

Filtering Algo Description

October 2025

Overview: we select padlock probes in two rounds: (i) a *global offender filter* that removes the worst $\approx 1\%$ probes by heterodimer interaction load, and (ii) a *tail-targeted scoring* of three criteria followed by top-10 per-gene selection.

Inputs:

- Probes indexed by $i = 1, \dots, N$ with gene label $g(i)$.
- Heterodimer fraction matrix $P \in [0, 1]^{N \times N}$ (diagonal 0).
- Three per-probe raw features x_{ik} with their “bad” tail:

$$k \in \{\text{armgap}, \text{binding}, \text{offtg}\},$$

where

- **armgap** = arm Tm difference (higher worse),
- **binding** = on-target binding fraction (lower worse),
- **offtg** = off-target Tm sum (higher worse).

Round 1: Global offender filter (remove 1%) Compute each probe’s total heterodimer fraction

$$s_i = \sum_{j=1}^N P_{ij}.$$

Iteratively remove the probe $j^* = \arg \max_{i \text{ active}} s_i$ and update $s_\ell \leftarrow s_\ell - P_{\ell j^*}$ for all remaining ℓ . Stop after removing $\lfloor 0.01 N \rfloor$ probes. Let $\mathcal{I} \subset \{1, \dots, N\}$ denote the remaining index set.

Round 2: Tail-targeted scoring on the survivors \mathcal{I} The goal is to build a single selection score that treats features with very different scales and distribution shapes and aggressively penalizes probes that are extreme in any feature’s undesirable direction.

For each metric k and each $i \in \mathcal{I}$:

1. **Percentile (rank) mapping.** Let $u_{ik} \in (0, 1)$ be the percentile (slightly shifted to avoid 0 and 1) of x_{ik} within $\{x_{jk} : j \in \mathcal{I}\}$:

$$u_{ik} = \frac{\text{rank}(x_{ik}) - 0.5}{|\mathcal{I}|}.$$

2. **Tail transform (symmetric score).** The idea here is to treat u_{ik} as the CDF of a zero centered, symmetric distribution (For normal distribution, this transformation is called “rank-based inverse normal transformation”). Define the tail score

$$s_{ik} = \begin{cases} T(u_{ik}), & \text{if upper tail is worse (armgap, offtg),} \\ -T(u_{ik}), & \text{if lower tail is worse (binding).} \end{cases}$$

With $T(u) = \Phi^{-1}(u)$ for inverse-normal or $T(u) = \log \frac{u}{1-u}$ for logit. We chose logit to emphasize on the very end of the rank. In practice, we also clip very big s_{ik} to a fixed value (current threshold set to 8).

3. **Per-metric exponential term.**

$$r_{ik} = w_k \exp(\beta_k s_{ik}).$$

4. **Total score (larger = worse).**

$$S_i = \sum_k r_{ik}.$$

Selection: For each gene g , with index set $\mathcal{I}_g = \{i \in \mathcal{I} : g(i) = g\}$:

- If $|\mathcal{I}_g| \leq 10$, keep all $i \in \mathcal{I}_g$.
- If $|\mathcal{I}_g| > 10$, sort \mathcal{I}_g by S_i ascending (smallest is best) and keep the first 10:

$$\mathcal{K}_g = \{i \in \mathcal{I}_g : \text{rank}_{\mathcal{I}_g}(S_i) \leq 10\}.$$

Outputs:

- Per-probe scores S_i , per-gene ranks, and keep sets \mathcal{K}_g (others dropped).
- QC: global ranks/percentiles of S_i , ranks of features.