

# Bayesian modeling of effective and functional brain connectivity - an exercise in probabilistic programming

Mattias Villani

Department of Statistics  
Stockholm University



[mattiasvillani.com](http://mattiasvillani.com)



@matvil



@matvil



[mattiasvillani](https://github.com/mattiasvillani)

# Overview

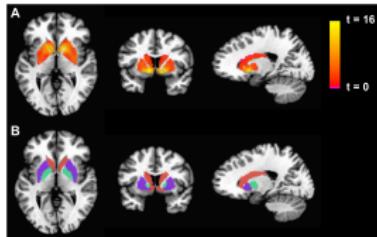
- fMRI data
  - Brain connectivity
  - Hierarchical vector autoregressions
  - Probabilistic programming
  - Variational inference for large-scale problems
- 
- Slides: <http://mattiasvillani.com/news>.

## Article and collaborators

- Based on recently accepted paper: 'Bayesian Modeling of Effective and Functional Brain Connectivity using Hierarchical Vector Autoregressions', *Journal of the Royal Statistical Society, Series C*. [journal online version](#)
- **Bertil Wegmann**, Statistics, Linköping University
- **Anders Lundquist**, Statistics, Umeå University
- **Anders Eklund**, Medical Informatics, Linköping University

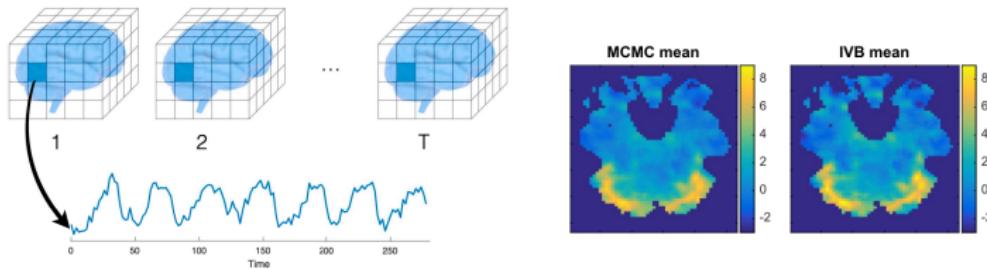
# fMRI for measuring brain activity

- Functional magnetic resonance imaging.
- Non-invasive indirect measure of brain activity.
- Locates brain activity by detecting changes in blood flow.
- Oxygenization is measured using its magnetic properties.
- Data problems:
  - ▶ Low signal buried in noise
  - ▶ Blood flows are sluggish
  - ▶ Head movements, breathing and other artefacts



# fMRI data and activity analysis

- fMRI data are recorded in MANY brain voxels (3D pixels).
- A **time series** in each voxel. An observation every 0.5-3 sec.



- **High-dim multivariate time series** with **spatial** dependence.
- **Voxel-wise time series regression** for voxel  $v$

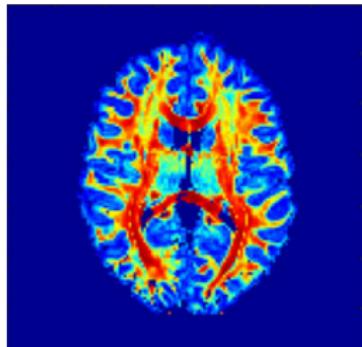
$$y_{vt} = \mathbf{x}_{vt}^\top \boldsymbol{\beta}_v + \varepsilon_{vt}, \quad \varepsilon_{vt} \sim ARIMA$$

where  $\mathbf{x}_{vt}$  contains covariates from experimental condition.

- **2D brain slices** with spatial dependence between  $\boldsymbol{\beta}_v$  (GMRF).
- **Whole-brain 3D** with spatial dependence.

# Resting state fMRI and brain connectivity

- **Connectivity**: how brain regions “communicate/cooperate”.
- **Functional connectivity**: correlations between brain regions.
- **Effective connectivity**: “causal” lead-lag cross-correlations.
- **Resting-state fMRI**: fMRI measurements for subjects at rest.
- Are there differences in brain connectivity between autism spectrum disorder (ASD) patients and healthy controls?



