

# GiRaF: a toolbox for Gibbs Random Fields analysis

Julien Stoehr<sup>\*1</sup>, Pierre Pudlo<sup>2</sup>, and Nial Friel<sup>1</sup>

<sup>1</sup>University College Dublin

<sup>2</sup>Aix-Marseille Université

February 24, 2016

## Abstract

**GiRaF** package offers various tools for the analysis of Gibbs (or discrete Markov) random fields. The latter form a class of intractable statistical models since, due to the Markovian dependence structure, the normalising constant of the fields cannot be computed using standard analytical or numerical methods. **GiRaF** substantially lowers the barrier for practitioners aiming at analysing such Gibbs random fields. It contains exact methods for small lattices and several approximate methods for larger lattices that make the analysis easier for practitioners. This document provides a short description of the models which are solved as well as short examples of the major functionalities.

## Contents

<b>1</b>	<b>Gibbs Random Fields</b>	<b>2</b>
1.1	Gibbs-Markov equivalence . . . . .	2
1.2	GiRaF inputs . . . . .	4
<b>2</b>	<b>Recursive computing and partition function</b>	<b>5</b>
2.1	NC.mrf . . . . .	5
<b>3</b>	<b>Sampling from a Gibbs random field</b>	<b>7</b>
3.1	exact.mrf . . . . .	7
3.2	sampler.mrf . . . . .	9
	<b>References</b>	<b>11</b>

---

\*julien.stoehr@ucd.ie

## Introduction

Gibbs (or discrete Markov) random fields have been used in many practical settings, surged by the development in the statistical community since the 1970's. Notable examples are the autologistic model (Besag, 1974) and its extension the Potts model used to describe the spatial dependency of discrete random variables (*e.g.*, shades of grey or colors) on the vertices of an undirected graph (*e.g.*, a regular grid of pixels). Despite their popularity, running a statistical analysis is difficult due to the dependence structure that leads to a considerable computational curse: the normalising constant is intractable and cannot be computed with standard analytical or numerical methods.

The **GiRaF** package aims at providing a toolbox for practitioners that makes the analysis of such models easier. The primary tool is the computation of the intractable normalising constant for small rectangular lattices (`NC.mrf`) based on the recursive algorithm of Reeves and Pettitt (2004). In practice, this function allows to have ground truth against which to compare. Beside the latter function, the **GiRaF** package contains methods that give exact sample from the likelihood for small enough rectangular lattices (`exact.mrf`, Friel and Rue (2007)) or approximate samples from the likelihood using MCMC samplers: the Gibbs sampler or the Swendsen-Wang algorithm (`sampler.mrf`) for large lattices.

The **GiRaF** package provides R functions based on C++ code for the statistical analysis of Gibbs random fields. As befits an introduction to the **GiRaF** package, a brief overview of Gibbs distribution is presented in Section 1 while pointing out the convention used throughout the package. Then we present the major functions through detailed examples, namely recursive computing in Section 2 and sampling methods in Section 3.

## 1 Gibbs Random Fields

### 1.1 Gibbs-Markov equivalence

A discrete random field  $\mathbf{X}$  is a collection of random variables  $X_i$  indexed by a finite set  $\mathcal{S} = \{1, \dots, n\}$ , whose elements are called sites, and taking values in a finite state space  $\mathcal{X}_i := \{0, \dots, K-1\}$ , interpreted as colors. For a given subset  $A \subset \mathcal{S}$ ,  $\mathbf{X}_A$  and  $\mathbf{x}_A$  respectively define the random process on  $A$ , *i.e.*,  $\{X_i, i \in A\}$ , and a realisation of  $\mathbf{X}_A$ . Denotes  $\mathcal{S} \setminus A = -A$  the complement of  $A$  in  $\mathcal{S}$ . When modeling a digital image, the sites lie on a regular 2D-grid of pixels, and their dependency is given by an undirected graph  $\mathcal{G}$  which induces a topology on  $\mathcal{S}$ : by definition, sites  $i$  and  $j$  are adjacent or are neighbor of each other if and only if  $i$  and  $j$  are linked by an edge in  $\mathcal{G}$ . A random field  $\mathbf{X}$  is a Markov random field with respect to  $\mathcal{G}$ , if for all configuration  $\mathbf{x}$  and for all sites  $i$

$$\mathbf{P}(X_i = x_i \mid \mathbf{X}_{-i} = \mathbf{x}_{-i}) = \mathbf{P}(X_i = x_i \mid \mathbf{X}_{\mathcal{N}(i)} = \mathbf{x}_{\mathcal{N}(i)}), \quad (1)$$



