Submitted to the Annals of Applied Statistics

STATISTICAL CURVE MODELS FOR INFERRING 3D CHROMATIN ARCHITECTURE

BY ELENA TUZHILINA[1], TREVOR HASTIE[2] AND MARK SEGAL[3]

[1]*Department of Statistical Sciences, University of Toronto, Toronto ON M5S3G3 Canada, elenatuz@stanford.edu*

[2]*Department of Statistics, Stanford University, Stanford CA 94305 USA, hastie@stanford.edu*

[3]*Department of Epidemiology and Biostatistics, University of California, San Francisco CA 94143 USA, Mark.Segal@ucsf.edu*

Reconstructing three dimensional (3D) chromatin structure from conformation capture assays (such as Hi-C) is a critical task in computational biology, since chromatin spatial architecture plays a vital role in numerous cellular processes and direct imaging is challenging. Most existing algorithms that operate on Hi-C contact matrices produce reconstructed 3D configurations in the form of a polygonal chain. However, none of the methods exploit the fact that the target solution is a (smooth) curve in 3D: this contiguity attribute is either ignored or indirectly addressed by imposing spatial constraints that are challenging to formulate. In this paper we develop both B-spline and smoothing spline techniques for directly capturing this potentially complex 1D curve. We subsequently combine these techniques with a Poisson model for contact counts and compare their performance on a real data example. In addition, motivated by the sparsity of Hi-C contact data, especially when obtained from single-cell assays, we appreciably extend the class of distributions used to model contact counts. We build a general distribution-based metric scaling (*DBMS*) framework, from which we develop zero-inflated and Hurdle Poisson models as well as negative binomial applications. Illustrative applications make recourse to bulk Hi-C data from IMR90 cells and single-cell Hi-C data from mouse embryonic stem cells.

1. Introduction. The task of reconstructing the three-dimensional (3D) configuration of chromatin (for a single chromosome) within the eukaryotic nucleus from pairwise contact assays, notably Hi-C (Lieberman-Aiden et al., 2009; Duan et al., 2010; Rao et al., 2014), is motivated by (at least) three considerations. First, such architecture is critical to an array of cellular processes, particularly transcription, but even memory formation (Marco et al., 2020). Second, armed with such an inferred configuration, we can superpose genomic attributes, enabling biological insights not accessible from the primary Hi-C contact matrix readout. Examples here include gene expression gradients and co-localization of virulence genes in the malaria parasite (Ay et al., 2014), the impact of spatial organization on double strand break repair (Lee et al., 2016), and elucidation of '3D hotspots' corresponding to (say) overlaid ChIP-Seq transcription factor extremes which can reveal novel regulatory interactions (Capurso, Bengtsson and Segal, 2016). Third, despite notable gains in imaging methodologies (Payne et al., 2021), such direct access to structure is yet to enjoy the resolution and uptake conferred by Hi-C assays.

This set of factors has led to a wealth of 3D reconstruction algorithms: a recent review (Oluwadare, Highsmith and Cheng, 2019) identified over 30 methods and there have numerous additions in subsequent years. However, the very notion of 'a' 3D reconstruction is simplistic, genomes being dynamic and variable with differences according to organism, tissue, cell-type, cell-cycle, and cell. Hi-C experiments are frequently performed on large, synchronized cell-type specific populations, so that resultant reconstructions are interpreted

Keywords and phrases: spatial structure, conformation reconstruction, metric scaling, splines.

as providing a consensus configuration. The emergence of single cell Hi-C (scHi-C, Ramani et al., 2017; Stevens et al., 2017) has enabled dissection of inter-cellular structural variation, at the expense of yielding appreciably sparser data. Developing reconstruction methodology to accommodate such sparsity is one of our contributions; see Sections 11-14.

Another component of structural variation is allelic: in diploid organisms maternal and paternal homologs can adopt differing configurations. This poses difficulties for reconstruction algorithms since Hi-C readout is generally unphased, and resultant contacts are ambiguous as to whether they are intra- or inter- homolog. Until recently, these concerns have been ignored but novel approaches attempt to resolve identifiability issues by either prescribing assumptions and/or invoking additional data sources (Cauer et al., 2019; Belyaeva et al., 2021). We do not address these aspects here, but note that the concerns can be sidestepped if phasing of Hi-C output can be achieved, HiCHap (Luo et al., 2020) being an accurate tool for doing so.

Clearly, the forefront structural feature of a single chromosome is its contiguity, followed by its folding complexity, necessary to achieve the compaction needed to fit within the nucleus. And it is these features that our original 3D reconstruction algorithm, Poisson metric scaling (*PoisMS*, Tuzhilina, Hastie and Segal, 2020) addressed. In prior work contiguity had been tackled by imposing constraints (Duan et al., 2010; Ay et al., 2014; Stevens et al., 2017), which are cell type specific and require prescription of constraint parameters. Given a paucity of relevant background biological measurement, these parameters can be difficult to specify. Further, their inclusion substantially increases computational burden. Other approaches (Zhang et al., 2013; Park and Lin, 2017; Rieber and Mahony, 2017) ignore contiguity in the reconstruction process, imposing it post hoc by "connecting the dots" of the 3D solution according to the ordering of corresponding genomic loci.

We review our *PoisMS* methodology, that extends principal curves (Hastie and Stuetzle, 1989) to the metric scaling problem, in Section 2. Previously, we had used B-spline bases as primitives for obtaining chromatin configuration but, as described in Section 3, this formulation is problematic when used with cross-validation to determine smoothness degree (for reasons detailed in Sections 3 and 10), which is critical for appropriately capturing the above-mentioned second key attribute, folding complexity. Accordingly, in Sections 3 through 9, we introduce a smoothing spline basis, and attendant algorithm *SPoisMS*. We describe how this improvement enables effective cross-validation and mitigates initialization concerns, which, subsequently, is demonstrated via a series of experiments. We also propose how *SPoisMS* can be efficiently implemented by exploiting its connections with *PoisMS*, and develop degrees-of-freedom estimates facilitating calibrated methods comparison.

We then turn attention to issues surrounding sparsity and overdispersion – characteristic of all contact matrices but especially extreme for data deriving from single cell Hi-C assays, the importance of which has already been noted. These features make Poisson assumptions inappropriate. To address these concerns we advance general distribution-based metric scaling methodology and algorithms (Section 12), then specialize to select special cases: hurdle Poisson, zero-inflated Poisson, negative binomial (Section 13), and finally make comparisons between these and the original Poisson formulation (Section 14) before providing concluding Discussion and directions for further work.

2. The *PoisMS* method. We provide a brief overview of our recently proposed Poisson metric scaling (*PoisMS*, Tuzhilina, Hastie and Segal, 2020), a novel approach which directly models 3D chromatin configuration by a 1D smooth curve. The result of a Hi-C experiment is the *contact map*, a symmetric matrix $C = [C_{ij}] \in \mathbb{Z}_+^{n\times n}$ of contact counts between n (binned) genomic loci i, j. While such counts are obtained on a genome-wide basis our focus is solely on individual chromosomes. The 3D chromatin reconstruction problem is to use the contact matrix C to obtain a 3D point configuration $x_1, \ldots, x_n \in \mathbb{R}^3$ corresponding to the spatial coordinates of loci $1, \ldots, n$ respectively.

Denote the matrix of the loci spacial coordinates by $X = \begin{pmatrix} -x_1^\top - \\ \cdots \\ -x_n^\top - \end{pmatrix} \in \mathbb{R}^{n\times 3}$. The core assumption of the model is

$$x_1, \ldots, x_n \in \gamma, \text{ where } \gamma \text{ is a smooth one-dimensional curve in } \mathbb{R}^3. \tag{1}$$

We express this assumption in matrix form as follows. First, if curve γ is parametrized by t and t_i indexes the genomic locus of x_i, then (1) can be reformulated as the set of equations $x_i = \gamma(t_i)$. Next, to impose smoothness, we specify that each component of $\gamma(t) = (\gamma_1(t), \gamma_2(t), \gamma_3(t))^\top$ is a cubic spline and so can be represented by a linear combination of basis functions:

$$\gamma_j(t) = \textstyle\sum_{\ell=1}^{k} \Theta_{\ell j}\, h_\ell(t), \text{ where } h_1(t), \ldots, h_k(t) \text{ is a cubic spline basis.}$$

Here k is the size of the basis and is the hyperparameter that controls the "wiggliness" of the resulting reconstruction, which we will subsequently refer to as *degrees-of-freedom*. Finally, if $H \in \mathbb{R}^{n\times k}$ is the matrix with elements $H_{i\ell} = h_\ell(t_i)$ and $\Theta \in \mathbb{R}^{k\times 3}$ represents the spline coefficient matrix, then the smoothing constraint can be rewritten in matrix form as $X = H\Theta$.

Next, we introduce the probabilistic model for the contact counts. To do so, we use the Poisson distribution and link the Poisson parameters to the pairwise distances between loci. Specifically, we assume that $C_{ij} \sim \text{Pois}(\lambda_{ij})$ and

$$\log(\lambda_{ij}) = -\|x_i - x_j\|^2 + \beta, \tag{2}$$

where β is an unknown intercept. The resulting negative log-likelihood objective is therefore

$$\ell_{\textit{PoisMS}}(X, \beta; C) = \frac{1}{n^2} \sum_{1\le i,j\le n} \left[e^{-\|x_i - x_j\|^2 + \beta} - C_{ij} \left(-\|x_i - x_j\|^2 + \beta \right) \right]. \tag{3}$$

Combining (3) with the smoothing constraint leads us to the following MLE problem

$$\underset{\Theta\in\mathbb{R}^{k\times 3},\ \beta\in\mathbb{R}}{\text{minimize}}\ \ell_{\textit{PoisMS}}(H\Theta, \beta; C). \tag{4}$$

Our particular formulation for the log-link (2) retains the heuristic that loci close in 3D should have higher expected contact counts and results in an objective that is equivalent to, but much easier to optimize, than, the more conventional exponential link $\lambda_{ij} = \beta\|x_i - x_j\|^\alpha$ (Rosenthal et al., 2019). In Tuzhilina, Hastie and Segal (2020) we propose an efficient iterative algorithm that recovers the solution to (4), schematically outlined as follows.

Poisson metric scaling (*PoisMS*)

---

1. For the fixed intercept β we compute the second order approximation (SOA) to the *PoisMS* loss. The approximation is done at the current guess of the reconstruction X and it has the form of weighted Frobenius norm:

   $$\ell_{\textit{PoisMS}}(X, \beta; C) \approx \ell_{\textit{WPCMS}}(X; Z, W) = \|\sqrt{W} * (Z - D^2(X))\|_F^2.$$

   Here $W = e^{-D^2(X)+\beta}$ and $Z = D^2(X) - \frac{C}{W} + 1$ are the weight and working response matrices, both depending on the current guesses for X and β, and $D(X)$ corresponds to the pairwise distance matrix with elements $D_{ij} = \|x_i - x_j\|$. *WPCMS* connotes weighted principal curve metric scaling.
2. Combining the SOA with the smoothing constraint $X = H\Theta$ gives the *WPCMS* problem:

   $$\underset{\Theta\in\mathbb{R}^{k\times 3}}{\text{minimize}}\, \ell_{\textit{WPCMS}}(H\Theta). \tag{5}$$

   This problem can be considered as an approximation of (4).

3. We solve (5) via the *WPCMS* algorithm using projected gradient descent. Performed in the space of similarity matrices $S(X) = XX^\top$, it alternates between two steps:
   - the gradient step moving S along its gradient;
   - the projection step mapping S back onto the space of spline similarity matrices.

   In detail, the gradient step updates the similarity matrix as

$$S := S - \nabla_S \ell_{WPCMS} = XX^\top - \Phi(W * (Z - D^2(X))),$$

   where $\Phi(G) = G - \mathrm{diag}(G \cdot 1)$. Then the projection step minimizes the distance between S and the space of similarity matrices

$$\ell_{PCMS}(X; S) = \|S - XX^\top\|_F^2.$$

   Combined with the smoothing constraint this leads us to the optimization problem that we hereafter will refer to as principal curve metric scaling (*PCMS*):

$$\underset{\Theta \in \mathbb{R}^k}{\text{minimize}}\, \ell_{PCMS}(H\Theta).$$

   The *PCMS* problem has an explicit solution: if H has orthonormal columns, it can be found via the singular value decomposition of $H^\top S H$.
4. *WPCMS* returns a new coefficient matrix Θ thereby updating the reconstruction $X = H\Theta$. Finally, for fixed X we optimize the *PoisMS* loss w.r.t. β thus updating the intercept:

$$\beta := \log\left(\frac{\sum_{1 \le i,j \le n} C_{ij}}{\sum_{1 \le i,j \le n} e^{-\|x_i - x_j\|^2}}\right).$$

---

3. Improved approach: smoothing splines. While the *PoisMS* approach has proved effective, there are some caveats. First, since the objective in (4) is non-convex, the *PoisMS* algorithm converges to a local minimum. Thus, initialization for Θ and β can impact the resulting reconstruction. Although we did not observe appreciable variation in the *PoisMS* solutions over a range of experiments, initialization choice remains an open question. We provide more details in the Appendix of Tuzhilina, Hastie and Segal (2020). Second, it is not clear how to perform cross-validation in the contact matrix context. Our original idea was to hold out some chromatin loci, pretending that they were unobserved, train the *PoisMS* model on the observed loci, then evaluate the fit on the held-out set. The training procedure is equivalent to removing a subset of rows and columns from the contact matrix, eliminating the corresponding rows in the spline basis matrix, and performing reconstructing using only the observed blocks of C and H. Although this approach may seem reasonable, care is needed in selecting unobserved loci: strong contact matrix correlations can derail cross-validation when loci are chosen at random.

