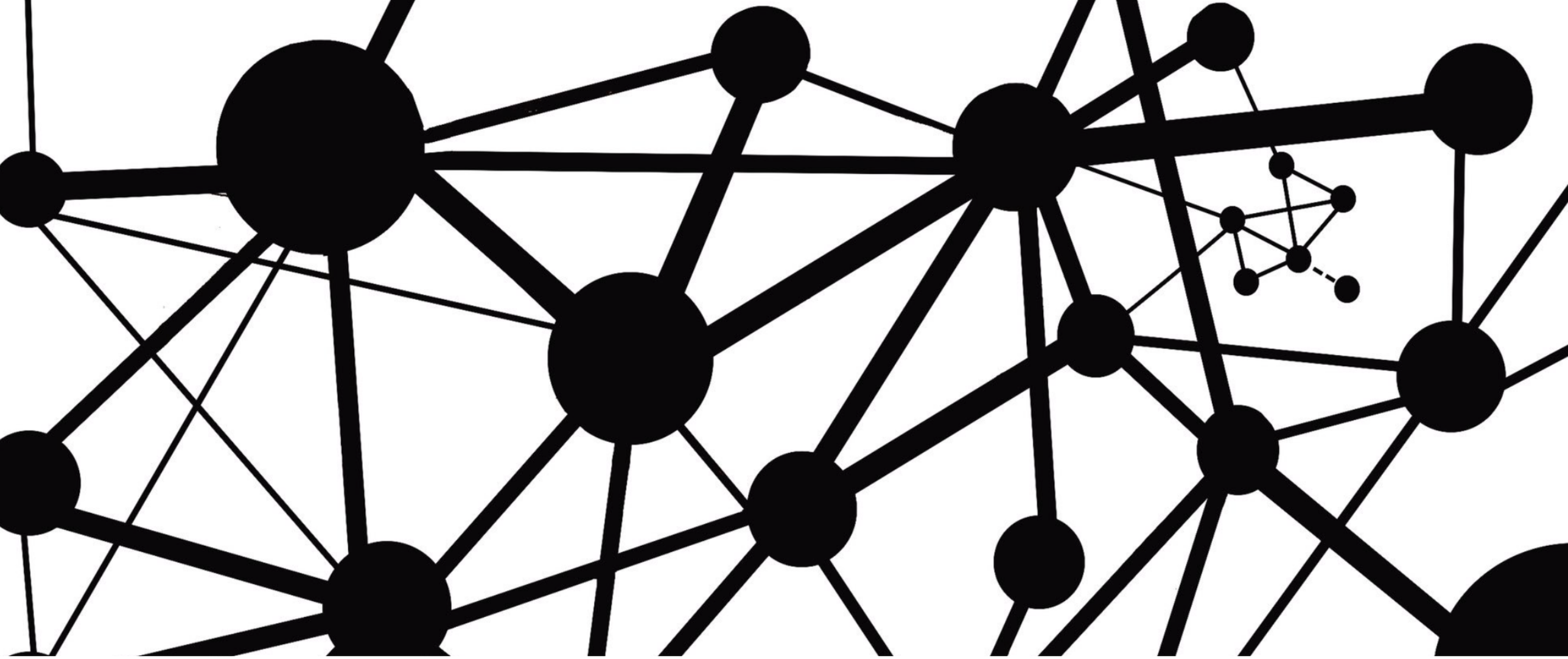


Final Considerations: Calibration, Computation, and Next Steps

Network Models for HIV/STI Transmission Dynamics with EpiModel

June 30, 2017

Overview

- Computational issues: tergmLite, HPCs
- Model calibration and sources of uncertainty in network models
- Steps of executing an EpiModel-based modeling project (on to Github…)

tergmLite

An **R package** (on Github but not CRAN, more on that later)

Handles the **simulation-side** of our epidemic modeling workflow

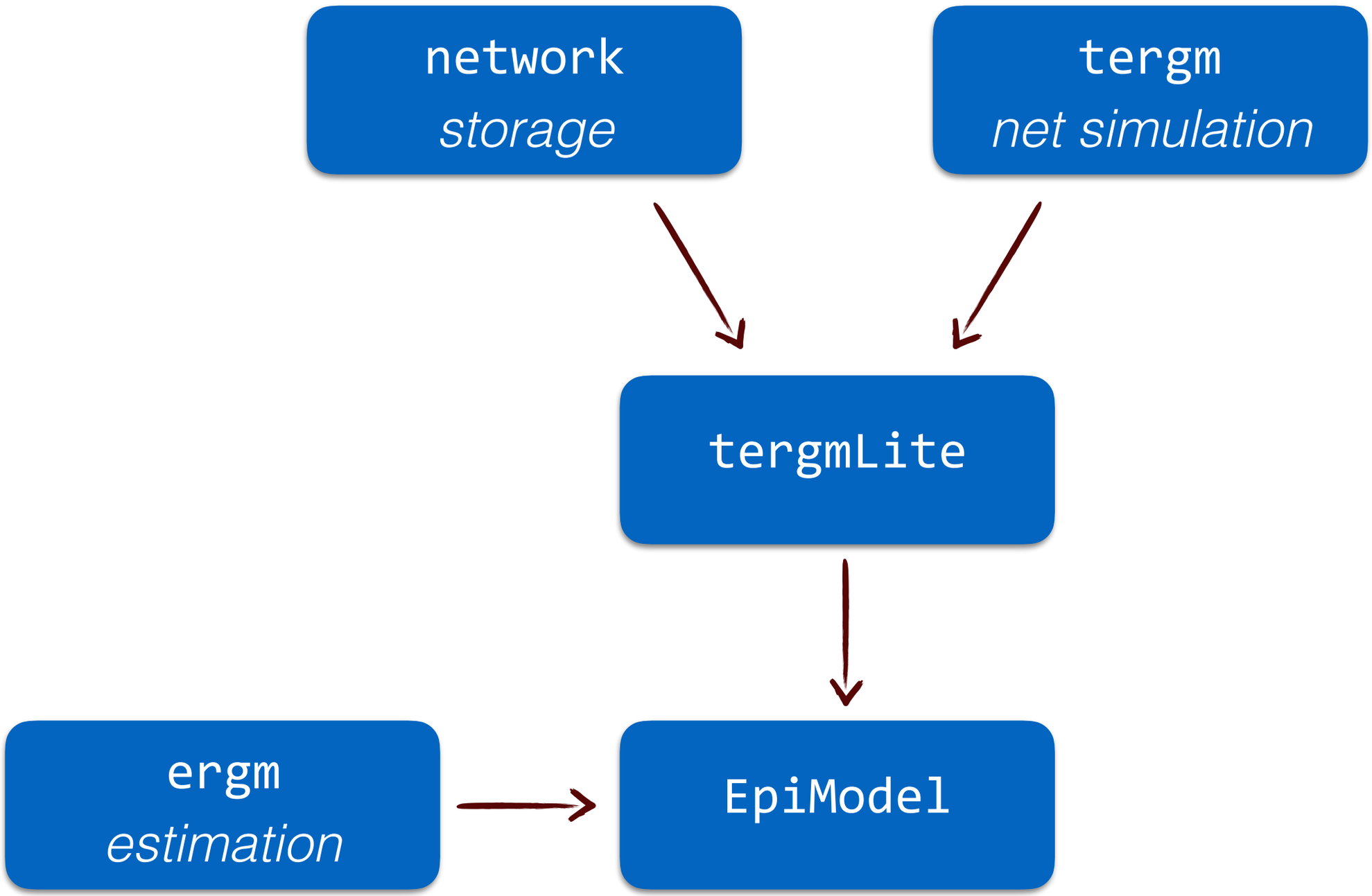
Represents the network structure as an edgelist and separate list of vectors for nodal attributes, along with ergm-defined metadata

Updates this network structure given vital dynamics and nodal covariate change

Updates the metadata needed by tergm/ergm at each time step given changing covariates and network composition