**Figure 1:** First and second order neighborhood graphs  $\mathcal{G}$ . (a) The four closest neighbors graph  $\mathcal{G}_4$ . Neighbors of the vertex in black are represented by vertices in gray. (b) The eight closest neighbors graph  $\mathcal{G}_8$ . Neighbors of the vertex in black are represented by vertices in gray.

where  $\mathcal{N}(i)$  denotes the set of all the adjacent sites to  $i$  in  $\mathcal{G}$ . The Hammersley-Clifford theorem states that if the distribution of a Markov random field with respect to a graph  $\mathcal{G}$  is positive for all configuration  $\mathbf{x}$  then it admits a Gibbs representation for the same topology (see for example [Grimmett \(1973\)](#), [Besag \(1974\)](#) and for a historical perspective [Clifford \(1990\)](#)), namely a probability measure  $\pi$  on  $\mathcal{X} = \prod_{i=1}^n \mathcal{X}_i$  given by

$$\pi(\mathbf{x} | \psi, \mathcal{G}) = \frac{1}{Z(\psi, \mathcal{G})} \exp\{-H(\mathbf{x} | \psi, \mathcal{G})\}, \quad (2)$$

where  $\psi$  is a vector of parameters and  $H$  denotes the energy function or Hamiltonian.

The **GiRaF** package solely focuses on Potts model, that is a pairwise Markov random field whose Hamiltonian linearly depends on the parameter  $\psi = \{\alpha, \beta\}$ , where  $\alpha$  stands for the parameter on sites and  $\beta$  stands for the parameter on edges. More precisely,

$$H(\mathbf{x} | \alpha, \beta, \mathcal{G}) = - \sum_{i=1}^n \sum_{k=0}^{K-1} \alpha_k \mathbf{1}\{x_i = k\} - \sum_{i \sim j} \beta_{ij} \mathbf{1}\{x_i = x_j\}, \quad (3)$$

where the above sum  $\sum_{i \sim j}$  ranges the set of edges of the graph  $\mathcal{G}$ .

At that stage, the **GiRaF** package handles homogeneous<sup>1</sup> and potentially anisotropic<sup>2</sup> Potts models defined on a rectangular  $n = h \times w$  lattice, with  $h \leq w$ . Lattice points are ordered from top to bottom in each column and columns from left to right. The **GiRaF** package contains two widely used adjacency structures, namely the graph  $\mathcal{G}_4$  (first order lattice), respectively  $\mathcal{G}_8$  (second order lattice), for which the neighborhood of a site is composed of the four, respectively eight, closest sites on a two-dimensional regular lattice, except on the boundaries of the lattice, see Figure 1.

Within the set of allowed distributions, the default model in **GiRaF** is set to the 2-state Potts

<sup>1</sup>Parameters  $\alpha$  and  $\beta$  are independent of the relative position of sites or edges

<sup>2</sup>Parameter  $\beta$  depends on the orientation of the edges.

models with a first order dependency structure  $\mathcal{G}_4$ , namely

$$\pi(\mathbf{x} | \beta, \mathcal{G}_4) = \frac{1}{Z(\beta, \mathcal{G}_4)} \exp \left( \sum_{i \sim_j} \beta_{ij} \mathbf{1}\{x_i = x_j\} \right).$$

## 1.2 GiRaF inputs

Most of the functions in the **GiRaF** package rely on the following inputs which are detailed throughout this Section.

