STATISTICAL CURVE MODELS FOR INFERRING 3D CHROMATIN ARCHITECTURE

BY ELENA TUZHILINA¹, TREVOR HASTIE^{1,*} AND MARK SEGAL^{2,†}

¹*Department of Statistics, Stanford University, Stanford CA 94305 USA, elenatuz@stanford.edu*

²*Department of Epidemiology and Biostatistics, University of California, San Francisco CA 94143 USA, hastie@stanford.edu; 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. We previously introduced Poisson metric scaling (PoisMS), a technique that models chromatin by a smooth curve, which yielded promising results. In this paper, we advance several ways for improving PoisMS. In particular, we address initialization issues by using a smoothing spline basis. The resulting SPoisMS method produces a sequence of reconstructions re-using previous solutions as warm starts. Importantly, this approach permits smoothing degree to be determined via cross-validation which was problematic using our prior B-spline basis. 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 (from 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 as providing a consensus configuration. The emergence of single cell Hi-C (scHi-C, [Ramani](#)

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

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 with regard deploying cross-validation to determine smoothness degree (for reasons detailed in Sections 3 and 10), which is critical for appropriately capturing the abovementioned second key attribute, folding complexity. Accordingly, in Sections 3 through 9, we introduce a smoothing spline basis, and attendant algorithm SPoisMS, and describe how this improvement enables effective cross-validation, mitigates initialization concerns, can be efficiently implemented by exploiting connections with PoisMS which, in turn, it is shown to outperform in a series of experiments, 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 (DBMS) (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. 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.

Suppose we are aiming to obtain the reconstruction of the chromatin with n genomic loci. Denote the matrix of the loci spacial coordinates by $X = \begin{pmatrix} -x_1^T - \\ \vdots \\ -x_n^T - \end{pmatrix} \in \mathbb{R}^{n \times 3}$. The core assumption of the model is

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

We start with transforming this assumption into a matrix form constraint as follows. First, if curve γ is parametrized by t and t_i index the genomic locus of x_i in the parameter space, then (1) can be reformulated as the set of equations $x_i = \gamma(t_i)$. Next, to incorporate the smoothing into the model, we claim that each component of $\gamma(t) = (\gamma_1(t), \gamma_2(t), \gamma_3(t))^T$ is a cubic spline and decompose and, thus, can be decomposed in a linear combination of the basis functions. In other words, we state that

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

Here k is the size of the basis and is the hyperparameter that controls the “wiggleness” 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 Poisson distribution and link the Poisson parameters to the pairwise distances between loci. To be precise, if $C = [C_{ij}] \in \mathbb{Z}_+^{n \times n}$ is the contact matrix then we assume that $C_{ij} \sim \text{Pois}(\lambda_{ij})$ and

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

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

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

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

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

Note that the particular choice of the log-link (2) exploits the following heuristic. First, the loci that are close to each other in 3D space should have a higher expected contact frequency value. Second, although equivalent to the more traditional exponential link $\lambda_{ij} = \beta \|x_i - x_j\|^\alpha$ proposed by, for example, Rosenthal et al. (2019), it results in the objective that is much easier to optimize. In Tuzhilina, Hastie and Segal (2020) we propose an efficient iterative algorithm that recovers the solution to (4), which, schematically, can be stated as follows.

Poisson Metric Scaling (PoisMS)

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

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

Here W and Z 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 $D_{ij} = \|x_i - x_j\|$.

2. Combining SOA with the smoothing constraint $X = H\Theta$ we state the WPCMS problem

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

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

3. We solve (5) via the WPCMS algorithm based on projected gradient descent (PGD). Performed in the space of the similarity matrices $S(X) = XX^T$, it alternates two steps:

- the gradient step moving S along its gradient $\nabla_S \ell_{WPCMS}$;
- the projection step mapping S onto the space of $H\Theta\Theta^T H^T$ matrices.

If H has orthonormal columns, the projection can be handled via the singular value decomposition of $H^T S H$.

4. WPCMS returns 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.

3. Alternative approach: smoothing splines. There are a few caveats about the PoisMS model. First, since the objective in (4) is non-convex, the PoisMS algorithm converges to a local minimum. Thus the initialization for Θ and β can impact the resulting reconstruction. Although we did not observe significant variation in the PoisMS solution in our experiments, an initialization choice remains an open question. We provide more details in the Appendix of [Tuzhilina, Hastie and Segal \(2020\)](#). Second, it is not exactly clear how to use cross-validation in the context of the contact matrix. The original idea was to hold out some chromatin loci pretending that they were unobserved, train the PoisMS model on the observed loci and evaluate the fit on the held-out set. The training procedure is, therefore, equivalent to removing a subset of rows and columns in the contact matrix, eliminating the corresponding rows in the spline basis matrix, and reconstructing the part of the chromatin using only the observed blocks of C and H . Although such an approach may seem reasonable, one should be very careful selecting the unobserved set of loci. Due to the strong correlation in the contact matrix choosing the loci at random may result in inadequate cross-validation scores.

To demonstrate the correlation we take the IMR90 cell data for chromosome 20 and 100kb resolution, which contains $n = 599$ loci, and compute the reconstruction X via PoisMS. In doing so, we use 25 degrees-of-freedom, the value which was identified as the optimal in [Tuzhilina, Hastie and Segal \(2020\)](#). Next, we evaluate the matrix of the Poisson parameters $\Lambda = e^{-D^2(X)+\beta}$ and compute the error matrix $E = C - \Lambda$. In Figure 1 we present the heatmap for the pairwise correlations between all the rows in E . According to the plot, in addition to the thick high correlation region along the diagonal, we observe multiple continuous off-diagonal areas, where the correlation is very closed to one. Both these patterns imply a strong correlation between the neighboring rows in the error matrix.

To counteract the interaction between the neighboring elements in the contact matrix, instead of selecting unobserved loci at random, we suggest using block cross-validation, thereby removing rows in H as well as rows and columns in C in continuous blocks. This is where PoisMS faces some 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 a very high reconstruction variance.

