Appendix to "Fast approximate Bayesian inference for small-area estimation of HIV indicators using the Naomi model"

Adam Howes* Alex Stringer† Seth R. Flaxman‡ Jeffrey W. Eaton§

Contents

S1 Simplified Naomi model description **2**
S1.1 Background . . . 2
S1.2 Process specification . . . 3
S1.3 Likelihood specification . . . 5
S1.4 Identifiability constaints . . . 6

S2 C++ `TMB` user template **8**

S3 MCMC convergence and suitability **22**

S4 Laplace marginals algorithm **23**

References **25**

*Department of Mathematics, Imperial College London
†Department of Statistics and Actuarial Science, University of Waterloo
‡Department of Computer Science, Oxford University
§Department of Infectious Disease Epidemiology, Imperial College London

S1 Simplified Naomi model description

In this section we describe the simplified Naomi model (Jeffrey W. Eaton et al. 2021) considered in the main text in more complete detail.

S1.1 Background

S1.1.1 Indexing

Consider the most recent national household survey with HIV testing which has taken place in the country of interest. Let $x \in \{1, \ldots, n\}$ refer to a district located within the Spectrum (Stover et al. 2010) region R_x. Let $s \in \{\text{F}, \text{M}\}$ be sex, and $a \in \{\text{0-5}, \text{5-10}, \ldots, \text{75-80}, \text{80+}\}$ be five-year age groups. As short-hand, we write $a = l$ to refer to the age group with lower bound l, e.g. $a = 20$ for $a = \text{20-25}$. We index the known quantity population size $N_{x,s,a}$ by district, sex and age-band, as well as the following unknown quantities: HIV prevalence $\rho_{x,s,a} \in [0, 1]$, ART coverage $\alpha_{x,s,a} \in [0, 1]$, annual HIV incidence rate $\lambda_{x,s,a} > 0$, and proportion of HIV positive persons recently infected $\kappa_{x,s,a} \in [0, 1]$. Sometimes data are observed at an aggregate level, rather than the more granular modelled level. In this instance, we use $\{\cdot\}$ to generically refer to a aggregate set over which an observation is made, e.g. $\{a\} = \{\text{15-19}, \ldots, \text{45-49}\}$ for the adult age range 15-49. We let $\sum_{\{x\}}$ be used as shorthand for $\sum_{x \in \{x\}}$, and likewise for s and a. In the main text we use a more concise, but less descriptive, approach.

S1.1.2 Structured random effects

We use use structured random effects to enable partial pooling of information across units assessed as being similar, such as those belonging to neighbouring districts or adjacent age-bands. Let u be a generic random effect, with length $\dim(u)$. Three types of structured random effects are used in the model. First, we specify the first order auto-regressive model by $u \sim \text{AR1}(\sigma, \phi)$ such that

$$
\begin{aligned}
u_1 &\sim \left(0, \frac{1}{1-\rho^2}\right), \\
u_i &= \rho u_{i-1} + \epsilon_i, \quad i = 2, \ldots, \dim(u)
\end{aligned}
$$

where $\epsilon_i \sim \mathcal{N}(0, 1)$ are independent and identically distributed (IID) Gaussian random variables, $\sigma > 0$ is the marginal standard deviation, and $|\rho| < 1$ is the (lag-one) correlation parameter. Second, we use $u \sim \text{ICAR}(\sigma)$ to refer to the Besag intrinsic conditional auto-regressive model (ICAR) (Besag, York, and Mollié 1991) with full conditionals

$$
u_i \,|\, u_{-i} \sim \mathcal{N}\left(\frac{\sum_{j:j\sim i} u_j}{n_{\delta i}}, \frac{\sigma^2}{n_{\delta i}}\right),
$$

where u_{-i} is u with the ith element removed i.e. $(u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_{\dim(u)})$, $j \sim i$ if the units i and j are defined as adjacent, $n_{\delta i} = |\{j : j \sim i\}|$ is the total number of adjacent units, and $\sigma > 0$ is the marginal standard deviation. We follow recommendations of Freni-Sterrantino, Ventrucci, and Rue (2018) on scaling of precision matrices, disconnected adjacency graph components, and islands. Third, for the reparameterised Besag-York-Mollie model (BYM2) (Simpson et al. 2017), we write $u \sim \text{BYM2}(\sigma, \phi)$, where u is comprised of a spatially structured ICAR component v with proportion $\phi \in (0, 1)$ and spatially unstructured IID component w with proportion $1 - \phi$, both scaled to have generalised variance equal to one, and $\sigma > 0$ is the marginal standard deviation such that

$$
u = \sigma\left(\sqrt{\phi} \cdot v + \sqrt{1-\phi} \cdot w\right).
$$

S1.1.3 Complex survey design

We assume the household survey was run according to a complex survey design where each individual j in the population U has non-zero probability $\pi_j \in (0, 1)$ of appearing in the sample $S \subseteq U$. Suppose we observe an outcome $\theta_j \in \{0, 1\}$ for $j \in S$. Let $w_j = 1/\pi_j \times 1/\omega_j$ be design weights, where ω_j is a non-response factor, then a survey weighted mean is given by

$$
\hat{\theta} = \frac{\sum_{j\in S} w_j \theta_j}{\sum_{j\in S} w_j}.
$$

We account for survey weighting in the variance via the Kish effective sample size (Kish 1965)

$$m^{\hat{\theta}} = \frac{\left(\sum_{j \in S} w_j\right)^2}{\sum_{j \in S} w_j^2}.$$

The observed number of indicator cases is then $y^{\hat{\theta}} = m^{\hat{\theta}} \cdot \hat{\theta}$. To make these computations we use the `survey` R package (Lumley 2004).

S1.2 Process specification

	Model component	Latent field	Hyperparameter
S1.2.1	HIV prevalence	$22 + 5n$	9
S1.2.2	ART coverage	$25 + 5n$	9
S1.2.3	HIV incidence rate	$2 + n$	3
S1.2.4	ANC testing	$2 + 2n$	2
S1.2.5	ART attendance	n	1
	Total	$51 + 14n$	24

Table S1: The numer of latent field parameters and hyperparameters in each model section.

We now describe the hyperparameter and latent field process specification for the model. Whereas in the main text process and likelihood specifications are written together, here we consider the likelihood equations separately in Section S1.3 to follow. Table S1.2 gives the number of latent field parameters and hyperparameters in each component of the model. In the case of Malawi then $n = 32$ such that the total number of latent field parameters is $51 + 14 \cdot 32 = 499$ and the total number of hyperparameters is 24.

S1.2.1 HIV prevalence

We model HIV prevalence $\rho_{x,s,a} \in [0, 1]$ on the logit scale using the linear predictor

$$\text{logit}(\rho_{x,s,a}) = \beta_0^\rho + \beta_S^{\rho,s=\text{M}} + u_a^\rho + u_a^{\rho,s=\text{M}} + u_x^\rho + u_x^{\rho,s=\text{M}} + u_x^{\rho,a<15} + \eta_{R_x,s,a}^\rho.$$

The term $\beta_0^\rho \sim \mathcal{N}(0, 5)$ is an intercept, and $\beta_s^{\rho,s=\text{M}} \sim \mathcal{N}(0, 5)$ is the difference in logit prevalence for men compared to women. The terms $u_a^\rho \sim \text{AR1}(\sigma_A^\rho, \phi_A^\rho)$ are age random effects for women, and $u_a^{\rho,s=\text{M}} \sim \text{AR1}(\sigma_{AS}^\rho, \phi_{AS}^\rho)$ are age random effects for the difference in logit prevalence for men compared to women age a. The terms $u_x^\rho \sim \text{BYM2}(\sigma_X^\rho, \phi_X^\rho)$ are spatial random effects for women, and $u_x^{\rho,s=\text{M}} \sim \text{BYM2}(\sigma_{XS}^\rho, \phi_{XS}^\rho)$ are spatial random effects for the difference in logit prevalence for men compared to women in district x. The terms $u_x^{\rho,a<15} \sim \text{ICAR}(\sigma_{XA}^\rho)$ are spatial random effects for the ratio of paediatric prevalence to adult women prevalence. Finally, $\eta_{R_x,s,a}^\rho$ are fixed offsets specifying assumed odds ratios for prevalence outside the age ranges for which data are available. The two BYM2 random effects are comprised of the following unit scaled spatially structured $\{v_x^\rho, v_x^{\rho,s=\text{M}}\}$ and spatially unstructured $\{w_x^\rho, w_x^{\rho,s=\text{M}}\}$ components

$$u_x^\rho = \sigma_X^\rho \left(\sqrt{\phi}_X^\rho \cdot v_x^\rho + \sqrt{1 - \phi_X^\rho} \cdot w_x^\rho\right),$$

$$u_x^{\rho,s=\text{M}} = \sigma_{XS}^\rho \left(\sqrt{\phi}_{XS}^\rho \cdot v_x^{\rho,s=\text{M}} + \sqrt{1 - \phi_{XS}^\rho} \cdot w_x^{\rho,s=\text{M}}\right).$$

We use half-normal priors for the standard deviation terms

$$\{\sigma_A^\rho, \sigma_{AS}^\rho, \sigma_X^\rho, \sigma_{XS}^\rho, \sigma_{XA}^\rho\} \sim \mathcal{N}^+(0, 2.5),$$

uniform priors for the AR1 correlation parameters

$$\{\phi_A^\rho, \phi_{AS}^\rho\} \sim \mathcal{U}(-1, 1),$$

and beta priors for the BYM2 proportion parameters

$$\{\phi_X^\rho, \phi_{XS}^\rho\} \sim \text{Beta}(0.5, 0.5).$$

S1.2.2 ART coverage

We model ART coverage $\alpha_{x,s,a} \in [0,1]$ on the logit scale using the linear predictor

$$\text{logit}(\alpha_{x,s,a}) = \beta_0^\alpha + \beta_S^{\alpha,s=\text{M}} + u_a^\alpha + u_a^{\alpha,s=\text{M}} + u_x^\alpha + u_x^{\alpha,s=\text{M}} + u_x^{\alpha,a<15} + \eta_{R_x,s,a}^\alpha$$

with terms and priors analogous to the HIV prevalence process model in Section S1.2.1 above.

