# Appendix to "Integrated nested Laplace approximations for extended latent Gaussian models with application to the Naomi HIV model"

Corresponding author: Adam Howes (ath19@ic.ac.uk)

## Contents

# 1 Simplified Naomi model description

In this section we describes more fully the simplified Naomi model considered in the main text. To do so, we draw substantially from the supplementary material of Jeffrey W. Eaton et al. (2021).

## 1.1 Background

### 1.1.1 Indexing

Let $\nu$ the most recent national household survey with HIV testing which has taken place in the country of interest at time $t = T_1$ Let $x$ 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 bands. As short-hand, we write $a = l$ to refer to the age band with lower bound $l$, e.g. $a = 20$ for $a =$ 20-25. We index the following quantities by district, sex and age-band: population size $N_{x,s,a}$, HIV prevalence $\rho_{x,s,a}$, ART coverage $\alpha_{x,s,a}$, annual HIV incidence rate $\lambda_{x,s,a}$, and proportion of HIV positive persons recently infected $\kappa_{x,s,a}$. 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 adults.

### 1.1.2 Structured random effects

Let $u$ be a generic random effect. We use use structured random effects to share information across units assessed as being similar, such as neighbouring districts or adjacent age-bands. 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 $i$th unit removed, $j \sim i$ if the units 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. 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 $v^\star$ with proportion $\phi \in (0, 1)$ and spatially unstructured IID component $w^\star$ with proportion $1 - \phi$, both scaled to have generalised variance equal to one, and $\sigma > 0$ is the marginal standard deviation such that

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

We specify the first order auto-regressive model by $u \sim \text{AR1}(\sigma, \phi)$ such that

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

where $\epsilon_i \sim \mathcal{N}(0, 1)$ is Gaussian white noise, $|\rho| < 1$ is the lag-one correlation parameter.

### 1.1.3 Complex survey design

We assume the household survey was run according to a complex survey design where each individual $j \in U$ has probability $\pi_j \in (0, 1)$ of appearing in the sample $S \subseteq U$. Suppose we observe an outcome $y_j$ for $j \in S$. Let $w_j = 1/\pi_j$ be design weights, then the weighted mean

$$\hat{y} = \frac{\sum_{j\in S} w_j y_j}{\sum_{j\in S} w_j}$$

Define Kish ESS. We make these computations using the `survey` R package (Lumley 2004).

## 1.2 Process specification

### 1.2.1 HIV prevalence

We model HIV prevalence on the logit scale using the linear predictor

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

where $\beta_0^\rho \sim \mathcal{N}(0,5)$ is an intercept term, $\beta_s^{\rho,s=\text{M}} \sim \mathcal{N}(0,5)$ is the difference in logit prevalence for men compared to women, $u_a^\rho \sim \text{AR1}(\sigma_A^\rho, \phi_A^\rho)$ are age random effects for women, $u_a^{\rho,s=\text{M}} \sim \text{AR1}(\sigma_{AS}^\rho, \phi_{AS}^\rho)$ are age random effects for the difference in logit prevalence for men compared to women age $a$, $u_x^\rho \sim \text{BYM2}(\sigma_X^\rho, \phi_X^\rho)$ are spatial random effects for women, $u_x^{\rho,s=\text{M}} \sim \text{BYM2}(\sigma_{XS}^\rho, \phi_{XS}^\rho)$ are spatial random effects for the difference in logit prevalence for men compared to women in district $x$, $u_x^{\rho,a<15} \sim \text{ICAR}(0, \sigma_{XA}^\rho)$ are spatial random effects for the ratio of paediatric prevalence to adult women prevalence, and $\eta_{R_x,s,a}^\rho$ are fixed offsets specifying assumed odds ratios for prevalence outside the age ranges for which data are available. We use the prior distributions $\mathcal{N}^+(0, 2.5)$ for all standard deviation terms, $\mathcal{U}(-1,1)$ for all AR1 correlation parameters, and $\text{Beta}(0.5, 0.5)$ for all BYM2 proportion parameters.

### 1.2.2 ART coverage

We model ART coverage 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 1.2.1 above.

### 1.2.3 HIV incidence rate

We model HIV incidence rate using the linear predictor

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

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, $\beta_S^{\lambda,s=\text{M}} \sim \mathcal{N}(0,5)$ is the log incidence rate ratio for men compared to women, $\rho_x^{\text{15-49}}$ is the HIV prevalence among adults 15-49 calculated by aggregating age-specific HIV prevalences

$$\rho_x^{\text{15-49}} = \frac{\sum_{s\in\{\text{F,M}\}} \sum_{a=15}^{45} N_{x,s,a} \cdot \rho_{x,s,a}}{\sum_{s\in\{\text{F,M}\}} \sum_{a=15}^{45} N_{x,s,a}}, \tag{2}$$

