Bayesian Analysis (2012) **7**, Number 3, pp. 547–566

Bayesian Matching of Unlabeled Point Sets Using Procrustes and Configuration Models

Kim Kenobi[*] and Ian L. Dryden[†]

Abstract. The problem of matching unlabeled point sets using Bayesian inference is considered. Two recently proposed models for the likelihood are compared, based on the Procrustes size-and-shape and the full configuration. Bayesian inference is carried out for matching point sets using Markov chain Monte Carlo simulation. An improvement to the existing Procrustes algorithm is proposed which improves convergence rates, using occasional large jumps in the burn-in period. The Procrustes and configuration methods are compared in a simulation study and using real data, where it is of interest to estimate the strengths of matches between protein binding sites. The performance of both methods is generally quite similar, and a connection between the two models is made using a Laplace approximation.

Keywords: Gibbs, Markov chain Monte Carlo, Metropolis-Hastings, molecule, protein, Procrustes, size, shape

1 Introduction

Matching configurations of points is an important but challenging problem in many application areas, including in bioinformatics and computer vision. In this paper we compare two Bayesian approaches that have been developed for matching unlabeled point sets. The matching problem, where the sets of points may be of different sizes, is relevant for the comparison of molecules and the comparison of objects from different views in computer vision. For example, if we have two protein surfaces, a question of interest is whether the two surfaces have a region of the same shape. This region may correspond to a binding site that the proteins have in common; for example they may both bind to the same protein molecule.

In this paper we compare and build on the Markov chain Monte Carlo (MCMC) methods recently independently developed by Green and Mardia (2006), Dryden et al. (2007) and Schmidler (2007), which themselves have connections with work stemming from Moss and Hancock (1996) and Rangarajan et al. (1997), among others.

Green and Mardia (2006) include details of a small dataset where the problem is one of matching unlabeled point sets, and we use this dataset as a testbed for our comparisons. The dataset consists of the coordinates of the centers of gravity of the amino acids that make up the nicotinamide adenine dinucleotide phosphate (NADP) binding sites of two proteins. Protein 1 is the human protein 17-beta hydroxysteroid dehydrogenase. Protein 2 is the mouse protein carbonyl reductase. The active site of protein 1 contains 40 amino acids and the active site of protein 2 contains 63 amino acids.

[*] School of Biosciences, University of Nottingham, Leicestershire, UK, kim.kenobi@nottingham.ac.uk
[†] Department of Statistics, University of South Carolina, Columbia, SC, dryden@mailbox.sc.edu

DOI:10.1214/12-BA718

Green and Mardia (2006) implemented their MCMC algorithm on the protein data. In Table 4 of Green and Mardia (2004) (which is not in Green and Mardia (2006)) each of the suggested pairings between amino acids in protein 1 and in protein 2 is assigned a probability. These probabilities were estimated by observing how often those matches were represented in long runs of the MCMC algorithm after convergence, and we use these findings as a basis for comparing the algorithms.

This paper consists of two main contributions. First we describe an improvement to the algorithm of Dryden et al. (2007) to prevent it from getting trapped in local modes in the burn-in period. This method involves introducing some irreversible big jumps to find a good starting point for the MCMC algorithm. Secondly we compare the performance of MCMC algorithms for simulating from two different Bayesian models: involving Procrustes matching (as in Dryden et al. (2007) and Schmidler (2007)) and involving the full configuration (as in Green and Mardia (2006)).

2 Procrustes model

2.1 Match matrix

Consider two configurations of M and N points in m dimensions, and we write X as an $M \times m$ matrix and μ as an $N \times m$ matrix of co-ordinates. In our application the configurations are molecules, the points are amino acid functional site centroids, and the configurations are in $m = 3$ dimensions. A key part of protein molecule matching is to identify which functional sites correspond between two molecules. In chemoinformatics when comparing smaller drug molecules the points are atoms and it is of interest to find correspondences between pairs of atoms in molecules.

In order to specify the labeling or correspondence between the points we use a *match matrix* Λ, which is an $M \times (N+1)$ matrix of 1s and 0s, in which every row sums to 1 to represent a particular matching of the points in X to the points in μ. For $1 \leq j \leq N$, if $\lambda_{ij} = 1$ then the i^{th} point of X matches to the j^{th} point in μ. If $\lambda_{i,N+1} = 1$ then the i^{th} point of X does not match to any point in μ. Note that there is no requirement for the columns to sum to 1, and so many-to-one matches are allowed. Also, the matching is not symmetric - in general the match from point set A to B will differ from the match from point set B to A.

We shall consider two approaches to molecule matching using different Bayesian models: a Procrustes size-and-shape model (Dryden et al. 2007; Schmidler 2007) and a configuration model (Green and Mardia 2006). The methods use Markov Chain Monte Carlo (MCMC) simulation to draw inferences about the match matrix and a concentration parameter, although the treatment of the rotation and translation nuisance parameters differs.

2.2 Likelihood

We first consider the Procrustes-based model and construct the likelihood which is a function of the size-and-shape distance, and requires registering configurations over translation and rotation. Consider a match matrix, Λ, with p matching points, and configuration matrices X and μ, where X as a random configuration and μ as fixed. Let X^Λ be a $p \times m$ matrix of the rows of X for which $\lambda_{i,N+1} = 0$ (i.e. the matched points in X), and let μ^Λ be a $p \times m$ matrix of the rows of μ which correspond to the points in μ to which the points of X^Λ are matched. We assume that X^Λ and μ^Λ are both centered configurations, i.e. $1_p^T X^\Lambda = 0_m^T = 1_p^T \mu^\Lambda$.

A rotation of X is given by post-multiplication by a rotation matrix $\Gamma \in SO(m)$, where $SO(m)$ is the special orthogonal group of $m \times m$ matrices such that $\Gamma^T\Gamma = \Gamma\Gamma^T = I_m$ and $|\Gamma| = 1$. A translation of X is given by addition of each row by $\gamma^T \in \mathbb{R}^m$. The size-and-shape of the configuration consists of all geometrical properties that are invariant under rotation and translation of X^Λ, i.e. the size-and-shape of $X^\Lambda\Gamma + 1_p\gamma^T$ is the same as that of X^Λ (see Dryden and Mardia (1992) and Dryden and Mardia (1998, Chapter 8)). Here 1_p is the p-vector of ones, 0_m is the m-vector of zeroes, and I_m is the $m \times m$ identity matrix.

