Stochastic SIR in continuous time using Gillespie.jl

Simon Frost

April 27, 2020

```julia
using Gillespie
using Random
using Plots
using BenchmarkTools

function sir_rates(x,parms)
  (S,I,R) = x
  (β,γ) = parms
  N = S+I+R
  infection = β*S*I/N
  recovery = γ*I
  [infection,recovery]
end
sir_transitions = [[-1 1 0];[0 -1 1]]
```

```
2×3 Array{Int64,2}:
 -1   1  0
  0  -1  1
```

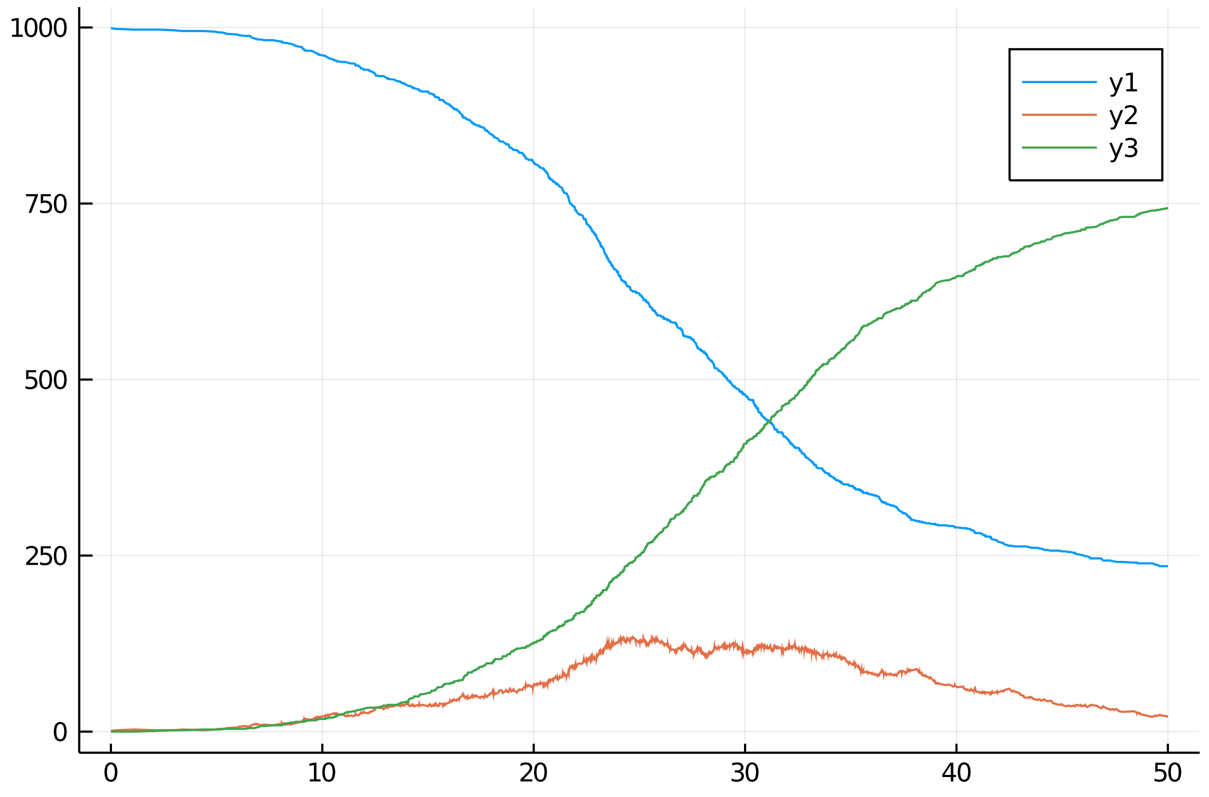
```julia
u0 = [999,1,0]
p = [0.5,0.25]
Random.seed!(1235)
tf = 50.0
```

```
50.0
```

```julia
sir_result = ssa(u0,sir_rates,sir_transitions,p,tf)
data = ssa_data(sir_result)
```

	time	x1	x2	x3
	Float64	Int64	Int64	Int64
1	0.0	999	1	0
2	0.236916	998	2	0
3	1.07532	997	3	0
4	2.16728	997	2	1
5	2.25891	997	1	2
6	2.96813	996	2	2
7	3.38675	995	3	2
8	4.38327	995	2	3
9	4.92125	994	3	3
10	5.10805	993	4	3
11	5.21266	993	3	4
12	5.40139	992	4	4
13	5.46484	991	5	4
14	5.98918	990	6	4
15	6.05614	989	7	4
16	6.33437	988	8	4
17	6.58878	988	7	5
18	6.60951	987	8	5
19	6.63446	986	9	5
20	6.75001	985	10	5
21	6.80489	984	11	5
22	6.8854	984	10	6
23	6.92483	983	11	6
24	7.01135	983	10	7
25	7.10276	983	9	8
26	7.22309	982	10	8
27	7.71406	982	9	9
28	7.8201	981	10	9
29	8.08454	980	11	9
30	8.08558	979	12	9
31	8.1044	979	11	10
32	8.23118	978	12	10
33	8.30206	978	11	11
34	8.33912	978	10	12
35	8.42502	977	11	12
36	8.49962	977	10	13
37	8.58602	976	11	13
38	8.69091	975	12	13
39	8.70621	974	13	13
40	8.74195	974	12	14
41	8.85186	973	13	14
42	9.0465	972	14	14
43	9.0628	971	15	14
44	9.0656	970	16	14
45	9.1423	969	17	14
46	9.14598	968	18	14
47	9.16416	968	17	15
48	9.19151	967	18	15
49	9.42744	967	17	16
50	9.57617	966	18	16
51	9.6132	965	19	16
52	9.68246	964	20	16

```
plot(data[:,1],data[:,2])
plot!(data[:,1],data[:,3])
plot!(data[:,1],data[:,4])
```



```
@benchmark ssa(u0,sir_rates,sir_transitions,p,tf)
```

```
BenchmarkTools.Trial:
  memory estimate:  1.13 KiB
  allocs estimate:  18
  --------------
  minimum time:     399.000 ns (0.00% GC)
  median time:      18.000 μs (0.00% GC)
  mean time:        116.258 μs (12.00% GC)
  maximum time:     11.313 ms (97.84% GC)
  --------------
  samples:          10000
  evals/sample:     1
```