S1.2.3 HIV incidence rate

We model HIV incidence rate $\lambda_{x,s,a} > 0$ on the log scale using the linear predictor

$$\log(\lambda_{x,s,a}) = \beta_0^\lambda + \beta_S^{\lambda,s=\text{M}} + \log(\rho_x^{\text{15-49}}) + \log(1 - \omega \cdot \alpha_x^{\text{15-49}}) + u_x^\lambda + \eta_{R_x,s,a}^\lambda,$$

where $\beta_0^\lambda \sim \mathcal{N}(0,5)$ is an intercept term proportional to the average HIV transmission rate for untreated HIV positive adults and $\beta_S^{\lambda,s=\text{M}} \sim \mathcal{N}(0,5)$ is the log incidence rate ratio for men compared to women. The term

$$\rho_x^{\text{15-49}} = \frac{\sum_{s\in\{\text{F,M}\}} \sum_{a=15}^{45} N_{x,s,a} \cdot \rho_{x,s,a}}{\sum_{s\in\{\text{F,M}\}} \sum_{a=15}^{45} N_{x,s,a}}$$

is the HIV prevalence among adults 15-49 calculated by aggregating age-specific HIV prevalences, and

$$\alpha_x^{\text{15-49}} = \frac{\sum_{s\in\{\text{F,M}\}} \sum_{a=15}^{45} N_{x,s,a} \cdot \rho_{x,s,a} \cdot \alpha_{x,s,a}}{\sum_{s\in\{\text{F,M}\}} \sum_{a=15}^{45} N_{x,s,a} \cdot \rho_{x,s,a}}$$

is the ART coverage among adults 15-49 calculated by aggregating age-specific ART coverages. The term ω is the average reduction in HIV transmission rate per 1% increase in population ART coverage and is fixed at $\omega = 0.7$ based on inputs to the Estimation and Projection Package (EPP) model (Jeffrey W. Eaton et al. 2019). The terms $u_x^\lambda \sim \mathcal{N}(0, \sigma^\lambda)$ with $\sigma^\lambda \sim \mathcal{N}^+(0,1)$ are IID spatial random effects. Finally, $\eta_{R_x,s,a}^\lambda$ specify fixed log incidence rate ratios by sex and age group calculated from Spectrum model output[1]

We model the proportion recently infected among HIV positive persons $\kappa_{x,s,a} \in [0,1]$ as

$$\kappa_{x,s,a} = 1 - \exp\left(-\lambda_{x,s,a} \cdot \frac{1-\rho_{x,s,a}}{\rho_{x,s,a}} \cdot (\Omega_T - \beta_T) - \beta_T\right),$$

where $\Omega_T \sim \mathcal{N}(\Omega_{T_0}, \sigma^{\Omega_T})$ is the mean duration of recent infection, and $\beta_T \sim \mathcal{N}^+(\beta_{T_0}, \sigma^{\beta_T})$ is the false recent ratio. We use an informative prior for Ω_T based on the characteristics of the recent infection testing algorithm. For PHIA surveys this is $\Omega_{T_0} = 130$ days and $\sigma^{\Omega_T} = 6.12$ days, and further we assume there is no false recency, such that $\beta_{T_0} = 0.0$ and $\sigma^{\beta_T} = 0.0$.

S1.2.4 ANC testing

HIV prevalence $\rho_{x,a}^{\text{ANC}}$ and ART coverage $\alpha_{x,a}^{\text{ANC}}$ among pregnant women are modelled as being offset on the logit scale from the corresponding district-age indicators $\rho_{x,F,a}$ and $\alpha_{x,F,a}$ according to

$$\text{logit}(\rho_{x,a}^{\text{ANC}}) = \text{logit}(\rho_{x,F,a}) + \beta^{\rho^{\text{ANC}}} + u_x^{\rho^{\text{ANC}}} + \eta_{R_x,a}^{\rho^{\text{ANC}}},$$
$$\text{logit}(\alpha_{x,a}^{\text{ANC}}) = \text{logit}(\alpha_{x,F,a}) + \beta^{\alpha^{\text{ANC}}} + u_x^{\alpha^{\text{ANC}}} + \eta_{R_x,a}^{\alpha^{\text{ANC}}},$$

where, for $\theta \in \{\rho, \alpha\}$, $\beta^{\theta^{\text{ANC}}} \sim \mathcal{N}(0,5)$ are the average differences between population and ANC outcomes, $u_x^{\theta^{\text{ANC}}} \sim \mathcal{N}(0, \sigma_X^{\theta^{\text{ANC}}})$ are IID district random effects with $\sigma_X^{\theta^{\text{ANC}}} \sim \mathcal{N}^+(0,1)$, and $\eta_{R_x,a}^{\theta^{\text{ANC}}}$ for are offsets for the

[1]Note that these outputs are the only source of age structure for this part of the model. Furthermore, Spectrum assumes that there are no new infections in children aged 5-9 or 10-14, or adults aged over 80. A consequence is that the posterior over new infections in these age groups is exactly zero, by definition. We remove these identically zero posteriors from any inference comparisons.

log fertility rate ratios for HIV positive women compared to HIV negative women and for women on ART to HIV positive women not on ART, calculated from Spectrum model outputs for region R_x.

In the full Naomi model, for adult women 15-49 the number of ANC clients $\Psi_{x,a} > 0$ are modelled as

$$\log(\Psi_{x,a}) = \log(N_{x,\mathrm{F},a}) + \psi_{R_x,a} + \beta^{\psi} + u_x^{\psi}$$

where $N_{x,\mathrm{F},a}$ are the female population sizes, $\psi_{R_x,a}$ are fixed age-sex fertility ratios in Spectrum region R_x, β^{ψ} are log rate ratios for the number of ANC clients relative to the predicted fertility, and $u_x^{\psi} \sim \mathcal{N}(0, \sigma^{\psi})$ are district random effects. Here we fix $\beta^{\psi} = u_x^{\psi} = 0$ such that $\Psi_{x,a}$ are simply constants.

S1.2.5 ART attendance

Let $\gamma_{x,x'} \in [0,1]$ be the probability that a person on ART residing in district x receives ART in district x'. We assume that $\gamma_{x,x'} = 0$ for $x \notin \{x, \mathrm{ne}(x)\}$ such that individuals seek treatment only in their residing district or its neighbours $\mathrm{ne}(x) = \{x' : x' \sim x\}$, where $\sim$ is an adjacency relation, and $\sum_{x' \in \{x, \mathrm{ne}(x)\}} \gamma_{x,x'} = 1$. To model $\gamma_{x,x'}$ for $x \sim x'$ we use a multinomial logistic regression model, based on the log-odds ratios

$$\tilde{\gamma}_{x,x'} = \log\left(\frac{\gamma_{x,x'}}{1 - \gamma_{x,x'}}\right) = \tilde{\gamma}_0 + u_x^{\tilde{\gamma}}, \tag{1}$$

where $\tilde{\gamma}_0 = -4$ is a fixed intercept, and $u_x^{\tilde{\gamma}} \sim \mathcal{N}(0, \sigma_X^{\tilde{\gamma}})$ are district random effects with $\sigma_X^{\tilde{\gamma}} \sim \mathcal{N}^+(0, 2.5)$. Note that Equation 1 does not depend on x', such that $\gamma_{x,x'}$ is only a function of x. Choice of $\tilde{\gamma}_0 = -4$ implies a prior mean on $\gamma_{x,x'}$ of 1.8%, such that $(100 - 1.8 \times \mathrm{ne}(x))\%$ of ART clients in district x obtain treatment in their home district, a-priori. We fix $\tilde{\gamma}_{x,x} = 0$ and recover the multinomial probabilities using the softmax

$$\gamma_{x,x'} = \frac{\exp(\tilde{\gamma}_{x,x'})}{\sum_{x^\star \in \{x, \mathrm{ne}(x)\}} \exp(\tilde{\gamma}_{x,x^\star})}.$$

Given the total number of PLHIV on ART $A_{x,s,a} = N_{x,s,a} \cdot \rho_{x,s,a} \cdot \alpha_{x,s,a}$, the number of ART clients who reside in district x and obtain ART in district x' are $A_{x,x',s,a} = A_{x,s,a} \cdot \gamma_{x,x'}$, and the total attending ART facilities in district x' are

$$\tilde{A}_{x',s,a} = \sum_{x \in \{x', \mathrm{ne}(x')\}} A_{x,x',s,a}.$$

S1.3 Likelihood specification

S1.3.1 Household survey data

For HIV prevalence, ART coverage and recent HIV infections, denoted by $\theta \in \{\rho, \alpha, \kappa\}$, the most recent household survey furnishes weighted observations $\hat{\theta}_{\{x\},\{s\},\{a\}}$ with respective Kish effective sample sizes $m^{\hat{\theta}}_{\{x\},\{s\},\{a\}} \in \mathbb{R}$, and observed number of cases

$$y^{\hat{\theta}}_{\{x\},\{s\},\{a\}} = m^{\hat{\theta}}_{\{x\},\{s\},\{a\}} \cdot \hat{\theta}_{\{x\},\{s\},\{a\}} \in \mathbb{R}.$$

