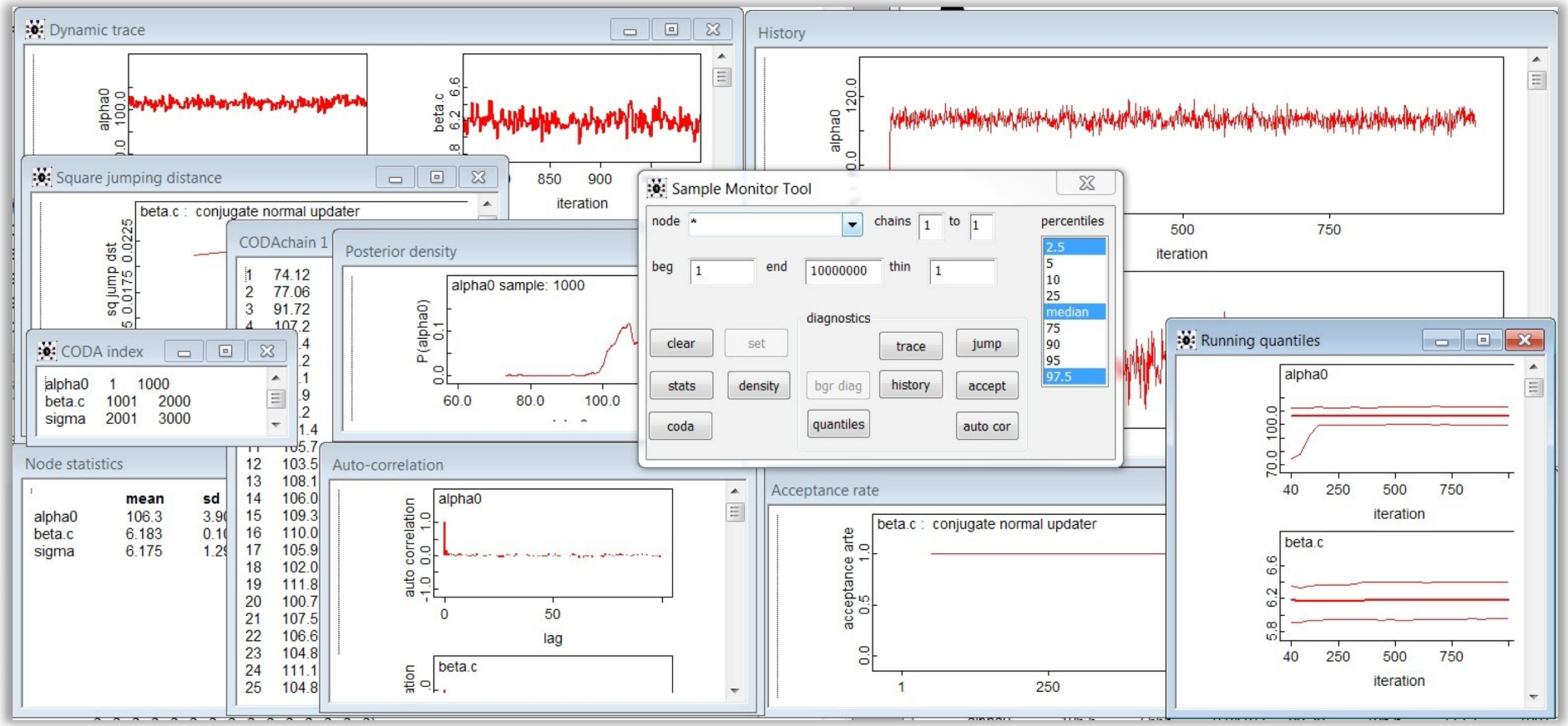
greta

simple and scalable statistical modelling in R

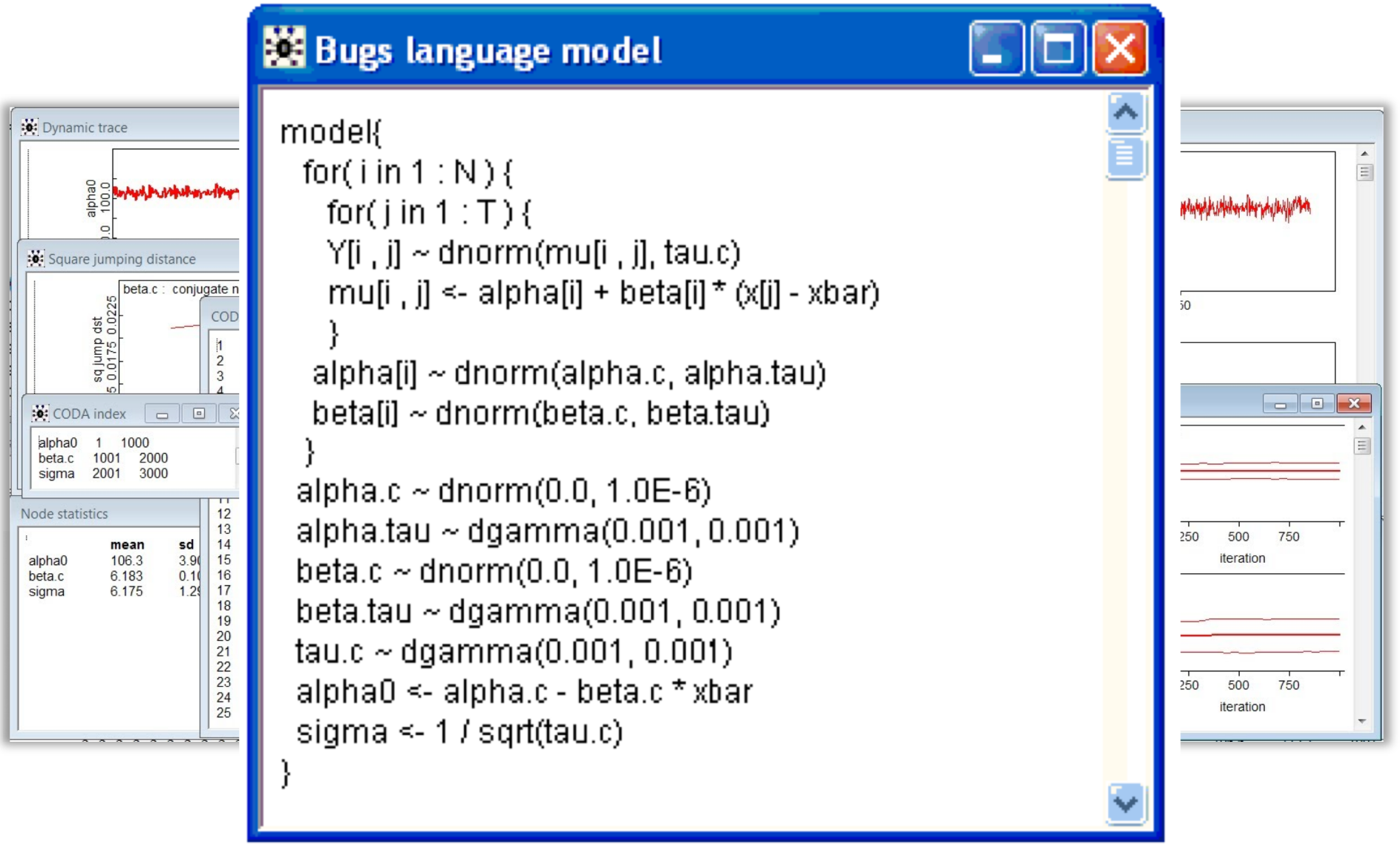
Nick Golding

greta-dev.github.io/greta

BUGS


Dynamic trace
alpha0
beta.c
iteration
History
alpha0
iteration
Square jumping distance
beta.c : conjugate normal updater
sq.jump dst
CODAchain 1
Posterior density
alpha0 sample: 1000
P(alpha0)
Sample Monitor Tool
node
chains
to
percentiles
beg
end
thin
diagnostics
clear
set
trace
jump
stats
density
bgr diag
history
accept
coda
quantiles
auto cor
2.5
5
10
25
median
75
90
95
97.5
CODA index
alpha0 1 1000
beta.c 1001 2000
sigma 2001 3000
Running quantiles
alpha0
beta.c
iteration
Node statistics
mean
sd
alpha0 106.3
beta.c 6.183
sigma 6.175
Auto-correlation
auto correlation
lag
Acceptance rate
beta.c : conjugate normal updater
acceptance arte

BUGS


Bugs language model
model{
for(i in 1 : N) {