# Hierarchical VAR for multi-subject analysis

- Subjects:  $s = 1, \dots, S$
- Brain regions:  $r = 1, \dots, R$
- Time points:  $t = 1, \dots, T$
- Data on subject  $s$  at time  $t$  over all  $R$  regions:  $\mathbf{y}_{s,t}$  ( $R$ -dim)
- VAR( $K$ ) model for each subject  $s$

$$\mathbf{y}_{s,t} = \sum_{k=1}^K \mathbf{B}_{s,k} \mathbf{y}_{s,t-k} + \boldsymbol{\varepsilon}_{s,t}, \quad \boldsymbol{\varepsilon}_{s,t} \stackrel{\text{iid}}{\sim} N(\mathbf{0}, \boldsymbol{\Sigma}_s)$$

$$\mathbf{B}_s = \left( \begin{array}{cccc} \mathbf{B}_{s,1} & \mathbf{B}_{s,2} & \cdots & \mathbf{B}_{s,K} \end{array} \right)$$

# Hierarchical prior to borrow strength across subjects

## ■ Prior on subject level parameters

- ▶  $\boldsymbol{B}_s$  centered around a group matrix  $\boldsymbol{B}$ .
- ▶  $\boldsymbol{\Sigma}_s$  centered around a group covariance matrix  $\boldsymbol{\Sigma}$

$$\begin{aligned}\text{vec}(\boldsymbol{B}_s) \mid \boldsymbol{\Sigma}_s &\sim N\left(\text{vec}(\boldsymbol{B}), \boldsymbol{\Sigma}_s \otimes \boldsymbol{P}_s^{-1}\right) \\ \boldsymbol{\Sigma}_s &\sim \text{IW}(\boldsymbol{\Sigma}, \nu)\end{aligned}$$

## ■ Prior on the group level parameters

$$\begin{aligned}\text{vec}(\boldsymbol{B}) \mid \boldsymbol{\Sigma}_s &\sim N\left(\text{vec}(\boldsymbol{B}_0), \boldsymbol{\Sigma} \otimes \boldsymbol{P}_0^{-1}\right) \\ \boldsymbol{\Sigma} &\sim \text{IW}(\boldsymbol{\Psi}_0, \nu_0)\end{aligned}$$

## ■ Main interest: group level posterior given the data

$$p(\boldsymbol{B}, \boldsymbol{\Sigma} \mid \mathbf{y}_{1:S, 1:T})$$

- Marginalize out all  $\boldsymbol{B}_s$  and  $\boldsymbol{\Sigma}_s$  to get  $p(\boldsymbol{B}, \boldsymbol{\Sigma} \mid \mathbf{y}_{1:S, 1:T})$ . 😍
- $p(\boldsymbol{B}, \boldsymbol{\Sigma} \mid \mathbf{y}_{1:S, 1:T})$  is not a known distribution. 😳

Stan and probabilistic programming to the rescue! 



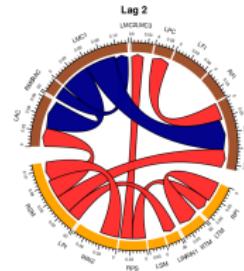
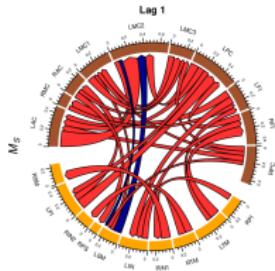
## Results from the ABIDE

- ABIDE data: resting state fMRI data
  - ▶ 539 individuals with Autism Spectrum Disorder (ASD)
  - ▶ 573 healthy controls.
- Randomly select 20 controls and 20 ASD from NY site.
- $T = 180$  time points
- $R = 20$  regions in
  - ▶ Default-Mode Network (DMN)
  - ▶ Sensory-Motor Network (SMN).