To model these observations, we use following three binomial working likelihoods

$$y^{\hat{\rho}}_{\{x\},\{s\},\{a\}} \sim \mathrm{xBin}(m^{\hat{\rho}}_{\{x\},\{s\},\{a\}}, \rho_{\{x\},\{s\},\{a\}}), \quad \rho_{\{x\},\{s\},\{a\}} = \frac{\sum_{\{x\}}\sum_{\{s\}}\sum_{\{a\}} N_{x,s,a} \cdot \rho_{x,s,a}}{\sum_{\{x\}}\sum_{\{s\}}\sum_{\{a\}} N_{x,s,a}},$$

$$y^{\hat{\alpha}}_{\{x\},\{s\},\{a\}} \sim \mathrm{xBin}(m^{\hat{\alpha}}_{\{x\},\{s\},\{a\}}, \alpha_{\{x\},\{s\},\{a\}}), \quad \alpha_{\{x\},\{s\},\{a\}} = \frac{\sum_{\{x\}}\sum_{\{s\}}\sum_{\{a\}} N_{x,s,a} \cdot \rho_{x,s,a} \cdot \alpha_{x,s,a}}{\sum_{\{x\}}\sum_{\{s\}}\sum_{\{a\}} N_{x,s,a} \cdot \rho_{x,s,a}},$$

$$y^{\hat{\kappa}}_{\{x\},\{s\},\{a\}} \sim \mathrm{xBin}(m^{\hat{\kappa}}_{\{x\},\{s\},\{a\}}, \kappa_{\{x\},\{s\},\{a\}}), \quad \kappa_{\{x\},\{s\},\{a\}} = \frac{\sum_{\{x\}}\sum_{\{s\}}\sum_{\{a\}} N_{x,s,a} \cdot \rho_{x,s,a} \cdot \kappa_{x,s,a}}{\sum_{\{x\}}\sum_{\{s\}}\sum_{\{a\}} N_{x,s,a} \cdot \rho_{x,s,a}}.$$

The generalised binomial $y \sim \mathrm{xBin}(m, p)$ is defined for $y, m \in \mathbb{R}^+$ with $y \leq m$ such that

$$\log p(y) = \log\Gamma(m+1) - \log\Gamma(y+1) - \log\Gamma(m-y+1) + y\log p + (m-y)\log(1-p),$$

where the gamma function Γ is such that $\forall n \in \mathbb{N}$, $\Gamma(n) = (n-1)!$.

S1.3.2 ANC testing data

We include ANC testing data for the year of the most recent survey. Let $W^{\text{ANC}}_{\{x\}}$ be the number of ANC clients, $X^{\text{ANC}}_{\{x\}}$ the number of those with ascertained status, $Y^{\text{ANC}}_{\{x\}}$ the number of those with positive status (either known or tested) and $Z^{\text{ANC}}_{\{x\}}$ the number of ANC clients already on ART prior to first ANC, such that

$$W^{\text{ANC}}_{x} \geq X^{\text{ANC}}_{x} \geq Y^{\text{ANC}}_{x} \geq Z^{\text{ANC}}_{x},$$

for all $x \in \{x\}$. When ANC testing data are only available for part of a given year, we denote $m^{\text{ANC}} \in \{1, \ldots, 12\}$ the number of months of reported data reflected in counts for that year. The observed number of HIV positive and already on ART among ANC clients is modelled by

$$\begin{aligned} Y^{\text{ANC}}_{\{x\}} &\sim \text{Bin}\left(X^{\text{ANC}}_{\{x\}}, \rho^{\text{ANC}}_{\{x\},\{15,\ldots 45\}}\right), \\ Z^{\text{ANC}}_{\{x\}} &\sim \text{Bin}\left(Y^{\text{ANC}}_{\{x\}}, \alpha^{\text{ANC}}_{\{x\},\{15,\ldots 45\}}\right), \end{aligned}$$

where prevalence and ART coverage are aggregated by the number of pregnant women $\Psi_{x,a}$

$$\begin{aligned} \rho^{\text{ANC}}_{\{x\}\{a\}} &= \frac{\sum_{\{x\}}\sum_{\{a\}} \Psi_{x,a} \cdot \rho^{\text{ANC}}_{x,a}}{\sum_{\{x\}}\sum_{\{a\}} \Psi_{x,a}}, \\ \alpha^{\text{ANC}}_{\{x\}\{a\}} &= \frac{\sum_{\{x\}}\sum_{\{a\}} \Psi_{x,a} \cdot \rho^{\text{ANC}}_{x,a} \cdot \alpha^{\text{ANC}}_{x,a}}{\sum_{\{x\}}\sum_{\{a\}} \Psi_{x,a} \cdot \rho^{\text{ANC}}_{x,a}}. \end{aligned}$$

S1.3.3 Number receiving ART

Let $\dot{A}_{\{x\},\{s\},\{a\}}$ be data for the number receiving ART

$$\dot{A}_{\{x\},\{s\},\{a\}} = \sum_{\{x\}}\sum_{\{s\}}\sum_{\{a\}}\sum_{x\sim x', x=x'} \dot{A}_{x',x,s,a}.$$

We model the unobserved numbers of ART clients travelling from x' to x as

$$\dot{A}_{x',x,s,a} \sim \text{Bin}(N_{x',s,a}, \pi_{x',x,s,a})$$

where $\pi_{x',x,s,a} = \rho_{x',s,a} \cdot \alpha_{x',s,a} \cdot \gamma_{x',x,s,a}$. This likelihood is approximated using a normal for the sum of binomials by

$$\dot{A}_{\{x\},\{s\},\{a\}} \sim \mathcal{N}(\tilde{A}_{\{x\},\{s\},\{a\}}, \sigma^{\tilde{A}}_{\{x\},\{s\},\{a\}})$$

where the mean is

$$\tilde{A}_{\{x\},\{s\}\{a\}} = \sum_{\{x\}}\sum_{\{s\}}\sum_{\{a\}}\sum_{x\sim x', x=x'} N_{x',s,a} \cdot \pi_{x',x,s,a},$$

and the variance is

$$\left(\sigma^{\tilde{A}}_{\{x\},\{s\}\{a\}}\right)^2 = \sum_{\{x\}}\sum_{\{s\}}\sum_{\{a\}}\sum_{x\sim x', x=x'} N_{x',s,a} \cdot \pi_{x',x,s,a} \cdot (1 - \pi_{x',x,s,a}).$$

S1.4 Identifiability constaints

If data are missing, some parameters are fixed to default values to help with identifiability. In particular:

1. If survey data on HIV prevalence or ART coverage by age and sex are not available then we set $u^{\theta}_{a} = 0$ and $u^{\theta}_{a,s=\text{M}} = 0$ and use the average age-sex pattern of from the Spectrum offset $\eta^{\theta}_{R_x,s,a}$. For the Malawi example considered in the main text HIV prevalence and ART coverage data are not available for those aged 65+. As a result, there are $|\{\text{0-4}, \ldots, \text{50-54}\}| = 13$ age groups included for the age random effects.

2. If no ART data, either survey or ART programme, are available but data on ART coverage among ANC clients are available, the level of ART coverage is not identifiable, but spatial variation is identifiable. In this instance, overall ART coverage is determined by the Spectrum offset, and only area random effects are estimated such that $\text{logit}\,(\alpha_{x,s,a}) = u^{\alpha}_{x} + \eta^{\alpha}_{R_x,s,a}$.
3. If survey data on recent HIV infection are not included in the model, then $\beta^{\lambda}_{0} = \beta^{\lambda,s=\text{M}}_{S} = u^{\lambda}_{x} = 0$. The sex ratio for HIV incidence is determined by the sex incidence rate ratio from Spectrum in the same years and the incidence rate in all districts is modelled assuming the same average HIV transmission rate for untreated adults, but varies according to district estimates of HIV prevalence and ART coverage.

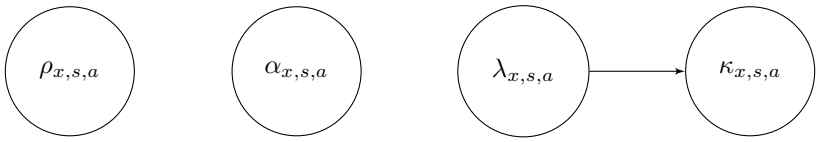

Figure S1: Directed acyclic graph describing the simplified Naomi model (work in progress).

S2 C++ `TMB` user template

This section contains C++ `TMB` code for the negative log-posterior of the simplified Naomi model For ease of understanding, Table S2 provides correspondence between the mathematical notation used in Section S1 and the variable names used in the `TMB` code, for all hyperparameters and latent field parameters. For further reference on the `TMB` software see `https://kaskr.github.io/adcomp/_book`, or on the method see Kristensen et al. (2016).

