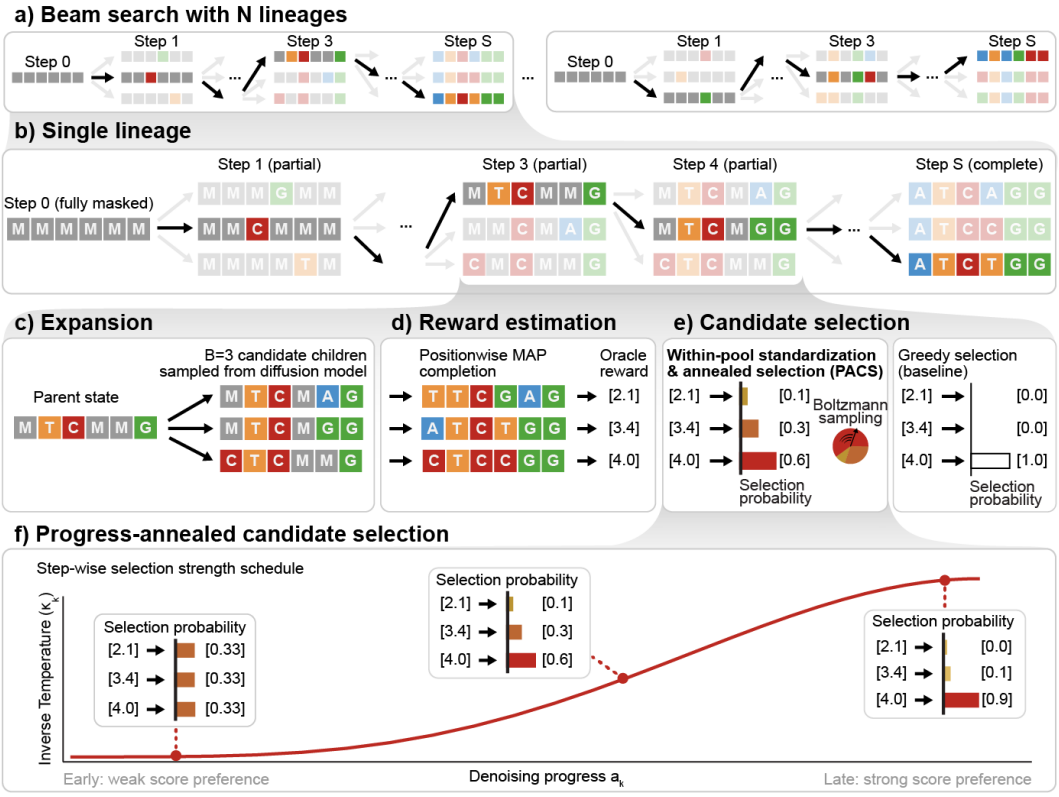


Figure 1: **Overview of Progress-Annealed Candidate Selection (PACS).** (a–b) Beam search maintains $N$ lineages. (c–e) Each parent proposes $B$ children from the reverse-diffusion model. Their positionwise MAP completions are scored by the oracle, and PACS samples one child using within-pool standardized scores instead of greedy selection. Selected partial states continue denoising. (f) Inverse temperature increases with the clean fraction, shifting selection from uniform choice toward stronger preference for high scores.

expected reward Oshima et al. (2025). Evaluating a partially masked candidate requires a surrogate for its achievable reward. Positionwise maximum a posteriori (MAP) lookahead fills each mask with its most probable nucleotide in one denoiser pass, followed by scoring of the resulting sequence using the oracle, thereby avoiding costly ancestral rollouts. However, this surrogate is not equally reliable throughout the search process Jain et al. (2025); Su et al. (2025). Early in denoising, when most positions are still masked, independent per-position completions may poorly capture cooperative motif interactions and yield unreliable rankings. As more context accumulates, MAP completions increasingly agree with the activity scores of the fully unmasked sequences.

Despite this variable reliability, standard discrete beam search applies greedy selection uniformly across all denoising steps. This can prematurely discard promising continuations when early lookahead scores are noisy. Recent inference-time search methods address this by reallocating compute across the trajectory by narrowing the beam or scheduling tree expansions (Oshima et al., 2025; Li et al., 2025; Jain et al., 2025). These strategies modulate where and how much to search while leaving the selection rule itself greedy at every step. A complementary approach is to modulate the selection rule directly, such as coupling selection strength to denoising progress. This motivates the question central to our approach: *How should inference-time search select lineages based on an approximate endpoint value whose reliability changes over the reverse-diffusion trajectory?*

Here we present **Progress-Annealed Candidate Selection** (**PACS**), a training-free method that schedules how strongly MAP lookahead scores influence candidate selection. PACS operates within beam search over the denoising trajectories of a frozen masked discrete diffusion model. PACS initializes $N$ persistent lineages each yielding one reported sequence (Figure 1a,b). At each step, each parent lineage generates its own next-step candidate children pool (Figure 1c). PACS standardizes MAP scores within each pool and

selects a continuation through sampling from a temperature-controlled distribution whose selection strength increases with the expected fraction of revealed nucleotides under the diffusion schedule (Figure 1d,e). This produces near-uniform selection early in generation and increasingly favors high-scoring candidates as denoising progresses (Figure 1f). We show that the selector exactly optimizes a local KL-regularized objective, balancing preference for higher standardized scores against departure from uniform candidate selection. Under stated assumptions, its selected-child distribution converges to a locally value-reweighted reverse transition as the candidate pool grows.

We evaluate PACS on cell-type-specific regulatory DNA design for three human cell lines using a class-conditioned masked discrete diffusion model trained on regulatory elements (Awasthi et al., 2026). In three design tasks, PACS improves sequence fidelity across several metrics while maintaining specificity compared to its greedy-limit during beam search. Increasing the candidate pool raises specificity under both selection rules, while PACS generally retains a fidelity advantage. Evaluation across eight immune cell types validates transfer to new cellular contexts and data modalities, where PACS achieves competitive specificity and improved sequence fidelity relative to the evaluated reinforcement learning (RL), inference-time alignment, and iterative optimization methods.

Our contributions are threefold:

- We introduce PACS, a training-free candidate-selection method that uses progress-dependent selection strength for MAP-valued diffusion beam search.
- We characterize PACS through a local KL-regularized objective, its limiting cases, and large-pool consistency under stated assumptions.
- Across three human cell lines and eight immune cell types, we demonstrate competitive specificity and improved sequence fidelity relative to the evaluated RL, inference-time alignment, and iterative optimization methods.

## 2 Related work

**Inference-time alignment and search.** These methods steer frozen generative models toward high-reward samples without retraining Uehara et al. (2025). For discrete diffusion models, they fall into three categories. Guidance-based methods reweight the reverse transition using gradients of a classifier or a learned value (Nisonoff et al., 2025; Schiff et al., 2025; Ye et al., 2025). Particle-based methods maintain a population and resample it under importance weights (Wu et al., 2024; Singhal et al., 2025; Skreta et al., 2025; Hasan et al., 2026; Phillips et al., 2024; Ou et al., 2026). Tree and beam search-based methods instead expand several continuations per step and commit to a subset (Tang et al., 2024; Yoon et al., 2025; 2026; Jain et al., 2025; Zhang et al., 2025). While guidance and particle-based methods sample from a modified reverse proposal, our approach (PACS) is a search-based method that differs by controlling which continuation each lineage retains.

**Approximate endpoint values and adaptive search.** Many of these methods need an estimate of the terminal reward of a partially masked state. Ancestral rollouts provide one but are costly, so practical systems score a single-pass surrogate instead, typically a MAP or posterior-mean completion evaluated by the oracle (Li et al., 2024; Oshima et al., 2025; Kim et al., 2026), whose reliability is known to degrade at high noise (Jain et al., 2025; Su et al., 2025). Existing approaches address this by reallocating compute across the beam search trajectory. DSearch adapts beam and expansion budgets Li et al. (2025), AdaLook adapts the lookahead cost Cui et al. (2026), and hierarchical schemes prune the active trajectory set on a fixed decay schedule (Bai et al., 2026). Others reshape the intermediate target by tempering (Kim et al., 2025), or by coupling variance-reduced guidance to selection (Kim, 2026). A separate line schedules the guidance weight over the trajectory, first in continuous diffusion (Kynkäänniemi et al., 2024) and recently in masked discrete diffusion (Rojas et al., 2026). None of these modulate the selection rule itself, which stays greedy or fixed-temperature at every denoising step. PACS instead ties selection strength to the clean fraction and anneals it from near-uniform retention toward increasingly greedy retention as sequence context accumulates. The resulting selector exactly optimizes a local KL-

regularized objective, relating it to soft best-of-$n$ alignment (Verdun et al., 2025) but with a schedule-dependent regularization strength.

**AI-driven cell-type-specific regulatory DNA design.** Sequence-optimization methods seek desired activity by using gradients from an oracle (Linder & Seelig, 2021; Schreiber et al., 2025). Generative approaches learn the sequence prior through autoregressive models (Lal et al., 2024), flow matching (Stark et al., 2024), and conditional discrete diffusion (Sarkar et al., 2024; DaSilva et al., 2026). Reward-driven adaptation then steers that prior toward activity by fine-tuning a diffusion backbone (Wang et al., 2025) or by reinforcement learning over an autoregressive generator (Chen et al., 2025), each requiring a training run per task. Closest to our method are training-free approaches that spend compute at inference (Awasthi et al., 2026). PACS retains the simpler beam structure and uses a progress-dependent selector for MAP-evaluated candidates.

# 3 Background and problem formulation

## 3.1 Problem Formulation: Cell-Type-Specific Regulatory DNA Design

Our objective is to design fixed-length regulatory DNA sequences that drive high activity in target cell types while minimizing it in off-target cell types. Let $\mathcal{V} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ denote the nucleotide vocabulary and $\boldsymbol{x} \in \mathcal{V}^L$ a sequence of length $L$. We assume access to a pretrained sequence-to-activity oracle that predicts scalar regulatory activities $s_c(\boldsymbol{x}) \in \mathbb{R}$ across a set of cell types $\mathcal{C}$. We partition $\mathcal{C}$ into disjoint sets of target types $\mathcal{C}^+$ and off-target types $\mathcal{C}^-$. To quantify target activity relative to the strongest predicted off-target activity, we use the **MinGap** objective from (Awasthi et al., 2026):

$$g(\boldsymbol{x}) = \frac{1}{|\mathcal{C}^+|} \sum_{c \in \mathcal{C}^+} s_c(\boldsymbol{x}) - \max_{c \in \mathcal{C}^-} s_c(\boldsymbol{x}). \tag{1}$$

## 3.2 Class-Conditioned Masked Diffusion as the Generative Prior

We employ a class-conditioned masked diffusion model as the generative prior, favoring broad regulatory-element structure (e.g., enhancer vs. promoter elements) via class label $y \in \mathcal{Y}$, while fine-grained cell-type specificity is steered by the search reward following Awasthi et al. (2026). Let $\boldsymbol{m}$ denote an absorbing mask token and $\boldsymbol{z}_t \in (\mathcal{V} \cup \{\boldsymbol{m}\})^L$ a partially masked sequence at continuous diffusion time $t \in [0, 1]$, where $t = 1$ is fully masked and $t = 0$ is fully resolved. A monotonically decreasing schedule $\alpha(t) \in [0, 1]$ specifies the marginal probability that a position is unmasked at time $t$. At each reverse step, a neural denoiser with parameters $\theta$ predicts nucleotide logits across masked positions. We incorporate classifier-free guidance (CFG) with scale $\gamma \geq 0$ to weight the element class condition $y$ (Schiff et al., 2025), yielding positionwise categorical distributions $d^{\gamma}_{\theta,\ell}(a \mid \boldsymbol{z}_t, t, y)$ over $a \in \mathcal{V}$. The factorized reverse proposal reveals each masked position independently according to the guided denoiser distribution, conditional on being revealed, while already visible nucleotides are deterministically retained (more details in Appendix A.2).

# 4 Methods

## 4.1 Beam Search with MAP Lookahead

To navigate the irreversible unmasking decisions, we augment standard ancestral sampling with beam search. The search maintains a population of $N$ parallel, independent lineages (parents) and explores a candidate pool of $B$ prospective continuations (siblings) per parent at each reverse step. Starting from the fully masked initialization $\boldsymbol{z}^{(n)}_{t_0} = \boldsymbol{m}^L$ for each lineage $n \in \{1, \ldots, N\}$, the transition from parent time $t$ to child time $s < t$ proceeds as follows. (i) For each parent, $B$ partially unmasked child states are sampled independently from the reverse proposal. (ii) Since the oracle scores only fully resolved sequences, each candidate is completed by copying its revealed positions and filling every masked position with the denoiser's position-wise MAP prediction:

$$\widehat{x}^{\ell}(\boldsymbol{z}_s) = \arg\max_{a \in \mathcal{V}} d^{\gamma}_{\theta,\ell}(a \mid \boldsymbol{z}_s, s, y) \quad \text{for } \boldsymbol{z}^{\ell}_s = \boldsymbol{m}, \qquad v_s(\boldsymbol{z}_s) = r\big(\widehat{\boldsymbol{x}}(\boldsymbol{z}_s)\big). \tag{2}$$

The scalar lookahead score $v_s(\boldsymbol{z}_s)$ serves as a computationally efficient proxy for child quality.

(iii) The highest-scoring sibling, $J_n^\star = \arg\max_j v_s(\boldsymbol{z}_s^{(n,j)})$, survives and becomes the parent for the next reverse transition. We also maintain a lineage-specific archive that stores the sequence with highest reward. After the final reverse step, we report the highest-scoring archived completion from each lineage. This archive does not affect the search and selecting a sequence from it is a architectural design choice. Implementation details of the beam search are in Appendix B.1

### 4.2 Diagnosing the Reliability of MAP Rankings

In our beam search formulation, candidate children compete strictly against their siblings for a given parent. Therefore, MAP lookahead should preserve the ranking of candidates within each sibling pool. We show that this ranking signal is weak early in denoising, when little sequence context is available, and strengthens as more nucleotides are revealed. To confirm this, we sample sibling pools at intermediate diffusion checkpoints and compare each candidate's MAP score with its ancestral continuation value, estimated as the mean terminal reward over independent ancestral rollouts from that candidate. We compute Spearman correlation within each sibling pool and then average across parents. This provides a measure of local ranking agreement on sampled greedy trajectories. Experimental details and independent-rollout-batch comparisons are in Appendix E.1.

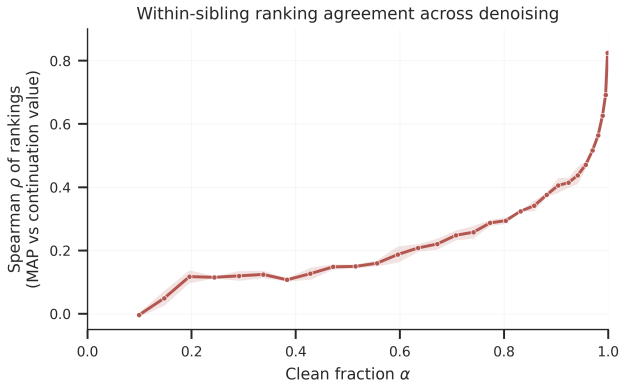


Figure 2: **MAP ranking agreement across denoising.** Within-sibling Spearman correlation between MAP lookaheads and continuation values estimated using ancestral rollouts per candidate. Points show means across three seeds. Shading denotes one standard deviation. The x-axis is the fraction of unmasked nucleotides.

Figure 2 shows weak ranking agreement early in denoising and substantially stronger agreement near completion. Thus, the amount of useful selection signal provided by MAP increases over the reverse trajectory, motivating a selection rule that places progressively greater weight on MAP scores as sequence context accumulates.

### 4.3 PACS: Sibling-Standardized Stochastic Selection

In PACS, we replace the per-parent greedy selection rule with stochastic selection with step-dependent strength across denoising steps. Since the scale and spread of MAP lookahead scores vary across steps and parent lineages, applying a fixed Boltzmann temperature to raw scores would impose inconsistent selection pressure across pools. We resolve this by standardizing each candidate's score within its sibling pool before selection.

Dropping the parent index, let $v_j$ denote sibling $j$'s MAP score for $j \in \{1, \dots, B\}$, with pool mean $\bar{v}$ and standard deviation $\sigma_B$ (population denominator). Given a step-dependent selection strength $\kappa_k \geq 0$, each parent independently draws one surviving index from the standardized Boltzmann distribution:

$$u_j = \frac{v_j - \bar{v}}{\sigma_B + \delta}, \qquad \pi_j = \frac{\exp\{\kappa_k u_j\}}{\sum_{\ell=1}^{B} \exp\{\kappa_k u_\ell\}}, \qquad J \sim \text{Categorical}(\pi_1, \dots, \pi_B), \tag{3}$$

with $\delta = 10^{-6}$ guarding against a degenerate pool. The parameter $\kappa_k$ acts as a dimensionless inverse temperature. $\kappa_k = 0$ yields uniform exploration, while $\kappa_k \to \infty$ recovers greedy selection. Since centering cancels in the softmax, standardization effectively rescales the inverse temperature to $\kappa_k/(\sigma_B + \delta)$, conferring invariance to additive score offsets and

approximate invariance to multiplicative rescaling across pools (Appendix C.2). A single temperature hyperparameter could therefore transfer across cell types and reward functions without per-task recalibration.

### 4.4 PACS: Scheduling Selection Strength by Clean Fraction

Section 4.2 showed that MAP ranking agreement improves with the clean fraction $\alpha$. We therefore parameterize $\kappa_k$ as a function of the clean fraction. Since the clean fraction increases with denoising progress, we call our method progress-annealed candidate selection.

Let $s_k$ denote the child time at transition grid $k$ of the reverse grid (Section 3.2). We define a normalized generative progress $a_k \in [0, 1]$, and $S$ the number of diffusion steps, and set the step-wise selection strength via a powered cubic smoothstep:

$$a_k = \mathrm{clip}_{[0,1]}\left(\frac{\alpha(s_k) - \alpha(s_0)}{\alpha(s_{S-1}) - \alpha(s_0)}\right), \qquad \kappa_k = \kappa_{\max}\big(3a_k^2 - 2a_k^3\big)^p, \tag{4}$$

with maximum selection strength $\kappa_{\max} > 0$ and exponent $p \in \mathbb{N}^+$. The smoothstep begins at zero stringency and saturates at $\kappa_{\max}$. Raising it to the power $p$ extends the early exploratory phase. We use $p = 2$ and $\kappa_{\max} = 40$ across both benchamarks. Alternative settings yield different operating points, and no configuration is uniformly optimal across targets and metrics. We parameterize selection strength by the schedule-defined clean fraction to connect candidate selection to denoising progress. Appendix E.2 further compares this choice with step-indexed schedules and alternative functional forms, revealing target-dependent specificity–fidelity trade-offs. The resulting complete algorithm for PACS is in Appendix B.3.

### 4.5 Theoretical Interpretation

The standardized Boltzmann sampler (Equation 3) is the unique solution to a local KL-regularized objective. For fixed standardized scores $\{u_j\}$ and selection strength $\kappa \geq 0$, the distribution $\pi_j \propto \exp\{\kappa u_j\}$ uniquely solves

$$\max_{p \in \Delta_B} \left\{ \kappa \textstyle\sum_j p_j\, u_j \;-\; \mathrm{KL}(p \,\|\, \mu) \right\},$$

where $\mu$ is the uniform selection reference over the sibling pool (Appendix C). Each reverse step therefore trades off expected standardized MAP score against deviation from uniform exploration, with the annealing schedule $\kappa_k$ controlling the balance. Following this interpretation, standard beam search corresponds to $\kappa \to \infty$ resulting in greedy selection while $\kappa = 0$ recovers the underlying diffusion proposal. PACS smoothly interpolates between these regimes, progressively increasing selection pressure as sequence context accumulates during denoising.

For a fixed parent, finite $\kappa$, and $\delta > 0$, the conditional selected-child law converges almost surely in total variation to a local MAP-score-tilted proposal as $B \to \infty$. Under the finite-support assumptions of Proposition 1, the unconditional law has an $O(B^{-1/2})$ total-variation error bound. This is a local selection result, not a guarantee of sampling an ideal global terminal-reward tilt or of the distribution of lineage-best outputs (Appendix C).

## 5 Experiments

### 5.1 Experimental setup

**Design tasks and predictive models.** We evaluate PACS on designing 200-bp regulatory DNA sequences for two tasks. (i) The human cell-line benchmark uses approximately 700,000 regulatory DNA sequences with massively parallel reporter assay (MPRA) measurements in HepG2, K562, and SK-N-SH (Gosai et al., 2024). (ii) The immune-cell benchmark targets differential chromatin accessibility across eight cell types: $CD4^+$ T cells, $CD8^+$ T cells, macrophages, mast cells, memory B cells, natural killer T cells, naive T cells, and

plasma B cells Zhang et al. (2021). Each task targets one cell type while minimizing activity in the remaining types. We use naturally occurring regions with target-specific activity as fidelity references. We follow a data splitting protocol that partitions the dataset into two homology-aware halves with MMseqs2 Steinegger & Söding (2017) and fine-tune separate oracles on each partition. We use a fine-tuned Enformer as `Design-Model`, and separate fine-tuned Enformer, Borzoi and AlphaGenome models as `Evaluation-Model` for final scoring (Avsec et al., 2021; Linder et al., 2025; Avsec et al., 2026).

**Generative backbone and design objective.** We use a class-conditioned masked diffusion model pretrained on approximately 3.2 million human and mouse candidate regulatory elements (Awasthi et al., 2026). The backbone is frozen during search and conditioned on the enhancer class. To smooth the optimization, we use an objective that replaces the maximum off-target term in MinGap (Equation 1) with a log-sum-exp penalty.

**Benchmarked methods.** We compare PACS with greedy selection, its infinite-limit control ($B = 2048$ for both); gradient-based sequence-optimization method (Ledidi, (Schreiber et al., 2025)); RL-based fine-tuning methods (DRAKES and Ctrl-DNA, (Wang et al., 2025; Chen et al., 2025)); class-conditioned discrete diffusion (D3, (Sarkar et al., 2024)); and inference-time alignment methods sharing the same backbone and objective (sequential Monte Carlo (SMC), twisted diffusion sampling (TDS), soft value-based decoding (SVDD), and discrete Feynman–Kac correctors, (Phillips et al., 2024; Wu et al., 2024; Li et al., 2024; Hasan et al., 2026)). We provide implementation details and baseline evaluations in Appendix D.4 and Appendix E.3.

**Evaluation metrics.** We determine specificity as the mean per-sequence MinGap computed using the AlphaGenome `Evaluation-Model` since it performs the best among other models (Table 2). We assess sequence fidelity by the Spearman correlation of transcription factor binding site (TFBS) frequencies using FIMO Bailey et al. (2015) and JASPAR motifs Rauluseviciute et al. (2024), 3-mer-frequency Pearson correlation, and Fréchet Biological Distance (FBD) in Nucleotide Transformer v2 (NTv2) embedding space Dalla-Torre et al. (2025). We further evaluate TFBS pair co-occurrence, inter-motif spacing, and relative orientation, to assess arrangement beyond marginal motif abundance. Diversity is assessed separately through mean positional nucleotide entropy.

### 5.2 PACS improves sequence fidelity while maintaining high specificity

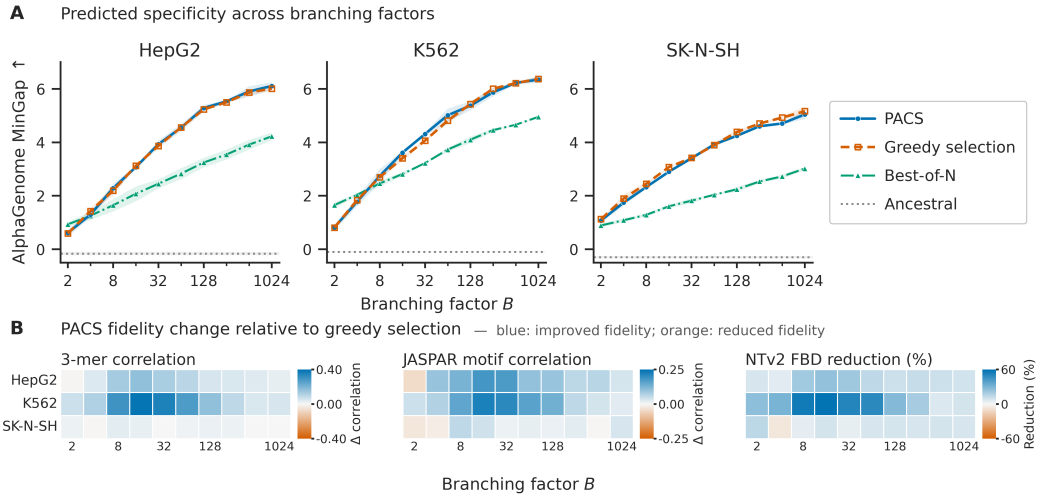


Figure 3: **Specificity and fidelity across branching factors. (A)** AlphaGenome MinGap at matched oracle-evaluation budgets, with ancestral sampling as an unguided reference. Shading indicates standard deviation across three runs of 128 sequences. **(B)** PACS fidelity changes relative to the greedy-limit control, shown as correlation differences and percentage FBD reduction. Blue indicates improvement; orange indicates deterioration.