To illustrate these issues we utilize Hi-C data for IMR90 cells chromosome 20 (Dixon et al., 2012), obtained from the Gene Expression Omnibus (GEO) with accession GSE35156, there being $n = 599$ loci when binned at 100kb resolution. We compute a 3D reconstruction X via *PoisMS* with 25 degrees-of-freedom, the value determined as optimal using the "elbow heuristic" (Tuzhilina, Hastie and Segal, 2020), an approach used in lieu of cross-validation. The matrix of the Poisson parameters $\Lambda = e^{-D^2(X)+\beta}$ and attendant error matrix $E = C - \Lambda$ are then computed, and a heatmap for the pairwise correlations between rows in E is displayed in the left panel of Figure 1. The numerous instances of extreme (near one) correlation, both on and off the diagonal, demonstrate the extent of strong correlations between neighboring rows of the error matrix.

To overcome the problems deriving from these correlations in performing cross-validation by selecting unobserved loci at random, we attempted using block cross-validation, removing

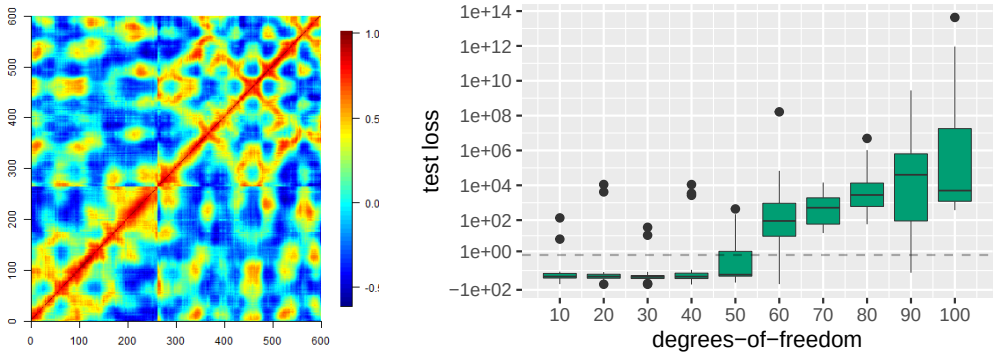


FIG 1. *Left panel: heatmap of the correlation matrix between the rows of the error matrix $E = \Lambda - C$, where Λ is the Poisson parameter matrix obtained from the PoisMS reconstruction. Spearman correlation is used for robustness. Numerous instances of high correlation between neighboring are evident. Right panel: Block-cross validation performance with 20 folds. The PoisMS approach is trained on 19/20 folds, the test score is measured on 1/20 held-out set. The plot represents the dependence of the test loss (log-scaled) on the degrees-of-freedom ads shows the dramatic increase in the average test score as well as the variance with the growth of degrees-of-freedom. The outliers for most boxplots correspond to holding out the two extreme blocks (number one 1 and 20). The grey dashed line corresponds to zero.*

rows in H and rows (and corresponding columns) in C in continuous blocks. This is where *PoisMS* encounters difficulties. Specifically, since each element of the B-spline basis has finite support, the spline basis matrix tends to be sparse. Thus, eliminating a block of rows in H can potentially exclude one or multiple elements from the basis, resulting in very high reconstruction variance. We illustrate this fact by measuring the test loss on a held-out set while performing 20-fold block cross-validation for the *PoisMS* technique (see Section 10 for more details on the procedure). Figure 1 demonstrates the dramatic decay in the test performance as well as the significant increase in the test score variance.

Accordingly, we develop an approach based on smoothing splines that resolves these initialization and cross-validation concerns. First, we rewrite (3) in terms of the curve γ as

$$\ell_{\mathit{SPoisMS}}(\gamma, \beta; C) = \frac{1}{n^2} \sum_{1 \le i,j \le n} \left[e^{-\|\gamma(t_i) - \gamma(t_j)\|^2 + \beta} - C_{ij} \left(-\|\gamma(t_i) - \gamma(t_j)\|^2 + \beta \right) \right]. \tag{6}$$

Instead of incorporating smoothness with the B-spline basis we form the penalized loss

$$\ell_{\mathit{SPoisMS}}(\gamma, \beta; C, \lambda) = \ell_{\mathit{PoisMS}}(\gamma, \beta; C) + \lambda \int \|\gamma''(t)\|^2 \, dt. \tag{7}$$

The second term in the objective measures the curve's smoothness via its second-order derivative, usually termed the *roughness penalty*. We state an alternative smoothing Poisson metric scaling (*SPoisMS*) optimization problem:

$$\underset{\gamma \in \Gamma,\ \beta \in \mathbb{R}}{\text{minimize}}\, \ell_{\mathit{SPoisMS}}(\gamma, \beta; C, \lambda). \tag{8}$$

Here Γ is the set of all smooth (second-order derivative exists) one-dimensional curves in $\mathbb{R}^3$. Note that in *SPoisMS* the "wiggliness" of the resulting reconstruction is controlled in a different way than in *PoisMS*, where it depends on the basis size k. In SpoisMS, increasing λ will place a greater penalty on the second derivative of γ, thereby encouraging smoother solutions.

From standard smoothing spline theory (e.g., Green and Silverman (1994) and Wahba (1990)) we have that each component of the optimal solution γ is a natural cubic spline with

knots $t_1, \ldots, t_n$. We restate problem (8) in matrix form as follows. Denote by $n_1(t), \ldots, n_n(t)$ the natural spline basis. Let $N \in \mathbb{R}^{n\times n}$ be the matrix representing the basis evaluations at the knots, i.e. $N_{i\ell} = n_\ell(t_i)$, and $\Omega \in \mathbb{R}^{n\times n}$ be the penalty matrix with elements $\Omega_{i\ell} = \int n_i''(t) n_\ell''(t)\, dt$. Then, for each γ, there exists $\Theta \in \mathbb{R}^{n\times 3}$ such that the objective (7) is equivalent to

$$\ell_{\textit{SPoisMS}}(\Theta, \beta; C, \Omega, \lambda) = \ell_{\textit{PoisMS}}(N\Theta, \beta; C) + \lambda \operatorname{tr}(\Theta^\top \Omega \Theta).$$

Now, let $K = N^{-T}\Omega N$. Since N is full-rank and non-singular, using a change of variables $X = N\Theta$ we restate the optimization problem (8) as

$$\underset{X\in\mathbb{R}^{n\times 3},\ \beta\in\mathbb{R}}{\text{minimize}}\ \ell_{\textit{SPoisMS}}(X, \beta; C, K, \lambda) = \ell_{\textit{PoisMS}}(X, \beta; C) + \lambda \operatorname{tr}(X^\top K X). \tag{9}$$

4. Link between *PoisMS* and the smoothing spline approach. The matrix K has several important properties. We exploit these to demonstrate that the *SPoisMS* loss calculated for the original contact matrix C is equivalent to the *PoisMS* loss computed for $C - \frac{\lambda n^2}{2} K$. Consequently, the *SPoisMS* problem can be readily solved by applying the original *PoisMS* method to an adjusted version of the contact matrix.

Let $K = UDU^\top$ be the eigendecomposition of the penalty matrix. K has two zero eigenvalues and the corresponding eigenvectors span the subspace of linear functions (Green and Silverman, 1994). This implies that $K\mathbf{1} = 0$ and $\mathbf{1}^\top K = 0$, which further implies $\sum_{1\le i,j,\le n} K_{ij} = 0$ (see Appendix A for details). Here $\mathbf{1} = (1, \ldots, 1)^\top \in \mathbb{R}^n$ is the n-dimensional vector of ones. These properties enable linking SpoisMS and *PoisMS* losses per Lemma 1.

LEMMA 1. *If K is the smoothing spline penalty matrix and $C^\lambda = C - \frac{\lambda n^2}{2} K$ then*

$$\ell_{\textit{SPoisMS}}(X, \beta; C, K, \lambda) = \ell_{\textit{PoisMS}}(X, \beta; C^\lambda).$$

PROOF. Recall that $S(X) = XX^\top$ is the matrix of the pairwise inner products between genomic loci and $D(X)$ is the pairwise distance matrix with elements $D_{ij} = \|x_i - x_j\|$. Denote the diagonal of the inner product matrix by

$$s(X) = \operatorname{diag}(S(X)) = \left(\|x_1\|^2, \|x_2\|^2, \ldots, \|x_n\|^2\right)^\top.$$

$S(X)$ and $D(X)$ are related via

$$D^2(X) = s(X) \cdot \mathbf{1}^\top + \mathbf{1} \cdot s(X)^\top - 2S(X). \tag{10}$$

Combining this equation with the properties of K and trace it is easy to show that

$$\operatorname{tr}(X^\top K X) = \operatorname{tr}(KS(X)) = -\frac{1}{2}\operatorname{tr}(KD^2(X)) = \frac{1}{2}\sum_{1\le i,j,\le n} K_{ij}(-\|x_i - x_j\|^2 + \beta).$$

The following steps conclude the proof:

$$\begin{aligned}
&\ell_{\textit{SPoisMS}}(X, \beta; C, K, \lambda) = \ell_{\textit{PoisMS}}(X, \beta; C) + \lambda \operatorname{tr}(X^\top K X) = \\
&= \frac{1}{n^2}\sum_{1\le i,j\le n}\left[e^{-\|x_i - x_j\|^2 + \beta} - \left(C_{ij} - \frac{\lambda n^2}{2} K_{ij}\right)\left(-\|x_i - x_j\|^2 + \beta\right)\right] = \\
&= \ell_{\textit{PoisMS}}(X, \beta; C^\lambda).
\end{aligned}$$

□

It is through this lemma that implementation of *SPoisMS* is appreciably simplified, since instead of requiring development of new iterative algorithms to solve (9), we can just use the existing *PoisMS* algorithm applied to the adjusted contact matrix C^λ.

5. Truncating the basis. Lemma 1 demonstrates that (9) is equivalent to solving the unconstrained *PoisMS* problem

$$\underset{X\in\mathbb{R}^{n\times 3},\ \beta\in\mathbb{R}}{\text{minimize}}\ \ell_{\textit{PoisMS}}\big(X,\beta;C-\frac{\lambda n^2}{2}K\big). \tag{11}$$

Recall that if there was the smoothing constraint $X = H\Theta$, the projection step in the *PoisMS* algorithm would involve the singular value decomposition of the $k \times k$ matrix $H^\top SH$, which is computationally cheap for small k (see Section 2 for details). However, in the case of the unconstrained problem (11), the projection is problematic as now it requires us to compute the SVD of the large $n \times n$ matrix S at each iteration.

There are several ways to circumvent this computational concern. Note that the *PoisMS* projection step aims to find Θ that minimizes the Frobenius distance between the similarity matrix S and $H\Theta\Theta^\top H^\top$, which, for orthogonal H, is equivalent to solving

$$\underset{\Theta\in\mathbb{R}^{k\times 3}}{\text{minimize}}\ \|H^\top SH-\Theta\Theta^\top\|_F^2.$$

Since this step seeks the rank-three approximation of $H^\top SH$, it is sufficient to compute only the first three singular vectors of this matrix. Our first computational trick is to use the block power method (Bentbib and Kanber, 2017), which is more efficient for finding a few singular vectors than calculating the full SVD.

The second trick involves including a constraint in (11) such that the resulting optimization problem accurately approximates (9) but requires less costly SVD steps. Recall that we decomposed the penalty matrix as $K = U^\top DU$. Since the columns of U form the natural spline basis (the *Demmler-Reinsch basis*), the resulting reconstruction can be represented as $X = U\Theta$ leading to an alternative form $\mathrm{tr}(\Theta^\top D\Theta)$ for the smoothing penalty. We then exploit a feature of the Demmler-Reinsch basis: the wigglier the basis element, the larger the corresponding eigenvalue. So, increased smoothness of the reconstruction can be achieved by increased penalization of the coefficients in Θ corresponding to the largest diagonal elements in D, and thus the wiggliest part of the basis U. Since some of the columns of U have little effect on the resulting reconstruction we can remove them from the basis. Accordingly, we can set $H \in \mathbb{R}^{n\times k}$ to the eigenvectors from U corresponding to the largest k eigenvalues and solve the alternative problem

$$\underset{\Theta\in\mathbb{R}^{k\times 3},\ \beta\in\mathbb{R}}{\text{minimize}}\ \ell_{\textit{PoisMS}}\big(H\Theta,\beta;C-\frac{\lambda n^2}{2}K\big). \tag{12}$$

Again, the solution is readily done using the *PoisMS* algorithm and, for k sufficiently large, the resulting will accurately approximate the solution from (11). However, since each projection step in *PoisMS* involves the SVD of a smaller $k \times k$ matrix, it is reached more efficiently.

6. *SPoisMS* and *PoisMS* comparison. There are several advantages in using smoothing splines (*SPoisMS*) rather than B-splines (*PoisMS*) for 3D chromatin reconstruction. Controlling smoothness by the size of the B-spline basis requires recomputing matrix H for each degrees-of-freedom value. In contrast, H is fixed for smoothing splines, with the penalty factor determining model flexibility. This implies that $\Theta \in \mathbb{R}^{k\times 3}$ is the same shape for any value of λ. This is important with regard to *SPoisMS* initialization. In particular, the *path of solutions* can be generated by gradually decreasing the penalty factor and re-using Θ obtained for larger λ as a warm start for smaller values. Additionally, the support for each Demmler-Reinsch basis function is $\mathbb{R}$. So, with respect to the cross-validation issues discussed in Section 3, block cross-validation would not be subject to the instabilities affecting use of the B-spline basis.

The main disadvantage of *SPoisMS* is that the basis size k needs to be relatively large in order to ensure an accurate solution. This makes each step of the *SPoisMS* computationally more expensive than a *PoisMS* step. Further, controlling smoothness by the (B-spline) basis size, as per *PoisMS*, admits a natural interpretation as degrees-of-freedom. However, while it is apparent that increasing λ should decrease the degrees-of-freedom (df) for *SPoisMS*, formally determining df in this case is less immediate, a problem we tackle next.