<b>h, w</b>	respectively the number of rows and columns of the rectangular lattice $\mathcal{G}$ .
<b>param</b>	numeric entry setting the interaction parameter $\beta$ (edges parameter). See details below.
<b>ncolors</b>	the number $K$ of states (or colors) for the discrete random variables $\mathbf{X}_i$ . By default, $\text{ncolors} = 2$ .
<b>nei</b>	the number of neighbors. The latter must be one of $\text{nei} = 4$ or $\text{nei} = 8$ , which respectively correspond to $\mathcal{G}_4$ and $\mathcal{G}_8$ . By default, $\text{nei} = 4$ .





Optional inputs

<b>pot</b>	numeric entry setting homogeneous potential $\alpha$ on singletons. By default, $\text{pot} = \text{NULL}$ .
<b>top, left, bottom, right, corner</b>	numeric entry setting constant borders for lattice $\mathcal{G}$ . By default, $\text{top} = \text{NULL}$ , $\text{left} = \text{NULL}$ , $\text{bottom} = \text{NULL}$ , $\text{right} = \text{NULL}$ , $\text{corner} = \text{NULL}$ . See details below.

**Interaction parameter (param)** The **GiRaF** packages allows to set an isotropic or anisotropic interaction parameter  $\beta$  via the entry `param`. In the isotropic configuration, `param` is simply a scalar while in the anisotropic case `param` is a vector  $c(\beta_1, \beta_2)$  or  $c(\beta_1, \beta_2, \beta_3, \beta_4)$  if  $\mathcal{G} = \mathcal{G}_4$  or  $\mathcal{G} = \mathcal{G}_8$ , respectively. The parameter  $\beta_{ij}$  in (3) stands for the component of `param` corresponding to the direction defined by the edge  $(i, j)$  but does not depend on the actual position of sites  $i$  and  $j$ , that is, given two edges  $(i_1, j_1)$  and  $(i_2, j_2)$  defining the same direction,  $\beta_{i_1, j_1} = \beta_{i_2, j_2}$  (see Table 1).

**Constant borders** The **GiRaF** package allows lattices to have fixed borders. More precisely, each site on the boundary of the lattice has neighbors with respect to the topology induced by  $\mathcal{G}$  that are set to constants. Setting the borders can be done using the optional inputs  $\text{top} = c(t_1, \dots, t_w)$ ,  $\text{left} = c(\ell_1, \dots, \ell_h)$ ,  $\text{bottom} = c(b_1, \dots, b_w)$ ,  $\text{right} = c(r_1, \dots, r_h)$  and  $\text{corner} = c(c_1, \dots, c_4)$  (see Figure 2). In the absence of a specified border, these will take the default value `NULL`.

**Table 1:** Interaction parametrisation for a homogeneous Gibbs random field for the isotropic and anisotropic cases. The table gives values of the parameter  $\beta_{ij}$  corresponding to the orientation of the edge  $(i, j)$ .

Orientation of edge $(i, j)$				
Dependency graph	$\mathcal{G}_4$ $\mathcal{G}_8$	$\mathcal{G}_4$ $\mathcal{G}_8$	$\mathcal{G}_8$	$\mathcal{G}_8$
Isotropic Gibbs	$\beta$			
Anisotropic Gibbs	$\beta_1$	$\beta_2$	$\beta_3$	$\beta_4$

## 2 Recursive computing and partition function

The inherent difficulty of Gibbs random fields arises from the intractable normalizing constant, called the partition function, defined by

$$Z(\psi, \mathcal{G}) = \sum_{\mathbf{x} \in \mathcal{X}} \exp\{-H(\mathbf{x} | \psi, \mathcal{G})\}. \quad (4)$$

The latter is a summation over the numerous possible realizations of the random field  $\mathbf{X}$ , that cannot be computed directly except for small grids and small number of colors  $K$ .

**GiRaF** contains generalised recursion proposed by [Reeves and Pettitt \(2004\)](#) for computing the normalising constant of general factorisable models such as the autologistic or the Potts model (NC.mrf). This method, based on an algebraic simplification due to the reduction in dependence arising from the Markov property, applies to lattices with a small number  $h$  of rows. Note that the complexity of the algorithm is exponential in the number of rows ( $\mathcal{O}(K^h)$ ) and linear in the number of columns. Hence the current limitation is  $K^h \leq 2^{25}$  for computing the normalising constant.