0.1 Appendix

Computer Information:

```
Julia Version 1.4.0
Commit b8e9a9ecc6 (2020-03-21 16:36 UTC)
Platform Info:
  OS: Windows (x86_64-w64-mingw32)
  CPU: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz
  WORD_SIZE: 64
  LIBM: libopenlibm
  LLVM: libLLVM-8.0.1 (ORCJIT, skylake)
Environment:
  JULIA_EDITOR = "C:\Users\sdwfr\AppData\Local\atom\app-1.45.0\atom.exe"  -a
  JULIA_NUM_THREADS = 4
```

Package Information:

```
Status `~\.julia\environments\v1.4\Project.toml`
[46ada45e-f475-11e8-01d0-f70cc89e6671] Agents 3.0.0
[b19378d9-d87a-599a-927f-45f220a2c452] ArrayFire 1.0.6
[c52e3926-4ff0-5f6e-af25-54175e0327b1] Atom 0.12.10
[6e4b80f9-dd63-53aa-95a3-0cdb28fa8baf] BenchmarkTools 0.5.0
[be33ccc6-a3ff-5ff2-a52e-74243cff1e17] CUDAnative 3.0.4
[3a865a2d-5b23-5a0f-bc46-62713ec82fae] CuArrays 2.0.1
[717857b8-e6f2-59f4-9121-6e50c889abd2] DSP 0.6.6
[2445eb08-9709-466a-b3fc-47e12bd697a2] DataDrivenDiffEq 0.2.0
[a93c6f00-e57d-5684-b7b6-d8193f3e46c0] DataFrames 0.20.2
[aae7a2af-3d4f-5e19-a356-7da93b79d9d0] DiffEqFlux 1.8.1
[41bf760c-e81c-5289-8e54-58b1f1f8abe2] DiffEqSensitivity 6.13.0
[6d1b261a-3be8-11e9-3f2f-0b112a9a8436] DiffEqTutorials 0.1.0
[0c46a032-eb83-5123-abaf-570d42b7fbaa] DifferentialEquations 6.13.0
[31c24e10-a181-5473-b8eb-7969acd0382f] Distributions 0.23.2
[634d3b9d-ee7a-5ddf-bec9-22491ea816e1] DrWatson 1.10.2
[587475ba-b771-5e3f-ad9e-33799f191a9c] Flux 0.10.4
[0c68f7d7-f131-5f86-a1c3-88cf8149b2d7] GPUArrays 3.1.0
[28b8d3ca-fb5f-59d9-8090-bfdbd6d07a71] GR 0.48.0
[523d8e89-b243-5607-941c-87d699ea6713] Gillespie 0.1.0
[7073ff75-c697-5162-941a-fcdaad2a7d2a] IJulia 1.21.2
[e5e0dc1b-0480-54bc-9374-aad01c23163d] Juno 0.8.1
[961ee093-0014-501f-94e3-6117800e7a78] ModelingToolkit 3.0.2
[429524aa-4258-5aef-a3af-852621145aeb] Optim 0.20.6
[1dea7af3-3e70-54e6-95c3-0bf5283fa5ed] OrdinaryDiffEq 5.34.1
[91a5bcdd-55d7-5caf-9e0b-520d859cae80] Plots 1.0.12
[e6cf234a-135c-5ec9-84dd-332b85af5143] RandomNumbers 1.4.0
[c5292f4c-5179-55e1-98c5-05642aab7184] ResumableFunctions 0.5.1
[428bdadb-6287-5aa5-874b-9969638295fd] SimJulia 0.8.0
[05bca326-078c-5bf0-a5bf-ce7c7982d7fd] SimpleDiffEq 1.1.0
[f3b207a7-027a-5e70-b257-86293d7955fd] StatsPlots 0.14.5
[789caeaf-c7a9-5a7d-9973-96adeb23e2a0] StochasticDiffEq 6.19.2
[44d3d7a6-8a23-5bf8-98c5-b353f8df5ec9] Weave 0.9.4
[37e2e46d-f89d-539d-b4ee-838fcccc9c8e] LinearAlgebra
[cf7118a7-6976-5b1a-9a39-7adc72f591a4] UUIDs
```