7. Degrees-of-freedom. The notion of *effective* degrees-of-freedom is very well studied in the context of linear models, one definition being the trace of the "hat" matrix (Hastie, Tibshirani and Friedman, 2009). In particular, in their chapter 5.4 the following formula was derived for smoothing splines: if $y, x \in \mathbb{R}^n$ and $d_1, \dots, d_n$ are the eigenvalues of the Demmler-Reinsch penalty matrix K then the loss function

$$\ell(f) = \frac{1}{n}\|y - x\|^2 + \lambda x^\top K x \tag{13}$$

implies the degrees-of-freedom as

$$df = \sum_{i=1}^{n} \frac{1}{1 + \lambda n d_i}. \tag{14}$$

In this section we propose a method for estimating df for the *SPoisMS* model. The general idea is to derive the appropriate approximation of the *SPoisMS* loss in such a way that it would resemble the standard smoothing spline loss (13). This will require us to prove the following lemma.

LEMMA 2. *The projection step of SPoisMS solves the following penalized PCMS problem for some $\tilde{S}$ independent of λ:*

$$\underset{X \in \mathbb{R}^{n\times 3}}{\text{minimize}} \frac{1}{n^2}\|\tilde{S} - XX^\top\|_F^2 + \lambda \operatorname{tr}(X^\top K X). \tag{15}$$

See Appendix B for the proof. This lemma gives us an advantage to simplify the *SPoisMS* loss as, at the convergence point and for some $\tilde{S}$, the *SPoisMS* solution coincides with the solution to problem (15). Therefore, if we derive a df formula for penalized *PCMS* it can serve as an accurate approximation to the *SPoisMS* degrees-of-freedom.

Now we need to account for the fact that $X \in \mathbb{R}^{n\times 3}$ is a matrix with three columns, whereas standard formula (14) is defined for a vector $x \in \mathbb{R}^n$. Note that the penalized *PCMS* problem can be solved in two ways. First, one can rewrite it as

$$\underset{X \in \mathbb{R}^{n}}{\text{minimize}} \left\| \left(\tilde{S} - \frac{\lambda n^2}{2} K\right) - XX^\top \right\|_F^2$$

and find X via the eigendecomposition of $\tilde{S} - \frac{\lambda n^2}{2} K$. Alternatively, one can apply the coordinate descent type of the algorithm alternating among the three columns of $X = (X_1, X_2, X_3)$. Specifically, if we fix X_2 and X_3, we can update X_1 by solving a one-dimensional penalized *PCMS* problem as follows

$$\underset{X_1 \in \mathbb{R}^{n}}{\text{minimize}} \frac{1}{n^2}\|(\tilde{S} - X_2 X_2^\top - X_3 X_3^\top) - X_1 X_1^\top\|_F^2 + \lambda X_1^\top K X_1. \tag{16}$$

By analogy, we can update X_2 and X_3 via fixing the remaining two columns. This approach implies that at the convergence point, the three-dimensional problem (15) can be replaced by three one-dimensional analogs (one per a column of X). In the following lemma we propose a definition of degrees-of-freedom for a one-dimensional version of penalized *PCMS*.

LEMMA 3. *The degrees-of-freedom for one-dimensional penalized PCMS*

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ \frac{1}{n^2}\|\tilde{S}-xx^\top\|_F^2+\lambda x^\top Kx$$

can be calculates as

$$df=\sum_{i=1}^{n}\frac{1}{1+\frac{\lambda n^2}{\|x_0\|^2}d_i},$$

where $x_0\in\mathbb{R}^n$ is the solution to the problem.

The proof can be found in Appendix B. An important observation from the lemma is that the proposed formula for df is independent from $\widetilde{S}$. Therefore, we can compute degrees-of-freedom separately for each column of X and use the average as df for *SPoisMS*. This leads us to the final definition of the *SPoisMS* degrees-of-freedom.

THEOREM 1. *If $X_0=(X_{01},X_{02},X_{03})$ is the solution to (9) then the effective degrees-of-freedom can be approximated by*

$$df=\frac{1}{3}\sum_{i=1}^{n}\sum_{j=1}^{3}\frac{1}{1+\frac{\lambda n^2}{\|X_{0j}\|^2}d_i}. \tag{17}$$

8. IMR90 cell experiments. We evaluated the *SPoisMS* approach using the IMR90 cell, chromosome 20 Hi-C data described in Section 3. First, we compute reconstructions via the *SPoisMS* method with the penalty factor ranging over $\lambda=10^4,10^3,100,10,1,0.1$. To speed up the computations we truncate the Demmler-Reinsch basis, including only the eigenvectors that correspond to the $k=300$ smallest eigenvalues of K (see Section 5 for details). We then generate the solution path by starting from the largest $\lambda=10^4$, and reusing Θ and β calculated at the previous step as a warm start for the subsequent value of λ. The resulting fits are presented in Figure 3 and reveal the increased wiggliness that results from decreasing the penalty factor thereby imposing less smoothness. Next, for each λ value we report the corresponding degrees-of-freedom computed via (17) in the table below which reveals that the range of λ considered covers degrees-of-freedom from $df\approx 5$ (very smooth reconstruction) to $df\approx 83$ (very wiggly). As a final step, to compare the performance of *SPoisMS* and the original *PoisMS* method, we round the resulting values of df and compute the *PoisMS* reconstructions using the B-spline basis of size df and random initialization. The resulting fits are displayed in Figure 3.

$\boldsymbol{\lambda}$	10^4	10^3	100	10	1	0.1
df	5.15	8.77	15.21	26.71	47.24	83.44

Analyzing Figures 2 and 3 leads us to the following conclusions. First, comparing the *PoisMS* and *SPoisMS* solutions we observe that the degrees-of-freedom formula derived in Section 7 produces quite an accurate approximation. To quantify how well our df definition represents nature, in Appendix C, we run additional experiments measuring the wiggliness of each reconstruction in terms of its average curvature. Second, we conclude that reusing *SPoisMS* solutions as a warm start produces a well-aligned sequence of fits that evolve in

a continuous way with decrease of λ. In contrast, we observe less agreement between the *PoisMS* solutions, which the random initialization of the method can explain. Finally, the left panel of Figure 4 demonstrates that using warm-starts for subsequent values of λ substantially decreases the total number of iterations required for the *SPoisMS* method to converge. This illustrates the utility of the smoothing spline from the computational point of view.

9. Model validation. The lack of 3D chromatin configuration image data for many cell types at suitable resolutions makes assessing accuracy of candidate Hi-C derived reconstructions challenging. Here we develop an approach to appraising model validity based on partitioning genomic loci into training and testing sets. Suppose we have only partial access to the contact matrix, i.e. we assume that some of loci are unobserved and denote by $\mathcal{O} \subseteq \{1, \ldots, n\}$ the set of observed loci and its complement by $\mathcal{U}$. To effect model evaluation we first use loci from $\mathcal{O}$ to compute the model coefficients Θ and intercept β, and then use $\mathcal{U}$ to evaluate the model performance. How this is accomplished for *PoisMS* and *SPoisMS* respectively is described next.

9.1. *PoisMS performance evaluation.* To simplify notation, without loss of generality we can assume that the rows and columns in C are permuted such that the first $|\mathcal{O}|$ of them correspond to contacts for the observed loci. Then the contact matrix has block-structure $C = \left(\begin{smallmatrix} C_{\mathcal{O}\mathcal{O}} & C_{\mathcal{O}\mathcal{U}} \\ C_{\mathcal{U}\mathcal{O}} & C_{\mathcal{U}\mathcal{U}} \end{smallmatrix}\right)$, which, in turn, implies the structure in the basis matrix $H = \left(\begin{smallmatrix} H_{\mathcal{O}} \\ H_{\mathcal{U}} \end{smallmatrix}\right)$.

To fit the *PoisMS* model on the observed data we evaluate the log-likelihood only at contacts from $\mathcal{O}$, yielding the training *PoisMS* loss

$$\ell^{\textit{train}}_{\textit{PoisMS}}(X, \beta; C) = \frac{1}{|\mathcal{O}|^2} \sum_{(i,j) \in \mathcal{O} \times \mathcal{O}} \left[e^{-\|x_i - x_j\|^2 + \beta} - C_{ij} \left(-\|x_i - x_j\|^2 + \beta \right) \right]. \tag{18}$$

Here $|\mathcal{O}|$ is the number of observed loci. It is not hard to show that

$$\ell^{\textit{train}}_{\textit{PoisMS}}(X, \beta; C) = \ell_{\textit{PoisMS}}(X_{\mathcal{O}}, \beta; C_{\mathcal{O}\mathcal{O}}),$$

therefore, minimizing (18) subject to the smoothing constraint $X = H\Theta$ is equivalent to solving the problem

$$\underset{\Theta \in \mathbb{R}^{k \times 3},\ \beta \in \mathbb{R}}{\text{minimize}}\ \ell_{\textit{PoisMS}}(H_{\mathcal{O}}\Theta, \beta; C_{\mathcal{O}\mathcal{O}}). \tag{19}$$

Thus, the solution can be found by applying *PoisMS* to contact matrix $C_{\mathcal{O}\mathcal{O}}$ with basis $H_{\mathcal{O}}$.

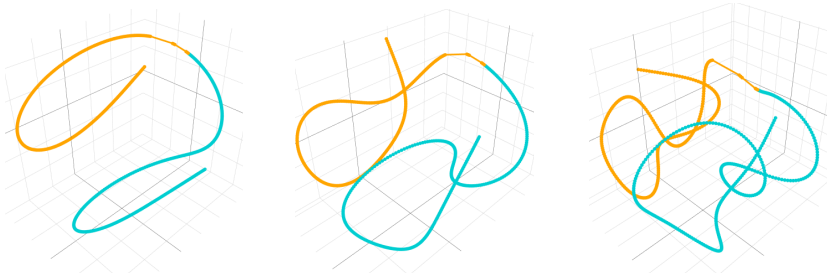
Now we proceed to the model evaluation step. Note that the original *PoisMS* algorithm requires the basis matrix to have orthonormal columns, which while true for the full H, but may not be the case for the sub-matrix $H_{\mathcal{O}}$. So, before fitting the *PoisMS* model, we compute the QR-decomposition $H_{\mathcal{O}} = QR$ and replace the basis matrix by Q. In order to evaluate how well the parameters (coefficient matrix Θ, intercept β) obtained from solving the *PoisMS* problem fit the remaining unseen contact counts we compute the complementary test loss

$$\ell^{\textit{test}}_{\textit{PoisMS}}(X, \beta; C) = \frac{1}{n^2 - |\mathcal{O}|^2} \sum_{(i,j) \notin \mathcal{O} \times \mathcal{O}} \left[e^{-\|x_i - x_j\|^2 + \beta} - C_{ij} \left(-\|x_i - x_j\|^2 + \beta \right) \right]. \tag{20}$$

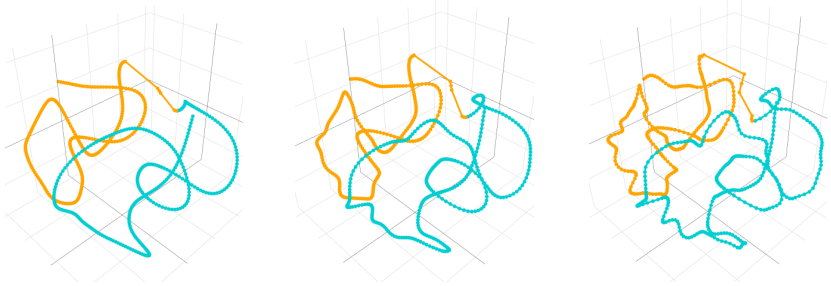
Since this loss involves contacts between both observed and unobserved genomic loci it is necessary to recover the full reconstruction, which is easily obtained via the formula

$$X = \left(\begin{smallmatrix} X_{\mathcal{O}} \\ X_{\mathcal{U}} \end{smallmatrix}\right) = \left(\begin{smallmatrix} Q\Theta \\ H_{\mathcal{U}} R^{-1} \Theta \end{smallmatrix}\right).$$

This reconstruction is subsequently plugged in (20) yielding the model test score.