In this section we propose a novel approach based on the smoothing spline technique that resolves the initialization and cross-validation issued discussed above. First, we rewrite (3) in terms of the γ curve as

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

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

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

The second term in the objective measures the curve smoothness via its second-order derivative and is usually called *roughness penalty*. Therefore, we state an alternative *Smoothing*

FIG 1. The heatmap of the correlation matrix measuring the association between the rows of the error matrix $E = \Lambda - C$. The reconstruction X is computed for $k = 25$ degrees-of-freedom. Since the diagonal elements of C are extremely large the error matrix has huge values on the diagonal as well. To mitigate the influence of the diagonal of E on the resulting correlation we measure the Spearman correlation instead. The thick red region along the diagonal and multiple red spots spread around the heatmap justify the strong correlation between the neighboring rows in the error matrix.

Poisson Metric Scaling (SPoisMS) optimization problem

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

Here Γ is the set of all smooth (second-order derivative exists) one-dimensional curves in $\mathbb{R}^3$. Note that in SPoisMS the “wiggleness” of the resulting reconstruction is controlled in a different way than in PoisMS, where it is done through the basis size k . In SPoisMS increasing λ will put more penalty on the second derivative of γ thereby encouraging more smoothness in the solution.

One can follow the standard smoothing spline theory to restate problem (8) in matrix form (see, for example, [Green and Silverman \(1994\)](#)). First, we claim that each component of the optimal solution γ is a natural cubic spline with knots $t_1, \dots, t_n$. Denote by $n_1(t), \dots, n_n(t)$ the natural spline basis.

If $N \in \mathbb{R}^{n \times n}$ is the matrix representing the basis evaluations at knots, i.e. $N_{i\ell} = n_\ell(t_i)$, and $\Omega \in \mathbb{R}^{n \times n}$ is 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

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

Now, let $K = N^{-T} \Omega N$. Since N is full-rank and non-singular one can use the change of variables $X = N\Theta$ thereby restating the optimization problem (8) as

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

4. Link between PoisMS and the smoothing spline approach. Matrix K has several important properties. We will use these properties to demonstrate that the SPoisMS loss calculated for the original contact matrix C is equivalent to the PoisMS loss computed for $C - \frac{\lambda}{2}K$. As a consequence, the SPoisMS problem can be easily solved by applying the original PoisMS method to adjusted version of the contact matrix.

Denote $K = UDU^T$ the eigendecomposition of the penalty matrix. One can prove that K has two zero eigenvalues and the corresponding eigenvectors span the subspace of linear

functions (see, for example, [Green and Silverman \(1994\)](#)). This fact can be used to validate the equations $K\mathbf{1} = 0$ and $\mathbf{1}^T K = 0$, which further imply $\sum_{1 \leq i, j \leq n} K_{ij} = 0$ (see Appendix A for details). Here $\mathbf{1} = (1, \dots, 1)^T \in \mathbb{R}^n$ refers to the n -dimensional vector of ones. Using these properties we link the SpoisMS and PoisMS losses as stated below.

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

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

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

$$s(X) = \text{diag}(S(X)) = (\|x_1\|^2, \|x_2\|^2, \dots, \|x_n\|^2)^T$$

the diagonal of the inner product matrix. There is a link between $S(X)$ and $D(X)$, i.e.

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

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

$$\text{tr}(X^T K X) = \text{tr}(K S(X)) = -\frac{1}{2} \text{tr}(K D^2(X)) = \frac{1}{2} \sum_{1 \leq i, j \leq n} K_{ij} (-\|x_i - x_j\|^2 + \beta).$$

The following steps conclude the proof:

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

□

It should be noted that this lemma significantly simplifies the practical application of SPoisMS. Instead of developing new iterative algorithms that would solve (9), we could just use the original PoisMS technique on the adjusted contact matrix C^λ .

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

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

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 $H^T S H$, which is a cheap operation for small basis (see Section 2 for the details). However, in the case of the unconstrained problem (11) the projection becomes 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 get around this computational issue. Note that the PoisMS projection step aims to find Θ that minimizes the Frobenius distance between the similarity matrix S and $H\Theta\Theta^T H^T$, which, for orthogonal H , is equivalent to solving

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

In other words, this step seeks for rank three approximation of $H^T SH$, so it is sufficient to compute only the first three singular vectors of this matrix. As the first computations trick, one can consider applying block power method (see, for example, [Bentbib and Kanber \(2017\)](#)), which can be more efficient for finding few singular vectors than calculating full SVD.

The second trick proposes to introduce a constraint to (11). It turns out that, for a proper constraint, the resulting problem would be an accurate approximation of (9) whereas requiring less costly SVD steps. Recall that we decomposed the penalty matrix as $K = U^T D U$. Since the columns of U form the natural spline basis (aka the *Demmler-Reinsch basis*), one can represent the resulting reconstruction as $X = U\Theta$ leading to an alternative form $\text{tr}(\Theta^T D \Theta)$ for the smoothing penalty. Next, we refer to the following nice property of the Demmler-Reinsch basis: the wigglier is the basis element, the larger is the eigenvalue corresponding to it. In terms of the penalty, the smoothness in the reconstruction is encouraged by penalizing the most the coefficients in Θ corresponding to the largest diagonal elements in D and, as a result, related to the wiggliest part of the basis U . Since some of the U columns have a little effect on the resulting reconstruction we can remove them from the basis. In other words, 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

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

This can be easily done via the PoisMS approach again. For k large enough, the resulting solution will be an accurate approximation of the one from (11). However, now each projection step in PoisMS involves the SVD of smaller $k \times k$ matrix.

6. SPoisMS and PoisMS comparison. There are several advantages of using smoothing splines for the chromatin reconstruction. First, the PoisMS approach, where the size of the B-spline basis controls the reconstruction smoothness, requires the user to recompute matrix H for each value of degrees-of-freedom. In contrast, H is fixed in smoothing splines, and the penalty factor controls the model flexibility, thereby implying $\Theta \in \mathbb{R}^{k \times 3}$ of the same shape for any value of λ . This observation can be especially beneficial while choosing the PoisMS algorithm initialization. For instance, one can produce the *path of solutions* gradually decreasing the penalty factor and re-using Θ obtained for larger λ as a warm start for smaller values. Second, the support for each Demmler-Reinsch basis function is $\mathbb{R}$. Thus, returning to the cross-validation issues discussed in Section 3, block cross-validation would demonstrate more robust behavior comparing to the B-spline case.