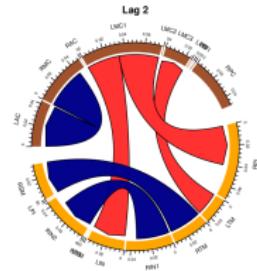
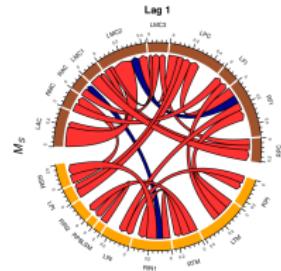


# Effective connectivity - *B* coefficients

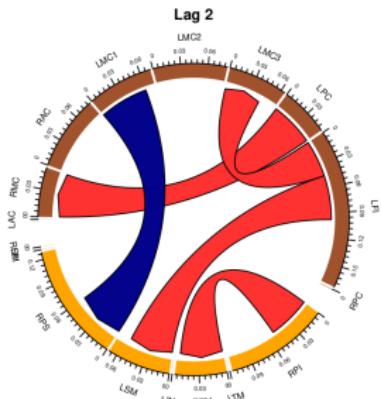
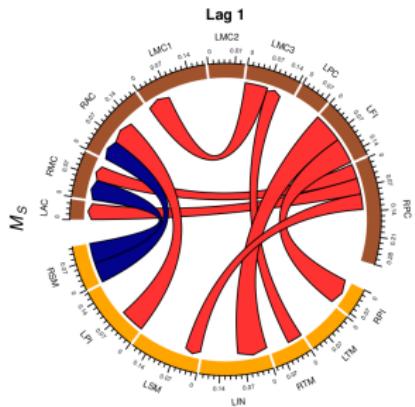
Controls



Autism

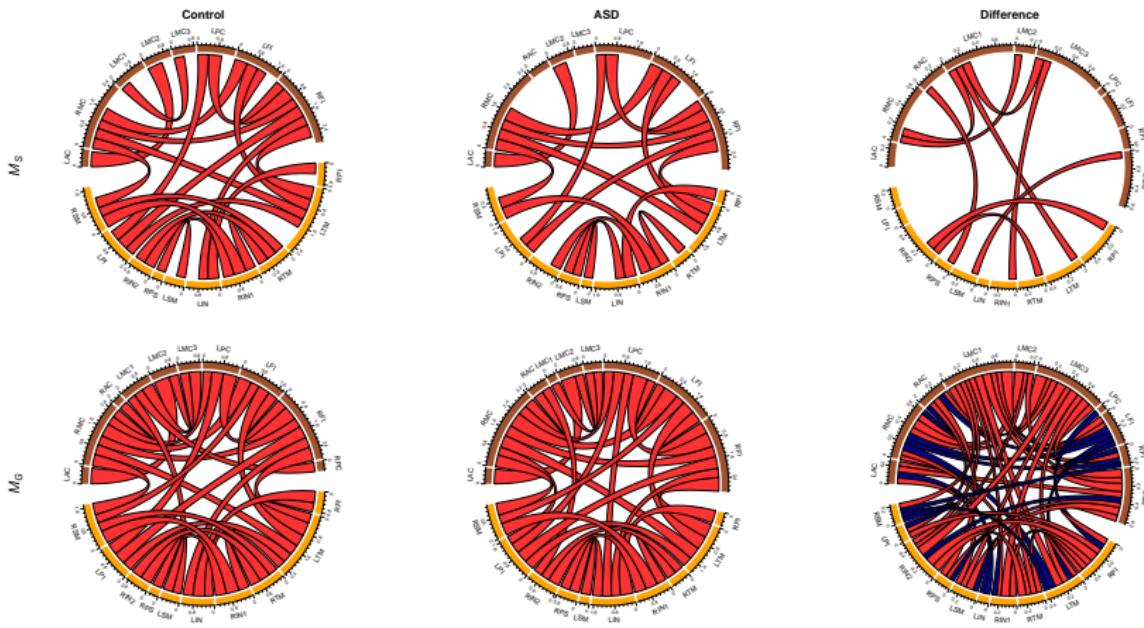


Controls - Autism contrast



# Functional connectivity - $\Sigma$

- Model  $M_S$ : different  $\Sigma_s$  across subjects around group  $\Sigma$
- Model  $M_G$ : all subjects have the common  $\Sigma$



# Probabilistic programming languages for Bayes

- Bayesian inference is usually performed by exploring the posterior distribution by **simulation**. **MCMC** etc.

**Probabilistic programming** (PP) is a programming paradigm in which probabilistic models are specified and inference for these models is performed automatically.