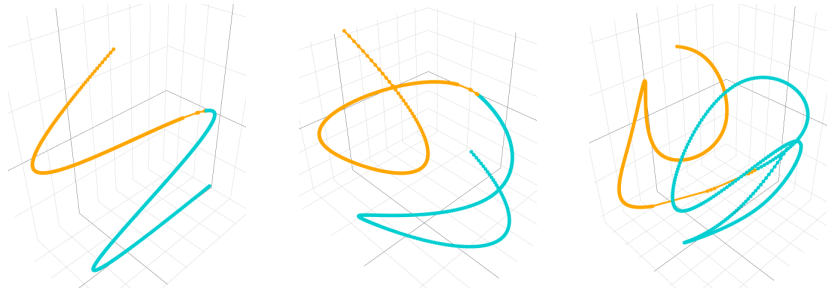


(a) $\lambda = 10^4$, $df \approx 5.15$ (b) $\lambda = 10^3$, $df \approx 8.77$ (c) $\lambda = 100$, $df \approx 15.21$

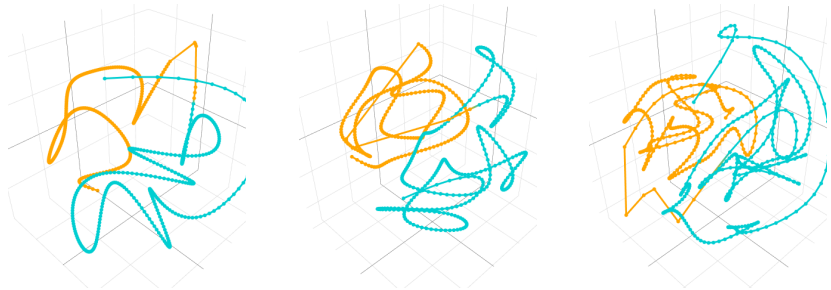


(c) $\lambda = 10$, $df \approx 26.71$ (d) $\lambda = 1$, $df \approx 47.24$ (d) $\lambda = 0.1$, $df \approx 83.44$

FIG 2. *Reconstructions obtained via the smoothing spline approach for different λ values. The solution path is produced in a sequential way: the reconstruction for larger λ is utilized as a warm start for the smaller λ. As a result, the solutions evolve in a continuous way. Colors (orange, teal) distinguish chromosome arms. Before potting the solutions were aligned via the Procrustes transformation.*



(a) $df = 5$ (b) $df = 9$ (c) $df = 15$



(d) $df = 27$ (e) $df = 47$ (f) $df = 83$

FIG 3. *Reconstructions obtained via the original PoisMS approach for different df values. The grid of df values is chosen to match the degrees-of-freedom in Figure 2. Each solution is produced by a random initialization of Θ. Random initialization for each value implies less agreement between the subsequent reconstructions than for the SPoisMS method. Before potting the solutions were aligned via the Procrustes transformation.*

9.2. *SPoisMS performance evaluation.* Now we extend the above approach to the smoothing spline technique. By analogy with (18), we can derive the *SPoisMS* training loss as

$$\ell^{train}_{SPoisMS}(\gamma,\beta;C,\lambda) = \frac{1}{|\mathcal{O}|^2}\sum_{(i,j)\in\mathcal{O}\times\mathcal{O}}\left[e^{-\|\gamma(t_i)-\gamma(t_j)\|^2+\beta} - C_{ij}\left(-\|\gamma(t_i)-\gamma(t_j)\|^2+\beta\right)\right]$$

$$+\lambda\int\|\gamma''(t)\|^2\,dt. \tag{21}$$

For this penalized loss the following lemma can be proved.

LEMMA 4. *The optimization problem*

$$\underset{\gamma\in\Gamma,\ \beta\in\mathbb{R}}{\text{minimize}}\,\ell^{train}_{SPoisMS}(\gamma,\beta;C,\lambda).$$

is equivalent to solving the reduced PoisMS problem

$$\underset{X_{\mathcal{O}}\in\mathbb{R}^{\mathcal{O}\times 3},\ \beta\in\mathbb{R}}{\text{minimize}}\,\ell_{PoisMS}\big(X_{\mathcal{O}},\beta;C_{\mathcal{O}\mathcal{O}}-\frac{\lambda|\mathcal{O}|^2}{2}(K/K_{\mathcal{U}\mathcal{U}})\big). \tag{22}$$

Here $K/K_{\mathcal{U}\mathcal{U}} = K_{\mathcal{O}\mathcal{O}} - K_{\mathcal{O}\mathcal{U}}K_{\mathcal{U}\mathcal{U}}^{-1}K_{\mathcal{U}\mathcal{O}}$ is the Schur complement of the penalty matrix, which has block structure similar to C, i.e. $K = \left(\begin{smallmatrix} K_{\mathcal{O}\mathcal{O}} & K_{\mathcal{O}\mathcal{U}} \\ K_{\mathcal{U}\mathcal{O}} & K_{\mathcal{U}\mathcal{U}}\end{smallmatrix}\right)$.

The proof is based on the properties of the smoothing spline basis (see Appendix D for the details). The lemma demonstrates that the solution to the *SPoisMS* problem can be computed by means of the *PoisMS* algorithm, even for a subset of knots. Moreover, it implies that if we treat some loci as unobserved we can re-use the full K to compute the penalty matrix for the reduced problem. In the lemma proof we also provide an explicit formula to impute the unobserved part of the reconstruction. The complete reconstruction

$$X = \begin{pmatrix} X_{\mathcal{O}} \\ -K_{\mathcal{U}\mathcal{U}}^{-1}K_{\mathcal{U}\mathcal{O}}X_{\mathcal{O}}\end{pmatrix}$$

will be subsequently plugged in (20) thereby producing the test score.

Note that the idea form Section 5 can be extended to partially observed data as well. Specifically, to reduce computations for the training fit, we can obtain an accurate approximation of (26) by obtaining the SVD of $K/K_{\mathcal{U}\mathcal{U}}$ and using a subset of the singular vectors as a basis for $X_{\mathcal{O}}$.

10. Block cross-validation. In Section 3 we raised the issue of high-correlation between the contact counts, which makes standard cross-validation approaches problematic for 3D chromatin reconstruction algorithms. To moderate the influence of correlated data on model evaluation process, we proposed the use of block cross-validation (BCV), noting difficulties encountered by the *PoisMS* algorithm. Here we investigate BCV performance for both *PoisMS* and *SPoisMS* using IMR90 cell Hi-C data.

First, we partition the $n = 599$ genomic bins into 20 folds, each containing consecutive loci. We assign one fold to $\mathcal{U}$ thereby treating 5% of the loci as unobserved; the remaining 19 folds (95% of the loci) are assigned to $\mathcal{O}$. Next, following the procedure from Section 9.1, we use $\mathcal{O}$ to compute solutions (Θ,β) via either *PoisMS* or *SPoisMS*. In our experiments, we test SPoiMS with penalty factors in the range $\lambda = 10^4, 10^3, 100, 10, 1, 0.1$. This facilitates making results compatible with *SPoisMS* by simply using the tabled degrees-of-freedom correspondence when computing the *PoisMS* solution (see Section 8 for details). Finally, to evaluate how well the solutions obtained for each method and each hyperparameter fit the unobserved

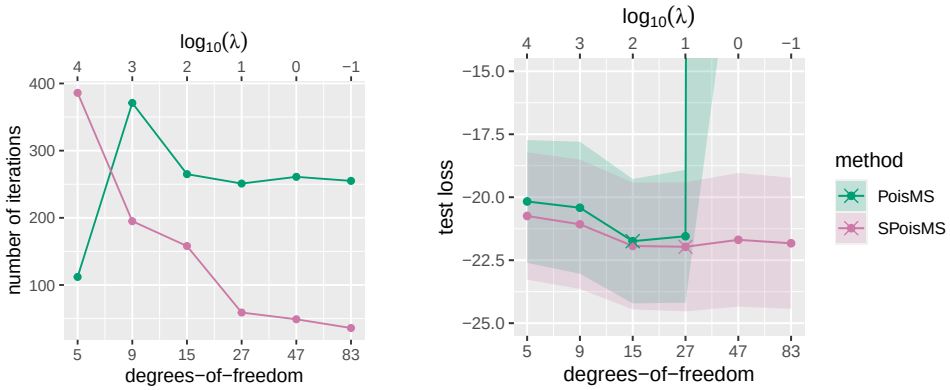


FIG 4. *Left panel: dependence between the number of iteration until the convergence and the reconstruction complexity (controlled by df in PoisMS and λ in SPoisMS). Using warm starts substantially decreases the number of iteration for the SPoisMS method for small λ. Random initialization for the PoisMS approach keeps the total number of iterations high for large degrees-of-freedom. Right panel: comparison of the block cross-validation performance for the PoisMS and SPoisMS methods. The train fit is obtained using 95% of the loci, the test score is computed on the remaining 5% of the loci. For df* $= 47$ *and* 83 *the mean PoisMS test score is so large that the confidence interval did not fit in the plot limits. The cross indicates the parameter value corresponding to the minimum score.*

set of loci $\mathcal{U}$ we impute the reconstruction and use the completed X to evaluate the test score (20). This entire procedure is repeated for all folds (excluding the two boundaries to mitigate edge effects). We present the mean as well as the 1SE intervals of the test score in the right panel of Figure 4.

From the left panel of the plot we see that, as anticipated, confidence intervals computed via *PoisMS* "explode" for high values of degrees-of-freedom. On the contrary, the *SPoisMS* test scores exhibit robust behavior regardless of the penalty factor λ. To summarise, replacing the B-spline basis of *PoisMS* with the smoothing spline (roughness penalty) approach of *SPoisMS* enables principled 3D reconstruction evaluation and degrees-of-freedom determination by facilitating block cross-validation.

11. Sparse contact matrices and over-dispersion. Contact matrices have two important attributes. First, they are exceedingly sparse, with large proportions of zero counts. This is especially true for subsequently described single cell Hi-C assays – as opposed to bulk cell experiments typically conducted using pools of $\sim 10^6$ cells. For comparison, the bulk cell data (IMR90) that we use for our experiments has 28% of zeros whereas the sparsity level of the single cell data is 99%. Second, they are diagonally dominant (Yang et al., 2017) reflecting, in part, chromatin contiguity. These factors tend to result in zero-inflation and over-dispersion with respect to assumed Poisson contact count distributions. To illustrate these concerns for the IMR90 cell data we use the 3D reconstruction X produced by the *PoisMS* approach with $df = 25$ and compute the expected mean count matrix Λ (see Section 3 for details). In Figure 5 we present the scatter plot for expected counts Λ_{ij} *vs* observed counts C_{ij}. The dense conglomeration of the points along the y-axis, as well as the substantial deviation of observed from predicted values for high contact counts highlights the presence of zero-inflation and over-dispersion.

12. Distribution-based metric scaling. As illustrated above, contact counts can violate the underlying Poisson assumptions. Indeed, the noted overdispersion has been widely documented in Hi-C data sets by Varoquaux, Noble and Vert (2021), who advance negative

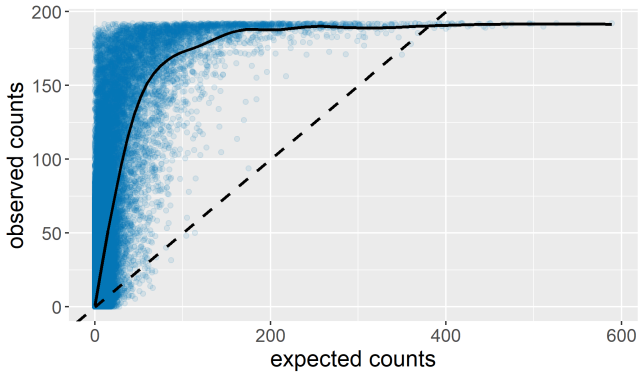


FIG 5. *Expected Λ_{ij} vs Observed C_{ij} counts where expected counts derive from a 3D reconstruction of IMR90 cell chromosome 20 obtained using PoisMS with 25 degrees-of-freedom. The dashed line depicts equality between observed and expected counts as would hold under Poisson assumptions. The solid curve is a scatter plot smooth obtained via generalized additive mode smoothing: zero-inflation of contact counts is demonstrated by the curve adhering to the expected counts axis even for moderate values thereof.*

binomial models to effect 3D reconstruction. Here we show that the *PoisMS* algorithm can be extended to a broad class of models, including the negative binomial, by developing a general approach distribution-based metric scaling (*DBMS*) which we subsequently specialize to three select chromatin 3D reconstruction techniques.

The *DBMS* model assumes that each contact count follows a discrete distribution with support $\mathbb{Z}_+$. This distribution depends on some parameters, which we subsequently link to the chromatin conformation X. We propose that the link function involve pairwise distances between loci as well as some nuisance parameters that we denote by Ω. The resulting negative log-likelihood loss has the form of $\ell_{DBMS}(X, \Omega; C)$. To solve the *DBMS* problem we minimize the loss w.r.t. unknown X and Ω, subject to the smoothing constraint $X = H\Theta$. Similarly to the *PoisMS* case, the optimization algorithm alternates between two steps.

Step 1: update the reconstruction. Since the negative log-likelihood depends only on distances we can re-write the loss as $\ell_{DBMS}(D^2(X), \Omega; C)$. As a first step, for the current reconstruction guess X, we compute the second order approximation of the loss at the point $D^2(X)$. To do so, we calculate the first and second order derivatives

$$\nabla = \nabla_{D^2} \ell_{DBMS}(D^2(X), \Omega; C) \text{ and } \nabla^2 = \nabla^2_{D^2} \ell_{DBMS}(D^2(X), \Omega; C)$$

and evaluate the weight and working response matrices

$$W = \nabla^2 \text{ and } Z = D^2(X) - \frac{\nabla}{\nabla^2}. \tag{23}$$

This leads to the second order approximation

$$\ell_{DBMS}(D^2(X), \Omega; C) \approx \|\sqrt{W} * (Z - D^2(X))\|_F^2. \tag{24}$$

Now, instead of optimizing the original loss, we minimize its second order approximation subject to the smoothing constraint $X = H\Theta$. That is we solve the problem

$$\underset{\Theta \in \mathbb{R}^{k \times 3}}{\text{minimize}} \|\sqrt{W} * (Z - D^2(H\Theta))\|_F^2,$$

which can be accomplished using *WPCMS*. As a result, we update Θ and the reconstruction $X = H\Theta$.

Step 2: update the nuisance parameters. We fix X and optimize the loss $\ell_{DBMS}(D^2(X), \Omega; C)$ w.r.t. Ω. Depending on the distribution, we will use one of the following approaches to update the nuisance parameters:

- compute the first order derivative of the loss w.r.t. Ω and use it to find an explicit formula for the optimal value;
- compute the first and second order derivatives of the loss w.r.t. Ω and use them to run Newton's algorithm.

13. Alternative contact count distributions. We now describe three alternative distributions for contact counts: hurdle Poisson, zero-inflated Poisson, and negative binomial. The optimization algorithm for each is readily obtained using *DBMS* as outlined above and detailed in the Appendix.

