BCR Work-Precision Diagrams

Samuel Isaacson and Chris Rackauckas

August 7, 2021

The following benchmark is of 1122 ODEs with 24388 terms that describe a stiff chemical reaction network modeling the BCR signaling network from Barua et al.. We use ReactionNetworkImporters to load the BioNetGen model files as a Catalyst model, and then use ModelingToolkit to convert the Catalyst network model to ODEs.

```julia
using DiffEqBase, OrdinaryDiffEq, Catalyst, ReactionNetworkImporters,
      Sundials, Plots, DiffEqDevTools, ODEInterface, ODEInterfaceDiffEq,
      LSODA, TimerOutputs, LinearAlgebra, ModelingToolkit

gr()
datadir  = joinpath(dirname(pathof(ReactionNetworkImporters)),"../data/bcr")
const to = TimerOutput()
tf       = 100000.0

# generate ModelingToolkit ODEs
@timeit to "Parse Network" prnbng = loadrxnetwork(BNGNetwork(), joinpath(datadir,
"bcr.net"))
rn    = prnbng.rn
@timeit to "Create ODESys" osys = convert(ODESystem, rn)

u_0    = prnbng.u_0
p      = prnbng.p
tspan = (0.,tf)
@timeit to "ODEProb No Jac" oprob = ODEProblem(osys, u_0, tspan, p)
@timeit to "ODEProb DenseJac" densejacprob = ODEProblem(osys, u_0, tspan, p, jac=true)
```

```
Parsing parameters...done
Adding parameters...done
Parsing species...done
Adding species...done
Creating ModelingToolkit versions of species and parameters...done
Parsing and adding reactions...done
Parsing groups...done
ODEProblem with uType Vector{Float64} and tType Float64. In-place: true
timespan: (0.0, 100000.0)
u0: 1122-element Vector{Float64}:
 299717.8348854
  47149.15480798
  46979.01102231
 290771.2428252
 299980.7396749
 300000.0
    141.3151575495
      0.1256496403614
      0.4048783555301
```

```
     140.8052338618
        ⋮
       1.005585387399e-24
       6.724953378237e-17
       3.395560698281e-16
       1.787990228838e-5
       8.761844379939e-13
       0.0002517949074779
       0.0005539124513976
       2.281251822741e-14
       1.78232055967e-8
```

```julia
@timeit to "ODEProb SparseJac" sparsejacprob = ODEProblem(osys, u_0, tspan, p, jac=true,
sparse=true)
show(to)
```

```
                          Time                    Allocations


     Tot / % measured:          303s / 100%           51.1GiB / 100%


 Section              ncalls    time   %tot    avg     alloc   %tot      a
vg

 ODEProb DenseJac        1      261s  86.2%    261s   42.4GiB  83.0%  42.4G
iB
 ODEProb SparseJac       1     28.1s  9.26%   28.1s   5.24GiB  10.3%  5.24G
iB
 ODEProb No Jac          1     11.2s  3.70%   11.2s   2.80GiB  5.49%  2.80G
iB
 Create ODESys           1     1.47s  0.48%   1.47s    501MiB  0.96%   501M
iB
 Parse Network           1     1.01s  0.33%   1.01s    163MiB  0.31%   163M
iB
```

```julia
@show numspecies(rn) # Number of ODEs
@show numreactions(rn) # Apprx. number of terms in the ODE
@show numparams(rn) # Number of Parameters
```

```
numspecies(rn) = 1122
numreactions(rn) = 24388
numparams(rn) = 128
128
```

0.1 Time ODE derivative function compilation

As compiling the ODE derivative functions has in the past taken longer than running a simulation, we first force compilation by evaluating these functions one time.

```julia
u  = copy(u_0)
```