We use partial Procrustes registration to register X^Λ to μ^Λ, in order to define a distance between the size-and-shapes. This aspect of the matching is present in both the Dryden et al. (2007) and Schmidler (2007) approaches. The Procrustes matching involves finding $\hat\Gamma \in SO(m)$ and $\hat\gamma \in \mathbb{R}^m$ such that

$$\| \mu^\Lambda - X^\Lambda\hat\Gamma - 1_p\hat\gamma^T \| = \inf_{\substack{\Gamma \in SO(m) \\ \gamma \in \mathbb{R}^p}} \| \mu^\Lambda - X^\Lambda\Gamma - 1_p\gamma^T \| = d_S(X^\Lambda, \mu^\Lambda),$$

where $d_S(X^\Lambda, \mu^\Lambda)$ is the Riemannian metric in size-and-shape space, $S\Sigma_m^p$ (see Kendall (1989); Dryden and Mardia (1992, 1998)). The Procrustes estimators of rotation and translation, $\hat\Gamma$ and $\hat\gamma$ are

$$\hat\gamma = 0_p, \quad \hat\Gamma = R_1 R_2^T, \qquad\qquad R_1, R_2 \in SO(m),$$

where $(\mu^\Lambda)^T X^\Lambda = \|X^\Lambda\| \|\mu^\Lambda\| R_2 D R_1^T$ and $D = \text{diag}(l_1, l_2, \ldots, l_m)$ is an $m \times m$ diagonal matrix where the eigenvalues, l_j, are optimally signed $(l_1 \geq l_2 \geq \ldots \geq |l_m| \geq 0)$ and non-degenerate $(l_{m-1} + l_m > 0)$, see Kent and Mardia (2001). Note that $\hat\gamma = 0_p$ because X^Λ and μ^Λ have both been centered.

Let $\hat X^\Lambda = X^\Lambda\hat\Gamma + 1_p\hat\gamma^T$. Then $\hat X^\Lambda$ is the partial Procrustes fit of X^Λ onto μ^Λ. (It is 'partial' because no scaling has been used, just rotation and translation.) The registration is computed efficiently using the `shapes` package in R (Dryden 2011).

The partial Procrustes tangent coordinates of X^Λ at μ^Λ are given by the $p \times m$ matrix

$$V^\Lambda = \hat X^\Lambda - \mu^\Lambda = X^\Lambda\hat\Gamma + 1_p\hat\gamma^T - \mu^\Lambda$$

which is in a $pm - m(m-1)/2 - m$ dimensional linear subspace of $\mathbb{R}^{mp}$.

We denote the unmatched points in X by $X^{-\Lambda}$. We transform these points using the same transformation parameters as for X^{Λ}. Let $\hat{X}^{-\Lambda} = X^{-\Lambda}\hat{\Gamma} + 1_{M-p}\hat{\gamma}^T$. We consider $X^{-\Lambda}$ to lie in $\mathbb{R}^{(M-p)m}$.

Given the match matrix, Λ, the size-and-shape of X^{Λ} lies in $S\Sigma_m^p$ and $X^{-\Lambda}$ lies in $\mathbb{R}^{(M-p)m}$.

We assume a zero mean isotropic Gaussian model for V^{Λ} in $Q = pm - m(m-1)/2 - m$ dimensions. (There are $m(m-1)/2 + m$ linear constraints on V^{Λ} due to the Procrustes registration.) We assume that $X^{-\Lambda}$, the non-matching part, is uniformly distributed in a bounded region, $\mathcal{A}$, with volume $|\mathcal{A}|$ of $\mathbb{R}^m$.

The likelihood of X given Λ and $\tau = 1/\sigma^2$, a precision parameter where σ^2 is a measure of the variability at each point, is

$$\begin{aligned} L(X|\Lambda,\tau,\mu) &= f_{V^{\Lambda}}(V^{\Lambda}|\tau,\Lambda,\mu) f_{X^{-\Lambda}}(X^{-\Lambda}|\Lambda) \\ &= (2\pi)^{-Q/2}\tau^{Q/2}\exp\left(-\frac{\tau}{2}\mathrm{trace}\{(V^{\Lambda})^T V^{\Lambda}\}\right) \times \frac{1}{|\mathcal{A}|^{M-p}} \\ &= (2\pi)^{-Q/2}\tau^{Q/2}\exp\left(-\frac{\tau}{2}d_S(X^{\Lambda},\mu^{\Lambda})^2\right) \times \frac{1}{|\mathcal{A}|^{M-p}}. \end{aligned}$$

This likelihood is given by Dryden et al. (2007) and is essentially that of Schmidler (2007) (with $Q = mp$ in the latter).

2.3 Prior and posterior distributions

We write $\pi(\tau)$ and $\pi(\Lambda)$ for the prior distributions of τ and Λ and assume τ and Λ are independent *a priori*. We use the prior distribution $\tau \sim \Gamma(\alpha_0, \beta_0)$.

For the prior distribution of Λ, we assume the rows are independently distributed with the i^{th} row having distribution

$$\pi(\lambda_{i,N+1} = 1) = \psi, \quad \pi(\lambda_{ij} = 1) = \frac{1-\psi}{N}, \qquad 1 \le j \le N,$$

for $1 \le i \le M$ and $0 \le \psi \le 1$. If $\psi = \frac{1}{N+1}$ then Λ is uniformly distributed in $\mathcal{M}_{M,N+1}$, the space of possible $M \times N+1$ match matrices. The posterior density of τ and Λ conditional on X is

$$\pi(\tau,\Lambda|X,\mu) = \frac{\pi(\tau)\pi(\Lambda)L(X|\Lambda,\tau,\mu)}{\sum_{\Lambda}\int_0^{\infty}\pi(\tau)\pi(\Lambda)L(X|\Lambda,\tau,\mu)\mathrm{d}\tau}.$$

2.4 MCMC Inference

The full conditional distribution of τ is available from the conjugacy of the Gamma distribution,

$$(\tau|X,\Lambda,\mu) \sim \Gamma\left(\alpha_0 + \frac{Q}{2}, \beta_0 + \frac{d_S(X^{\Lambda},\mu^{\Lambda})^2}{2}\right),$$

so we update τ with a Gibbs step.

We make updates to the match matrix using a Metropolis-Hastings step. We select a row at random and move the 1 to a new position in $[1,\ldots,N+1]$. In particular, if the selected point is already matched then it becomes unmatched with probability p_{reject}, or it is matched to another point i with probability $(1-p_{reject})/(N-1)$. If the selected point is unmatched then it becomes matched to point i with probability $1/N$.

We accept the new proposal, Λ^*, with probability

$$\alpha_\Lambda = \min\left\{1, \frac{\pi(\Lambda^*|X,\mu,\tau)q}{\pi(\Lambda|X,\mu,\tau)q^*}\right\},$$

where

$$q/q^* = \begin{cases} p_{reject}/(1/N) & \text{if making an unmatched point matched,} \\ (1/N)/p_{reject} & \text{if making a matched point unmatched,} \\ 1 & \text{if making a matched point match to a different point.} \end{cases}$$

If $p_{reject} = 1/N$ then $q/q^* = 1$, which was the value used by Dryden et al. (2007).

