GeneBench-Pro Case Study: SV-masked Hi-C Loop-strength Estimation

GeneBench-Pro

June 26, 2026

1 Overview

This case study asks an analyst to quantify a focal chromatin loop gain at one 20 kb Hi-C interaction while avoiding a series of plausible shortcuts. In Hi-C, each matrix pixel is a contact count between two genomic bins, and a loop is a focal excess above a smooth background driven mainly by genomic distance and technical biases. The released files include two contact tables, two bin-annotation tables, and a minimal prompt naming the target pair and the required output schema. The reported outputs are the replicate-mean target-pair $\log_2$(observed/expected) values in case and control, plus their difference. The target is therefore not a raw count contrast, not a local ring-of-neighbors background score, and not a same-distance average. It is a loop-strength contrast after reconstructing the appropriate expected-count background from the released data [1–3].

2 Released Prompt and Files

Prompt

```
You are given Hi-C contact matrices at 20 kb and 40 kb resolution plus bin annotations. Estimate
    the loop enrichment at the 20 kb interaction between ‘bin_id = 8‘ and ‘bin_id = 17‘ in ‘
    bins_20kb.tsv.gz‘. Report three quantities: ‘case_loop_strength‘ (mean log2(observed/expected)
    across case replicates), ‘control_loop_strength‘ (mean log2(observed/expected) across control
    replicates), and ‘delta_loop_strength‘ (case minus control).
These data came from a real experiment; you will be graded not just on numerical correctness but
    the quality of analytical reasoning you exhibit; do not attempt to take any shortcuts.
Return your final answer as exactly one JSON object.
Do not wrap the JSON in markdown.
Do not add prose before or after the JSON.
Do not omit any keys shown in the example.
Return the JSON object in your final answer:
{
  "answer": {
    "case_loop_strength": <float>,
    "control_loop_strength": <float>,
    "delta_loop_strength": <float>
  },
  "reasoning": "<description of method and QC>"
}
```

Released data files

File	Format	Contents
`bins_20kb.tsv.gz`	.tsv.gz	Twenty-four 20 kb chr8 bins with `bin_id`, coordinates, GC content, mappability, and restriction-site count.
`contacts_20kb.tsv.gz`	.tsv.gz	Replicate-level upper-triangle 20 kb contact counts with `rep`, `condition`, `bin_i`, `bin_j`, `dist_bin`, and `count`.
`bins_40kb.tsv.gz`	.tsv.gz	Twelve coarsened 40 kb chr8 bins with the same annotation fields, used as context for resolution checks rather than the requested target surface.
`contacts_40kb.tsv.gz`	.tsv.gz	Replicate-level 40 kb contact counts aggregated from the 20 kb surface, included to make the wrong-resolution shortcut visible.

3 Answer Fields and Tolerances

The GLM parameters themselves are not reported. They are used to compute fitted expected counts $\widehat{\mu}_{r,i^\star j^\star}$, and the released JSON fields map to those fitted quantities as

$$\texttt{case\_loop\_strength}\ (\widehat{L}_{\text{case}}) = \frac{1}{2} \sum_{r \in R_{\text{case}}} \log_2 \left(\frac{Y_{r,i^\star j^\star}}{\widehat{\mu}_{r,i^\star j^\star}} \right) \tag{1}$$

$$\texttt{control\_loop\_strength}\ (\widehat{L}_{\text{control}}) = \frac{1}{2} \sum_{r \in R_{\text{control}}} \log_2 \left(\frac{Y_{r,i^\star j^\star}}{\widehat{\mu}_{r,i^\star j^\star}} \right) \tag{2}$$

$$\texttt{delta\_loop\_strength}\ (\widehat{\Delta}) = \widehat{L}_{\text{case}} - \widehat{L}_{\text{control}}. \tag{3}$$

Answer field	Ground truth	Tolerance / matching rule	Interpretation
`case_loop_strength`	1.880794	Absolute error ≤ 0.020	Mean case-replicate $\log_2$(observed/expected) at the target 20 kb interaction.
`control_loop_strength`	-0.518554	Absolute error ≤ 0.020	Mean control-replicate $\log_2$(observed/expected) at the same target interaction.
`delta_loop_strength`	2.399347	Absolute error ≤ 0.020	Case minus control loop-strength contrast.