2

```
du = similar(u)
@timeit to "ODERHS Eval1" oprob.f(du,u,p,0.)
@timeit to "ODERHS Eval2" oprob.f(du,u,p,0.)

# force compilation for dense and sparse problem rhs
densejacprob.f(du,u,p,0.)
sparsejacprob.f(du,u,p,0.)

J = zeros(length(u),length(u))
@timeit to "DenseJac Eval1" densejacprob.f.jac(J,u,p,0.)
@timeit to "DenseJac Eval2" densejacprob.f.jac(J,u,p,0.)
```

```
Error: syntax: expression too large
```

```
Js = similar(sparsejacprob.f.jac_prototype)
@timeit to "SparseJac Eval1" sparsejacprob.f.jac(Js,u,p,0.)
@timeit to "SparseJac Eval2" sparsejacprob.f.jac(Js,u,p,0.)
show(to)
```
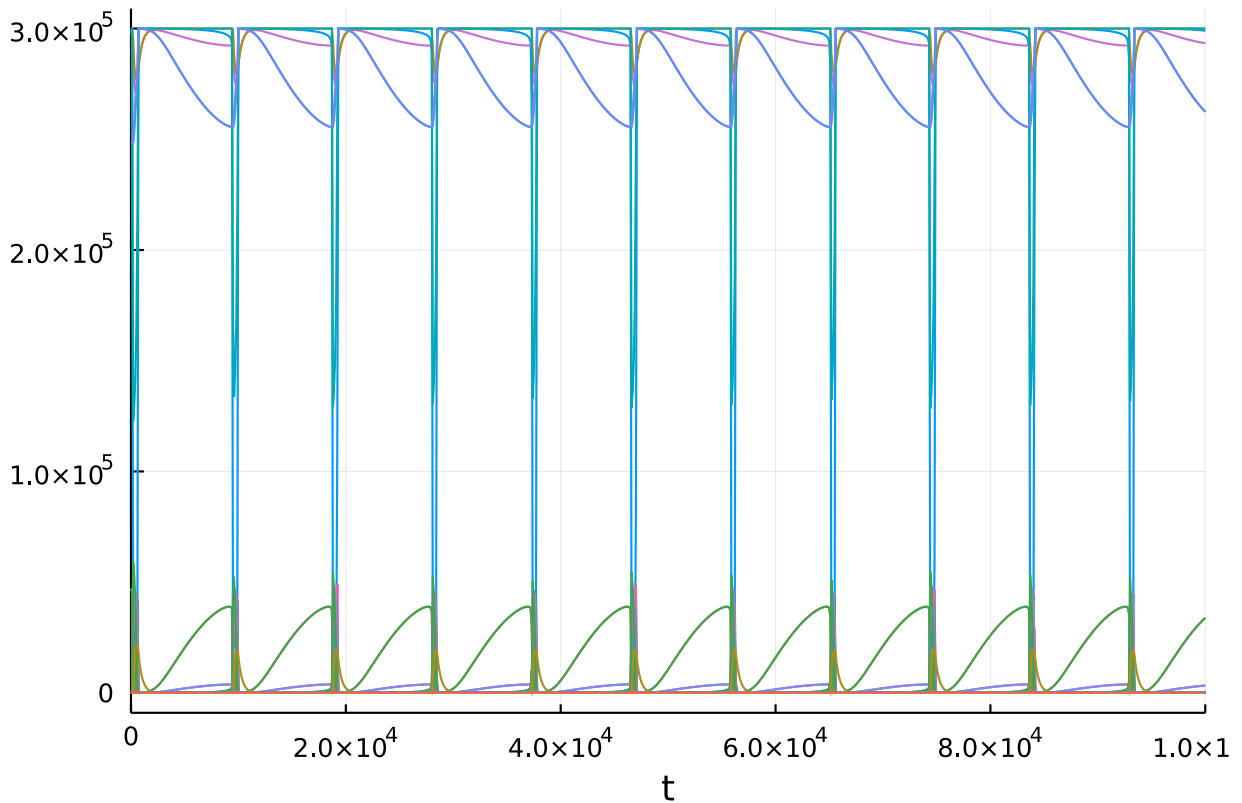
```
Error: syntax: expression too large
```

0.2 Picture of the solution

```
sol = solve(oprob, CVODE_BDF(), saveat=tf/1000., reltol=1e-5, abstol=1e-5)
plot(sol, legend=false, fmt=:png)
```


For these benchmarks we will be using the time-series error with these saving points since the final time point is not well-indicative of the solution behavior (capturing the oscillation

is the key!).