13.1. *Hurdle Poisson.* The hurdle Poisson model (Gurmu, 1998) has two components: (i) zero counts follow a Bernoulli distribution; and (ii) non-zero counts follow a zero-truncated Poisson (ZTP) distribution. Thus

$$C_{ij} = 0 \text{ with probability } \pi,$$
$$C_{ij} | C_{ij} > 0 \sim \text{ZTP}(\lambda_{ij})$$

which leads to the following distribution for C_{ij}

$$P(C_{ij} = c) = \begin{cases} \pi, \text{ if } c = 0 \\ \frac{1-\pi}{1-e^{-\lambda_{ij}}} \frac{\lambda_{ij}^c e^{-\lambda_{ij}}}{c!}, \text{ if } c > 0 \end{cases}$$

We again use the log-link (2) to introduce dependence between truncated Poisson parameters λ_{ij} and chromatin 3D spacial structure. If $\mathcal{N} = \{(i,j) : C_{ij} = 0\}$ represents the subset of zero contact counts, then the goal of the Hurdle Poisson metric scaling (*HPoisMS*) is to minimize the negative log-likelihood

$$\ell_{\textit{HPoisMS}}(X, \beta, \pi; C) = \frac{1}{n^2} \Big[- \sum_{(i,j) \in \mathcal{N}} \log(\pi) - \sum_{(i,j) \notin \mathcal{N}} \log(1-\pi) +$$
$$+ \sum_{(i,j) \notin \mathcal{N}} \left[e^{-\|x_i - x_j\|^2 + \beta} - C_{ij} \left(-\|x_i - x_j\|^2 + \beta \right) + \log \left(1 - e^{-e^{-\|x_i - x_j\|^2 + \beta}} \right) \right] \Big]$$

The optimal solution can be found via the *HPoisMS* algorithm described in Appendix E.1.

13.2. *Zero-inflated Poison.* The zero-inflated Poisson distribution (Lambert, 1992) addresses excess zero counts by assuming that zero count observations have two different sources: "structural" and "sampling". Structural zeros occur with some fixed probability π, while sampling zeros correspond to a null Poisson outcome

$$C_{ij} = 0 \text{ with probability } \pi$$
$$C_{ij} \sim \text{Pois}(\lambda_{ij}) \text{ with probability } 1 - \pi.$$

This yields the contact count distribution

$$P(C_{ij} = c) = \begin{cases} \pi + (1-\pi) e^{-\lambda_{ij}}, \text{ if } c = 0 \\ (1-\pi) \frac{\lambda_{ij}^c e^{-\lambda_{ij}}}{c!}, \text{ if } c > 0 \end{cases}$$

which, when combined with the same log-link (2), produces the loss function

$$\ell_{\textit{ZIPoisMS}}(X, \beta, \pi; C) = \frac{1}{n^2} \Big[- \sum_{(i,j) \in \mathcal{N}} \log \left(\pi + (1-\pi) e^{-e^{-\|x_i - x_j\|^2 + \beta}} \right) -$$
$$- \sum_{(i,j) \notin \mathcal{N}} \log(1-\pi) + \sum_{(i,j) \notin \mathcal{N}} \left[e^{-\|x_i - x_j\|^2 + \beta} - C_{ij} \left(-\|x_i - x_j\|^2 + \beta \right) \right] \Big]$$

Zero-inflated Poisson metric scaling (*ZIPoisMS*), an optimization procedure that finds the optimal X, β nd π, is presented in Appendix E.2.

13.3. *Negative binomial.* The negative binomial distribution models contact counts as

$$P(C_{ij} = c) = \frac{\Gamma(c+r)}{\Gamma(c+1)\Gamma(r)} \frac{\lambda_{ij}^c r^r}{(\lambda_{ij}+r)^{c+r}}.$$

While the Poisson distribution implies $\mathrm{Var}(C_{ij}) = \mathrm{E}(C_{ij})$, the negative binomial mean - variance relationship is $\mathrm{Var}(C_{ij}) = \mathrm{E}(C_{ij}) + \frac{\mathrm{E}(C_{ij})^2}{r}$, with the nuisance parameter r enabling the model to capture contact count overdispersion.

By linking the parameter λ_{ij} to the pairwise distances $\|x_i - x_j\|$ through equation (2) we arrive at our negative binomial metric scaling (*NBMS*) criterion: find optimal X, β, r that minimize the negative log-likelihood

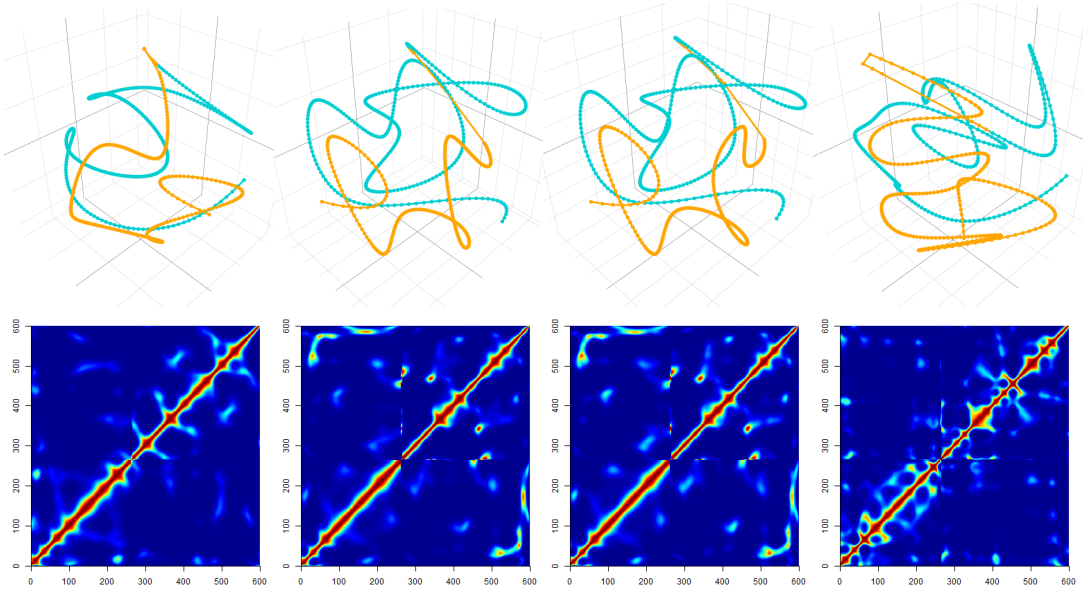
$$\ell_{NBMS}(X, \beta, r; C) = \frac{1}{n^2}\Big[\sum_{1 \le i,j \le n} \log\Gamma(r) - \log\Gamma(C_{ij}+r) - r\log r + \\ \sum_{1 \le i,j \le n} (C_{ij}+r)\log(e^{-\|x_i - x_j\|^2 + \beta} + r) - C_{ij}(-\|x_i - x_j\|^2 + \beta)\Big].$$

Algorithm details are provided in Appendix E.3.

14. Discrete distribution model comparison. We explore modeling contact counts by means of the four distributions: Poisson, Hurdle Poisson, zero-inflated Poisson and negative binomial, starting with simple visual comparisons. We use IMR90 chromosome 20 data and we train the four models for a grid of degrees-of-freedom values $df = 10, 20 \ldots, 100$ using a B-spline basis. Note that it is also possible to combine the two proposed advances, i.e. distribution-based approach with the smoothing spline techniques, which we discuss in Appendix F. In the left panel of Figure 7 we present the dependence of the train score on the model complexity. For each model we pick the "optimal" degrees-of-freedom using the elbow heuristic (Tuzhilina, Hastie and Segal, 2020) and obtain the corresponding 3D chromatin reconstructions. We plot these reconstruction as well as the heatmaps of $\Lambda(X) = e^{-D^2(X)+\beta}$ in Figure 6. Comparing the images, we can conclude that *HPoisMS* and *ZIPoisMS* produce quite similar results. At the same time, both of these models encourage more interaction between before and after centromere parts of the reconstruction than *PoisMS* and *NBMS*; however, the negative binomial model still demonstrates more interaction within these two blocks.

As indicated, concerns surrounding excess zeroes are considerably amplified when performing 3D chromatin configuration reconstruction using exceedingly sparse single-cell Hi-C data, in contrast with still sparse data from bulk cell experiments. Accordingly, to further compare our suite of *DBMS* models, we use contact matrices for chromosome 1 from eight single mouse embryonic stem cells (Stevens et al., 2017, Gene Expression Ominbus (GEO) repository GSE80280) denoted $C^{(1)}, \ldots, C^{(8)}$. Using data at 100kb resolution results in $n = 1924$ genomic loci. We run the following experiment. We select at random a subset of four indices $\mathcal{T} \in \{1, \ldots, 8\}$ and calculate train and test contact matrices as $C_{train} = \sum_{i \in \mathcal{T}} C^{(i)}$ and $C_{test} = \sum_{i \notin \mathcal{T}} C^{(i)}$. We use C_{train} to fit a model, thereby obtaining the reconstruction X and the optimal nuisance parameters. We subsequently use C_{test} to evaluate the model's test score represented by the negative log-likelihood value. We repeat this experiment $N = 30$ times and calculate the average test score (across the random splits) as well as 90% confidence interval.

In Figure 7 we plot the dependence of test performance on degrees-of-freedom value for each of the four models. Comparing the *PoisMS* and *NBMS* curves (blue and green), we



(a) Poisson (b) Hurdle Poisson (c) Zero-inflated Poisson (d) Negative binomial

FIG 6. *Upper row: the projections of the resulting reconstructions obtained via four methods: PoisMS, HPoisMS, ZIPoisMS, NBMS. The degrees-of-freedom is set to the value picked by the elbow method. Bottow row: the heatmaps for* $\Lambda = e^{-D^2(X)+\beta}$.

conclude that using negative binomial distribution for contact counts in place of Poisson does not enhance the conformation reconstruction. On the other hand, we observe a substantial improvement in the test score for the other two models, where the best log-likelihood value is attained by *ZIPoisMS* with $df = 15$. The extreme sparsity of the contact matrix can explain such a difference in performance: the data has around 99% of zero contact counts, thus, it strongly benefits from zero-inflated models.

15. Discussion. In this paper we propose several improvements to the *PoisMS* methodology described in Tuzhilina, Hastie and Segal (2020). First, we suggest an alternative way to encourage the reconstruction smoothness. Specifically, instead of using constraint (1), we combine the *PoisMS* objective with the roughness penalty. The resulting *SPoisMS* optimization problem (7) is a blend of *PoisMS* and smoothing splines, and, as we prove in Section 4, it has the following nice property: the reconstruction can be found by means of the original *PoisMS* algorithm. The motivation for such an alternative smoothing technique was the incompatibility of B-splines with block cross-validation and the inability to re-use the solution for smaller degrees-of-freedom as a warm start for the larger ones. We demonstrate the advantages of *SPoisMS* via IMR90 data set the in Section 10.

Next, we extend *PoisMS* in a different direction. We propose three alternative models for the contact counts: zero-inflated and Hurdle Poisson as well as Negative Binomial. These distributions were motivated by various artifacts present in the Hi-C data such as sparsity or diagonal dominance of the contact matrix. In section 12 we introduce a general framework that allows us to build optimization algorithm for a wide class of distributions. We subsequently use it for developing the *ZIPoisMS*, *HPoisMS* and *NBMS* techniques, which we finally compare via the single cell data.

There is still much scope for future experiments. First, we plan to test the novel techniques and validate our findings on data sets involving other chromosomes and resolutions. From

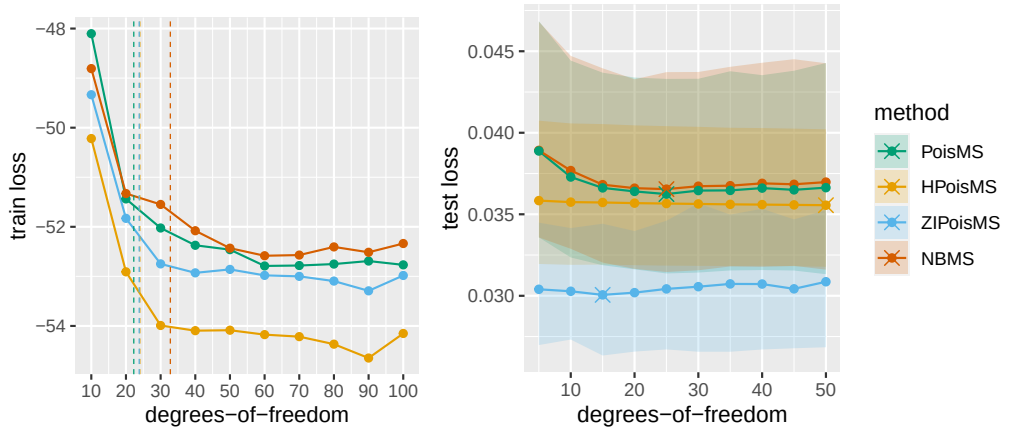


FIG 7. *Left panel: comparing four models via the bulk cell data. Each model was trained on the full contact matrix and the train score is reported. The dashed lines represent the optimal degrees-of-freedom values detected for each model by the elbow method. Right panel: comparing four models via the single cell data. Each model was fitted on* C_{train} *and subsequently evaluated on* C_{test}*. The plot represents the average (across different train-test splits) test score vs degrees-of-freedom; the intervals correspond to the 90% CI bands of the test scores. The plot evidences the superior performance of the ZIPoisMS models to the competitors.*

a methodological point of view, we aim to enrich the class of models. In particular, we will explore Hurdle Negative Binomial distribution, which would simultaneously address sparsity and overdispersion. Moreover, we can refine the methods proposed in this paper by linking the nuisance parameters to the chromatin spatial structure. As an example, in the Hurdle model one can replace the common parameter π by π_{ij}, different for each pair of loci, which we will link to the pair-wise distances via the logit link $\log(\frac{\pi_{ij}}{1-\pi_{ij}}) = aD^2(X) + b$.