To isolate the effect of candidate selection, we compare PACS with ancestral sampling (no reward-based selection), best-of-$N$ sampling (matched oracle calls), and greedy-limit

control baselines, keeping model parameters identical. Figure 3 shows predicted specificity and fidelity as the branching factor increases for three cell lines. The results suggest that the mean AlphaGenome MinGap increases monotonically with $B$ under PACS across all three cell lines. Both PACS and its greedy-limit control consistently exceed best-of-$N$ specificity at every evaluated setting with $B \geq 8$. At the same time, PACS improves sequence fidelity relative to greedy selection across a broad range of branching factors. In K562 and HepG2, TF motif frequency correlation, 3-mer correlation, and NTv2 FBD improve at almost every evaluated $B$. The gains are particularly pronounced at intermediate branching factors with detailed results given in F.1. In K562 at $B = 16$, PACS increases motif correlation from 0.692 to 0.914 and 3-mer correlation from 0.571 to 0.961, while reducing FBD by 59.2% and increasing MinGap from 3.402 to 3.611. Improvements in SK-N-SH remain smaller and less consistent. Although the fidelity advantage generally narrows at larger branching factors, all three metrics remain improved across all three cell lines at $B = 1024$. Together, these findings support adaptive selection as a means of improving the specificity–fidelity trade-off across search scales, with benefits that depend on the target cell type and branching factor.

### 5.3 Evaluating PACS on regulatory DNA design across human cell lines

| | | SMC | TDS | SVDD-PM | DFKC | Ledidi | DRAKES | D3 | Ctrl-DNA | **Greedy selection** | **PACS** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **AlphaGenome MinGap ↑** | HepG2 | 1.14 | 0.99 | 2.30 | 4.91 | 5.98 | -1.35 | -0.09 | 5.97 | 6.29 | 6.37 |
| | K562 | 2.84 | 0.24 | 2.85 | 6.27 | 6.47 | 0.08 | -0.00 | 6.22 | 6.39 | 6.46 |
| | SK-N-SH | 0.20 | 0.65 | 2.21 | 3.80 | 3.21 | -0.15 | -0.13 | 3.57 | 5.39 | 5.12 |
| **JASPAR motif correlation ↑** | HepG2 | 0.290 | 0.330 | 0.632 | 0.448 | 0.539 | -0.126 | 0.793 | 0.545 | 0.830 | 0.834 |
| | K562 | 0.408 | 0.430 | 0.664 | 0.531 | 0.650 | 0.060 | 0.794 | 0.528 | 0.837 | 0.865 |
| | SK-N-SH | 0.500 | 0.343 | 0.796 | 0.504 | 0.491 | -0.001 | 0.766 | 0.580 | 0.784 | 0.819 |
| **Pairwise TFBS co-occurrence ↑** | HepG2 | 0.117 | 0.027 | 0.404 | 0.279 | 0.338 | 0.225 | 0.271 | 0.208 | 0.349 | 0.326 |
| | K562 | 0.189 | 0.079 | 0.345 | 0.201 | 0.253 | 0.341 | 0.409 | 0.069 | 0.318 | 0.385 |
| | SK-N-SH | 0.162 | 0.045 | 0.637 | 0.315 | 0.465 | 0.140 | 0.432 | 0.428 | 0.512 | 0.563 |
| **Inter-motif spacing JSD ↓** | HepG2 | 0.141 | 0.253 | 0.007 | 0.020 | 0.009 | 0.037 | 0.022 | 0.022 | 0.012 | 0.011 |
| | K562 | 0.083 | 0.164 | 0.006 | 0.033 | 0.015 | 0.025 | 0.021 | 0.053 | 0.014 | 0.015 |
| | SK-N-SH | 0.086 | 0.188 | 0.013 | 0.038 | 0.024 | 0.066 | 0.033 | 0.042 | 0.008 | 0.011 |
| **Relative motif orientation correlation ↑** | HepG2 | 0.480 | 0.577 | 0.620 | 0.598 | 0.764 | 0.686 | 0.452 | 0.455 | 0.663 | 0.722 |
| | K562 | 0.823 | 0.591 | 0.775 | 0.785 | 0.790 | 0.664 | 0.713 | 0.759 | 0.899 | 0.847 |
| | SK-N-SH | 0.713 | 0.806 | 0.910 | 0.880 | 0.902 | — | 0.711 | 0.871 | 0.896 | 0.877 |
| **3-mer correlation ↑** | HepG2 | 0.603 | 0.622 | 0.842 | 0.672 | 0.685 | -0.361 | 0.977 | 0.568 | 0.932 | 0.960 |
| | K562 | 0.687 | 0.351 | 0.515 | 0.624 | 0.718 | -0.354 | 0.969 | 0.388 | 0.901 | 0.942 |
| | SK-N-SH | 0.766 | 0.713 | 0.923 | 0.508 | 0.524 | -0.382 | 0.932 | 0.431 | 0.897 | 0.915 |
| **NTv2 FBD ↓** | HepG2 | 4.98 | 4.77 | 2.07 | 3.35 | 2.44 | 13.23 | 1.63 | 3.75 | 2.04 | 1.79 |
| | K562 | 4.56 | 10.02 | 4.94 | 3.30 | 2.27 | 8.67 | 1.67 | 4.91 | 2.10 | 1.76 |
| | SK-N-SH | 5.36 | 5.58 | 2.00 | 3.72 | 2.58 | 18.78 | 1.78 | 4.87 | 2.40 | 2.13 |
| **Shannon diversity ↑** | HepG2 | 1.234 | 0.833 | 1.981 | 1.926 | 1.981 | 1.864 | 1.978 | 1.914 | 1.981 | 1.981 |
| | K562 | 0.913 | 0.482 | 1.936 | 1.929 | 1.982 | 1.958 | 1.980 | 1.851 | 1.979 | 1.980 |
| | SK-N-SH | 1.515 | 0.772 | 1.982 | 1.932 | 1.981 | 1.826 | 1.969 | 1.847 | 1.971 | 1.972 |

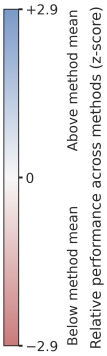


Figure 4: **Regulatory DNA design across three human cell lines.** Annotations show means across three runs of 128 sequences. Colours indicate direction-adjusted scores standardized across the ten methods separately for each metric and cell type; blue indicates above-average performance and red indicates below-average performance. Arrows indicate preferred metric directions, and the dash marks an unavailable measurement.

Figure 4 compares predicted specificity and sequence fidelity across three human cell-line design tasks. PACS achieves the highest mean AlphaGenome MinGap in HepG2, closely approaches Ledidi in K562 (6.463 versus 6.475), and ranks second to greedy selection in SK-N-SH (5.125 versus 5.387). Relative to its greedy-limit control, PACS improves JASPAR motif correlation, 3-mer correlation, and NTv2 FBD in all three cell lines, reducing FBD by 12.3%, 16.0%, and 11.5% in HepG2, K562, and SK-N-SH, respectively. Figure 8 in Appendix F.2 visualizes these specificity–fidelity trade-offs across methods. PACS also achieves the highest mean JASPAR motif correlation across all three cell lines. Although D3 attains higher 3-mer correlations and lower FBD, its negative MinGap scores illustrate that distributional fidelity alone does not ensure cell-type specificity. Arrangement-aware metrics show that PACS improves TFBS co-occurrence over greedy selection in K562 and SK-N-SH, whereas spacing and orientation improve only in HepG2. Together, these results support PACS as a means of improving sequence-composition and embedding fidelity while retaining competitive predicted specificity, with target-dependent effects on motif arrangement.

### 5.4 Evaluating PACS on regulatory DNA design across immune cell types

We next evaluated PACS across eight immune cell types to assess transfer new cell types and chromatin accessibility prediction without fine-tuning (Figure 5, full results in Appendix F.3). PACS achieves positive mean AlphaGenome MinGap across all eight targets and exceeds every non-greedy selection approach except CD8$^+$ T cells where Ctrl-DNA achieves higher MinGap. The CD4$^+$ T-cell task illustrates PACS's utility for distinguishing closely related cellular states. Both PACS and its greedy-limit control achieve positive mean MinGap, while PACS raises JASPAR motif correlation from 0.580 to 0.745, 3-mer correlation from 0.783 to 0.868, and reduces FBD by 30.9% compared to the greedy baseline. Across all eight targets, PACS improves JASPAR motif correlation, 3-mer correlation, and NTv2 FBD relative to greedy selection, with FBD reductions ranging from 5.7% to 36.2%. These gains also extend to motif arrangement metrics. TFBS co-occurrence improves in seven targets, spacing JSD in six, and orientation correlation in five. Per-position Shannon diversity is maintained or increased. Together, these results extend the benefits of adaptive selection to new cellular contexts and a chromatin accessibility-based design objective, supporting its ability to generate sequences with competitive predicted specificity and improved fidelity across complementary measurements.

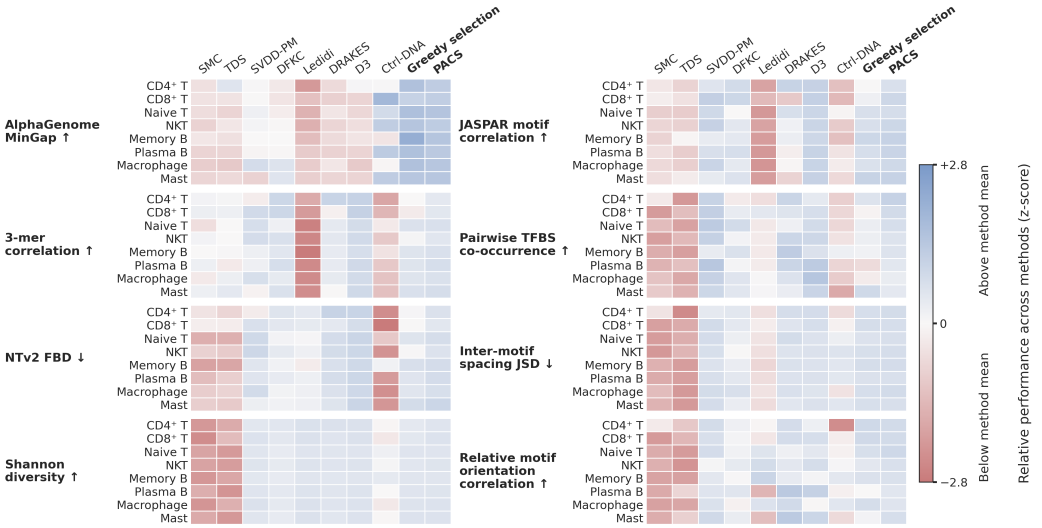


Figure 5: **Regulatory DNA design across eight immune cell types.** Colours show direction-adjusted mean scores standardized across the ten methods separately for each metric and cell type. Blue indicates above-average performance and red indicates below-average performance; arrows indicate preferred metric directions.

## 6 Conclusion and Limitations

PACS is a training-free candidate-selection method for masked discrete diffusion beam search. Across the two main benchmarks, it improves motif-frequency correlation, 3-mer correlation, and embedding fidelity relative to its greedy-limit control, with target-dependent changes in predicted specificity. The designs remain computational predictions, may reflect oracle-specific biases, and will eventually require experimental validation. While our progress-based annealing schedule is methodologically supported by ablations, future work could investigate selection rules informed by candidate-specific uncertainty. Finally, each beam-search step evaluates the denoiser and oracle for every sibling candidate, so inference cost scales linearly with beam width and output count. Distilling the search procedure into the diffusion model with additional training could reduce inference costs for large DNA libraries.

### AI use statement

Generative AI was used during manuscript preparation to review grammar and text. It was also used for retrieving papers relevant to this study. The authors manually checked the paper suggestions against the source material and take responsibility for all text, claims, figures, and other content in the submission.

### Ethics statement

This work studies computational design of cell-type-specific regulatory DNA using public data. It does not report experiments in human participants or the synthesis and testing of the proposed sequences. Activity and specificity predictions should be experimentally validated for the designed sequences.

### Reproducibility statement

The main text specifies the design objective, the PACS selector, and the evaluation criteria. The appendix provides the reverse proposal and sampling grid (Appendix A.2), the complete algorithm and search configuration (Appendices B.3 and D.3), data partitioning and oracle training (Appendices D.1 and D.2), baseline implementations (Appendix D.4), metric definitions and natural-reference construction (Appendix D.5), complete numerical results (Appendices F.1, F.2, and F.3), and proofs (Appendix C). We will release the complete code and model weights with the final manuscript.

## References

Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R. Ledsam, Agnieszka Grabska-Barwinska, Kyle R. Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R. Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods*, 18(10):1196–1203, October 2021. ISSN 1548-7105. doi: 10.1038/s41592-021-01252-x. URL https://www.nature.com/articles/s41592-021-01252-x. Publisher: Nature Publishing Group.

Žiga Avsec, Natasha Latysheva, Jun Cheng, Guido Novati, Kyle R. Taylor, Tom Ward, Clare Bycroft, Lauren Nicolaisen, Eirini Arvaniti, Joshua Pan, Raina Thomas, Vincent Dutordoir, Matteo Perino, Soham De, Alexander Karollus, Adam Gayoso, Toby Sargeant, Anne Mottram, Lai Hong Wong, Pavol Drotár, Adam Kosiorek, Andrew Senior, Richard Tanburn, Taylor Applebaum, Souradeep Basu, Demis Hassabis, and Pushmeet Kohli. Advancing regulatory variant effect prediction with AlphaGenome. 649(8099):1206–1218, 2026. ISSN 1476-4687. doi: 10.1038/s41586-025-10014-0. URL https://www.nature.com/articles/s41586-025-10014-0.

Animesh Awasthi, Raphael Bednarsky, Moritz Schaefer, and Christoph Bock. Conditional monte carlo tree diffusion for designing cell-type-specific and biologically faithful regulatory dna, 2026. URL https://arxiv.org/abs/2604.20488.

Jinbin Bai, Yixuan Li, Yuchen Zhu, Yi Xin, Qingyu Shi, Aosong Feng, Xiaohong Liu, Molei Tao, Jianru Xue, Xiangtai Li, and Ming-Hsuan Yang. Prism: Efficient test-time scaling via hierarchical search and self-verification for discrete diffusion language models, 2026. URL https://arxiv.org/abs/2602.01842.

Timothy L Bailey, James Johnson, Charles E Grant, and William S Noble. The meme suite. *Nucleic acids research*, 43(W1):W39–W49, 2015.

Danila Bredikhin, Alejandro Buendia, Martin Kjellberg, Christopher Zou, Xinming Tu, and Anshul Kundaje. Porting AlphaGenome to PyTorch. *Genomics x AI Blog*, March 2026. doi: 10.5281/zenodo.20272906. URL https://genomicsxai.github.io/blogs/2026-004/.

Xingyu Chen, Shihao Ma, Runsheng Lin, Jiecong Lin, and Bo Wang. Ctrl-DNA: Controllable Cell-Type-Specific Regulatory DNA Design via Constrained RL, May 2025. URL http://arxiv.org/abs/2505.20578. arXiv:2505.20578 [cs].

Yingqian Cui, Wei Deng, Lantao Mei, Hang Li, Charu C. Aggarwal, Hui Liu, and Yue Xing. Adaptive multi-step lookahead decoding for diffusion language models, 2026. URL https://arxiv.org/abs/2607.15655.

Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P. de Almeida, Hassan Sirelkhatim, Guillaume Richard, Marcin Skwark, Karim Beguir, Marie Lopez, and Thomas Pierrot. Nucleotide Transformer: building and evaluating robust foundation models for human genomics. *Nature Methods*, 22 (2):287–297, February 2025. ISSN 1548-7105. doi: 10.1038/s41592-024-02523-z. URL https://www.nature.com/articles/s41592-024-02523-z.

Lucas Ferreira DaSilva, Simon Senan, Judith F. Kribelbauer-Swietek, Zain Munir Patel, Lithin Karmel Louis, Aniketh Janardhan Reddy, Sameer Gabbita, Jonathan D. Rosen, Zach Nussbaum, César Miguel Valdez Córdova, Aaron Wenteler, Noah Weber, Tin M. Tunjic, Martino Mansoldo, Talha Ahmad Khan, Gue-Ho Hwang, Vincent Gardeux, David T. Humphreys, Cameron Smith, Matei Bejan, Peter Bromley, Will Connell, Bart Deplancke, Michael I. Love, Emily S. Wong, Wouter Meuleman, and Luca Pinello. Designing synthetic regulatory elements using the generative AI framework DNA-Diffusion. *Nature Genetics*, 58(1):180–194, January 2026. ISSN 1546-1718. doi: 10.1038/s41588-025-02441-6. URL https://www.nature.com/articles/s41588-025-02441-6. Publisher: Nature Publishing Group.

Cynthia E. Dunbar, Katherine A. High, J. Keith Joung, Donald B. Kohn, Keiya Ozawa, and Michel Sadelain. Gene therapy comes of age. *Science*, 359(6372):eaan4672, January 2018. doi: 10.1126/science.aan4672. URL https://www.science.org/doi/10.1126/science.aan4672. Publisher: American Association for the Advancement of Science.

Sager J Gosai, Rodrigo I Castro, Natalia Fuentes, John C Butts, Kousuke Mouri, Michael Alasoadura, Susan Kales, Thanh Thanh L Nguyen, Ramil R Noche, Arya S Rao, et al. Machine-guided design of cell-type-targeting cis-regulatory elements. *Nature*, pp. 1–10, 2024.

Mohsin Hasan, Viktor Ohanesian, Artem Gazizov, Yoshua Bengio, Alán Aspuru-Guzik, Roberto Bondesan, Marta Skreta, and Kirill Neklyudov. Discrete feynman-kac correctors, 2026. URL https://arxiv.org/abs/2601.10403.

Vineet Jain, Kusha Sareen, Mohammad Pedramfar, and Siamak Ravanbakhsh. Diffusion tree sampling: Scalable inference-time alignment of diffusion models, 2025. URL https://arxiv.org/abs/2506.20701.

Kwanyoung Kim. Vgas: Variance-reduced guidance and adaptive selection for training-free reward alignment in discrete diffusion, 2026. URL https://arxiv.org/abs/2608.26585.

Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward over-optimization, 2025. URL https://arxiv.org/abs/2501.05803.

Yeongmin Kim, Donghyeok Shin, Byeonghu Na, Minsang Park, Richard Lee Kim, and Il-Chul Moon. Lookahead sample reward guidance for test-time scaling of diffusion models, 2026. URL https://arxiv.org/abs/2602.03211.

Tasuku Kitada, Breanna DiAndreth, Brian Teague, and Ron Weiss. Programming gene and engineered-cell therapies with synthetic biology. *Science*, 359(6376):eaad1067, 2018. doi: 10.1126/science.aad1067. URL https://www.science.org/doi/abs/10.1126/science.aad1067.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models, 2024. URL https://arxiv.org/abs/2404.07724.

Avantika Lal, David Garfield, Tommaso Biancalani, and Gokcen Eraslan. Designing realistic regulatory dna with autoregressive language models. *Genome Research*, 34(9):1411–1420, 2024.

Avantika Lal, Laura Gunsalus, Surag Nair, Tommaso Biancalani, and Gokcen Eraslan. gReLU: a comprehensive framework for DNA sequence modeling and design. *Nature Methods*, 22(11):2253–2257, November 2025. ISSN 1548-7105. doi: 10.1038/s41592-025-02868-z. URL https://www.nature.com/articles/s41592-025-02868-z. Publisher: Nature Publishing Group.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, and Masatoshi Uehara. Derivative-Free Guidance in Continuous and Discrete Diffusion Models with Soft Value-Based Decoding, October 2024. URL http://arxiv.org/abs/2408.08252. arXiv:2408.08252 [cs].

Xiner Li, Masatoshi Uehara, Xingyu Su, Gabriele Scalia, Tommaso Biancalani, Aviv Regev, Sergey Levine, and Shuiwang Ji. Dynamic search for inference-time alignment in diffusion models, 2025. URL https://arxiv.org/abs/2503.02039.

Johannes Linder and Georg Seelig. Fast activation maximization for molecular sequence design. *BMC bioinformatics*, 22(1):510, October 2021. ISSN 1471-2105. doi: 10.1186/s12859-021-04437-5.

Johannes Linder, Divyanshi Srivastava, Han Yuan, Vikram Agarwal, and David R. Kelley. Predicting RNA-seq coverage from DNA sequence as a unifying model of gene regulation. *Nature Genetics*, 57(4):949–961, April 2025. ISSN 1546-1718. doi: 10.1038/s41588-024-02053-6. URL https://www.nature.com/articles/s41588-024-02053-6. Publisher: Nature Publishing Group.

Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking Guidance for Discrete State-Space Diffusion and Flow Models, March 2025. URL http://arxiv.org/abs/2406.01572. arXiv:2406.01572 [cs].

Yuta Oshima, Masahiro Suzuki, Yutaka Matsuo, and Hiroki Furuta. Inference-time text-to-video alignment with diffusion latent beam search, 2025. URL https://arxiv.org/abs/2501.19252.

Zijing Ou, Chinmay Pani, and Yingzhen Li. Inference-time scaling of discrete diffusion models via importance weighting and optimal proposal design, 2026. URL https://arxiv.org/abs/2505.22524.

Angus Phillips, Hai-Dang Dau, Michael John Hutchinson, Valentin De Bortoli, George Deligiannidis, and Arnaud Doucet. Particle Denoising Diffusion Sampler, June 2024. URL http://arxiv.org/abs/2402.06320. arXiv:2402.06320 [stat].

Ieva Rauluseviciute, Rafael Riudavets-Puig, Romain Blanc-Mathieu, Jaime A Castro-Mondragon, Katalin Ferenc, Vipin Kumar, Roza Berhanu Lemma, Jérémy Lucas, Jeanne Chéneby, Damir Baranasic, Aziz Khan, Oriol Fornes, Sveinung Gundersen, Morten Johansen, Eivind Hovig, Boris Lenhard, Albin Sandelin, Wyeth W Wasserman, Fran**c**cois Parcy, and Anthony Mathelier. JASPAR 2024: 20th anniversary of the open-access database of transcription factor binding profiles. *Nucleic Acids Research*, 52(D1):D174–D182, January 2024. ISSN 0305-1048. doi: 10.1093/nar/gkad1059. URL https://doi.org/10.1093/nar/gkad1059.

Kevin Rojas, Ye He, Chieh-Hsin Lai, Yuhta Takida, Yuki Mitsufuji, and Molei Tao. Improving classifier-free guidance in masked diffusion: Low-dim theoretical insights with high-dim impact, 2026. URL https://arxiv.org/abs/2507.08965.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and Effective Masked Diffusion Language Models, November 2024. URL http://arxiv.org/abs/2406.07524. arXiv:2406.07524 [cs].

Anirban Sarkar, Ziqi Tang, Chris Zhao, and Peter K Koo. Designing dna with tunable regulatory activity using discrete diffusion. *bioRxiv*, pp. 2024–05, 2024.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-torre, Bernardo P. de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple Guidance Mechanisms for Discrete Diffusion Models, May 2025. URL http://arxiv.org/abs/2412.10193. arXiv:2412.10193 [cs].

Jacob Schreiber, Franziska Katharina Lorbeer, Monika Heinzl, Franziska Reiter, Baptiste Rafanel, Yang Young Lu, Alexander Stark, and William Stafford Noble. Programmatic design and editing of cis-regulatory elements, December 2025. URL https://www.biorxiv.org/content/10.1101/2025.04.22.650035v2. ISSN: 2692-8205 Pages: 2025.04.22.650035 Section: New Results.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models, 2025. URL https://arxiv.org/abs/2501.06848.

Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alán Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-kac correctors in diffusion: Annealing, guidance, and product of experts, 2025. URL https://arxiv.org/abs/2503.02819.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to dna sequence design, 2024. URL https://arxiv.org/abs/2402.05841.

Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, November 2017. ISSN 1546-1696. doi: 10.1038/nbt.3988. URL https://www.nature.com/articles/nbt.3988.

Xun Su, Jianming Huang, Yang Yusen, Zhongxi Fang, and Hiroyuki Kasai. Navigating the exploration-exploitation tradeoff in inference-time scaling of diffusion models, 2025. URL https://arxiv.org/abs/2508.12361.

Sophia Tang, Yinuo Zhang, and Pranam Chatterjee. PepTune: De Novo Generation of Therapeutic Peptides with Multi-Objective-Guided Discrete Diffusion, December 2024. URL https://arxiv.org/abs/2412.17780v4.

Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review, 2025. URL https://arxiv.org/abs/2501.09685.

Claudio Mayrink Verdun, Alex Oesterling, Himabindu Lakkaraju, and Flavio P. Calmon. Soft best-of-n sampling for model alignment, 2025. URL https://arxiv.org/abs/2505.03156.

Chenyu Wang, Masatoshi Uehara, Yichun He, Amy Wang, Tommaso Biancalani, Avantika Lal, Tommi Jaakkola, Sergey Levine, Hanchen Wang, and Aviv Regev. Fine-Tuning Discrete Diffusion Models via Reward Optimization with Applications to DNA and Protein Design, March 2025. URL http://arxiv.org/abs/2410.13643. arXiv:2410.13643 [cs].

Luhuan Wu, Brian L. Trippe, Christian A. Naesseth, David M. Blei, and John P. Cunningham. Practical and Asymptotically Exact Conditional Sampling in Diffusion Models, November 2024. URL http://arxiv.org/abs/2306.17775. arXiv:2306.17775 [stat].