The main disadvantage of the smoothing spline technique is that you need to keep the basis size k relatively large to get an accurate solution. This makes each step of the SPoisMS computationally more expensive than the PoisMS one. Further, in the case of PoisMS the flexibility of the resulting reconstruction is controlled by k , which corresponds to the B-spline basis size and which can be naturally interpreted as the degrees-of-freedom of the model. On the contrary, although it is obvious that increasing λ should decrease the degrees-of-freedom for SPoisMS, it is hard to formally measure the df in this case. We tackle this problem in the following section.

7. Degrees-of-freedom. The notion of effective degrees-of-freedom is very well studied in the context of linear models and can be defined as the trace of the "hat" matrix (see, for example, [Hastie, Tibshirani and Friedman \(2009\)](#)). In this section we suggest a method for estimating df for the SPoisMS model. The general idea is to approximate the loss ℓ_{SPoisMS} by a quadratic function (equivalently, approximate the original problem by least squares) and use the definition developed for the linear regression.

LEMMA 2. Given some square matrix $M \in \mathbb{R}^{n \times n}$ denote by $\Phi(M)$ the operator that subtracts the row sums of M from its diagonal, i.e. $\Phi(M) = M - \text{diag}(M\mathbf{1})$. Suppose X_0 is the resulting SPoisMS reconstruction. One can show that

$$\ell_{\text{SPoisMS}}(X, \beta; C, K, \lambda) \approx \text{tr}[(X - X_0)^T B(X - X_0)] + \text{tr}[X^T P X]$$

where $B = -\frac{2}{3}\Phi(\Lambda * D^2)$ and $P^\lambda = \Phi(\Lambda - C^\lambda)$.

See Appendix B for the proof. Next recall that for the reduced Demmler-Reinsch basis we should add the constraint $X = H\Theta$. Thus, the resulting approximation has a form of a weighted regression with a quadratic penalty, where the response corresponds to X_0 and the feature matrix is H . Borrowing the definition we derive the formula for degrees-of-freedom

$$(13) \quad df = \text{tr}[(H^T(B + P^\lambda)H)^{-1}(H^T B H)].$$

Before using (13), we slightly adjust it, so it follows two natural assumptions. First, if no penalty is imposed then degrees-of-freedom should be equal to the size of the basis used to produce the fit. In other words, for $\lambda = 0$ we expect $df = \text{rk}(H)$, which can be achieved if $P^0 = \Phi(\Lambda - C) = 0$. Second, in the ideal setting, the degrees-of-freedom should not depend on the resulting reconstruction X . Both statements can be satisfied through replacing Λ with C or, equivalently, substituting expected counts with the observed ones. Note that, by analogy, one can replace $D^2 = \beta - \log(\Lambda)$ by $\beta - \log(C)$, thereby implying the updated formulas

$$(14) \quad B = \frac{2}{3}\Phi(C * \log(C) - \beta C) \text{ and } P^\lambda = \frac{\lambda n^2}{2}\Phi(K).$$

Here we assume that $0 \cdot \log(0) = 0$ for zero contact counts.

8. IMR90 cell experiments. In this section we test the smoothing spline approach on the IMR90 cell chromosome 20 data.

First, we compute reconstructions via the SPoisMS method varying the penalty factor in the grid $\lambda = 10^{-4}, 10^{-5}, \dots, 10^{-9}$. To speed up the computations we truncate the Demmler-Reinsch basis including only the eigen-vectors that correspond to the $k = 300$ smallest eigen-values of K (see Section 5 for more details). We produce the path of the solutions: we start from the largest $\lambda = 10^{-4}$, and reuse 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. In the plot we observe that decreasing the penalty factor produces wigglier result as it imposes less smoothing penalty on the reconstruction. Next, for each λ value we report the corresponding degrees-of-freedom computed via (13) and (14), which we present in table below. According to the table, the considered range of λ covers degrees-of-freedom from $df \approx 6$ (very smooth reconstruction) to $df \approx 117$ (very wiggly). As a final step, to compare the performance of SPoisMS and the original method we round the resulting values of df and compute the PoisMS reconstructions using the B-spline basis of size df and random initialization. We demonstrate the resulting fits in Figure 3.

λ	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}
df	6.15	11.32	20.73	37.56	67.07	117.43

Several conclusions could be drawn from Figures 2 and 3. First, comparing the PoisMS and SPoisMS solutions we observe that the degrees-of-freedom formula derived in Section 7

FIG 2. The projections of the resulting 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 . Colors (orange, teal) distinguish chromosome arms. Before plotting the solutions were aligned via the Procrustes transformation.

produces quite an accurate approximation of d_f for large λ , although it may underestimate d_f for small λ . Second, we conclude that reusing SPoisMS solutions as a warm start produces a well-aligned sequence of fits which allows us to track the evolution of the solution with the growth of λ . In contrast, we observe less agreement between the PoisMS solutions, which the random initialization of the method can explain. Finally, we notice that using warm-starts for subsequent λ values substantially decreases the number of iterations required for the method to converge. This illustrates the utility of the smoothing spline from the computational point of view.

9. Model validation. To further demonstrate the advantages of the SPoisMS approach we discuss how one can evaluate the reconstruction model using the contact matrix. In this section we propose an evaluation method based on a train-test split of genomic loci. Suppose we have only partial access to the contact matrix, i.e. we assume that some of the loci are unobserved. We denote by $\mathcal{O} \subseteq \{1, \dots, n\}$ the set of observed loci and its complement by $\mathcal{U}$. To assess our model we follow two steps:

1. We use the loci from $\mathcal{O}$ to compute the model coefficients Θ and intercept β .

FIG 3. The projections of the resulting reconstructions obtained via the original PoisMS approach for different λ values. The grid of λ values is chosen to match the degrees-of-freedom in Figure 2. Each solution is produced by a random initialization of β and Θ . Colors (orange, teal) distinguish chromosome arms. Before plotting the solutions were aligned via the Procrustes transformation.

2. Then we use $\mathcal{U}$ to evaluate the model's performance.

In what follows, we explain how to tackle these two tasks for both the PoisMS and SPoisMS techniques.

9.1. PoisMS performance evaluation. To simplify the notations, without loss of generality, we can assume that the rows and columns in C are permuted in such a way that the first $|\mathcal{O}|$ of them represent the contact counts for the observed loci. In other words, the contact matrix has block-structure $C = \begin{pmatrix} C_{\mathcal{O}\mathcal{O}} & C_{\mathcal{O}\mathcal{U}} \\ C_{\mathcal{U}\mathcal{O}} & C_{\mathcal{U}\mathcal{U}} \end{pmatrix}$, which, in turn, implies the structure in the basis matrix $H = \begin{pmatrix} H_{\mathcal{O}} \\ H_{\mathcal{U}} \end{pmatrix}$.