0.3 Generate Test Solution

```
@time sol = solve(oprob, CVODE_BDF(), abstol=1/10^12, reltol=1/10^12)
test_sol = TestSolution(sol)
```

```
630.057369 seconds (4.71 M allocations: 2.204 GiB, 0.36% gc time, 0.10% com
pilation time)
retcode: Success
Interpolation: 3rd order Hermite
t: nothing
u: nothing
```

0.4 Setups

```
abstols = 1.0 ./ 10.0 .^ (5:8)
reltols = 1.0 ./ 10.0 .^ (5:8);
setups = [
          #Dict(:alg=>Rosenbrock23(autodiff=false)),
          Dict(:alg=>TRBDF2(autodiff=false)),
          Dict(:alg=>QNDF(autodiff=false)),
          Dict(:alg=>CVODE_BDF()),
          Dict(:alg=>CVODE_BDF(linear_solver=:LapackDense)),
          #Dict(:alg=>rodas()),
          #Dict(:alg=>radau()),
          #Dict(:alg=>Rodas4(autodiff=false)),
          #Dict(:alg=>Rodas5(autodiff=false)),
          Dict(:alg=>KenCarp4(autodiff=false)),
          Dict(:alg=>KenCarp47(autodiff=false)),
          #Dict(:alg=>RadauIIA5(autodiff=false)),
          #Dict(:alg=>lsoda()),
          ]
```

```
6-element Vector{Dict{Symbol, V} where V}:
 Dict{Symbol, OrdinaryDiffEq.TRBDF2{0, false, DiffEqBase.DefaultLinSolve, D
iffEqBase.NLNewton{Rational{Int64}, Rational{Int64}, Rational{Int64}}, Data
Type}}(:alg => OrdinaryDiffEq.TRBDF2{0, false, DiffEqBase.DefaultLinSolve,
DiffEqBase.NLNewton{Rational{Int64}, Rational{Int64}, Rational{Int64}}, Dat
aType}(DiffEqBase.DefaultLinSolve(nothing, nothing), DiffEqBase.NLNewton{Ra
tional{Int64}, Rational{Int64}, Rational{Int64}}(1//100, 10, 1//5, 1//5), V
al{:forward}, true, :linear, :PI))
 Dict{Symbol, OrdinaryDiffEq.QNDF{5, 0, false, DiffEqBase.DefaultLinSolve,
DiffEqBase.NLNewton{Rational{Int64}, Rational{Int64}, Rational{Int64}}, Dat
aType, Nothing, Nothing, NTuple{5, Float64}}}(:alg => OrdinaryDiffEq.QNDF{5
, 0, false, DiffEqBase.DefaultLinSolve, DiffEqBase.NLNewton{Rational{Int64}
, Rational{Int64}, Rational{Int64}}, DataType, Nothing, Nothing, NTuple{5,
Float64}}(Val{5}(), DiffEqBase.DefaultLinSolve(nothing, nothing), DiffEqBas
e.NLNewton{Rational{Int64}, Rational{Int64}, Rational{Int64}}(1//100, 10, 1
//5, 1//5), Val{:forward}, nothing, nothing, :linear, (-0.185, -0.111111111
1111111, -0.0823, -0.0415, 0.0), :Standard))
 Dict{Symbol, Sundials.CVODE_BDF{:Newton, :Dense, Nothing, Nothing}}(:alg =
> Sundials.CVODE_BDF{:Newton, :Dense, Nothing, Nothing}(0, 0, 0, false, 10,
 5, 7, 3, 10, nothing, nothing, 0))
 Dict{Symbol, Sundials.CVODE_BDF{:Newton, :LapackDense, Nothing, Nothing}}(
:alg => Sundials.CVODE_BDF{:Newton, :LapackDense, Nothing, Nothing}(0, 0, 0
, false, 10, 5, 7, 3, 10, nothing, nothing, 0))
 Dict{Symbol, OrdinaryDiffEq.KenCarp4{0, false, DiffEqBase.DefaultLinSolve,
 DiffEqBase.NLNewton{Rational{Int64}, Rational{Int64}, Rational{Int64}}, Da
```

```
taType}}(:alg => OrdinaryDiffEq.KenCarp4{0, false, DiffEqBase.DefaultLinSol
ve, DiffEqBase.NLNewton{Rational{Int64}, Rational{Int64}, Rational{Int64}},
 DataType}(DiffEqBase.DefaultLinSolve(nothing, nothing), DiffEqBase.NLNewto
n{Rational{Int64}, Rational{Int64}, Rational{Int64}}(1//100, 10, 1//5, 1//5
), Val{:forward}, true, :linear, :PI))
 Dict{Symbol, OrdinaryDiffEq.KenCarp47{0, false, DiffEqBase.DefaultLinSolve
, DiffEqBase.NLNewton{Rational{Int64}, Rational{Int64}, Rational{Int64}}, D
ataType}}(:alg => OrdinaryDiffEq.KenCarp47{0, false, DiffEqBase.DefaultLinS
olve, DiffEqBase.NLNewton{Rational{Int64}, Rational{Int64}, Rational{Int64}
}, DataType}(DiffEqBase.DefaultLinSolve(nothing, nothing), DiffEqBase.NLNew
ton{Rational{Int64}, Rational{Int64}, Rational{Int64}}(1//100, 10, 1//5, 1/
/5), Val{:forward}, true, :linear, :PI))
```