for(j in 1 : T) {
Y[i , j] ~ dnorm(mu[i , j], tau.c)
mu[i , j] <- alpha[i] + beta[i] * (x[j] - xbar)
}
alpha[i] ~ dnorm(alpha.c, alpha.tau)
beta[i] ~ dnorm(beta.c, beta.tau)
}
alpha.c ~ dnorm(0.0, 1.0E-6)
alpha.tau ~ dgamma(0.001, 0.001)
beta.c ~ dnorm(0.0, 1.0E-6)
beta.tau ~ dgamma(0.001, 0.001)
tau.c ~ dgamma(0.001, 0.001)
alpha0 <- alpha.c - beta.c * xbar
sigma <- 1 / sqrt(tau.c)
}
Dynamic trace
Square jumping distance
CODA index
alpha0 1 1000
beta.c 1001 2000
sigma 2001 3000
Node statistics
mean sd
alpha0 106.3
beta.c 6.183
sigma 6.175
iteration
iteration

Stan

Stan

```
data {
  int<lower=0> N;   // number of data items
  int<lower=0> K;   // number of predictors
  matrix[N, K] x;   // predictor matrix
  vector[N] y;      // outcome vector
}
transformed data {
  matrix[N, K] Q_ast;
  matrix[K, K] R_ast;
  matrix[K, K] R_ast_inverse;
  // thin and scale the QR decomposition
  Q_ast = qr_Q(x)[, 1:K] * sqrt(N - 1);
  R_ast = qr_R(x)[1:K, ] / sqrt(N - 1);
  R_ast_inverse = inverse(R_ast);
}
parameters {
  real alpha;           // intercept
  vector[K] theta;      // coefficients on Q_ast
  real<lower=0> sigma;  // error scale
}
model {
  y ~ normal(Q_ast * theta + alpha, sigma);  // likelihood
}
generated quantities {
  vector[K] beta;
  beta = R_ast_inverse * theta; // coefficients on x
}
```

Stan

```
data {
  real alpha;
  real beta;
  real<lower=0> sigma2;
  int<lower=0> J;
  int y[J];
  vector[J] Z;
  int n[J];
}

transformed data {
  real<lower=0> sigma;
  sigma <- sqrt(sigma2);
}

parameters {
   real theta1;
   real theta2;
   vector[J] X;
}

model {
  real p[J];
  theta1 ~ normal(0, 32);    // 32^2 = 1024
  theta2 ~ normal(0, 32);
  X ~ normal(alpha + beta * Z, sigma);
  y ~ binomial_logit(n, theta1 + theta2 * X);
}
```