4 Structure Diagram

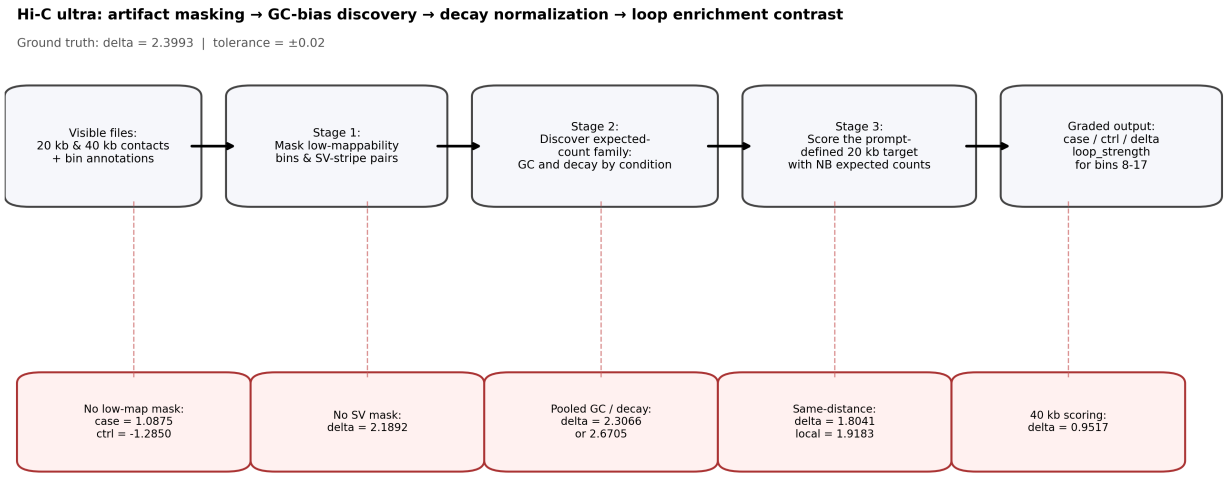

Figure 1: Three-stage cascade from the released contact tables to the loop-strength outputs. The gray boxes connected by black arrows are the intended analysis path: start from the visible 20 kb and 40 kb contact/bin files, mask low-mappability and structural-variant (SV) stripe pairs, learn a condition-specific GC and distance background, and score the prompt-defined 20 kb target with negative-binomial (NB) expected counts. The red outlined boxes below the path are incorrect shortcuts, and the red dashed connectors show which stage each shortcut skips or distorts. "ctrl" means control.

5 Variables and Assumptions

- $i, j \in \{0, \ldots, 23\}$: 20 kb bin indices with $i < j$.
- $I, J \in \{0, \ldots, 11\}$: 40 kb parent-bin indices with $I < J$.
- $B_I = \{2I, 2I + 1\}$: the two 20 kb children aggregated into 40 kb parent bin I.
- $r \in \{\texttt{ctrl\_r1}, \texttt{ctrl\_r2}, \texttt{case\_r1}, \texttt{case\_r2}\}$: replicate label.
- $C_r \in \{\text{control}, \text{case}\}$: replicate condition.
- $g_i \in [0, 1]$: 20 kb bin GC content, drawn blockwise from uniform distributions and then constructively overridden at bins 8 and 17.
- $m_i \in [0, 1]$: 20 kb bin mappability, drawn from Uniform$(0.86, 0.98)$ except for the low bins 10 and 11.
- $s_i \in \mathbb{N}$: 20 kb bin restriction-enzyme site count, drawn from DiscreteUniform$\{3, \ldots, 8\}$.
- $d_{ij} = j - i$: genomic distance in 20 kb bins.
- $\eta^{\text{bg}}_{r,ij} \in \mathbb{R}$ and $\mu^{\text{bg}}_{r,ij} = e^{\eta^{\text{bg}}_{r,ij}}$: smooth background log-mean and mean before low-mappability inflation, SV-stripe contamination, and focal loop gain.

- $\eta^{\text{obs}}_{r,ij} \in \mathbb{R}$ and $\mu^{\text{obs}}_{r,ij} = e^{\eta^{\text{obs}}_{r,ij}}$: sampling log-mean and mean after deterministic artifacts and the case focal-loop gain are added.
- $Y_{r,ij} \in \mathbb{N}_0$: observed 20 kb Hi-C contact count, sampled from a negative-binomial law conditional on $\mu^{\text{obs}}_{r,ij}$.
- $g_I^{(40)}, m_I^{(40)}, s_I^{(40)}$: coarsened 40 kb covariates obtained from the two children in B_I.
- $Y^{(40)}_{r,IJ}$: observed 40 kb contact count obtained by summing the four 20 kb child counts in $B_I \times B_J$.
- $\mathcal{L} = \{10, 11\}$: low-mappability bins.
- $\mathcal{S} = \{(i, j) : i \in 9{:}12, j \in 15{:}18\}$: SV-stripe bin-pair set.
- $(i^\star, j^\star) = (8, 17)$: target pair.

6 Data-Generating Process

6.1 Bin-level covariates

The simulation begins by creating 24 adjacent 20 kb bins on chromosome 8, spanning base positions 400,000 through 880,000. GC content is sampled blockwise and then the target bins are pushed high so the GC-interaction decision point has leverage:

$$g_i \sim \begin{cases} \text{Uniform}(0.35, 0.52), & i \in \{0, \dots, 7\} \\ \text{Uniform}(0.42, 0.62), & i \in \{8, \dots, 15\} \\ \text{Uniform}(0.38, 0.58), & i \in \{16, \dots, 23\}, \end{cases} \qquad g_8 = 0.58,\ g_{17} = 0.57 \tag{4}$$

$$m_i \sim \text{Uniform}(0.86, 0.98), \qquad m_{10} = 0.42,\ m_{11} = 0.46 \tag{5}$$

$$s_i \sim \text{DiscreteUniform}\{3, \dots, 8\}. \tag{6}$$

The first equation creates a realistic GC spectrum with a deliberately GC-rich target pair. The second creates a clean low-mappability tail at bins 10 and 11. The third gives each bin a small integer restriction-site count for use in the expected-count background.