0.5   Automatic Jacobian Solves

Due to the computational cost of the problem, we are only going to focus on the methods
which demonstrated computational efficiency on the smaller biochemical benchmark prob-
lems. This excludes the exponential integrator, stabilized explicit, and extrapolation classes
of methods.

First we test using auto-generated Jacobians (finite difference)

```
wp = WorkPrecisionSet(oprob,abstols,reltols,setups;error_estimate=:l2,
                saveat=tf/10000.,appxsol=test_sol,maxiters=Int(1e5),numruns=1)
plot(wp)
```
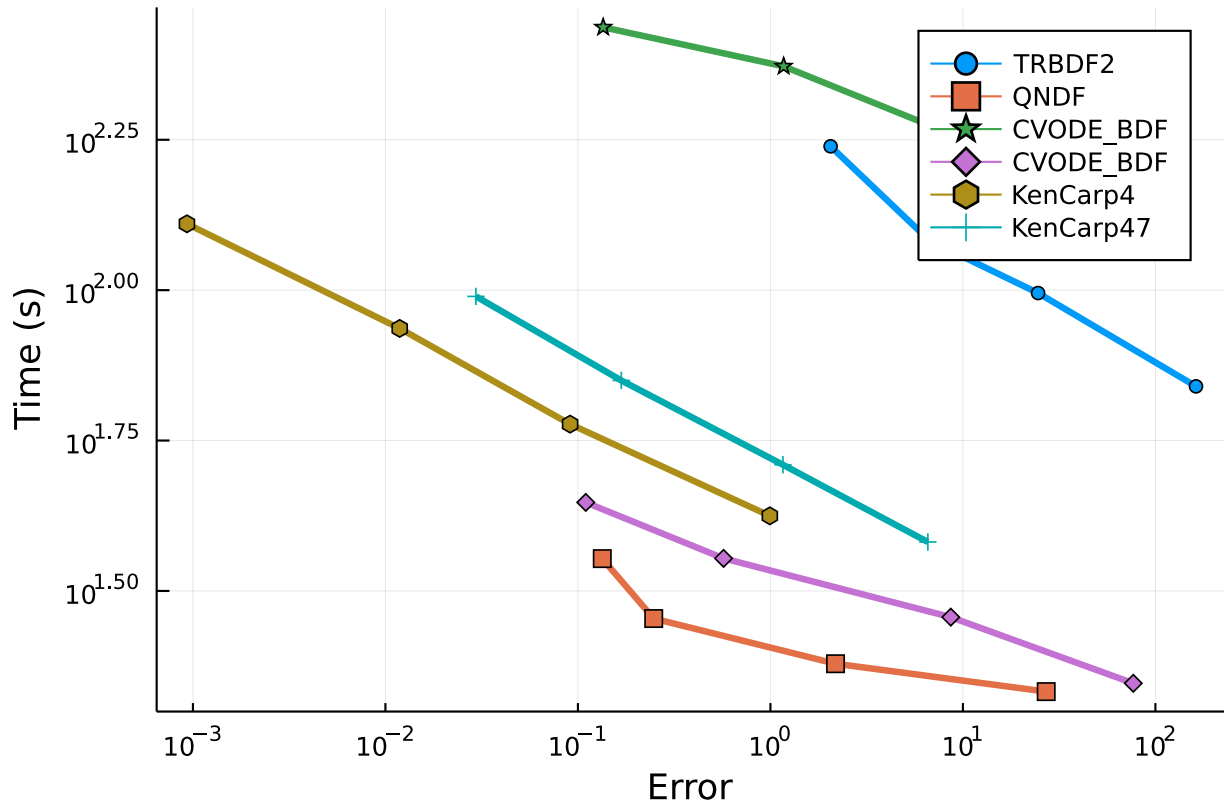

0.6   Analytical Jacobian

Now we test using the generated analytic Jacobian function.

```
wp = WorkPrecisionSet(densejacprob,abstols,reltols,setups;error_estimate=:l2,
                      saveat=tf/10000.,appxsol=test_sol,maxiters=Int(1e5),numruns=1)
plot(wp)

Error: syntax: expression too large
```