Variable name	Notation	Type	Size	Domain	ρ input?	α input?	λ input?
`logit_phi_rho_x`	$\text{logit}(\phi_X^\rho)$	Hyper	1	$\mathbb{R}$	✓		
`log_sigma_rho_x`	$\log(\sigma_X^\rho)$	Hyper	1	$\mathbb{R}$	✓		
`logit_phi_rho_xs`	$\text{logit}(\phi_{XS}^\rho)$	Hyper	1	$\mathbb{R}$	✓		
`log_sigma_rho_xs`	$\log(\sigma_{XS}^\rho)$	Hyper	1	$\mathbb{R}$	✓		
`logit_phi_rho_a`	$\text{logit}(\phi_A^\rho)$	Hyper	1	$\mathbb{R}$	✓		
`log_sigma_rho_a`	$\log(\sigma_A^\rho)$	Hyper	1	$\mathbb{R}$	✓		
`logit_phi_rho_as`	$\text{logit}(\phi_{AS}^\rho)$	Hyper	1	$\mathbb{R}$	✓		
`log_sigma_rho_as`	$\log(\sigma_{AS}^\rho)$	Hyper	1	$\mathbb{R}$	✓		
`log_sigma_rho_xa`	$\log(\sigma_{XA}^\rho)$	Hyper	1	$\mathbb{R}$	✓		
`logit_phi_alpha_x`	$\text{logit}(\phi_X^\alpha)$	Hyper	1	$\mathbb{R}$		✓	
`log_sigma_alpha_x`	$\log(\sigma_X^\alpha)$	Hyper	1	$\mathbb{R}$		✓	
`logit_phi_alpha_xs`	$\text{logit}(\phi_{XS}^\alpha)$	Hyper	1	$\mathbb{R}$		✓	
`log_sigma_alpha_xs`	$\log(\sigma_{XS}^\alpha)$	Hyper	1	$\mathbb{R}$		✓	
`logit_phi_alpha_a`	$\text{logit}(\phi_A^\alpha)$	Hyper	1	$\mathbb{R}$		✓	
`log_sigma_alpha_a`	$\log(\sigma_A^\alpha)$	Hyper	1	$\mathbb{R}$		✓	
`logit_phi_alpha_as`	$\text{logit}(\phi_{AS}^\alpha)$	Hyper	1	$\mathbb{R}$		✓	
`log_sigma_alpha_as`	$\log(\sigma_{AS}^\alpha)$	Hyper	1	$\mathbb{R}$		✓	
`log_sigma_alpha_xa`	$\log(\sigma_{XA}^\alpha)$	Hyper	1	$\mathbb{R}$		✓	
`OmegaT_raw`	Ω_T	Hyper	1	$\mathbb{R}$			✓
`log_betaT`	$\log(\beta_T)$	Hyper	1	$\mathbb{R}$			✓
`log_sigma_lambda_x`	$\log(\sigma^\lambda)$	Hyper	1	$\mathbb{R}$			✓
`log_sigma_ancrho_x`	$\log(\sigma_X^{\rho^{\text{ANC}}})$	Hyper	1	$\mathbb{R}$			
`log_sigma_ancalpha_x`	$\log(\sigma_X^{\alpha^{\text{ANC}}})$	Hyper	1	$\mathbb{R}$			
`log_sigma_or_gamma`	$\log(\sigma_X^{\tilde{\gamma}})$	Hyper	1	$\mathbb{R}$			
`beta_rho`	$(\beta_0^\rho, \beta_s^{\rho,s=\text{M}})$	Latent	2	$\mathbb{R}^2$	✓		
`beta_alpha`	$(\beta_0^\alpha, \beta_S^{\alpha,s=\text{M}})$	Latent	2	$\mathbb{R}^2$		✓	
`beta_lambda`	$(\beta_0^\lambda, \beta_S^{\lambda,s=\text{M}})$	Latent	2	$\mathbb{R}^2$			✓
`beta_anc_rho`	$\beta^{\rho^{\text{ANC}}}$	Latent	1	$\mathbb{R}$			
`beta_anc_alpha`	$\beta^{\alpha^{\text{ANC}}}$	Latent	1	$\mathbb{R}$			
`u_rho_x`	w_x^ρ	Latent	n	$\mathbb{R}^n$	✓		
`us_rho_x`	v_x^ρ	Latent	n	$\mathbb{R}^n$	✓		
`u_rho_xs`	$w_x^{\rho,s=\text{M}}$	Latent	n	$\mathbb{R}^n$	✓		
`us_rho_xs`	$v_x^{\rho,s=\text{M}}$	Latent	n	$\mathbb{R}^n$	✓		
`u_rho_a`	u_a^ρ	Latent	10	$\mathbb{R}^{10}$	✓		
`u_rho_as`	$u_a^{\rho,s=\text{M}}$	Latent	10	$\mathbb{R}^{10}$	✓		
`u_rho_xa`	$u_x^{\rho,a<15}$	Latent	n	$\mathbb{R}^n$	✓		
`u_alpha_x`	w_x^α	Latent	n	$\mathbb{R}^n$		✓	
`us_alpha_x`	v_x^α	Latent	n	$\mathbb{R}^n$		✓	
`u_alpha_xs`	$w_x^{\alpha,s=\text{M}}$	Latent	n	$\mathbb{R}^n$		✓	
`us_alpha_xs`	$v_x^{\alpha,s=\text{M}}$	Latent	n	$\mathbb{R}^n$		✓	
`u_alpha_a`	u_a^α	Latent	13	$\mathbb{R}^{13}$		✓	
`u_alpha_as`	$u_a^{\alpha,s=\text{M}}$	Latent	10	$\mathbb{R}^{10}$		✓	
`u_alpha_xa`	$u_x^{\alpha,a<15}$	Latent	n	$\mathbb{R}^n$		✓	
`ui_lambda_x`	u_x^λ	Latent	n	$\mathbb{R}^n$			✓
`ui_anc_rho_x`	$u_x^{\rho^{\text{ANC}}}$	Latent	n	$\mathbb{R}^n$			
`ui_anc_alpha_x`	$u_x^{\alpha^{\text{ANC}}}$	Latent	n	$\mathbb{R}^n$			
`log_or_gamma`	$u_x^{\tilde{\gamma}}$	Latent	n	$\mathbb{R}^n$			