6.2 Expected-count mean model

For each replicate and bin pair, the smooth background contact-count mean is generated on the log scale. The correct expected-count family has replicate intercepts, condition-specific GC effects, a restriction-site term, and condition-specific distance decay. In this problem, mappability is used to define a mask rather than as a continuous regression covariate. The background mean model therefore mirrors the standard Hi-C bias-modeling ingredients of genomic distance, sequence composition, and restriction-fragment structure, while handling low mappability through Stage 1 filtering [1,3].

$$\eta^{\text{bg}}_{r,ij} = \beta_{0,r} - \lambda_{C_r} \log(d_{ij} + 1) + \beta_{g,C_r}(g_i + g_j - 1) + \beta_{re}\left[\log(s_i s_j) - \log(25)\right] \tag{7}$$

with

$$\lambda_{\text{control}} = 1.90, \quad \lambda_{\text{case}} = 1.55, \quad \beta_{g,\text{control}} = 1.30, \quad \beta_{g,\text{case}} = 0.50, \quad \beta_{re} = 0.08,$$

and replicate intercepts

$$\beta_{0,\texttt{ctrl\_r1}} = 6.25,\ \beta_{0,\texttt{ctrl\_r2}} = 6.05,\ \beta_{0,\texttt{case\_r1}} = 6.20,\ \beta_{0,\texttt{case\_r2}} = 6.10.$$

The λ values make the control matrix decay more steeply with distance than the case matrix. The β_g values create the condition-specific GC bias, and the replicate intercepts produce realistic between-library baseline offsets without changing the target estimand.

6.3 Constructive artifacts

The construction then copies the background log-mean into η^{obs} and adds three independent log-mean terms before sampling counts:

$$\begin{aligned}\eta^{\text{obs}}_{r,ij} = \eta^{\text{bg}}_{r,ij} &+ 1.20\,\mathbf{1}\{i \in \mathcal{L} \text{ or } j \in \mathcal{L}\} \\ &+ 0.75\,\mathbf{1}\{C_r = \text{case and } (i,j) \in \mathcal{S}\} \\ &+ 1.00\,\mathbf{1}\{C_r = \text{case and } (i,j) = (8,17)\}.\end{aligned} \tag{8}$$

These terms are cumulative: a case pixel in the SV stripe that also touches a low-mappability bin receives both the 1.20 and 0.75 increments. The first term creates the low-map artifact, the second creates the SV stripe, and the third encodes the true case-only loop gain.

6.4 Observed counts

Observed counts follow a negative-binomial sampling model with variance $\mu + \alpha\mu^2$ rather than a strictly Poisson variance law of μ [7]:

$$Y_{r,ij} \sim \text{NegBin}\Big(\mu^{\text{obs}}_{r,ij} = e^{\eta^{\text{obs}}_{r,ij}},\ \text{size} = 18\Big). \tag{9}$$

This yields overdispersed counts consistent with the negative-binomial generalized linear model used later for expected-count fitting [1,7].

6.5 40 kb coarsening

The released 40 kb context files are deterministic aggregations of the 20 kb surface:

$$g^{(40)}_I = \frac{1}{2}\sum_{i \in B_I} g_i, \qquad m^{(40)}_I = \frac{1}{2}\sum_{i \in B_I} m_i, \qquad s^{(40)}_I = \sum_{i \in B_I} s_i \tag{10}$$

$$Y^{(40)}_{r,IJ} = \sum_{i \in B_I}\sum_{j \in B_J} Y_{r,ij}, \qquad d^{(40)}_{IJ} = J - I. \tag{11}$$

The coarsened covariates average GC and mappability across the two child bins while summing restriction sites. The coarsened count surface sums the four child pixels inside each parent block, which is why the resolution-choice step must treat the 40 kb files as context rather than an interchangeable quantification surface.

7 Analyst Walkthrough

The analysis is a sequence of background-identification decisions rather than a single contrast at the target pixel. A correct solution must keep the 20 kb target surface, mask low-mappability bins, remove the case-only SV stripe from the expected-count fit, allow GC and distance-decay terms to differ by condition, and reject local-window backgrounds that leak the focal structural signal into the null model.

Stage 1, Step 1: Start from the target pair, but do not trust raw counts

The first quick check is the raw target contrast:

```
target = contacts20.query("bin_i == 8 and bin_j == 17")
raw = target.groupby("condition")["count"].mean()
delta_raw = np.log2(raw["case"]) - np.log2(raw["control"])
```

A careful analyst naturally starts here: the target pair $(8, 17)$ is visibly stronger in case than control, and a raw log-count contrast gives $\Delta = 3.4160$. That answer initially looks defensible because the case matrix really does contain a bright focal signal at the target. It nevertheless targets the wrong quantity because it treats the target count as self-normalizing. The released contact tables plainly show strong distance decay and condition-level baseline differences, so raw counts cannot be the final estimand [2,3].

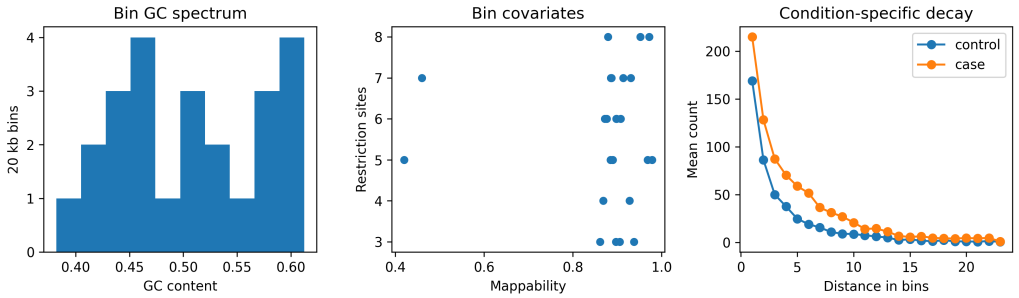

Figure 2: Overview of the released 20 kb data surface. Left: a histogram of bin GC content, where each bar counts the number of 20 kb bins in a GC-content interval. Middle: each blue point is one bin, plotted by mappability on the horizontal axis and restriction-enzyme site count on the vertical axis. Right: mean contact count by genomic distance bin, with blue circles/line for control replicates and orange circles/line for case replicates; the separated decay curves show why the raw target count is not self-normalizing.