Dryden et al. (2007) also describe a computationally faster approximate Metropolis-Hastings update to the match matrix which does not require the use of the whole configuration in the calculation of the density. If we propose the change $(i \to l_1)$ to $(i \to l_2)$ then the alternative Hastings ratio, α^*_Λ is given by

$$\alpha^*_\lambda = \min\{g(x_i,\mu_{l_2})q/(g(x_i,\mu_{l_1})q^*),1\}, \qquad (1)$$

where

$$g(x_i,\mu_j) = \begin{cases} \frac{1-\psi}{N}\left(\frac{\tau}{2\pi}\right)^{m/2}\exp\left(-\frac{\tau}{2}|x_i-\mu_j|^2\right), & \text{if } j < N+1 \\ \psi\frac{1}{|\mathcal{A}|} & \text{if } j = N+1. \end{cases}$$

When a new match is accepted the ordinary partial Procrustes registration is carried out on the new matching points to ensure the configuration of matching points has rotation removed.

For brevity we shall refer to the size-and-shape model as the “Procrustes model”, and matching using MCMC simulation with this model as the “Procrustes method”.

2.5 Improving the Procrustes algorithm

One of the problems with the MCMC scheme is that because of the multimodality of the likelihood function for the match matrix Λ, the molecules often get stuck in a local mode. In order to circumvent this problem Dryden et al. (2007) ran the algorithm from a number of different start points until the algorithm had reached a position which satisfied certain convergence criteria.

We propose a new initialization algorithm which involves proposing much more radical changes to the match matrix than changing just one row. The four types of bigger moves are called 'nearness','rotation','translation' and 'flip'. All four types of proposal are non reversible, and therefore we only allow these big jumps at the start of the MCMC algorithm. Effectively the use of these proposals helps to find a good starting point for the subsequent MCMC inference. The new moves are:

1. **Nearness.** Each of the matched points in X (i.e. those rows of Λ that have a 0 in the last column) is matched to the point in μ that is nearest to it. Let I^{Λ} be the index of matched points, so $I^{\Lambda} = \{i \in \{1, 2, \ldots, M\} : \lambda_{i,N+1} = 0\}$. We define $\Lambda^* = (\lambda^*_{ij})$ by

$$\lambda^*_{ij} = \begin{cases} 1 & i \notin I^{\Lambda}, j = N+1 \\ 1 & i \in I^{\lambda}, \parallel (X)_i - (\mu)_j \parallel = \min_{l \in \{1,\ldots,N\}} \parallel (X)_i - (\mu)_l \parallel \\ 0 & \text{otherwise.} \end{cases}$$

   Let $N(X, \mu, \Lambda) = \Lambda^*$ as defined above. Note that Λ^* has the same number of matched points as Λ. The other three methods (rotation, translation and flip) use this nearness step at the end.

2. **Rotation.** Randomly choose an angle $\theta \sim U[-\pi, \pi]$. Randomly choose an axis (x, y or z) about which to rotate, and set $R = R_x(\theta), R_y(\theta)$ or $R_z(\theta)$ as appropriate, where R_x, R_y, R_z are defined in (2) and (3) below. Let $X^* = XR$ then map each point in X^* to the nearest point in μ, i.e. $\Lambda^* \equiv R^{\Lambda}(X, \mu, \Lambda) = N(X^*, \mu, \Lambda)$.

3. **Translation.** Choose $\gamma \sim N_3(0, \sigma^2)$. Define $X^* = X + 1_M \gamma^T$ and then map each point in X^* to its nearest point in μ. Thus $\Lambda^* \equiv T^{\Lambda}(X, \mu, \Lambda) = N(X^*, \mu, \Lambda)$.

4. **Flip.** This move has the same form as the rotation step, but instead of selecting θ from a $U[-\pi, \pi]$ distribution we set $\theta = \pi$.

We define an initialization phase by setting a maximum number of initial jumps, $N_{initialization}$. We also define a settling time, N_{settle}. During the initialization phase (i.e. $\leq N_{initialization}$ interactions) at least N_{settle} default updates are proposed between any two big jump proposals. The rationale behind this is to explore the region of the parameter space we 'land in' after making a big jump before immediately jumping somewhere else. The hope is that the settling time allows the algorithm to home in on a solution if a big jump takes us somewhere close to the optimal solution. Provided at least N_{settle} default updates have been proposed we randomly choose an update type from {nearness, rotation, translation, flip, default}, with probabilities $p_n, p_r, p_t, p_f, 1 - (p_n + p_r + p_t + p_f)$, say. Whichever update method is chosen, a new match matrix, Λ^* is generated. We then accept the new match matrix with probability

$$\alpha_{\Lambda} = \min\{1, \pi(\Lambda^*|X, \mu, \tau)/\pi(\Lambda|X, \mu, \tau)\}.$$

After $N_{initialization}$ iterations the algorithm proceeds exactly as described in Dryden et al. (2007).

Other work where large jumps have been used to help address multimodality includes Tjelmeland and Hegstad (2001) and Tjelmeland and Eidsvik (2004). Their scenarios are simpler where they can make use of local optimization to construct reversible jumps between modes. In our case we are not carrying out local optimization involving gradients, but rather we use the nearness of points in physical space for constructing a large jump into a different mode in this large discrete combinatorial search space of matchings. It is not clear how to make such moves reversible, and so we just carry them out during the initialization phase.

3 Configuration model

3.1 Likelihood

We now consider an alternative model for the configuration of points which is very similar to that of Green and Mardia (2006). We again assume that μ is a fixed $N \times m$ configuration and X is an $M \times m$ configuration that we apply rigid-body transformations to.

This model for the co-ordinates of the points does not involve removing rotation and translation by Procrustes matching. Rather, the rotation matrix $\Gamma \in SO(m)$ and the translation parameter γ will be parameters in the model. The matched points in X^{Λ} are taken as Gaussian perturbations of the matching points in μ, and we assume that the rows of $X^{-\Lambda}$ are distributed uniformly over a bounded region $\mathcal{A} \subset \mathbb{R}^m$ of volume $|\mathcal{A}|$. We concentrate on the $m = 3$ dimensional case here.