We start with the PoisMS approach. To fit the model on observed data we evaluate the log-likelihood only at contact counts from $\mathcal{O}$ thereby leading to the train PoisMS loss

$$(15) \quad \ell_{PoisMS}^{train}(X, \beta; C) = \frac{1}{|\mathcal{O}|^2} \sum_{(i,j) \in \mathcal{O} \times \mathcal{O}} \left[e^{-\|x_i - x_j\|^2 + \beta} - C_{ij} (-\|x_i - x_j\|^2 + \beta) \right].$$

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

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

therefore, minimizing (15) subject to the smoothing constraint $X = H\Theta$ is equivalent to solving the problem

$$(16) \quad \underset{\Theta \in \mathbb{R}^{k \times 3}, \beta \in \mathbb{R}}{\text{minimize}} \quad \ell_{PoisMS}(H_O\Theta, \beta; C_{OO}).$$

In other words, the solution can be found by running PoisMS for the contact matrix C_{OO} and basis H_O .

Now we proceed to the model evaluation step. Note that the original PoisMS algorithm requires the basis matrix to have orthonormal columns, which is true for the full H , but may not be the case for the sub-matrix H_O . Thus before fitting the PoisMS model we need to compute the QR -decomposition $H_O = QR$ and replace the basis matrix by Q . Solving the PoisMS problem results in the coefficient matrix Θ as well as the intercept β . In order to evaluate how well these parameters fit the remaining contact counts we need to compute the complementary test loss

$$(17) \quad \ell_{PoisMS}^{test}(X, \beta; C) = \frac{1}{n^2 - |\mathcal{O}|^2} \sum_{(i,j) \notin \mathcal{O} \times \mathcal{O}} \left[e^{-\|x_i - x_j\|^2 + \beta} - C_{ij} (-\|x_i - x_j\|^2 + \beta) \right].$$

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

$$X = \begin{pmatrix} X_O \\ X_U \end{pmatrix} = \begin{pmatrix} Q\Theta \\ H_U R^{-1} \Theta \end{pmatrix}.$$

This reconstruction will be subsequently plugged in (17) thus producing the model test score.

9.2. SPoisMS performance evaluation. Now we extend the approach proposed in the previous section to the smoothing spline technique. By analogy with (15), one can derive the SPoisMS train loss as follows

$$(18) \quad \begin{aligned} \ell_{SPoisMS}^{train}(\gamma, \beta; C, \lambda) &= \frac{1}{|\mathcal{O}|^2} \sum_{(i,j) \in \mathcal{O} \times \mathcal{O}} \left[e^{-\|\gamma(t_i) - \gamma(t_j)\|^2 + \beta} - C_{ij} (-\|\gamma(t_i) - \gamma(t_j)\|^2 + \beta) \right] \\ &+ \lambda \int \|\gamma''(t)\|^2 dt. \end{aligned}$$

For the penalized loss the following lemma can be proved.

LEMMA 3. *The optimization problem*

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

is equivalent to solving the reduced PoisMS problem

$$(19) \quad \underset{X_O \in \mathbb{R}^{O \times 3}, \beta \in \mathbb{R}}{\text{minimize}} \quad \ell_{PoisMS}(X_O, \beta; C_{OO} - \frac{\lambda |\mathcal{O}|^2}{2} (K/K_{UU})).$$

Here $K/K_{UU} = K_{OO} - K_{OU} K_{UU}^{-1} K_{UO}$ is the Schur complement of the penalty matrix, which has block structure similar to C , i.e. $K = \begin{pmatrix} K_{OO} & K_{OU} \\ K_{UO} & K_{UU} \end{pmatrix}$.

The proof is based on the properties of the smoothing spline basis (see Appendix C 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 one can re-use full K to compute the penalty matrix for the

reduced problem. In the lemma proof we also presented the 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 (17) thereby producing the test score.

Note that the idea from Section 5 can be extended to the partially observed data as well. Specifically, to reduce the computations of the train fit one can obtain an accurate approximation of (24) by taking the SVD of $K/K_{\mathcal{U}\mathcal{U}}$ and using a subset of singular vectors as a basis for $X_{\mathcal{O}}$.

10. Block cross-validation. In Section 3 we raise the issue of high-correlation between the contact counts, which makes the standard cross-validation approach problematic in the context of the chromatin reconstruction. To moderate the influence of correlated data on the model evaluation process, we suggest to use block cross-validation (BCV). In this section we investigate the BCV performance of the PoisMS and SPoisMS approaches using the IMR90 cell data.

First, we split $n = 599$ genomic loci into 20 folds, each containing consecutive loci. We assign one fold to $\mathcal{U}$ thereby treating 5% of loci as unobserved; the remaining 19 folds, or 95% of loci, are assigned to $\mathcal{O}$. Next, following the procedure from Section 9.1, we use $\mathcal{O}$ to compute the solution (Θ, β) either via PoisMS or via SPoisMS. In our experiments, we test SPoisMS with different penalty factors $\lambda = 10^{-4}, 10^{-5}, \dots, 10^{-9}$. To make the PoisMS results compatible to the SPoisMS ones, we use the degrees-of-freedom from Table ?? when computing the PoisMS solution (see Section 8 for the details). As a result, for each of the method and each of the hyperparameter we obtain the coefficient matrix Θ and the intercept β . Finally, to evaluate how well these parameters fit the unobserved set of loci $\mathcal{U}$ we impute the reconstruction and use the completed X to evaluate the test score (17). We repeat the entire experiment for all folds except the boundary ones and present the mean as well as the 1SE intervals of the test score in Figure 4.

From the left panel of the plot we infer that, as expected, the confidence intervals computed via PoisMS "explode" for high values of degrees-of-freedom. On the contrary, the SPoisMS test score exhibits robust behavior regardless of the penalty factor λ . To summarise, introducing smoothness to PoisMS through the roughness penalty instead of the B-spline basis finally enables reconstruction evaluation by means of block cross-validation.