Table S2: Correspondence between mathematical notation and variable names used in our `TMB` code. The total number of hyperparameters is 24, and the total number of latent field parameters is $51 + 14n$, where n is the number of districts. We use the notation ✓ to refer to direct dependence of the parameter on the variable, ✗ to refer to no dependence, and a blank entry to refer to dependence conditional on the data.

```
// #define TMB_LIB_INIT R_init_naomi_simple
#include <TMB.hpp>


/** Log posterior density of BYM2 with INLA conditional parameterisation
 *
 * Calculate the joint LPDF of parameter vector (x, u) where
 * x = sigma * (sqrt(phi) * u + sqrt(1-phi) * v) with u a ICAR structured
 * component u ~ N(0, Q^-1) and v is an IID effect v ~ N(0, 1). Calculation
 * proceeds by conditioning P(<x, u>) = P(x | u) * P(u). See Reibler et al.
 * Section 3.4.
 *
 * @param x vector of random effects.
 * @param u vector of spatial component of random effect.
 * @param sigma marginal standard deviation (>0).
 * @param phi proportion of marginal variance explained by spatial structured
 *            component u (phi \in [0, 1]).
 * @param Q scaled structure matrix for spatial component.
 *
 * @return Log probability density of x and u.
 *
 * @note
 * The $\sqrt(2\pi)^{-2*n}$ and $|Q|^{1/2}$ terms are dropped.
 * Returns the _positive_ log PDF (different from builtin TMB
 * functions. Thus should typically be implemented as `nll -= bym2_conditional_lpdf(...)`.
 */
template<class Type>
Type bym2_conditional_lpdf(const vector<Type> x,
                           const vector<Type> u,
                           const Type sigma,
                           const Type phi,
                           const Eigen::SparseMatrix<Type> Q) {

  Type val(0.0);

  // constant terms omitted: -0.5 * (n + rank(Q)) * log(2*pi) + 0.5 * log|Q|
  val += -0.5 * x.size() * (2 * log(sigma) + log(1 - phi));  // normalising constant
  val += -0.5 / (sigma * sigma * (1 - phi)) * (x * x).sum();
  val += sqrt(phi) / (sigma * (1 - phi)) * (x * u).sum();
  val += -0.5 * (u * (Q * u)).sum();
  val += -0.5 * phi / (1 - phi) * (u * u).sum();

  return(val);
}

template<class Type>
Type objective_function<Type>::operator() ()
{

  // indexing:
  //
  // * rho: HIV prevalence model
  // * alpha: ART coverage model
```

```
// * lambda: HIV incidence model
//
// * _x: area
// * _a: age
// * _s: sex
// * _t: time

using namespace density;

// ** Data **

// Population
DATA_VECTOR(population_t1);

// Design matrices
DATA_MATRIX(X_rho);
DATA_MATRIX(X_alpha);
DATA_MATRIX(X_lambda);

DATA_MATRIX(X_ancrho);
DATA_MATRIX(X_ancalpha);

DATA_SPARSE_MATRIX(Z_rho_x);
DATA_SPARSE_MATRIX(Z_rho_xs);
DATA_SPARSE_MATRIX(Z_rho_a);
DATA_SPARSE_MATRIX(Z_rho_as);
DATA_SPARSE_MATRIX(Z_rho_xa);

DATA_SPARSE_MATRIX(Z_alpha_x);
DATA_SPARSE_MATRIX(Z_alpha_xs);
DATA_SPARSE_MATRIX(Z_alpha_a);
DATA_SPARSE_MATRIX(Z_alpha_as);
DATA_SPARSE_MATRIX(Z_alpha_xa);

DATA_SPARSE_MATRIX(Z_x);
DATA_SPARSE_MATRIX(Z_lambda_x);

DATA_VECTOR(logit_rho_offset);
DATA_VECTOR(logit_alpha_offset);

DATA_VECTOR(log_asfr_t1_offset);

DATA_VECTOR(logit_anc_rho_t1_offset);

DATA_VECTOR(logit_anc_alpha_t1_offset);

DATA_SPARSE_MATRIX(Z_ancrho_x);
DATA_SPARSE_MATRIX(Z_ancalpha_x);


// Precision matrix for ICAR area model
DATA_SPARSE_MATRIX(Q_x);
DATA_SCALAR(Q_x_rankdef);
```

```
DATA_VECTOR(n_prev);
DATA_VECTOR(x_prev);
DATA_SPARSE_MATRIX(A_prev);

DATA_VECTOR(n_artcov);
DATA_VECTOR(x_artcov);
DATA_SPARSE_MATRIX(A_artcov);

DATA_VECTOR(n_recent);
DATA_VECTOR(x_recent);
DATA_SPARSE_MATRIX(A_recent);

DATA_VECTOR(n_anc_prev_t1);
DATA_VECTOR(x_anc_prev_t1);
DATA_SPARSE_MATRIX(A_anc_prev_t1);

DATA_VECTOR(n_anc_artcov_t1);
DATA_VECTOR(x_anc_artcov_t1);
DATA_SPARSE_MATRIX(A_anc_artcov_t1);

DATA_SPARSE_MATRIX(A_artattend_t1);
DATA_VECTOR(x_artnum_t1);

DATA_SPARSE_MATRIX(A_artattend_mf);
DATA_SPARSE_MATRIX(A_art_reside_attend);

DATA_IVECTOR(n_nb);
DATA_IVECTOR(adj_i);
DATA_IVECTOR(adj_j);

DATA_SPARSE_MATRIX(Xgamma);
DATA_VECTOR(log_gamma_offset);

DATA_SPARSE_MATRIX(Xart_idx);
DATA_SPARSE_MATRIX(Xart_gamma);


// Incidence model
DATA_SCALAR(omega);
DATA_SCALAR(OmegaT0);
DATA_SCALAR(sigma_OmegaT);
DATA_SCALAR(betaT0);
DATA_SCALAR(sigma_betaT);
DATA_SCALAR(ritaT);

DATA_SPARSE_MATRIX(X_15to49);
DATA_VECTOR(log_lambda_t1_offset);

// Paediatric prevalence and incidence ratio model

DATA_SPARSE_MATRIX(X_15to49f);
DATA_SPARSE_MATRIX(X_paed_rho_ratio);
DATA_VECTOR(paed_rho_ratio_offset);
```

```
DATA_SPARSE_MATRIX(X_paed_lambda_ratio_t1);

// ** Initialize nll **
Type val(0);

// ** Parameters **

// fixed effects
// diffuse N(0.0, 5.0) prior distribution

PARAMETER_VECTOR(beta_rho);
val -= dnorm(beta_rho, 0.0, 5.0, true).sum();

PARAMETER_VECTOR(beta_alpha);
val -= dnorm(beta_alpha, 0.0, 5.0, true).sum();

PARAMETER_VECTOR(beta_lambda);
val -= dnorm(beta_lambda, 0.0, 5.0, true).sum();

PARAMETER_VECTOR(beta_anc_rho);
val -= dnorm(beta_anc_rho, 0.0, 5.0, true).sum();

PARAMETER_VECTOR(beta_anc_alpha);
val -= dnorm(beta_anc_alpha, 0.0, 5.0, true).sum();

// * HIV prevalence model *

// hyper parameters

PARAMETER(logit_phi_rho_x);
Type phi_rho_x(invlogit(logit_phi_rho_x));
val -= log(phi_rho_x) +  log(1 - phi_rho_x);  // change of variables: logit_phi_x ->v phi_x
val -= dbeta(phi_rho_x, Type(0.5), Type(0.5), true);

PARAMETER(log_sigma_rho_x);
Type sigma_rho_x(exp(log_sigma_rho_x));
val -= dnorm(sigma_rho_x, Type(0.0), Type(2.5), true) + log_sigma_rho_x;

PARAMETER(logit_phi_rho_xs);
Type phi_rho_xs(invlogit(logit_phi_rho_xs));
val -= log(phi_rho_xs) +  log(1 - phi_rho_xs);  // change of variables: logit_phi_xs -> phi_xs
val -= dbeta(phi_rho_xs, Type(0.5), Type(0.5), true);

PARAMETER(log_sigma_rho_xs);
Type sigma_rho_xs(exp(log_sigma_rho_xs));
val -= dnorm(sigma_rho_xs, Type(0.0), Type(2.5), true) + log_sigma_rho_xs;

PARAMETER(logit_phi_rho_a);
val -= dnorm(logit_phi_rho_a, Type(0.0), Type(2.582), true);  // INLA default
Type phi_rho_a(2.0 * invlogit(logit_phi_rho_a) - 1.0);

PARAMETER(log_sigma_rho_a);
Type sigma_rho_a(exp(log_sigma_rho_a));
```

```
val -= dnorm(sigma_rho_a, Type(0.0), Type(2.5), true) + log_sigma_rho_a;

PARAMETER(logit_phi_rho_as);
val -= dnorm(logit_phi_rho_as, Type(0.0), Type(2.582), true);  // INLA default
Type phi_rho_as(2.0 * invlogit(logit_phi_rho_as) - 1.0);

PARAMETER(log_sigma_rho_as);
Type sigma_rho_as(exp(log_sigma_rho_as));
val -= dnorm(sigma_rho_as, Type(0.0), Type(2.5), true) + log_sigma_rho_as;

PARAMETER(log_sigma_rho_xa);
Type sigma_rho_xa(exp(log_sigma_rho_xa));
val -= dnorm(sigma_rho_xa, Type(0.0), Type(0.5), true) + log_sigma_rho_xa;

// latent effects

PARAMETER_VECTOR(u_rho_x);
PARAMETER_VECTOR(us_rho_x);
val -= dnorm(sum(us_rho_x), Type(0.0), Type(0.001) * us_rho_x.size(), true); // soft sum-to-zero cons
val -= bym2_conditional_lpdf(u_rho_x, us_rho_x, sigma_rho_x, phi_rho_x, Q_x);


PARAMETER_VECTOR(u_rho_xs);
PARAMETER_VECTOR(us_rho_xs);
if (u_rho_xs.size()) {
  val -= dnorm(sum(us_rho_xs), Type(0.0), Type(0.001) * us_rho_xs.size(), true); // soft sum-to-zero
  val -= bym2_conditional_lpdf(u_rho_xs, us_rho_xs, sigma_rho_xs, phi_rho_xs, Q_x);
}

PARAMETER_VECTOR(u_rho_a);
if(u_rho_a.size() > 0)
  val += SCALE(AR1(phi_rho_a), sigma_rho_a)(u_rho_a);

PARAMETER_VECTOR(u_rho_as);
if(u_rho_a.size() > 0)
  val += SCALE(AR1(phi_rho_as), sigma_rho_as)(u_rho_as);

PARAMETER_VECTOR(u_rho_xa);
if (u_rho_xa.size() > 0) {
  val -= dnorm(sum(u_rho_xa), Type(0.0), sigma_rho_xa * Type(0.001) * u_rho_xa.size(), true); // soft

  val -= -(Q_x.rows() - Q_x_rankdef) * log_sigma_rho_xa -
    0.5 / (sigma_rho_xa * sigma_rho_xa) * (u_rho_xa * (Q_x * u_rho_xa)).sum();
}


// * ART coverage model *

PARAMETER(logit_phi_alpha_x);
Type phi_alpha_x(invlogit(logit_phi_alpha_x));
val -= log(phi_alpha_x) +  log(1 - phi_alpha_x);  // change of variables: logit_phi_x -> phi_x
val -= dbeta(phi_alpha_x, Type(0.5), Type(0.5), true);
```

```
PARAMETER(log_sigma_alpha_x);
Type sigma_alpha_x(exp(log_sigma_alpha_x));
val -= dnorm(sigma_alpha_x, Type(0.0), Type(2.5), true) + log_sigma_alpha_x;

PARAMETER(logit_phi_alpha_xs);
Type phi_alpha_xs(invlogit(logit_phi_alpha_xs));
val -= log(phi_alpha_xs) +  log(1 - phi_alpha_xs);  // change of variables: logit_phi_xs -> phi_xs
val -= dbeta(phi_alpha_xs, Type(0.5), Type(0.5), true);

PARAMETER(log_sigma_alpha_xs);
Type sigma_alpha_xs(exp(log_sigma_alpha_xs));
val -= dnorm(sigma_alpha_xs, Type(0.0), Type(2.5), true) + log_sigma_alpha_xs;

PARAMETER(logit_phi_alpha_a);
val -= dnorm(logit_phi_alpha_a, Type(0.0), Type(2.582), true);  // INLA default
Type phi_alpha_a(2.0 * invlogit(logit_phi_alpha_a) - 1.0);

PARAMETER(log_sigma_alpha_a);
Type sigma_alpha_a(exp(log_sigma_alpha_a));
val -= dnorm(sigma_alpha_a, Type(0.0), Type(2.5), true) + log_sigma_alpha_a;

PARAMETER(logit_phi_alpha_as);
val -= dnorm(logit_phi_alpha_as, Type(0.0), Type(2.582), true);  // INLA default
Type phi_alpha_as(2.0 * invlogit(logit_phi_alpha_as) - 1.0);

PARAMETER(log_sigma_alpha_as);
Type sigma_alpha_as(exp(log_sigma_alpha_as));
val -= dnorm(sigma_alpha_as, Type(0.0), Type(2.5), true) + log_sigma_alpha_as;

PARAMETER(log_sigma_alpha_xa);
Type sigma_alpha_xa(exp(log_sigma_alpha_xa));
val -= dnorm(sigma_alpha_xa, Type(0.0), Type(2.5), true) + log_sigma_alpha_xa;

PARAMETER_VECTOR(u_alpha_x);
PARAMETER_VECTOR(us_alpha_x);
val -= dnorm(sum(us_alpha_x), Type(0.0), Type(0.001) * us_alpha_x.size(), true); // soft sum-to-zero
val -= bym2_conditional_lpdf(u_alpha_x, us_alpha_x, sigma_alpha_x, phi_alpha_x, Q_x);

PARAMETER_VECTOR(u_alpha_xs);
PARAMETER_VECTOR(us_alpha_xs);
if (u_alpha_xs.size()) {
  val -= dnorm(sum(us_alpha_xs), Type(0.0), Type(0.001) * us_alpha_xs.size(), true); // soft sum-to-z
  val -= bym2_conditional_lpdf(u_alpha_xs, us_alpha_xs, sigma_alpha_xs, phi_alpha_xs, Q_x);
}

PARAMETER_VECTOR(u_alpha_a);
if(u_alpha_a.size() > 0)
  val += SCALE(AR1(phi_alpha_a), sigma_alpha_a)(u_alpha_a);

PARAMETER_VECTOR(u_alpha_as);
if(u_alpha_as.size() > 0)
  val += SCALE(AR1(phi_alpha_as), sigma_alpha_as)(u_alpha_as);
```

```
PARAMETER_VECTOR(u_alpha_xa);
val -= dnorm(u_alpha_xa, 0.0, sigma_alpha_xa, true).sum();


// * HIV incidence model *

PARAMETER(OmegaT_raw);
val -= dnorm(OmegaT_raw, Type(0.0), Type(1.0), true);
Type OmegaT = OmegaT0 + OmegaT_raw * sigma_OmegaT;

PARAMETER(log_betaT);
val -= dnorm(exp(log_betaT), Type(0.0), Type(1.0), true) + log_betaT;