Together, these panels show why the first decision cannot be a raw-count contrast: the target sits in a data surface where GC content, restriction-site count, mappability, and genomic distance all affect the expected background, so the later masking and model-selection steps are practical requirements rather than optional diagnostics.

Stage 1, Step 2: Mask low-mappability bins

If low-map pairs are left in the training set, the expected-count background is badly distorted and the target values move to $(1.0875, -1.2850, 2.3725)$ for (case, control, delta). That answer can look deceptively close on Δ alone, so the three released output fields distinguish a numerically nearby

contrast from the intended background model. The diagnostic is a clean low-mappability gap: bins 10 and 11 sit at 0.42 and 0.46, while every other bin is above 0.86. That structural gap means any cutoff between roughly 0.6 and 0.8 isolates exactly the same two bins; the implementation uses $m_i < 0.6$, and the figure's dashed line at pairwise mappability sum 1.45 marks the corresponding pair-scale separation. Pairs touching those bins have mean count 118.58 versus 31.10 for the rest. The corrective action is simple: drop those pairs before any background fit so Stage 1 hands a clean non-target matrix to the next decision. This diagnostic and mask are justified by the long-standing role of sequence uniqueness/mappability in Hi-C bias correction [3].

```
low_bins = bins20.query("mappability < 0.6")["bin_id"]
is_low_pair = contacts20.bin_i.isin(low_bins) | contacts20.bin_j.isin(low_bins)
contacts20.groupby(is_low_pair)["count"].mean()
```

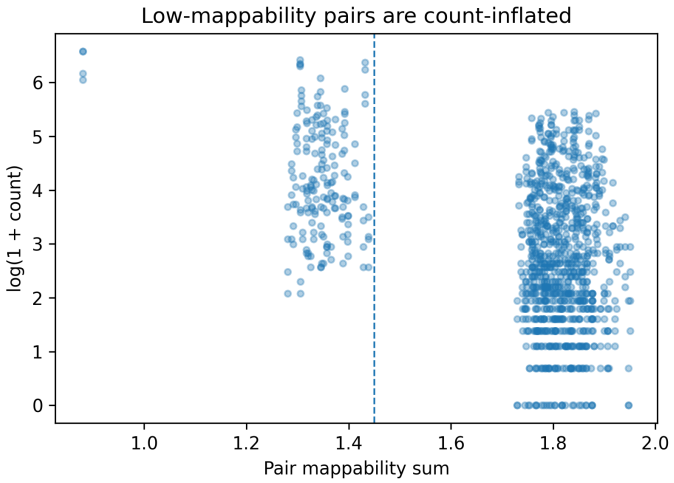

Figure 3: Low-mappability diagnostic. Each blue point is one replicate-specific contact pair, with pair mappability sum $m_i + m_j$ on the horizontal axis and $\log(1 + \text{count})$ on the vertical axis. The blue dashed vertical line at pair mappability sum 1.45 marks the pair-scale separation induced by the bin-level mask $m_i < 0.6$: pairs left of the line touch one of the two low-mappability bins (10 or 11), while pairs to the right use only high-mappability bins. The left cluster is visibly count-inflated and should not train the expected-count background.

Stage 1, Step 3: Mask the case-only SV stripe

If stripe pairs are kept in the background fit, the target delta falls to 2.1892. That answer still points in the right qualitative direction, which is why the shortcut is tempting. After the low-mappability bins have been removed, the stripe diagnostic remains condition-specific: stripe pairs in the case matrix average 70.38 counts versus 37.14 elsewhere, while control stripe pairs average only 19.31 versus 24.07 elsewhere. The region could initially look like part of a broad case-specific domain, but the condition asymmetry is too sharp for ordinary local background. The corrective action is to exclude those stripe-pair locations from the training surface in both conditions before moving on to GC and decay discovery, so the expected-count fit uses matched non-target geometry rather than letting the case-only anomaly leak into one condition's fit. Structural variants can reorganize 3D contacts and create enhancer-hijacking events in rearranged genomes [4]; here, the released count surface makes that general concern visible as a case-only stripe that should not train the background.

```
stripe = train.query("9 <= bin_i <= 12 and 15 <= bin_j <= 18")
stripe.groupby("condition")["count"].mean()
train = train.query("not (9 <= bin_i <= 12 and 15 <= bin_j <= 18)")
```

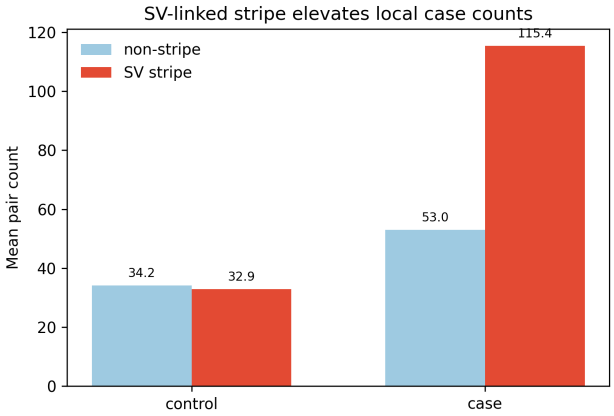

Figure 4: Structural-variant (SV) stripe diagnostic. The horizontal groups are control and case conditions; the vertical axis is mean contact-pair count. Light-blue bars summarize non-stripe pairs, red bars summarize SV-stripe pairs, and the numbers printed above bars are the corresponding mean counts. The red case bar is much higher than the light-blue case bar, while the two control bars are similar, showing that the stripe is a case-only background contaminant rather than the focal target loop itself.