11. Sparse contact matrices and over-dispersion. The contact matrix has two important features. First, usually it is very sparse, i.e. contains big portion of zero counts. Second, it is diagonally dominant (Yang et al., 2017). Both of these factors may cause zero-inflation and over-dispersion of the data thereby making the Poisson distribution unsuitable for modeling the contact counts. To check if this is the case for the IMR90 data we take the reconstruction X produced by the PoisMS approach with $df = 25$ and compute the expected mean count matrix Λ (see Section 3 for more details). In Figure 5 we present the scatter plot for the observed counts C_{ij} vs. the expected ones Λ_{ij} . The dense conglomeration of the points along the x-axis of the figure as well as the significant deviation of the observed values from the predicted ones for high contact counts reveals the potential presence of zero-inflation and over-dispersion in the data.

12. Distribution-based metric scaling. The main concern for using Poisson distribution when modeling the contact counts is that the resulting reconstruction is highly influenced by abundant zeroes and the diagonal of the contact matrix. To mitigate the influence of these data artifacts, one can consider alternative distributions such as zero-inflated Poisson or

FIG 4. Comparison of the block cross-validation performance for the *PoisMS* and *SPoisMS* methods. Left panel: the test score is computed via *PoisMS* with different degrees-of-freedom. For $df = 37$ the confidence interval is much wider than for small degrees-of-freedom, for $df = 67$ and 117 the mean test score is so large that the confidence interval did not fit in the plot limits. Right panel: the test score is computed via *SPoisMS* for a grid of penalty factors (aligned with the df values from the left panel). The train fit is obtained using 95% of the loci, the test score is computed on the remaining 5% of the loci.

FIG 5. Observed vs. expected counts. In this plot we present the dependence between the contact counts C_{ij} and the Poisson parameters Λ_{ij} obtained by *PoisMS* with $k = 25$ degrees-of-freedom. The reconstruction is computed for IMR90 data. Red line: represents the ideal fit where observed and expected counts are equal. Yellow line: represents the smooth approximation of the scatter plot; the zero-inflation of the contact counts is demonstrated by its clinging to the expected counts axis.

negative binomial. It turns out that one can easily extend the *PoisMS* algorithm outline from Section 2 to other distributions as well. Below we propose general *distribution-based metric scaling (DBMS)* and demonstrate how one can use this technique to build three alternative chromatin reconstruction models.

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 . In other words, the resulting negative log-likelihood loss has the form of $\ell(X, \Omega; C)$. To solve the DBMS problem we minimize the loss w.r.t. unknown X and Ω and subject to the smoothing constraint $X = H\Theta$. Similar to the PoisMS case, the optimization algorithm alternates two steps.

Step 1: update the reconstruction. Since the negative log-likelihood depends on the distances only, we can re-write the loss in the form $\ell(D^2(X), \Omega; C)$. 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

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

and evaluate the weights as well as the working response matrices

$$(20) \quad W = \nabla^2 \text{ and } Z = D^2(X) - \frac{\nabla}{\nabla^2}.$$

This leads us to the second order approximation

$$(21) \quad \ell(D^2(X), \Omega; C) \approx \|\sqrt{W} * (Z - D^2(X))\|_F^2.$$

Now, instead of optimizing the original loss we minimize its second order approximation subject to the smoothing constraint $X = H\Theta$. In other words, 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 done via 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 $\ell(D^2(X), \Omega; C)$ w.r.t. the 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 distributions. Below we suggest three alternative distributions for the contact counts: Hurdle Poisson, zero-inflated Poisson, and negative binomial. The optimization algorithm for each of them can be easily built using the DBMS outline from Section 12, which we discuss in detail in the Appendix.

13.1. Hurdle Poisson. Modeling the contact counts with Hurdle Poisson consist of two parts (see, for example, Gurmu (1998)). We first assume that attaining zero count follows the Bernoulli distribution; we further introduce the zero-truncated Poisson (ZTP) distribution for non-zero counts, i.e.

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

This leads us to the following distribution of 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 the dependence between the truncated Poisson parameters λ_{ij} and the 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

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

The optimal solution can be found via the HPoisMS algorithm that we discuss in Appendix D.1.

13.2. Zero Inflated Poisson. The zero-inflated Poisson distribution addresses the abundance of zero contact in a different way (Lambert, 1992). It assumes that zero observations have two different origins: “structural” and “sampling”. The structural zeros occur with some fixed probability π , whereas the sampling zeros appear as a null Poisson outcome, i.e.

$$C_{ij} = 0 \text{ with probability } \pi$$

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

This leads us to the following distribution of the contact counts

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

which, combined with the same log-link (2), implies the corresponding loss function

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

Zero-inflated Poisson metric scaling (ZIPoisMS), an optimization procedure that finds the optimal X, β and π , is proposed in Appendix D.2.

13.3. Negative Binomial. Next, we consider the negative binomial model. We use the following distribution for the contact frequencies:

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

Comparing to the Poisson approach, which implies $E(C_{ij}) = \text{Var}(C_{ij})$, the negative binomial technique links the contact count mean and variance through the relation $\text{Var}(C_{ij}) = E(C_{ij}) + \frac{E(C_{ij})^2}{r}$. The nuisance parameter r adds more flexibility to the model thereby allowing overdispersion to be addressed.

Linking parameter λ_{ij} to the pair-wise distances $\|x_i - x_j\|$ through equation (2) one can state the goal of *negative binomial metric scaling (NBMS)*: find the optimal X, β, r that minimize the negative log-likelihood

$$\begin{aligned} \ell_{NBMS}(X, \beta, r; C) = & \sum_{1 \leq i, j \leq n} \log \Gamma(r) - \log \Gamma(C_{ij} + r) - r \log r + \\ & + \sum_{1 \leq i, j \leq n} (C_{ij} + r) \log(e^{-\|x_i - x_j\|^2 + \beta} + r) - C_{ij} (-\|x_i - x_j\|^2 + \beta) \end{aligned}$$

FIG 6. The projections of the resulting reconstructions obtained via four methods: *PoisMS*, *HPoisMS*, *ZIPoisMS*, *NBMS*. The degrees-of-freedom is set to 25 for the B-spline basis.

See Appendix D.3 for the details on the NBMS optimization algorithm.