$\alpha_x^{\text{15-49}}$ is the ART coverage among adults 15-49 calculated by aggregating age-specific ART coverages

$$\alpha_x^{\text{15-49}} = \frac{\sum_{s\in\{\text{F,M}\}} \sum_{a=15}^{45} N_{x,s,a} \cdot \rho_{x,s,a} \cdot \alpha_{x,s,a}}{\sum_{s\in\{\text{F,M}\}} \sum_{a=15}^{45} N_{x,s,a} \cdot \rho_{x,s,a}}, \tag{3}$$

$\omega$ 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), $u_x^\lambda \sim \mathcal{N}(0, \sigma^\lambda)$ with $\sigma^\lambda \sim \mathcal{N}^+(0,1)$ are IID spatial random effects, and $\eta_{R_x,s,a}^\lambda$ specify log incidence rate ratios by sex and age group calculated from Spectrum model output. Note that the HIV incidence rate process (Equation 1) depends on those of the HIV prevalence (Section 1.2.1) and the ART coverage (Section 1.2.2) via the adult aggregate indicators (Equations 2 and 3) violating the conditions required being within the latent Gaussian model (LGM) class as defined by Rue, Martino, and Chopin (2009).

### 1.2.4 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

$$\text{logit}(\rho_{x,a}^{\text{ANC}}) = \text{logit}(\rho_{x,F,a}) + \beta^{\rho^{\text{ANC}}} + u_x^{\rho^{\text{ANC}}} + \eta_{R_x,a}^{\rho^{\text{ANC}}}, \tag{4}$$

$$\text{logit}(\alpha_{x,a}^{\text{ANC}}) = \text{logit}(\alpha_{x,F,a}) + \beta^{\alpha^{\text{ANC}}} + u_x^{\alpha^{\text{ANC}}} + \eta_{R_x,a}^{\alpha^{\text{ANC}}}, \tag{5}$$

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 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 $R_x$. Dependence here too on $\rho_{x,F,a}$ and $\alpha_{x,F,a}$ in Equations 4 and 5 violates the conditions required to be a LGM.

#### 1.2.5 ART attendance

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

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

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 1.2.5 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 on $\gamma_{x,x'}$ of 1.8%, such that $(100 - 1.8 \times \text{ne}(x))\%$ 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

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

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

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

### 1.3 Likelihood specification

#### 1.3.1 Household survey data

For $\theta \in \{\rho, \alpha, \kappa\}$, the household survey $\nu$ furnishes weighted observations $\hat{\theta}_{\{x\},\{s\},\{a\}}$ with respective Kish effective sample sizes $m^{\hat{\theta}}_{\{x\},\{s\},\{a\}}$, and observed number of cases

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

For HIV prevalence and ART coverage we use the binomial working likelihoods

$$y^{\hat{\rho}}_{\{x\},\{s\},\{a\}} \sim \text{xBin}(m^{\hat{\rho}}_{\{x\},\{s\},\{a\}}, \rho_{\{x\},\{s\},\{a\}}), \tag{6}$$

$$\rho_{\{x\},\{s\},\{a\}} = \frac{\sum_{x \in \{x\}} \sum_{s \in \{s\}} \sum_{a \in \{a\}} N_{x,s,a} \cdot \rho_{x,s,a}}{\sum_{x \in \{x\}} \sum_{s \in \{s\}} \sum_{a \in \{a\}} N_{x,s,a}}, \tag{7}$$

and

$$y^{\hat{\alpha}}_{\{x\},\{s\},\{a\}} \sim \text{xBin}(m^{\hat{\alpha}}_{\{x\},\{s\},\{a\}}, \alpha_{\{x\},\{s\},\{a\}}), \tag{8}$$

$$\alpha_{\{x\},\{s\},\{a\}} = \frac{\sum_{x \in \{x\}} \sum_{s \in \{s\}} \sum_{a \in \{a\}} N_{x,s,a} \cdot \rho_{x,s,a} \cdot \alpha_{x,s,a}}{\sum_{x \in \{x\}} \sum_{s \in \{s\}} \sum_{a \in \{a\}} N_{x,s,a} \cdot \rho_{x,s,a}}. \tag{9}$$

For recent infections we also use a binomial working likelihood

$$y^{\hat{\kappa}}_{\{x\},\{s\},\{a\}} \sim \text{xBin}(m^{\hat{\kappa}}_{\{x\},\{s\},\{a\}}, \kappa_{\{x\},\{s\},\{a\}}), \tag{10}$$

$$\kappa_{x,s,a} = 1 - \exp(-\lambda_{x,s,a} \cdot \frac{1 - \rho_{x,s,a}}{\rho_{x,s,a}} \cdot (\Omega_T - \beta_T) - \beta_T), \tag{11}$$

where $\kappa_{x,s,a}$ are the predicted proportion recently infected among HIV positive persons, $\Omega_T \sim \mathcal{N}(\Omega_{T_0}, \sigma^{\Omega_T})$ is the mean duration of recent infection (MDRI), and $\beta_T \sim \mathcal{N}(\beta_{T_0}, \sigma^{\beta_T})$ is the false recent ratio (FRR). We use an informative prior on $\Omega_T$ based on the characteristics of the recent infection testing algorithm (RITA). For PHIA surveys this is $\Omega_{T_0} = 130$ days and $\sigma^{\Omega_T} = 6.12$ days. For PHIA surveys we assume there is no false recency, such that $\beta_{T_0} = 0.0$ and $\sigma^{\beta_T} = 0.0$.