- Write up the model, no need to code up sampling algorithms.
- **Automatic differentiation**: computer computes derivatives.
- **Stan** - language built upon **Hamiltonian Monte Carlo**.
- **HMC**. Gradient-based MCMC for high-dim.
- C++ using the R package `rstan`. Bindings from Python.
- **Turing.jl** is a PP language in Julia. Fast natively. 😍

# Poisson regression in Turing.jl

## ■ Poisson regression

$$y_i | \mathbf{x}_i \sim \text{Poisson}(\lambda_i)$$

$$\lambda_i = \exp(\mathbf{x}_i^\top \boldsymbol{\beta})$$

$$\boldsymbol{\beta} \sim N(\mathbf{0}, \tau^2 \mathbf{I}_p)$$

```
using Turing

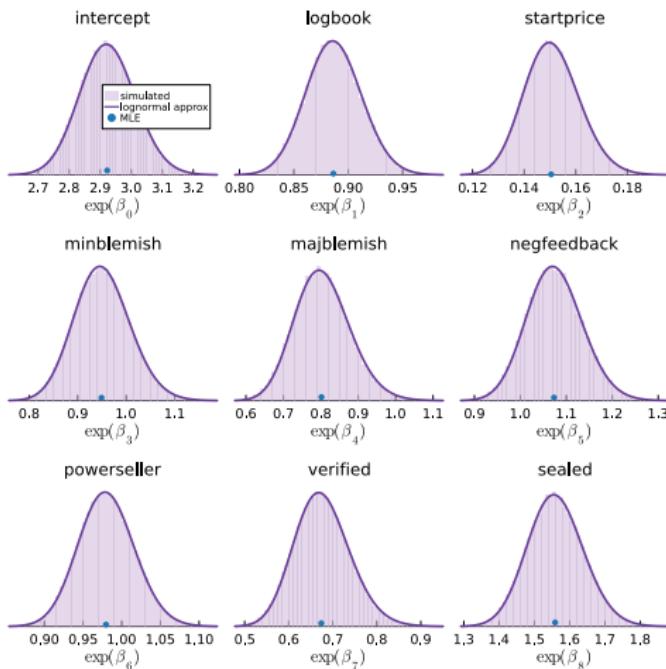
# Setting up the poisson regression model
@model function poissonReg(y, X, τ)
    p = size(X,2)
    β ~ filldist(Normal(0, τ), p) # all βj are iid Normal(0, τ)
    λ = exp.(X*β)
    n = length(y)
    for i in 1:n
        | y[i] ~ Poisson(λ[i])
    end
end

# HMC sampling from posterior of β
τ = 10      # Prior standard deviation
α = 0.70    # target acceptance probability in NUTS sampler
model = poissonReg(y, X, τ)
chain = sample(model, Turing.NUTS(α), 10000, discard_initial = 1000)
```

# Poisson regression in Stan

```
data {  
    int<lower=0> n;      // number of observations  
    int<lower=0> p;      // number of predictors  
    matrix[n, p] X;      // predictor matrix  
    vector[n] y;         // outcome vector  
    real<lower=0> tau;    // prior standard deviation  
}  
parameters {  
    vector[p] beta;  
}  
model {  
    // Prior  
    beta ~ normal(0,tau^2);  
  
    // Model/likelihood  
    y ~ poisson_log(X*beta);  
}  
  
fit1 <- stan(model_code = poissonRegModel,  
             data = poisRegData,  
             warmup = 1000,  
             iter = 10000  
           )
```

# HMC sampling



## TuringGLM.jl with R's formula syntax

```
# Using TuringGLM.jl
using TuringGLM
fm = @formula(nbids ~ logbook + startprice + minblemish +
| majblemish + negfeedback + powerseller + verified + sealed)
model = turing_model(fm, ebay_df; model = Poisson)
chain = sample(model, NUTS(), 10000)
```

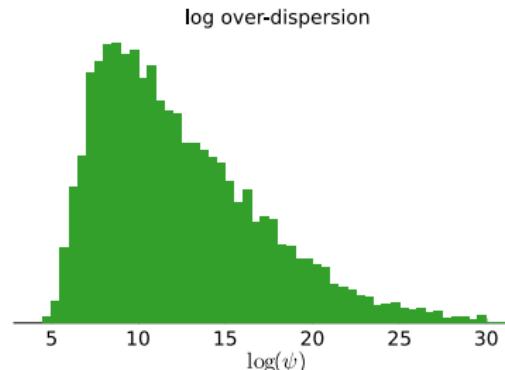
- Inspired by the [brms](#) package in R.