Another interesting and more practical research direction would be to develop a more efficient implementation of the proposed algorithms. For instance, various acceleration techniques such as Nesterov and Anderson acceleration are known to perform well when combined with the projected gradient descent (see, for example, Tuzhilina and Hastie, 2021). Thus, these techniques can significantly speed up the convergence of the main building block *WPCMS* thereby enabling much faster computations for the *DBMS* solutions. Finally, the sparsity of the contact matrix can be an important feature used for improving the storage and computation time.

Importance of accurate 3D reconstructions from single-cell Hi-C data: allow dissection of inter-cell structural variation unclouded by reliance of consensus reconstructions as derived from bulk cell Hi-C experiments. Further, large numbers ($\sim 10^3$) of single cells, assayed in differing conditions (e.g. cell cycle phase) and for differing cell types (Ramani et al., 2017), allow a variety of compelling questions to be addressed. However, such analyses require means for capturing and quantifying structural differences, ideally more refined than global mean square error following Procrustes alignment. We have commenced work pursuing these objectives via two distinct approaches. First, we are devising local Procrustes alignment techniques, effected by iteratively using kernel weighting schemes along a current alignment. Second, when provided with an alignment, either global or local, between two 3D chromosome configurations and extracted per-locus between-structure residuals therefrom, we have advanced use of the patient rule induction method (PRIM, Hastie, Tibshirani and Friedman, 2009) to identify maximally divergent subregions (Segal, 2021). For any of these methods to have merit, the input 3D structures need to be good approximations, underscoring the potential of our *DBMS* approach to single-cell 3D reconstruction.

Funding. E.T. was supported by Stanford Data Science scholarship. T.H. was partially supported by grants DMS-1407548 and IIS 1837931 from the National Science Foundation, and grant 5R01 EB 001988-21 from the National Institutes of Health. M.S. was partially supported by grant GM-109457 from the National Institutes of Health.

SUPPLEMENTARY MATERIAL

Code and Data

Proposed methods and data sets are published in the R package `DBMS`; the software is available from Github (https://github.com/ElenaTuzhilina/DBMS).

REFERENCES

AY, F., BUNNIK, E. M., VAROQUAUX, N., BOL, S. M., PRUDHOMME, J., VERT, J. P., NOBLE, W. S. and LE ROCH, K. G. (2014). Three-dimensional modeling of the P. falciparum genome during the erythrocytic cycle reveals a strong connection between genome architecture and gene expression. *Genome Research* **24** 974-88.

BELYAEVA, A., KUBJAS, K., SUN, L. J. and UHLER, C. (2021). Identifying 3D genome organization in diploid organims via Euclidean distance geometry.

BENTBIB, A. H. and KANBER, A. (2017). Block Power Method for SVD Decomposition. *Analele Universitatii "Ovidius" Constanta - Seria Matematica* **23** 45–58.

CAPURSO, D., BENGTSSON, H. and SEGAL, M. R. (2016). Discovering hotspots in functional genomic data superposed on 3D chromatin configuration reconstructions. *Nucleic Acids Research* **44** 2028-2035.

CAUER, A. G., YARDIMCI, G., VERT, J.-P., VAROQUAUX, N. and NOBLE, W. S. (2019). Inferring Diploid 3D Chromatin Structures from Hi-C Data. *19th International Workshop on Algorithms in Bioinformatics (WABI 2019)* **143** 11:1-11:13.

DIXON, J. R., SELVARAJ, S., YUE, F., KIM, A., LI, Y., SHEN, Y., HU, M., LIU, J. S. and REN, B. (2012). Topological domains in mammalian genomes identified by analysis of chromatin contacts. *Nature* **485** 376-380.

DUAN, Z., ANDRONESCU, M., SCHUTZ, K., MCILWAIN, S., KIM, Y. J., LEE, C., SHENDURE, J., FIELDS, S., BLAU, C. A. and NOBLE, W. S. (2010). A three-dimensional model of the yeast genome. *Nature* **465** 363-367.

GREEN, P. J. and SILVERMAN, B. W. (1994). *Nonparametric regression and generalized linear models: a roughness penalty approach*. Chapman and Hall, United Kingdom.

GURMU, S. (1998). Generalized hurdle count data regression models. *Economics Letters* **58** 263-268.

HASTIE, T. J. and STUETZLE, W. (1989). Principal curves. *Journal of the American Statistical Association* **406** 502-516.

HASTIE, T. J., TIBSHIRANI, R. J. and FRIEDMAN, J. H. (2009). *The Elements of Statistical Learning*. Springer, New York.

LAMBERT, D. (1992). Zero-Inflated Poisson Regression, with an Application to Defects in Manufacturing. *Technometrics* **34** 1–14.

LEE, C. S., WANG, R. W., CHANG, H. H., CAPURSO, D., SEGAL, M. R. and HABER, J. E. (2016). Chromosome position determines the success of double-strand break repair. *Proceedings of the National Academy of Science* **113** 146-154.

LIEBERMAN-AIDEN, E., VAN BERKUM, N. L., WILLIAMS, L., IMAKAEV, M., RAGOCZY, T., TELLING, A., AMIT, I., LAJOIE, B. R., SABO, P. J., DORSCHNER, M. O., SANDSTROM, R., BERNSTEIN, B., BENDER, M. A., GROUDINE, M., GNIRKE, A., STAMATOYANNOPOULOS, J., MIRNY, L. A., LANDER, E. S. and DEKKER, J. (2009). Comprehensive mapping of long-range contacts reveals folding principles of the human genome. *Science* **326** 289-293.

LUO, H., LI, X., FU, H. and PENG, C. (2020). HiCHap: a package to correct and analyze the diploid Hi-C data. *BMC Genomics* **21** 746.

MARCO, A., MEHARENA, H. S., DILEEP, V., RAJU, R. M., DAVILA-VELDERRAIN, J., ZHANG, A. L., ADAIKKAN, C., YOUNG, J. Z., GAO, F., KELLIS, M. and TSAI, L. H. (2020). Mapping the epigenomic and transcriptomic interplay during memory formation and recall in the hippocampal engram ensemble. *Nature Neuroscience* **23** 1606–1617.

OLUWADARE, O., HIGHSMITH, M. and CHENG, J. (2019). An overview of methods for reconstructing 3-D chromosome and genome structures from Hi-C data. *Biological Procedures Online* **21** 1-20.

PARK, J. and LIN, S. (2017). A random effect model for reconstruction of spatial chromatin structure. *Biometrics* **73** 52-62.

PAYNE, A. C., CHIANG, Z. D., REGINATO, P. L., MANGIAMELI, S. M., MURRAY, E. M., YAO, C.-C., MARKOULAKI, S., EARL, A. S., LABADE, A. S., JAENISCH, R., CHURCH, G. M., BOYDEN, E. S., BUENROSTRO, J. D. and CHEN, F. (2021). In situ genome sequencing resolves DNA sequence and structure in intact biological samples. *Science* **371** eaay3446.

RAMANI, V., DENG, X., GUNDERSON, K. L., STEEMERS, F. J., DISTECHE, C. M., NOBLE, W. S., DUAN, Z. and SHENDURE, J. (2017). Massively multiplex single-cell Hi-C. *Nature Methods* **14** 263-266.

RAO, S. S., HUNTLEY, M. H., DURAND, N. C., STAMENOVA, E. K., BOCHKOV, I. D., ROBINSON, J. T., SANBORN, A. L., MACHOL, I., OMER, A. D., LANDER, E. S. and AIDEN, E. L. (2014). A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell* **159** 1665-1680.

RIEBER, L. and MAHONY, S. (2017). miniMDS: 3D structural inference from high-resolution Hi-C data. *Bioinformatics* **33** 261-266.

ROSENTHAL, M., BRYNER, D., HUFFER, F., EVANS, S., SRIVASTAVA, A. and NERETTI, N. (2019). Bayesian Estimation of 3D Chromosomal Structure from Single Cell Hi-C Data. *Journal of Computational Biology* **26** 1191–1202.

SEGAL, M. R. (2021). Assessing chromatin relocalization in 3D using the patient rule induction method. https://doi.org/10.1093/biostatistics/kxab033.

STEVENS, T. J., LANDO, D., BASU, S., ATKINSON, L. P., CAO, Y., LEE, S. F., LEEB, M., WOHLFAHRT, K. J., BOUCHER, W., O'SHAUGHNESSY-KIRWAN, A., CRAMARD, J., FAURE, A. J., RALSER, M., BLANCO, E., MOREY, L., SANSO, M., PALAYRET, M. G. S., LEHNER, B., DI CROCE, L., WUTZ, A., HENDRICH, B., KLENERMAN, D. and LAUE, E. D. (2017). 3D structures of individual mammalian genomes studied by single-cell Hi-C. *Nature* **544** 59-64.

TUZHILINA, E., HASTIE, T. J. and SEGAL, M. R. (2020). Principal curve approaches for inferring 3D chromatin architecture. *Biostatistics*.

TUZHILINA, E. and HASTIE, T. (2021). Weighted Low Rank Matrix Approximation and Acceleration.

VAROQUAUX, N., NOBLE, W. S. and VERT, J. P. (2021). Inference of genome 3D architecture by modeling overdispersion of Hi-C data. https://www.biorxiv.org/content/10.1101/2021.02.04.429864v1.

WAHBA, G. (1990). Spline Models for Observational Data. *Regional Conference Series in Applied Mathematics* **59**.

YANG, T., ZHANG, F., YARDIMCI, G. G., SONG, F., HARDISON, R. C., NOBLE, W. S., YUE, F. and LI, Q. (2017). HiCRep: assessing the reproducibility of Hi-C data using a stratum- adjusted correlation coefficient. *Genome Research* **27** 1939-1949.

ZHANG, Z., LI, G., C., T. K. and SUNG, W. K. (2013). 3D chromosome modeling with semi-definite programming and Hi-C data. *Journal of Computational Biology* **20** 831-846.

APPENDIX A: PROPERTY OF THE DEMMLER-REINSCH PENALTY MATRIX

LEMMA 1. *If $\mathbf{1} = (1, \dots, 1)^\top \in \mathbb{R}^n$ is n-dimensional vector of ones and K is the smoothing spline penalty matrix, then*

- $K\mathbf{1} = 0$;
- $\mathbf{1}^\top K = 0$;
- $\sum_{1 \le i,j, \le n} K_{ij} = \mathbf{1}^\top K \mathbf{1} = 0$.

PROOF. Denote $K = UDU^\top$ the eigendecomposition of the penalty matrix. One can show that K has two zero eigenvalues and the corresponding eigenvectors span the subspace of linear functions (see, for example, Green and Silverman, 1994). For simplicity, we assume that the eigenvalues are sorted in increasing order in D, so $d_1 = d_2 = 0$ and matrix U has the following structure: $U = (U_0, U_0^\perp)$, where $U_0 \in \mathbb{R}^{n \times 2}$ corresponds to the null-space of K and $U_0^\perp \in \mathbb{R}^{n \times (n-2)}$ is the orthogonal space. Therefore, if $g(t)$ is some linear function of t and $G = (g(t_1), \dots, g(t_n))^\top$, we get $G \in span(U_0)$ as well as $(U_0^\perp)^\top G = 0$. This leads us to the relation

$$KG = UD \begin{pmatrix} U_0^\top G \\ (U_0^\perp)^\top G \end{pmatrix} = U \begin{pmatrix} \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix} U_0^\top G \\ 0 \end{pmatrix} = 0.$$

In particular, setting $g(t) = 1$ implies $KG = K\mathbf{1} = 0$. The second relation $\mathbf{1}^\top K = 0$ automatically follows form the fact that K is positive semi-definite (PSD), which immediately implies $\sum_{1 \le i,j, \le n} K_{ij} = \mathbf{1}^\top K \mathbf{1} = 0$. □

APPENDIX B: DEGREES-OF-FREEDOM

LEMMA 2. *The projection step of SPoisMS solves the following penalized PCMS problem for some $\tilde{S}$ independent of λ:*

$$\underset{X \in \mathbb{R}^{n \times 3}}{\text{minimize}} \frac{1}{n^2} \|\tilde{S} - XX^\top\|_F^2 + \lambda \operatorname{tr}(X^\top K X).$$

PROOF. Recall, that *SPoisMS* is equivalent to *PoisMS* with C replaced by

$$C^\lambda = C - \frac{\lambda n^2}{2} K.$$

Suppose that the current reconstruction guess at the beginning of a new *WPCMS* loop is X_0. Thus, the working response in the *WPCMS* problem is $Z^\lambda = D^2(X_0) - \frac{C^\lambda}{W} + 1$.

Next, each gradient step within the *WPCMS* loop updates the similarity matrix as

$$S = XX^\top - \Phi(W * (Z^\lambda - D^2(X))) =$$

$$= \underbrace{XX^\top - \Phi(W * (Z - D^2(X)))}_{\tilde{S}} + \frac{\lambda n^2}{2} K.$$

Here we use the linearity of the operator $\Phi(\cdot)$ and Lemma 1 implying

$$\Phi(K) = K - \operatorname{diag}(K \cdot 1) = K.$$

As a result, the projection step will minimize the *PCMS* loss as

$$\|S - XX^\top\|_F^2 = \|\tilde{S} + \frac{\lambda n^2}{2} K - XX^\top\|_F^2 =$$

$$= \|\tilde{S} - XX^\top\|_F^2 - \lambda n^2 \operatorname{tr}(X^\top K X) + \frac{\lambda^2 n^4}{4} \|K\|_F^2.$$

Removing the terms independent of X and dividing by n^2 this leads us to the optimization problem from the lemma. □

LEMMA 3. *The degrees-of-freedom for one-dimensional penalized PCMS*

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\,\frac{1}{n^2}\|\tilde{S}-xx^\top\|_F^2+\lambda x^\top Kx$$

can be calculated as

$$df=\sum_{i=1}^{n}\frac{1}{1+\frac{\lambda n^2}{\|x_0\|^2}d_i},$$

where $x_0\in\mathbb{R}^n$ is the solution to the problem.