Jinshou Yang, Feihan Zhou, Xiyuan Luo, Yuan Fang, Xing Wang, Xiaohong Liu, Ruiling Xiao, Decheng Jiang, Yuemeng Tang, Gang Yang, Lei You, and Yupei Zhao. Enhancer reprogramming: critical roles in cancer and promising therapeutic strategies. *Cell Death Discovery*, 11(1):84, March 2025. ISSN 2058-7716. doi: 10.1038/s41420-025-02366-3. URL https://www.nature.com/articles/s41420-025-02366-3. Publisher: Nature Publishing Group.

He Ye, Rojas Kevin, and Tao Molei. What exactly does guidance do in masked discrete diffusion models, 2025. URL https://arxiv.org/abs/2506.10971.

Jaesik Yoon, Hyeonseo Cho, Yoshua Bengio, and Sungjin Ahn. Fast monte carlo tree diffusion: 100x speedup via parallel sparse planning, 2025. URL https://arxiv.org/abs/2506.09498.

Jaesik Yoon, Hyeonseo Cho, Doojin Baek, Yoshua Bengio, and Sungjin Ahn. Monte carlo tree diffusion for system 2 planning, 2026. URL https://arxiv.org/abs/2502.07202.

Kai Zhang, James D. Hocker, Michael Miller, Xiaomeng Hou, Joshua Chiou, Olivier B. Poirion, Yunjiang Qiu, Yang E. Li, Kyle J. Gaulton, Allen Wang, Sebastian Preissl, and Bing Ren. A single-cell atlas of chromatin accessibility in the human genome. *Cell*, 184 (24):5985–6001.e19, November 2021. ISSN 0092-8674, 1097-4172. doi: 10.1016/j.cell.2021.10.024. URL https://www.cell.com/cell/abstract/S0092-8674(21)01279-4. Publisher: Elsevier.

Xiangcheng Zhang, Haowei Lin, Haotian Ye, James Zou, Jianzhu Ma, Yitao Liang, and Yilun Du. Inference-time scaling of diffusion models through classical search, 2025. URL https://arxiv.org/abs/2505.23614.

## Appendix contents

A **Objectives and generative model** ..... 16
A.1 Design objectives ..... 16
A.2 Class-conditioned sampling ..... 16

B **PACS: search and selection** ..... 17
B.1 Beam Search with MAP Lookahead ..... 17
B.2 Stochastic selection and implementation ..... 18
B.3 Algorithm and computational complexity ..... 19

C **Theoretical analysis** ..... 20
C.1 Finite-pool selection and consistency ..... 20
C.2 Score scaling and selection strength ..... 23
C.3 Global reward-tilted reference ..... 23

D **Experimental setup** ..... 24
D.1 Datasets and data partitioning ..... 24
D.2 Oracle training and validation ..... 24
D.3 Search settings and infrastructure ..... 25
D.4 Baseline implementations ..... 26
D.5 Metrics and natural-reference sets ..... 27

E **Diagnostics and additional experiments** ..... 29
E.1 MAP ranking reliability ..... 29
E.2 Selection schedules and ablations ..... 30
E.3 Candidate-budget and DFKC experiments ..... 33

F **Complete benchmark results** ..... 34
F.1 Branching-factor results ..... 34
F.2 Human-cell-line benchmark ..... 37
F.3 Immune-cell benchmark ..... 41

# A Objectives and Generative Model

## A.1 Design Objectives

**Reward during evaluation.** Following Section 3.1, let $\mathcal{C}^+$ and $\mathcal{C}^-$ denote nonempty, disjoint sets of target and off-target cell types. We distinguish the design oracle, whose predictions guide search, from the evaluation oracle, whose predictions are used to assess the generated sequences. For the evaluation oracle, the MinGap score is

$$g(\boldsymbol{x}) = \frac{1}{|\mathcal{C}^+|} \sum_{c \in \mathcal{C}^+} s_c(\boldsymbol{x}) - \max_{c \in \mathcal{C}^-} s_c(\boldsymbol{x}). \tag{5}$$

This score is used to measure and report predicted specificity. When multiple target cell types are specified, the target term averages their predictions rather than imposing a minimum activity for every target.

**Reward used during search.** For all our experiments, we use the log-sum-exp specificity reward during design and search

$$r(\boldsymbol{x}) = \frac{1}{|\mathcal{C}^+|} \sum_{c \in \mathcal{C}^+} s_c^{\text{design}}(\boldsymbol{x}) - \log\left( \sum_{c \in \mathcal{C}^-} \exp s_c^{\text{design}}(\boldsymbol{x}) \right). \tag{6}$$

Search and evaluation use different reward functions and separately trained oracles, so they need not rank sequences identically.

**Connection to partial-state valuation.** Both objectives evaluate complete DNA sequences. During search, a partially masked candidate is assigned the reward of its positionwise MAP completion, as defined in equation 2. This completion provides a scoring surrogate, while the partially masked candidate remains the state from which subsequent denoising proceeds. Sequence fidelity and diversity are assessed separately from this specificity reward.

## A.2 Class-Conditioned Sampling

**Classifier-free guidance.** Following Schiff et al. (2025), we factorize the guided distribution across token positions and apply per-token renormalization. The CFG-guided distribution is defined as a product of experts, which requires normalization for $\gamma > 1$. At each masked position $\ell$, we combine the conditional and unconditional denoiser distributions and normalize over nucleotides. The resulting categorical distribution determines the nucleotide identity when a masked position is revealed. The masking schedule separately determines whether that position is revealed.

Concretely, the guided sampling distribution is:

$$d_\theta^\gamma(\boldsymbol{z}_s^{1:L} | \boldsymbol{z}_t^{1:L}, \boldsymbol{y}) = \prod_{\ell=1}^{L} \frac{1}{Z^{(\ell)}} d_\theta(\boldsymbol{z}_s^\ell | \boldsymbol{y}, \boldsymbol{z}_t^{1:L})^\gamma d_\theta(\boldsymbol{z}_s^\ell | \boldsymbol{z}_t^{1:L})^{1-\gamma} \tag{7}$$

where $Z^{(\ell)} = \sum_{\boldsymbol{z}_s'} d_\theta(\boldsymbol{z}_s' \mid \boldsymbol{z}_t^{1:L}, \boldsymbol{y})^\gamma d_\theta(\boldsymbol{z}_s' \mid \boldsymbol{z}_t^{1:L})^{(1-\gamma)}$ is the per-token normalization constant.

**Reverse proposal.** For $s < t$, the guided proposal at position $\ell$ is

$$Q_{s|t}^\ell(a \mid \boldsymbol{z}_t, y) = \begin{cases} \mathbf{1}\{a = z_t^\ell\}, & z_t^\ell \neq \boldsymbol{m}, \\ \dfrac{1-\alpha_s}{1-\alpha_t}, & z_t^\ell = \boldsymbol{m},\ a = \boldsymbol{m}, \\ \dfrac{\alpha_s - \alpha_t}{1-\alpha_t} d_{\theta,\ell}^\gamma(a \mid \boldsymbol{z}_t, t, y), & z_t^\ell = \boldsymbol{m},\ a \in \mathcal{V}. \end{cases} \tag{8}$$

The sequence-level proposal factorizes as

$$Q_{s|t}(\boldsymbol{z}_s \mid \boldsymbol{z}_t, y) = \prod_{\ell=1}^{L} Q_{s|t}^\ell(z_s^\ell \mid \boldsymbol{z}_t, y). \tag{9}$$

Positions are sampled independently conditional on the current sequence, but each denoiser prediction uses the full sequence context. Guidance changes which nucleotide is proposed at a newly revealed position, while the masking schedule determines the probability of revealing that position. Revealed nucleotides remain fixed in subsequent transitions.

**Sampling grid and output.** Our PACS sampler initializes all positions as masked and uses the reverse-time grid with $S$ number of steps

$$t_k = 1 - k\frac{1-\varepsilon}{S}, \qquad k = 0, \ldots, S, \qquad \varepsilon = 10^{-5}. \tag{10}$$

Transition $k$ proposes children at $s_k = t_{k+1}$. The clean-fraction coordinate is evaluated from the configured noise schedule as $\alpha(s_k) = \exp[-\sigma(s_k)]$. It is not the measured fraction of revealed positions in an individual candidate. At the final step, any remaining masked positions in each selected child are filled using the denoiser's positionwise MAP prediction while preserving revealed nucleotides. As at earlier steps, this completion is compared with the lineage-specific archive. We report the highest-scoring MAP completion archived among the selected children along each lineage, rather than necessarily reporting the completion of the final retained state.

**Separation of proposal and selection.** The pretrained denoiser, regulatory class, and guidance scale define the reverse proposal. PACS acts after candidates have been sampled from this proposal, changing their probabilities of being retained. It therefore leaves the proposal parameters unchanged while altering the distribution of selected continuations.

# B PACS: Search and Selection

## B.1 Beam Search with MAP Lookahead: Implementation Details

**Search population and candidate expansion.** The search maintains $N$ parent states, each associated with a persistent and independent lineage initialized from the fully masked sequence. At transition $t_k \to s_k = t_{k+1}$, each parent independently generates $B$ candidate children from the diffusion reverse proposal. Thus, $N$ controls the number of output trajectories, whereas $B$ controls the number of alternatives considered for each parent at each step. The denoiser prediction for a parent is computed once and reused to sample its $B$ children.

**MAP lookahead evaluation.** Each candidate is evaluated using a fresh denoiser prediction conditioned on that candidate at its child time $s_k$. Its completion preserves all revealed nucleotides and fills each masked position with the highest-probability nucleotide:

$$\widehat{x}^{\ell}_{n,j} = \begin{cases} z^{(n,j),\ell}_{s_k}, & z^{(n,j),\ell}_{s_k} \neq \boldsymbol{m}, \\ \arg\max_{a \in \mathcal{V}} d^{\gamma}_{\theta,\ell}\Big(a \mid \boldsymbol{z}^{(n,j)}_{s_k}, s_k, y\Big), & z^{(n,j),\ell}_{s_k} = \boldsymbol{m}. \end{cases} \tag{11}$$

The design oracle assigns the score $v_{n,j} = r(\widehat{\boldsymbol{x}}_{n,j})$. This positionwise MAP completion requires a single denoiser prediction per candidate rather than a stochastic rollout through the remaining diffusion steps. It is neither a joint MAP sequence nor an estimate obtained by averaging terminal rewards.

**Parentwise retention.** Exactly one child is retained from each parent's candidate pool. Greedy selection retains the candidate with the largest MAP score, whereas PACS samples a candidate according to the sibling-standardized probabilities in equation 3. Candidates from different parents do not compete with each other. This parentwise beam-search formulation therefore differs from global beam pruning, which retains the highest-scoring candidates across all parents. The population contains $N$ persistent lineages throughout generation, although different lineages may produce identical sequences.

**Reported sequences and implementation.** During search, each lineage continues from its retained partially masked state. Separately, we maintain an archive containing the highest-scoring MAP completion of the states retained along that lineage. After each selection step, the archive is updated if the selected child's completion has a strictly higher reward than the archived completion. Ties preserve the earlier sequence. After all reverse steps, we report one archived sequence per lineage. Thus, the reported sequence may be a completion evaluated before the final step, rather than the completion of the terminal state. This reporting rule is applied identically to PACS and greedy selection and reuses previously computed completions and rewards without additional denoiser or oracle evaluations. Furthermore, this archive does not affect the search process and is only used for reporting the final sequence.

### B.2 Stochastic Selection and Numerical Implementation

**Parentwise standardization.** For a fixed parent, let $v_1, \dots, v_B$ be the MAP-completion rewards of its sampled children. PACS standardizes these scores within the sibling pool and samples one surviving index according to

$$\pi_j = \frac{\exp[\kappa_k(v_j - \bar{v}_B)/(\sigma_B + \delta)]}{\sum_{\ell=1}^{B} \exp[\kappa_k(v_\ell - \bar{v}_B)/(\sigma_B + \delta)]}, \tag{12}$$

with

$$\sigma_B^2 = \frac{1}{B}\sum_{j=1}^{B}(v_j - \bar{v}_B)^2, \qquad \delta = 10^{-6}. \tag{13}$$

Because the common centering term cancels in the softmax, selection depends on score differences through the effective inverse temperature

$$\frac{\kappa_k}{\sigma_B + \delta}. \tag{14}$$

Thus, the same value of $\kappa_k$ acts relative to the spread of scores within each sibling pool, rather than to their absolute numerical scale. Importantly, $\sigma_B$ measures only within-pool score variation and is not an estimate of uncertainty or of MAP-score reliability.

**Dependence on reward scale.** Adding the same constant to every score leaves the selection probabilities unchanged. Under a positive affine rescaling

$$v_j' = a v_j + b, \qquad a > 0, \tag{15}$$

the standardized scores become

$$\frac{v_j' - \bar{v}_B'}{\sigma_B' + \delta} = \frac{v_j - \bar{v}_B}{\sigma_B + \delta/a}. \tag{16}$$

Hence multiplicative scale invariance is exact when $\delta = 0$ and $\sigma_B > 0$, and is effectively recovered when $\delta$ is negligible relative to the score spread. With a fixed positive stabilizer, positive-scale invariance is approximate when $\delta$ is negligible relative to both $\sigma_B$ and $a\sigma_B$. It is exact when the stabilizer is also rescaled to $a\delta$, or when $\delta = 0$ and $\sigma_B > 0$. For the implemented choice $\delta > 0$, an all-equal pool has $u_i = 0$ and yields uniform selection for every finite $\kappa$.

**Degenerate pools and tie handling.** If all candidates have the same score, then all standardized scores are zero and PACS reduces to uniform sampling over candidate indices for any finite $\kappa_k$. More generally, equal-score indices receive equal probability.

If the same sequence appears at multiple indices, its total probability is the sum of the probabilities assigned to those indices.

Sampling is implemented with the Gumbel–max construction applied to the standardized logits, using a separate random-number stream for selection. The greedy baseline instead keeps the first maximizing index. It therefore agrees with the $\kappa \to \infty$ limit when the maximizer is unique, whereas under ties the finite-temperature limit distributes probability uniformly over the maximizing indices.

### B.3 Complete Algorithm and Computational Complexity

**Algorithm 1** Progress-Annealed Candidate Selection (PACS)

**Require:** Sequence length $L$, regulatory class $y$, population size $N$, sibling pool size $B$, reverse grid $\{t_k\}_{k=0}^{S}$, proposal $Q$ incorporating the frozen denoiser and CFG scale $\gamma$, MAP completion operator $\widehat{\boldsymbol{x}}$, oracle reward $r$, masking schedule $\alpha$, parameters $\delta > 0$, $p$, and $\kappa_{\max}$.

Initialize $r_{\text{out}}^{(n)} \leftarrow -\infty$ and $\boldsymbol{x}_{\text{out}}^{(n)} \leftarrow \varnothing$ for all $n$
Initialize $\boldsymbol{z}_{t_0}^{(n)} \leftarrow \boldsymbol{m}^L$ for all $n \in \{1, \dots, N\}$
**for** $k = 0, \dots, S-1$ **do**
  Set child time $s_k \leftarrow t_{k+1}$
  $a_k \leftarrow \text{clip}_{[0,1]}\left(\dfrac{\alpha(s_k) - \alpha(t_1)}{\alpha(t_S) - \alpha(t_1)}\right)$
  $\kappa_k \leftarrow \kappa_{\max}(3a_k^2 - 2a_k^3)^p$
  **for** $n = 1, \dots, N$ **do**
    Sample $\boldsymbol{z}_{s_k}^{(n,1)}, \dots, \boldsymbol{z}_{s_k}^{(n,B)} \overset{\text{iid}}{\sim} Q_{s_k|t_k}(\cdot \mid \boldsymbol{z}_{t_k}^{(n)}, y)$
    For each $j$, compute $\widehat{\boldsymbol{x}}_{n,j} \leftarrow \widehat{\boldsymbol{x}}(\boldsymbol{z}_{s_k}^{(n,j)}; s_k, y)$, preserving revealed bases
    Compute $v_{n,j} \leftarrow r(\widehat{\boldsymbol{x}}_{n,j})$ for all $j \in \{1, \dots, B\}$
    Compute the mean and population standard deviation within this parent's pool:

$$\bar{v}_n \leftarrow \frac{1}{B}\sum_{j=1}^{B} v_{n,j}, \qquad \sigma_n \leftarrow \sqrt{\frac{1}{B}\sum_{j=1}^{B}(v_{n,j} - \bar{v}_n)^2}$$

    For each $j$, set $u_{n,j} \leftarrow \dfrac{v_{n,j} - \bar{v}_n}{\sigma_n + \delta}, \qquad \pi_{n,j} \leftarrow \dfrac{\exp(\kappa_k u_{n,j})}{\sum_{\ell=1}^{B}\exp(\kappa_k u_{n,\ell})}$
    Sample $J_n \sim \text{Categorical}(\pi_{n,1}, \dots, \pi_{n,B})$
    Update partial state: $\boldsymbol{z}_{s_k}^{(n)} \leftarrow \boldsymbol{z}_{s_k}^{(n,J_n)}$
    **if** $v_{n,J_n} > r_{\text{out}}^{(n)}$ **then**
      $\boldsymbol{x}_{\text{out}}^{(n)} \leftarrow \widehat{\boldsymbol{x}}_{n,J_n}$
      $r_{\text{out}}^{(n)} \leftarrow v_{n,J_n}$
    **end if**
  **end for**
**end for**
**return** $\{(\boldsymbol{x}_{\text{out}}^{(n)}, r_{\text{out}}^{(n)})\}_{n=1}^{N}$

*Greedy comparison.* Greedy selection replaces standardization and stochastic selection with $J_n \leftarrow \arg\max_j v_{n,j}$, resolving ties by the first maximizing index. All other steps remain unchanged.

**Computational Complexity.** The computational cost of PACS is governed primarily by neural forward passes:

- **Proposal generation:** Requires $\mathcal{O}(NS)$ denoiser forward passes. Because sibling proposals share their parent's sequence context, denoiser representations are evaluated once per parent lineage at each step.

- **Candidate evaluation:** Generating MAP completions and querying the activity oracle scales as $\mathcal{O}(NBS)$, which dominates neural evaluation cost.
- **Selection overhead:** Sibling standardization and categorical sampling require $\mathcal{O}(NBS)$ elementary scalar operations, contributing negligible runtime relative to model inference.

# C Theoretical Analysis

## C.1 Finite-Pool Selection and Consistency

The finite-pool results concern randomness in the selected index after candidate states have been proposed. This is distinct from the randomness that generates the candidate pool itself.

**Optimality of the finite-pool selector.** Let $\mu \in \Delta_B$ be a reference distribution over the $B$ candidate indices. In PACS, this reference is uniform, and

$$Z_\kappa = \sum_i \mu_i e^{\kappa u_i}, \qquad \pi_i = \frac{\mu_i e^{\kappa u_i}}{Z_\kappa}.$$

Since $\mu_i > 0$ and the scores are finite, $\pi_i > 0$. For any $p \in \Delta_B$,

$$\mathrm{KL}(p||\pi) = \mathrm{KL}(p||\mu) - \kappa \sum_i p_i u_i + \log Z_\kappa, \tag{17}$$

and therefore

$$\kappa \sum_i p_i u_i - \mathrm{KL}(p||\mu) = \log Z_\kappa - \mathrm{KL}(p||\pi). \tag{18}$$

Since $\mathrm{KL}(p||\pi) \geq 0$, the objective is uniquely maximized at $p = \pi$. For $\kappa = 0$, this gives $\pi = \mu$, recovering the reference distribution. If $\mu$ has zero entries, the same argument applies on its support.

**Limiting cases.** As $\kappa \to 0$, continuity of the exponential gives $e^{\kappa u_i} \to 1$ for every $i$. Hence $Z_\kappa \to 1$ and $\pi_i \to \mu_i$.

For $\kappa \to \infty$, let $u_{\max} = \max_j u_j$ and $\mathcal{M} = i : u_i = u_{\max}$. Writing

$$\pi_i = \frac{\mu_i e^{\kappa(u_i - u_{\max})}}{\sum_j \mu_j e^{\kappa(u_j - u_{\max})}},$$

shows that every nonmaximal term vanishes exponentially, while the maximal terms remain unchanged. Therefore $\pi$ converges componentwise (equivalently, in total variation on the finite candidate set) to

$$\pi_i^{(\infty)} = \begin{cases} \dfrac{\mu_i}{\sum_{j \in \mathcal{M}} \mu_j}, & i \in \mathcal{M}, \\ 0, & i \notin \mathcal{M}. \end{cases}$$

In particular, if the maximizer is unique, $\pi$ converges to a point mass on the highest-scoring candidate, recovering greedy selection.

Under uniform selection, the surviving index is drawn as $I \sim \mathrm{Unif}1, \ldots, B$, independently of the candidate scores. Conditional on the current parent $z_t$, each candidate $Z_i$ is itself sampled i.i.d. from the reverse proposal $Q_{s|t}(\cdot \mid z_t, y)$. Therefore, selecting one of these candidates uniformly does not change its marginal distribution: for any measurable set $A$,

$$\Pr(Z_I \in A \mid z_t) = \frac{1}{B} \sum_{i=1}^{B} \Pr(Z_i \in A \mid z_t) = Q_{s|t}(A \mid z_t, y). \tag{19}$$

Thus, when $\kappa = 0$, PACS simply samples the next state from the original reverse-diffusion transition. Repeating this uniform-selection rule at every reverse step therefore reproduces ancestral sampling, provided no additional reranking or filtering is applied.

**Large-pool consistency 1.** Fix a parent state and one reverse-diffusion transition. Let $Q$ be the reverse proposal on a finite support $\mathcal{S}$, let $v : \mathcal{S} \to \mathbb{R}$ be finite, and draw

$$Z_1, \dots, Z_B \overset{\text{iid}}{\sim} Q.$$

Write

$$\bar{v}_B = \frac{1}{B}\sum_{i=1}^{B} v(Z_i), \qquad \sigma_B^2 = \frac{1}{B}\sum_{i=1}^{B} v(Z_i)^2 - \bar{v}_B^2,$$

and let PACS select index $I$ with probabilities

$$\pi_i^{(B)} \propto \exp\left\{\kappa \frac{v(Z_i) - \bar{v}_B}{\sigma_B + \delta}\right\}, \qquad \delta > 0,\ \kappa < \infty.$$

The conditional law of the selected child is

$$\widehat{Q}_{B,\kappa}(z) := \Pr(Z_I = z \mid Z_1, \dots, Z_B).$$

At the population level, let

$$m = \mathbb{E}_Q[v(Z)], \qquad \sigma^2 = \mathrm{Var}_Q(v(Z)),$$

and define

$$Q_\kappa(z) = \frac{Q(z)\exp\left\{\kappa \frac{v(z)-m}{\sigma+\delta}\right\}}{\sum_{z' \in \mathcal{S}} Q(z')\exp\left\{\kappa \frac{v(z')-m}{\sigma+\delta}\right\}}. \tag{20}$$

Finally, $\overline{Q}_{B,\kappa} := \mathbb{E}[\widehat{Q}_{B,\kappa}]$ denotes the unconditional law of the selected child.

**Proposition 1** (Large-pool consistency)**.** *As $B \to \infty$,*

$$\|\widehat{Q}_{B,\kappa} - Q_\kappa\|_{\mathrm{TV}} \to 0 \quad \textit{almost surely,}$$

*and consequently*

$$\|\overline{Q}_{B,\kappa} - Q_\kappa\|_{\mathrm{TV}} \to 0.$$

*Moreover, for fixed finite $\kappa$,*

$$\|\overline{Q}_{B,\kappa} - Q_\kappa\|_{\mathrm{TV}} = O(B^{-1/2}).$$

*If $\sigma = 0$, then $Q_\kappa = Q$ and $\overline{Q}_{B,\kappa} = Q$ for every $B$.*

*Proof.* Define the empirical candidate distribution

$$\widehat{q}_B(z) = \frac{1}{B}\sum_{i=1}^{B} \mathbf{1}\{Z_i = z\}.$$

For any probability vector $q$ on $\mathcal{S}$, set

$$m(q) = \sum_z q(z)v(z), \qquad \sigma(q)^2 = \sum_z q(z)v(z)^2 - m(q)^2,$$

and define the standardized tilt

$$T_\kappa(q)(z) = \frac{q(z)\exp\left\{\kappa \frac{v(z)-m(q)}{\sigma(q)+\delta}\right\}}{\sum_{z'} q(z')\exp\left\{\kappa \frac{v(z')-m(q)}{\sigma(q)+\delta}\right\}}.$$

Grouping equal candidate states shows exactly that

$$\widehat{Q}_{B,\kappa} = T_\kappa(\widehat{q}_B), \qquad Q_\kappa = T_\kappa(Q).$$

For each fixed $z$, the indicators $\mathbf{1}\{Z_i = z\}$ are iid Bernoulli with mean $Q(z)$. Hence, by the strong law of large numbers,

$$\widehat{q}_B(z) \to Q(z) \qquad \text{a.s.}$$

Because $\mathcal{S}$ is finite, this holds simultaneously for all $z$, so $\|\widehat{q}_B - Q\|_1 \to 0$ a.s. The map $T_\kappa$ is continuous. $m(q)$ and $\sigma(q)$ are continuous, $\sigma(q) + \delta \geq \delta > 0$, and the normalizing denominator is positive. Therefore the continuous mapping theorem gives

$$\widehat{Q}_{B,\kappa} = T_\kappa(\widehat{q}_B) \to T_\kappa(Q) = Q_\kappa \qquad \text{a.s.}$$

Since $\mathcal{S}$ is finite, componentwise convergence is equivalent to total-variation convergence, proving the first claim.