Given an $M \times (N+1)$ match matrix, Λ (with p matching points), rotation matrix Γ and translation vector γ the likelihood is therefore defined as:

$$L^*(X|\Lambda, \mu, \tau, \Gamma, \gamma) = \left(\frac{1}{2\pi}\right)^{3p/2} \frac{\tau^{3p/2}}{|\mathcal{A}|^{M-p}} \exp\left(-\frac{\tau}{2}\text{trace}\{(\tilde{X}^{\Lambda} - \mu^{\Lambda})^T(\tilde{X}^{\Lambda} - \mu^{\Lambda})\}\right),$$

where $\tilde{X}^{\Lambda} = X^{\Lambda}\Gamma + 1_p\gamma^T$, $\Gamma = R_z(\theta_{12})R_y(\theta_{13})R_x(\theta_{23})$, and the rotation matrices about the x, y, z axes are:

$$R_x(\theta_{23}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_{23} & \sin\theta_{23} \\ 0 & -\sin\theta_{23} & \cos\theta_{23} \end{pmatrix}, \quad R_y(\theta_{13}) = \begin{pmatrix} \cos\theta_{13} & 0 & \sin\theta_{13} \\ 0 & 1 & 0 \\ -\sin\theta_{13} & 0 & \cos\theta_{13} \end{pmatrix}, \quad (2)$$

$$R_z(\theta_{12}) = \begin{pmatrix} \cos\theta_{12} & \sin\theta_{12} & 0 \\ -\sin\theta_{12} & \cos\theta_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (3)$$

with Euler angles $\theta_{12} \in [-\pi, \pi), \theta_{13} \in [-\pi/2, \pi/2], \theta_{23} \in [-\pi, \pi)$. There are many choices of Euler angle representations and all have singularities (Stuelpnagel, 1964), although the singularities have measure zero with respect to Haar measure which is given by

$$\frac{1}{8\pi^2}\cos(\theta_{13})d\theta_{12}d\theta_{13}d\theta_{23}$$

in this case (e.g. see Khatri and Mardia, 1977).

Note that Green and Mardia (2006)'s model is constructed with X and μ as Gaussian perturbations from an underlying Poisson process. However, the likelihood is actually of the same form as the one sided version, where X is perturbed from μ, although the variance parameter is doubled.

3.2 Prior and posterior distributions

We take $\tau, \Lambda, \Gamma, \gamma$ to be mutually independent *a priori*, and the priors of τ and Λ are taken as in Section 2.3. We also take the prior for γ as:

$$\gamma \sim N_3(\mu_\gamma, \sigma_\gamma^2 I_3),$$

and we take Γ to be uniform with respect to Haar measure on $SO(m)$. The posterior density of $(\Lambda, \tau, \Gamma, \gamma)$ conditioned on X is

$$\pi(\tau, \Lambda, \Gamma, \gamma | X, \mu) = \frac{\pi(\tau)\pi(\Lambda)\pi(\Gamma)\pi(\gamma)L(X|\Lambda, \tau, \mu, \Gamma, \gamma)}{\sum_\Lambda \int_0^\infty \pi(\tau)\pi(\Lambda)\pi(\Gamma)\pi(\gamma)L(X|\Lambda, \tau, \mu, \Gamma, \gamma)\mathrm{d}\tau}.$$

3.3 MCMC simulation

The full conditional distribution of τ is given by

$$(\tau | X, \Lambda, \Gamma, \gamma, \mu) \sim \Gamma\left(\alpha_0 + \frac{3p}{2}, \beta_0 + \frac{\|\tilde{X}^\Lambda - \mu^\Lambda\|^2}{2}\right),$$

and so a Gibbs update can be used for τ.

We update the rotation angles using a Metropolis-Hastings step, drawing the proposal perturbations from a uniform distribution on $[-0.2, 0.2]$ for θ_{12}, θ_{23}, and uniform on $[-0.1, 0.1]$ for θ_{13}, to give proposed angles $\theta_{12}^*, \theta_{13}^*, \theta_{23}^*$. The Hastings ratio is:

$$\min\left(1, \frac{\pi(\tau, \Lambda, \Gamma(\theta_{12}^*, \theta_{13}^*, \theta_{23}^*), \gamma | X, \mu) \cos\theta_{13}^*}{\pi(\tau, \Lambda, \Gamma(\theta_{12}, \theta_{13}, \theta_{23}), \gamma | X, \mu) \cos\theta_{13}}\right),$$

and the extra cosine terms are due to the Haar measure on the special orthogonal rotation group.

The full conditional distribution of γ is given by

$$\gamma | X, \mu, \tau, \Lambda, \Gamma \sim N\left(\frac{\mu_\gamma/\sigma_\gamma^2 + \tau \sum_{j \le M, k \le N, \lambda_{jk}=1}(\mu_k - x_j\Gamma)}{p\tau + 1/\sigma_\gamma^2}, \frac{1}{p\tau + 1/\sigma_\gamma^2} I_3\right), \tag{4}$$

and so we use a Gibbs update for γ.

We update the match matrix Λ in the same way as in the Procrustes model using the acceptance probability

$$\begin{aligned}\alpha_\Lambda &= \min\left(1, \frac{\pi(\Lambda^*|X,\mu,\tau,\Gamma,\gamma)q}{\pi(\Lambda|X,\mu,\tau,\Gamma,\gamma)q^*}\right) \\ &= \min\left(1, \frac{L(X|\Lambda^*,\mu,\tau,\Gamma,\gamma)\pi(\Lambda^*)q}{L(X|\Lambda,\mu,\tau,\Gamma,\gamma)\pi(\Lambda)q^*}\right).\end{aligned}$$

Suppose Λ^* contains the match $(i \longrightarrow l_1)$ and Λ contains the match $(i \longrightarrow l_2)$, where $l_1 \neq l_2$ and the match matrices Λ^* and Λ are otherwise identical. The acceptance probability α_Λ is exactly the same as that given in Equation (1), i.e. the fast method of Dryden et al. (2007). Hence the MCMC updates of Λ for the Procrustes and Configuration models are more similar than they first appear.

Note that our implementation of the MCMC simulation differs slightly from Green and Mardia (2006) who use a matrix Fisher conjugate prior for the rotation, and update two of the rotation angles with a Gibbs step. In addition, Green and Mardia (2006) ensure that the matching is 1-1 between the points, whereas we do allow the possibility of many-to-one matches.

For brevity we shall refer to this model as the "Configuration model", and matching using this model as the "Configuration method". The Configuration model has been demonstrated to work well in a variety of situations (Mardia et al. 2007).

4 Applications and simulations

4.1 Assessment of initialization procedure

Here, we use the NADP-binding site protein data to assess the efficiency of the Procrustes algorithm, both with and without the large jump proposals. There are 40 centers of gravity of amino acids for protein 1 and 63 for protein 2. Following Green and Mardia (2006) we take the prior hyperparameters to be $\alpha_0 = 1, \beta_0 = 36, \mu_\gamma = 0, \sigma_\gamma = 50$, and we take $\psi = 0.2$. The proposal parameters $p_n = 0.001, p_r = 0.02, p_f = 0.01, p_t = 0.09, N_{settle} = 850$ for this application. For the volume parameter we use $|\mathcal{A}| \approx 23000$ which is the volume of a bounding box obtained by multiplying the maximum lengths in the x, y, z directions for each protein.