PROOF. Again, note that the one-dimensional problem can be solved in two ways. One can rewrite it as

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\,\big\|\big(\tilde{S}-\frac{\lambda n^2}{2}K\big)-xx^\top\big\|_F^2$$

and find x via the eigendecomposition of $\tilde{S}-\frac{\lambda n^2}{2}K$. Alternatively, one can apply alternating algorithm. If the current solution guess is x_0 then updated x can be found via solving a spline problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\,\frac{1}{n^2}\|\tilde{S}-xx_0^\top\|_F^2+\lambda x^\top Kx\iff$$

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\,\frac{1}{n^2}\|y-x\|_F^2+\frac{\lambda}{\|x_0\|^2}\,x^\top Kx. \tag{25}$$

Here we set $y=\tilde{S}\cdot\frac{x_0}{\|x_0\|}$ and reparameterize the problem as $x:=\|x_0\|\cdot x$. The second approach implies that at the convergence point we can replace the original *PCMS* problem by (25). Thus, we can borrow the definition of degrees-of-freedom from regular smoothing splines to compute *df* for one-dimensional penalized *PCMS*. The last step is to plug-in the corresponding values to formula (14).

□

APPENDIX C: *POISMS* AND *SPOISMS* RECONSTRUCTION COMPARISON

To understand how well formula (17) aligns our *SPoisMS* and *PoisMS* reconstructions from Section 8 we introduce the notion of *discrete curvature* as follows. For a 3D conformation $X=\begin{pmatrix}-x_1^\top-\\ \cdots\\ -x_n^\top-\end{pmatrix}$ we calculate directed edges $\vec{e}_i=\vec{x}_{i+1}-\vec{x}_i$ and compute the *average angle deficit* as

$$\kappa(X)=\frac{1}{n-2}\sum_{i=1}^{n-2}\angle(\vec{e}_i;\vec{e}_{i+1}).$$

This quantity measures how wiggly the reconstruction X is, and can serve as an approximation to the formal curvature for the corresponding smooth one-dimensional curve γ.

Now we take the *SPoisMS* reconstructions calculated for the grid of penalty factors

$$\lambda=10^4,10^3,100,10,1,0.1$$

and the accordant *PoisMS* reconstructions with

$$df=5,9,15,27,47,83.$$

We plot the average angle deficit for each reconstruction in Figure 8. As expected, the reconstruction wiggliness increases with the growth of the model complexity (increase in *df* or decrease in λ). Moreover, the plot suggests that the corresponding *SPoisMS* and *PoisMS* reconstructions have similar curvature.

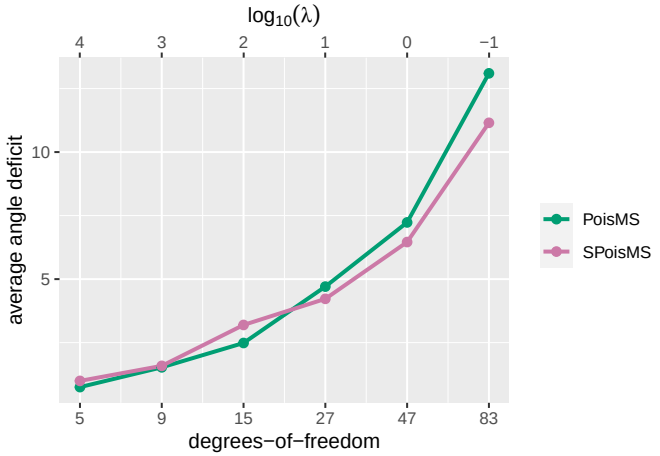


FIG 8. *Dependence between the average angle deficit (measuring how wiggly is a reconstruction) and the model complexity (controlled by df in PoisMS and λ in SPoisMS). The plot demonstrates that the corresponding SPoisMS and PoisMS reconstructions have similar curvature, which implies the accuracy of the proposed formula for estimating degrees-of-freedom.*

APPENDIX D: *SPOISMS* PERFORMANCE EVALUATION

LEMMA 4. *The optimization problem*

$$\underset{\gamma\in\Gamma,\ \beta\in\mathbb{R}}{\text{minimize}}\, \ell^{train}_{SPoisMS}(\gamma,\beta;C,\lambda).$$

is equivalent to solving the reduced PoisMS problem

$$\underset{X_{\mathcal{O}}\in\mathbb{R}^{n\times 3},\ \beta\in\mathbb{R}}{\text{minimize}}\, \ell_{PoisMS}\big(X_{\mathcal{O}},\beta;C_{\mathcal{O}\mathcal{O}}-\frac{\lambda|\mathcal{O}|^2}{2}(K/K_{\mathcal{U}\mathcal{U}})\big). \tag{26}$$

Here $K/K_{\mathcal{U}\mathcal{U}}=K_{\mathcal{O}\mathcal{O}}-K_{\mathcal{O}\mathcal{U}}K_{\mathcal{U}\mathcal{U}}^{-1}K_{\mathcal{U}\mathcal{O}}$ is the Schur complement of the penalty matrix, which has block structure similar to C, i.e. $K=\left(\begin{smallmatrix}K_{\mathcal{O}\mathcal{O}} & K_{\mathcal{O}\mathcal{U}}\\ K_{\mathcal{U}\mathcal{O}} & K_{\mathcal{U}\mathcal{U}}\end{smallmatrix}\right)$.

PROOF. First we note that Denote the solution by $X=\left(\begin{smallmatrix}X_{\mathcal{O}}\\ X_{\mathcal{U}}\end{smallmatrix}\right)$. We begin with rewriting the loss function in matrix form as

$$\ell^{train}_{SPoisMS}(X,\beta;C,K,\lambda)=\ell_{PoisMS}(X_{\mathcal{O}},\beta;C_{\mathcal{O}\mathcal{O}})+\lambda\,\mathrm{tr}(X^\top KX).$$

Next, we notice that the first part of the loss does not depend on $X_{\mathcal{U}}$. If $X_{\mathcal{O}}$ is fixed then $X_{\mathcal{U}}$ can be found as a minimum of

$$\mathrm{tr}(X^\top KX)=\mathrm{tr}(X_{\mathcal{O}}^\top K_{\mathcal{O}\mathcal{O}}X_{\mathcal{O}}+2X_{\mathcal{U}}^\top K_{\mathcal{U}\mathcal{O}}X_{\mathcal{O}}+X_{\mathcal{U}}^\top K_{\mathcal{U}\mathcal{U}}X_{\mathcal{U}}).$$

Taking the derivative w.r.t. $X_{\mathcal{U}}$ and setting it to zero leads us to the stationary point $X_{\mathcal{U}}=-K_{\mathcal{U}\mathcal{U}}^{-1}K_{\mathcal{U}\mathcal{O}}X_{\mathcal{O}}$. Plugging it back to the original loss function implies

$$\begin{aligned}
&\ell^{train}_{SPoisMS}(X,\beta;C,K,\lambda)=\\
&=\ell_{PoisMS}(X_{\mathcal{O}},\beta;C_{\mathcal{O}\mathcal{O}})+\lambda\,\mathrm{tr}(X_{\mathcal{O}}^\top K_{\mathcal{O}\mathcal{O}}X_{\mathcal{O}}+2X_{\mathcal{U}}^\top K_{\mathcal{U}\mathcal{O}}X_{\mathcal{O}}+X_{\mathcal{U}}^\top K_{\mathcal{U}\mathcal{U}}X_{\mathcal{U}})=\\
&=\ell_{PoisMS}(X_{\mathcal{O}},\beta;C_{\mathcal{O}\mathcal{O}})+\lambda\,\mathrm{tr}\left(X_{\mathcal{O}}^\top(K_{\mathcal{O}\mathcal{O}}-K_{\mathcal{O}\mathcal{U}}K_{\mathcal{U}\mathcal{U}}^{-1}K_{\mathcal{U}\mathcal{O}})X_{\mathcal{O}}\right)=\\
&=\ell_{SPoisMS}(X_{\mathcal{O}},\beta;C_{\mathcal{O}\mathcal{O}},K/K_{\mathcal{U}\mathcal{U}},\lambda).
\end{aligned}$$

Therefore, the optimal $X_{\mathcal{O}}$ can be found by solving the *SPoisMS* problem for the contact matrix $C_{\mathcal{OO}}$ and the penalty matrix $K/K_{\mathcal{UU}}$, i.e.

$$\underset{X_{\mathcal{O}}\in\mathbb{R}^{|\mathcal{O}|\times 3},\ \beta\in\mathbb{R}}{\text{minimize}}\ \ell_{\textit{SPoisMS}}(X_{\mathcal{O}},\beta;C_{\mathcal{OO}},K/K_{\mathcal{UU}},\lambda). \tag{27}$$

Note that in the main paper we demonstrated that the *SPoisMS* problem with the Demmler-Reinsch penalty matrix can be efficiently solved via the *PoisMS* algorithm. We extend this result to the optimization (27) as well. From the penalty matrix property $K\mathbf{1}=0$ one can derive the system on equations involving the blocks of K, i.e. $\begin{cases} K_{\mathcal{OO}}\mathbf{1}=-K_{\mathcal{OU}}\mathbf{1} \\ K_{\mathcal{UO}}\mathbf{1}=-K_{\mathcal{UU}}\mathbf{1} \end{cases}$, leading us to analogous property of the Schur complement

$$(K/K_{\mathcal{UU}})\mathbf{1}=K_{\mathcal{OO}}\mathbf{1}-K_{\mathcal{OU}}K_{\mathcal{UU}}^{-1}K_{\mathcal{UO}}\mathbf{1}=K_{\mathcal{OO}}\mathbf{1}+K_{\mathcal{OU}}\mathbf{1}=0.$$

Following the proof in the main paper, one can show that the smoothing penalty in the loss function $\ell_{\textit{SPoisMS}}(X_{\mathcal{O}},\beta;C_{\mathcal{OO}},K/K_{\mathcal{UU}},\lambda)$ can be absorbed into the *PoisMS* part thereby proving the equivalence between (27) and the *PoisMS* problem

$$\underset{X_{\mathcal{O}}\in\mathbb{R}^{|\mathcal{O}|\times 3},\ \beta\in\mathbb{R}}{\text{minimize}}\ \ell_{\textit{PoisMS}}\big(X_{\mathcal{O}},\beta;C_{\mathcal{OO}}-\frac{\lambda}{2}(K/K_{\mathcal{UU}})\big). \tag{28}$$

□

APPENDIX E: DISTRIBUTION BASED METRIC SCALING

Denote Γ the matrix with element $\Gamma_{ij}=\mathbb{1}_{C_{ij}=0}$ and note that the derivatives of the log-link $\Lambda=e^{-D^2(X)+\beta}$ w.r.t. $D^2(X)$ and β are

$$\nabla_{D^2}\Lambda=-\Lambda \text{ and } \nabla_{\beta}\Lambda=\Lambda.$$

We will use these relations to derive the update steps for the optimization algorithm.

E.1. Hurdle Poisson. Recall that the loss function has the following form

$$\ell_{\textit{HPoisMS}}(X,\beta,\pi;C)=\frac{1}{n^2}\Big[-\sum_{(i,j)\in\mathcal{N}}\log(\pi)-\sum_{(i,j)\notin\mathcal{N}}\log(1-\pi)+ \\ +\sum_{(i,j)\notin\mathcal{N}}\Big[\lambda_{ij}-C_{ij}\log\lambda_{ij}+\log\left(1-e^{-\lambda_{ij}}\right)\Big]\Big].$$

We first compute the derivatives of the loss w.r.t. D^2. Note that

$$\nabla_{D^2}e^{-\Lambda}=e^{-\Lambda}*\Lambda \text{ and } \nabla_{\beta}e^{-\Lambda}=-e^{-\Lambda}*\Lambda.$$

Denote $M=\frac{\Lambda}{1-e^{-\Lambda}}$, then (up to $\frac{1}{n^2}$ scale)

$$\nabla_{D^2}=(1-\Gamma)*\left(-\Lambda+C-\frac{e^{-\Lambda}*\Lambda}{1-e^{-\Lambda}}\right)=(1-\Gamma)*(C-M)$$

$$\nabla^2_{D^2}=(1-\Gamma)*\frac{\Lambda*(1-e^{-\Lambda})-e^{-\Lambda}*\Lambda^2}{(1-e^{-\Lambda})^2}=(1-\Gamma)*M*(1-e^{-\Lambda}*M)$$

Thus we can calculate the weights and the working response for the second-order approximation and apply *WPCMS* to update the reconstruction X.

Now we deal with the nuisance parameters. First, by analogy with the distance matrix one can compute the first and second order derivatives w.r.t. β (up to $\frac{1}{n^2}$ scale)

$$\nabla_\beta = \sum_{(i,j)\notin\mathcal{N}} (M_{ij} - C_{ij}) \text{ and } \nabla^2_\beta = \sum_{(i,j)\notin\mathcal{N}} M_{ij}(1 - e^{-\lambda_{ij}} M_{ij}).$$

We will use these derivatives to find optimal β when applying Newton's method. Next, we note that the optimal value for the Bernoulli parameter π can be found from the equation

$$\nabla_\pi = \frac{|\mathcal{N}|}{\pi} - \frac{n^2 - |\mathcal{N}|}{1-\pi} = 0.$$

This leads us to the explicit formula $\pi = \frac{|\mathcal{N}|}{n^2}$. Combining all the above steps leads us to the following *HPoisMS algorithm*.