Stage 2, Step 4: Fit a diagnostic background model and discover the GC interaction

If you keep a pooled GC term because it feels simpler and still "mostly" right, the target delta is 2.3066 and both per-condition values fail tolerance. To isolate the GC decision from the distance-decay decision, fit a diagnostic model that already allows condition-specific distance slopes but still forces one pooled GC coefficient. The residual mismatch remains: Pearson residuals against *gc_sum* rise with slope about +0.687 in control but only about +0.149 in case. That mismatch means the GC effect is not shared across conditions, so the background model needs separate GC slopes by condition. GC-sensitive background modeling is standard in Hi-C count normalization [1,3,5], while the condition interaction is justified here by the released residual pattern rather than by a prespecified recipe.

```
fit_gc = smf.glm("count ~ 0 + C(rep) + gc_sum + re_log + log_dist:C(condition)",
                 data=train, family=nb_family).fit()
train["pearson"] = fit_gc.resid_pearson
train.groupby("condition").apply(lambda d: np.polyfit(d.gc_sum, d.pearson, 1)[0])
```

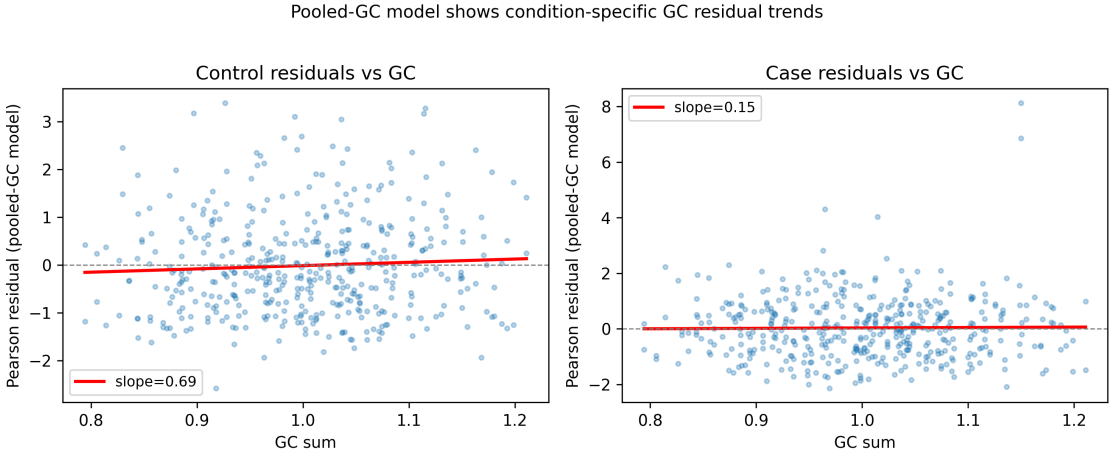

Figure 5: Condition-stratified pooled-GC residual diagnostic. The left panel is control and the right panel is case. Each blue point is a masked training contact pair, plotted by GC sum $(g_i + g_j - 1)$ on the horizontal axis and Pearson residual from a model with one pooled GC coefficient on the vertical axis. The gray dashed horizontal line is residual zero. The red line in each panel is the fitted residual trend, with its slope printed in the legend. The larger positive control slope compared with the near-flat case slope is the stage-2 signal that the GC term must interact with condition rather than be pooled.

Stage 2, Step 5: Fit condition-specific distance decay

If you pool distance decay because both curves are monotone and look qualitatively similar, the fitted target delta jumps to 2.6705. The released-data diagnostic is the separation between the two decay curves: a simple log-log slope summary over all released pairs gives -2.305 in control versus -1.923 in case, and the same separation persists after the Stage 1 masks with slopes -2.090 versus -1.658. The target sits at distance bin 9, where the case curve remains systematically above the control curve; a pooled slope therefore predicts the wrong expected baseline for both conditions at exactly the queried separation. The corrective action is to split the log-distance slope by condition; once you do that, the model family is finally right and the remaining question is where to score the target interaction. Distance-aware expected modeling and contact-frequency-versus-distance summaries are standard in Hi-C analysis [1–3,5], but the exact condition interaction here is again a released-data repair chosen because the pooled model leaves a visible condition mismatch.

```
formula = (
    "count ~ 0 + C(rep) + gc_sum:C(condition) + re_log "
    "+ log_dist:C(condition)"
)
fit_decay = smf.glm(formula, data=train, family=nb_family).fit()
fit_decay.params.filter(like="log_dist")
```