For the unconditional law, thanks to Jensen's inequality,

$$\|\overline{Q}_{B,\kappa} - Q_\kappa\|_{\mathrm{TV}} \leq \mathbb{E}\|\widehat{Q}_{B,\kappa} - Q_\kappa\|_{\mathrm{TV}}.$$

The quantity inside the expectation converges a.s. to zero and is bounded by 1, so bounded convergence proves the second claim.

For the rate, assume first $\sigma > 0$. Then $T_\kappa$ is locally Lipschitz: for some $C < \infty$,

$$\|T_\kappa(q) - T_\kappa(Q)\|_{\mathrm{TV}} \leq C\|q - Q\|_1$$

whenever $q$ is sufficiently close to $Q$. For every $z$,

$$B\widehat{q}_B(z) \sim \mathrm{Binomial}(B, Q(z)),$$

so, by Cauchy-Schwarz inequality,

$$\mathbb{E}|\widehat{q}_B(z) - Q(z)| \leq \sqrt{\mathrm{Var}(\widehat{q}_B(z))} = \sqrt{\frac{Q(z)(1 - Q(z))}{B}}.$$

Summing over the finite support yields

$$\mathbb{E}\|\widehat{q}_B - Q\|_1 = O(B^{-1/2}).$$

To control the rare event where the local Lipschitz bound may not apply, choose $\varepsilon > 0$ such that

$$U_\varepsilon := \{q : \|q - Q\|_2 < \varepsilon\}$$

is contained in the Lipschitz neighborhood of $Q$. Since

$$\mathbb{E}\|\widehat{q}_B - Q\|_2^2 = \sum_{z \in \mathcal{S}} \mathrm{Var}(\widehat{q}_B(z)) = \frac{1}{B} \sum_{z \in \mathcal{S}} Q(z)(1 - Q(z)) = O(B^{-1}),$$

Markov's inequality gives

$$\Pr(\widehat{q}_B \notin U_\varepsilon) \leq \Pr(\|\widehat{q}_B - Q\|_2 \geq \varepsilon) \leq \frac{\mathbb{E}\|\widehat{q}_B - Q\|_2^2}{\varepsilon^2} = O(B^{-1}).$$

We then split the expected error according to whether $\widehat{q}_B$ lies inside this neighborhood:

$$\begin{aligned}\mathbb{E}\|T_\kappa(\widehat{q}_B) - T_\kappa(Q)\|_{\mathrm{TV}} = \mathbb{E}\Big[\|T_\kappa(\widehat{q}_B) - T_\kappa(Q)\|_{\mathrm{TV}} \mathbf{1}_{\{\widehat{q}_B \in U_\varepsilon\}}\Big] \\ + \mathbb{E}\Big[\|T_\kappa(\widehat{q}_B) - T_\kappa(Q)\|_{\mathrm{TV}} \mathbf{1}_{\{\widehat{q}_B \notin U_\varepsilon\}}\Big].\end{aligned}$$

On $U_\varepsilon$, the Lipschitz bound and $\mathbb{E}\|\widehat{q}_B - Q\|_1 = O(B^{-1/2})$ give an $O(B^{-1/2})$ contribution. Outside $U_\varepsilon$, total variation is at most 1, so the contribution is at most $\Pr(\widehat{q}_B \notin U_\varepsilon) = O(B^{-1})$. Therefore

$$\mathbb{E}\|\widehat{Q}_{B,\kappa} - Q_\kappa\|_{\mathrm{TV}} = O(B^{-1/2}),$$

and consequently

$$\|\overline{Q}_{B,\kappa} - Q_\kappa\|_{\mathrm{TV}} = O(B^{-1/2}).$$

Finally, if $\sigma = 0$, then $v(z) = m$ on the support of $Q$, so all exponential weights are equal. Thus $Q_\kappa = Q$, while conditional selection reduces to the empirical law $\widehat{q}_B$; taking expectation gives $\overline{Q}_{B,\kappa} = Q$ exactly for every $B$. □

### C.2 Score Scaling and Selection Strength

**Score invariance and degenerate pools.** Adding the same constant to all raw scores leaves standardized scores and selection probabilities unchanged. If $v_i' = av_i + b$ with $a > 0$, then

$$u_i' = \frac{a(v_i - \bar{v})}{a\sigma_B + \delta} = \frac{a(\sigma_B + \delta)}{a\sigma_B + \delta}\, u_i. \tag{21}$$

Thus positive-scale invariance holds when $\delta$ is negligible relative to both $\sigma_B$ and $a\sigma_B$. Cases of null $\sigma_B$ are also covered : an all-equal pool has $u_i = 0$ and yields uniform selection for every finite $\kappa$, and a single-candidate pool always returns its sole proposal.

**Selection strength on a fixed pool.** For a fixed candidate pool with standardized scores $u = (u_1, \ldots, u_B)$, define

$$\pi_{\kappa,i} = \frac{\mu_i e^{\kappa u_i}}{\sum_j \mu_j e^{\kappa u_j}},$$

where $\mu$ is a fixed reference distribution over candidate indices; in PACS, $\mu_i = 1/B$. Thus $\pi_0 = \mu$, while increasing $\kappa$ increasingly favors high-scoring candidates.

Let

$$F(\kappa) = \log \sum_i \mu_i e^{\kappa u_i}.$$

Then

$$F'(\kappa) = \mathbb{E}_{\pi_\kappa}[u], \qquad F''(\kappa) = \mathrm{Var}_{\pi_\kappa}(u),$$

so

$$\frac{d}{d\kappa}\mathbb{E}_{\pi_\kappa}[u] = \mathrm{Var}_{\pi_\kappa}(u) \geq 0. \tag{22}$$

Moreover,

$$\frac{d}{d\kappa}\,\mathrm{KL}(\pi_\kappa || \mu) = \kappa\,\mathrm{Var}_{\pi_\kappa}(u) \geq 0. \tag{23}$$

Thus, on a fixed pool, larger $\kappa$ increases the expected selected score and moves selection farther from its reference distribution. For uniform $\mu$, $H(\pi_\kappa) = \log B - \mathrm{KL}(\pi_\kappa|\mu)$, so selection entropy is nonincreasing in $\kappa$. This gives $\kappa$ a precise interpretation as selection strength. These guarantees are local to a fixed pool, across diffusion steps, both candidates and scores change.

### C.3 Global Reward-Tilted Reference

**Global reward-tilted reference.** Here $P_Q$ denotes the distribution obtained from the fixed reverse proposal followed by MAP completion of the final partial state, with a fixed class condition and finite $\eta > 0$. For comparison, the classical KL-regularized target associated with a reference terminal distribution $P_Q$ and reward $r$ is

$$P^\star(x) \propto P_Q(x) e^{\eta r(x)}.$$

The corresponding exact reward-guided Markov process is given by the Doob $h$-transform, with

$$h_t(z_t) = \mathbb{E}_Q\left[e^{\eta r(X_0)} \mid Z_t = z_t\right], \qquad Q^\star_{s|t}(z_s \mid z_t) = Q_{s|t}(z_s \mid z_t)\frac{h_s(z_s)}{h_t(z_t)}$$

(Uehara et al., 2025). The exact globally reward-tilted sampler reweights each reverse transition according to a soft continuation value. PACS follows the same general principle

Table 1: Immune-cell targets. Prevalence is the fraction of the original peaks annotated as accessible in each cell type.

| Benchmark name | Source atlas name | Short code | Prevalence (%) |
|---|---|---|---|
| $CD8^+$ T cells | T Lymphocyte 1 (CD8+) | `CD8T` | 5.24 |
| $CD4^+$ T cells | T lymphocyte 2 (CD4+) | `CD4T` | 4.43 |
| Natural killer T | Natural Killer T | `NKT` | 4.04 |
| Naive T cells | Naive T | `NaiveT` | 4.67 |
| Memory B cells | Memory B | `MemoryB` | 4.02 |
| Plasma B cells | Plasma B | `PlasmaB` | 5.25 |
| Macrophages | Macrophage General | `Macrophage` | 9.23 |
| Mast cells | Mast | `Mast` | 5.22 |

of locally favoring high-value continuations, but replaces this intractable quantity with the MAP-completion reward, standardized within each sibling pool and modulated by the step-dependent strength $\kappa_k$. Its large-pool limit therefore has the same exponential-reweighting structure as the ideal reward-guided transition, while remaining computable with a single lookahead score per candidate.

# D Experimental Setup

## D.1 Datasets and Data Partitioning

### D.1.1 Human cell-line activity benchmark

The human cell-line benchmark uses 735,156 regulatory DNA sequences from the MPRA dataset of Gosai et al. (2024). Each sequence is 200 bp long and has measured activity in HepG2, K562, and SK-N-SH. We use the distributed sequences and $\log_2$ fold-change measurements without additional sequence resizing or label transformation.

To reduce sequence similarity between the design and evaluation partitions, we cluster sequences using MMseqs2 easy linclust with minimum sequence identity 0.8, coverage threshold 0.8, and coverage mode 1 (Steinegger & Söding, 2017). Clusters are shuffled and assigned intact to the smaller of two partitions. Each partition contains 367,578 sequences.

### D.1.2 Immune-cell accessibility benchmark

The immune benchmark uses the chromatin accessibility atlas of Zhang et al. (2021), which contains binary accessibility annotations for 1,121,319 peaks across 204 cell types. We extract the central 200 bp of each 400 bp peak from hg38 and retain windows consisting entirely of A, C, G, and T. Removing 19 invalid windows leaves 1,121,300 sequences. A sequence can be accessible in more than one cell type. The same clustering and partitioning procedure yields two sets of 560,650 sequences.

The predictive models have 204 output tracks. Design and specificity evaluation restrict these outputs to the eight immune cell types in Table 1; each task has one target and seven off-targets. Thus, specificity in this benchmark is defined relative to these seven alternatives, not the remaining 203 cell types in the atlas.

## D.2 Predictive-Oracle Training and Validation

### D.2.1 Architectures and training settings

Enformer and Borzoi were adapted using gReLU Lal et al. (2025). We replaced the output head and fine-tuned the full network. The reported settings used Adam with learning rate $10^{-4}$, batch size 1024, and no reverse-complement or shift augmentation. MPRA models have three outputs and are trained for 15 epochs with mean-squared-error loss. Accessibility models have 204 outputs and are trained for 15 epochs with binary cross-entropy loss.

The AlphaGenome-based evaluators used a PyTorch implementation Bredikhin et al. (2026) initialized from fold-0 pretrained weights, with a layer-normalized two-layer MLP replacing the original heads. Training used AdamW, encoder and head learning rates of $10^{-4}$ and $10^{-3}$, respectively, and 15 epochs with one epoch of linear warmup followed by cosine decay. These are task-specific fine-tuned evaluators, rather than the unmodified pretrained models.

Enformer and AlphaGenome receive the 200 bp sequence directly. For Borzoi, each sequence is padded symmetrically with 28 ambiguous bases on either side to give 256 bp, at both training and evaluation. These input adaptations differ from the models' original long-context use.

**Prediction scale.** MPRA predictions are used on the trained regression scale without an additional transformation. Accessibility models are trained with binary labels, and their output logits are transformed with a sigmoid before computing specificity. Thus, immune MinGap and the other specificity measures use predicted accessibility probabilities.

#### D.2.2 Reported performance

Tables 2 and 3 summarize the validation-performance records. MPRA performance is measured by Pearson and Spearman correlation. Accessibility performance is measured by AUROC, with class prevalence reported to contextualize the precision–recall results.

For every architecture and both benchmarks, the checkpoint with the best validation performance is used for final scoring. The validation data are the opposite partition from the training data.

Table 2: MPRA validation-performance records for design- and evaluation-partition oracles, evaluated on the opposite partition. The reported models use best-validation checkpoints.

| Oracle | Cell type | Pearson $r$ | Spearman $\rho$ |
|---|---|---|---|
| Enformer (design) | HepG2 | 0.865 | 0.779 |
| | K562 | 0.862 | 0.744 |
| | SK-N-SH | 0.862 | 0.789 |
| Enformer (eval) | HepG2 | 0.863 | 0.775 |
| | K562 | 0.857 | 0.739 |
| | SK-N-SH | 0.863 | 0.788 |
| AlphaGenome (design) | HepG2 | 0.889 | 0.811 |
| | K562 | 0.885 | 0.772 |
| | SK-N-SH | 0.884 | 0.812 |
| AlphaGenome (eval) | HepG2 | 0.887 | 0.805 |
| | K562 | 0.883 | 0.770 |
| | SK-N-SH | 0.883 | 0.809 |
| Borzoi (design) | HepG2 | 0.865 | 0.781 |
| | K562 | 0.865 | 0.750 |
| | SK-N-SH | 0.865 | 0.788 |
| Borzoi (eval) | HepG2 | 0.864 | 0.779 |
| | K562 | 0.857 | 0.746 |
| | SK-N-SH | 0.857 | 0.784 |

### D.3 Search Hyperparameters and Computational Infrastructure

**Shared benchmark configuration.** We use the same search hyperparameters for the human-cell-line and immune-cell benchmarks (Table 4). PACS and greedy selection share the frozen class-conditioned diffusion checkpoint, candidate-generation procedure, MAP valuation, reward definition, and lineage-best reporting rule. Task-specific oracle checkpoints and target/off-target cell-type sets are specified in the corresponding experimental setup. Here, $B$ denotes the number of candidate children per parent.

Table 3: Validation performance for chromatin accessibility prediction.

| Cell type | Prev. (%) | Enformer (design) AUROC | Enformer (eval) AUROC | AlphaGenome (eval) AUROC | Borzoi (eval) AUROC |
|---|---|---|---|---|---|
| $CD8^+$ T | 5.24 | 0.925 | 0.923 | 0.928 | 0.922 |
| $CD4^+$ T | 4.43 | 0.931 | 0.928 | 0.933 | 0.927 |
| NKT | 4.04 | 0.934 | 0.934 | 0.941 | 0.934 |
| Naive T | 4.67 | 0.925 | 0.922 | 0.926 | 0.919 |
| Memory B | 4.02 | 0.930 | 0.928 | 0.933 | 0.926 |
| Plasma B | 5.25 | 0.917 | 0.914 | 0.922 | 0.909 |
| Macrophage | 9.23 | 0.907 | 0.905 | 0.909 | 0.903 |
| Mast | 5.22 | 0.918 | 0.918 | 0.922 | 0.912 |
| **Mean (8 immune)** | 5.26 | **0.923** | **0.922** | **0.927** | **0.919** |
| Mean (all 204 tracks) | – | 0.900 | 0.898 | 0.903 | 0.894 |

Table 4: Shared search configuration for the human-cell-line and immune-cell benchmarks. Each run produces 128 sequences. Branching-factor and other ablations vary the corresponding settings explicitly.

| Parameter | Setting |
|---|---|
| Diffusion backbone | Frozen class-conditioned DiMamba checkpoint |
| Regulatory class $y$ | Enhancer |
| CFG scale $\gamma$ | 1.0 |
| Reverse steps $S$ | 32 |
| Population size $N$ | 128 |
| Output sequences per run | 128 |
| Branching factor $B$ | 2,048 |
| Candidate valuation | Positionwise MAP completion |
| Population retention | One child per parent |
| Output rule | Best retained-state completion |
| Greedy selection | Highest-scoring sibling |
| PACS selection | Sibling-standardized Boltzmann sampling |
| PACS progress coordinate | Child clean fraction, normalized over the full trajectory |
| PACS schedule | Powered cubic smoothstep |
| PACS exponent $p$ | 2 |
| PACS maximum inverse temperature $\kappa_{\max}$ | 40 |
| Standardization stabilizer $\delta$ | $10^{-6}$ |

**Computational infrastructure.** All experiments were conducted on NVIDIA H100 GPUs (80 GB VRAM).

### D.4 Implementation of Compared Design Methods

All benchmark runs return 128 sequences of length 200 bp. Shared-backbone search methods use the frozen regulatory diffusion model conditioned on the enhancer class with CFG scale $\gamma = 1$. The current greedy-selection control and PACS use the same sampler, reward, candidate count, and output rule, differing in candidate selection.

**PACS and greedy selection.** Both maintain $N = 128$ lineages and draw $B$ children per parent over $S = 32$ reverse steps. Each child is scored through its positionwise MAP completion. Greedy selection retains the highest-scoring child within each parent whereas PACS standardizes scores with the population standard deviation of that parent's pool and samples a survivor with a clean-fraction-based smoothstep schedule, $p = 2$ and $\kappa_{\max} = 40$. The main benchmark uses $B = 2048$, whereas the branching experiment varies $B$ from 2 to 1024. Selection is applied at every reverse step.

**Ancestral sampling and best-of-$N$.** Ancestral sampling uses the same frozen reverse proposal without reward-based selection. Best-of-$N$ generates complete ancestral trajectories and retains one sequence per independent group according to the design reward. For a search setting with $S$ steps and $B$ siblings, each of the 128 groups contains $SB$ complete

candidates, matching the $NSB$ design-oracle sequence evaluations of search. This matches oracle evaluations rather than total generation cost.

**Sequential Monte Carlo (SMC) and twisted diffusion sampler (TDS).** These baselines use the shared frozen backbone. SMC weights and resamples particles using reward information and TDS additionally guides the proposal with a reward-gradient twist. Our configuration uses 128 particles, 32 reverse steps, SMC temperature 0.5, and guidance strength 100 for TDS.

**Soft Value-based Decoding in Diffusion models (SVDD).** The benchmark adapts soft value-based decoding to the same backbone and specificity task, with $M = 10$ candidates and 32 reverse steps. A candidate is retained separately within each chain using greedy value selection, and the terminal sequence is returned. We use SVDD-PM, which evaluates the oracle on a posterior-mean nucleotide representation, whereas PACS evaluates a positionwise MAP completion.

**Discrete Feynman–Kac correctors (DFKC).** DFKC uses the shared backbone with reward scale $\beta = 100$, groups of four particles, and 32 groups, producing 128 outputs. We use 200 updates, low-confidence remasking, resampling, and a corrector cap of 10.

**Ledidi.** We use the gReLU Lal et al. (2025) implementation that optimizes the log-sum-exp specificity objective for 20,000 iterations from independently sampled random DNA sequences. Each optimization uses batch size one and returns its final iterate.

**DRAKES.** DRAKES is fine-tuned separately for each target. The learning rate was $10^{-4}$, 100 epochs, batch size 32, gradient accumulation over four steps, truncated backpropagation through 50 denoising steps, Gumbel temperature 1, gradient clipping at 1, and KL weight 0.001. Each trained model generates 128 sequences over 128 ancestral steps without post-hoc reward selection.

**D3.** The conditional D3 model uses an analytic sampler with 200 steps. Conditioning sets the target activity to 4 and off-target activities to 0 for MPRA, and to 1 and 0 for binary accessibility. All sampled sequences are retained. The final immune checkpoint is trained on the design partition.

**Ctrl-DNA.** The implementation performs 100 iterations of five epochs, batch size 128, and Lagrange learning rate $3 \times 10^{-4}$. Initialization uses 128 natural sequences with high target activity. The first 128 outputs of the final round are reported. In our configuration, the TFBS regularization term is disabled. The reported comparison therefore concerns this adaptation of Ctrl-DNA, not the complete published objective.

### D.5 Evaluation Metrics and Natural-Reference Sets

Metrics are computed separately for each run of 128 sequences. We report the mean and standard deviation across three runs. Natural-reference sets are fixed across methods for each target.

#### D.5.1 Predicted specificity

Let $s_c(\boldsymbol{x})$ denote a prediction from the specified oracle and $c_\star$ the target cell type. We report

$$\mathrm{MinGap}(\boldsymbol{x}) = s_{c_\star}(\boldsymbol{x}) - \max_{c \in \mathcal{C}^-} s_c(\boldsymbol{x}), \tag{24}$$

$$r_{\mathrm{LSE}}(\boldsymbol{x}) = s_{c_\star}(\boldsymbol{x}) - \log \sum_{c \in \mathcal{C}^-} \exp s_c(\boldsymbol{x}), \tag{25}$$

$$r_{\mathrm{mean}}(\boldsymbol{x}) = s_{c_\star}(\boldsymbol{x}) - \frac{1}{|\mathcal{C}^-|} \sum_{c \in \mathcal{C}^-} s_c(\boldsymbol{x}). \tag{26}$$

The log-sum-exp uses unit temperature. All inference-time methods optimize $r_{\mathrm{LSE}}$ with the design oracle.

#### D.5.2 Motif frequency and sequence composition

The motif reference uses 720 human motifs from JASPAR2024 CORE Rauluseviciute et al. (2024) and scans sequences with FIMO Bailey et al. (2015). Both strands are scanned with a uniform nucleotide background, motif pseudocount $10^{-4}$, score discretization width 0.1, and hit threshold $p < 10^{-3}$. For motif-frequency correlation, all retained hits are counted, including overlaps. The generated and reference count vectors are aligned to the same motif vocabulary and compared using Spearman correlation.

For 3-mer fidelity, all overlapping forward-strand 3-mers are counted without reverse-complement merging, giving 64 categories. We compare generated and reference frequency vectors by Pearson correlation. GC content is the mean fraction of G or C bases per sequence and is descriptive rather than a quantity to maximize.

#### D.5.3 NTv2 embedding fidelity

We compute Fréchet Biological Distance using the 500M-parameter multispecies Nucleotide Transformer v2 model Dalla-Torre et al. (2025). Each sequence is represented by the attention-masked mean of the final token representations, including the classification token under our protocol. Let $(\mu_g, \Sigma_g)$ and $(\mu_r, \Sigma_r)$ denote the empirical embedding means and covariances of the generated and reference sets. Then

$$\mathrm{FBD} = \|\mu_g - \mu_r\|_2^2 + \mathrm{tr}\Big(\Sigma_g + \Sigma_r - 2(\Sigma_g^{1/2}\Sigma_r\Sigma_g^{1/2})^{1/2}\Big). \tag{27}$$

We add a covariance diagonal regularizer of $10^{-6}$, compute the matrix square root through the covariance product, take its real component, and clip the resulting distance below at zero.

#### D.5.4 Motif arrangement

The arrangement metrics use the 50 most frequent motifs in the target-specific natural reference. Hits must satisfy $p < 10^{-4}$. Same-motif hits within 10 bp are greedily deduplicated by score, and at most ten highest-scoring hits are retained per sequence. All distances refer to motif-center positions. These metrics characterize the retained motif configurations under this filtering rule, rather than the complete regulatory grammar.

**Pairwise TFBS co-occurrence.** For each unordered pair of distinct motifs $(a, b)$, we estimate binary sequence-level occurrence probabilities and compute

$$q_{ab} = \log_2 \frac{P(a \wedge b) + \epsilon}{P(a)P(b) + \epsilon}, \qquad \epsilon = 10^{-9}. \tag{28}$$

The score is the Spearman correlation between generated and reference vectors over the same $\binom{50}{2}$ motif pairs.

**Inter-motif spacing.** Center-to-center distances between distinct motifs are pooled into twenty 10 bp bins spanning the sequence length. We report base-2 Jensen–Shannon divergence between the normalized reference and generated histograms. Lower divergence indicates closer agreement in the pooled spacing distribution.

**Relative motif orientation.** For each motif pair within 50 bp, we compute the fraction of co-occurrences on the same strand. The score is the Spearman correlation of these pair-specific fractions between generated and reference sequences. A motif pair must occur at least five times in both sets. At least five eligible pairs and nonconstant vectors are required. This statistic does not distinguish every possible order and strand arrangement.

**Diversity.** Positional Shannon entropy is computed from the empirical nucleotide frequencies across the 128 sequences at each position and then averaged across positions, with a maximum of 2 bits. Mean pairwise Levenshtein edit distance is averaged over all unordered sequence pairs. It counts the minimum number of single-base insertions, deletions, and substitutions required to transform one sequence into another.

**Natural-reference construction.** For each MPRA target, reference sequences are selected from the full measured corpus at or above the 99.9th percentile of target-versus-off-target MinGap, giving 736 sequences per target. Selection uses measured activities rather than oracle predictions.

For each immune target, references are atlas peaks accessible in that target and none of the other seven immune types. If fewer than 100 peaks satisfy this criterion, it is relaxed to allow accessibility in at most one additional immune type. The eligible set is subsampled with a fixed seed to a maximum of 2048 sequences.

# E Diagnostics and Additional Experiments

## E.1 MAP Ranking Reliability

**Task and parent trajectories.** We perform 200-bp HepG2-specific sequence design, with K562 and SK-N-SH as off-targets. Parent trajectories follow the greedy search procedure in Appendix B.1, using $N = 128$ lineages, 128 children per parent, 32 reverse steps, enhancer conditioning, and $\gamma = 1$. Retained partial states are saved after every step, and diagnostic probes are performed after trajectory generation. The analysis uses three independent seeds and concerns retained partial states, irrespective of which archived completion is subsequently reported.

**Checkpoint selection and fresh candidate pools.** We probe 30 checkpoints from each saved trajectory. With zero-based indexing, parent checkpoint $p \in \{0, \dots, 29\}$ uses the state saved after step $p$, proposes diagnostic children at step $p + 1$, and begins their subsequent continuation at step $p + 2$. Thus, the probe excludes both the initial decision from the fully masked state and the final decision with no remaining continuation step. Each parent generates $B = 32$ fresh diagnostic siblings, giving 4,096 candidates per checkpoint and seed. These pools are sampled independently of the candidates used to construct the saved trajectory. All candidates are evaluated, including duplicates, without MAP-based prefiltering.