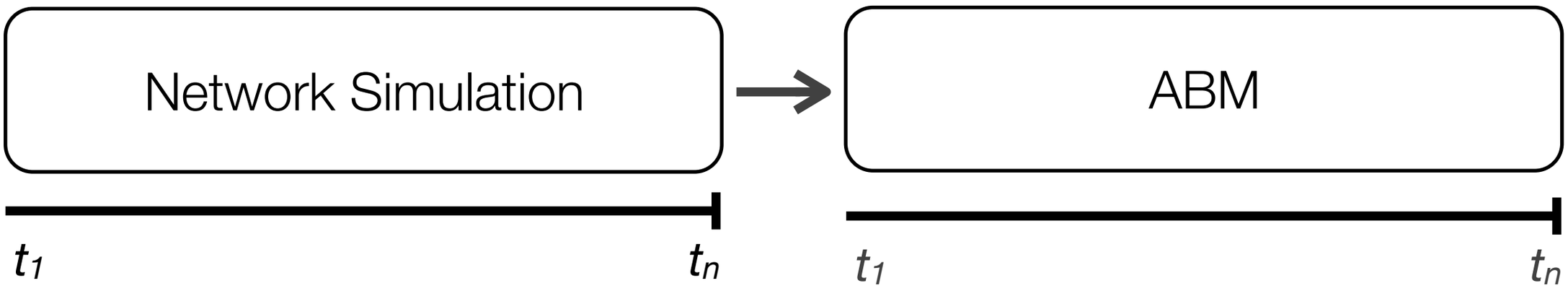
This represents a tradeoff of features (loss of network data, fewer supported network models) for speed

tergmLite

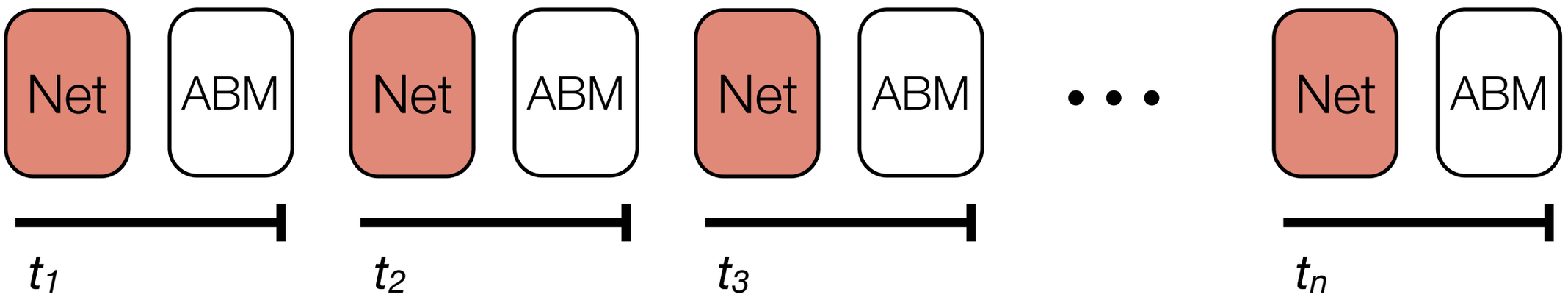

network
storage
tergm
net simulation
tergmLite
ergm
estimation
EpiModel

Model Dependence = Computational Bottleneck

Independent Models



Dependent Models



Implications

Time

~0.0002 seconds per node per time step

10k network, 3 partnership types, 100 years of simulation = 14 hours

Costs

Money: Grant investment in HPCs

Time: efforts devoted towards parallelization

Methods: limits new statistical applications

Applied research: takes too long

Optimization Work outside tergmLite

In our **epidemic** modeling of networks, we never/rarely use:

- Individual nodal covariate histories
- Individual edge histories
- Complex attribute storage

The `delete.nodes` control setting in EpiModel

- `networkDynamic` ➠ `network` class

Helps considerably (7 days ➠ 14 hours), but **not enough**

Important changes to `tergm` storage by Statnet collaborators

So: the use of the `network` class itself is the **residual burden**

Minimum Data for Epidemic Models

Isolates not explicitly represented

Total network size in external data

Edgelist

```
      [,1] [,2]
 [1,]    1    3
 [2,]    1    5
 [3,]    1    8
 [4,]    4    9
 [5,]    6    3
 [6,]    7    5
 [7,]    7    9
 [8,]    9    5
 [9,]    9    8
attr(,"n")
[1] 10
```

Attributes

```
      age male
 [1,]  24    0
 [2,]  46    0
 [3,]  33    1
 [4,]  39    0
 [5,]  28    1
 [6,]  21    1
 [7,]  47    1
 [8,]  20    1
 [9,]  42    1
[10,]  26    0
```

Two Matrices

Isolates explicitly represented

Specifically, attributes referenced in the ERGM

The network Object Class

A Vertex

```
nw$val[[1]]
$na
[1] FALSE

$vertex.names
[1] "1"

$age
[1] 29

$male
[1] 1
```

network

>100x larger

than the minimum

double-matrix

An Edge

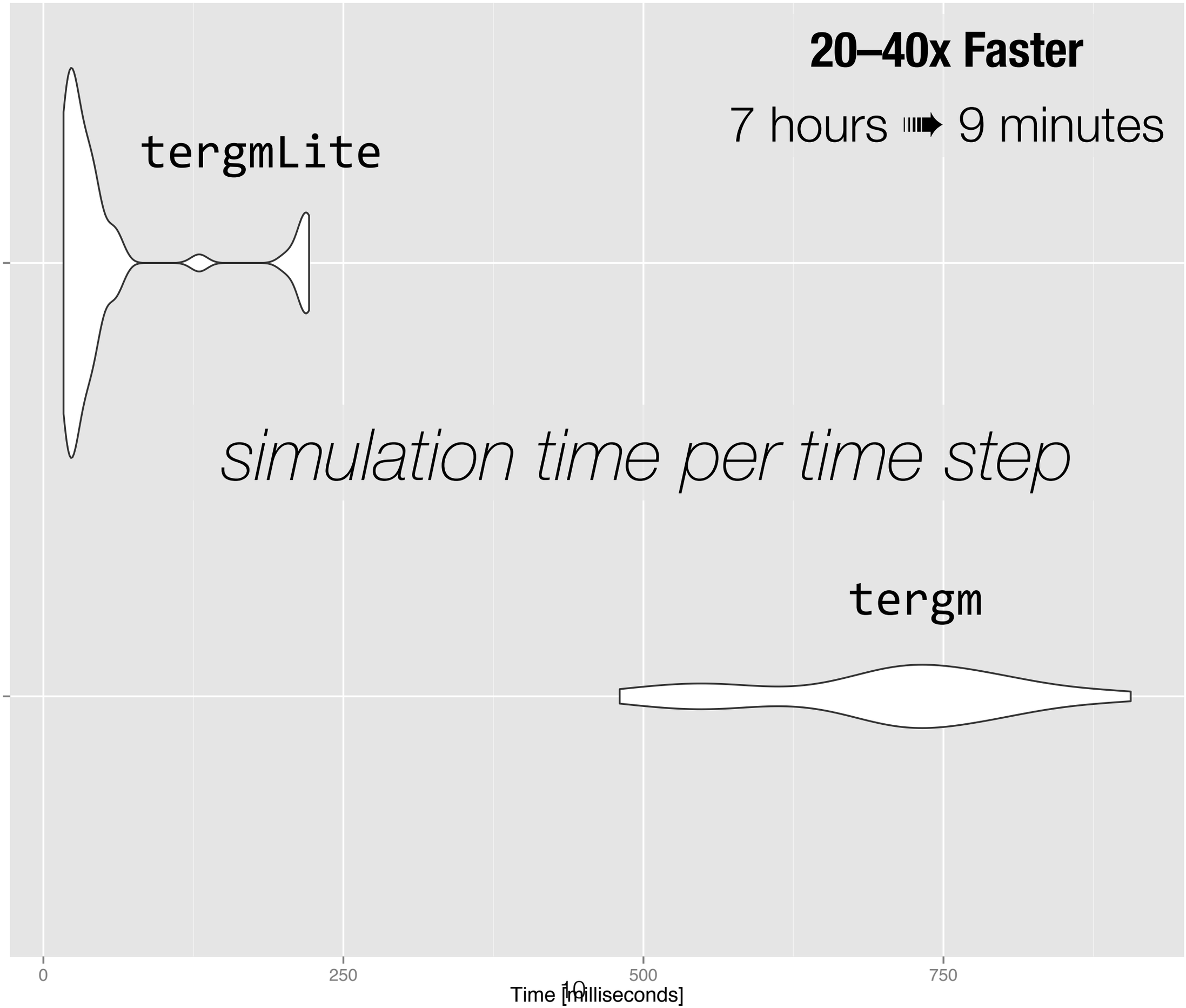
```
nw$mel[[1]]
$inl
[1] 1

$outl
[1] 280

$atl
$atl$na
[1] FALSE
```

"…sufficiently general to encode all major types of network data collected presently or in the foreseeable future."