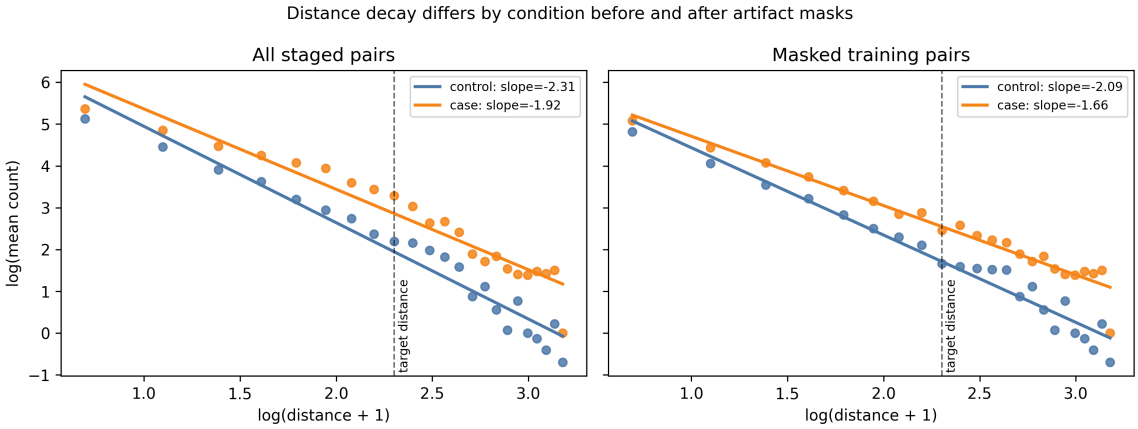

Figure 6: Condition-specific distance-decay diagnostic before and after Stage 1 masking. The left panel uses all staged contact pairs; the right panel uses only the masked training pairs after low-mappability, SV-stripe, and target-pixel exclusions. Blue points/line are control and orange points/line are case. Points are mean contact counts by distance bin plotted as log(mean count) versus $\log(\text{distance} + 1)$; solid lines are the fitted log-log decay trends, with slopes reported in the legends. The gray dashed vertical line marks the target distance for bins 8 and 17 ($d = 9$, so $\log(d + 1) = \log 10$). The condition-specific slope separation remains after masking, supporting $\log(d + 1) : C(\text{condition})$ rather than one pooled decay term.

Stage 3, Step 6: Reject same-distance and local-window backgrounds

Even after the bias-aware model family is identified, two simpler expected-count definitions remain tempting. A same-distance background compares the target only to other pairs at the same genomic separation, while a local-window background compares it to nearby pixels. Both are reasonable exploratory diagnostics, but neither estimates the counterfactual smooth background for the target pair: the same-distance average ignores GC, restriction-site, replicate, and condition-specific decay effects, and the local window is contaminated by the same focal structural signal the analysis is trying to quantify. In the released instance, these shortcuts return $\Delta = 1.8041$ and $\Delta = 1.9183$, respectively, outside the tolerance around the reference $\Delta = 2.3993$.

```
same = train.groupby(["condition", "dist_bin"])["count"].mean().reset_index()
target_dist = contacts20.query("bin_i == 8 and bin_j == 17").dist_bin.iloc[0]
same.query("dist_bin == @target_dist")
```

Stage 3, Step 7: Stay at 20 kb for quantification

If you repeat the otherwise correct background logic after aggregating to 40 kb, the resulting target delta is only 0.9517. That failure happens because the parent pixel mixes the focal loop with surrounding background, even though the coarse view still looks informative at first glance. The 40 kb data helps reveal broader structure but, in this case study, it is not the reported quantification surface. The corrective action is to bring the fully corrected background model back to the prompt-defined 20 kb surface before scoring the target pair. Multi-resolution Hi-C tooling is resolution-sensitive [2,5], and this ablation shows that the coarse files are useful for context but not for the final target quantification requested in the prompt.

```
fit_40 = fit_expected_model(contacts40, bins40, resolution="40kb")
score_target(fit_40, pair=(4, 8), coarse=True)
```

Stage 3, Step 8: Compute the requested outputs

With the correct 20 kb masked training set and the correct expected-count model, predict $\widehat{\mu}_{r,ij}$ for all pairs, then compute

$$\widehat{L}_C = \frac{1}{|R_C|} \sum_{r \in R_C} \log_2 \left(\frac{Y_{r,i^\star j^\star}}{\widehat{\mu}_{r,i^\star j^\star}} \right), \qquad \widehat{\Delta} = \widehat{L}_{\text{case}} - \widehat{L}_{\text{control}}. \tag{12}$$

On the released data this yields

$$\widehat{L}_{\text{case}} = 1.8808, \qquad \widehat{L}_{\text{control}} = -0.5186, \qquad \widehat{\Delta} = 2.3993.$$

```
target = contacts20.query("bin_i == 8 and bin_j == 17").copy()
target["expected"] = fit.predict(target)
target["loe"] = np.log2(target["count"] / target["expected"])
target.groupby("condition")["loe"].mean()
```

The replicate-level target-pair pieces are $\log_2(O/E) = -0.2300$ for `ctrl_r1`, -0.8071 for `ctrl_r2`, 1.7809 for `case_r1`, and 1.9807 for `case_r2`. Averaging the two case values gives 1.8808, averaging the two control values gives -0.5186, and subtracting those means gives $\Delta = 2.3993$, matching the answer-field table.

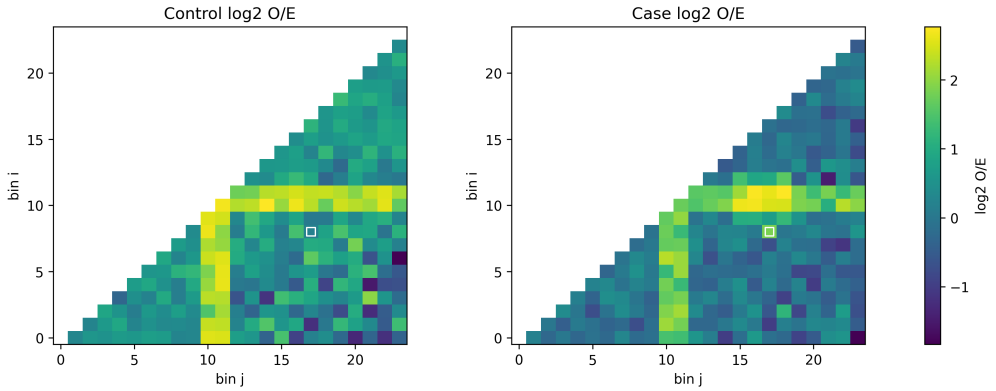

Figure 7: Correct-model 20 kb replicate-mean heatmaps. The left panel is control and the right panel is case; axes are 20 kb bin indices i and j. Color encodes $\log_2$ observed/expected (O/E) loop enrichment using $\log_2((\mathrm{O}+0.5)/(\mathrm{E}+0.5))$ only for display stability across zero-count non-target pixels: darker purple/blue is lower than expected, green is near zero enrichment, and yellow is higher than expected. The shared color bar gives the displayed $\log_2$ O/E scale. The white open square marks the prompt-defined target pixel $(i, j) = (8, 17)$ in each condition. The released target values use the exact no-pseudocount $\log_2$(O/E) values summarized in the answer-field table.