**Denoising coordinate and MAP scores.** Checkpoints are indexed by the proposed child's schedule-defined clean fraction $\alpha_s = \exp[-\sigma(s)]$, spanning approximately 0.0989 to 0.9988 under the cosine schedule. This coordinate is not the measured revealed fraction of an individual candidate, and the final checkpoint precedes complete denoising. For parent $i$ and sibling $j$, the MAP score is $v_{s,ij} = r(\widehat{\boldsymbol{x}}_{s,ij})$, using the completion procedure in Appendix B.1 and the raw log-sum-exp reward in Appendix A.1. The same reward is applied to stochastic continuation endpoints.

**Independent continuation estimates.** Each candidate receives two independently seeded batches, $\mathcal{A}$ and $\mathcal{B}$, of $M = 64$ stochastic terminal continuations:

$$\widehat{V}^{\mathcal{A}}_{s,ij} = \frac{1}{M}\sum_{m=1}^{M} r(X^{\mathcal{A}}_{ijm}), \qquad \widehat{V}^{\mathcal{B}}_{s,ij} = \frac{1}{M}\sum_{m=1}^{M} r(X^{\mathcal{B}}_{ijm}).$$

Each continuation follows the remaining ancestral reverse steps with the same class condition and guidance scale, without further oracle-based selection. Only its completed endpoint is scored. The estimates therefore approximate expected terminal reward under ancestral continuation, rather than the value of continued PACS or greedy search. Batch $\mathcal{B}$ alone provides the MAP-ranking reference, while batch $\mathcal{A}$ provides an independent comparison for assessing reference reproducibility.

**Ranking agreement and aggregation.** At each checkpoint, we compute Spearman correlation separately within every parent's sibling pool:

$$\rho_{s,i}^{\mathrm{MAP}} = \rho_{\mathrm{S}}\left(\{v_{s,ij}\}_{j=1}^{B}, \{\widehat{V}_{s,ij}^{\mathcal{B}}\}_{j=1}^{B}\right).$$

We also assess the reproducibility of the Monte Carlo reference using

$$\rho_{s,i}^{\mathrm{rollout}} = \rho_{\mathrm{S}}\left(\{\widehat{V}_{s,ij}^{\mathcal{A}}\}_{j=1}^{B}, \{\widehat{V}_{s,ij}^{\mathcal{B}}\}_{j=1}^{B}\right).$$

We average correlations over valid parent pools within each seed and then equally across three seeds. Shading denotes the standard deviation of seed-level means. Curves connect measured checkpoints without fitting a functional relationship.

**Reference uncertainty and interpretation.** Across the measured checkpoints, mean MAP–rollout agreement rises overall from −0.004 to 0.824, while rollout–rollout agreement rises from 0.156 to 0.986 (Figure 6). Thus, the Monte Carlo reference is itself less reproducible early in denoising, but MAP agrees less strongly with it than an independent rollout estimate at every checkpoint. Rollout–rollout agreement is not a strict reliability ceiling or a correction factor. These results support progress-dependent ranking informativeness on the sampled greedy trajectories.

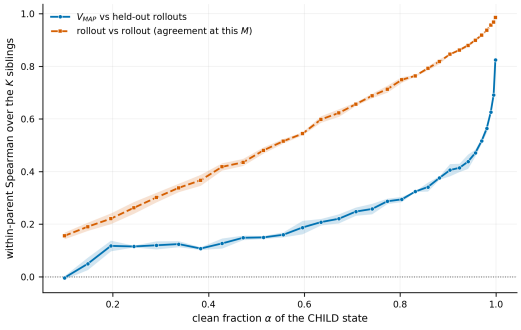


Figure 6: MAP ranking agreement across denoising.

### E.2 Clean-Fraction Scheduling and Selector Ablations

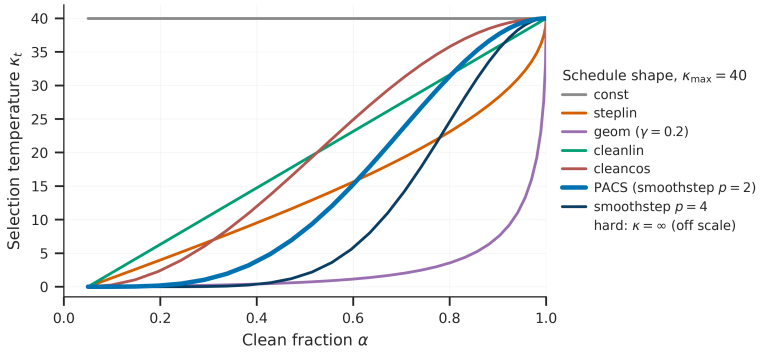


Figure 7: Selection-strength schedules

#### E.2.1 Progress coordinates and schedule families

**Progress coordinates.** At reverse step $k \in \{0, \ldots, S-1\}$, candidates are proposed at child time $s_k = t_{k+1}$. We compare step-indexed progress with the normalized schedule-defined clean fraction:

$$a_k^{\mathrm{step}} = \frac{k}{S-1}, \qquad a_k^{\mathrm{clean}} = \mathrm{clip}_{[0,1]}\left(\frac{\alpha(s_k) - \alpha(s_0)}{\alpha(s_{S-1}) - \alpha(s_0)}\right). \tag{29}$$

The clean coordinate uses the masking schedule, not the realized revealed fraction of an individual candidate. It is shared across parents at a given step. Both coordinates run from zero at the first candidate-selection step to one at the last.

**Selection schedules.** Finite-temperature selectors take the form $\kappa_k = \kappa_{\max} h_k$. The tested families are

$$h_k = \begin{cases} 1, & \text{constant}, \\ a_k^{\text{step}}, & \text{step-linear}, \\ a_k^{\text{clean}}, & \text{clean-linear}, \\ \left[1 - \cos(\pi a_k^{\text{clean}})\right]/2, & \text{clean-cosine}, \\ \left[3(a_k^{\text{clean}})^2 - 2(a_k^{\text{clean}})^3\right]^p, & \text{clean-smoothstep}, \\ \dfrac{(1+\eta)^k - 1}{(1+\eta)^{S-1} - 1}, & \text{geometric}, \quad \eta = 0.20. \end{cases} \tag{30}$$

The geometric growth parameter $\eta$ is different from the CFG scale $\gamma$. PACS uses clean-smoothstep with $p = 2$ and $\kappa_{\max} = 40$. Increasing $p$ reduces the inverse temperature at intermediate clean fractions while preserving its endpoints. This delays strong selection but does not guarantee a particular selection-entropy trajectory, since candidate scores also change during generation.

**Matching convention.** All nonconstant ramps reach their specified $\kappa_{\max}$ at the final step. We evaluate $\kappa_{\max} \in \{10, 20, 40\}$ for each family, including clean-smoothstep with $p = 1$ and $p = 2$, and additionally evaluate $p = 4$ at $\kappa_{\max} = 40$. Together with greedy selection, this gives 23 configurations per cell type. Matching endpoint strength does not match the trajectory-average inverse temperature $\bar{\kappa} = S^{-1} \sum_k \kappa_k$. Therefore, comparisons at fixed $\kappa_{\max}$ vary both the temporal profile and the average strength. Even matching $\bar{\kappa}$ would not match selection entropy or divergence from uniform choice, because these also depend on the realized sibling scores.

### E.2.2 Experimental protocol

We evaluate HepG2, K562, and SK-N-SH specificity using 200-bp sequences, $N = 128$ parent lineages, $B = 32$ children per parent, and $S = 32$ reverse steps. Every step performs MAP valuation and parentwise selection. The frozen diffusion backbone uses enhancer-class conditioning and $\gamma = 1$. All configurations share the design oracle, raw log-sum-exp reward, proposal, and completion procedure. Only the selector changes. The experiment uses five seeds, with 128 sequences per seed.

**Output rule and evaluation.** The comparisons characterize endpoint generation at $B = 32$. Tables 5–7 report AlphaGenome MinGap, JASPAR motif-frequency Spearman correlation, 3-mer Pearson correlation, and NTv2 embedding Fréchet distance (FBD). Entries are means with sample standard deviations across the five runs. Differences discussed below are descriptive comparisons of these run means, rather than claims of statistical significance or equivalence.

### E.2.3 Results and interpretation

**Scheduled versus constant selection.** The default PACS configuration has higher mean AlphaGenome MinGap, higher motif and 3-mer correlations, and lower NTv2 FBD than each of the three tested constant selectors in all three cell types. At the shared endpoint strength $\kappa_{\max} = 40$, the FBD reductions relative to constant selection are approximately 16.7%, 51.3%, and 7.1% for HepG2, K562, and SK-N-SH, respectively. Relative to greedy selection, PACS increases motif correlation by 0.145, 0.214, and 0.013 and 3-mer correlation by 0.118, 0.351, and 0.014, while reducing FBD by approximately 17.4%, 51.1%, and 5.2%. The corresponding changes in mean AlphaGenome MinGap are +0.082, +0.182, and −0.016. These results support the scheduled selector over the tested constant settings, with the largest fidelity gains on K562. They do not establish superiority over every possible constant strength.

**Effect of schedule shape and exponent.** At a fixed clean-fraction coordinate and $\kappa_{\max} = 40$, smoothstep with $p = 1$ reduces FBD relative to the linear ramp in all three cell types. Increasing the smoothstep exponent from $p = 1$ to $p = 2$ further reduces FBD

Table 5: Selector schedule sweep for HepG2. Values are mean (sample SD) across five runs, using final-state completions, $N = 128$, $B = 32$, and $S = 32$. Specificity is AlphaGenome MinGap; motif and 3-mer columns report correlations, and FBD uses NTv2 embeddings. The default PACS configuration is labeled; it is not asserted to be best in every column.

| Selector | $p$ | $\kappa_{\max}$ | MinGap ↑ | Motif ↑ | 3-mer ↑ | FBD ↓ |
|---|---|---|---|---|---|---|
| Greedy | – | $\infty$ | 3.558 (0.145) | 0.605 (0.022) | 0.803 (0.010) | 2.154 (0.060) |
| Constant | – | 10 | 3.466 (0.154) | 0.624 (0.045) | 0.817 (0.012) | 2.122 (0.055) |
| Constant | – | 20 | 3.588 (0.259) | 0.610 (0.019) | 0.810 (0.009) | 2.158 (0.025) |
| Constant | – | 40 | 3.593 (0.239) | 0.611 (0.031) | 0.813 (0.021) | 2.136 (0.082) |
| Step-linear | 1 | 10 | 3.333 (0.075) | 0.744 (0.034) | 0.899 (0.021) | 1.829 (0.045) |
| Step-linear | 1 | 20 | 3.504 (0.086) | 0.673 (0.012) | 0.866 (0.012) | 1.935 (0.057) |
| Step-linear | 1 | 40 | 3.635 (0.070) | 0.670 (0.037) | 0.851 (0.017) | 2.009 (0.055) |
| Clean-linear | 1 | 10 | 3.434 (0.164) | 0.721 (0.032) | 0.895 (0.011) | 1.868 (0.048) |
| Clean-linear | 1 | 20 | 3.571 (0.100) | 0.679 (0.039) | 0.854 (0.005) | 1.981 (0.042) |
| Clean-linear | 1 | 40 | 3.788 (0.138) | 0.669 (0.019) | 0.849 (0.014) | 2.030 (0.080) |
| Clean-cosine | 1 | 10 | 3.444 (0.138) | 0.746 (0.049) | 0.912 (0.023) | 1.804 (0.063) |
| Clean-cosine | 1 | 20 | 3.605 (0.063) | 0.723 (0.028) | 0.889 (0.008) | 1.863 (0.041) |
| Clean-cosine | 1 | 40 | 3.713 (0.180) | 0.685 (0.017) | 0.874 (0.012) | 1.945 (0.064) |
| Geometric | – | 10 | 1.162 (0.127) | 0.768 (0.020) | 0.951 (0.007) | 1.704 (0.022) |
| Geometric | – | 20 | 1.957 (0.130) | 0.770 (0.039) | 0.944 (0.013) | 1.709 (0.056) |
| Geometric | – | 40 | 2.660 (0.202) | 0.731 (0.054) | 0.937 (0.013) | 1.718 (0.026) |
| Clean-smoothstep | 1 | 10 | 3.463 (0.172) | 0.748 (0.040) | 0.913 (0.025) | 1.803 (0.069) |
| Clean-smoothstep | 1 | 20 | 3.522 (0.067) | 0.719 (0.020) | 0.886 (0.004) | 1.862 (0.031) |
| Clean-smoothstep | 1 | 40 | 3.708 (0.181) | 0.681 (0.024) | 0.873 (0.015) | 1.932 (0.056) |
| Clean-smoothstep | 2 | 10 | 3.186 (0.135) | 0.749 (0.029) | 0.932 (0.008) | 1.787 (0.102) |
| Clean-smoothstep | 2 | 20 | 3.486 (0.093) | 0.758 (0.049) | 0.926 (0.011) | 1.751 (0.029) |
| **PACS (default)** | 2 | 40 | 3.640 (0.133) | 0.750 (0.048) | 0.921 (0.012) | 1.779 (0.046) |
| Clean-smoothstep | 4 | 40 | 3.278 (0.188) | 0.747 (0.037) | 0.928 (0.008) | 1.755 (0.022) |

from 1.932 to 1.779 in HepG2, from 2.296 to 1.804 in K562, and from 2.071 to 1.982 in SK-N-SH. Mean AlphaGenome MinGap changes from 3.708 to 3.640, from 3.872 to 4.003, and from 3.179 to 3.142, respectively. Increasing $p$ to 4 further lowers FBD but also lowers mean specificity relative to $p = 2$ in all three targets. These comparisons show that the temporal profile offers different specificity–fidelity trade-offs rather than a universally optimal exponent.

**Strength dependence and the geometric control.** Increasing $\kappa_{\max}$ does not improve every outcome monotonically. Within PACS with $p = 2$, K562 attains higher mean MinGap and lower FBD at $\kappa_{\max} = 20$ than at 40 (4.049 versus 4.003 and 1.724 versus 1.804), whereas increasing the strength from 20 to 40 raises mean specificity in HepG2 and SK-N-SH with higher FBD. The geometric ramp concentrates selection late in the trajectory and produces low FBD but substantially reduced specificity at weak endpoint strengths. At $\kappa_{\max} = 10$, its AlphaGenome MinGap is 1.162, 1.718, and 0.972 across the three targets. Thus, improved fidelity alone is insufficient to choose a schedule.

**Scope of the conclusions.** The sweep supports varying reliance on MAP scores during generation and shows that constant finite-temperature selection over the tested range does not reproduce the default PACS trade-off. It does not distinguish schedule timing from average inverse temperature, establish the empirical necessity of sibling standardization, or validate the same schedule ranking under lineage-best reporting. The clean-fraction coordinate is motivated by denoising progress. Its optimality is not implied by these experiments.

Table 6: Selector schedule sweep for K562. Values are mean (sample SD) across five runs, using final-state completions, $N = 128$, $B = 32$, and $S = 32$. Specificity is AlphaGenome MinGap; motif and 3-mer columns report correlations, and FBD uses NTv2 embeddings. The default PACS configuration is labeled; it is not asserted to be best in every column.

| Selector | $p$ | $\kappa_{\max}$ | MinGap ↑ | Motif ↑ | 3-mer ↑ | FBD ↓ |
|---|---|---|---|---|---|---|
| Greedy | – | $\infty$ | 3.821 (0.155) | 0.712 (0.014) | 0.619 (0.016) | 3.691 (0.310) |
| Constant | – | 10 | 3.603 (0.140) | 0.701 (0.020) | 0.605 (0.021) | 3.778 (0.199) |
| Constant | – | 20 | 3.686 (0.179) | 0.705 (0.022) | 0.607 (0.023) | 3.873 (0.131) |
| Constant | – | 40 | 3.702 (0.080) | 0.700 (0.014) | 0.613 (0.027) | 3.708 (0.236) |
| Step-linear | 1 | 10 | 3.545 (0.173) | 0.916 (0.011) | 0.954 (0.010) | 2.006 (0.092) |
| Step-linear | 1 | 20 | 3.716 (0.162) | 0.900 (0.014) | 0.928 (0.011) | 2.271 (0.093) |
| Step-linear | 1 | 40 | 3.846 (0.150) | 0.870 (0.012) | 0.890 (0.022) | 2.389 (0.148) |
| Clean-linear | 1 | 10 | 3.580 (0.117) | 0.909 (0.019) | 0.945 (0.016) | 2.125 (0.163) |
| Clean-linear | 1 | 20 | 3.696 (0.154) | 0.874 (0.020) | 0.886 (0.019) | 2.611 (0.223) |
| Clean-linear | 1 | 40 | 3.799 (0.109) | 0.852 (0.004) | 0.855 (0.018) | 2.670 (0.291) |
| Clean-cosine | 1 | 10 | 3.825 (0.188) | 0.916 (0.004) | 0.965 (0.003) | 1.834 (0.104) |
| Clean-cosine | 1 | 20 | 3.819 (0.254) | 0.913 (0.015) | 0.950 (0.007) | 2.096 (0.034) |
| Clean-cosine | 1 | 40 | 3.872 (0.129) | 0.910 (0.008) | 0.937 (0.007) | 2.214 (0.121) |
| Geometric | – | 10 | 1.718 (0.161) | 0.903 (0.008) | 0.968 (0.010) | 1.649 (0.032) |
| Geometric | – | 20 | 2.567 (0.098) | 0.914 (0.017) | 0.970 (0.010) | 1.639 (0.032) |
| Geometric | – | 40 | 3.322 (0.164) | 0.910 (0.013) | 0.971 (0.005) | 1.662 (0.019) |
| Clean-smoothstep | 1 | 10 | 3.785 (0.165) | 0.916 (0.002) | 0.966 (0.004) | 1.871 (0.125) |
| Clean-smoothstep | 1 | 20 | 3.933 (0.178) | 0.910 (0.020) | 0.945 (0.011) | 2.112 (0.069) |
| Clean-smoothstep | 1 | 40 | 3.872 (0.043) | 0.903 (0.005) | 0.936 (0.008) | 2.296 (0.139) |
| Clean-smoothstep | 2 | 10 | 3.859 (0.216) | 0.919 (0.012) | 0.970 (0.003) | 1.675 (0.062) |
| Clean-smoothstep | 2 | 20 | 4.049 (0.130) | 0.919 (0.007) | 0.969 (0.007) | 1.724 (0.035) |
| **PACS (default)** | 2 | 40 | 4.003 (0.182) | 0.926 (0.009) | 0.970 (0.007) | 1.804 (0.067) |
| Clean-smoothstep | 4 | 40 | 3.953 (0.221) | 0.919 (0.011) | 0.969 (0.003) | 1.669 (0.028) |

## E.3 Candidate-Budget Experiments

### E.3.1 Candidate-budget and selection-rule comparisons

We report sweeps using $N = 128$, $S = 32$, and sibling counts $B \in \{8, 128, 1024, 2048\}$, comparing SMC-, TDS-, and SVDD branching. These runs distinguish resampling across a population from selection within a parent. We observe mode collapse for SMC and TDS as the branching factor is increased.

Table 8: Candidate-budget comparison across three human cell lines. Here $K$ denotes the sibling count $B$. MinGap is predicted specificity computed using the AlphaGenome evaluation model and is reported as mean (SD) across three runs. All other entries are means across the same runs. Motif and 3-mer denote motif-frequency and 3-mer-frequency correlations, respectively. FBD uses NTv2 embeddings. Entropy is mean positional Shannon entropy in bits; Div. is mean pairwise Levenshtein distance; Wall is elapsed time in seconds.

| | | | AlphaGenome | | | | | | Wall |
|---|---|---|---|---|---|---|---|---|---|
| Cell | Method | $K$ | MinGap (SD) | Motif | 3-mer | FBD | Entropy | Div. | (s) |
| HepG2 | SMC | 8 | 6.20 (0.35) | 0.37 | 0.57 | 5.47 | 0.16 | 12.8 | 9 |
| HepG2 | SMC | 128 | 7.06 (0.09) | 0.40 | 0.62 | 6.61 | 0.17 | 13.9 | 108 |
| HepG2 | SMC | 1024 | 6.98 (0.29) | 0.45 | 0.65 | 6.09 | 0.27 | 20.6 | 847 |
| HepG2 | SMC | 2048 | 6.81 (0.14) | 0.46 | 0.68 | 5.01 | 0.19 | 14.7 | 1707 |
| HepG2 | SVDD | 8 | 2.08 (0.12) | 0.61 | 0.83 | 2.09 | 1.98 | 149.8 | 10 |
| HepG2 | SVDD | 128 | 4.99 (0.16) | 0.74 | 0.90 | 1.95 | 1.98 | 149.8 | 108 |
| HepG2 | SVDD | 1024 | 6.08 (0.05) | 0.84 | 0.94 | 1.91 | 1.98 | 149.7 | 848 |
| HepG2 | SVDD | 2048 | 5.89 (0.16) | 0.84 | 0.94 | 1.97 | 1.98 | 149.8 | 1703 |

Continued on next page

Table 8 (continued)

| Cell | Method | $K$ | AlphaGenome MinGap (SD) | Motif | 3-mer | FBD | Entropy | Div. | Wall (s) |
|---|---|---|---|---|---|---|---|---|---|
| HepG2 | TDS | 8 | 6.56 (0.37) | 0.43 | 0.58 | 4.76 | 0.30 | 23.9 | 16 |
| HepG2 | TDS | 128 | 6.70 (0.14) | 0.41 | 0.51 | 5.83 | 0.24 | 18.5 | 116 |
| HepG2 | TDS | 1024 | 7.22 (0.10) | 0.39 | 0.53 | 5.23 | 0.12 | 9.4 | 858 |
| HepG2 | TDS | 2048 | 7.05 (0.68) | 0.48 | 0.60 | 4.50 | 0.32 | 24.3 | 1723 |
| K562 | SMC | 8 | 6.13 (0.50) | 0.42 | 0.48 | 8.38 | 0.18 | 13.5 | 9 |
| K562 | SMC | 128 | 6.95 (0.71) | 0.44 | 0.59 | 5.34 | 0.21 | 16.4 | 109 |
| K562 | SMC | 1024 | 6.38 (0.45) | 0.42 | 0.59 | 5.87 | 0.32 | 26.7 | 850 |
| K562 | SMC | 2048 | 6.32 (0.32) | 0.39 | 0.55 | 7.11 | 0.22 | 16.0 | 1698 |
| K562 | SVDD | 8 | 2.60 (0.25) | 0.69 | 0.56 | 4.24 | 1.95 | 147.5 | 9 |
| K562 | SVDD | 128 | 4.92 (0.21) | 0.79 | 0.78 | 2.53 | 1.97 | 149.2 | 109 |
| K562 | SVDD | 1024 | 6.05 (0.06) | 0.85 | 0.90 | 1.92 | 1.98 | 149.7 | 849 |
| K562 | SVDD | 2048 | 6.04 (0.06) | 0.85 | 0.91 | 1.91 | 1.98 | 149.8 | 1703 |
| K562 | TDS | 8 | 6.10 (0.89) | 0.36 | 0.46 | 7.58 | 0.16 | 13.1 | 16 |
| K562 | TDS | 128 | 5.82 (0.29) | 0.41 | 0.57 | 5.24 | 0.13 | 9.1 | 116 |
| K562 | TDS | 1024 | 6.22 (1.25) | 0.30 | 0.54 | 5.66 | 0.09 | 6.8 | 856 |
| K562 | TDS | 2048 | 5.40 (1.77) | 0.35 | 0.53 | 5.76 | 0.22 | 17.8 | 1718 |
| SK-N-SH | SMC | 8 | 5.51 (0.80) | 0.49 | 0.65 | 8.44 | 0.40 | 33.4 | 9 |
| SK-N-SH | SMC | 128 | 5.82 (0.74) | 0.47 | 0.66 | 6.03 | 0.24 | 17.5 | 108 |
| SK-N-SH | SMC | 1024 | 5.88 (0.36) | 0.43 | 0.62 | 6.78 | 0.32 | 25.9 | 849 |
| SK-N-SH | SMC | 2048 | 5.15 (0.07) | 0.39 | 0.66 | 5.55 | 0.40 | 32.2 | 1705 |
| SK-N-SH | SVDD | 8 | 2.14 (0.09) | 0.79 | 0.93 | 2.01 | 1.98 | 149.8 | 9 |
| SK-N-SH | SVDD | 128 | 4.04 (0.14) | 0.83 | 0.93 | 2.17 | 1.98 | 149.6 | 108 |
| SK-N-SH | SVDD | 1024 | 4.88 (0.04) | 0.81 | 0.92 | 2.31 | 1.97 | 149.1 | 851 |
| SK-N-SH | SVDD | 2048 | 5.14 (0.06) | 0.80 | 0.89 | 2.45 | 1.97 | 149.2 | 1718 |
| SK-N-SH | TDS | 8 | 4.22 (1.60) | 0.45 | 0.60 | 9.74 | 0.10 | 7.4 | 16 |
| SK-N-SH | TDS | 128 | 4.28 (0.14) | 0.39 | 0.68 | 5.84 | 0.24 | 20.7 | 115 |
| SK-N-SH | TDS | 1024 | 4.89 (1.74) | 0.38 | 0.58 | 6.93 | 0.09 | 7.3 | 858 |
| SK-N-SH | TDS | 2048 | 3.72 (1.75) | 0.40 | 0.65 | 6.33 | 0.25 | 21.1 | 1705 |

# F Complete Benchmark Results

## F.1 Complete Branching-Factor Results

Tables 9–11 report absolute predicted-specificity and fidelity measurements underlying the branching-factor comparison. Each entry is the mean (standard deviation) across three runs. PACS, greedy selection, and best-of-$N$ are reported at each branching factor $B \in \{2, 4, 8, 16, 32, 64, 128, 256, 512, 1024\}$. For best-of-$N$, $B$ identifies the search configuration whose design-oracle budget is matched. Ancestral sampling is independent of $B$ and is listed once per target. Higher AlphaGenome MinGap and correlations indicate better scores under the respective measures; lower NTv2 FBD indicates closer agreement with the natural-reference embedding distribution.