Type betaT = betaT0 + exp(log_betaT) * sigma_betaT;

PARAMETER(log_sigma_lambda_x);
Type sigma_lambda_x(exp(log_sigma_lambda_x));
val -= dnorm(sigma_lambda_x, Type(0.0), Type(1.0), true) + log_sigma_lambda_x;

PARAMETER_VECTOR(ui_lambda_x);
val -= sum(dnorm(ui_lambda_x, 0.0, sigma_lambda_x, true));


// * ANC testing model *

// ANC prevalence and ART coverage random effects
PARAMETER(log_sigma_ancrho_x);
Type sigma_ancrho_x(exp(log_sigma_ancrho_x));
val -= dnorm(sigma_ancrho_x, Type(0.0), Type(2.5), true) + log_sigma_ancrho_x;

PARAMETER(log_sigma_ancalpha_x);
Type sigma_ancalpha_x(exp(log_sigma_ancalpha_x));
val -= dnorm(sigma_ancalpha_x, Type(0.0), Type(2.5), true) + log_sigma_ancalpha_x;

PARAMETER_VECTOR(ui_anc_rho_x);
val -= sum(dnorm(ui_anc_rho_x, 0.0, sigma_ancrho_x, true));

PARAMETER_VECTOR(ui_anc_alpha_x);
val -= sum(dnorm(ui_anc_alpha_x, 0.0, sigma_ancalpha_x, true));


// * ART attendance model *

PARAMETER(log_sigma_or_gamma);
Type sigma_or_gamma(exp(log_sigma_or_gamma));
val -= dnorm(sigma_or_gamma, Type(0.0), Type(2.5), true) + log_sigma_or_gamma;

PARAMETER_VECTOR(log_or_gamma);
val -= dnorm(log_or_gamma, 0.0, sigma_or_gamma, true).sum();


// *** Process model ***

// HIV prevalence time 1
```

```
vector<Type> mu_rho(X_rho * beta_rho +
  logit_rho_offset +
  Z_rho_x * u_rho_x +
  Z_rho_xs * u_rho_xs +
  Z_rho_a * u_rho_a +
  Z_rho_as * u_rho_as +
  Z_rho_xa * u_rho_xa);

// paediatric prevalence

vector<Type> rho_15to49f_t1((X_15to49f * vector<Type>(invlogit(mu_rho) * population_t1)) / (X_15to49f
vector<Type> mu_rho_paed(X_paed_rho_ratio * rho_15to49f_t1 + paed_rho_ratio_offset);
mu_rho_paed = logit(mu_rho_paed);
mu_rho += mu_rho_paed;

// ART coverage time 1

vector<Type> mu_alpha(X_alpha * beta_alpha +
  logit_alpha_offset +
  Z_alpha_x * u_alpha_x +
  Z_alpha_xs * u_alpha_xs +
  Z_alpha_a * u_alpha_a +
  Z_alpha_as * u_alpha_as +
  Z_alpha_xa * u_alpha_xa);


vector<Type> rho_t1(invlogit(mu_rho));
vector<Type> alpha_t1(invlogit(mu_alpha));

vector<Type> plhiv_t1(population_t1 * rho_t1);
vector<Type> prop_art_t1(rho_t1 * alpha_t1);
vector<Type> artnum_t1(population_t1 * prop_art_t1);

vector<Type> plhiv_15to49_t1(X_15to49 * plhiv_t1);
vector<Type> rho_15to49_t1(plhiv_15to49_t1 / (X_15to49 * population_t1));
vector<Type> alpha_15to49_t1((X_15to49 * artnum_t1) / plhiv_15to49_t1);

vector<Type> mu_lambda_t1(X_lambda * beta_lambda + log_lambda_t1_offset +
  Z_x * vector<Type>(log(rho_15to49_t1) + log(1.0 - omega * alpha_15to49_t1)) +
  Z_lambda_x * ui_lambda_x);

vector<Type> lambda_adult_t1(exp(mu_lambda_t1));

// Add paediatric incidence
vector<Type> lambda_paed_t1(X_paed_lambda_ratio_t1 * rho_15to49f_t1);
vector<Type> lambda_t1(lambda_adult_t1 + lambda_paed_t1);

vector<Type> infections_t1(lambda_t1 * (population_t1 - plhiv_t1));

// likelihood for household survey data

vector<Type> rho_obs_t1((A_prev * plhiv_t1) / (A_prev * population_t1));
vector<Type> hhs_prev_ll = dbinom(x_prev, n_prev, rho_obs_t1, true);
```

```
val -= sum(hhs_prev_ll);

vector<Type> alpha_obs_t1((A_artcov * artnum_t1) / (A_artcov * plhiv_t1));
vector<Type> hhs_artcov_ll = dbinom(x_artcov, n_artcov, alpha_obs_t1, true);
val -= sum(hhs_artcov_ll);

vector<Type> pR_infections_obs_t1(A_recent * infections_t1);
vector<Type> pR_plhiv_obs_t1(A_recent * plhiv_t1);
vector<Type> pR_population_obs_t1(A_recent * population_t1);
vector<Type> pR_lambda_obs_t1(pR_infections_obs_t1 / (pR_population_obs_t1 - pR_plhiv_obs_t1));
vector<Type> pR_rho_obs_t1(pR_plhiv_obs_t1 / pR_population_obs_t1);
vector<Type> pR(1.0 - exp(-(pR_lambda_obs_t1 * (1.0 - pR_rho_obs_t1) / pR_rho_obs_t1 *
  (OmegaT - betaT * ritaT) + betaT)));
val -= dbinom(x_recent, n_recent, pR, true).sum();


// ANC prevalence and ART coverage model
// Note: currently this operates on the entire population vector, producing
//       lots of zeros for males and female age groups not exposed to fertility.
//       It would be more computationally efficient to project this to subset
//       of female age 15-49 age groups. But I don't know if it would be
//       meaningfully more efficient.

vector<Type> mu_anc_rho_t1(mu_rho +
  logit_anc_rho_t1_offset +
  X_ancrho * beta_anc_rho +
  Z_ancrho_x * ui_anc_rho_x);
vector<Type> anc_rho_t1(invlogit(mu_anc_rho_t1));

vector<Type> mu_anc_alpha_t1(mu_alpha +
  logit_anc_alpha_t1_offset +
  X_ancalpha * beta_anc_alpha +
  Z_ancalpha_x * ui_anc_alpha_x);
vector<Type> anc_alpha_t1(invlogit(mu_anc_alpha_t1));

// JE NOTE 6 Jan 2022: removed mu_asfr term -- should not use for aggregate ANC.
vector<Type> anc_clients_t1(population_t1 * exp(log_asfr_t1_offset));
vector<Type> anc_plhiv_t1(anc_clients_t1 * anc_rho_t1);
vector<Type> anc_already_art_t1(anc_plhiv_t1 * anc_alpha_t1);

// likelihood for ANC testing observations

vector<Type> anc_rho_obs_t1(A_anc_prev_t1 * anc_plhiv_t1 / (A_anc_prev_t1 * anc_clients_t1));
vector<Type> anc_rho_obs_t1_ll = dbinom(x_anc_prev_t1, n_anc_prev_t1, anc_rho_obs_t1, true);
val -= sum(anc_rho_obs_t1_ll);

vector<Type> anc_alpha_obs_t1(A_anc_artcov_t1 * anc_already_art_t1 / (A_anc_artcov_t1 * anc_plhiv_t1))
vector<Type> anc_alpha_obs_t1_ll = dbinom(x_anc_artcov_t1, n_anc_artcov_t1, anc_alpha_obs_t1, true);
val -= sum(anc_alpha_obs_t1_ll);

// * ART attendance model *

vector<Type> gamma_art_t1(exp(Xgamma * log_or_gamma + log_gamma_offset));
```

```
int cum_nb = 0;
for(int i = 0; i < n_nb.size(); i++){
  Type cum_exp_or_gamma_i = 0.0;
  for(int j = 0; j < n_nb[i]+1; j++)
    cum_exp_or_gamma_i += gamma_art_t1[cum_nb + i + j];
  for(int j = 0; j < n_nb[i]+1; j++)
    gamma_art_t1[cum_nb + i + j] /= cum_exp_or_gamma_i;
  cum_nb += n_nb[i];
}

vector<Type> prop_art_ij_t1((Xart_idx * prop_art_t1) * (Xart_gamma * gamma_art_t1));
vector<Type> population_ij_t1(Xart_idx * population_t1);

vector<Type> artnum_ij_t1(population_ij_t1 * prop_art_ij_t1);
vector<Type> A_j_t1(A_artattend_t1 * artnum_ij_t1);
vector<Type> sd_A_j_t1(A_artattend_t1 * vector<Type>(population_ij_t1 * prop_art_ij_t1 * (1 - prop_ar
sd_A_j_t1 = sd_A_j_t1.sqrt();

vector<Type> artnum_t1_ll = dnorm(x_artnum_t1, A_j_t1, sd_A_j_t1, true);
val -= sum(artnum_t1_ll);

// Calculate model outputs

DATA_SPARSE_MATRIX(A_out);
DATA_SPARSE_MATRIX(A_anc_out);
DATA_INTEGER(calc_outputs);
DATA_INTEGER(report_likelihood)

  if(calc_outputs) {

    vector<Type> population_t1_out(A_out * population_t1);

    vector<Type> plhiv_t1_out(A_out * plhiv_t1);
    vector<Type> rho_t1_out(plhiv_t1_out / population_t1_out);

    vector<Type> artnum_t1_out(A_out * artnum_t1);
    vector<Type> alpha_t1_out(artnum_t1_out / plhiv_t1_out);
    vector<Type> artattend_t1_out(A_out * (A_artattend_mf * artnum_ij_t1));
    vector<Type> artattend_ij_t1_out(A_art_reside_attend * artnum_ij_t1);
    vector<Type> untreated_plhiv_num_t1_out(plhiv_t1_out - artnum_t1_out);

    // Calculate number of PLHIV who attend facility in district i; denominator for artattend
    vector<Type> plhiv_attend_ij_t1((Xart_idx * plhiv_t1) * (Xart_gamma * gamma_art_t1));
    vector<Type> plhiv_attend_t1_out(A_out * (A_artattend_mf * plhiv_attend_ij_t1));
    vector<Type> untreated_plhiv_attend_t1_out(plhiv_attend_t1_out - artattend_t1_out);

    vector<Type> infections_t1_out(A_out * infections_t1);
    vector<Type> lambda_t1_out(infections_t1_out / (population_t1_out - plhiv_t1_out));

    vector<Type> anc_clients_t1_out(A_anc_out * anc_clients_t1);
    vector<Type> anc_plhiv_t1_out(A_anc_out * anc_plhiv_t1);
    vector<Type> anc_already_art_t1_out(A_anc_out * anc_already_art_t1);
```

```
    // Note: assuming that:
    //  (1) anc_known_pos is equivalent to anc_already_art
    //  (2) All ANC attendees are diagnosed and initated on ART.
    vector<Type> anc_art_new_t1_out(anc_plhiv_t1_out - anc_already_art_t1_out);
    vector<Type> anc_known_pos_t1_out(anc_already_art_t1_out);
    vector<Type> anc_tested_pos_t1_out(anc_plhiv_t1_out - anc_known_pos_t1_out);
    vector<Type> anc_tested_neg_t1_out(anc_clients_t1_out - anc_plhiv_t1_out);

    vector<Type> anc_rho_t1_out(anc_plhiv_t1_out / anc_clients_t1_out);
    vector<Type> anc_alpha_t1_out(anc_already_art_t1_out / anc_plhiv_t1_out);

    REPORT(population_t1_out);
    REPORT(rho_t1_out);
    REPORT(plhiv_t1_out);
    REPORT(alpha_t1_out);
    REPORT(artnum_t1_out);
    REPORT(artattend_t1_out);
    REPORT(artattend_ij_t1_out);
    REPORT(untreated_plhiv_num_t1_out);
    REPORT(plhiv_attend_t1_out);
    REPORT(untreated_plhiv_attend_t1_out);
    REPORT(lambda_t1_out);
    REPORT(infections_t1_out);
    REPORT(anc_clients_t1_out);
    REPORT(anc_plhiv_t1_out);
    REPORT(anc_already_art_t1_out);
    REPORT(anc_art_new_t1_out);
    REPORT(anc_known_pos_t1_out);
    REPORT(anc_tested_pos_t1_out);
    REPORT(anc_tested_neg_t1_out);
    REPORT(anc_rho_t1_out);
    REPORT(anc_alpha_t1_out);

  }

  if(report_likelihood){

    REPORT(hhs_prev_ll);
    REPORT(hhs_artcov_ll);
    REPORT(artnum_t1_ll);
    REPORT(anc_rho_obs_t1_ll);

  }

  // Adam addition to include reporting of all latent field and hyper-parameter elements

  // Hyper-parameter
  REPORT(logit_phi_rho_x)
  REPORT(log_sigma_rho_x)
  REPORT(logit_phi_rho_xs)
  REPORT(log_sigma_rho_xs)
  REPORT(logit_phi_rho_a)
  REPORT(log_sigma_rho_a)
```

```
    REPORT(logit_phi_rho_as)
    REPORT(log_sigma_rho_as)
    REPORT(log_sigma_rho_xa)
    REPORT(logit_phi_alpha_x)
    REPORT(log_sigma_alpha_x)
    REPORT(logit_phi_alpha_xs)
    REPORT(log_sigma_alpha_xs)
    REPORT(logit_phi_alpha_a)
    REPORT(log_sigma_alpha_a)
    REPORT(logit_phi_alpha_as)
    REPORT(log_sigma_alpha_as)
    REPORT(log_sigma_alpha_xa)
    REPORT(OmegaT_raw)
    REPORT(log_betaT)
    REPORT(log_sigma_lambda_x)
    REPORT(log_sigma_ancrho_x)
    REPORT(log_sigma_ancalpha_x)
    REPORT(log_sigma_or_gamma)

    // Latent field
    REPORT(beta_rho)
    REPORT(beta_alpha)
    REPORT(beta_lambda)
    REPORT(beta_anc_rho)
    REPORT(beta_anc_alpha)
    REPORT(u_rho_x)
    REPORT(us_rho_x)
    REPORT(u_rho_xs)
    REPORT(us_rho_xs)
    REPORT(u_rho_a)
    REPORT(u_rho_as)
    REPORT(u_rho_xa)
    REPORT(u_alpha_x)
    REPORT(us_alpha_x)
    REPORT(u_alpha_xs)
    REPORT(us_alpha_xs)
    REPORT(u_alpha_a)
    REPORT(u_alpha_as)
    REPORT(u_alpha_xa)
    REPORT(ui_lambda_x)
    REPORT(ui_anc_rho_x)
    REPORT(ui_anc_alpha_x)
    REPORT(log_or_gamma)

    return val;
}
```