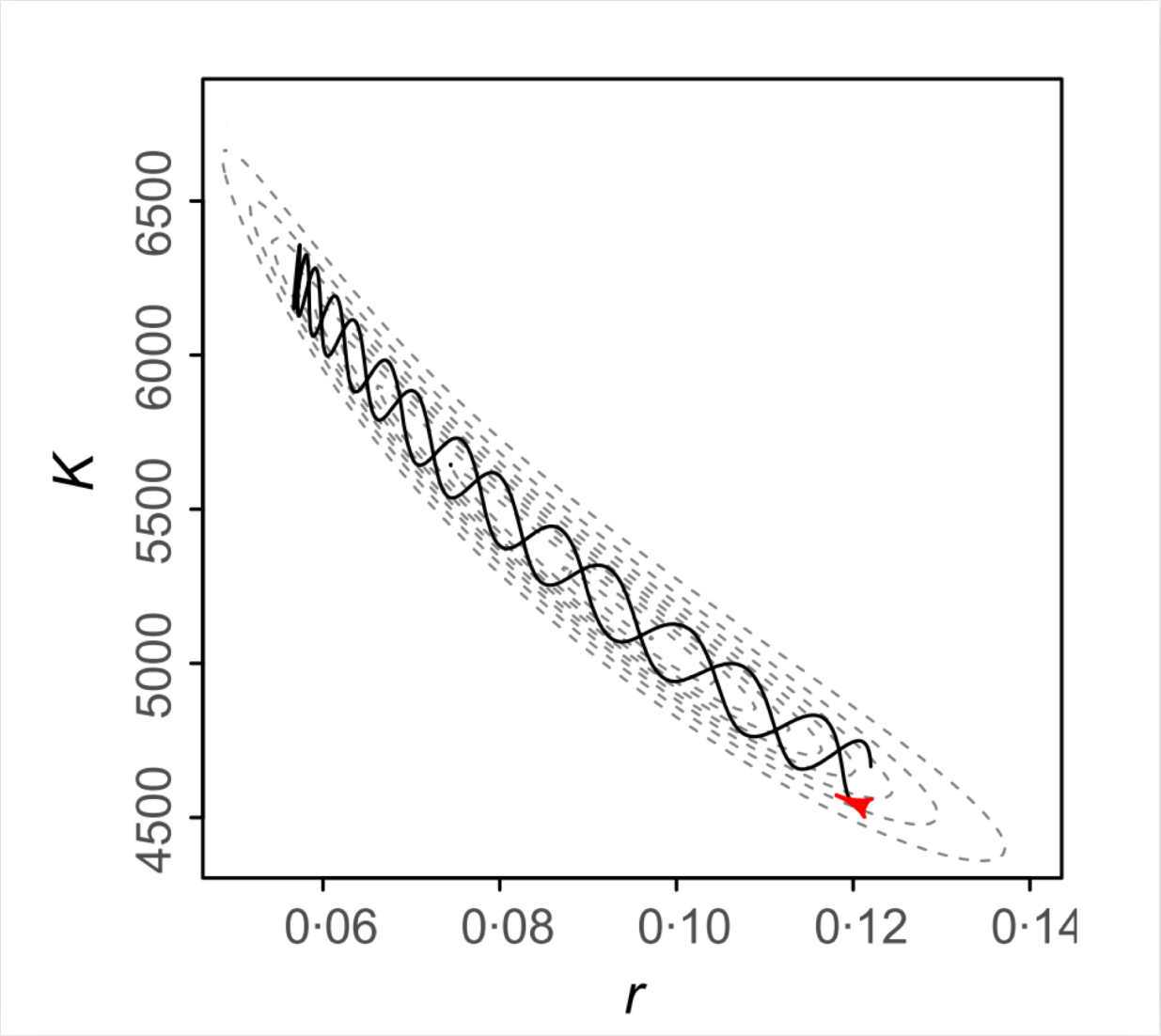
JAGS

```
for(j in 1 : J) {
   y[j] ~ dbin(p[j], n[j])
   logit(p[j]) <- theta[1] + theta[2] * X[j]
   X[j] ~ dnorm(mu[j], tau)
   mu[j] <- alpha + beta * Z[j]
}
theta[1] ~ dnorm(0.0, 0.001)
theta[2] ~ dnorm(0.0, 0.001)
```

greta

```
theta = normal(0, 32, dim = 2)
mu <- alpha + beta * Z
X = normal(mu, sigma)
p <- ilogit(theta[1] + theta[2] * X)
distribution(y) = binomial(n, p)
```