Table 9: Absolute branching-factor results for HepG2. Values are mean (SD) over three runs. The ancestral reference is shown once because it does not depend on branching factor.

| Method | $B$ | AlphaGenome MinGap | JASPAR motif correlation | 3-mer correlation | NTv2 FBD |
|---|---|---|---|---|---|
| Ancestral | – | -0.158 (0.063) | 0.774 (0.028) | 0.958 (0.005) | 1.672 (0.017) |

Continued on next page

Table 9 (continued)

| Method | $B$ | AlphaGenome MinGap | JASPAR motif correlation | 3-mer correlation | NTv2 FBD |
|---|---|---|---|---|---|
| Best-of-N | 2 | 0.934 (0.088) | 0.787 (0.028) | 0.928 (0.020) | 1.730 (0.050) |
| Best-of-N | 4 | 1.258 (0.152) | 0.838 (0.005) | 0.954 (0.011) | 1.959 (0.412) |
| Best-of-N | 8 | 1.648 (0.226) | 0.874 (0.012) | 0.970 (0.010) | 1.689 (0.033) |
| Best-of-N | 16 | 2.075 (0.276) | 0.868 (0.027) | 0.974 (0.005) | 1.701 (0.019) |
| Best-of-N | 32 | 2.452 (0.168) | 0.869 (0.016) | 0.972 (0.003) | 1.688 (0.035) |
| Best-of-N | 64 | 2.830 (0.206) | 0.853 (0.013) | 0.966 (0.001) | 1.725 (0.027) |
| Best-of-N | 128 | 3.246 (0.175) | 0.845 (0.014) | 0.959 (0.003) | 1.755 (0.026) |
| Best-of-N | 256 | 3.543 (0.188) | 0.843 (0.015) | 0.960 (0.003) | 1.740 (0.029) |
| Best-of-N | 512 | 3.913 (0.179) | 0.809 (0.022) | 0.949 (0.007) | 1.795 (0.032) |
| Best-of-N | 1024 | 4.223 (0.149) | 0.792 (0.030) | 0.940 (0.007) | 1.801 (0.048) |
| Greedy selection | 2 | 0.594 (0.080) | 0.750 (0.033) | 0.912 (0.015) | 2.423 (0.058) |
| Greedy selection | 4 | 1.431 (0.100) | 0.642 (0.033) | 0.855 (0.014) | 2.188 (0.056) |
| Greedy selection | 8 | 2.193 (0.160) | 0.580 (0.051) | 0.774 (0.013) | 2.421 (0.109) |
| Greedy selection | 16 | 3.121 (0.043) | 0.572 (0.033) | 0.754 (0.056) | 2.399 (0.273) |
| Greedy selection | 32 | 3.866 (0.015) | 0.601 (0.027) | 0.790 (0.021) | 2.263 (0.102) |
| Greedy selection | 64 | 4.556 (0.057) | 0.649 (0.040) | 0.820 (0.025) | 2.183 (0.071) |
| Greedy selection | 128 | 5.231 (0.039) | 0.723 (0.024) | 0.886 (0.021) | 2.008 (0.060) |
| Greedy selection | 256 | 5.495 (0.035) | 0.767 (0.042) | 0.899 (0.011) | 2.026 (0.036) |
| Greedy selection | 512 | 5.867 (0.114) | 0.769 (0.023) | 0.918 (0.010) | 2.024 (0.070) |
| Greedy selection | 1024 | 6.004 (0.071) | 0.816 (0.001) | 0.925 (0.011) | 1.975 (0.033) |
| PACS | 2 | 0.612 (0.129) | 0.704 (0.041) | 0.903 (0.025) | 2.240 (0.214) |
| PACS | 4 | 1.299 (0.098) | 0.738 (0.020) | 0.898 (0.002) | 2.075 (0.138) |
| PACS | 8 | 2.284 (0.175) | 0.708 (0.043) | 0.908 (0.027) | 1.919 (0.167) |
| PACS | 16 | 3.070 (0.056) | 0.754 (0.065) | 0.919 (0.023) | 1.823 (0.085) |
| PACS | 32 | 3.929 (0.135) | 0.774 (0.043) | 0.918 (0.019) | 1.783 (0.054) |
| PACS | 64 | 4.547 (0.134) | 0.749 (0.035) | 0.918 (0.019) | 1.883 (0.070) |
| PACS | 128 | 5.281 (0.121) | 0.817 (0.044) | 0.937 (0.018) | 1.798 (0.058) |
| PACS | 256 | 5.532 (0.038) | 0.826 (0.018) | 0.947 (0.013) | 1.782 (0.036) |
| PACS | 512 | 5.901 (0.196) | 0.833 (0.032) | 0.956 (0.007) | 1.806 (0.014) |
| PACS | 1024 | 6.106 (0.143) | 0.849 (0.013) | 0.952 (0.012) | 1.790 (0.023) |

Table 10: Absolute branching-factor results for K562. Values are mean (SD) over three runs. The ancestral reference is shown once because it does not depend on branching factor.

| Method | $B$ | AlphaGenome MinGap | JASPAR motif correlation | 3-mer correlation | NTv2 FBD |
|---|---|---|---|---|---|
| Ancestral | – | -0.092 (0.012) | 0.802 (0.056) | 0.959 (0.006) | 1.647 (0.017) |
| Best-of-N | 2 | 1.652 (0.113) | 0.906 (0.012) | 0.964 (0.004) | 1.670 (0.008) |
| Best-of-N | 4 | 2.039 (0.044) | 0.906 (0.002) | 0.965 (0.013) | 1.896 (0.384) |
| Best-of-N | 8 | 2.462 (0.117) | 0.902 (0.004) | 0.958 (0.009) | 1.664 (0.022) |
| Best-of-N | 16 | 2.813 (0.075) | 0.906 (0.009) | 0.959 (0.011) | 1.674 (0.046) |
| Best-of-N | 32 | 3.224 (0.025) | 0.909 (0.006) | 0.963 (0.013) | 1.669 (0.014) |
| Best-of-N | 64 | 3.728 (0.076) | 0.904 (0.009) | 0.961 (0.006) | 1.692 (0.016) |
| Best-of-N | 128 | 4.097 (0.115) | 0.907 (0.012) | 0.958 (0.005) | 1.739 (0.023) |
| Best-of-N | 256 | 4.455 (0.065) | 0.907 (0.006) | 0.957 (0.001) | 1.693 (0.024) |
| Best-of-N | 512 | 4.656 (0.019) | 0.897 (0.006) | 0.954 (0.003) | 1.732 (0.046) |
| Best-of-N | 1024 | 4.950 (0.049) | 0.899 (0.008) | 0.948 (0.011) | 1.744 (0.025) |
| Greedy selection | 2 | 0.813 (0.046) | 0.807 (0.011) | 0.548 (0.014) | 18.120 (1.484) |
| Greedy selection | 4 | 1.836 (0.095) | 0.827 (0.011) | 0.708 (0.037) | 8.343 (1.915) |
| Greedy selection | 8 | 2.695 (0.020) | 0.765 (0.015) | 0.639 (0.059) | 4.816 (0.423) |

Continued on next page

Table 10 (continued)

| Method | $B$ | AlphaGenome MinGap | JASPAR motif correlation | 3-mer correlation | NTv2 FBD |
|---|---|---|---|---|---|
| Greedy selection | 16 | 3.402 (0.103) | 0.692 (0.017) | 0.571 (0.019) | 4.750 (0.526) |
| Greedy selection | 32 | 4.064 (0.037) | 0.720 (0.001) | 0.617 (0.003) | 4.161 (0.040) |
| Greedy selection | 64 | 4.808 (0.107) | 0.751 (0.016) | 0.688 (0.038) | 3.475 (0.156) |
| Greedy selection | 128 | 5.431 (0.088) | 0.790 (0.021) | 0.802 (0.034) | 2.654 (0.274) |
| Greedy selection | 256 | 6.001 (0.064) | 0.835 (0.006) | 0.867 (0.012) | 2.160 (0.144) |
| Greedy selection | 512 | 6.212 (0.027) | 0.854 (0.012) | 0.902 (0.004) | 1.903 (0.028) |
| Greedy selection | 1024 | 6.370 (0.112) | 0.864 (0.003) | 0.904 (0.006) | 1.998 (0.090) |
| PACS | 2 | 0.822 (0.106) | 0.851 (0.016) | 0.619 (0.051) | 13.461 (3.519) |
| PACS | 4 | 1.778 (0.166) | 0.895 (0.019) | 0.823 (0.073) | 5.818 (1.453) |
| PACS | 8 | 2.741 (0.263) | 0.917 (0.007) | 0.938 (0.045) | 2.135 (0.525) |
| PACS | 16 | 3.611 (0.060) | 0.914 (0.007) | 0.961 (0.006) | 1.936 (0.079) |
| PACS | 32 | 4.310 (0.072) | 0.927 (0.007) | 0.969 (0.006) | 2.017 (0.163) |
| PACS | 64 | 5.011 (0.250) | 0.910 (0.006) | 0.961 (0.005) | 1.759 (0.112) |
| PACS | 128 | 5.368 (0.169) | 0.924 (0.004) | 0.965 (0.003) | 1.914 (0.087) |
| PACS | 256 | 5.866 (0.149) | 0.892 (0.008) | 0.959 (0.004) | 1.743 (0.088) |
| PACS | 512 | 6.220 (0.051) | 0.891 (0.006) | 0.951 (0.008) | 1.778 (0.047) |
| PACS | 1024 | 6.345 (0.121) | 0.887 (0.003) | 0.956 (0.001) | 1.821 (0.053) |

Table 11: Absolute branching-factor results for SK-N-SH. Values are mean (SD) over three runs. The ancestral reference is shown once because it does not depend on branching factor.

| Method | $B$ | AlphaGenome MinGap | JASPAR motif correlation | 3-mer correlation | NTv2 FBD |
|---|---|---|---|---|---|
| Ancestral | – | -0.291 (0.042) | 0.743 (0.042) | 0.934 (0.003) | 1.810 (0.030) |
| Best-of-N | 2 | 0.893 (0.076) | 0.835 (0.022) | 0.936 (0.010) | 1.769 (0.049) |
| Best-of-N | 4 | 1.087 (0.063) | 0.873 (0.017) | 0.945 (0.011) | 1.756 (0.024) |
| Best-of-N | 8 | 1.291 (0.056) | 0.873 (0.025) | 0.964 (0.003) | 1.748 (0.036) |
| Best-of-N | 16 | 1.610 (0.060) | 0.876 (0.023) | 0.963 (0.004) | 1.821 (0.030) |
| Best-of-N | 32 | 1.820 (0.096) | 0.844 (0.024) | 0.948 (0.003) | 1.907 (0.078) |
| Best-of-N | 64 | 2.041 (0.007) | 0.817 (0.030) | 0.929 (0.007) | 2.026 (0.016) |
| Best-of-N | 128 | 2.244 (0.091) | 0.803 (0.028) | 0.922 (0.006) | 2.102 (0.007) |
| Best-of-N | 256 | 2.537 (0.062) | 0.772 (0.029) | 0.917 (0.010) | 2.117 (0.068) |
| Best-of-N | 512 | 2.726 (0.068) | 0.757 (0.035) | 0.900 (0.011) | 2.204 (0.060) |
| Best-of-N | 1024 | 3.021 (0.061) | 0.753 (0.015) | 0.891 (0.011) | 2.270 (0.060) |
| Greedy selection | 2 | 1.130 (0.038) | 0.826 (0.021) | 0.855 (0.009) | 3.883 (0.291) |
| Greedy selection | 4 | 1.901 (0.138) | 0.820 (0.023) | 0.915 (0.005) | 2.162 (0.099) |
| Greedy selection | 8 | 2.452 (0.063) | 0.777 (0.018) | 0.900 (0.009) | 2.083 (0.014) |
| Greedy selection | 16 | 3.075 (0.014) | 0.783 (0.026) | 0.909 (0.008) | 2.089 (0.025) |
| Greedy selection | 32 | 3.420 (0.056) | 0.788 (0.032) | 0.918 (0.003) | 2.061 (0.035) |
| Greedy selection | 64 | 3.900 (0.092) | 0.817 (0.042) | 0.922 (0.007) | 2.131 (0.048) |
| Greedy selection | 128 | 4.387 (0.079) | 0.823 (0.027) | 0.917 (0.008) | 2.186 (0.069) |
| Greedy selection | 256 | 4.704 (0.022) | 0.815 (0.030) | 0.917 (0.004) | 2.232 (0.052) |
| Greedy selection | 512 | 4.924 (0.076) | 0.820 (0.017) | 0.923 (0.005) | 2.262 (0.047) |
| Greedy selection | 1024 | 5.165 (0.165) | 0.795 (0.006) | 0.912 (0.003) | 2.270 (0.031) |
| PACS | 2 | 1.077 (0.068) | 0.797 (0.025) | 0.872 (0.010) | 3.324 (0.220) |
| PACS | 4 | 1.747 (0.116) | 0.801 (0.038) | 0.911 (0.012) | 2.309 (0.240) |
| PACS | 8 | 2.329 (0.051) | 0.835 (0.023) | 0.934 (0.007) | 1.991 (0.091) |
| PACS | 16 | 2.902 (0.044) | 0.823 (0.015) | 0.932 (0.005) | 1.909 (0.015) |
| PACS | 32 | 3.409 (0.076) | 0.799 (0.007) | 0.921 (0.002) | 1.977 (0.035) |
| PACS | 64 | 3.922 (0.035) | 0.833 (0.018) | 0.937 (0.007) | 1.969 (0.008) |

Continued on next page

Table 11 (continued)

| Method | $B$ | AlphaGenome MinGap | JASPAR motif correlation | 3-mer correlation | NTv2 FBD |
|---|---|---|---|---|---|
| PACS | 128 | 4.245 (0.067) | 0.835 (0.021) | 0.937 (0.003) | 2.009 (0.099) |
| PACS | 256 | 4.601 (0.016) | 0.836 (0.016) | 0.935 (0.011) | 2.064 (0.056) |
| PACS | 512 | 4.704 (0.026) | 0.817 (0.010) | 0.920 (0.011) | 2.065 (0.036) |
| PACS | 1024 | 5.044 (0.173) | 0.827 (0.019) | 0.913 (0.009) | 2.095 (0.113) |

### F.2 Complete Human-Cell-Line Benchmark Results

Tables 12–14 report the full absolute benchmark measurements, grouped into predicted specificity, sequence fidelity, and sequence composition and diversity. Entries give means with standard deviations across three runs in parentheses. MinGap is reported separately for the Enformer, Borzoi, and AlphaGenome evaluation models. Log-sum-exp specificity and mean-difference specificity are retained as separate measurements. Motif co-occurrence, inter-motif spacing JSD, and relative motif orientation complement the marginal motif and 3-mer correlations. Embedding fidelity is reported using NTv2 FBD. GC content is reported in its original fractional units without assigning a preferred direction.

Levenshtein distance and positional Shannon entropy summarize sequence diversity. Bracketed annotations $[n = a/b]$ retain the number of valid runs for a metric; a single valid run has no reported standard deviation, and a dash denotes no valid measurement.

Figure 8 summarizes the predicted-specificity–fidelity trade-offs underlying the human-cell-line benchmark.

Table 12: Predicted specificity for the three human cell lines. Values are mean (SD) across three runs; valid-run annotations are retained where fewer runs are available.

| Method | MinGap Enformer | MinGap Borzoi | MinGap AlphaGenome | Log-sum-exp Enformer specificity | Mean-difference Enformer specificity |
|---|---|---|---|---|---|
| **HepG2** | | | | | |
| SMC | 1.586 (2.364) | 1.017 (1.593) | 1.140 (1.715) | 1.025 (2.324) | 1.739 (2.313) |
| TDS | 0.823 (0.869) | 0.616 (0.720) | 0.987 (1.174) | 0.218 (0.870) | 0.919 (0.871) |
| SVDD | 3.024 (0.075) | 2.144 (0.013) | 2.299 (0.107) | 2.511 (0.069) | 3.241 (0.067) |
| DFKC | 5.910 (0.562) | 4.179 (0.434) | 4.905 (0.574) | 5.359 (0.545) | 6.077 (0.534) |
| DRAKES | -1.314 (0.024) | -1.100 (0.040) | -1.351 (0.042) | -1.811 (0.049) | -1.072 (0.061) |
| D3 | -0.075 (0.021) | -0.060 (0.011) | -0.093 (0.007) | -0.616 (0.020) | 0.110 (0.019) |
| Ledidi | 7.127 (0.126) | 5.295 (0.092) | 5.976 (0.068) | 6.555 (0.116) | 7.263 (0.114) |
| Ctrl-DNA | 7.464 (0.270) | 6.961 (0.614) | 5.970 (0.280) | 7.001 (0.298) | 7.744 (0.303) |
| Greedy selection | 7.581 (0.060) | 5.770 (0.001) | 6.291 (0.053) | 7.047 (0.059) | 7.764 (0.058) |
| PACS | 7.691 (0.020) | 5.825 (0.023) | 6.372 (0.061) | 7.152 (0.018) | 7.868 (0.018) |
| **K562** | | | | | |
| SMC | 3.193 (0.707) | 2.652 (0.835) | 2.839 (1.068) | 2.720 (0.735) | 3.460 (0.744) |
| TDS | 0.228 (0.513) | 0.525 (0.857) | 0.238 (0.691) | -0.280 (0.578) | 0.447 (0.593) |
| SVDD | 3.229 (0.156) | 2.737 (0.128) | 2.854 (0.059) | 2.729 (0.149) | 3.463 (0.145) |
| DFKC | 7.488 (0.071) | 6.347 (0.190) | 6.268 (0.118) | 7.178 (0.084) | 8.011 (0.093) |
| DRAKES | -0.033 (0.032) | -0.027 (0.047) | 0.077 (0.022) | -0.457 (0.024) | 0.333 (0.018) |
| D3 | 0.003 (0.070) | -0.042 (0.049) | -0.004 (0.056) | -0.566 (0.078) | 0.155 (0.081) |
| Ledidi | 7.988 (0.071) | 6.923 (0.077) | 6.475 (0.012) | 7.659 (0.074) | 8.472 (0.074) |
| Ctrl-DNA | 8.490 (0.153) | 7.656 (0.050) | 6.222 (0.193) | 8.221 (0.136) | 9.090 (0.119) |
| Greedy selection | 8.047 (0.071) | 7.030 (0.085) | 6.389 (0.074) | 7.699 (0.079) | 8.500 (0.084) |
| PACS | 7.883 (0.070) | 6.757 (0.073) | 6.463 (0.162) | 7.528 (0.070) | 8.326 (0.071) |
| **SK-N-SH** | | | | | |
| SMC | 0.227 (0.166) | 0.130 (0.150) | 0.204 (0.049) | -0.352 (0.155) | 0.363 (0.161) |

Continued on next page

Table 12 (continued)

| Method | MinGap Enformer | MinGap Borzoi | MinGap AlphaGenome | Log-sum-exp specificity | Mean-difference specificity |
|---|---|---|---|---|---|
| TDS | 1.012 (1.102) | 0.889 (0.961) | 0.652 (0.608) | 0.530 (1.252) | 1.287 (1.331) |
| SVDD | 2.619 (0.112) | 1.956 (0.056) | 2.210 (0.093) | 2.253 (0.122) | 3.113 (0.138) |
| DFKC | 3.674 (0.072) | 3.121 (0.134) | 3.799 (0.112) | 3.435 (0.080) | 4.390 (0.097) |
| DRAKES | -0.081 (0.040) | -0.036 (0.040) | -0.151 (0.067) | -0.369 (0.042) | 0.496 (0.045) |
| D3 | -0.111 (0.014) | -0.158 (0.040) | -0.127 (0.040) | -0.683 (0.013) | 0.037 (0.019) |
| Ledidi | 3.651 (0.198) | 2.888 (0.152) | 3.210 (0.206) | 3.353 (0.183) | 4.261 (0.165) |
| Ctrl-DNA | 3.988 (0.856) | 3.650 (0.886) | 3.569 (0.412) | 3.717 (0.910) | 4.706 (0.990) |
| Greedy selection | 5.735 (0.009) | 4.801 (0.072) | 5.387 (0.036) | 5.473 (0.014) | 6.380 (0.019) |
| PACS | 5.529 (0.072) | 4.546 (0.054) | 5.125 (0.120) | 5.285 (0.061) | 6.226 (0.062) |

Table 13: Sequence fidelity for the three human cell lines. Values are mean (SD) across three runs; valid-run annotations are retained where fewer runs are available.

| Method | JASPAR motif correlation | 3-mer correlation | TFBS pair co-occurrence | Inter-motif spacing JSD | Motif orientation correlation | NTv2 FBD |
|---|---|---|---|---|---|---|
| **HepG2** | | | | | | |
| SMC | 0.290 (0.112) | 0.603 (0.076) | 0.117 (0.124) | 0.141 (0.023) | 0.480 (0.520) | 4.983 (0.320) |
| TDS | 0.330 (0.020) | 0.622 (0.024) | 0.027 (0.034) | 0.253 (0.027) | 0.577 (0.272) | 4.772 (0.286) |
| SVDD | 0.632 (0.050) | 0.842 (0.018) | 0.404 (0.011) | 0.007 (0.002) | 0.620 (0.104) | 2.067 (0.054) |
| DFKC | 0.448 (0.070) | 0.672 (0.034) | 0.279 (0.006) | 0.020 (0.003) | 0.598 (0.133) | 3.347 (0.110) |
| DRAKES | -0.126 (0.014) | -0.361 (0.012) | 0.225 (0.046) | 0.037 (0.012) | 0.686 (0.198) | 13.231 (1.080) |
| D3 | 0.793 (0.009) | 0.977 (0.006) | 0.271 (0.016) | 0.022 (0.006) | 0.452 (0.147) | 1.629 (0.013) |
| Ledidi | 0.539 (0.017) | 0.685 (0.034) | 0.338 (0.026) | 0.009 (0.004) | 0.764 (0.003) | 2.442 (0.124) |
| Ctrl-DNA | 0.545 (0.063) | 0.568 (0.098) | 0.208 (0.086) | 0.022 (0.002) | 0.455 (0.115) | 3.745 (0.352) |
| Greedy selection | 0.830 (0.022) | 0.932 (0.005) | 0.349 (0.037) | 0.012 (0.003) | 0.663 (0.078) | 2.039 (0.047) |
| PACS | 0.834 (0.036) | 0.960 (0.008) | 0.326 (0.009) | 0.011 (0.002) | 0.722 (0.037) | 1.789 (0.066) |
| **K562** | | | | | | |
| SMC | 0.408 (0.084) | 0.687 (0.125) | 0.189 (0.056) | 0.083 (0.040) | 0.823 (0.029) | 4.558 (0.565) |
| TDS | 0.430 (0.100) | 0.351 (0.272) | 0.079 (0.015) | 0.164 (0.129) | 0.591 (0.274) | 10.020 (6.994) |
| SVDD | 0.664 (0.034) | 0.515 (0.033) | 0.345 (0.033) | 0.006 (0.002) | 0.775 (0.006) | 4.940 (0.628) |
| DFKC | 0.531 (0.045) | 0.624 (0.024) | 0.201 (0.054) | 0.033 (0.016) | 0.785 (0.082) | 3.296 (0.062) |
| DRAKES | 0.060 (0.008) | -0.354 (0.007) | 0.341 (0.027) | 0.025 (0.008) | 0.664 (0.058) | 8.666 (0.638) |
| D3 | 0.794 (0.057) | 0.969 (0.013) | 0.409 (0.024) | 0.021 (0.003) | 0.713 (0.171) | 1.667 (0.023) |
| Ledidi | 0.650 (0.030) | 0.718 (0.006) | 0.253 (0.027) | 0.015 (0.004) | 0.790 (0.140) | 2.272 (0.046) |
| Ctrl-DNA | 0.528 (0.121) | 0.388 (0.043) | 0.069 (0.041) | 0.053 (0.016) | 0.759 (0.074) | 4.911 (0.356) |
| Greedy selection | 0.837 (0.013) | 0.901 (0.006) | 0.318 (0.027) | 0.014 (0.001) | 0.899 (0.006) | 2.095 (0.050) |
| PACS | 0.865 (0.003) | 0.942 (0.005) | 0.385 (0.052) | 0.015 (0.005) | 0.847 (0.014) | 1.760 (0.012) |
| **SK-N-SH** | | | | | | |
| SMC | 0.500 (0.044) | 0.766 (0.071) | 0.162 (0.130) | 0.086 (0.043) | 0.713 (0.161) | 5.360 (2.693) |
| TDS | 0.343 (0.025) | 0.713 (0.049) | 0.045 (0.085) | 0.188 (0.122) | 0.806 (0.121) | 5.579 (0.550) |
| SVDD | 0.796 (0.013) | 0.923 (0.007) | 0.637 (0.036) | 0.013 (0.001) | 0.910 (0.010) | 1.998 (0.029) |
| DFKC | 0.504 (0.022) | 0.508 (0.061) | 0.315 (0.021) | 0.038 (0.020) | 0.880 (0.034) | 3.720 (0.053) |
| DRAKES | -0.001 (0.009) | -0.382 (0.001) | 0.140 (0.122) | 0.066 (0.031) | — [n=0/3] | 18.779 (0.856) |
| D3 | 0.766 (0.022) | 0.932 (0.017) | 0.432 (0.038) | 0.033 (0.009) | 0.711 (0.098) | 1.785 (0.016) |
| Ledidi | 0.491 (0.013) | 0.524 (0.015) | 0.465 (0.022) | 0.024 (0.007) | 0.902 (0.041) | 2.581 (0.056) |
| Ctrl-DNA | 0.580 (0.042) | 0.431 (0.074) | 0.428 (0.080) | 0.042 (0.050) | 0.871 (0.054) | 4.866 (0.750) |
| Greedy selection | 0.784 (0.013) | 0.897 (0.010) | 0.512 (0.031) | 0.008 (0.003) | 0.896 (0.020) | 2.403 (0.058) |
| PACS | 0.819 (0.029) | 0.915 (0.016) | 0.563 (0.010) | 0.011 (0.003) | 0.877 (0.033) | 2.126 (0.033) |