14. Four models comparison. In this section we model contact counts by means of the four distributions discussed above (Poisson, Hurdle and zero-inflated Poisson as well as negative binomial). We start with a simple visual comparison of the models. We use IMR90 data and obtain four reconstructions approximation the chromatin curve by a B-spline with 25 degrees-of-freedom. Comparing the images in Figures 6, we can conclude that the Negative Binomial model tends to separate before and after centromere parts of the chromatin (blue and orange color in the plot). In contrast, the models that account for excessive zero counts, i.e. Hurdle and zero-inflated Poisson, imply more interaction between these two parts. This observation is also supported by Figure 7, where we present the heatmap for $\Lambda(X) = e^{-D(X)+\beta}$. In the NB panel we clearly see two prominent diagonal blocks evidencing strong interaction within each of the two pieces of chromatin; the two off-diagonal blocks, however, have values closed to zero and imply scarce cross-contact. From the same plot one can infer that, unlike *HPoisMS* and *ZIPoisMS*, the *NBPoisMS* model was able to moderate the influence of the diagonal elements of the contact matrix on the resulting reconstruction.

To compare the models performance, we use contact matrices for chromosome 1 from eight single mouse embryonic stem cells (Stevens et al., 2017, Gene Expression Omnibus

FIG 7. The heatmaps for $\Lambda = e^{-D(X)+\beta}$ the resulting reconstructions obtained via four methods: *PoisMS*, *HPoisMS*, *ZIPoisMS*, *NBMB*. The degrees-of-freedom is set to 25 for the B-spline basis.

(GEO) repository GSE80280) denoted $C^{(1)}, \dots, C^{(8)}$. 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 $\mathcal{T} \in \{1, \dots, 8\}$ 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 1SE confidence interval.

In Figure 8 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 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 $df = 10$. The extreme sparsity of the contact matrix can explain such a difference in performance: the data has more than 98% of zero contact counts, thus, it strongly benefits from zero-inflated models.

FIG 8. 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 1SE bands of the test scores. The plot evidences the superior performance of the ZIPoisMS models to the competitors.

15. Discussion. In this paper we propose several extensions 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 constrain (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 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.

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

Proposed methods are implemented 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.
- BELIAEVA, A., KUBJAS, K., SUN, L. J. and UHLER, C. (2021). Identifying 3D genome organization in diploid organisms 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., BENGTTSSON, 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.
- 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.
- 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., SANZO, 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.
- 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 $\mathbf{1} = (1, \dots, 1)^T \in \mathbb{R}^n$ is n -dimensional vector of ones and K is the smoothing spline penalty matrix, then

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

PROOF. Denote $K = UDU^T$ 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 $G = (g(t_1), \dots, g(t_n))^T$, we get $G \in \text{span}(U_0)$ as well as $(U_0^\perp)^T G = 0$. This leads us to the relation

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

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

APPENDIX B: DEGREES-OF-FREEDOM

LEMMA 2. Given some square matrix $M \in \mathbb{R}^{n \times n}$ denote by $\Phi(M)$ the operator that subtracts the row sums of M from its diagonal, i.e. $\Phi(M) = M - \text{diag}(M\mathbf{1})$. Suppose X_0 is the resulting SPoisMS reconstruction. One can show that

$$\ell_{\text{SPoisMS}}(X, \beta; C, K, \lambda) \approx \text{tr}[(X - X_0)^T B(X - X_0)] + \text{tr}[X^T P X]$$

where $B = -\frac{2}{3}\Phi(\Lambda * D^2)$ and $P^\lambda = \Phi(\Lambda - C^\lambda)$.

PROOF. Denote by $A^{(ij)}$ the matrix such that all the elements are equal to zero except

$$A_{ii}^{(ij)} = A_{jj}^{(ij)} = 1 \text{ and } A_{ij}^{(ij)} = A_{ji}^{(ij)} = -1.$$

If $i = j$ we can assume that $A^{(ij)} = 0$. One can show that for any matrix $M \in \mathbb{R}^{n \times n}$ the following relation holds

$$\sum_{1 \leq i, j \leq n} M_{ij} A^{(ij)} = -\Phi(M + M^T).$$

Further, combining

$$\nabla_X \|x_i - x_j\|^2 = 2 \begin{pmatrix} 0 \\ -(x_i - x_j)^T \\ \dots \\ -(x_j - x_i)^T \\ 0 \end{pmatrix} = 2A^{(ij)} X \text{ and } \lambda_{ij} = e^{-\|x_i - x_j\|^2 + \beta}$$

we derive that $\nabla_X \lambda_{ij} = -2\lambda_{ij} A^{(ij)} X$. Thus we compute the first order derivative of (3) as

$$\begin{aligned} \nabla_X \ell_{\text{SPoisMS}}(X, \beta; C, K, \lambda) &= \frac{1}{n^2} \nabla_X \left[\sum_{1 \leq i, j \leq n} (\lambda_{ij} - C_{ij}^\lambda \cdot \log(\lambda_{ij})) \right] = \\ &= -\frac{2}{n^2} \sum_{1 \leq i, j \leq n} (\lambda_{ij} - C_{ij}^\lambda) A^{(ij)} \cdot X = \frac{4}{n^2} \Phi(\Lambda - C^\lambda) \cdot X = \frac{4}{n^2} P^\lambda X \end{aligned}$$

Here we exploit the fact that both Λ and C are symmetric matrices. Next we compute the second order derivatives. To make it simpler we compute them with respect to each column in X separately. Denote the columns of X by X_i , i.e. $X = (X_1, X_2, X_3)$. If $f : \mathbb{R}^{n \times 3} \rightarrow \mathbb{R}$ is some arbitrary scalar function, then

$$\nabla_{X_k}(f(X) \cdot X_k) = (\nabla_{X_k} f(X)) \cdot X_k^T + \text{diag}(f(X))$$