We used the *a priori* 'correct' matches, as identified in Green and Mardia (2004) to define a convergence criterion. Although we use the word 'correct' match, note that these are not ground truth matches but rather these top 36 matches are very reliably obtained by the matching algorithms, and so it is in this sense that we use the word 'correct'.

To assess the efficacy of the criterion for determining convergence, we started 50 MCMC runs from distinct initial configurations in each of which 10 correct matches were selected at random. Each run was allowed to run for 10000 iterations, and we measured the number of correct matches after each 100 iterations. The results are shown in Figure 1 and we define convergence to have been reached when 30 'correct' matches have been obtained and $\sigma < 1$. The plot also shows a mean number of ‘correct’ matches, where after convergence the value of 30 ‘correct’ matches is used in the calculation. It is interesting to note that the Procrustes and Configuration methods converge at a similar rate here.

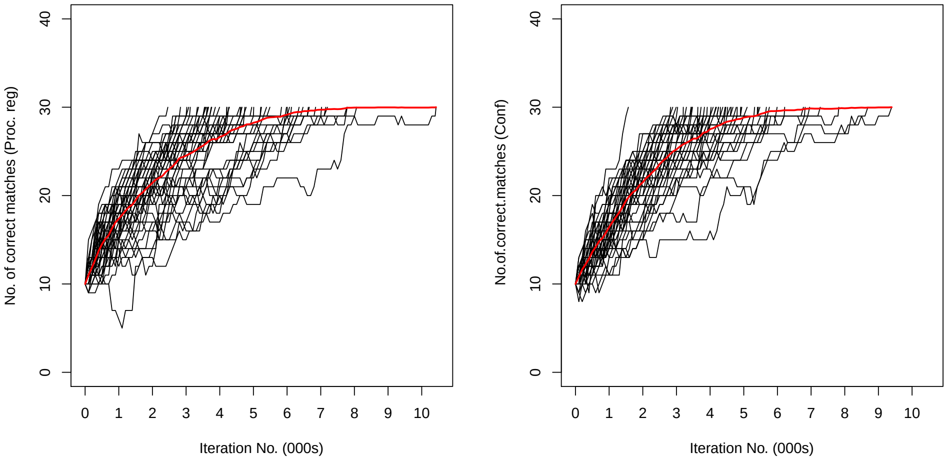

Figure 1: A comparison of the numbers of ‘correct’ matches against iteration number (using the 36 ‘correct matches from Green and Mardia (2004)) for the Procrustes (left) and Configuration (right) models, initializing by choosing 10 correct matches at random. We define convergence as 30 ‘correct matches and $\sigma < 1$, and each trace is plotted until convergence has been reached. The red lines show the mean number of correct matches at each iteration number, using 30 ‘correct’ matches after convergence has been reached.

Also, rather than starting with 10 ‘known’ matches we have rerun the algorithms starting from just two matches selected from the ‘correct’ matches or from two randomly selected matches, which is comparable to assuming no matches. With Procrustes registration, the algorithm converged 12/100 times from two randomly selected points, and 54/100 times from two ‘correct’ matches. For the Configuration method, the algorithm converged 12/100 times from two randomly selected points, and 36/100 times from two ‘correct’ matches. Hence, it can be seen in this application that it is helpful to initialize by assuming some of the ‘correct’ matches.

To compare further the convergence performance of the Procrustes and Configuration methods, we initiated 25 runs from random starting points. We allowed each run to continue for a maximum of a million iterations, monitoring the number of correct matches after every thousand iterations. On the basis of the results described above, we stipulated that if within these million iterations the number of correct matches reached 10 then that counted as convergence. Such runs were allowed to continue for a further 50000 iterations. The Procrustes method was used both with and without the big jumps described above; these were only used during the initial $N_{initialization} = 1000000$ iterations.

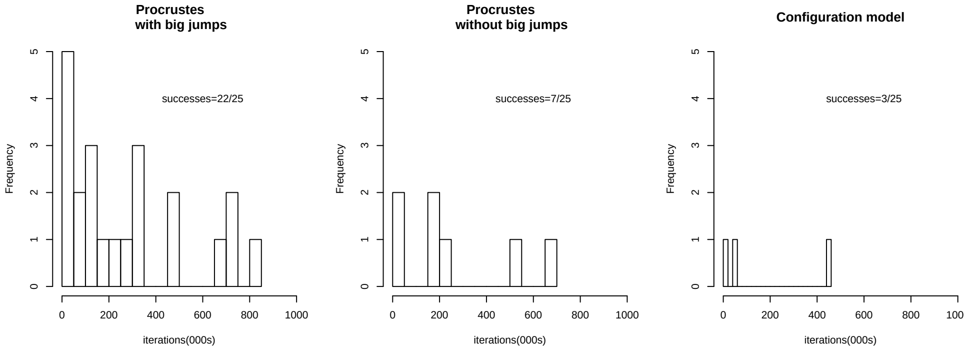

Figure 2: Histograms of the number of iterations to convergence (defined as 10 'correct' matches and $\sigma < 1$ from random starting points, i.e. two matches selected at random) for the size-and-shape model, with and without large jumps, and the configuration model. The parameters are: $\alpha_0 = 1, \beta_0 = 36, \psi = 0.5, \sigma_T = 2.2, p_n = 0.001, p_r = 0.02, p_f = 0.01, p_t = 0.09, N_{settle} = 850$.

Figure 2 shows histograms of the number of iterations before the algorithms converged to 10 correct matches for the successful runs. The success rates of 7/25 for the Procrustes method without big jumps and 3/25 for the Configuration method were not too encouraging. However, when big jumps were included for the Procrustes method, the success rate increased to 22/25, a very impressive result.

Green and Mardia (2006) report convergence within a million iterations on 83 out of 100 tests run from random starting points. They define convergence differently to us, looking for runs in which the log-posterior goes higher than some threshold. It is important to note three things when looking at this result and comparing it with the results of Figure 2. Firstly, in the Green and Mardia paper, they update the match matrix 10 times per sweep, so they are effectively looking at the convergence within 10 million iterations. Secondly, their proposal methods for the angles in particular are different; they use Gibbs steps instead of Metropolis-Hastings updates, making use of conjugacy of the matrix Fisher distribution. This may also improve their convergence performance, with the form of the proposals being closer to the true distribution. Finally, the way the model is formulated is different, with 1-1 matches and a hidden Poisson process being used.

Although the algorithm was much more likely to converge within a million iterations if the big jumps were included, it did mean that from certain starting points the algorithm took a lot longer to converge if the big jumps were included than if they were not.