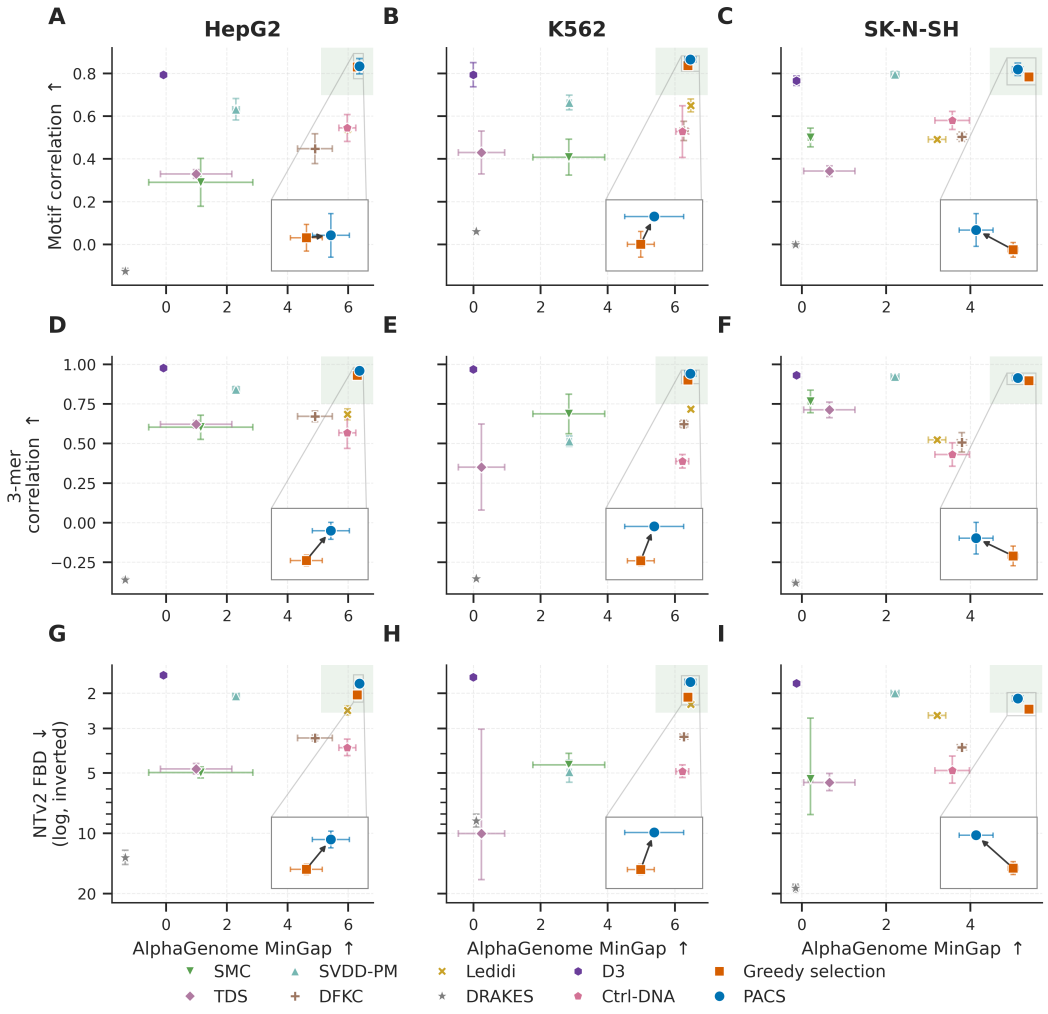


Figure 8: **Predicted specificity–fidelity trade-offs across human cell lines.** AlphaGenome MinGap is plotted against JASPAR motif correlation (A–C), 3-mer correlation (D–F), and NTv2 FBD (G–I). Points and error bars show means and standard deviations across three runs. Insets enlarge the greedy-selection and PACS comparison. Arrows point from greedy selection to PACS. FBD uses an inverted logarithmic axis, so the upper-right direction indicates better performance in every panel.

Table 7: Selector schedule sweep for SK-N-SH. Values are mean (sample SD) across five runs, using final-state completions, $N = 128$, $B = 32$, and $S = 32$. Specificity is AlphaGenome MinGap; motif and 3-mer columns report correlations, and FBD uses NTv2 embeddings. The default PACS configuration is labeled; it is not asserted to be best in every column.

| Selector | $p$ | $\kappa_{\max}$ | MinGap ↑ | Motif ↑ | 3-mer ↑ | FBD ↓ |
|---|---|---|---|---|---|---|
| Greedy | – | $\infty$ | 3.158 (0.115) | 0.796 (0.031) | 0.915 (0.006) | 2.090 (0.085) |
| Constant | – | 10 | 2.941 (0.098) | 0.802 (0.037) | 0.925 (0.012) | 2.115 (0.121) |
| Constant | – | 20 | 3.084 (0.082) | 0.785 (0.018) | 0.919 (0.012) | 2.091 (0.056) |
| Constant | – | 40 | 3.116 (0.071) | 0.793 (0.043) | 0.918 (0.008) | 2.133 (0.037) |
| Step-linear | 1 | 10 | 2.743 (0.127) | 0.814 (0.035) | 0.938 (0.009) | 2.015 (0.079) |
| Step-linear | 1 | 20 | 3.151 (0.130) | 0.809 (0.015) | 0.935 (0.010) | 2.031 (0.035) |
| Step-linear | 1 | 40 | 3.271 (0.121) | 0.805 (0.017) | 0.932 (0.008) | 2.158 (0.066) |
| Clean-linear | 1 | 10 | 2.945 (0.145) | 0.804 (0.026) | 0.941 (0.005) | 2.046 (0.096) |
| Clean-linear | 1 | 20 | 3.201 (0.112) | 0.811 (0.021) | 0.924 (0.010) | 2.118 (0.084) |
| Clean-linear | 1 | 40 | 3.197 (0.131) | 0.815 (0.016) | 0.927 (0.014) | 2.135 (0.062) |
| Clean-cosine | 1 | 10 | 2.991 (0.110) | 0.812 (0.035) | 0.945 (0.010) | 1.983 (0.049) |
| Clean-cosine | 1 | 20 | 3.230 (0.100) | 0.821 (0.030) | 0.936 (0.008) | 2.020 (0.050) |
| Clean-cosine | 1 | 40 | 3.184 (0.099) | 0.815 (0.004) | 0.930 (0.010) | 2.045 (0.053) |
| Geometric | – | 10 | 0.972 (0.067) | 0.811 (0.029) | 0.948 (0.003) | 1.787 (0.022) |
| Geometric | – | 20 | 1.692 (0.115) | 0.806 (0.039) | 0.951 (0.010) | 1.849 (0.063) |
| Geometric | – | 40 | 2.332 (0.062) | 0.817 (0.013) | 0.950 (0.005) | 1.863 (0.071) |
| Clean-smoothstep | 1 | 10 | 2.966 (0.084) | 0.809 (0.032) | 0.942 (0.010) | 2.017 (0.056) |
| Clean-smoothstep | 1 | 20 | 3.232 (0.109) | 0.812 (0.029) | 0.936 (0.007) | 2.033 (0.066) |
| Clean-smoothstep | 1 | 40 | 3.179 (0.071) | 0.809 (0.009) | 0.927 (0.011) | 2.071 (0.060) |
| Clean-smoothstep | 2 | 10 | 2.828 (0.044) | 0.845 (0.013) | 0.951 (0.005) | 1.860 (0.055) |
| Clean-smoothstep | 2 | 20 | 3.094 (0.095) | 0.822 (0.018) | 0.947 (0.006) | 1.936 (0.065) |
| **PACS (default)** | 2 | 40 | 3.142 (0.091) | 0.809 (0.016) | 0.929 (0.009) | 1.982 (0.040) |
| Clean-smoothstep | 4 | 40 | 2.994 (0.108) | 0.835 (0.033) | 0.952 (0.004) | 1.839 (0.036) |

Table 14: Sequence composition and diversity for the three human cell lines. Values are mean (SD) across three runs; valid-run annotations are retained where fewer runs are available.

| Method | Levenshtein distance | Positional Shannon entropy | GC content |
|---|---|---|---|
| **HepG2** | | | |
| SMC | 96.470 (52.084) | 1.234 (0.638) | 0.501 (0.067) |
| TDS | 65.447 (18.829) | 0.833 (0.252) | 0.476 (0.050) |
| SVDD | 149.874 (0.020) | 1.981 (0.000) | 0.513 (0.003) |
| DFKC | 146.183 (0.088) | 1.926 (0.002) | 0.501 (0.006) |
| DRAKES | 141.970 (0.118) | 1.864 (0.002) | 0.698 (0.002) |
| D3 | 149.687 (0.217) | 1.978 (0.003) | 0.458 (0.009) |
| Ledidi | 149.840 (0.086) | 1.981 (0.001) | 0.475 (0.002) |
| Ctrl-DNA | 145.024 (2.293) | 1.914 (0.033) | 0.458 (0.032) |
| Greedy selection | 149.831 (0.064) | 1.981 (0.001) | 0.483 (0.001) |
| PACS | 149.845 (0.040) | 1.981 (0.001) | 0.470 (0.003) |
| **K562** | | | |
| SMC | 72.629 (34.786) | 0.913 (0.386) | 0.479 (0.017) |
| TDS | 39.492 (25.934) | 0.482 (0.332) | 0.616 (0.087) |
| SVDD | 146.720 (0.557) | 1.936 (0.008) | 0.617 (0.009) |
| DFKC | 146.367 (0.253) | 1.929 (0.004) | 0.493 (0.011) |
| DRAKES | 148.352 (0.213) | 1.958 (0.003) | 0.567 (0.003) |
| D3 | 149.809 (0.171) | 1.980 (0.002) | 0.468 (0.010) |
| Ledidi | 149.933 (0.111) | 1.982 (0.002) | 0.490 (0.000) |
| Ctrl-DNA | 141.204 (0.978) | 1.851 (0.019) | 0.480 (0.035) |

Continued on next page

Table 14 (continued)

| Method | Levenshtein distance | Positional Shannon entropy | GC content |
|---|---|---|---|
| Greedy selection | 149.732 (0.108) | 1.979 (0.002) | 0.491 (0.002) |
| PACS | 149.795 (0.132) | 1.980 (0.002) | 0.484 (0.009) |
| **SK-N-SH** | | | |
| SMC | 120.525 (9.384) | 1.515 (0.153) | 0.476 (0.047) |
| TDS | 63.468 (27.163) | 0.772 (0.274) | 0.416 (0.047) |
| SVDD | 149.907 (0.022) | 1.982 (0.000) | 0.480 (0.006) |
| DFKC | 146.509 (0.141) | 1.932 (0.002) | 0.500 (0.009) |
| DRAKES | 139.740 (0.052) | 1.826 (0.001) | 0.720 (0.001) |
| D3 | 149.011 (0.246) | 1.969 (0.004) | 0.433 (0.005) |
| Ledidi | 149.882 (0.119) | 1.981 (0.002) | 0.483 (0.005) |
| Ctrl-DNA | 140.473 (4.902) | 1.847 (0.066) | 0.477 (0.063) |
| Greedy selection | 149.177 (0.065) | 1.971 (0.001) | 0.441 (0.002) |
| PACS | 149.218 (0.070) | 1.972 (0.001) | 0.445 (0.006) |

### F.3 Complete Immune-Cell Benchmark Results

Tables 15–17 report the full absolute benchmark measurements, grouped into predicted specificity, sequence fidelity, and sequence composition and diversity. Entries give means with standard deviations across three runs in parentheses. MinGap is reported separately for the Enformer, Borzoi, and AlphaGenome evaluation models. Log-sum-exp specificity and mean-difference specificity are retained as separate measurements computed with the Enformer model. Motif co-occurrence, inter-motif spacing JSD, and relative motif orientation complement the marginal motif and 3-mer correlations. Embedding fidelity is reported using NTv2 FBD. GC content is reported in its original fractional units without assigning a preferred direction.

Levenshtein distance and positional Shannon entropy summarize sequence diversity. Bracketed annotations $[n = a/b]$ retain the number of valid runs for a metric. A single valid run has no reported standard deviation, and a dash denotes no valid measurement.

Table 15: Predicted specificity for the eight immune cell types. Values are mean (SD) across three runs; valid-run annotations are retained where fewer runs are available.

| Method | MinGap Enformer | MinGap Borzoi | MinGap AlphaGenome | Log-sum-exp Enformer specificity | Mean-difference Enformer specificity |
|---|---|---|---|---|---|
| **CD4$^+$ T** | | | | | |
| SMC | -0.065 (0.069) | -0.062 (0.063) | -0.070 (0.074) | -1.955 (0.015) | -0.006 (0.011) |
| TDS | -0.016 (0.002) | -0.019 (0.004) | -0.016 (0.002) | -1.944 (0.008) | 0.002 (0.008) |
| SVDD | -0.014 (0.007) | -0.036 (0.009) | -0.042 (0.011) | -1.678 (0.001) | 0.292 (0.001) |
| DFKC | -0.034 (0.005) | -0.044 (0.004) | -0.061 (0.005) | -1.759 (0.042) | 0.209 (0.047) |
| DRAKES | -0.068 (0.009) | -0.058 (0.009) | -0.077 (0.009) | -1.954 (0.005) | -0.006 (0.005) |
| D3 | -0.043 (0.002) | -0.039 (0.003) | -0.041 (0.002) | -1.951 (0.002) | -0.004 (0.002) |
| Ledidi | -0.105 (0.006) | -0.106 (0.004) | -0.150 (0.007) | -1.962 (0.002) | -0.014 (0.002) |
| Ctrl-DNA | -0.047 (0.005) | -0.119 (0.020) | -0.042 (0.009) | -1.557 (0.009) | 0.438 (0.008) |
| Greedy selection | 0.093 (0.002) | 0.021 (0.003) | 0.036 (0.008) | -1.438 (0.002) | 0.539 (0.003) |
| PACS | 0.061 (0.007) | -0.006 (0.004) | 0.020 (0.008) | -1.485 (0.005) | 0.492 (0.004) |
| **CD8$^+$ T** | | | | | |
| SMC | -0.022 (0.019) | -0.022 (0.015) | -0.021 (0.015) | -1.939 (0.018) | 0.007 (0.017) |
| TDS | -0.022 (0.018) | -0.017 (0.021) | -0.019 (0.013) | -1.934 (0.015) | 0.012 (0.015) |
| SVDD | 0.044 (0.003) | 0.066 (0.001) | 0.008 (0.006) | -1.637 (0.012) | 0.333 (0.013) |
| DFKC | 0.024 (0.009) | 0.036 (0.006) | -0.018 (0.006) | -1.711 (0.030) | 0.255 (0.033) |
| DRAKES | -0.046 (0.010) | -0.038 (0.011) | -0.045 (0.011) | -1.943 (0.004) | 0.005 (0.004) |

Continued on next page

Table 15 (continued)

| Method | MinGap Enformer | MinGap Borzoi | MinGap AlphaGenome | Log-sum-exp Enformer specificity | Mean-difference Enformer specificity |
|---|---|---|---|---|---|
| D3 | -0.038 (0.002) | -0.031 (0.001) | -0.040 (0.003) | -1.943 (0.002) | 0.004 (0.001) |
| Ledidi | -0.048 (0.003) | -0.033 (0.002) | -0.063 (0.007) | -1.898 (0.001) | 0.050 (0.001) |
| Ctrl-DNA | 0.084 (0.004) | 0.180 (0.013) | 0.115 (0.009) | -1.422 (0.001) | 0.570 (0.001) |
| Greedy selection | 0.064 (0.005) | 0.117 (0.002) | 0.069 (0.004) | -1.460 (0.002) | 0.523 (0.002) |
| PACS | 0.088 (0.002) | 0.107 (0.004) | 0.077 (0.002) | -1.491 (0.004) | 0.490 (0.004) |
| **Naive T** | | | | | |
| SMC | -0.031 (0.017) | -0.045 (0.023) | -0.033 (0.009) | -1.954 (0.005) | -0.008 (0.005) |
| TDS | -0.064 (0.037) | -0.071 (0.045) | -0.069 (0.040) | -1.960 (0.012) | -0.013 (0.011) |
| SVDD | 0.181 (0.005) | 0.100 (0.018) | 0.125 (0.008) | -1.510 (0.004) | 0.448 (0.004) |
| DFKC | 0.079 (0.026) | 0.029 (0.020) | 0.031 (0.025) | -1.709 (0.052) | 0.247 (0.054) |
| DRAKES | -0.028 (0.019) | 0.011 (0.024) | 0.004 (0.014) | -1.773 (0.017) | 0.184 (0.018) |
| D3 | -0.056 (0.008) | -0.049 (0.004) | -0.057 (0.008) | -1.953 (0.005) | -0.006 (0.005) |
| Ledidi | -0.126 (0.007) | -0.156 (0.006) | -0.218 (0.009) | -1.988 (0.003) | -0.041 (0.003) |
| Ctrl-DNA | 0.229 (0.031) | 0.204 (0.036) | 0.248 (0.059) | -1.355 (0.112) | 0.615 (0.119) |
| Greedy selection | 0.624 (0.006) | 0.424 (0.008) | 0.405 (0.006) | -1.107 (0.004) | 0.844 (0.004) |
| PACS | 0.537 (0.009) | 0.372 (0.007) | 0.380 (0.008) | -1.153 (0.008) | 0.801 (0.007) |
| **NKT** | | | | | |
| SMC | -0.039 (0.023) | -0.038 (0.019) | -0.077 (0.060) | -1.952 (0.014) | -0.005 (0.014) |
| TDS | -0.016 (0.005) | -0.017 (0.007) | -0.013 (0.006) | -1.945 (0.004) | 0.001 (0.004) |
| SVDD | 0.069 (0.014) | -0.051 (0.008) | 0.060 (0.013) | -1.653 (0.030) | 0.310 (0.032) |
| DFKC | 0.038 (0.014) | -0.053 (0.007) | 0.033 (0.027) | -1.731 (0.044) | 0.231 (0.047) |
| DRAKES | -0.063 (0.011) | -0.054 (0.005) | -0.063 (0.016) | -1.954 (0.002) | -0.007 (0.001) |
| D3 | -0.037 (0.007) | -0.037 (0.006) | -0.036 (0.007) | -1.948 (0.002) | -0.001 (0.002) |
| Ledidi | -0.122 (0.005) | -0.127 (0.003) | -0.184 (0.004) | -1.983 (0.002) | -0.036 (0.002) |
| Ctrl-DNA | 0.130 (0.039) | -0.123 (0.021) | 0.218 (0.028) | -1.459 (0.025) | 0.523 (0.022) |
| Greedy selection | 0.348 (0.002) | -0.031 (0.006) | 0.253 (0.007) | -1.373 (0.004) | 0.583 (0.004) |
| PACS | 0.306 (0.009) | -0.038 (0.003) | 0.225 (0.020) | -1.397 (0.017) | 0.562 (0.017) |
| **Memory B** | | | | | |
| SMC | -0.042 (0.030) | -0.053 (0.014) | -0.035 (0.016) | -1.955 (0.006) | -0.008 (0.004) |
| TDS | -0.055 (0.055) | -0.056 (0.059) | -0.063 (0.059) | -1.960 (0.014) | -0.013 (0.012) |
| SVDD | 0.030 (0.040) | -0.136 (0.020) | 0.039 (0.033) | -1.571 (0.013) | 0.396 (0.011) |
| DFKC | 0.032 (0.018) | -0.055 (0.009) | 0.041 (0.019) | -1.753 (0.039) | 0.202 (0.041) |
| DRAKES | -0.079 (0.015) | -0.073 (0.007) | -0.082 (0.015) | -1.962 (0.002) | -0.014 (0.002) |
| D3 | -0.046 (0.002) | -0.044 (0.004) | -0.042 (0.001) | -1.955 (0.003) | -0.008 (0.003) |
| Ledidi | -0.151 (0.005) | -0.171 (0.007) | -0.221 (0.007) | -2.016 (0.002) | -0.069 (0.002) |
| Ctrl-DNA | -0.044 (0.007) | -0.045 (0.012) | -0.043 (0.008) | -1.958 (0.004) | -0.011 (0.004) |
| Greedy selection | 0.474 (0.002) | -0.032 (0.007) | 0.477 (0.011) | -1.150 (0.002) | 0.809 (0.002) |
| PACS | 0.312 (0.008) | -0.078 (0.010) | 0.342 (0.021) | -1.204 (0.003) | 0.765 (0.003) |
| **Plasma B** | | | | | |
| SMC | -0.019 (0.014) | -0.018 (0.012) | -0.025 (0.017) | -1.946 (0.005) | 0.000 (0.005) |
| TDS | -0.037 (0.019) | -0.032 (0.022) | -0.023 (0.009) | -1.948 (0.006) | -0.002 (0.006) |
| SVDD | 0.238 (0.022) | 0.175 (0.008) | 0.207 (0.011) | -1.448 (0.011) | 0.508 (0.011) |
| DFKC | 0.203 (0.022) | 0.182 (0.013) | 0.217 (0.015) | -1.472 (0.021) | 0.486 (0.023) |
| DRAKES | -0.045 (0.008) | -0.040 (0.007) | -0.053 (0.009) | -1.939 (0.005) | 0.008 (0.005) |
| D3 | -0.033 (0.004) | -0.028 (0.006) | -0.034 (0.006) | -1.944 (0.002) | 0.002 (0.002) |
| Ledidi | -0.081 (0.001) | -0.081 (0.003) | -0.106 (0.003) | -1.935 (0.003) | 0.013 (0.003) |
| Ctrl-DNA | 0.458 (0.063) | 0.383 (0.018) | 0.691 (0.029) | -1.184 (0.024) | 0.776 (0.021) |
| Greedy selection | 0.787 (0.003) | 0.414 (0.004) | 0.797 (0.005) | -1.039 (0.001) | 0.908 (0.001) |
| PACS | 0.690 (0.011) | 0.394 (0.003) | 0.709 (0.007) | -1.085 (0.006) | 0.864 (0.006) |
| **Macrophage** | | | | | |
| SMC | 0.056 (0.027) | 0.060 (0.050) | 0.066 (0.038) | -1.864 (0.033) | 0.082 (0.033) |

Continued on next page

Table 15 (continued)

| Method | MinGap Enformer | MinGap Borzoi | MinGap AlphaGenome | Log-sum-exp Enformer specificity | Mean-difference Enformer specificity |
|---|---|---|---|---|---|
| TDS | 0.053 (0.036) | 0.061 (0.023) | 0.033 (0.020) | -1.880 (0.048) | 0.066 (0.048) |
| SVDD | 0.707 (0.018) | 0.648 (0.020) | 0.651 (0.025) | -1.172 (0.015) | 0.775 (0.015) |
| DFKC | 0.660 (0.018) | 0.584 (0.043) | 0.590 (0.011) | -1.189 (0.024) | 0.759 (0.025) |
| DRAKES | 0.226 (0.032) | 0.245 (0.031) | 0.254 (0.035) | -1.565 (0.022) | 0.389 (0.022) |
| D3 | 0.041 (0.003) | 0.044 (0.003) | 0.040 (0.005) | -1.881 (0.003) | 0.065 (0.003) |
| Ledidi | 0.072 (0.013) | 0.044 (0.010) | 0.058 (0.010) | -1.814 (0.011) | 0.133 (0.011) |
| Ctrl-DNA | 0.423 (0.123) | 0.303 (0.113) | 0.361 (0.143) | -1.182 (0.059) | 0.786 (0.052) |
| Greedy selection | 0.949 (0.001) | 0.874 (0.009) | 0.926 (0.004) | -0.973 (0.000) | 0.973 (0.000) |
| PACS | 0.901 (0.003) | 0.849 (0.002) | 0.874 (0.008) | -1.003 (0.002) | 0.943 (0.002) |
| **Mast** | | | | | |
| SMC | -0.047 (0.038) | -0.032 (0.016) | -0.032 (0.024) | -1.954 (0.008) | -0.008 (0.007) |
| TDS | -0.021 (0.012) | -0.018 (0.018) | -0.007 (0.019) | -1.945 (0.007) | 0.001 (0.007) |
| SVDD | 0.035 (0.011) | -0.101 (0.007) | -0.059 (0.009) | -1.608 (0.010) | 0.355 (0.011) |
| DFKC | 0.512 (0.067) | 0.221 (0.037) | 0.360 (0.033) | -1.352 (0.079) | 0.596 (0.079) |
| DRAKES | -0.042 (0.012) | -0.038 (0.005) | -0.033 (0.009) | -1.950 (0.003) | -0.003 (0.003) |
| D3 | -0.039 (0.010) | -0.036 (0.002) | -0.034 (0.006) | -1.944 (0.006) | 0.003 (0.006) |
| Ledidi | -0.056 (0.007) | -0.057 (0.002) | -0.075 (0.004) | -1.912 (0.009) | 0.036 (0.009) |
| Ctrl-DNA | 0.852 (0.099) | 0.396 (0.175) | 0.576 (0.080) | -1.034 (0.045) | 0.913 (0.043) |
| Greedy selection | 0.879 (0.007) | 0.465 (0.008) | 0.629 (0.008) | -1.020 (0.003) | 0.927 (0.003) |
| PACS | 0.864 (0.011) | 0.433 (0.020) | 0.628 (0.020) | -1.024 (0.003) | 0.922 (0.002) |

