Mendes Multistate Model

Samuel Isaacson, Chris Rackauckas

July 1, 2020

Taken from Gupta and Mendes, *An Overview of Network-Based and -Free Approaches for Stochastic Simulation of Biochemical Systems*, Computation, 6 (9), 2018.

```
using DiffEqBase, DiffEqBiological, DiffEqJump, DiffEqProblemLibrary.JumpProblemLibrary,
Plots, Statistics
```

```
Error: ArgumentError: Package DiffEqBase not found in current path:
- Run `import Pkg; Pkg.add("DiffEqBase")` to install the DiffEqBase package
.
```

```
gr()
```

```
Error: UndefVarError: gr not defined
```

```
fmt = :png
JumpProblemLibrary.importjumpproblems()
```

```
Error: UndefVarError: JumpProblemLibrary not defined
```

1 Plot solutions by each method

```
methods = (Direct(),DirectFW(),FRM(),FRMFW(),SortingDirect(),NRM(),DirectCR(),RSSA())
```

```
Error: UndefVarError: Direct not defined
```

```
shortlabels = [string(leg)[12:end-2] for leg in methods]
```

```
Error: MethodError: no method matching iterate(::typeof(methods))
Closest candidates are:
  iterate(!Matched::Core.SimpleVector) at essentials.jl:603
  iterate(!Matched::Core.SimpleVector, !Matched::Any) at essentials.jl:603
  iterate(!Matched::ExponentialBackOff) at error.jl:253
  ...
```

```
jprob   = prob_jump_multistate
```

```
Error: UndefVarError: prob_jump_multistate not defined
```

```
tf      = 10.0*jprob.tstop
```

```
Error: UndefVarError: jprob not defined
```

```
prob    = DiscreteProblem(jprob.u0, (0.0,tf), jprob.rates)
```

```
Error: UndefVarError: jprob not defined
```

```
rn      = jprob.network
```

```
Error: UndefVarError: jprob not defined
```

```
varlegs = ["A_P", "A_bound_P", "A_unbound_P", "RLA_P"]
varsyms = [
    [:S7,:S8,:S9],
    [:S9],
    [:S7,:S8],
    [:S7]
]
varidxs = []
for vars in varsyms
    push!(varidxs, [findfirst(isequal(sym),rn.syms) for sym in vars])
end
```

```
Error: UndefVarError: rn not defined
```

```
p = []
for (i,method) in enumerate(methods)
    jump_prob = JumpProblem(prob, method, rn, save_positions=(false,false))
    sol = solve(jump_prob, SSAStepper(), saveat=tf/1000.)
    solv = zeros(1001,4)
    for (i,varidx) in enumerate(varidxs)
        solv[:,i] = sum(sol[varidx,:], dims=1)
    end
    if i < length(methods)
        push!(p, plot(sol.t,solv,title=shortlabels[i],legend=false,format=fmt))
    else
        push!(p,
plot(sol.t,solv,title=shortlabels[i],legend=true,labels=varlegs,format=fmt))
    end
end
```

```
Error: MethodError: no method matching iterate(::typeof(methods))
Closest candidates are:
  iterate(!Matched::Core.SimpleVector) at essentials.jl:603
  iterate(!Matched::Core.SimpleVector, !Matched::Any) at essentials.jl:603
  iterate(!Matched::ExponentialBackOff) at error.jl:253
  ...
```

```
plot(p...,format=fmt)
```

```
Error: UndefVarError: plot not defined
```