* Carter B, JSS, 2008

20–40x Faster

7 hours ➠ 9 minutes



tergmLite within EpiModelHIV

Prep work and conversions

```
netsim(est, param, init, control)
    t1 initialization module
    for (t in 2 to tn) {
      module 1
      module 2
      network re-simulation module
      transmission module
      module 5
}
tn Process and save output
```

Fast version of simulate.network

Current Terms Supported in tergmLite

edges, nodematch, nodemix, nodefactor, nodecov, degree, concurrent, absdiff, absdiffby, absdiffnodemix

Other ergm-terms and newly coded terms easy to add once the term-specific format of the data structure is defined

Still working on a generalizable testing approach for this, but trust me it works!

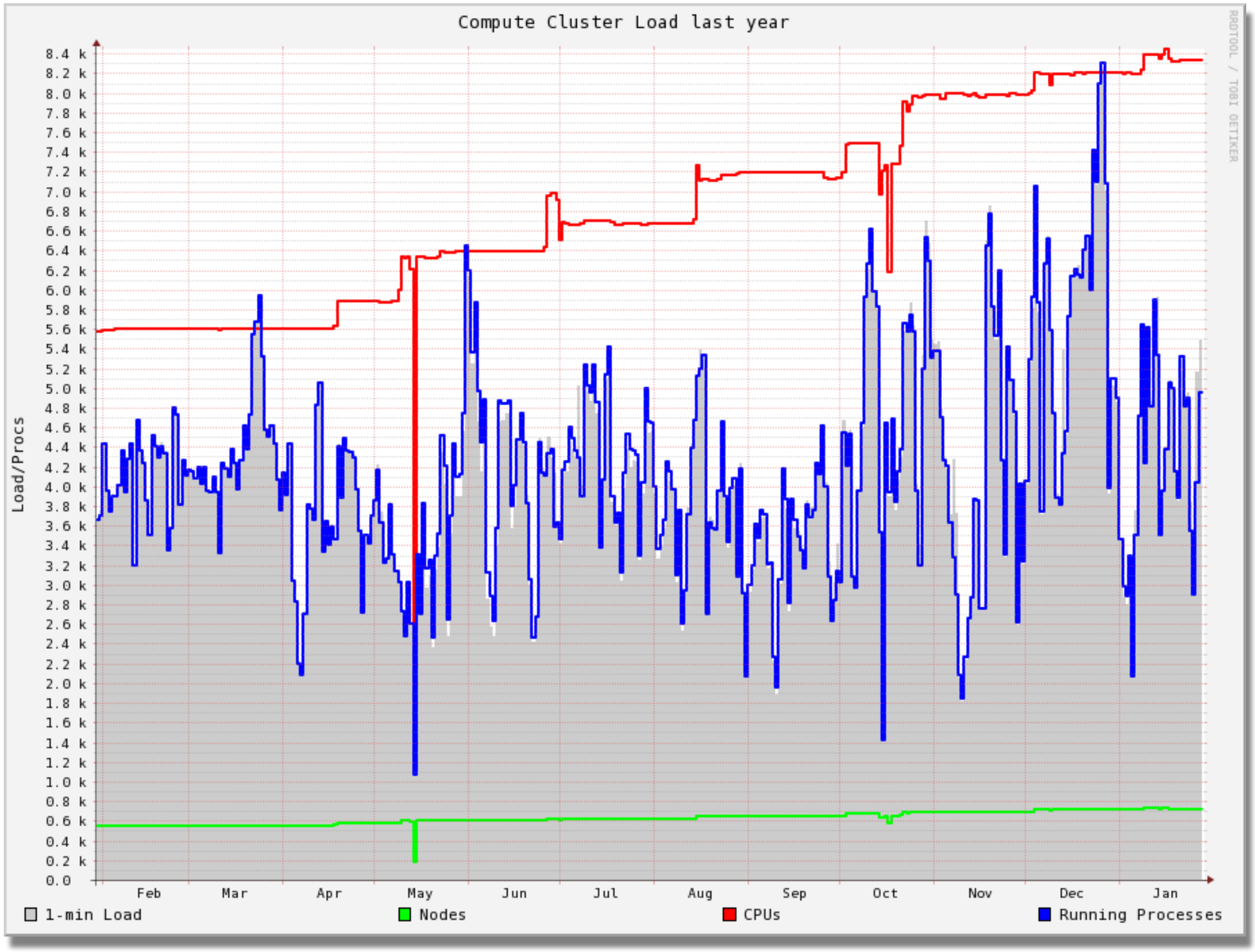
Overview

- Computational issues: tergmLite, HPCs
- Model calibration and sources of uncertainty in network models
- Steps of executing an EpiModel-based modeling project (on to Github…)

High Performance Computing (HPC)

- We run all of our research-level models on HPCs in parallel
- UW has Hyak system
  - Unix-based super cluster
  - >700 nodes, >10k cores
  - Group leases mean dedicated cores
  - Backfill queue allows broader access

Hyak Capacity and Utilization



Hyak Capacity and Utilization

```
2603764[1]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603766[1]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603761[3]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603767[1]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603765[1]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603762[1]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603762[2]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603761[4]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603766[2]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603766[3]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603761[1]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603763[4]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603763[1]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603764[2]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603765[3]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603764[4]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603762[3]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603763[3]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603765[4]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603764[3]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603761[2]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603763[2]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603765[2]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603762[4]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603766[4]      sjenness   Running   16   12:39:28  Sat Jan 31 09:02:58
2603767[4]      sjenness   Running   16   12:40:02  Sat Jan 31 09:03:32
2603767[3]      sjenness   Running   16   12:40:02  Sat Jan 31 09:03:32
2603768[1]      sjenness   Running   16   12:40:02  Sat Jan 31 09:03:32
2603768[3]      sjenness   Running   16   12:40:02  Sat Jan 31 09:03:32
2603768[4]      sjenness   Running   16   12:40:02  Sat Jan 31 09:03:32
2603767[2]      sjenness   Running   16   12:40:02  Sat Jan 31 09:03:32
2603768[2]      sjenness   Running   16   12:40:02  Sat Jan 31 09:03:32
```

```
127 active jobs          2032 of 7968 processors in use by local jobs (25.50%)
                           638 of 683 nodes active      (93.41%)
```