This is a consequence of the choice of the settling time parameter between large jump proposals. One way to avoid this might be to let the algorithms run for an initial period of 100000, say, before introducing any big jumps. This way, if the algorithm converged within that period then it would not be necessary to use the big jumps at all. Also, the settling time between large jump proposals could be increased. Despite the fact that it often took longer for the algorithm to converge with the large jumps, the evidence is compelling that the big jumps vastly improve convergence.

We experimented with the probabilities of acceptance for the four types of large jumps. At the levels we settled on (given in the caption of Figure 2) the nearness proposal was always accepted (which is always the case since the likelihood always increases for the nearness proposal), and the other three types were accepted roughly a quarter (flip), a third (rotation) and half (translation) of the times when they were proposed.

4.2 Long run comparisons

In order to compare further the Procrustes and Configuration algorithms we apply the MCMC scheme from a number of long runs of the method. In order to ensure that we started the algorithms close to convergence, we initialized the proteins by aligning the first 10 pairs of amino acids as given in Table 4 of Green and Mardia (2004).

We ran the two algorithms and looked at the proportion of the accepted match matrices after convergence in which particular matches were represented. Although in principle many to one matches were possible, they did not tend to occur in the long runs after convergence. We ran the experiment for five values of ψ, the prior probability of a particular point being unmatched, and five values of the proposal probability p_{reject}, the probability of moving a matched point to an unmatched status in the proposal for the change to the match matrix. For each parameter the five values we used were $0.001, 1/63, 0.1, 0.2$ and 0.4. (The $1/63$ is there because $N = 63$ and in the case of ψ, this corresponds to a uniform prior for Λ.)

Altering p_{reject} had little effect on the results. We fix $p_{reject} = 0.2$ and consider the effects of varying ψ, the prior probability of each point in protein 1 being unmatched (independently of the other points). We ran each MCMC algorithm for 1000000 iterations after convergence, adding the match matrices together.

In Figure 3, we show how often the 36 most likely matches from Table 4 of Green and Mardia (2004) appear in our match matrices after convergence. These percentages are calculated as the number of times each match occurred divided by the total number of match matrices.

We have calculated a 'threshold match matrix' by putting a 1 in each position that corresponds to the maximum entry in a row of the summed match matrices and a 0 everywhere else. This gives us a method for comparing how many points are matched for each value of ψ. For values of $\psi \in \{0.001, 1/63, 0.1, 0.2, 0.4\}$ the number of unmatched points are $\{0, 0, 1, 4, 4\}$ respectively, for both the Procrustes and Configuration methods.

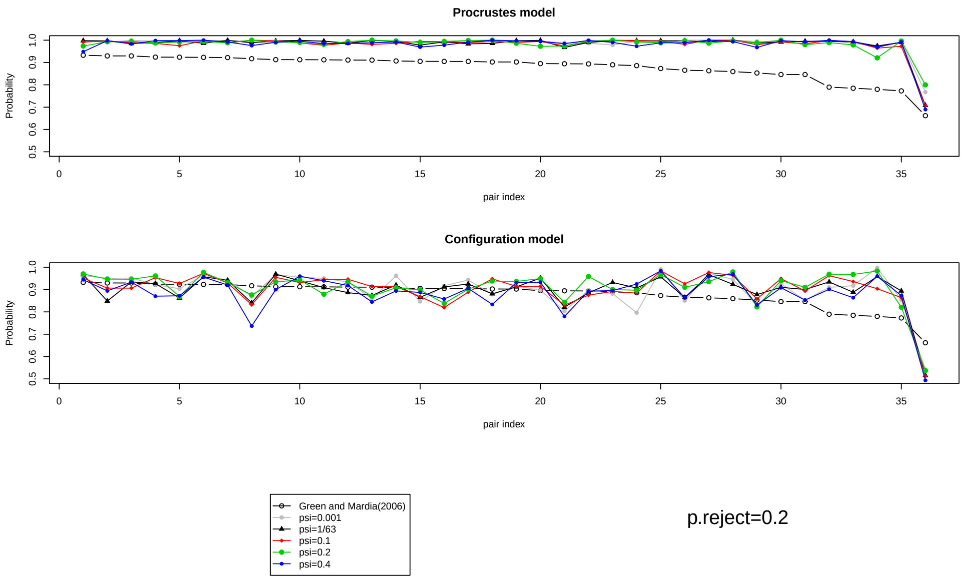

Figure 3: The proportions of match matrices containing particular pairings, based on 1000000 iterations after convergence for the Procrustes and Configuration models for five values of ψ - a comparison with the percentages quoted in Green and Mardia (2006).

Clearly changing the prior distribution of Λ by altering ψ has an effect on the number of points that are matched.

Figure 3 shows that using the Configuration model, we obtain probabilities for the top 36 matches reported in Green and Mardia (2006) that are similar to the figures quoted in that paper. However, using the Procrustes model, the probabilities are all significantly closer to 1 than using the Configuration model. This suggests that the Procrustes model is 'stickier' than the Configuration model, in the sense that matches are released less readily after convergence. The simulation study below investigates the relationship of long run convergence probabilities with different variances, and the results suggest that there is a possibility that the results observed in Figure 3 may be a contingent property of the variability of the points. We return to this in the discussion of the simulation study.

Note that the posterior standard deviation $\sigma = 1/\sqrt{\tau}$ was smaller for the Procrustes model. In particular, the means of the 10000 values well after burn-in were 0.869 for the Procrustes model and 1.355 for the Configuration model.

We have used R (R Development Core Team 2011) for coding, and so this could be speeded up considerably using C or C++ for example. With the Procrustes method, a million iterations take approximately 27 minutes on a standard desktop PC. For the Configuration method, a million iterations take approximately 1 hour 30 minutes. Thus the method with Procrustes registration is approximately three times faster than the Configuration method in our implementation.

4.3 A simulation study

We consider now a simulation study where we know what the true probabilities of matching are and compare the MCMC algorithms both with and without Procrustes registration to see how they perform. The details of this simulation are as follows:

Step 1 Define a length, $L > 0$ and a minimum distance $0 < d_{min} < L$. Fix $M, N \in \mathbb{N}$, $n_{ones} < M$. As before, M is the number of points in the point set X and N is the number of points in the point set μ. Define a vector of probabilities, $p = (p_1, p_2, \ldots, p_M)$, where $p_1 = p_2 = \ldots = p_{n_{ones}} = 1$ and $p_i = 0$ for $i = n_{ones} + 1, \ldots, M$. Fix $s < d_{min}$; this is the standard deviation of the perturbations of the random points.