0.7   Sparse Jacobian

Finally we test using the generated sparse analytic Jacobian function.

```
setups = [
          #Dict(:alg=>Rosenbrock23(autodiff=false)),
          Dict(:alg=>TRBDF2(autodiff=false)),
          Dict(:alg=>QNDF(autodiff=false)),
          #Dict(:alg=>CVODE_BDF(linear_solver=:KLU)), # Fails!
          #Dict(:alg=>rodas()),
          #Dict(:alg=>radau()),
          #Dict(:alg=>Rodas4(autodiff=false)),
          #Dict(:alg=>Rodas5(autodiff=false)),
          Dict(:alg=>KenCarp4(autodiff=false)),
          Dict(:alg=>KenCarp47(autodiff=false)),
          #Dict(:alg=>RadauIIA5(autodiff=false)),
          #Dict(:alg=>lsoda()),
          ]
wp = WorkPrecisionSet(sparsejacprob,abstols,reltols,setups;error_estimate=:l2,
                      saveat=tf/10000.,appxsol=test_sol,maxiters=Int(1e5),numruns=1)
plot(wp)

Error: syntax: expression too large
```

0.8   Appendix

These benchmarks are a part of the SciMLBenchmarks.jl repository, found at: https://github.com/SciML/
For more information on high-performance scientific machine learning, check out the SciML
Open Source Software Organization https://sciml.ai.

To locally run this benchmark, do the following commands:

```
using SciMLBenchmarks
SciMLBenchmarks.weave_file("benchmarks/Bio","BCR.jmd")
```

Computer Information:

```
Julia Version 1.6.2
Commit 1b93d53fc4 (2021-07-14 15:36 UTC)
Platform Info:
  OS: Linux (x86_64-pc-linux-gnu)
  CPU: AMD EPYC 7502 32-Core Processor
  WORD_SIZE: 64
  LIBM: libopenlibm
  LLVM: libLLVM-11.0.1 (ORCJIT, znver2)
```

Environment:
    JULIA_DEPOT_PATH = /root/.cache/julia-buildkite-plugin/depots/5b300254-1738-4989-ae0a

Package Information:

        Status `/var/lib/buildkite-agent/builds/amdci3-julia-csail-mit-edu/julialang/scir
  [479239e8] Catalyst v6.12.1
  [2b5f629d] DiffEqBase v6.62.2
  [f3b72e0c] DiffEqDevTools v2.27.2
  [7f56f5a3] LSODA v0.7.0
  [961ee093] ModelingToolkit v5.17.3
  [54ca160b] ODEInterface v0.5.0
  [09606e27] ODEInterfaceDiffEq v3.10.0
  [1dea7af3] OrdinaryDiffEq v5.56.0
  [91a5bcdd] Plots v1.15.2
  [b4db0fb7] ReactionNetworkImporters v0.8.0
  [31c91b34] SciMLBenchmarks v0.1.0
  [c3572dad] Sundials v4.4.3
  [a759f4b9] TimerOutputs v0.5.9

And the full manifest:

        Status `/var/lib/buildkite-agent/builds/amdci3-julia-csail-mit-edu/julialang/scir
  [c3fe647b] AbstractAlgebra v0.16.0
  [1520ce14] AbstractTrees v0.3.4
  [79e6a3ab] Adapt v3.3.0
  [ec485272] ArnoldiMethod v0.1.0
  [4fba245c] ArrayInterface v3.1.15
  [9e28174c] BinDeps v1.0.2
  [fa961155] CEnum v0.4.1
  [00ebfdb7] CSTParser v2.5.0
  [479239e8] Catalyst v6.12.1
  [d360d2e6] ChainRulesCore v0.9.44
  [b630d9fa] CheapThreads v0.2.5
  [35d6a980] ColorSchemes v3.12.1
  [3da002f7] ColorTypes v0.11.0
  [5ae59095] Colors v0.12.8
  [861a8166] Combinatorics v1.0.2
  [a80b9123] CommonMark v0.8.1
  [38540f10] CommonSolve v0.2.0
  [bbf7d656] CommonSubexpressions v0.3.0
  [34da2185] Compat v3.30.0
  [8f4d0f93] Conda v1.5.2
  [187b0558] ConstructionBase v1.2.1
  [d38c429a] Contour v0.5.7
  [a8cc5b0e] Crayons v4.0.4
  [9a962f9c] DataAPI v1.6.0

```
[864edb3b] DataStructures v0.18.9
[e2d170a0] DataValueInterfaces v1.0.0
[2b5f629d] DiffEqBase v6.62.2
[f3b72e0c] DiffEqDevTools v2.27.2
[c894b116] DiffEqJump v6.14.2
[77a26b50] DiffEqNoiseProcess v5.7.3
[163ba53b] DiffResults v1.0.3
[b552c78f] DiffRules v1.0.2
[b4f34e82] Distances v0.10.3
[31c24e10] Distributions v0.24.18
[ffbed154] DocStringExtensions v0.8.4
[e30172f5] Documenter v0.26.3
[d4d017d3] ExponentialUtilities v1.8.4
[e2ba6199] ExprTools v0.1.3
[c87230d0] FFMPEG v0.4.0
[7034ab61] FastBroadcast v0.1.8
[9aa1b823] FastClosures v0.3.2
[1a297f60] FillArrays v0.11.7
[6a86dc24] FiniteDiff v2.8.0
[53c48c17] FixedPointNumbers v0.8.4
[59287772] Formatting v0.4.2
[f6369f11] ForwardDiff v0.10.18
[069b7b12] FunctionWrappers v1.1.2
[28b8d3ca] GR v0.57.4
[5c1252a2] GeometryBasics v0.3.12
[d7ba0133] Git v1.2.1
[42e2da0e] Grisu v1.0.2
[cd3eb016] HTTP v0.9.9
[eafb193a] Highlights v0.4.5
[0e44f5e4] Hwloc v2.0.0
[7073ff75] IJulia v1.23.2
[b5f81e59] IOCapture v0.1.1
[615f187c] IfElse v0.1.0
[d25df0c9] Inflate v0.1.2
[83e8ac13] IniFile v0.5.0
[c8e1da08] IterTools v1.3.0
[42fd0dbc] IterativeSolvers v0.9.1
[82899510] IteratorInterfaceExtensions v1.0.0
[692b3bcd] JLLWrappers v1.3.0
[682c06a0] JSON v0.21.1
[98e50ef6] JuliaFormatter v0.13.7
[7f56f5a3] LSODA v0.7.0
[b964fa9f] LaTeXStrings v1.2.1
[2ee39098] LabelledArrays v1.6.1
[23fbe1c1] Latexify v0.15.5
[093fc24a] LightGraphs v1.3.5
[d3d80556] LineSearches v7.1.1
[2ab3a3ac] LogExpFunctions v0.2.4
[bdcacae8] LoopVectorization v0.12.23
```

```
[1914dd2f] MacroTools v0.5.6
[739be429] MbedTLS v1.0.3
[442fdcdd] Measures v0.3.1
[e1d29d7a] Missings v1.0.0
[961ee093] ModelingToolkit v5.17.3
[46d2c3a1] MuladdMacro v0.2.2
[ffc61752] Mustache v1.0.10
[d41bc354] NLSolversBase v7.8.0
[2774e3e8] NLsolve v4.5.1
[77ba4419] NaNMath v0.3.5
[8913a72c] NonlinearSolve v0.3.8
[54ca160b] ODEInterface v0.5.0
[09606e27] ODEInterfaceDiffEq v3.10.0
[6fe1bfb0] OffsetArrays v1.9.0