Table 16: Sequence fidelity for the eight immune cell types. Values are mean (SD) across three runs; valid-run annotations are retained where fewer runs are available.

| Method | JASPAR motif correlation | 3-mer correlation | TFBS pair co-occurrence | Inter-motif spacing JSD | Motif orientation correlation | NTv2 FBD |
|---|---|---|---|---|---|---|
| **CD4$^+$ T** | | | | | | |
| SMC | 0.478 (0.048) | 0.811 (0.056) | 0.380 (0.070) | 0.083 (0.015) | 0.755 (0.025) | 4.439 (0.172) |
| TDS | 0.363 (0.045) | 0.802 (0.073) | 0.051 (0.126) | 0.201 (0.011) | 0.655 (0.218) | 4.977 (0.744) |
| SVDD | 0.714 (0.037) | 0.741 (0.009) | 0.630 (0.010) | 0.032 (0.007) | 0.900 (0.011) | 4.111 (0.207) |
| DFKC | 0.784 (0.007) | 0.960 (0.013) | 0.382 (0.040) | 0.028 (0.001) | 0.823 (0.114) | 2.532 (0.332) |
| DRAKES | 0.848 (0.047) | 0.974 (0.004) | 0.620 (0.037) | 0.011 (0.009) | 0.897 (0.017) | 1.613 (0.046) |
| D3 | 0.855 (0.045) | 0.984 (0.002) | 0.515 (0.038) | 0.010 (0.006) | 0.828 (0.055) | 1.630 (0.046) |
| Ledidi | 0.094 (0.155) | 0.479 (0.039) | 0.337 (0.105) | 0.088 (0.014) | 0.784 [n=1/3] | 2.748 (0.058) |
| Ctrl-DNA | 0.261 (0.031) | 0.461 (0.035) | 0.272 (0.183) | 0.093 (0.013) | 0.481 (0.079) | 8.029 (1.494) |
| Greedy selection | 0.580 (0.019) | 0.783 (0.018) | 0.543 (0.053) | 0.013 (0.003) | 0.913 (0.026) | 3.471 (0.349) |
| PACS | 0.745 (0.029) | 0.868 (0.007) | 0.634 (0.004) | 0.009 (0.004) | 0.903 (0.003) | 2.400 (0.124) |
| **CD8$^+$ T** | | | | | | |
| SMC | 0.443 (0.158) | 0.726 (0.115) | 0.030 (0.088) | 0.197 (0.046) | 0.458 (0.589) | 5.267 (0.651) |
| TDS | 0.360 (0.302) | 0.716 (0.137) | 0.138 (0.099) | 0.181 (0.089) | 0.556 (0.284) | 5.307 (0.442) |
| SVDD | 0.893 (0.025) | 0.901 (0.015) | 0.479 (0.046) | 0.007 (0.000) | 0.874 (0.013) | 2.091 (0.071) |
| DFKC | 0.823 (0.024) | 0.930 (0.008) | 0.364 (0.041) | 0.033 (0.011) | 0.787 (0.046) | 2.887 (0.146) |
| DRAKES | 0.188 (0.013) | 0.590 (0.007) | 0.395 (0.020) | 0.026 (0.008) | 0.887 (0.037) | 2.939 (0.134) |
| D3 | 0.886 (0.030) | 0.979 (0.005) | 0.420 (0.015) | 0.012 (0.006) | 0.881 (0.036) | 1.799 (0.036) |
| Ledidi | 0.084 (0.215) | 0.045 (0.140) | 0.382 (0.085) | 0.081 (0.040) | 0.712 (0.008) [n=2/3] | 3.084 (0.049) |
| Ctrl-DNA | 0.076 (0.054) | 0.221 (0.010) | 0.217 (0.142) | 0.075 (0.030) | 0.730 (0.204) | 13.995 (0.897) |
| Greedy selection | 0.592 (0.032) | 0.560 (0.029) | 0.300 (0.016) | 0.011 (0.001) | 0.899 (0.014) | 4.379 (0.301) |
| PACS | 0.820 (0.043) | 0.789 (0.051) | 0.432 (0.030) | 0.013 (0.003) | 0.893 (0.027) | 2.795 (0.120) |
| **Naive T** | | | | | | |
| SMC | 0.422 (0.123) | 0.618 (0.145) | 0.163 (0.158) | 0.189 (0.054) | 0.643 (0.214) | 4.902 (0.273) |
| TDS | 0.410 (0.080) | 0.769 (0.085) | 0.032 (0.144) | 0.163 (0.052) | 0.603 (0.305) | 4.889 (0.679) |
| SVDD | 0.907 (0.002) | 0.970 (0.008) | 0.749 (0.023) | 0.007 (0.001) | 0.920 (0.014) | 1.775 (0.060) |
| DFKC | 0.671 (0.057) | 0.830 (0.046) | 0.645 (0.061) | 0.041 (0.015) | 0.853 (0.056) | 2.716 (0.059) |

Continued on next page

Table 16 (continued)

| Method | JASPAR motif correlation | 3-mer correlation | TFBS pair co-occurrence | Inter-motif spacing JSD | Motif orientation correlation | NTv2 FBD |
|---|---|---|---|---|---|---|
| DRAKES | 0.859 (0.025) | 0.881 (0.005) | 0.508 (0.036) | 0.006 (0.004) | 0.859 (0.023) | 1.889 (0.041) |
| D3 | 0.930 (0.014) | 0.986 (0.002) | 0.642 (0.021) | 0.018 (0.012) | 0.884 (0.050) | 1.625 (0.065) |
| Ledidi | 0.105 (0.154) | 0.061 (0.128) | 0.383 (0.106) | 0.086 (0.035) | 0.896 [n=1/3] | 2.780 (0.040) |
| Ctrl-DNA | 0.644 (0.052) | 0.684 (0.038) | 0.301 (0.155) | 0.040 (0.053) | 0.815 (0.128) | 4.198 (1.164) |
| Greedy selection | 0.852 (0.003) | 0.938 (0.002) | 0.522 (0.053) | 0.018 (0.006) | 0.914 (0.017) | 2.177 (0.025) |
| PACS | 0.874 (0.011) | 0.950 (0.007) | 0.625 (0.093) | 0.012 (0.008) | 0.936 (0.013) | 1.912 (0.038) |
| **NKT** | | | | | | |
| SMC | 0.384 (0.124) | 0.780 (0.065) | 0.086 (0.117) | 0.172 (0.030) | 0.609 (0.188) | 4.601 (0.321) |
| TDS | 0.353 (0.271) | 0.742 (0.121) | 0.153 (0.175) | 0.135 (0.113) | 0.547 (0.447) | 4.822 (0.322) |
| SVDD | 0.885 (0.006) | 0.957 (0.017) | 0.442 (0.004) | 0.009 (0.005) | 0.770 (0.029) | 1.685 (0.017) |
| DFKC | 0.713 (0.026) | 0.932 (0.023) | 0.329 (0.026) | 0.049 (0.017) | 0.800 (0.091) | 2.515 (0.073) |
| DRAKES | 0.640 (0.033) | 0.867 (0.009) | 0.426 (0.041) | 0.016 (0.002) | 0.806 (0.062) | 1.887 (0.069) |
| D3 | 0.857 (0.047) | 0.976 (0.003) | 0.421 (0.024) | 0.014 (0.007) | 0.835 (0.034) | 1.630 (0.018) |
| Ledidi | 0.078 (0.202) | 0.043 (0.161) | 0.277 (0.167) | 0.081 (0.028) | 0.839 (0.191) [n=2/3] | 2.838 (0.069) |
| Ctrl-DNA | 0.315 (0.026) | 0.455 (0.037) | 0.259 (0.072) | 0.016 (0.011) | 0.769 (0.063) | 6.829 (0.454) |
| Greedy selection | 0.778 (0.008) | 0.765 (0.023) | 0.344 (0.059) | 0.009 (0.002) | 0.806 (0.059) | 3.110 (0.174) |
| PACS | 0.844 (0.001) | 0.882 (0.003) | 0.380 (0.038) | 0.007 (0.001) | 0.827 (0.048) | 2.187 (0.098) |
| **Memory B** | | | | | | |
| SMC | 0.392 (0.026) | 0.785 (0.048) | 0.106 (0.065) | 0.116 (0.041) | 0.626 (0.057) | 4.260 (0.215) |
| TDS | 0.591 (0.038) | 0.798 (0.023) | 0.074 (0.163) | 0.137 (0.040) | 0.623 (0.218) | 4.194 (0.352) |
| SVDD | 0.706 (0.018) | 0.849 (0.039) | 0.317 (0.042) | 0.012 (0.008) | 0.878 (0.009) | 1.940 (0.110) |
| DFKC | 0.729 (0.123) | 0.918 (0.022) | 0.350 (0.062) | 0.026 (0.017) | 0.832 (0.036) | 2.262 (0.105) |
| DRAKES | 0.687 (0.064) | 0.883 (0.002) | 0.413 (0.016) | 0.010 (0.005) | 0.868 (0.045) | 1.776 (0.050) |
| D3 | 0.860 (0.027) | 0.979 (0.008) | 0.347 (0.092) | 0.012 (0.002) | 0.840 (0.089) | 1.587 (0.006) |
| Ledidi | 0.076 (0.181) | 0.063 (0.139) | 0.246 (0.047) | 0.076 (0.013) | 0.935 (0.057) [n=2/3] | 2.704 (0.051) |
| Ctrl-DNA | 0.287 (0.203) | 0.659 (0.147) | 0.307 (0.038) | 0.017 (0.004) | 0.878 (0.036) | 2.263 (0.249) |
| Greedy selection | 0.863 (0.007) | 0.936 (0.011) | 0.300 (0.060) | 0.005 (0.001) | 0.895 (0.015) | 1.775 (0.018) |
| PACS | 0.897 (0.006) | 0.963 (0.007) | 0.342 (0.022) | 0.004 (0.001) | 0.897 (0.013) | 1.674 (0.013) |
| **Plasma B** | | | | | | |
| SMC | 0.454 (0.072) | 0.800 (0.034) | 0.085 (0.024) | 0.138 (0.085) | 0.395 (0.184) | 5.210 (0.964) |
| TDS | 0.348 (0.277) | 0.643 (0.038) | 0.118 (0.036) | 0.152 (0.072) | 0.536 (0.187) | 4.567 (0.343) |
| SVDD | 0.796 (0.003) | 0.796 (0.019) | 0.431 (0.035) | 0.006 (0.003) | 0.721 (0.042) | 2.266 (0.117) |
| DFKC | 0.811 (0.032) | 0.937 (0.015) | 0.278 (0.100) | 0.019 (0.002) | 0.718 (0.087) | 2.548 (0.068) |
| DRAKES | 0.541 (0.058) | 0.838 (0.013) | 0.382 (0.021) | 0.010 (0.004) | 0.836 (0.090) | 1.898 (0.041) |
| D3 | 0.897 (0.015) | 0.978 (0.009) | 0.419 (0.052) | 0.009 (0.003) | 0.805 (0.084) | 1.609 (0.049) |
| Ledidi | 0.024 (0.165) | 0.042 (0.167) | 0.232 (0.113) | 0.076 (0.025) | 0.433 [n=1/3] | 2.748 (0.056) |
| Ctrl-DNA | 0.482 (0.057) | 0.433 (0.033) | 0.175 (0.041) | 0.018 (0.010) | 0.614 (0.268) | 6.413 (1.113) |
| Greedy selection | 0.809 (0.016) | 0.889 (0.012) | 0.190 (0.004) | 0.009 (0.002) | 0.644 (0.065) | 2.261 (0.060) |
| PACS | 0.887 (0.001) | 0.939 (0.005) | 0.320 (0.010) | 0.007 (0.001) | 0.715 (0.044) | 1.881 (0.071) |
| **Macrophage** | | | | | | |
| SMC | 0.498 (0.048) | 0.660 (0.207) | 0.147 (0.196) | 0.086 (0.020) | 0.665 (0.261) | 4.565 (1.027) |
| TDS | 0.447 (0.079) | 0.825 (0.027) | 0.080 (0.100) | 0.116 (0.031) | 0.674 (0.174) | 4.338 (0.502) |
| SVDD | 0.897 (0.008) | 0.952 (0.004) | 0.393 (0.041) | 0.006 (0.003) | 0.858 (0.041) | 1.643 (0.028) |
| DFKC | 0.756 (0.022) | 0.858 (0.052) | 0.331 (0.064) | 0.010 (0.001) | 0.796 (0.041) | 2.747 (0.126) |
| DRAKES | 0.636 (0.011) | 0.859 (0.008) | 0.338 (0.058) | 0.011 (0.003) | 0.872 (0.046) | 1.977 (0.081) |
| D3 | 0.901 (0.023) | 0.982 (0.010) | 0.430 (0.017) | 0.015 (0.007) | 0.804 (0.040) | 1.502 (0.022) |
| Ledidi | 0.147 (0.154) | 0.079 (0.150) | 0.277 (0.053) | 0.054 (0.010) | 0.855 (0.041) [n=2/3] | 2.662 (0.039) |
| Ctrl-DNA | 0.591 (0.088) | 0.430 (0.050) | 0.168 (0.048) | 0.054 (0.007) | 0.846 (0.031) | 6.794 (0.984) |
| Greedy selection | 0.810 (0.009) | 0.909 (0.006) | 0.243 (0.030) | 0.008 (0.002) | 0.866 (0.017) | 2.151 (0.026) |
| PACS | 0.867 (0.002) | 0.933 (0.002) | 0.309 (0.059) | 0.005 (0.002) | 0.850 (0.043) | 1.843 (0.059) |
| **Mast** | | | | | | |
| SMC | 0.474 (0.082) | 0.800 (0.080) | 0.136 (0.084) | 0.117 (0.062) | 0.617 (0.299) | 4.053 (0.382) |
| TDS | 0.540 (0.080) | 0.856 (0.000) | 0.080 (0.051) | 0.146 (0.056) | 0.696 (0.197) | 4.170 (0.219) |
| SVDD | 0.705 (0.020) | 0.697 (0.021) | 0.351 (0.054) | 0.005 (0.001) | 0.874 (0.017) | 2.548 (0.147) |
| DFKC | 0.762 (0.048) | 0.913 (0.013) | 0.303 (0.070) | 0.028 (0.008) | 0.818 (0.047) | 2.653 (0.182) |
| DRAKES | 0.410 (0.020) | 0.744 (0.009) | 0.281 (0.083) | 0.019 (0.003) | 0.920 (0.008) | 2.169 (0.104) |
| D3 | 0.882 (0.025) | 0.977 (0.006) | 0.360 (0.013) | 0.012 (0.004) | 0.885 (0.036) | 1.623 (0.075) |
| Ledidi | 0.113 (0.182) | 0.085 (0.150) | 0.256 (0.085) | 0.073 (0.028) | 0.656 (0.362) [n=2/3] | 2.601 (0.035) |

Continued on next page

Table 16 (continued)

| Method | JASPAR motif correlation | 3-mer correlation | TFBS pair co-occurrence | Inter-motif spacing JSD | Motif orientation correlation | NTv2 FBD |
|---|---|---|---|---|---|---|
| Ctrl-DNA | 0.444 (0.070) | 0.395 (0.123) | 0.059 (0.059) | 0.019 (0.019) | 0.720 (0.099) | 5.597 (1.325) |
| Greedy selection | 0.873 (0.014) | 0.922 (0.008) | 0.344 (0.021) | 0.008 (0.001) | 0.800 (0.068) | 1.873 (0.085) |
| PACS | 0.877 (0.019) | 0.944 (0.008) | 0.296 (0.017) | 0.009 (0.005) | 0.860 (0.025) | 1.656 (0.014) |

Table 17: Sequence composition and diversity for the eight immune cell types. Values are mean (SD) across three runs; valid-run annotations are retained where fewer runs are available.

| Method | Levenshtein distance | Positional Shannon entropy | GC content |
|---|---|---|---|
| **CD4$^+$ T** | | | |
| SMC | 98.963 (14.070) | 1.227 (0.160) | 0.446 (0.053) |
| TDS | 106.667 (24.137) | 1.339 (0.285) | 0.448 (0.035) |
| SVDD | 146.423 (0.435) | 1.931 (0.006) | 0.370 (0.006) |
| DFKC | 146.946 (0.096) | 1.938 (0.002) | 0.456 (0.008) |
| DRAKES | 149.319 (0.128) | 1.973 (0.002) | 0.442 (0.006) |
| D3 | 149.524 (0.219) | 1.976 (0.003) | 0.452 (0.007) |
| Ledidi | 149.974 (0.083) | 1.983 (0.001) | 0.500 (0.003) |
| Ctrl-DNA | 135.474 (1.451) | 1.779 (0.022) | 0.310 (0.009) |
| Greedy selection | 145.742 (0.200) | 1.921 (0.003) | 0.362 (0.001) |
| PACS | 147.821 (0.135) | 1.951 (0.002) | 0.397 (0.001) |
| **CD8$^+$ T** | | | |
| SMC | 90.003 (22.891) | 1.088 (0.282) | 0.472 (0.041) |
| TDS | 115.238 (5.277) | 1.400 (0.097) | 0.472 (0.052) |
| SVDD | 149.237 (0.124) | 1.972 (0.002) | 0.551 (0.007) |
| DFKC | 146.555 (0.301) | 1.931 (0.005) | 0.509 (0.009) |
| DRAKES | 148.232 (0.104) | 1.957 (0.001) | 0.407 (0.003) |
| D3 | 149.995 (0.084) | 1.983 (0.001) | 0.496 (0.007) |
| Ledidi | 149.975 (0.082) | 1.983 (0.001) | 0.500 (0.003) |
| Ctrl-DNA | 127.088 (1.148) | 1.684 (0.018) | 0.306 (0.004) |
| Greedy selection | 146.541 (0.706) | 1.934 (0.009) | 0.386 (0.008) |
| PACS | 149.219 (0.367) | 1.972 (0.005) | 0.446 (0.014) |
| **Naive T** | | | |
| SMC | 114.816 (6.758) | 1.413 (0.103) | 0.434 (0.033) |
| TDS | 112.122 (14.523) | 1.361 (0.230) | 0.479 (0.014) |
| SVDD | 149.684 (0.030) | 1.978 (0.000) | 0.534 (0.005) |
| DFKC | 147.160 (0.086) | 1.941 (0.002) | 0.457 (0.010) |
| DRAKES | 149.493 (0.079) | 1.976 (0.001) | 0.541 (0.005) |
| D3 | 150.000 (0.057) | 1.983 (0.001) | 0.508 (0.007) |
| Ledidi | 149.976 (0.084) | 1.983 (0.001) | 0.500 (0.003) |
| Ctrl-DNA | 144.725 (2.499) | 1.906 (0.040) | 0.579 (0.083) |
| Greedy selection | 149.779 (0.114) | 1.980 (0.002) | 0.526 (0.002) |
| PACS | 149.851 (0.078) | 1.981 (0.001) | 0.517 (0.002) |
| **NKT** | | | |
| SMC | 116.154 (3.006) | 1.431 (0.033) | 0.474 (0.031) |
| TDS | 113.291 (6.863) | 1.401 (0.096) | 0.455 (0.052) |
| SVDD | 149.732 (0.046) | 1.979 (0.001) | 0.485 (0.009) |
| DFKC | 147.074 (0.161) | 1.940 (0.003) | 0.458 (0.004) |
| DRAKES | 149.215 (0.068) | 1.972 (0.001) | 0.440 (0.003) |
| D3 | 149.878 (0.094) | 1.981 (0.001) | 0.478 (0.007) |
| Ledidi | 149.974 (0.080) | 1.983 (0.001) | 0.500 (0.003) |

Continued on next page

Table 17 (continued)

| Method | Levenshtein distance | Positional Shannon entropy | GC content |
|---|---|---|---|
| Ctrl-DNA | 142.302 (1.610) | 1.871 (0.023) | 0.414 (0.008) |
| Greedy selection | 148.629 (0.177) | 1.963 (0.002) | 0.532 (0.003) |
| PACS | 149.812 (0.133) | 1.980 (0.002) | 0.508 (0.005) |
| **Memory B** | | | |
| SMC | 98.202 (20.591) | 1.236 (0.264) | 0.461 (0.025) |
| TDS | 108.126 (12.194) | 1.330 (0.187) | 0.494 (0.046) |
| SVDD | 148.837 (0.267) | 1.966 (0.004) | 0.427 (0.010) |
| DFKC | 147.133 (0.274) | 1.941 (0.004) | 0.450 (0.010) |
| DRAKES | 149.315 (0.100) | 1.973 (0.001) | 0.443 (0.005) |
| D3 | 149.811 (0.140) | 1.980 (0.002) | 0.473 (0.008) |
| Ledidi | 149.974 (0.087) | 1.983 (0.001) | 0.500 (0.003) |
| Ctrl-DNA | 147.614 (0.492) | 1.950 (0.008) | 0.415 (0.023) |
| Greedy selection | 149.883 (0.080) | 1.981 (0.001) | 0.502 (0.005) |
| PACS | 149.894 (0.055) | 1.981 (0.001) | 0.482 (0.004) |
| **Plasma B** | | | |
| SMC | 105.030 (8.107) | 1.310 (0.098) | 0.478 (0.029) |
| TDS | 91.517 (22.957) | 1.138 (0.268) | 0.445 (0.022) |
| SVDD | 149.329 (0.110) | 1.973 (0.002) | 0.554 (0.006) |
| DFKC | 146.859 (0.125) | 1.937 (0.002) | 0.484 (0.008) |
| DRAKES | 148.771 (0.111) | 1.965 (0.001) | 0.424 (0.003) |
| D3 | 149.855 (0.155) | 1.981 (0.002) | 0.478 (0.009) |
| Ledidi | 149.975 (0.082) | 1.983 (0.001) | 0.500 (0.003) |
| Ctrl-DNA | 136.296 (5.910) | 1.806 (0.071) | 0.351 (0.013) |
| Greedy selection | 149.016 (0.301) | 1.969 (0.004) | 0.451 (0.004) |
| PACS | 149.603 (0.126) | 1.977 (0.002) | 0.459 (0.000) |
| **Macrophage** | | | |
| SMC | 103.650 (10.851) | 1.275 (0.125) | 0.490 (0.087) |
| TDS | 115.140 (5.446) | 1.429 (0.076) | 0.486 (0.023) |
| SVDD | 149.963 (0.116) | 1.982 (0.002) | 0.493 (0.002) |
| DFKC | 146.340 (0.361) | 1.929 (0.005) | 0.515 (0.019) |
| DRAKES | 148.677 (0.164) | 1.964 (0.002) | 0.425 (0.002) |
| D3 | 149.874 (0.152) | 1.981 (0.002) | 0.477 (0.007) |
| Ledidi | 149.979 (0.087) | 1.983 (0.001) | 0.500 (0.003) |
| Ctrl-DNA | 133.094 (0.595) | 1.738 (0.005) | 0.502 (0.017) |
| Greedy selection | 149.401 (0.057) | 1.974 (0.001) | 0.448 (0.003) |
| PACS | 149.626 (0.117) | 1.978 (0.002) | 0.471 (0.004) |
| **Mast** | | | |
| SMC | 117.627 (2.522) | 1.482 (0.012) | 0.467 (0.045) |
| TDS | 113.193 (12.024) | 1.397 (0.179) | 0.487 (0.031) |
| SVDD | 148.397 (0.263) | 1.960 (0.004) | 0.586 (0.008) |
| DFKC | 146.292 (0.319) | 1.928 (0.005) | 0.482 (0.012) |
| DRAKES | 148.183 (0.185) | 1.957 (0.002) | 0.405 (0.004) |
| D3 | 149.944 (0.125) | 1.982 (0.002) | 0.482 (0.008) |
| Ledidi | 149.977 (0.079) | 1.983 (0.001) | 0.500 (0.003) |
| Ctrl-DNA | 139.741 (5.926) | 1.835 (0.086) | 0.443 (0.038) |
| Greedy selection | 149.869 (0.009) | 1.981 (0.000) | 0.493 (0.003) |
| PACS | 149.851 (0.076) | 1.981 (0.001) | 0.474 (0.003) |