S3 MCMC convergence and suitability

We assessed MCMC convergence and suitability using a range of graphical and numerical tests. These included the potential scale reduction factor $\hat{R}$, bulk and tail effective sample size (ESS), autocorrelation decay plots, univariate traceplots, pairs density plots, and NUTS specific divergent transition and energy assessments.

For the time being, this analysis is available from `athowes.github.io/elgm-inf/mcmc-convergence`. Once the MCMC results are finalised, the analysis will be moved to this appendix and expanded upon. The following draft text and references may be useful in that expanded write-up.

- Improved $\hat{R}$ statistic from Vehtari et al. (2021). Recommended only to use the sample if the value is less than 1.05.
- ESS and autocorrelation related to efficiency
- Traceplots helpful for diagnosis of problems
- Pairs density plots helpful for understanding relationships between parameters, including possible identifiability issues
- Energy plot from Betancourt (2017)

S4 Laplace marginals algorithm

1. Calculate the mode, Hessian at the mode, and lower Cholesky

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg\max_{\boldsymbol{\theta}} \tilde{p}_{\texttt{LA}}(\boldsymbol{\theta}, \mathbf{y}),\\ \mathbf{H} &= -\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top}\log\tilde{p}_{\texttt{LA}}(\boldsymbol{\theta}, \mathbf{y})|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}\\ \mathbf{H}^{-1} &= \mathbf{L}\mathbf{L}^\top,\end{aligned}$$

of the Laplace approximation

$$\tilde{p}_{\texttt{LA}}(\boldsymbol{\theta}, \mathbf{y}) = \frac{p(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta})}{\tilde{p}_{\texttt{G}}(\mathbf{x}\,|\,\boldsymbol{\theta}, \mathbf{y})}\Big|_{\mathbf{x}=\hat{\mathbf{x}}(\boldsymbol{\theta})}$$

where $\tilde{p}_{\texttt{G}}(\mathbf{x}\,|\,\boldsymbol{\theta}, \mathbf{y}) = \mathcal{N}(\mathbf{x}\,|\,\hat{\mathbf{x}}(\boldsymbol{\theta}), \mathbf{H}(\boldsymbol{\theta})^{-1})$ is a Gaussian approximation to $p(\mathbf{x}\,|\,\boldsymbol{\theta}, \mathbf{y})$ with mode and precision matrix given by

$$\begin{aligned}\hat{\mathbf{x}}(\boldsymbol{\theta}) &= \arg\max_{\mathbf{x}} \log p(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta}),\\ \mathbf{H}(\boldsymbol{\theta}) &= -\frac{\partial^2}{\partial x \partial x^\top}\log p(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta})|_{\mathbf{x}=\hat{\mathbf{x}}(\boldsymbol{\theta})}.\end{aligned}$$

2. Generate a set of nodes $\mathbf{u} \in \mathcal{Q}(m, k)$ and weights $\omega : \mathbf{u} \to \mathbb{R}$ from a Gauss-Hermite quadrature rule with k nodes per dimension, which are then adapted based on the mode and lower Choleksy via $\boldsymbol{\theta}(\mathbf{u}) = \hat{\boldsymbol{\theta}} + \mathbf{L}\mathbf{u}$. If possible $k \geq 3$ is preferred, though the number of grid points scales exponentially with choice of k. Then, use this quadrature rule to calculate the normalising constant $\tilde{p}_{\texttt{AQ}}(\mathbf{y})$ as follows

$$\tilde{p}_{\texttt{AQ}}(\mathbf{y}) = \sum_{\mathbf{u}\in\mathcal{Q}(m,k)} \tilde{p}_{\texttt{LA}}(\boldsymbol{\theta}(\mathbf{u}), \mathbf{y})\omega(\mathbf{u}). \tag{2}$$

3. For $i \in [n]$ generate l nodes $x_i(\mathbf{v})$ via a Gauss-Hermite quadrature rule $\mathbf{v} \in \mathcal{Q}(1, l)$ adapted based on the mode $\hat{\mathbf{x}}(\boldsymbol{\theta})_i$ and standard deviation $\sqrt{\text{diag}[\mathbf{H}(\boldsymbol{\theta})^{-1}]_i}$ of the Gaussian marginal. A value of $l \geq 4$ is recommended to enable B-spline interpolation. Then, for $x_i \in \{x_i(\mathbf{v})\}_{\mathbf{v}\in\mathcal{Q}(1,l)}$ and $\boldsymbol{\theta} \in \{\boldsymbol{\theta}(\mathbf{u})\}_{\mathbf{u}\in\mathcal{Q}(m,k)}$ calculate the modes and Hessians

$$\begin{aligned}\hat{\mathbf{x}}_{-i}(x_i, \boldsymbol{\theta}) &= \arg\max_{\mathbf{x}_{-i}} \log p(\mathbf{y}, x_i, \mathbf{x}_{-i}, \boldsymbol{\theta}),\\ \mathbf{H}_{-i,-i}(x_i, \boldsymbol{\theta}) &= -\frac{\partial^2}{\partial\mathbf{x}_{-i}\partial\mathbf{x}_{-i}^\top}\log p(\mathbf{y}, x_i, \mathbf{x}_{-i}, \boldsymbol{\theta})|_{\mathbf{x}_{-i}=\hat{\mathbf{x}}_{-i}(x_i,\boldsymbol{\theta})},\end{aligned}$$

where optimisation to obtain $\hat{\mathbf{x}}_{-i}(x_i, \boldsymbol{\theta})$ is initialised at $\hat{\mathbf{x}}(\boldsymbol{\theta})_{-i}$.