# Negative binomial regression in Turing.jl

## ■ Negative binomial regression

$$y_i | \mathbf{x}_i \sim \text{NegBinomial} \left( \psi, p = \frac{\psi}{\psi + \lambda_i} \right), \quad \lambda_i = \exp(\mathbf{x}_i^\top \boldsymbol{\beta})$$

```
# Negative binomial regression
@model function negbinomialReg(y, X, τ, μ₀, σ₀)
    p = size(X,2)
    β ~ filldist(Normal(0, τ), p)
    λ = exp.(X*β)
    ψ ~ LogNormal(μ₀, σ₀)
    n = length(y)
    for i in 1:n
        | y[i] ~ NegativeBinomial(ψ, ψ/(ψ + λ[i]))
    end
end
```



# Stan code for the marginalized model

```
data {
    int<lower=0> R; // number of brain regions
    int<lower=0> LR; // number of covariates L+R
    int<lower=0> S; // number of subjects
    int<lower=0> LRR; // number of VAR coefficients, LRR = LR+R
    int<lower=0> T; // number of time points
    array[S] matrix [R,R] M_s; // array with matrices M_s for all subjects
    array[S] matrix [LR,LR] E_s; // array with matrices E_s for all subjects
    array[S] matrix [LR,LR] Q_s_inv; // array with matrices Q_s_inv for all subjects
    // Prior settings
    vector[LRR] B_0_spec; // prior Mean
    cov_matrix[LR] Chol_Cov_B; // cholesky decomposition of the covariance matrix for B
    int nu_0; // degrees of freedom in the prior for Sigma
    cov_matrix[R] Psi_0; // scale matrix in the prior for Sigma
    cov_matrix[LRR] I_Mat; // identity matrix
}
parameters (
    cov_matrix[R] Sigma; // covariance matrix
    matrix[LR,LR] B_spec; // matrix of VAR coefficients
    real<lower=R+2> nu; // degrees of freedom in the prior for Sigma_s
)
transformed parameters {
    matrix[LR,R] B; // matrix of VAR coefficients
    B = Chol_Cov_B * B_spec * cholesky_decompose(Sigma);
}
model {
    real Sum_logdet;
    matrix[R,R] Part_s;
    // priors
    Sigma ~ inv_wishart(nu_0,Psi_0); // prior for the covariance matrix Sigma
    to_vector(B_spec) ~ multi_normal(B_0_spec,I_Mat); // special prior for parameterization
    Sum_logdet = 0;
    // log-likelihood
    for (s in 1:S){
        Part_s = nu*Sigma + M_s[s] + quad_form(Q_s_inv[s] , B-E_s[s]);
        Sum_logdet = Sum_logdet + log_determinant(Part_s);
    }
    target += S*( lmgamma(R,0.5*(T+nu)) - lmgamma(R,0.5*nu) );
    target += 0.5*S*nu*log_determinant(nu*Sigma) - 0.5*(nu+T)*Sum_logdet;
}
```

# Stan code for the marginalized model

$$p(\mathbf{y}_{1:S,1:T} | \mathbf{B}, \Sigma) = c |\nu \Sigma|^{\nu/2} \prod_{s=1}^S \left| \nu \Sigma + M_s + (\mathbf{B} - \mathbf{E}_s)^T Q_s^{-1} (\mathbf{B} - \mathbf{E}_s) \right|$$

```
model {
    real Sum_logdet;
    matrix[R,R] Part_s;
    // priors
    Sigma ~ inv_wishart(nu_0,Psi_0); // prior for the covariance matrix Sigma
    to_vector(B_spec) ~ multi_normal(B_0_spec,I_Mat); // special prior for parameterization
    Sum_logdet = 0;
    // log-likelihood
    for (s in 1:S){
        Part_s = nu*Sigma + M_s[s] + quad_form(Q_s_inv[s] , B-E_s[s]);
        Sum_logdet = Sum_logdet + log_determinant(Part_s);
    }
    target += S*( lmgamma(R,0.5*(T+nu)) - lmgamma(R,0.5*nu) );
    target += 0.5*S*nu*log_determinant(nu*Sigma) - 0.5* (nu+T)*Sum_logdet;
}
```