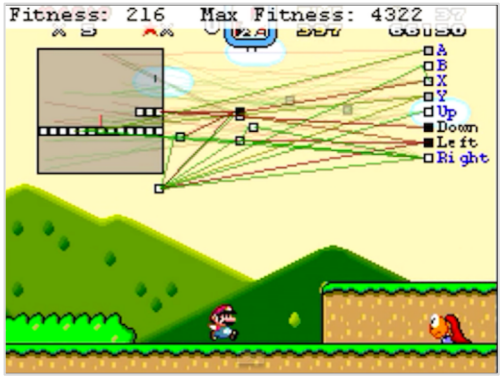
gradient-based inference



Monnahan *et al.* 2016

google tensorflow

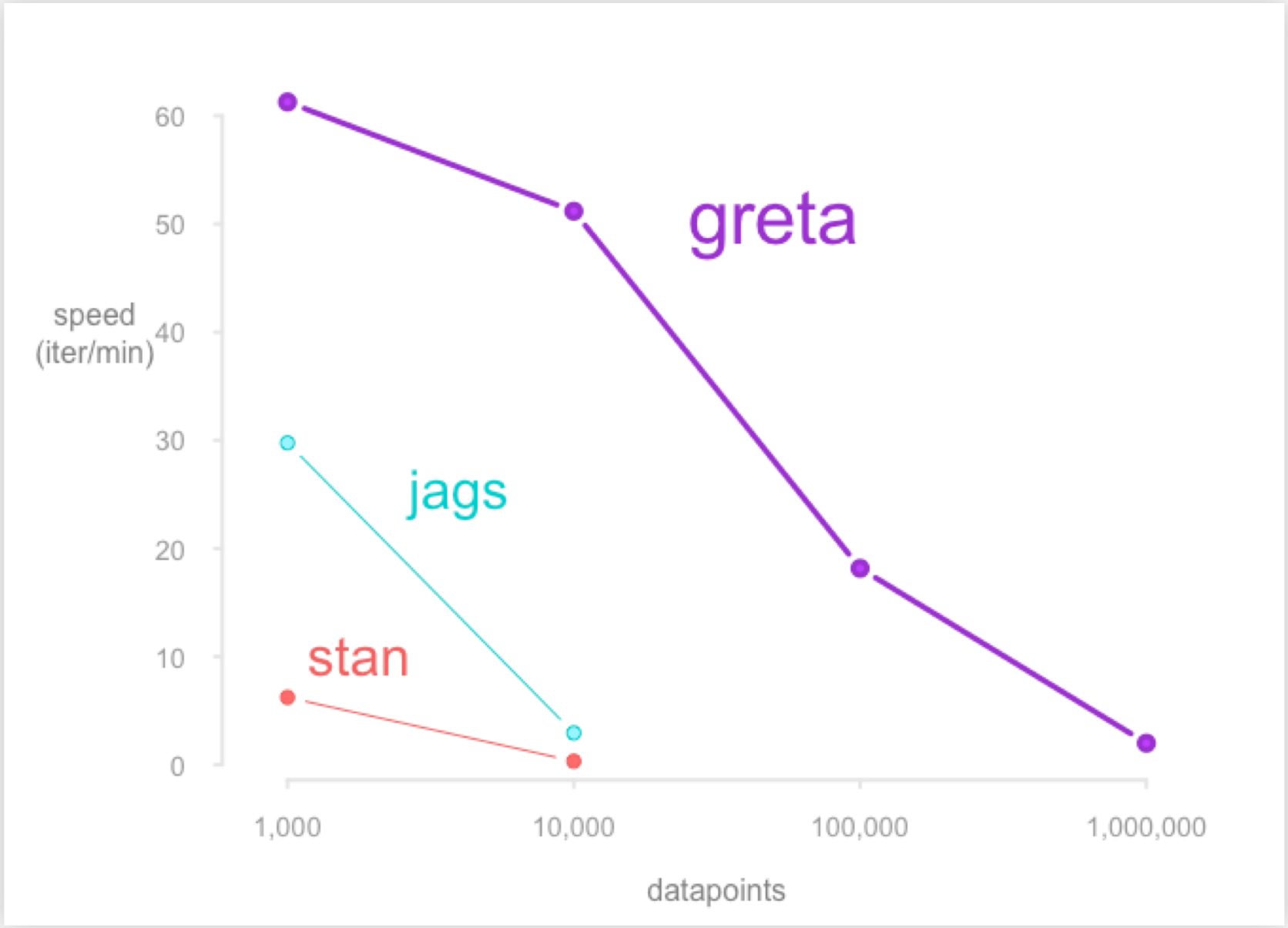
- automatic differentiation
- efficient linear algebra
- highly parallel