[429524aa] Optim v1.3.0
[bac558e1] OrderedCollections v1.4.1
[1dea7af3] OrdinaryDiffEq v5.56.0
[90014a1f] PDMats v0.11.0
[d96e819e] Parameters v0.12.2
[69de0a69] Parsers v1.1.0
[ccf2f8ad] PlotThemes v2.0.1
[995b91a9] PlotUtils v1.0.10
[91a5bcdd] Plots v1.15.2
[e409e4f3] PoissonRandom v0.4.0
[f517fe37] Polyester v0.3.1
[85a6dd25] PositiveFactorizations v0.2.4
[21216c6a] Preferences v1.2.2
[1fd47b50] QuadGK v2.4.1
[74087812] Random123 v1.3.1
[fb686558] RandomExtensions v0.4.3
[e6cf234a] RandomNumbers v1.4.0
[b4db0fb7] ReactionNetworkImporters v0.8.0
[3cdcf5f2] RecipesBase v1.1.1
[01d81517] RecipesPipeline v0.3.2
[731186ca] RecursiveArrayTools v2.11.4
[f2c3362d] RecursiveFactorization v0.1.12
[189a3867] Reexport v1.0.0
[ae029012] Requires v1.1.3
[ae5879a3] ResettableStacks v1.1.0
[79098fc4] Rmath v0.7.0
[47965b36] RootedTrees v1.0.0
[7e49a35a] RuntimeGeneratedFunctions v0.5.2
[476501e8] SLEEFPirates v0.6.20
[1bc83da4] SafeTestsets v0.0.1
[0bca4576] SciMLBase v1.13.4
[31c91b34] SciMLBenchmarks v0.1.0
[6c6a2e73] Scratch v1.0.3
[efcf1570] Setfield v0.7.0
[992d4aef] Showoff v1.0.3
```

```
[699a6c99] SimpleTraits v0.9.3
[b85f4697] SoftGlobalScope v1.1.0
[a2af1166] SortingAlgorithms v1.0.0
[47a9eef4] SparseDiffTools v1.13.2
[276daf66] SpecialFunctions v1.4.1
[aedffcd0] Static v0.2.4
[90137ffa] StaticArrays v1.2.0
[82ae8749] StatsAPI v1.0.0
[2913bbd2] StatsBase v0.33.8
[4c63d2b9] StatsFuns v0.9.8
[7792a7ef] StrideArraysCore v0.1.11
[09ab397b] StructArrays v0.5.1
[c3572dad] Sundials v4.4.3
[d1185830] SymbolicUtils v0.11.2
[0c5d862f] Symbolics v0.1.25
[3783bdb8] TableTraits v1.0.1
[bd369af6] Tables v1.4.2
[8290d209] ThreadingUtilities v0.4.4
[a759f4b9] TimerOutputs v0.5.9
[0796e94c] Tokenize v0.5.16
[a2a6695c] TreeViews v0.3.0
[30578b45] URIParser v0.4.1
[5c2747f8] URIs v1.3.0
[3a884ed6] UnPack v1.0.2
[1986cc42] Unitful v1.7.0
[3d5dd08c] VectorizationBase v0.20.11
[81def892] VersionParsing v1.2.0
[19fa3120] VertexSafeGraphs v0.1.2
[44d3d7a6] Weave v0.10.8
[ddb6d928] YAML v0.4.6
[c2297ded] ZMQ v1.2.1
[700de1a5] ZygoteRules v0.2.1
[6e34b625] Bzip2_jll v1.0.6+5
[83423d85] Cairo_jll v1.16.0+6
[5ae413db] EarCut_jll v2.1.5+1
[2e619515] Expat_jll v2.2.10+0
[b22a6f82] FFMPEG_jll v4.3.1+4
[a3f928ae] Fontconfig_jll v2.13.1+14
[d7e528f0] FreeType2_jll v2.10.1+5
[559328eb] FriBidi_jll v1.0.5+6
[0656b61e] GLFW_jll v3.3.4+0
[d2c73de3] GR_jll v0.57.2+0
[78b55507] Gettext_jll v0.20.1+7
[f8c6e375] Git_jll v2.31.0+0
[7746bdde] Glib_jll v2.59.0+4
[e33a78d0] Hwloc_jll v2.4.1+0
[aacddb02] JpegTurbo_jll v2.0.1+3
[c1c5ebd0] LAME_jll v3.100.0+3
[aae0fff6] LSODA_jll v0.1.1+0
```

```
[dd4b983a]  LZO_jll v2.10.1+0
[dd192d2f]  LibVPX_jll v1.9.0+1
[e9f186c6]  Libffi_jll v3.2.2+0
[d4300ac3]  Libgcrypt_jll v1.8.7+0
[7e76a0d4]  Libglvnd_jll v1.3.0+3
[7add5ba3]  Libgpg_error_jll v1.42.0+0