# Variational Inference

- Approx the posterior  $p(\theta|x)$  with a (simpler) distribution  $q(\theta)$ .
- Mean field Variational Inference (VI):

$$q(\theta) = \prod_{i=1}^p q_i(\theta_i)$$

- Parametric VI: Parametric family  $q_\lambda(\theta)$  with parameters  $\lambda$ .
- Find  $q(\theta)$  that minimizes the Kullback-Leibler divergence

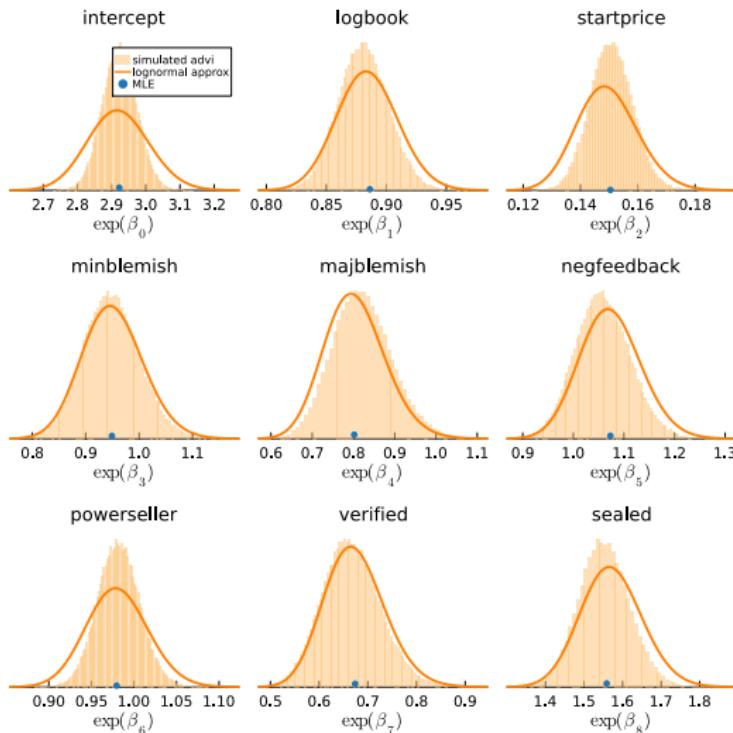
$$KL(q, p) = \int \ln \frac{q(\theta)}{p(\theta|x)} q(\theta) d\theta.$$

- Enough with proportional form  $p(\theta|x) \propto p(x|\theta)p(\theta)$ .

## Variational inference - Poisson regression in Turing.jl

```
# Variational inference for posterior of β
τ = 10      # Prior standard deviation
model = poissonReg(y, X, τ)
nSamples = 10
nGradSteps = 1000
approx_post = vi(model, ADVI(nSamples, nGradSteps))
βsample = rand(approx_post, 1000)
```

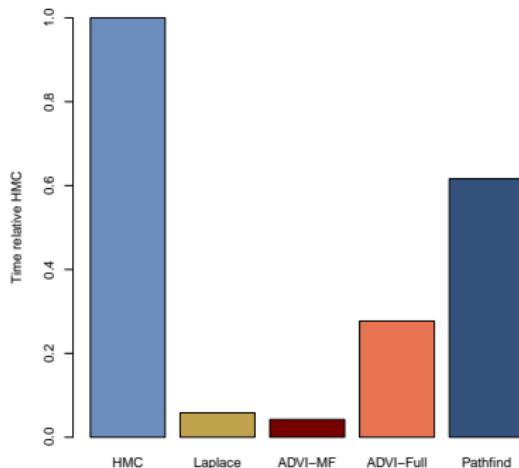
# Variational inference - Poisson regression in Turing.jl