Google Translate

scalable

probit regression with 50 predictors



extendable

greta.dynamics

a greta extension for modelling dynamical systems

R ★ 1 2 Updated 12 days ago

greta.gp

a greta extension for Gaussian process modelling

R ★ 4 1 Updated on 9 Sep

greta.multivariate

a greta extension for multivariate modelling

1 Updated on 26 Aug

greta.gam

a greta extension for generalised additive modelling using mgcv

R ★ 3 2 Updated on 23 Aug

extendable

greta.gp

Gaussian processes in greta

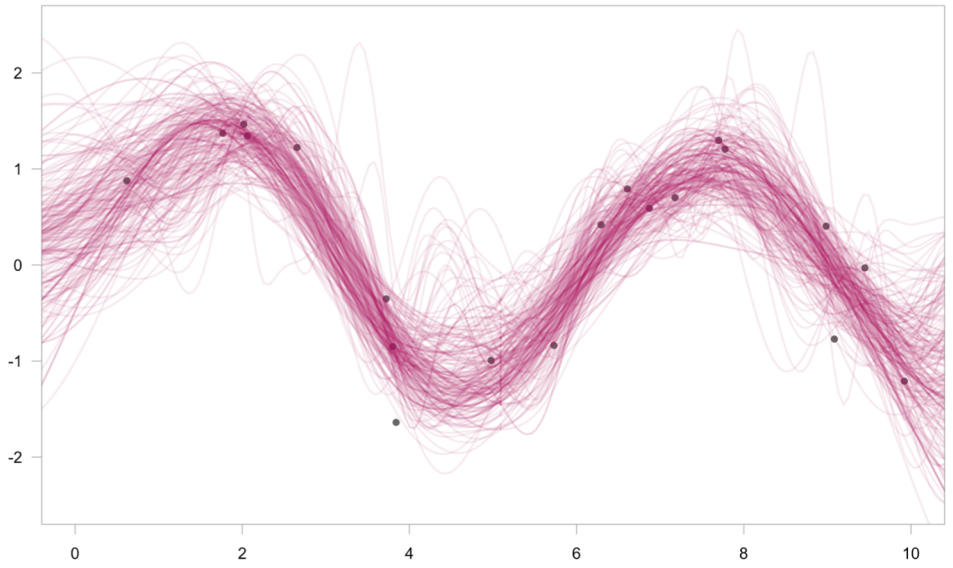
`greta.gp` extends greta to let you define Gaussian processes as part of your model. It provides a syntax to create and combine GP kernels, and use them to define either full rank or sparse Gaussian processes.

build passing codecov 98%

```
# kernel & GP
kernel <- rbf(rbf_len, rbf_var) + bias(1)
f = gp(x, kernel)

# likelihood
distribution(y) = normal(f, obs_sd)

# prediction
f_plot <- project(f, x_plot)
```



github.com/greta-dev/greta.gp

what next?

- variational inference
- samplers for big data
- discrete samplers
- differential equations

why ‘greta’?

Grete Hermann (1901-1984)

wrote the first algorithms for computer algebra

… without a computer

(I didn’t want people saying ‘greet’, so spelled the package *greta* instead)