2 Benchmarking performance of the methods

```
function run_benchmark!(t, jump_prob, stepper)
    sol = solve(jump_prob, stepper)
    @inbounds for i in 1:length(t)
        t[i] = @elapsed (sol = solve(jump_prob, stepper))
    end
end
```

```
run_benchmark! (generic function with 1 method)
```

```
nsims = 100
benchmarks = Vector{Vector{Float64}}()
for method in methods
    jump_prob = JumpProblem(prob, method, rn, save_positions=(false,false))
    stepper = SSAStepper()
    t = Vector{Float64}(undef, nsims)
    run_benchmark!(t, jump_prob, stepper)
    push!(benchmarks, t)
end
```

```
Error: MethodError: no method matching iterate(::typeof(methods))
Closest candidates are:
  iterate(!Matched::Core.SimpleVector) at essentials.jl:603
  iterate(!Matched::Core.SimpleVector, !Matched::Any) at essentials.jl:603
  iterate(!Matched::ExponentialBackOff) at error.jl:253
  ...
```

```
medtimes = Vector{Float64}(undef,length(methods))
```

```
Error: MethodError: no method matching length(::typeof(methods))
Closest candidates are:
  length(!Matched::Core.SimpleVector) at essentials.jl:596
  length(!Matched::Base.MethodList) at reflection.jl:852
  length(!Matched::Core.MethodTable) at reflection.jl:938
  ...
```

```
stdtimes = Vector{Float64}(undef,length(methods))
```

```
Error: MethodError: no method matching length(::typeof(methods))
Closest candidates are:
  length(!Matched::Core.SimpleVector) at essentials.jl:596
  length(!Matched::Base.MethodList) at reflection.jl:852
  length(!Matched::Core.MethodTable) at reflection.jl:938
  ...
```

```
avgtimes = Vector{Float64}(undef,length(methods))
```

```
Error: MethodError: no method matching length(::typeof(methods))
Closest candidates are:
  length(!Matched::Core.SimpleVector) at essentials.jl:596
  length(!Matched::Base.MethodList) at reflection.jl:852
  length(!Matched::Core.MethodTable) at reflection.jl:938
  ...
```

```
for i in 1:length(methods)
    medtimes[i] = median(benchmarks[i])
    avgtimes[i] = mean(benchmarks[i])
    stdtimes[i] = std(benchmarks[i])
end
```

```
Error: MethodError: no method matching length(::typeof(methods))
Closest candidates are:
  length(!Matched::Core.SimpleVector) at essentials.jl:596
  length(!Matched::Base.MethodList) at reflection.jl:852
  length(!Matched::Core.MethodTable) at reflection.jl:938
  ...
```

```
using DataFrames
```

```
Error: ArgumentError: Package DataFrames not found in current path:
- Run `import Pkg; Pkg.add("DataFrames")` to install the DataFrames package
.
```

```
df =
DataFrame(names=shortlabels,medtimes=medtimes,relmedtimes=(medtimes/medtimes[1]),avgtimes=avgtimes,
std=stdtimes, cv=stdtimes./avgtimes)
```

```
Error: UndefVarError: medtimes not defined
```

```
sa = [text(string(round(mt,digits=3),"s"),:center,12) for mt in df.medtimes]
```

```
Error: UndefVarError: df not defined
```

```
bar(df.names,df.relmedtimes,legend=:false, fmt=fmt)
```

```
Error: UndefVarError: bar not defined
```

```
scatter!(df.names, .05 .+ df.relmedtimes, markeralpha=0, series_annotations=sa, fmt=fmt)
```

```
Error: UndefVarError: df not defined
```

```
ylabel!("median relative to Direct")
```

```
Error: UndefVarError: ylabel! not defined
```

```
title!("Multistate Model")
```

```
Error: UndefVarError: title! not defined
```

```
using DiffEqBenchmarks
DiffEqBenchmarks.bench_footer(WEAVE_ARGS[:folder],WEAVE_ARGS[:file])
```

2.1 Appendix

These benchmarks are a part of the DiffEqBenchmarks.jl repository, found at: https://github.com/JuliaDi

To locally run this tutorial, do the following commands:

```
using DiffEqBenchmarks
DiffEqBenchmarks.weave_file("Jumps","Mendes_multistate_example.jmd")
```

Computer Information:

```
Julia Version 1.4.2
Commit 44fa15b150* (2020-05-23 18:35 UTC)
Platform Info:
  OS: Linux (x86_64-pc-linux-gnu)
  CPU: Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz
  WORD_SIZE: 64
  LIBM: libopenlibm
  LLVM: libLLVM-8.0.1 (ORCJIT, skylake)
Environment:
  JULIA_DEPOT_PATH = /builds/JuliaGPU/DiffEqBenchmarks.jl/.julia
  JULIA_CUDA_MEMORY_LIMIT = 536870912
  JULIA_PROJECT = @.
  JULIA_NUM_THREADS = 4
```

Package Information:

```
Status: `/builds/JuliaGPU/DiffEqBenchmarks.jl/Project.toml`
[7073ff75-c697-5162-941a-fcdaad2a7d2a] IJulia 1.21.2
[44d3d7a6-8a23-5bf8-98c5-b353f8df5ec9] Weave 0.10.2
[b77e0a4c-d291-57a0-90e8-8db25a27a240] InteractiveUtils
[d6f4376e-aef5-505a-96c1-9c027394607a] Markdown
[44cfe95a-1eb2-52ea-b672-e2afdf69b78f] Pkg
```