8 Estimand

The released estimand is the per-condition replicate-mean target-pair loop enrichment on the $\log_2$(O/E) scale, where E is the counterfactual smooth background mean rather than the post-

loop sampling mean:

$$L_{\text{case}} = \mathbb{E}_{r \in R_{\text{case}}} \left[\log_2 \left(\frac{Y_{r,i^\star j^\star}}{\mu^{\text{bg}}_{r,i^\star j^\star}} \right) \right] \tag{13}$$

$$L_{\text{control}} = \mathbb{E}_{r \in R_{\text{control}}} \left[\log_2 \left(\frac{Y_{r,i^\star j^\star}}{\mu^{\text{bg}}_{r,i^\star j^\star}} \right) \right] \tag{14}$$

$$\Delta = L_{\text{case}} - L_{\text{control}}. \tag{15}$$

This is intentionally narrower than "is there a loop?" and narrower than "what is the case-control raw count ratio?" The naive answers fail because they either omit the expected-count background entirely or estimate it on the wrong training surface.

9 Estimator

The estimator has the same three formal stages named in the Overview and Analyst Walkthrough and matches the reference implementation used to generate the public answer.

Stage 1: artifact masking and training-set definition. Let

$$\mathcal{T} = \{(r,i,j) : (i,j) \neq (i^\star, j^\star),\ i \notin \mathcal{L},\ j \notin \mathcal{L},\ (i,j) \notin \mathcal{S}\}\,. \tag{16}$$

These are the non-target 20 kb pairs used to learn the expected-count background after the low-map and SV-stripe artifacts have been removed.

Stage 2: expected-count family discovery and fitting. The expected-count model is a negative-binomial generalized linear model (GLM), i.e. a count-regression model with a log-linked mean and negative-binomial variance, fit on $\mathcal{T}$ [1,6,7]. This is the formal version of the Stage 2 discovery problem: the correct family keeps separate GC and distance slopes by condition rather than pooling them. Under the statsmodels NB2 parameterization,

$$Y_{r,ij} \mid \mu_{r,ij} \sim \text{NegBin}(\mu_{r,ij}, \alpha), \qquad \text{Var}(Y_{r,ij} \mid \mu_{r,ij}) = \mu_{r,ij} + \alpha \mu^2_{r,ij}, \tag{17}$$

with fixed dispersion $\alpha = 1/18$. Let $z^g_{ij} = g_i + g_j - 1$, $z^{re}_{ij} = \log(s_i s_j) - \log(25)$, and $z^d_{ij} = \log(d_{ij}+1)$. The fitted parameter vector $\widehat{\theta}$ maximizes

$$\ell(\theta) = \sum_{(r,i,j) \in \mathcal{T}} \log f_{\text{NB}}(Y_{r,ij}; \mu_{r,ij}(\theta), \alpha)\,, \tag{18}$$

where the linear predictor is

$$\begin{aligned} \log \mu_{r,ij} = \alpha_r &+ \gamma_{g,\text{control}}\, z^g_{ij} \mathbf{1}\{C_r = \text{control}\} + \gamma_{g,\text{case}}\, z^g_{ij} \mathbf{1}\{C_r = \text{case}\} \\ &+ \gamma_{re}\, z^{re}_{ij} + \gamma_{d,\text{control}} z^d_{ij} \mathbf{1}\{C_r = \text{control}\} \\ &+ \gamma_{d,\text{case}} z^d_{ij} \mathbf{1}\{C_r = \text{case}\}. \end{aligned} \tag{19}$$

The implementation uses the `statsmodels` GLM fit with log link and fixed negative-binomial dispersion $\alpha = 1/18$ [6,7]. This fixed dispersion is a reference-estimator convention matching the

synthetic count construction; it is not a default asserted by the software documentation. The covariates are `gc_sum` $= g_i + g_j - 1$, `re_log` $= \log(s_i s_j) - \log(25)$, and `log_dist` $= \log(d_{ij} + 1)$. The target pixel is excluded from training, and no pseudocount is used for the released output values. In Patsy/statsmodels notation this is

```
count ~ 0 + C(rep) + gc_sum:C(condition) + re_log
        + log_dist:C(condition).
```

Dispersion robustness check. The target values are not tuned to the single fixed dispersion. Refitting the same masked model with dispersion sizes $9, 12, 27$, and 36, corresponding to $\alpha = 1/9, 1/12, 1/27$, and $1/36$, keeps every answer field within the ± 0.020 tolerance around the $\alpha = 1/18$ reference. Across dispersion sizes $9, 12, 18, 27$, and 36, the worst absolute field error is 0.0073.

Stage 3: 20 kb target scoring. After fitting $\widehat{\theta}$ on $\mathcal{T}$, predict $\widehat{\mu}_{r,ij}$ for all released 20 kb pairs, including the held-out target pair $(i^\star, j^\star)$, and then map those fitted expectations to the requested outputs. The 40 kb files are not part of this stage because the prompt-defined estimand is the 20 kb target pixel.