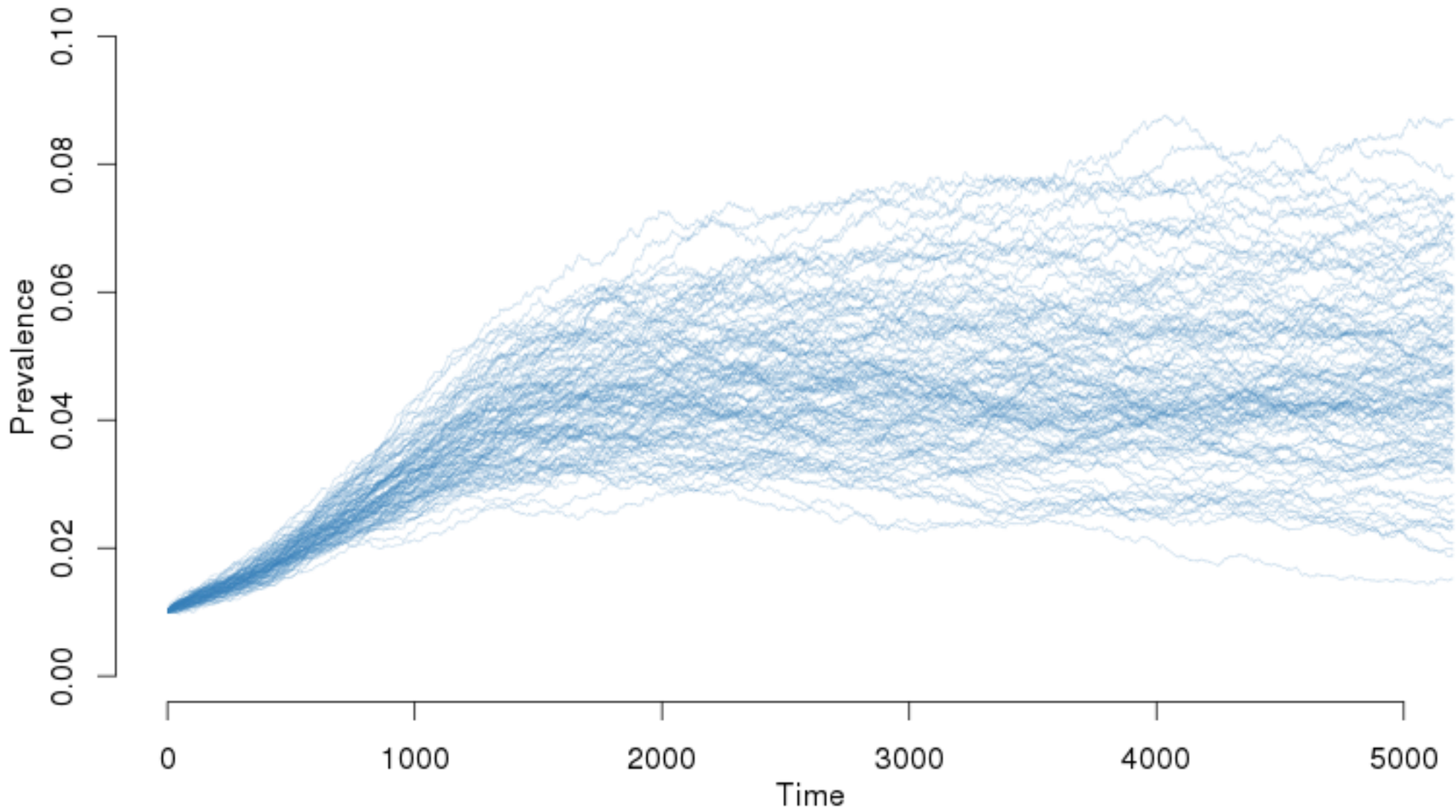
```
2603780[4]      sjenness   Running   16   14:53:59  Sat Jan 31 11:17:29
2603784[1]      sjenness   Running   16   14:53:59  Sat Jan 31 11:17:29
2603782[3]      sjenness   Running   16   14:53:59  Sat Jan 31 11:17:29
2603786[1]      sjenness   Running   16   14:53:59  Sat Jan 31 11:17:29
2603781[3]      sjenness   Running   16   14:53:59  Sat Jan 31 11:17:29
2603783[2]      sjenness   Running   16   14:53:59  Sat Jan 31 11:17:29
2603783[3]      sjenness   Running   16   14:53:59  Sat Jan 31 11:17:29
2603785[3]      sjenness   Running   16   14:53:59  Sat Jan 31 11:17:29
2603785[2]      sjenness   Running   16   14:53:59  Sat Jan 31 11:17:29
2603780[3]      sjenness   Running   16   14:53:59  Sat Jan 31 11:17:29
2603780[1]      sjenness   Running   16   14:53:59  Sat Jan 31 11:17:29
2603784[4]      sjenness   Running   16   14:53:59  Sat Jan 31 11:17:29
2603781[2]      sjenness   Running   16   14:53:59  Sat Jan 31 11:17:29
2603781[4]      sjenness   Running   16   14:53:59  Sat Jan 31 11:17:29
2603783[4]      sjenness   Running   16   14:53:59  Sat Jan 31 11:17:29
2603785[4]      sjenness   Running   16   14:53:59  Sat Jan 31 11:17:29
2603784[3]      sjenness   Running   16   14:53:59  Sat Jan 31 11:17:29
2603780[2]      sjenness   Running   16   14:53:59  Sat Jan 31 11:17:29
2603782[4]      sjenness   Running   16   14:53:59  Sat Jan 31 11:17:29
2603788[2]      sjenness   Running   16   14:54:33  Sat Jan 31 11:18:03
2603789[1]      sjenness   Running   16   14:54:33  Sat Jan 31 11:18:03
2603787[3]      sjenness   Running   16   14:54:33  Sat Jan 31 11:18:03
2603788[3]      sjenness   Running   16   14:54:33  Sat Jan 31 11:18:03
2603786[2]      sjenness   Running   16   14:54:33  Sat Jan 31 11:18:03
2603786[4]      sjenness   Running   16   14:54:33  Sat Jan 31 11:18:03
2603787[4]      sjenness   Running   16   14:54:33  Sat Jan 31 11:18:03
2603789[2]      sjenness   Running   16   14:54:33  Sat Jan 31 11:18:03
2603789[4]      sjenness   Running   16   14:54:33  Sat Jan 31 11:18:03
2603787[1]      sjenness   Running   16   14:54:33  Sat Jan 31 11:18:03
2603789[3]      sjenness   Running   16   14:54:33  Sat Jan 31 11:18:03
2603788[1]      sjenness   Running   16   14:54:33  Sat Jan 31 11:18:03
2603787[2]      sjenness   Running   16   14:54:33  Sat Jan 31 11:18:03
2603786[3]      sjenness   Running   16   14:54:33  Sat Jan 31 11:18:03
2603788[4]      sjenness   Running   16   14:54:33  Sat Jan 31 11:18:03
```

HPC Software Stack for EpiModel

1. Sequential process **application**
2. SOCK/MPI-based parallel **method**
3. R **script** running parallelized method
4. Torque **script** call to app script
5. Shell **call** torque script

Parallel Processes

Embarrassingly parallel multiples of **inherently serial** problems



Parallel Processes

- Parallelization of R-based apps is relatively new
- Our examples
  - EpiModel's `ncores` parameter
  - ergm's `parallel` parameter
- Commonalities
  - Set up clusters
  - Separate computations
  - Merge of data

Getting Parallel in EpiModel

```
cluster.size <- nsims

cl <- startMPIcluster(cluster.size)
registerDoMPI(cl)

out <- foreach(i = 1:nsims) %dopar% {
  require(EpiModel)
  control$nsims <- 1
  netsim(x, param, init, control)
}

closeCluster(cl)
```

Stack L2

Stack L1

HPC Job Organization and Scheduling

- Hyak jobs managed by Torque
  - Used by most super HPCs today
- Job schedule based on user specs
  - How many **cores** → # of simulations
  - How much **time** → # of time steps
  - How much **memory** → NW size x time steps
- Interactive mode also “scheduled”