From the first derivative equations it follows that

$$\begin{aligned} \nabla_{X_k}^2 \ell_{SPoisMS}(X, \beta; C, K, \lambda) &= -\frac{2}{n^2} \nabla_{X_k} \left[\sum_{1 \leq i, j \leq n} (\lambda_{ij} - C_{ij}^\lambda) A^{(ij)} \cdot X_k \right] = \\ &= -\frac{2}{n^2} \sum_{1 \leq i, j \leq n} \nabla_{X_k}^2 [(\lambda_{ij} - C_{ij}^\lambda) \cdot X_k] A^{(ij)T} = \\ &= -\frac{2}{n^2} \sum_{1 \leq i, j \leq n} [-2\lambda_{ij} A^{(ij)} X_k \cdot X_k^T + \text{diag}(\lambda_{ij} - C_{ij}^\lambda)] A^{(ij)T} = \\ &= \frac{4}{n^2} \sum_{1 \leq i, j \leq n} \lambda_{ij} \cdot A^{(ij)} X_k \cdot (A^{(ij)} X_k)^T - \frac{2}{n^2} \sum_{1 \leq i, j \leq n} (\lambda_{ij} - C_{ij}^\lambda) A^{(ij)} = \\ &= \frac{4}{n^2} \sum_{1 \leq i, j \leq n} \lambda_{ij} (X_{ik} - X_{jk})^2 A^{(ij)} + \frac{4}{n^2} \Phi(\Lambda - C^\lambda) \end{aligned}$$

To make it simple we will approximate the actual Hessian by the average of the second order derivatives, which has a nice closed form

$$\begin{aligned} \frac{1}{3} (\nabla_{X_1}^2 + \nabla_{X_2}^2 + \nabla_{X_3}^2) &= \frac{4}{3n^2} \sum_{1 \leq i, j \leq n} \lambda_{ij} \cdot \|x_i - x_j\|^2 A^{(ij)} + \frac{4}{n^2} \Phi(\Lambda - C^\lambda) = \\ &= -\frac{8}{3n^2} \Phi(\Lambda * D^2(X)) + \frac{4}{n^2} \Phi(\Lambda - C^\lambda) = \frac{4}{n^2} (P^\lambda + B) \end{aligned}$$

This leads us to the approximation (up to a constant) as follows

$$\begin{aligned} \ell_{SPoisMS}(X, \beta; C, K, \lambda) &\approx \text{tr} [X_0^T P^\lambda (X - X_0)] + \frac{1}{2} \text{tr} [(X - X_0)^T (P^\lambda + B) (X - X_0)] \propto \\ &\propto \text{tr} [(X - X_0)^T B (X - X_0)] + \text{tr} [X^T P^\lambda X] \end{aligned}$$

Note that the second order approximation in the original form should include terms $X_k^T \nabla_{X_k}^2 X_k$ as well as the cross-terms $X_k^T \nabla_{X_k X_\ell}^2 X_\ell$, where $\nabla_{X_k X_\ell}^2$ is the mixed second derivative computed w.r.t. both X_k and X_ℓ . In our case we ignored the mixed terms and replaced all $\nabla_{X_k}^2$ by their average. In other words, we approximate the original Hessian by a block-diagonal version with the same block $\frac{1}{3} (\nabla_{X_1}^2 + \nabla_{X_2}^2 + \nabla_{X_3}^2)$ on the diagonal. $\square$

APPENDIX C: SPOISMS PERFORMANCE EVALUATION

LEMMA 3. *The optimization problem*

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

is equivalent to solving the reduced PoisMS problem

$$(22) \quad \underset{X_O \in \mathbb{R}^{n \times 3}, \beta \in \mathbb{R}}{\text{minimize}} \ell_{PoisMS}(X_O, \beta; C_{OO} - \frac{\lambda |\mathcal{O}|^2}{2} (K/K_{UU})).$$

Here $K/K_{UU} = K_{OO} - K_{OU} K_{UU}^{-1} K_{UO}$ is the Schur complement of the penalty matrix, which has block structure similar to C , i.e. $K = \begin{pmatrix} K_{OO} & K_{OU} \\ K_{UO} & K_{UU} \end{pmatrix}$.

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

$$\ell_{SPoisMS}^{train}(X, \beta; C, K, \lambda) = \ell_{PoisMS}(X_{\mathcal{O}}, \beta; C_{\mathcal{O}\mathcal{O}}) + \lambda \text{tr}(X^T K X).$$

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

$$\text{tr}(X^T K X) = \text{tr}(X_{\mathcal{O}}^T K_{\mathcal{O}\mathcal{O}} X_{\mathcal{O}} + 2X_{\mathcal{U}}^T K_{\mathcal{U}\mathcal{O}} X_{\mathcal{O}} + X_{\mathcal{U}}^T K_{\mathcal{U}\mathcal{U}} X_{\mathcal{U}}).$$

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

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

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

$$(23) \quad \underset{X_{\mathcal{O}} \in \mathbb{R}^{|\mathcal{O}| \times 3}, \beta \in \mathbb{R}}{\text{minimize}} \quad \ell_{SPoisMS}(X_{\mathcal{O}}, \beta; C_{\mathcal{O}\mathcal{O}}, K/K_{\mathcal{U}\mathcal{U}}, \lambda).$$

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 (23) 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. $\begin{cases} K_{\mathcal{O}\mathcal{O}}\mathbf{1} = -K_{\mathcal{O}\mathcal{U}}\mathbf{1} \\ K_{\mathcal{U}\mathcal{O}}\mathbf{1} = -K_{\mathcal{U}\mathcal{U}}\mathbf{1} \end{cases}$, leading us to analogous property of the Schur complement

$$(K/K_{\mathcal{U}\mathcal{U}})\mathbf{1} = K_{\mathcal{O}\mathcal{O}}\mathbf{1} - K_{\mathcal{O}\mathcal{U}}K_{\mathcal{U}\mathcal{U}}^{-1}K_{\mathcal{U}\mathcal{O}}\mathbf{1} = K_{\mathcal{O}\mathcal{O}}\mathbf{1} + K_{\mathcal{O}\mathcal{U}}\mathbf{1} = 0.$$

Following the proof in the main paper, one can show that the smoothing penalty in the loss function $\ell_{SPoisMS}(X_{\mathcal{O}}, \beta; C_{\mathcal{O}\mathcal{O}}, K/K_{\mathcal{U}\mathcal{U}}, \lambda)$ can be absorbed into the PoisMS part thereby proving the equivalence between (23) and the PoisMS problem

$$(24) \quad \underset{X_{\mathcal{O}} \in \mathbb{R}^{|\mathcal{O}| \times 3}, \beta \in \mathbb{R}}{\text{minimize}} \quad \ell_{PoisMS}(X_{\mathcal{O}}, \beta; C_{\mathcal{O}\mathcal{O}} - \frac{\lambda}{2}(K/K_{\mathcal{U}\mathcal{U}})).$$

□

APPENDIX D: 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.