4. For $x_i \in \{x_i(\mathbf{v})\}_{\mathbf{v}\in\mathcal{Q}(1,l)}$ calculate

$$\tilde{p}_{\texttt{AQ}}(x_i\,|\,\mathbf{y}) = \frac{\tilde{p}_{\texttt{LA}}(x_i, \mathbf{y})}{\tilde{p}_{\texttt{AQ}}(\mathbf{y})}. \tag{3}$$

where

$$\tilde{p}_{\texttt{LA}}(x_i, \mathbf{y}) = \sum_{\mathbf{u}\in\mathcal{Q}(m,k)} \tilde{p}_{\texttt{LA}}(x_i, \boldsymbol{\theta}(\mathbf{u}), \mathbf{y})\omega(\mathbf{u}).$$

and

$$\tilde{p}_{\texttt{LA}}(x_i, \boldsymbol{\theta}, \mathbf{y}) = \frac{p(x_i, \mathbf{x}_{-i}, \boldsymbol{\theta}, \mathbf{y})}{\tilde{p}_{\texttt{G}}(\mathbf{x}_{-i}\,|\,x_i, \boldsymbol{\theta}, \mathbf{y})}\Big|_{\mathbf{x}_{-i}=\hat{\mathbf{x}}_{-i}(x_i,\boldsymbol{\theta})}.$$

Although Equation 3 can be calculated using the estimate of the evidence in Equation 2 it is more numerically accurate to use the estimate

$$\tilde{p}_{\texttt{AQ}}(\mathbf{y}) = \sum_{\mathbf{v}\in\mathcal{Q}(1,l)} \tilde{p}_{\texttt{LA}}(x_i(\mathbf{v}), \mathbf{y})\omega(\mathbf{v})$$

5. Given $\{x_i(\mathbf{v}), \tilde{p}_{\mathtt{AQ}}(x_i(\mathbf{v}) \,|\, \mathbf{y})\}_{\mathbf{v} \in \mathcal{Q}(1,l)}$ create a spline interpolant to each posterior marginal on the log-scale. Samples, and thereby relevant posterior marginal summaries, may be obtained using inverse transform sampling.

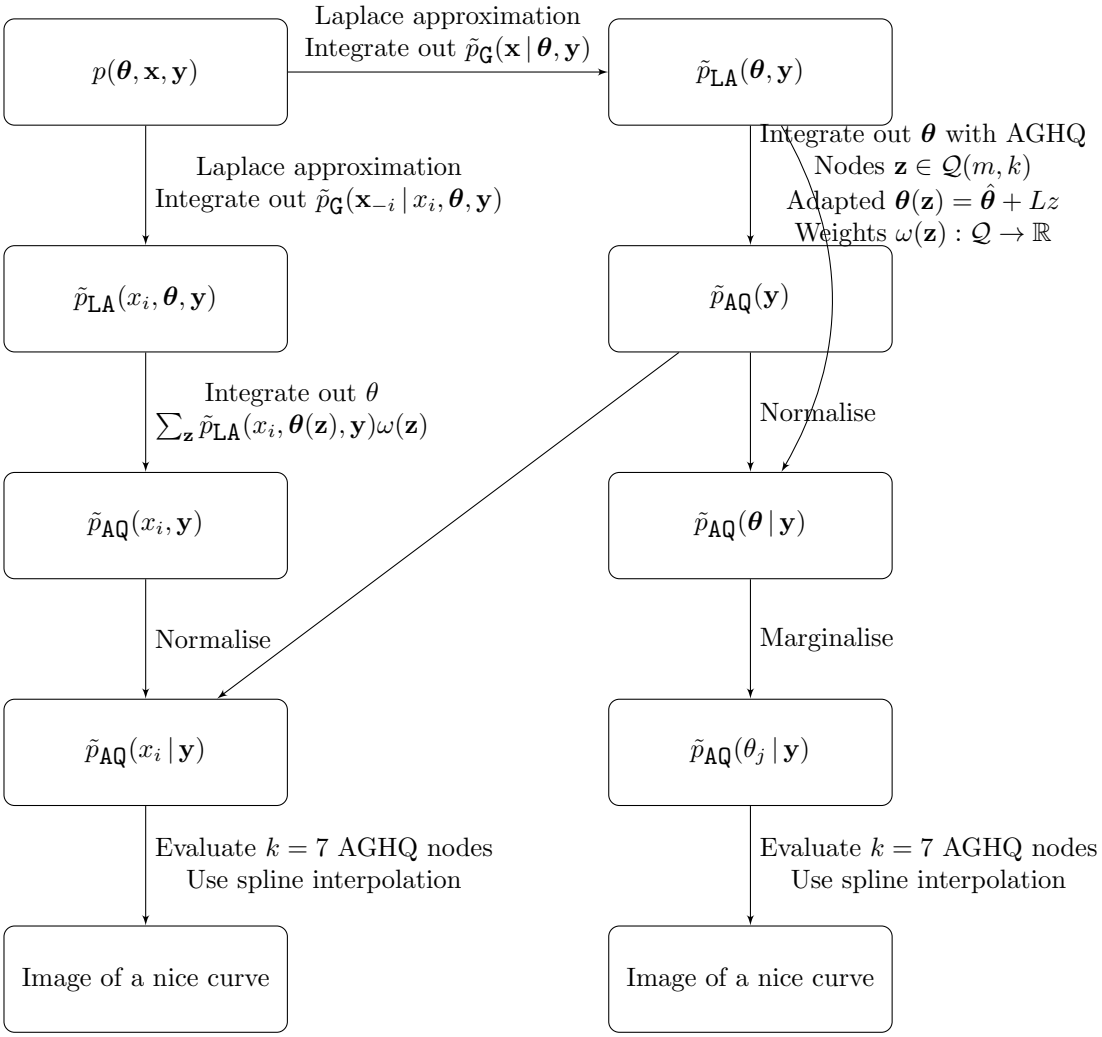

Figure S2: Flowchart describing the algorithm we propose (work in progress).

References

Besag, Julian, Jeremy York, and Annie Mollié. 1991. "Bayesian image restoration, with two applications in spatial statistics." *Annals of the Institute of Statistical Mathematics* 43 (1): 1–20.

Betancourt, Michael. 2017. "A Conceptual Introduction to Hamiltonian Monte Carlo." *arXiv Preprint arXiv:1701.02434*.

Eaton, Jeffrey W, Tim Brown, Robert Puckett, Robert Glaubius, Kennedy Mutai, Le Bao, Joshua A Salomon, John Stover, Mary Mahy, and Timothy B Hallett. 2019. "The Estimation and Projection Package Age-Sex Model and the r-Hybrid Model: New Tools for Estimating HIV Incidence Trends in Sub-Saharan Africa." *AIDS (London, England)* 33 (Suppl 3): S235.

Eaton, Jeffrey W., Laura Dwyer-Lindgren, Steve Gutreuter, Megan O'Driscoll, Oliver Stevens, Sumali Bajaj, Rob Ashton, et al. 2021. "Naomi: A New Modelling Tool for Estimating HIV Epidemic Indicators at the District Level in Sub-Saharan Africa." *Journal of the International AIDS Society* 24 (S5): e25788. https://doi.org/https://doi.org/10.1002/jia2.25788.

Freni-Sterrantino, Anna, Massimo Ventrucci, and Håvard Rue. 2018. "A Note on Intrinsic Conditional Autoregressive Models for Disconnected Graphs." *Spatial and Spatio-Temporal Epidemiology* 26: 25–34.

Kish, Leslie. 1965. "Survey Sampling."

Kristensen, Kasper, Anders Nielsen, Casper W Berg, Hans Skaug, Bradley M Bell, et al. 2016. "TMB: Automatic Differentiation and Laplace Approximation." *Journal of Statistical Software* 70 (i05).

Lumley, Thomas. 2004. "Analysis of Complex Survey Samples." *Journal of Statistical Software* 9: 1–19.

Simpson, Daniel, Håvard Rue, Andrea Riebler, Thiago G Martins, and Sigrunn H Sørbye. 2017. "Penalising Model Component Complexity: A Principled, Practical Approach to Constructing Priors." *Statistical Science* 32 (1): 1–28.

Stover, J, P Johnson, T Hallett, M Marston, R Becquet, and IM Timaeus. 2010. "The Spectrum Projection Package: Improvements in Estimating Incidence by Age and Sex, Mother-to-Child Transmission, HIV Progression in Children and Double Orphans." *Sexually Transmitted Infections* 86 (Suppl 2): ii16–21.

Vehtari, Aki, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. 2021. "Rank-Normalization, Folding, and Localization: An Improved r for Assessing Convergence of MCMC (with Discussion)." *Bayesian Analysis* 16 (2): 667–718.