Hurdle Poisson metric scaling (*HPoisMS*)

Set $\pi = \frac{|\mathcal{N}|}{n^2}$, then repeat until convergence:

1. For the current guess of X and β compute SOA:
   - evaluate $\Lambda = e^{-D^2(X)+\beta}$ and $M = \frac{\Lambda}{1-e^{-\Lambda}}$
   - calculate $\nabla_{D^2} = (1-\Gamma) * (C - M)$ and $\nabla^2_{D^2} = (1-\Gamma) * M * (1 - e^{-\Lambda} * M)$
   - compute $W = \nabla^2$ and $Z = D^2(X) - \frac{\nabla}{\nabla^2}$
2. Solve *WPCMS* problem with W and Z thereby updating X.
3. For fixed X run Newton's method to update β. Repeat until convergence:
   - evaluate $\Lambda = e^{-D^2(X)+\beta}$ and $M = \frac{\Lambda}{1-e^{-\Lambda}}$
   - calculate $\nabla_\beta = \sum_{(i,j)\notin\mathcal{N}} (M_{ij} - C_{ij})$ and $\nabla^2_\beta = \sum_{(i,j)\notin\mathcal{N}} M_{ij}(1 - e^{-\lambda_{ij}} M_{ij})$
   - update $\beta := \beta - \frac{\nabla}{\nabla^2}$.

E.2. Zero-inflated Poisson. In this section be build an optimization algorithm for the zero-inflated model. Let $a = \frac{\pi}{1-\pi}$ and rewrite the *ZIPoisMS* loss function as

$$\ell_{\textit{ZIPoisMS}}(X, \beta, \pi; C) =$$

$$= \frac{1}{n^2}\Big[- \sum_{(i,j)\in\mathcal{N}} \log\Big(\pi + (1-\pi)e^{-\lambda_{ij}}\Big) - \sum_{(i,j)\notin\mathcal{N}} \log(1-\pi) + \sum_{(i,j)\notin\mathcal{N}} [\lambda_{ij} - C_{ij}\log\lambda_{ij}]\Big] =$$

$$= \frac{1}{n^2}\Big[- \sum_{(i,j)\in\mathcal{N}} \log\Big(a + e^{-\lambda_{ij}}\Big) - \sum_{1\le i,j\le n} \log(1-\pi) + \sum_{(i,j)\notin\mathcal{N}} [\lambda_{ij} - C_{ij}\log\lambda_{ij}]\Big]$$

We first calculate the derivatives of the loss w.r.t. D^2 (up to $\frac{1}{n^2}$ scale):

$$\nabla_{D^2} = -\Gamma * \frac{\Lambda * e^{-\Lambda}}{a + e^{-\Lambda}} + (1-\Gamma) * (-\Lambda + C) =$$

$$= -\Gamma * \frac{\Lambda}{ae^{\Lambda} + 1} + (1-\Gamma) * (C - \Lambda)$$

$$\nabla^2_{D^2} = -\Gamma * \frac{-\Lambda * (ae^{\Lambda} + 1) + ae^{\Lambda} * \Lambda^2}{(ae^{\Lambda} + 1)^2} + (1-\Gamma) * \Lambda =$$

$$= \Gamma * \Lambda * \frac{ae^{\Lambda}(1-\Lambda) + 1}{(ae^{\Lambda} + 1)^2} + (1-\Gamma) * \Lambda$$

Next, by analogy, we compute the derivatives w.r.t. the intercept β (up to $\frac{1}{n^2}$ scale):

$$\nabla_\beta = \sum_{(i,j)\in\mathcal{N}} \frac{\lambda_{ij}}{ae^{\lambda_{ij}}+1} - \sum_{(i,j)\notin\mathcal{N}} (C_{ij}-\lambda_{ij})$$

$$\nabla^2_\beta = \sum_{(i,j)\in\mathcal{N}} \lambda_{ij}\frac{ae^{\lambda_{ij}}(1-\lambda_{ij})+1}{(ae^{\lambda_{ij}}+1)^2} + \sum_{(i,j)\notin\mathcal{N}} \lambda_{ij}$$

We will use these formulas to update the intercept via Newton's method. Now, we handle the nuisance parameter π. Unlike the Hurdle model, there is no explicit formula for optimal π; therefore, we also apply Newton's method to update it at each iteration. This requires us to compute the derivatives w.r.t. π (up to $\frac{1}{n^2}$ scale)

$$\nabla_\pi = -\sum_{(i,j)\in\mathcal{N}} \frac{1-e^{-\lambda_{ij}}}{\pi+(1-\pi)e^{-\lambda_{ij}}} + \sum_{(i,j)\notin\mathcal{N}} \frac{1}{1-\pi} =$$

$$= -\sum_{(i,j)\in\mathcal{N}} \frac{e^{\lambda_{ij}}-1}{\pi(e^{\lambda_{ij}}-1)+1} + \sum_{(i,j)\notin\mathcal{N}} \frac{1}{1-\pi}$$

$$\nabla^2_\pi = \sum_{(i,j)\in\mathcal{N}} \left(\frac{e^{\lambda_{ij}}-1}{\pi(e^{\lambda_{ij}}-1)+1}\right)^2 + \sum_{(i,j)\notin\mathcal{N}} \frac{1}{(1-\pi)^2}$$

We combine all the steps in the *ZIPoisMS algorithm* stated below.

Zero-inflated Poisson metric scaling (*ZIPoisMS*)

Repeat until convergence:

1. For the current guess of X, β and π compute SOA:
   - evaluate $\Lambda = e^{-D^2(X)+\beta}$ and $a = \frac{\pi}{1-\pi}$
   $$\nabla_{D^2} = -\Gamma * \frac{\Lambda}{ae^{\Lambda}+1} + (1-\Gamma)*(C-\Lambda)$$
   - calculate
   $$\nabla^2_{D^2} = \Gamma * \Lambda * \frac{ae^{\Lambda}(1-\Lambda)+1}{(ae^{\Lambda}+1)^2} + (1-\Gamma)*\Lambda$$
   - compute $W = \nabla^2$ and $Z = D^2(X) - \frac{\nabla}{\nabla^2}$
2. Solve *WPCMS* problem with W and Z thereby updating X.
3. For fixed X, π compute $a = \frac{\pi}{1-\pi}$ and run Newton's method to update β.
   Repeat until convergence:
   - evaluate $\Lambda = e^{-D^2(X)+\beta}$
   $$\nabla_\beta = \sum_{(i,j)\in\mathcal{N}} \frac{\lambda_{ij}}{ae^{\lambda_{ij}}+1} - \sum_{(i,j)\notin\mathcal{N}} (C_{ij}-\lambda_{ij})$$
   - calculate
   $$\nabla^2_\beta = \sum_{(i,j)\in\mathcal{N}} \lambda_{ij}\frac{ae^{\lambda_{ij}}(1-\lambda_{ij})+1}{(ae^{\lambda_{ij}}+1)^2} + \sum_{(i,j)\notin\mathcal{N}} \lambda_{ij}$$
   - update $\beta := \beta - \frac{\nabla}{\nabla^2}$.
4. For fixed X, β compute $\Lambda = e^{-D^2(X)+\beta}$ and run Newton's method to update π.
   Repeat until convergence:

$$\nabla_\pi = -\sum_{(i,j)\in\mathcal{N}} \frac{e^{\lambda_{ij}}-1}{\pi(e^{\lambda_{ij}}-1)+1} + \sum_{(i,j)\notin\mathcal{N}} \frac{1}{1-\pi}$$

- calculate
$$\nabla_\pi^2 = \sum_{(i,j)\in\mathcal{N}} \left(\frac{e^{\lambda_{ij}}-1}{\pi(e^{\lambda_{ij}}-1)+1}\right)^2 + \sum_{(i,j)\notin\mathcal{N}} \frac{1}{(1-\pi)^2}$$
- update $\pi := \pi - \frac{\nabla}{\nabla^2}$.

E.3. Negative Binomial. We extend the methodology to the negative binomial model. We start with taking the derivatives of the loss function

$$\ell_{NBMS}(X,\beta,r;C) = \frac{1}{n^2}\Big[\sum_{1\le i,j\le n} \log\Gamma(r) - \log\Gamma(C_{ij}+r) - r\log r + \\ + \sum_{1\le i,j\le n} (C_{ij}+r)\log(\lambda_{ij}+r) - C_{ij}\log\lambda_{ij}\Big]$$

w.r.t. D^2 leads us to the following equations (up to $\frac{1}{n^2}$ scale)

$$\nabla_{D^2} = -\frac{(C+r)*\Lambda}{\Lambda+r} + C = r\frac{C-\Lambda}{\Lambda+r}$$
$$\nabla^2_{D^2} = r\frac{\Lambda*(\Lambda+r)+\Lambda*(C-\Lambda)}{(\Lambda+r)^2} = r\frac{\Lambda*(C+r)}{(\Lambda+r)^2}$$

We will use these derivatives to calculate the *WPCMS* parameters and, subsequently, update the reconstruction X. Next, we calculate the derivatives w.r.t. β (up to $\frac{1}{n^2}$ scale)

$$\nabla_\beta = -r\sum_{1\le i,j\le n} \frac{C_{ij}-\lambda_{ij}}{\lambda_{ij}+r} \text{ and } \nabla^2_\beta = r\sum_{1\le i,j\le n} \frac{\lambda_{ij}(C_{ij}+r)}{(\lambda_{ij}+r)^2}$$

and use them to update β via Newton's method. Finally, to find optimal r we compute the derivatives w.r.t. this nuisance parameter (up to $\frac{1}{n^2}$ scale):

$$\nabla_r = \sum_{1\le i,j\le n} \psi_0(r) - \psi_0(C_{ij}+r) - \log r + \log(\lambda_{ij}+r) + \frac{C_{ij}-\lambda_{ij}}{\lambda_{ij}+r}$$
$$\nabla^2_r = \sum_{1\le i,j\le n} \psi_1(r) - \psi_1(C_{ij}+r) - \frac{1}{r} + \frac{1}{\lambda_{ij}+r} - \frac{C_{ij}-\lambda_{ij}}{(\lambda_{ij}+r)^2}$$

Here $\psi_0(\cdot)$ and $\psi_1(\cdot)$ correspond to the di- and tri-gamma function. We conclude this section with the *NBMS* algorithm.

Negative binomial metric scaling (*NBMS*)

Repeat until convergence:

1. For the current guess of X, β and r compute SOA:
   - evaluate $\Lambda = e^{-D^2(X)+\beta}$
   - calculate $\nabla_{D^2} = r\frac{C-\Lambda}{\Lambda+r}$ and $\nabla^2_{D^2} = r\frac{\Lambda*(C+r)}{(\Lambda+r)^2}$
   - compute $W = \nabla^2$ and $Z = D^2(X) - \frac{\nabla}{\nabla^2}$
2. Solve *WPCMS* problem with W and Z thereby updating X.
3. For fixed X, r run Newton's method to update β.
   Repeat until convergence:

- evaluate $\Lambda = e^{-D^2(X)+\beta}$
- calculate $\nabla_\beta = -r\sum_{1\le i,j\le n}\frac{C_{ij}-\lambda_{ij}}{\lambda_{ij}+r}$ and $\nabla^2_\beta = r\sum_{1\le i,j\le n}\frac{\lambda_{ij}(C_{ij}+r)}{(\lambda_{ij}+r)^2}$
- update $\beta := \beta - \frac{\nabla}{\nabla^2}$.

4. For fixed X, β compute $\Lambda = e^{-D^2(X)+\beta}$ and run Newton's method to update π. Repeat until convergence:
   - calculate
   $$\nabla_r = \sum_{1\le i,j\le n} \psi_0(r) - \psi_0(C_{ij}+r) - \log r + \log(\lambda_{ij}+r) + \frac{C_{ij}-\lambda_{ij}}{\lambda_{ij}+r}$$
   $$\nabla^2_r = \sum_{1\le i,j\le n} \psi_1(r) - \psi_1(C_{ij}+r) - \frac{1}{r} + \frac{1}{\lambda_{ij}+r} - \frac{C_{ij}-\lambda_{ij}}{(\lambda_{ij}+r)^2}$$
   - update $r := r - \frac{\nabla}{\nabla^2}$.

APPENDIX F: SMOOTH DISTRIBUTION-BASED METRIC SCALING (*SDBMS*)

By analogy with the *SPoisMS* loss (9) one can combine the smoothing spline technique with general distribution-based metric scaling leading to the *SDBMS* loss

$$\ell_{SDBMS}(X, \Omega; C, K, \lambda) = \ell_{DBMS}(X, \Omega; C) + \lambda \operatorname{tr}(X^\top K X) \tag{29}$$

Following the outline from Section (12) one can build the optimization algorithm as follows.

First, we recall that $\operatorname{tr}(KS(X)) = -\frac{1}{2}\operatorname{tr}(KD^2(X))$, so the penalty term in the *SDBMS* loss is a linear function of $D^2(X)$. The updated first derivative involved in the second order approximation is therefore $\tilde\nabla = \nabla - \frac{\lambda}{2}K$ while the second derivative ∇^2 stays the same. The new working response matrix involved in the *WPCMS* is $\tilde Z = Z + \frac{\lambda}{2}\frac{K}{W}$. Finally, as $\Phi(K) = K$ the gradient step in PGD update is

$$\tilde S = XX^\top - \Phi(W * (Z + \frac{\lambda}{2}\frac{K}{W} - D^2(X))) = S + \frac{\lambda}{2}K$$

whereas the projection step minimizes the loss

$$\ell_{PCMS}(X; \tilde S) = \|S - XX^\top\|^2_F + \lambda \operatorname{tr}(X^\top K X).$$

In other words, adding the smoothing penalty to the original *DBMS* problem is equivalent to replacing the *PCMS* projection step with its smoothing spline analog.