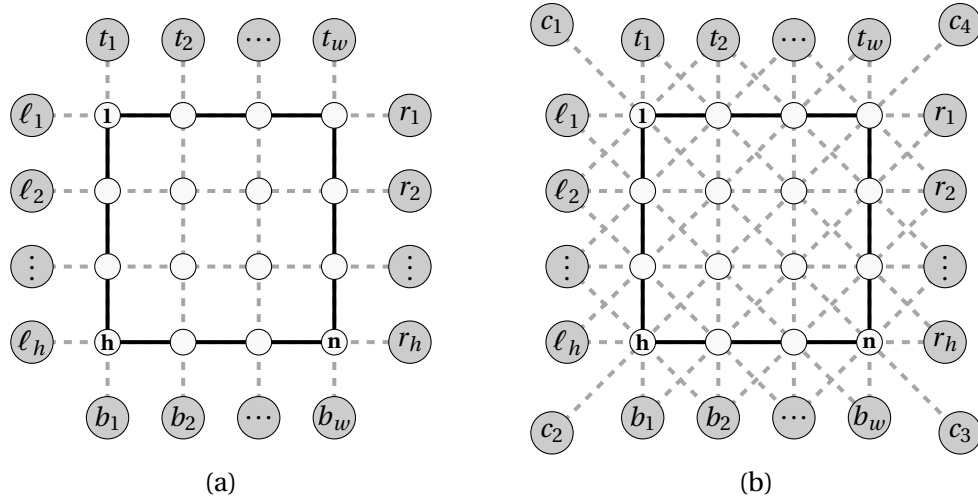
### 2.1 NC.mrf

The function NC.mrf computes the partition function (4) of a  $K$ -state Potts model defined on a rectangular  $h \times w$  lattices ( $h \leq w$ ), with either a first order or a second order dependency structure (see Figure 1) and a small number of rows (up to 25 for 2-states models).

**Usage:**

```
NC.mrf(h, w, param, ncolors = 2, nei = 4, pot = NULL, top = NULL,
left = NULL, bottom = NULL, right = NULL, corner = NULL)
```

**Inputs:** see Section 1.2.



**Figure 2:** First and second order neighborhood graphs  $\mathcal{G}$ . (a) The four closest neighbors graph  $\mathcal{G}_4$ . Neighbors of the vertex in black are represented by vertices in gray. (b) The eight closest neighbors graph  $\mathcal{G}_8$ . Neighbors of the vertex in black are represented by vertices in gray.

**Example:** # *NC.mrf*

# Dimension of the lattice

height <- 8

width <- 10

# Interaction parameter

Beta <- 0.6 # Isotropic configuration

# Beta <- c(0.6, 0.6) # Anisotropic configuration for a first  
# order dependency structure (nei = 4).

# Beta <- c(0.6, 0.6, 0.6, 0.6) # Anisotropic configuration for a second  
# order dependency structure (nei = 8).

# Number of colors. Automatically set to 2 if not specified.

K <- 2

# Number of neighbors. Automatically set to 4 if not specified.

G <- 4

# Optional potential on sites. Automatically set to NULL if not specified

potential <- runif(K,-1,1)

# Optional borders. Automatically set to NULL if not specified

Top <- Bottom <- sample(0:(K-1), width, replace = TRUE)

Left <- Right <- sample(0:(K-1), height, replace = TRUE)

Corner <- sample(0:(K-1), 4, replace = TRUE)

# Partition function for the default setting

```

NC.mrf(h = height, w = width, param = Beta)
# When specifying the number of colors and neighbors
NC.mrf(h = height, w = width, ncolors = K, nei = G, param = Beta)
# When specifying an optional potential on sites
NC.mrf(h = height, w = width, ncolors = K, nei = G, param = Beta,
       pot = potential)
# When specifying possible borders. The users will omit to mention all
# the non-existing borders
NC.mrf(h = height, w = width, ncolors = K, nei = G, param = Beta,
       top = Top, left = Left, bottom = Bottom, right = Right,
       corner = Corner)

```

### 3 Sampling from a Gibbs random field

Sampling from a Gibbs distribution can be a daunting task due to the correlation structure on a high dimensional space, and standard Monte Carlo methods are impracticable except for very specific cases. For rectangular lattices with a small number of row, [Friel and Rue \(2007\)](#) have extended the recursive algorithm of [Reeves and Pettitt \(2004\)](#) to allow exact sampling from the likelihood (`exact.mrf`). But for larger lattices, Markov chain Monte Carlo (MCMC) methods have played a dominant role in dealing with such intractable likelihood, the idea being to generate a Markov chain whose stationary distribution is the distribution of interest. The **GiRaF** package contains two widely used MCMC procedure to give approximate samples: the Gibbs sampler and the Swendsen-Wang algorithm (`sampler.mrf`).