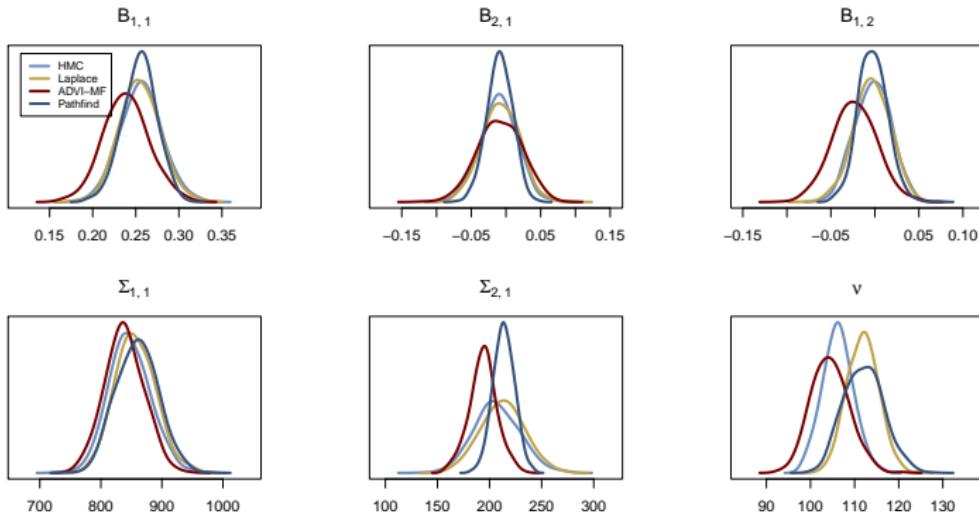
# Variational inference for hierarchical VAR - Time

HMC compute times

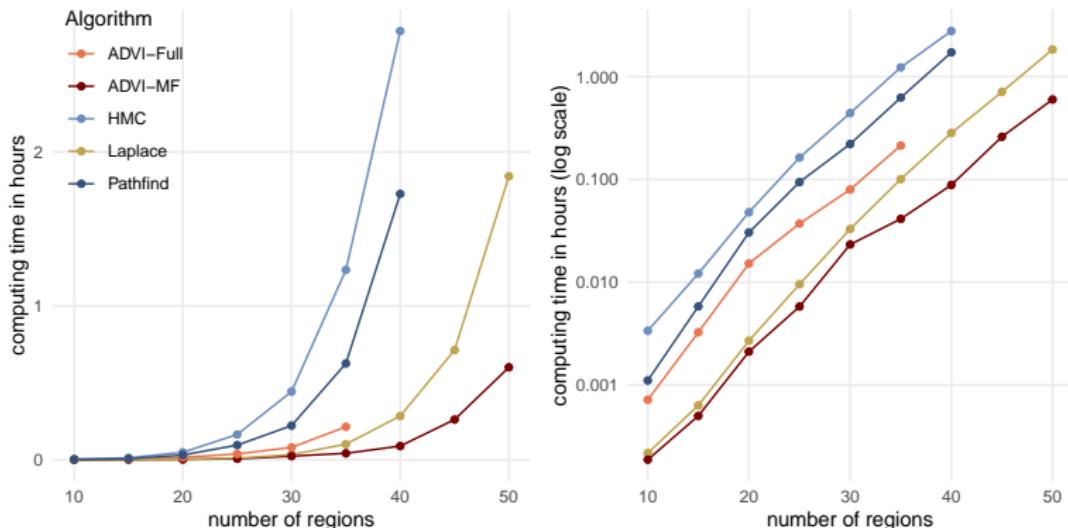
	$R = 10$	$R = 20$	$R = 30$
$L = 1$	0.005	0.094	0.479
$L = 2$	0.008	0.385	9.242
$L = 3$	0.018	1.313	48.805
$L = 4$	0.048	5.288	N.A.



# Variational inference for hierarchical VAR - Accuracy



# Variational inference for hierarchical VAR - Scaling



# Conclusions

- A Bayesian hierarchical vector autoregression motivated by group comparison of brain connectivity.
- The model is applied to 20 healthy controls and 20 Autism patients from the ABIDE dataset.
- Computations is performed by
  - ▶ analytical marginalization of all subject level parameters
  - ▶ HMC sampling in Stan of the group level parameters.
  - ▶ variational inference to handle many brain regions/lags.
- Future work:
  - ▶ further analysis and development of variational inference.
  - ▶ using the model to classify subjects as healthy/autism.