Defining priors in R; ‘rprior‘

This prior allow the user to use a prior for θ in a form of an R-function which return $\log \pi(\theta)$ as a function of θ. The function needs `not` to vectorize. The prior function needs to be prepared using `inla.rprior.define()` before use (see below for an example).

Example

As an example, we implement the `loggamma`-prior, which for $\theta = \log(\text{precision})$ is

```
rprior.func = function(lprec) {
    return (dgamma(exp(lprec), a, b, log = TRUE) + lprec)
}
```

The last term is the log-Jacobian for the change of variable. Note that the prior-parameters, a and b, can be passed on when *preparing* the prior (a required step), using

```
rprior <- inla.rprior.define(rprior.func, a=1.0, b=0.1)
```

`rprior` can then be used as the argument `prior = rprior` in the `hyper` argument; see the following example.

```
rprior.func = function(lprec) {
    return (dgamma(exp(lprec), a, b, log = TRUE) + lprec)
}
rprior <- inla.rprior.define(rprior.func, a = 1, b = 0.1)

prior.expression = "expression:
            a = 1;
            b = 0.1;
            precision = exp(lprec);
            logdens = log(b^a) - lgamma(a)
                      + (a-1)*lprec - b*precision;
            ljacobian = lprec;
            return(logdens + ljacobian);"

prior.func = function(lprec) {
    a = 1; b = 0.1;
    return (dgamma(exp(lprec), a, b, log = TRUE) + lprec)
}
lprec = seq(-10, 10, len=1000)
prior.table = paste(c("table:", cbind(lprec, prior.func(lprec))),
        sep = "", collapse = " ")

n = 100
y = rnorm(n)

r = inla(y~1,
        data = data.frame(y),
        control.family = list(
                hyper = list(
                        prec = list(
                                prior = "loggamma",
```

```
                                param = c(1, 0.1)))))

rr = inla(y~1,
        data = data.frame(y),
        control.family = list(
                hyper = list(
                        prec = list(
                                prior = prior.expression))))

rrr = inla(y~1,
        data = data.frame(y),
        control.family = list(
                hyper = list(
                        prec = list(
                                prior = prior.table))))

rrrr = inla(y~1,
        data = data.frame(y),
        control.family = list(
                hyper = list(
                        prec = list(
                                prior = rprior))))

round(c(r$mlik[1], rr$mlik[1], rrr$mlik[1], rrrr$mlik[1]), 5)
```