HPC Job Organization and Scheduling

Example HPC Calendar

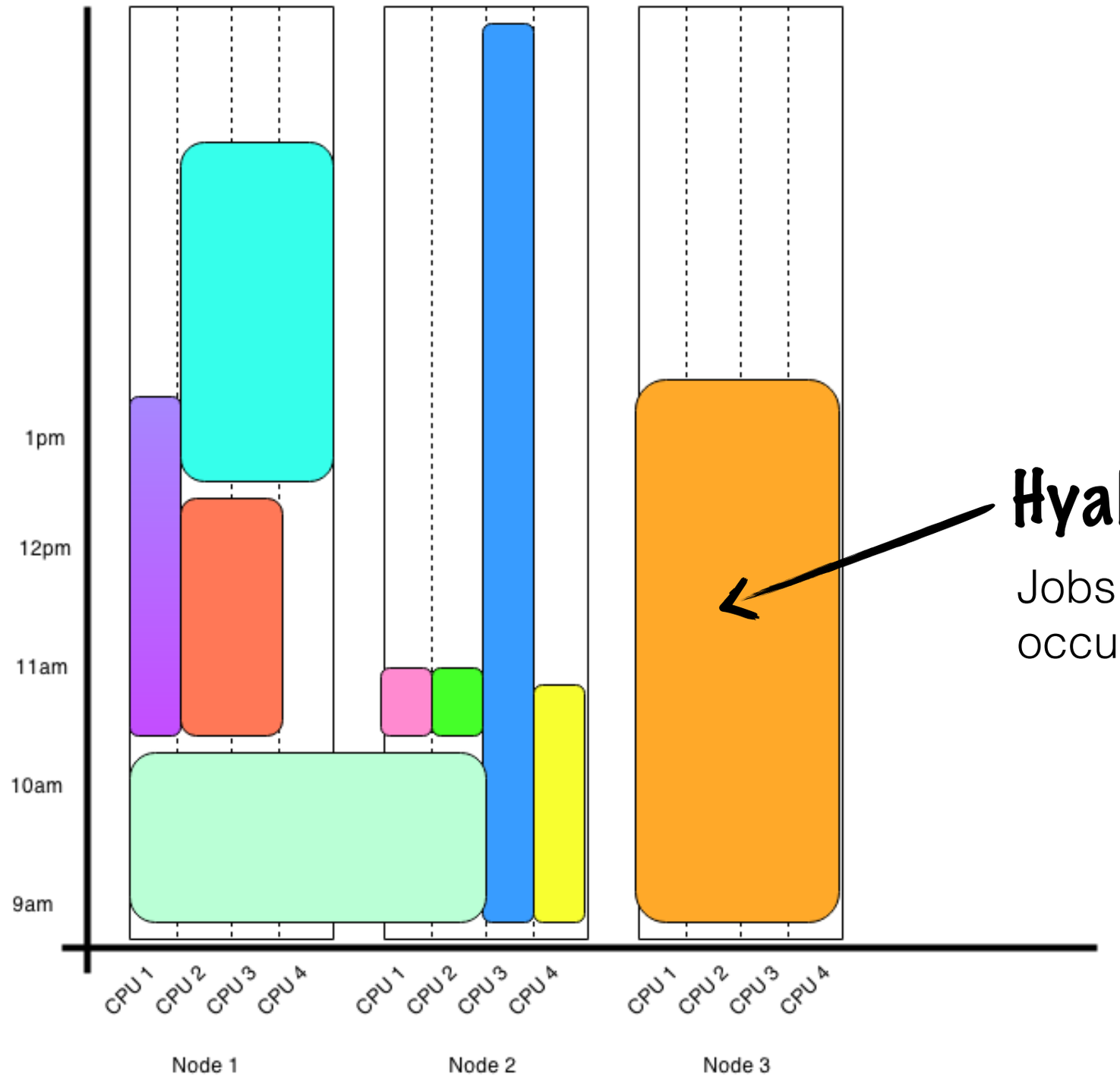
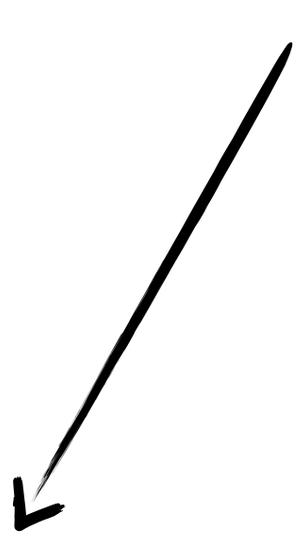


Image from NYU Torque Wiki

Example Torque Script

```
#!/bin/bash

#PBS -N intel-openmpi

#PBS -l nodes=2:ppn=16,mem=22gb,feature=16core
#PBS -l walltime=00:10:00
#PBS -N dissmod
#PBS -o /gscratch/csde/sjenness
#PBS -j oe
#PBS -d /gscratch/csde/sjenness
#PBS -m ae
#PBS -M sjenness@u.washington.edu

module load r_3.1.1 icc_14.0.3-ompi_1.8.3

mpirun -np 1 R --slave CMD BATCH --vanilla sim$SIMNO.R
```

Stack L3



Example Torque Script

In terminal

```
qsub runsim1.sh
qsub -l nodes=1:ppn=16,mem=30gb,feature=16core -v SIMNO=1 runsim.sh
qsub -l nodes=7:ppn=16,mem=210gb,feature=16core -v SIMNO=2 runsim.sh
```

Or define a bash shell function

```
runsim () {
   qsub -l nodes=$2:ppn=16,mem=$3gb,feature=16core -v SIMNO=$1
            runsim.sh
}

runsim 1 1 30
runsim 2 7 210
```

Level 5: Shell Call

```
qsub -q bf –t 1-4 -v SIMNO=363 runsim.sh
```

Level 4: Torque Script

```
mpirun -np 1 R --slave CMD BATCH -$SIMNO.${PBS_ARRAYID} sim$SIMNO.R
```

Level 3: R Script

```
fsimno <- sub(“-“, "", tail(commandArgs(FALSE), 1))
control <- control.hiv(simno = fsimno, nsims = 15, ncores = 15)
runsimHPC(“est/estfile.rda”, param, init, control)
```

Level 2: Parallel Method

```
out <- foreach(i = 1:nsims) %dopar% {
 control$currsim <- i
 netsim(x, param, init, control)
}
```

Level 1: Sequential App

```
if (at %% save.int == 0) {
 fn <- paste0("sim", simno, "/sim", currsim, ".cp.rda")
 save(x, file = fn)
}
```

Model Calibration

- Critical for applied intervention models in which the starting condition (disease prevalence/incidence) reflects real-world observations
- Reflects uncertainty in the underlying parameter values
- Nothing in our calibration approach specific to network models or ERGMs, but is specific to stochastic systems
- Calibration addresses one of three sources of uncertainty in the inputs and outputs of our models
    - Next a review/recap from NEEMA 2016 Meeting…

Uncertainty in the Inputs: Calibration Methods

Approximate Bayesian Computation with Sequential Monte Carlo (ABC-SMC)

- Define prior distributions for uncertain input parameters
- Draw from one sample from the distribution
- Simulate the model with that parameter set
- Compare outcome statistics (prev/inc) to external target data
- If sufficiently close to targets, add parameters to posterior; if not, throw them out
- Use accepted parameters to inform choice of next parameter draw

Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems

Tina Toni[1,2,*], David Welch[3,†], Natalja Strelkowa[4], Andreas Ipsen[5] and Michael P. H. Stumpf[1,2,*]