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

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

We first compute the derivatives of the loss w.r.t. D^2 . Denote $M = \frac{\Lambda}{1 - e^{-\Lambda}}$, then

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

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

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

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

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 $\nabla_{D^2}^2 = (1 - \Gamma) * M * (1 - e^{-\Lambda} * M)$
 - compute $W = \nabla^2$ and $Z = D^2(X) - \frac{\nabla^2}{\nabla_{D^2}^2}$
 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 $\nabla_{\beta}^2 = \sum_{(i,j) \notin \mathcal{N}} M_{ij} (1 - e^{-\lambda_{ij}} M_{ij})$
 - update $\beta := \beta - \frac{\nabla_{\beta}}{\nabla_{\beta}^2}$.
-

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

$$\begin{aligned} \ell_{ZIPoisMS}(X, \beta, \pi; C) &= \\ &= - \sum_{(i,j) \in \mathcal{N}} \log \left(\pi + (1 - \pi) e^{-\lambda_{ij}} \right) - \sum_{(i,j) \notin \mathcal{N}} \log(1 - \pi) + \sum_{(i,j) \notin \mathcal{N}} [\lambda_{ij} - C_{ij} \log \lambda_{ij}] = \end{aligned}$$

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

We first calculate the derivatives of the loss w.r.t. D^2 :

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

Next, by analogy, we compute the derivatives w.r.t. the intercept β :

$$\begin{aligned} \nabla_{\beta} &= \sum_{(i,j) \in \mathcal{N}} \frac{\lambda_{ij}}{ae^{\lambda_{ij}} + 1} - \sum_{(i,j) \notin \mathcal{N}} (C_{ij} - \lambda_{ij}) \\ \nabla_{\beta}^2 &= \sum_{(i,j) \in \mathcal{N}} \lambda_{ij} \frac{ae^{\lambda_{ij}}(1 - \lambda_{ij}) + 1}{(ae^{\lambda_{ij}} + 1)^2} + \sum_{(i,j) \notin \mathcal{N}} \lambda_{ij} \end{aligned}$$

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

$$\begin{aligned} \nabla_{\pi} &= - \sum_{(i,j) \in \mathcal{N}} \frac{1 - e^{-\lambda_{ij}}}{\pi + (1 - \pi)e^{-\lambda_{ij}}} + \sum_{(i,j) \notin \mathcal{N}} \frac{1}{1 - \pi} = \\ &= - \sum_{(i,j) \in \mathcal{N}} \frac{e^{\lambda_{ij}} - 1}{\pi(e^{\lambda_{ij}} - 1) + 1} + \sum_{(i,j) \notin \mathcal{N}} \frac{1}{1 - \pi} \\ \nabla_{\pi}^2 &= \sum_{(i,j) \in \mathcal{N}} \left(\frac{e^{\lambda_{ij}} - 1}{\pi(e^{\lambda_{ij}} - 1) + 1} \right)^2 + \sum_{(i,j) \notin \mathcal{N}} \frac{1}{(1 - \pi)^2} \end{aligned}$$

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

Zero-inflated Poisson Metric Scaling (ZIPoisMS)

Repeat until convergence:

- 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

$$\nabla_{D^2}^2 = \Gamma * \Lambda * \frac{ae^{\Lambda}(1 - \Lambda) + 1}{(ae^{\Lambda} + 1)^2} + (1 - \Gamma) * \Lambda$$
 - compute $W = \nabla^2$ and $Z = D^2(X) - \frac{\nabla}{\nabla^2}$
- 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

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

- update $\beta := \beta - \frac{\nabla_{\beta}}{\nabla_{\beta}^2}$.

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 $\pi := \pi - \frac{\nabla_{\pi}}{\nabla_{\pi}^2}$.

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

$$\begin{aligned} \ell_{NBMS}(X, \beta, r; C) &= \sum_{1 \leq i, j \leq n} \log \Gamma(r) - \log \Gamma(C_{ij} + r) - r \log r + \\ &\quad + \sum_{1 \leq i, j \leq n} (C_{ij} + r) \log(\lambda_{ij} + r) - C_{ij} \log \lambda_{ij} \end{aligned}$$

w.r.t. D^2 leads us to the following equations

$$\begin{aligned} \nabla_{D^2} &= - \frac{(C + r) * \Lambda}{\Lambda + r} + C = r \frac{C - \Lambda}{\Lambda + r} \\ \nabla_{D^2}^2 &= r \frac{\Lambda * (\Lambda + r) + \Lambda * (C - \Lambda)}{(\Lambda + r)^2} = r \frac{\Lambda * (C + r)}{(\Lambda + r)^2} \end{aligned}$$

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

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

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:

$$\begin{aligned} \nabla_r &= \sum_{1 \leq i, j \leq n} \psi_0(r) - \psi_0(C_{ij} + r) - \log r + \log(\lambda_{ij} + r) + \frac{C_{ij} - \lambda_{ij}}{\lambda_{ij} + r} \\ \nabla_r^2 &= \sum_{1 \leq i, j \leq n} \psi_1(r) - \psi_1(C_{ij} + r) - \frac{1}{r} + \frac{1}{\lambda_{ij} + r} - \frac{C_{ij} - \lambda_{ij}}{(\lambda_{ij} + r)^2} \end{aligned}$$

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 $\nabla_{D^2}^2 = r \frac{\Lambda*(C+r)}{(\Lambda+r)^2}$
- 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 $\nabla_{\beta} = -r \sum_{1 \leq i, j \leq n} \frac{C_{ij} - \lambda_{ij}}{\lambda_{ij} + r}$ and $\nabla_{\beta}^2 = r \sum_{1 \leq i, j \leq n} \frac{\lambda_{ij}(C_{ij} + r)}{(\lambda_{ij} + r)^2}$
- 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 $\nabla_r = \sum_{1 \leq i, j \leq n} \psi_0(r) - \psi_0(C_{ij} + r) - \log r + \log(\lambda_{ij} + r) + \frac{C_{ij} - \lambda_{ij}}{\lambda_{ij} + r}$
 - calculate $\nabla_r^2 = \sum_{1 \leq i, j \leq n} \psi_1(r) - \psi_1(C_{ij} + r) - \frac{1}{r} + \frac{1}{\lambda_{ij} + r} - \frac{C_{ij} - \lambda_{ij}}{(\lambda_{ij} + r)^2}$
 - update $r := r - \frac{\nabla}{\nabla^2}$.
-