10 Decision-Point and Ablation Walkthrough

The table below combines the full ablation suite with the failure stage and mechanism. The released answer contract checks case, control, and delta loop strength; a row passes only if all three values are within tolerance.

Decision point	Analysis / ablation	Quantitative output	Pass?	Failure point	Why the approach is wrong
Reference pipeline	correct	delta 2.3993, case 1.8808, control -0.5186; errors 0.0000/0.0000/0.0000	yes	none	Reference masked, condition-specific, bias-aware expected-count GLM at 20 kb.
Reference sensitivity	dispersion_size_9_to_36	all tested sizes pass; worst field error 0.0073 at size 9	yes	none	Same masked NB GLM refit across a reasonable dispersion range.
GC modeling	pooled_gc	delta 2.3066, case 1.8394, control -0.4672; errors 0.0927/0.0414/0.0514	no	Stage 2	Uses one GC coefficient even though residual GC trends differ by condition.
Distance decay	pooled_decay	delta 2.6705, case 1.9818, control -0.6887; errors 0.2711/0.1010/0.1701	no	Stage 2	Pools condition-specific distance-decay curves.
GC and distance	pooled_gc_and_decay	delta 2.5978, case 1.9480, control -0.6498; errors 0.1985/0.0672/0.1312	no	Stage 2	Underfits both condition-dependent GC and distance structure.
Low-mappability mask	no_lowmap_filter	delta 2.3725, case 1.0875, control -1.2850; errors 0.0269/0.7933/0.7664	no	Stage 1	Lets duplicated or uncallable low-mappability bins train the background.
SV mask	no_sv_mask	delta 2.1892, case 1.6739, control -0.5154; errors 0.2101/0.2069/0.0032	no	Stage 1	Absorbs the case-only SV stripe into the expected background.
All masks	no_masks	delta 2.1236, case 0.7827, control -1.3409; errors 0.2758/1.0981/0.8223	no	Stage 1	Leaves both low-map inflation and SV-stripe contamination in the fit.
All masks and GC	no_masks_pooled_gc	delta 2.1298, case 0.7855, control -1.3443; errors 0.2696/1.0953/0.8257	no	Stages 1–2	Combines unmasked background contamination with a pooled GC term.
Shortcut expected count	same_distance_mean	delta 1.8041, case 1.8017, control -0.0023; errors 0.5953/0.0790/0.5162	no	Stage 3	Same-distance averaging ignores GC, RE-site, and local artifact structure.
Shortcut expected count	same_distance_pooled	delta 3.4160, case 2.3934, control -1.0226; errors 1.0166/0.5126/0.5040	no	Stage 3	Pooled same-distance background collapses condition structure.
Shortcut expected count	local_donut	delta 1.9183, case 1.4581, control -0.4603; errors 0.4810/0.4227/0.0583	no	Stage 3	Local window does not estimate the bias-aware expected background.
Shortcut expected count	same_distance_by_rep	delta 1.7976, case 1.8018, control 0.0041; errors 0.6017/0.0790/0.5227	no	Stage 3	Replicate-wise same-distance averaging still misses condition-specific covariate effects.
Resolution	coarse_40kb_model	delta 0.9517, case 1.0110, control 0.0593; errors 1.4477/0.8698/0.5778	no	Stage 3	Coarse bins blur the focal 20 kb loop into neighboring background.
Resolution	coarse_40kb_same_distance	delta 1.0059, case 1.2014, control 0.1955; errors 1.3934/0.6794/0.7140	no	Stage 3	Combines wrong resolution with a same-distance shortcut.
Raw-count shortcut	raw_count_log2	delta 3.4160, case 5.8197, control 2.4037; errors 1.0166/3.9389/2.9222	no	Stage 3	Treats observed counts as self-normalizing and ignores expected background.
Null/default shortcut	default_zero	delta 0.0000, case 0.0000, control 0.0000; errors 2.3993/1.8808/0.5186	no	Output contract	Does not perform the loop-strength analysis.

Table 1: Unified decision-point and ablation walkthrough for the Hi-C masked loop-strength problem.

11 References

1. Sahin M, Wong W, Zhan Y, et al. *HiC-DC+ enables systematic 3D interaction calls and differential analysis for Hi-C and HiChIP*. Nature Communications. 2021. DOI: 10.1038/s41467-021-23749-x.

2. Open2C, Abdennur N, Abraham S, et al. *Cooltools: enabling high-resolution Hi-C analysis in Python*. PLoS Computational Biology. 2024. DOI: 10.1371/journal.pcbi.1012067.

3. Yaffe E, Tanay A. *Probabilistic modeling of Hi-C contact maps eliminates systematic biases to characterize global chromosomal architecture*. Nature Genetics. 2011. DOI: 10.1038/ng.947.

4. Wang X, Xu J, Zhang B, et al. *Genome-wide detection of enhancer-hijacking events from chromatin interaction data in rearranged genomes*. Nature Methods. 2021. DOI: 10.1038/s41592-021-01164-w.

5. Jorge E, Foissac S, Neuvial P, Zytnicki M, Vialaneix N. *A comprehensive review and benchmark of differential analysis tools for Hi-C data*. Briefings in Bioinformatics. 2025. DOI: 10.1093/bib/bbaf074.

6. Seabold S, Perktold J. *Statsmodels: Econometric and Statistical Modeling with Python*. Proceedings of the 9th Python in Science Conference. 2010. DOI: 10.25080/Majora-92bf1922-011.

7. statsmodels developers. *statsmodels.genmod.families.family.NegativeBinomial*. statsmodels stable documentation, accessed May 31, 2026. `https://www.statsmodels.org/stable/generated/statsmodels.genmod.families.family.NegativeBinomial.html`.