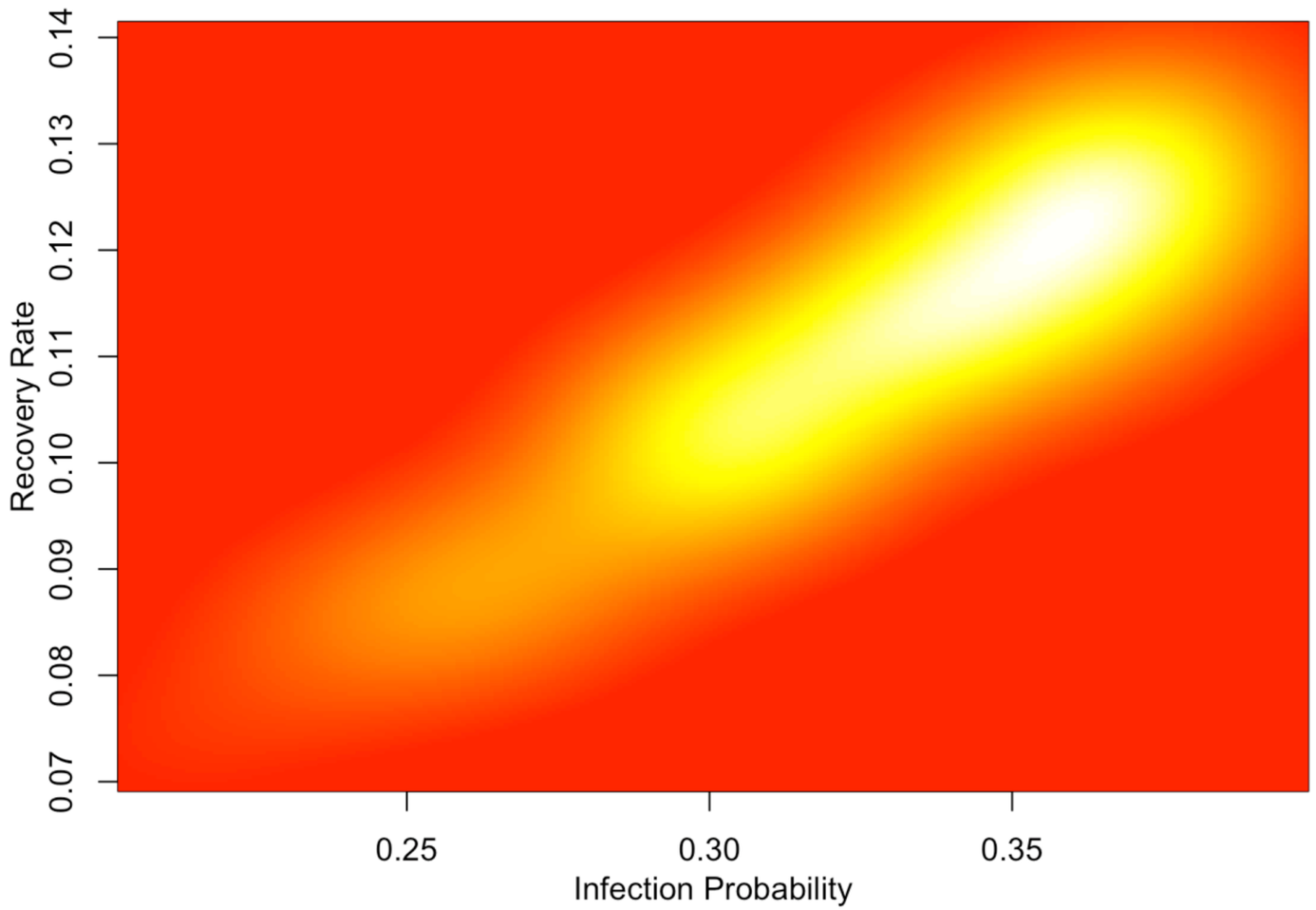
Table S1. Parameter Definitions, Posterior Value, Prior Distributions, and Sources for STI-related Model Parameters

Parameter	Posterior Mean	Priors	Sources
Per-act Transmission Probability			
Rectal Gonorrhea	0.3577	0.30 – 0.60	6–8
Urethral Gonorrhea	0.2481	0.20 – 0.50	9,10
Rectal Chlamydia	0.3216	0.30 – 0.60	11
Urethral Chlamydia	0.2130	0.20 – 0.50	12–14
Probability of Symptoms Given Infection			

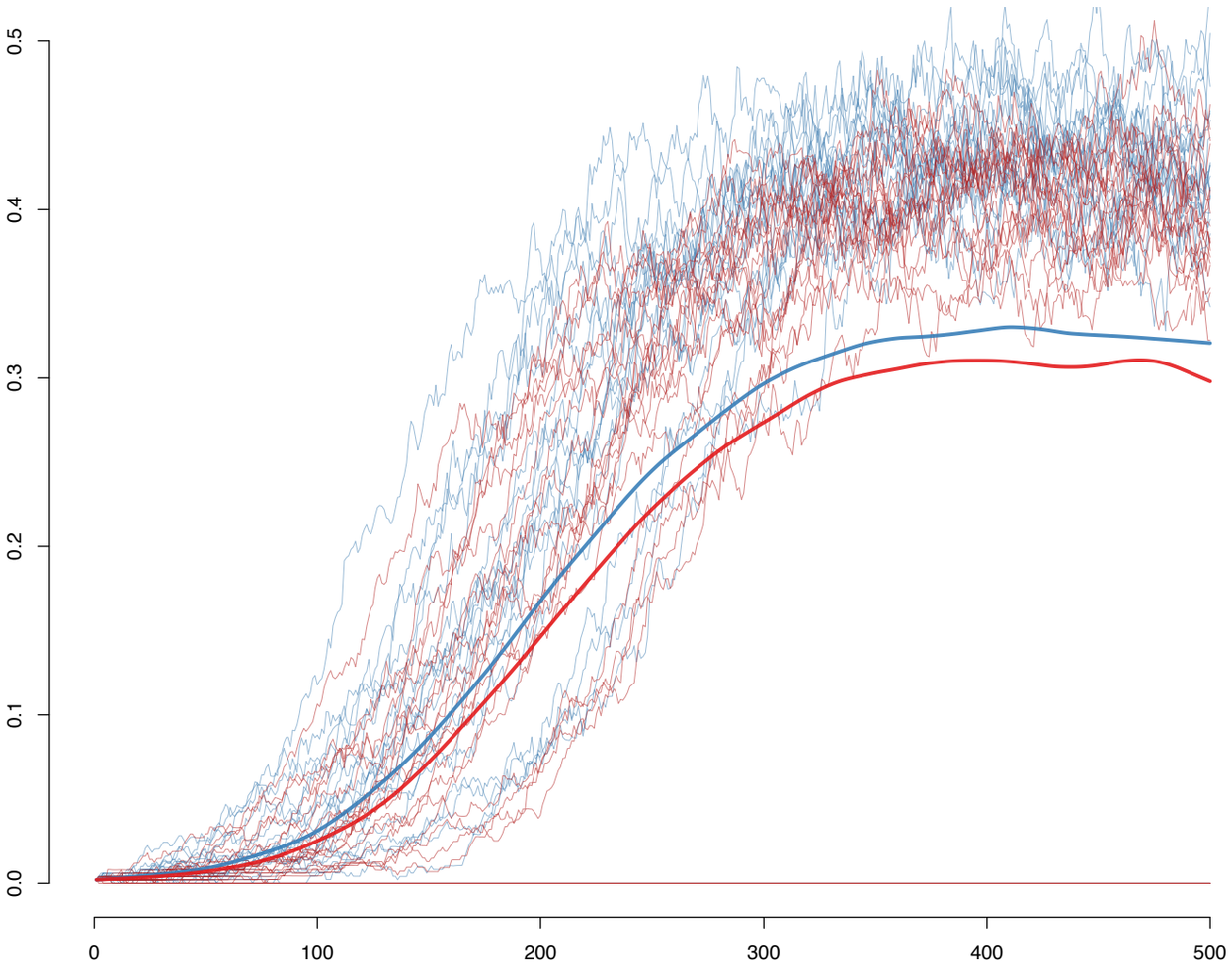
Uncertainty in the Inputs: Calibration Methods

ABC Challenges

- Computationally demanding (dynamic feedback loops with correlated parameters)
- Choice of which parameters to fit and the prior distributions not always clear
- The identifiability problem: not enough degrees of freedom to find a unique solution
- Number of inputs typically larger than the number of outputs
- Parameter inference vs. model calibration