[94ce4f54]  Libiconv_jll v1.16.1+0
[4b2f31a3]  Libmount_jll v2.35.0+0
[89763e89]  Libtiff_jll v4.1.0+2
[38a345b3]  Libuuid_jll v2.36.0+0
[c771fb93]  ODEInterface_jll v0.0.1+0
[e7412a2a]  Ogg_jll v1.3.4+2
[458c3c95]  OpenSSL_jll v1.1.1+6
[efe28fd5]  OpenSpecFun_jll v0.5.4+0
[91d4177d]  Opus_jll v1.3.1+3
[2f80f16e]  PCRE_jll v8.44.0+0
[30392449]  Pixman_jll v0.40.1+0
[ea2cea3b]  Qt5Base_jll v5.15.2+0
[f50d1b31]  Rmath_jll v0.3.0+0
[fb77eaff]  Sundials_jll v5.2.0+1
[a2964d1f]  Wayland_jll v1.17.0+4
[2381bf8a]  Wayland_protocols_jll v1.18.0+4
[02c8fc9c]  XML2_jll v2.9.12+0
[aed1982a]  XSLT_jll v1.1.34+0
[4f6342f7]  Xorg_libX11_jll v1.6.9+4
[0c0b7dd1]  Xorg_libXau_jll v1.0.9+4
[935fb764]  Xorg_libXcursor_jll v1.2.0+4
[a3789734]  Xorg_libXdmcp_jll v1.1.3+4
[1082639a]  Xorg_libXext_jll v1.3.4+4
[d091e8ba]  Xorg_libXfixes_jll v5.0.3+4
[a51aa0fd]  Xorg_libXi_jll v1.7.10+4
[d1454406]  Xorg_libXinerama_jll v1.1.4+4
[ec84b674]  Xorg_libXrandr_jll v1.5.2+4
[ea2f1a96]  Xorg_libXrender_jll v0.9.10+4
[14d82f49]  Xorg_libpthread_stubs_jll v0.1.0+3
[c7cfdc94]  Xorg_libxcb_jll v1.13.0+3
[cc61e674]  Xorg_libxkbfile_jll v1.1.0+4
[12413925]  Xorg_xcb_util_image_jll v0.4.0+1
[2def613f]  Xorg_xcb_util_jll v0.4.0+1
[975044d2]  Xorg_xcb_util_keysyms_jll v0.4.0+1
[0d47668e]  Xorg_xcb_util_renderutil_jll v0.3.9+1
[c22f9ab0]  Xorg_xcb_util_wm_jll v0.4.1+1
[35661453]  Xorg_xkbcomp_jll v1.4.2+4
[33bec58e]  Xorg_xkeyboard_config_jll v2.27.0+4
[c5fb5394]  Xorg_xtrans_jll v1.4.0+3
[8f1865be]  ZeroMQ_jll v4.3.2+6
[3161d3a3]  Zstd_jll v1.5.0+0
[0ac62f75]  libass_jll v0.14.0+4
[f638f0a6]  libfdk_aac_jll v0.1.6+4
```

```
[b53b4c65] libpng_jll v1.6.38+0
[a9144af2] libsodium_jll v1.0.20+0
[f27f6e37] libvorbis_jll v1.3.6+6
[1270edf5] x264_jll v2020.7.14+2
[dfaa095f] x265_jll v3.0.0+3
[d8fb68d0] xkbcommon_jll v0.9.1+5
[0dad84c5] ArgTools
[56f22d72] Artifacts
[2a0f44e3] Base64
[ade2ca70] Dates
[8bb1440f] DelimitedFiles
[8ba89e20] Distributed
[f43a241f] Downloads
[7b1f6079] FileWatching
[9fa8497b] Future
[b77e0a4c] InteractiveUtils
[b27032c2] LibCURL
[76f85450] LibGit2
[8f399da3] Libdl
[37e2e46d] LinearAlgebra
[56ddb016] Logging
[d6f4376e] Markdown
[a63ad114] Mmap
[ca575930] NetworkOptions
[44cfe95a] Pkg
[de0858da] Printf
[3fa0cd96] REPL
[9a3f8284] Random
[ea8e919c] SHA
[9e88b42a] Serialization
[1a1011a3] SharedArrays
[6462fe0b] Sockets
[2f01184e] SparseArrays
[10745b16] Statistics
[4607b0f0] SuiteSparse
[fa267f1f] TOML
[a4e569a6] Tar
[8dfed614] Test
[cf7118a7] UUIDs
[4ec0a83e] Unicode
[e66e0078] CompilerSupportLibraries_jll
[deac9b47] LibCURL_jll
[29816b5a] LibSSH2_jll
[c8ffd9c3] MbedTLS_jll
[14a3606d] MozillaCACerts_jll
[4536629a] OpenBLAS_jll
[efcefdf7] PCRE2_jll
[bea87d4a] SuiteSparse_jll
[83775a58] Zlib_jll
```

```
[8e850ede] nghttp2_jll
[3f19e933] p7zip_jll
```