#### 3.1 `exact.mrf`

The function `exact.mrf` gives exact sample from the likelihood (2) of a  $K$ -state Potts model defined on a rectangular  $h \times w$  lattice ( $h \leq w$ ), with either a first order or a second order dependency structure (see Figure 1) and a small number of rows (up to 19 for 2-states models). More generally, due to the complexity of the algorithm, the current limitation for exact sampling is  $K^h \leq 2^{19}$ .

**Usage:**

```

exact.mrf(h, w, param, ncolors = 2, nei = 4, pot = NULL,
         top = NULL, left = NULL, bottom = NULL,
         right = NULL, corner = NULL, view = FALSE)

```

**Inputs:** see also Section 1.2.

**view** Logical value indicating whether the draw should be printed. Do not display the optional borders.

```

Example: # exact.mrf
# Dimension of the lattice
height <- 8
width <- 10

# Interaction parameter
Beta <- 0.6 # Isotropic configuration
# Beta <- c(0.6, 0.6) # Anisotropic configuration for a first
# order dependency structure (nei = 4).
# Beta <- c(0.6, 0.6, 0.6, 0.6) # Anisotropic configuration for a second
# order dependency structure (nei = 8).

# Number of colors. Automatically set to 2 if not specified.
K <- 2
# Number of neighbors. Automatically set to 4 if not specified.
G <- 4

# Optional potential on sites. Automatically set to NULL if not specified
potential <- runif(K,-1,1)
# Optional borders. Automatically set to NULL if not specified
Top <- Bottom <- sample(0:(K-1), width, replace = TRUE)
Left <- Right <- sample(0:(K-1), height, replace = TRUE)
Corner <- sample(0:(K-1), 4, replace = TRUE)

# Exact sampling for the default setting
exact.mrf(h = height, w = width, param = Beta, view = TRUE)
# When specifying the number of colors and neighbors
exact.mrf(h = height, w = width, ncolors = K, nei = G, param = Beta,
view = TRUE)
# When specifying an optional potential on sites
exact.mrf(h = height, w = width, ncolors = K, nei = G, param = Beta,
pot = potential, view = TRUE)
# When specifying possible borders. The users will omit to mention all
# the non-existing borders
exact.mrf(h = height, w = width, ncolors = K, nei = G, param = Beta,
top = Top, left = Left, bottom = Bottom,
right = Right, corner = Corner, view = TRUE)

```



### 3.2 sampler.mrf

The function `sampler.mrf` gives approximate sample from the likelihood (2) of a  $K$ -state Potts model defined on a rectangular  $h \times w$  lattice ( $h \leq w$ ), with either a first order or a second order dependency structure (see Figure 1), using one of the following MCMC samplers.

The Gibbs sampler is a highly popular MCMC algorithm in Bayesian analysis starting with the influential development of (Geman and Geman, 1984). It can be seen as a component-wise Metropolis-Hastings algorithm (Metropolis et al., 1953, Hastings, 1970) where variables are updated one at a time and for which proposal distributions are the full conditionals themselves.

The Swendsen-Wang algorithm (Swendsen and Wang, 1987) originally designed to speed up simulation of Potts model close to the phase transition<sup>3</sup> and bypass slow mixing issues of the Gibbs sampler. This algorithm makes a use of auxiliary variables in order to incorporate simultaneous updates of large homogeneous regions. Note that we solely use Swendsen-Wang updates and we do not run the chain as a *complete coupling* chain as proposed by Huber (1999) to get exact sample from the likelihood.