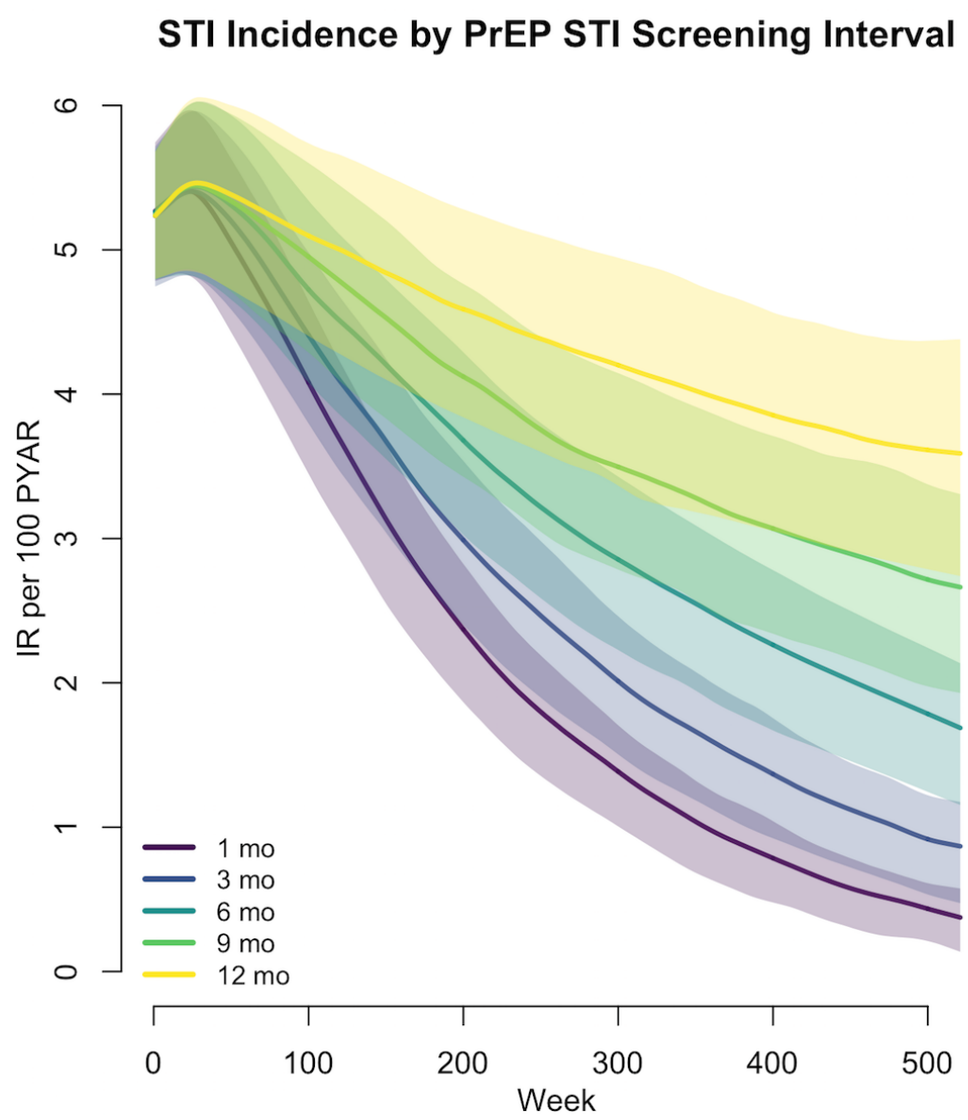
Uncertainty in the Outputs: Agent-Based Models

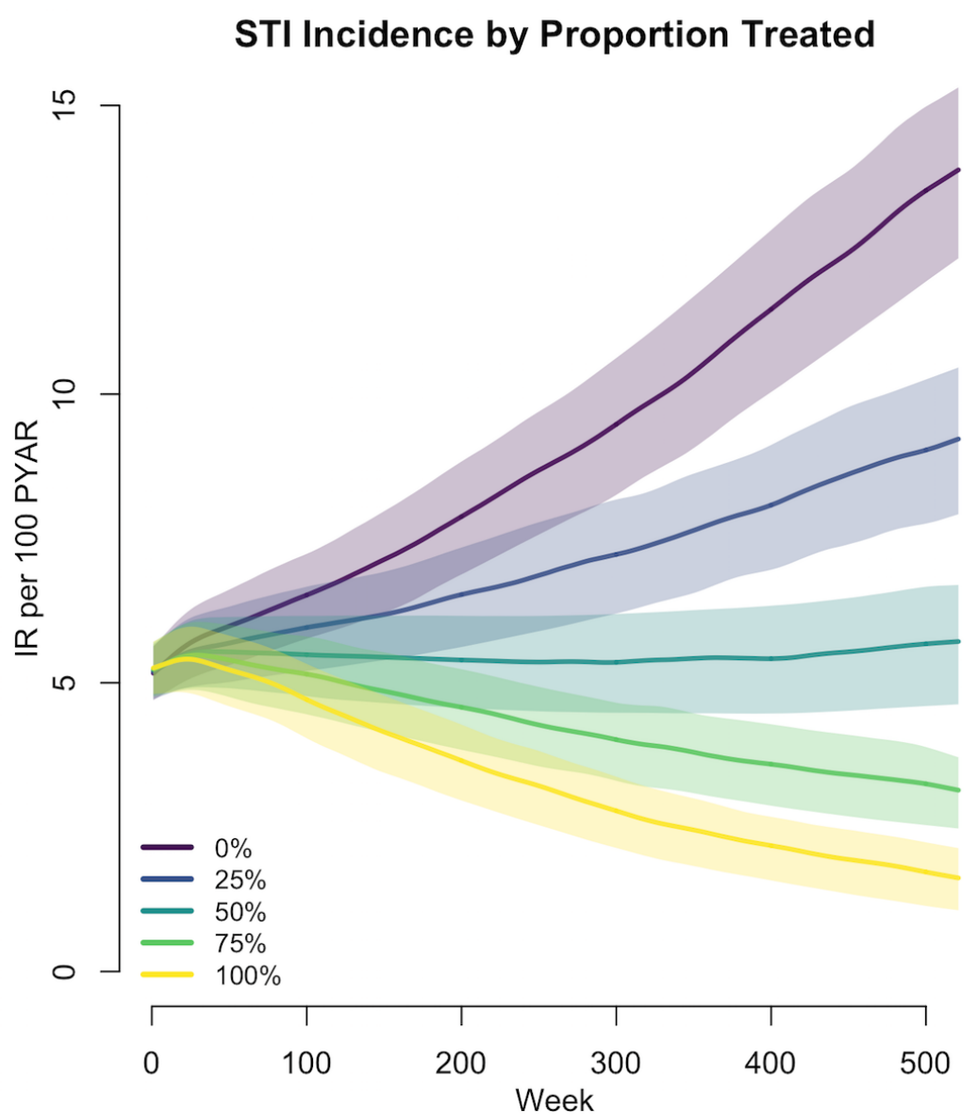


Distribution of Outcomes

- Major strength of ABMs (including our network models) in generating a distribution of outcomes given a chosen parameter set
- Critical implications for non-normal distributed outcomes (e.g., bimodal) common with low-incidence transmission systems
- Formal analysis of variance provides insight into key drivers of uncertainty