Step 2 Sample the N points of μ from a uniform distribution on the cube with corners

$$\{(-L, -L, -L), (-L, -L, L), \ldots, (L, L, L)\}$$

subject to the constraint that each new point is at least a distance d_{min} from every other point. For the i^{th} point in X, denoted $(X)_i$, if $p_i = 1$ then we sample from a Normal distribution centered on the i^{th} point in μ,

$$(X)_i \sim N_3((\mu)_i, s^2 I_3),$$

else we sample uniformly from the cube with corners as above,

$$(X)_i \sim U[\text{cube as above}].$$

Step 3 Run the two MCMC algorithms for N_{iter} iterations starting from the match matrix which matches $(X)_j$ to $(\mu)_j$ for $j = 1, 2, \ldots, n_{ones}$. (In other words we start the algorithms from convergence.) For $i = 1, \ldots, n_{ones}$, record the proportion of the N_{iter} match matrices that match $(X)_i$ to $(\mu)_i$. For $i = n_{ones} + 1, \ldots, M$, record the proportion of the N_{iter} match matrices for which $(X)_i$ is unmatched.

Step 4 Hold μ constant and sample a new X as described in step 2. Repeat step 3. Continue this process until the proportions of successful matches and successfully unmatched points have been recorded for K runs of the MCMC algorithm.

Step 5 Repeat experiment for various values of $s < d_{min}$.

Figure 4 shows the results of running this experiment with $M = 20$, $N = 24$ and $n_{ones} = 12$. The values chosen for L and d_{min} were 10 and 2 respectively. The experiment was run for four values of s, the standard deviation parameter. These were

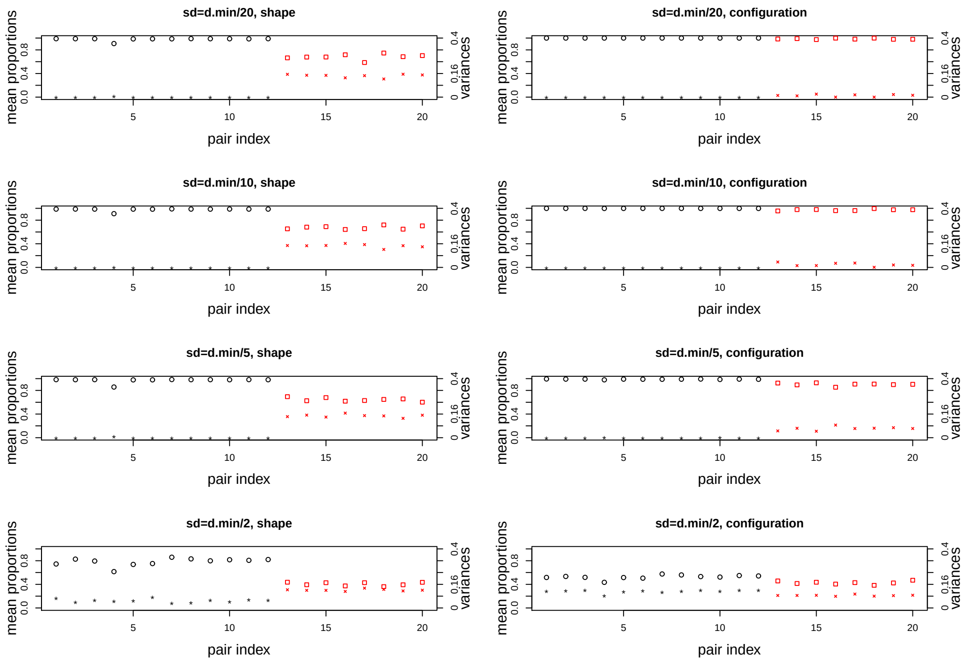

Figure 4: The means (circles and squares, left hand scale) and variances (stars, right hand scale) of the proportions of successful matches (black circles and black small stars) and successfully unmatched points (red squares and red large stars) for the Procrustes model (top row) and Configuration model (bottom row) on long runs (100000 iterations) after convergence. Here, there are 20 points in configuration X and 24 points in configuration μ. The points in μ are sampled uniformly from a cube of side length 20 subject to the constraint that they are a minimum distance $d_{min} = 2$ from the nearest neighbor. The first 12 points in X are sampled from Normal distributions centered at the corresponding points in μ and the last 8 points in X are sampled uniformly on the cube of radius $2L$. The means and variances are calculated over 100 runs, with μ held constant and X resampled each time.

$d_{min}/20, d_{min}/10, d_{min}/5, d_{min}/2$, or 0.1, 0.2, 0.4 and 1. The value of N_{iter}, the number of iterations after convergence, was 100000.

Figure 4 has a curious feature. When the value of the standard deviation is less than or equal to $d_{min}/5$, the Configuration model seems to estimate the probabilities for both matched and unmatched points more reliably than the Procrustes model. For both models the matched points are rarely released when the matching is very precise, but the Configuration model gives probabilities closer to 1 than the Procrustes model. (This is not clear from just looking at the graphs). When the standard deviation is increased to $d_{min}/2$, the Configuration model still performs better than the Procrustes model on the unmatched points. Interestingly, now the Procrustes model gives significantly better (i.e. higher) estimates for the probabilities for the matched points.

With reference to the results illustrated in Figure 3, this simulation study poses an interesting question. In Figure 3, we found that the Procrustes method appeared 'stickier' than the Configuration method. In the light of the findings of this simulation study, it is possible that this result is a feature of the particular relationship between the variance parameter and the minimum distance between points in this particular dataset. From the simulation, it appears there may be a critical value of the standard deviation parameter, somewhere between $d_{min}/5$ and $d_{min}/2$, for which the two MCMC methods swap over in terms of which one gives the higher probabilities for particular matches.

4.4 Laplace approximation

We have observed in all our simulation studies and practical applications that neither the Procrustes nor the Configuration models are uniformly superior. In fact, closer inspection indicates that the two models are more similar than they first appear.

Let us consider the posterior density $\pi(\Lambda, \tau, \Gamma, \gamma | X)$. Note that the rotation and translation Γ, γ are nuisance parameters, and one has a choice about how to deal with them. In the Configuration approach one samples from the full joint distribution of $(\Lambda, \tau, \Gamma, \gamma | X)$ and so joint inference of all the parameters can be carried out. However, if Γ, γ are considered nuisance parameters then we can integrate them out to give the marginal density of (Λ, τ)

$$\pi_C(\Lambda, \tau | X) = \int_{\Gamma, \gamma} \pi(\Lambda, \tau, \Gamma, \gamma | X) d\Gamma d\gamma. \tag{5}$$

In the Procrustes approach the match is obtained by optimizing over the nuisance parameters, and so we consider the different posterior density based on

$$\pi_P(\Lambda, \tau | X) \propto \sup_{\Gamma, \gamma} \pi(\Lambda, \tau, \Gamma, \gamma | X). \tag{6}$$

We can consider (6) to be an approximation to the marginal density (5) where the integral is approximated using Laplace's method (Tierney and Kadane 1986).