#### Usage:

```
sampler.mrf(iter, sampler = "Gibbs", h, w, ncolors = 2, nei = 4,
            param, pot = NULL, top = NULL, left = NULL,
            bottom = NULL, right = NULL, corner = NULL,
            initialise = TRUE, random = TRUE, view = FALSE)
```

**Inputs:** see also Section 1.2.

<b>iter</b>	Number of iterations of the algorithm.
<b>sampler</b>	The method to be used. The latter must be one of "Gibbs" or "SW" corresponding respectively to the Gibbs sampler and the Swendsen-Wang algorithm.
<b>initialise</b>	Logical value indicating whether initial guess should be randomly drawn.
<b>random</b>	Logical value indicating whether the sites should be updated sequentially or randomly. Used only with the "Gibbs" option.
<b>view</b>	Logical value indicating whether the draw should be printed. Do not display the optional borders.

```
Example: # sampler.mrf
# Algorithm settings
```

<sup>3</sup>Symmetry breaking for large values of parameter  $\beta$  due to a discontinuity of the partition function when the number of sites  $n$  tends to infinity. When the parameter is above the critical value  $\log\{1 + \sqrt{K}\}$  of phase transition, one moves gradually to a multi-modal distribution, that is, values  $x_i$  are almost all equal

```

n <- 200
method <- "Gibbs"

# Dimension of the lattice
height <- width <- 100

# Interaction parameter
Beta <- 0.6 # Isotropic configuration
# Beta <- c(0.6, 0.6) # Anisotropic configuration for a first
# order dependency structure (nei = 4).
# Beta <- c(0.6, 0.6, 0.6, 0.6) # Anisotropic configuration for a second
# order dependency structure (nei = 8).

# Number of colors. Automatically set to 2 if not specified.
K <- 2
# Number of neighbors. Automatically set to 4 if not specified.
G <- 4

# Optional potential on sites. Automatically set to NULL if not specified
potential <- runif(K,-1,1)
# Optional borders. Automatically set to NULL if not specified
Top <- Bottom <- sample(0:(K-1), width, replace = TRUE)
Left <- Right <- sample(0:(K-1), height, replace = TRUE)
Corner <- sample(0:(K-1), 4, replace = TRUE)

# Sampling method for the default setting
sampler.mrf(iter = n, sampler = method, h = height, w = width,
            param = Beta, view = TRUE)
# Sampling using an existing configuration as starting point
sampler.mrf(iter = n, sampler = method, h = height, w = width,
            ncolors = K, nei = G, param = Beta,
            initialise = FALSE, view = TRUE)
# Specifying optional arguments. The users may omit to mention all
# the non-existing borders
sampler.mrf(iter = n, sampler = method, h = height, w = width,
            ncolors = K, nei = G, param = Beta,
            pot = potential, top = Top, left = Left, bottom = Bottom,
            right = Right, corner = Corner, view = TRUE)
# Gibbs sampler with sequential updates of the sites.
sampler.mrf(iter = n, sampler = "Gibbs", h = height, w = width,
            ncolors = K, nei = G, param = Beta,
            random = FALSE, view = TRUE)

```

---

## References

- J. E. Besag. Spatial Interaction and the Statistical Analysis of Lattice Systems (with Discussion). *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236, 1974.
- P. Clifford. Markov random fields in statistics. *Disorder in physical systems: A volume in honour of John M. Hammersley*, pages 19–32, 1990.
- N. Friel and H. Rue. Recursive computing and simulation-free inference for general factorizable models. *Biometrika*, 94(3):661–672, 2007.
- S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.
- G. R. Grimmett. A theorem about random fields. *Bulletin of the London Mathematical Society*, 5(1):81–84, 1973.
- W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- M. Huber. Efficient Exact Sampling from the Ising Model Using Swendsen-wang. In *SODA*, pages 921–922, 1999.
- N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of Chemical Physics*, 21(6):1087–1092, 1953.
- R. Reeves and A. N. Pettitt. Efficient recursions for general factorisable models. *Biometrika*, 91(3):751–757, 2004.
- R. H. Swendsen and J.-S. Wang. Nonuniversal critical dynamics in Monte Carlo simulations. *Physical Review Letters*, 58(2):86–88, 1987.