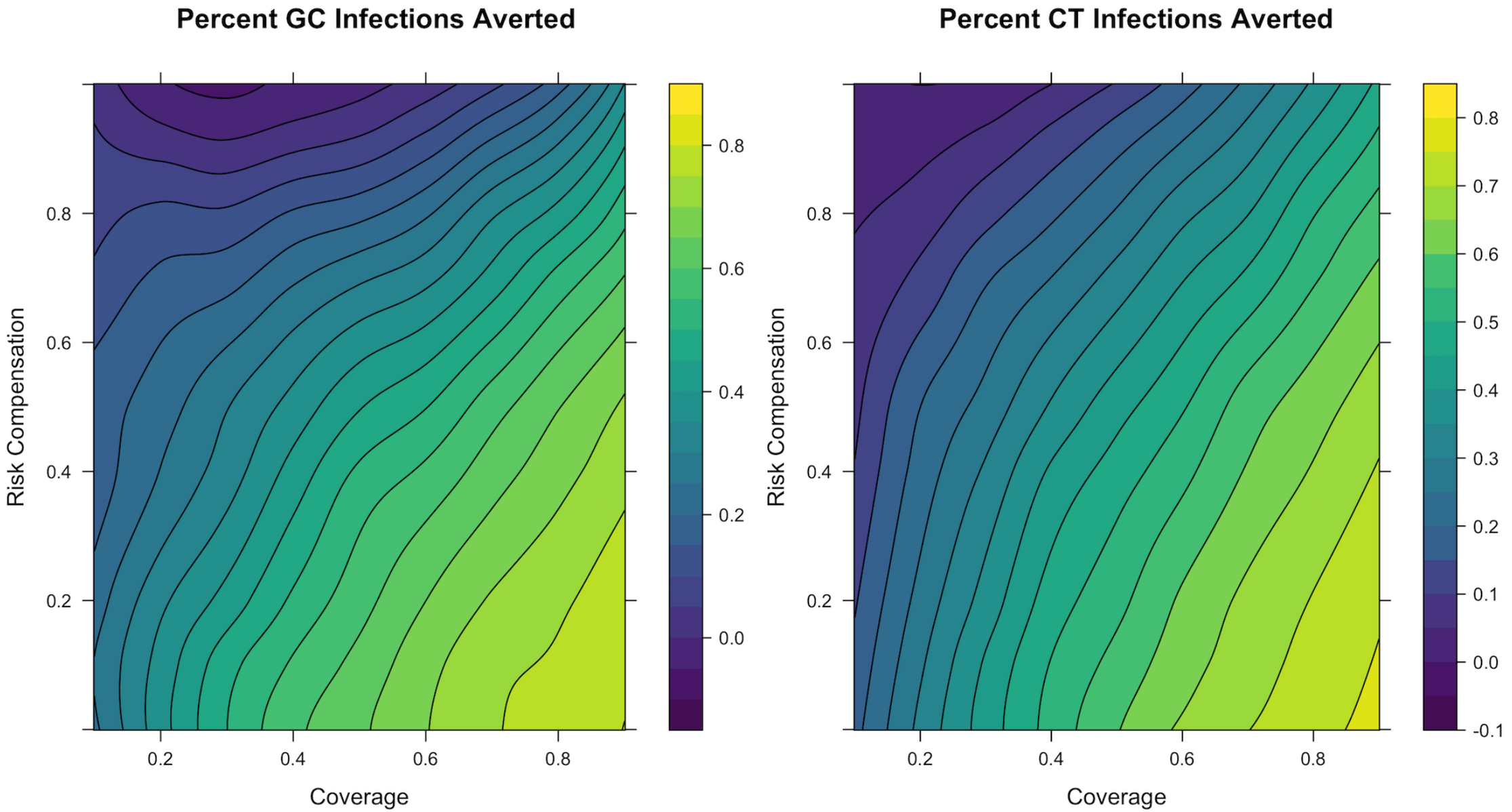
Uncertainty in the Outputs: Agent-Based Models





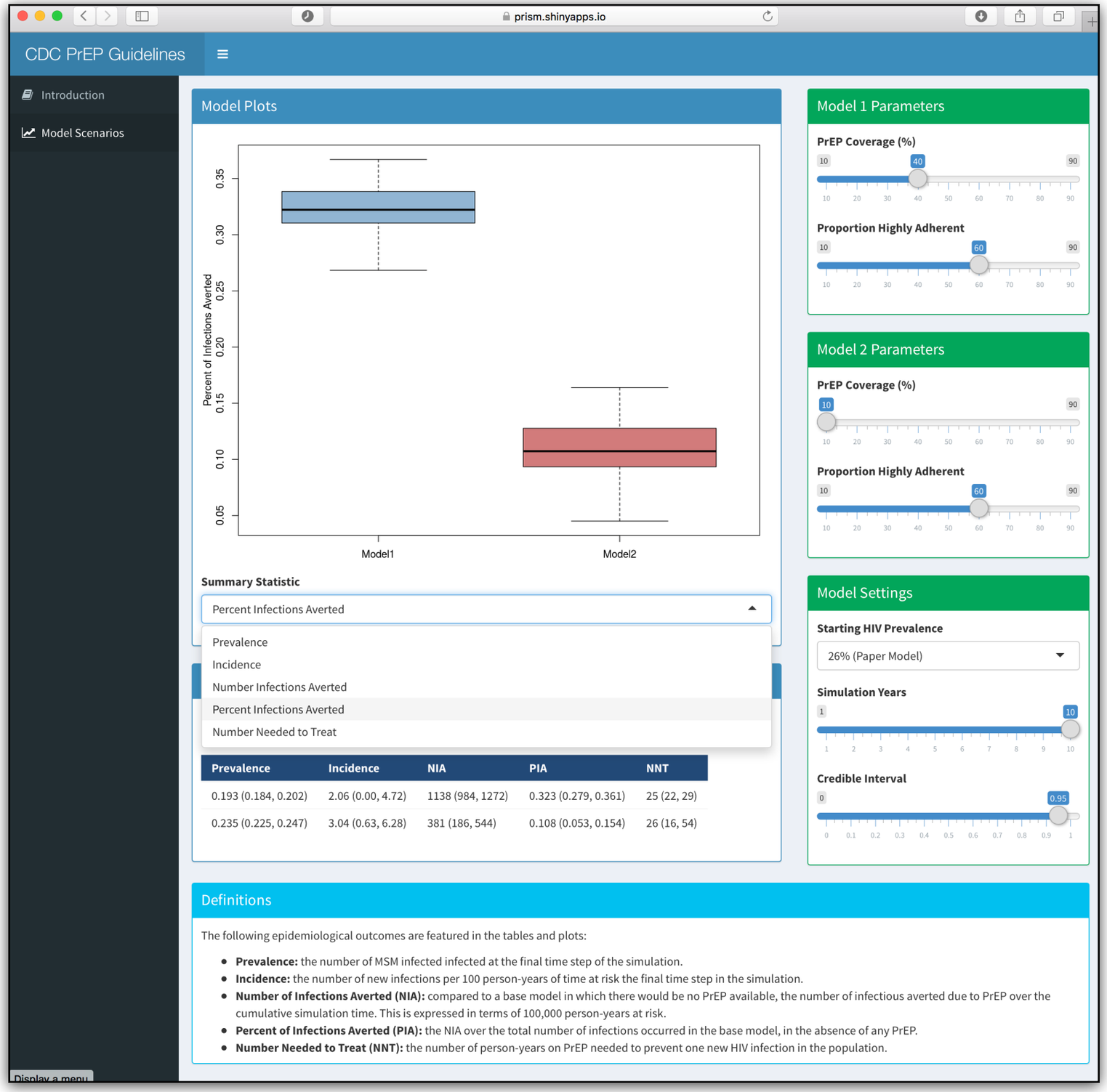
- Challenge is that the variability is a function of somewhat arbitrary simulation control settings: number of simulations, population size
- Difficult to explain that overlapping simulation intervals does not imply evidence against rejecting the “null hypothesis” of no difference

Uncertainty in the Future: Sensitivity Analyses



- Main sensitivity analyses for our intervention models focused on what we can't know over our future prediction interval
- Still makes large assumptions about stability to all other components of the transmission system for sake of study scope and clarity

Uncertainty in the Future: Sensitivity Analyses



Punting the Uncertainty

- Goal of translating our model outcomes into useful tools for public health practice
- Example of PrEP guidelines tool
- Allows for users to experiment with ranges of parameters and model conditions for local data

http://prism.shinyapps.io/cdc-prep-guidelines/

Overview

- Computational issues: tergmLite, HPCs
- Model calibration and sources of uncertainty in network models
- Steps of executing an EpiModel-based modeling project (on to Github…)

How to Execute an EpiModel

- Organization of code on Github into repositories
- Organization of project code by steps in the process
- Team coding, code review
- Analysis, writing, and publication