### 1.3.2 ANC testing data

For women 15-49 the predicted number of ANC clients $\Psi_{x,a}$ is a log-linear model

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

where $N_{x,\mathrm{F},a}$ are the female population sizes, $\psi_{R_x,a}$ are ASFR in Spectrum region $R_x$ at time $t$, $\beta^{\psi}$ are the log rate ratio for the number of ANC clients relative to the predicted fertility, $u_x^{\psi} \sim \mathcal{N}(0, \sigma^{\psi})$ are district random effects.

We include ANC testing data for the year of the most recent survey $Y[T_1]$. Let $W^{\mathrm{ANC}}_{\{x\},Y[t]}$ be the number of ANC clients, $X^{\mathrm{ANC}}_{\{x\},Y[t]}$ the number of those with ascertained status, $y^{\mathrm{ANC}}_{\{x\},Y[t]}$ the number of those with positive status (either known or tested) and $Z^{\mathrm{ANC}}_{\{x\},Y[t]}$ the number of ANC clients already on ART prior to first ANC, such that $W^{\mathrm{ANC}}_{x,Y[t]} \geq X^{\mathrm{ANC}}_{x,Y[t]} \geq y^{\mathrm{ANC}}_{x,Y[t]} \geq Z^{\mathrm{ANC}}_{x,Y[t]}$ When ANC testing data are only available for part of a given year, we denote $m^{\mathrm{ANC}}_{Y[t]} \in \{1, \ldots, 12\}$ the number of months of reported data reflected in counts for year $Y[t]$.

The observed number of HIV positive and already on ART among ANC clients at $Y[T_1]$ is modelled by

$$Y^{\mathrm{ANC}}_{\{x\}Y[T_1]} \sim \mathrm{Bin}\left(X^{\mathrm{ANC}}_{\{x\}Y[T_1]}, \rho^{\mathrm{ANC}}_{\{x\},\{15,\ldots 45\}}\right)$$
$$Z^{\mathrm{ANC}}_{\{x\}Y[T_1]} \sim \mathrm{Bin}\left(Y^{\mathrm{ANC}}_{\{x\}Y[T_1]}, \alpha^{\mathrm{ANC}}_{\{x\},\{15,\ldots 45\}}\right)$$

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

$$\rho^{\mathrm{ANC}}_{\{x\}\{a\}} = \frac{\sum_{x\in\{x\}}\sum_{a\in\{a\}} \Psi_{x,a} \cdot \rho^{\mathrm{ANC}}_{x,a}}{\sum_{x\in\{x\}}\sum_{a\in\{a\}} \Psi_{x,a}}$$
$$\alpha^{\mathrm{ANC}}_{\{x\}\{a\}} = \frac{\sum_{x\in\{x\}}\sum_{a\in\{a\}} \Psi_{x,a} \cdot \rho^{\mathrm{ANC}}_{x,a} \cdot \alpha^{\mathrm{ANC}}_{x,a}}{\sum_{x\in\{x\}}\sum_{a\in\{a\}} \Psi_{x,a} \cdot \rho^{\mathrm{ANC}}_{x,a}}$$

### 1.3.3 Number receiving ART

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

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

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

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

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

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

and

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

## 1.4 Identifiability constaints

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

- If survey data on ART coverage by age and sex are not available then we set $u_a^\alpha = 0$ and $u_{a,s=\mathrm{M}}^\alpha = 0$ and use the average age/sex pattern of ART coverage from the Spectrum offset $\eta_{R_x,s,a}^\alpha$.
- If no ART data (survey or ART programme) are available at $T_1$ 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

$$\text{logit}\left(\alpha_{x,s,a}\right) = u_x^\alpha + \eta_{R_x,s,a}^\alpha$$

- If survey data on recent HIV infection are not included in the model, then $\beta_0^\lambda = \beta_S^{\lambda,s=\mathrm{M}} = u_x^\lambda = 0$. 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.

# 2 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 indentifiability issues
- Energy plot from Betancourt (2017)

## 3 C++ TMB user template

```
// #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;
}
```

# 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.

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

Rue, Håvard, Sara Martino, and Nicolas Chopin. 2009. "Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 71 (2): 319–92.

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.