From a Bayesian analysis perspective it is natural to work with the marginal posterior distribution (5). From a shape theory perspective the analysis should be invariant under rotations or translations of the data, and so a uniform prior for Γ, γ in (5) or a distribution of the form (6) are both natural.

Regardless of which model is used we have seen generally similar performance in a wide range of scenarios. In situations where the posterior distribution of Γ, γ is close to multivariate normal then we expect the Laplace approximation in (6) to be accurate, and hence this provides an explanation for the similarity of the two models in many practical situations.

5 Discussion

In conclusion, it is clear that the Procrustes method is significantly improved by considering the initial large jumps. However, despite quite extensive comparisons there is not an overall preference between the Procrustes or Configuration methods for all situations. The Procrustes and Configuration methods appear to converge similarly reliably to the true solution when the proteins are initialized by selecting 10 correct matches at random. For simulated datasets where the variance is small, the Configuration method more reliably predicts the probabilities of matches, and the Procrustes method was more likely to suffer from false matches. For larger variances the Procrustes method was more effective at estimating correct matches, without more false matches. However, in essence both models are fairly similar, and inference using marginal posteriors (5) or (6) is similar in practice due to the Laplace approximation. These observations are relevant to a wider discussion of whether to specify distributions directly in a quotient space (as for the Procrustes model) or in a configuration space, and the appropriate choice will depend on the application and the objectives of the analysis.

The way we have set up the MCMC procedures, we do not exclude the possibility of many-to-one matches. We have followed the methodology of Dryden et al. (2007) and found that in general many-to-one matches are not selected in long runs after convergence. However, it would be easy to constrain the choice of match matrices such that only one-to-one matches were proposed. This is the method adopted by Green and Mardia (2006).

MCMC tools are an effective way of finding the optimal correspondence and registration between two point sets where we wish to match a subset of points from one set to a subset of points from the other set. However, because of the combinatoric nature of looking for possible correspondences, the algorithms are currently prohibitively time consuming for large data sets (e.g. more than a few hundred points). Suppose we were interested in comparing two large protein surfaces to look for regions of a similar shape (such as binding sites that are common to both proteins). It may be possible to use an efficient search algorithm to scan the surface of the two proteins for small regions that are potential candidates for binding sites and then apply the MCMC methods to those small sites individually to confirm whether or not there are subsets of the two regions that match well. Schmidler (2007) notes the difficulties of using MCMC methods for large problems and suggests the use of geometric hashing to compute approximate posterior quantities efficiently.

Although we have just considered pairwise matching of two configurations here, the methods extend to matching multiple molecules. Extensions of the Procrustes and Configuration models for multiple alignments have been given by Dryden et al. (2007), Ruffieux and Green (2009) and Mardia et al. (2011)

References

Dryden, I. L. (2011). `shapes` *package*. R Foundation for Statistical Computing, Vienna, Austria. Contributed package.
URL http://www.R-project.org 549

Dryden, I. L., Hirst, J. D., and Melville, J. L. (2007). "Statistical analysis of unlabeled point sets: comparing molecules in cheminformatics." *Biometrics*, 63: 237–251. 547, 548, 549, 550, 551, 552, 555, 563

Dryden, I. L. and Mardia, K. V. (1992). "Size and shape analysis of landmark data." *Biometrika*, 79: 57–68. 549

— (1998). *Statistical Shape Analysis*. Chichester: Wiley. 549

Green, P. J. and Mardia, K. V. (2004). "Bayesian alignment using hierarchical models, with applications in protein bioinformatics." Technical report, University of Bristol. ArXiv:math/0503712v1. 548, 555, 556, 558

— (2006). "Bayesian alignment using hierarchical models, with applications in protein bioinformatics." *Biometrika*, 93: 235–254. 547, 548, 553, 554, 555, 557, 559, 563

Kendall, D. G. (1989). "A Survey Of The Statistical Theory Of Shape (with discussion)." *Statistical Science*, 4: 87–120. 549

Kent, J. T. and Mardia, K. V. (2001). "Shape, tangent projections and bilateral symmetry." *Biometrika*, 88: 469–485. 549

Khatri, C. G. and Mardia, K. V. (1977). "The von Mises-Fisher matrix distribution in orientation statistics." *Journal of the Royal Statistical Society. Series B.*, 39(1): 95–106.

Mardia, K., Nyirongo, V., Green, P., Gold, N., and Westhead, D. (2007). "Bayesian refinement of protein functional site matching." *BMC Bioinformatics*, 8: 257. 555

Mardia, K. V., Nyirongo, V. B., Fallaize, C. J., Barber, S., and Jackson, R. M. (2011). "Hierarchical Bayesian Modeling of Pharmacophores in Bioinformatics." *Biometrics*, 67: 611–619. 563

Moss, S. and Hancock, E. R. (1996). "Registering Incomplete Radar Images using the EM Algorithm." In Fisher, R. B. and Trucco, E. (eds.), *Proceedings of the Seventh British Machine Vision Conference*, 685–694. British Machine Vision Association. 547

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.
URL http://www.R-project.org/ 559

Rangarajan, A., Chui, H., and Bookstein, F. L. (1997). "The Softassign Procrustes Matching Algorithm." In Duncan, J. and Gindi, G. (eds.), *Information Processing in Medical Imaging*, 29–42. Springer. 547

Ruffieux, Y. and Green, P. J. (2009). "Alignment of multiple configurations using hierarchical models." *Journal of Computational and Graphical Statistics*, 18: 756–773. 563

Schmidler, S. C. (2007). "Fast Bayesian Shape Matching Using Geometric Algorithms." In Bernado, J. M., Herckerman, D., Berger, J. O., and Dawid, A. P. (eds.), *Bayesian Statistics*, 8. Oxford University Press. 547, 548, 549, 550, 563

Stuelpnagel, J. (1964). "On the parametrization of the three-dimensional rotation group." *Society for Industrial and Applied Mathematics Review*, 6: 422–430.

Tierney, L. and Kadane, J. B. (1986). "Accurate approximations for posterior moments and marginal densities." *Journal of the American Statistical Association*, 81(393): 82–86. 562

Tjelmeland, H. and Eidsvik, J. (2004). "On the use of local optimizations within Metropolis-Hastings updates." *Journal of the Royal Statistical Society. Series B.*, 66(2): 411–427. 553

Tjelmeland, H. and Hegstad, B. K. (2001). "Mode jumping proposals in MCMC." *Scandinavian Journal of Statistics.*, 28(1): 205–223. 553

Acknowledgments

The authors are grateful to two anonymous referees for their helpful comments on the paper. The work was supported by an EPSRC studentship.