Integrated-cell

February 4, 2025

[1]:
```
function virtual_cell!(du, u, p, t)
    # Molecular Scale (Enzyme Kinetics)
    S, E, P = u[1], u[2], u[3]
    reaction_rate = (k_cat * S * E) / (Km + S)

    du[1] = -reaction_rate
    du[2] = 0  # Enzyme is constant
    du[3] = reaction_rate

    # Organelle Scale (ATP Synthesis)
    ADP, ATP = u[4], u[5]
    synthesis_rate = (Vmax * ADP) / (K_ATP + ADP)

    du[4] = -synthesis_rate
    du[5] = synthesis_rate

    # Glycolysis
    Glucose, Pyruvate = u[6], u[7]
    k1, k2 = 0.8, 0.5

    du[6] = -k1 * Glucose
    du[7] = k2 * Glucose

    # Cellular Signaling
    GF, Receptor, Response = u[8], u[9], u[10]
    k_bind, k_signal = 0.6, 0.8

    du[8] = -k_bind * GF * Receptor
    du[9] = -k_bind * GF * Receptor
    du[10] = k_signal * GF * Receptor

    # Cell Cycle (G1, S, G2, M)
    G1, S_phase, G2, M_phase = u[11], u[12], u[13], u[14]
    k_G1, k_S, k_G2, k_M = 0.5, 0.3, 0.2, 0.1

    du[11] = -k_G1 * G1
    du[12] = k_G1 * G1 - k_S * S_phase
```

```
    du[13] = k_S * S_phase - k_G2 * G2
    du[14] = k_G2 * G2 - k_M * M_phase
end

# Initial conditions for all variables
u0 = [1.0, 0.1, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0, 1.0, 0.0, 1.0, 0.0, 0.0, 0.0]
tspan = (0.0, 20.0)

prob = ODEProblem(virtual_cell!, u0, tspan)
sol = solve(prob, Tsit5())

plot(sol, title="Integrated Virtual Cell Simulation")
```

```
UndefVarError: `ODEProblem` not defined in `Main`
Suggestion: check for spelling errors or missing imports.

Stacktrace:
 [1] top-level scope
   @ In[1]:46
```

[2]:
```
using Flux

# Generate training data from ODE simulation
X_train = [sol.t]  # Time as input
Y_train = [sol.u]  # State variables as output

# Define a simple neural network to predict cell state
model = Chain(
    Dense(1, 32, relu),
    Dense(32, 32, relu),
    Dense(32, length(u0))  # Output size matches number of cell variables
)

loss(x, y) = Flux.mse(model(x), y)
opt = Adam(0.01)

# Training loop
for epoch in 1:100
    Flux.train!(loss, Flux.params(model), [(X_train, Y_train)], opt)
end

# Predict cell state under new conditions
X_test = [15.0]  # Example: Predict state at t=15
predicted_state = model(X_test)

println("Predicted cell state at t=15:", predicted_state)
```

```
UndefVarError: `sol` not defined in `Main`
Suggestion: check for spelling errors or missing imports.

Stacktrace:
 [1] top-level scope
   @ In[2]:4
```

[3]:
```julia
using DiffEqJump

# Define stochastic reaction system
function stochastic_cell!(du, u, p, t)
    du[1] = -0.5 * u[1] + randn() * 0.1  # Random substrate fluctuations
    du[2] = 0.5 * u[1] - randn() * 0.1  # Random enzyme fluctuations
    du[3] = randn() * 0.05  # Product stochasticity
end

u0 = [1.0, 0.1, 0.0]
tspan = (0.0, 10.0)

prob = SDEProblem(stochastic_cell!, u0, tspan)
sol = solve(prob, EM())

plot(sol, labels=["Substrate" "Enzyme" "Product"], title="Stochastic Virtual␣
 ↪Cell")
```

```
[ Info: Precompiling DiffEqJump
[c894b116-72e5-5b58-be3c-e6d8d4ac2b12]
WARNING: Method definition rrule(Type{SciMLBase.ODEProblem{uType, tType,
isinplace, P, F, K, PT} where PT where K where F where P where isinplace where
tType where uType}, Any…) in module SciMLBaseChainRulesCoreExt at /Users/kevin
/.julia/packages/SciMLBase/8XHkk/ext/SciMLBaseChainRulesCoreExt.jl:80
overwritten in module DiffEqBase at
/Users/kevin/.julia/packages/DiffEqBase/H7x4Q/src/chainrules.jl:1.
ERROR: Method overwriting is not permitted during Module precompilation. Use
`__precompile__(false)` to opt-out of precompilation.
  Info: Skipping precompilation due to
precompilable error. Importing DiffEqJump
[c894b116-72e5-5b58-be3c-e6d8d4ac2b12].
    exception = Error when precompiling module, potentially
caused by a __precompile__(false) declaration in the module.
[ Info: Precompiling DiffEqBase
[2b5f629d-d688-5b77-993f-72d75c75574e] (cache misses: dep missing source (2))
WARNING: Method definition rrule(Type{SciMLBase.ODEProblem{uType, tType,
isinplace, P, F, K, PT} where PT where K where F where P where isinplace where
tType where uType}, Any…) in module SciMLBaseChainRulesCoreExt at /Users/kevin
/.julia/packages/SciMLBase/8XHkk/ext/SciMLBaseChainRulesCoreExt.jl:80
```

```
overwritten in module DiffEqBase at
/Users/kevin/.julia/packages/DiffEqBase/H7x4Q/src/chainrules.jl:1.
ERROR: Method overwriting is not permitted during Module precompilation. Use
`__precompile__(false)` to opt-out of precompilation.
 Info: Skipping precompilation due to
precompilable error. Importing DiffEqBase
[2b5f629d-d688-5b77-993f-72d75c75574e].
    exception = Error when precompiling module, potentially
caused by a __precompile__(false) declaration in the module.
[ Info: Precompiling DiffEqBaseZygoteExt
[0f3e1095-8eee-5b88-8671-b96e40350fa1]
 Warning: Module DiffEqBase with build ID
ffffffff-ffff-ffff-0001-42fa7369b62b is missing from the cache.
 This may mean DiffEqBase
[2b5f629d-d688-5b77-993f-72d75c75574e] does not support precompilation but is
imported by a module that does.
 @ Base loading.jl:2541
 Info: Skipping precompilation due to
precompilable error. Importing DiffEqBaseZygoteExt
[0f3e1095-8eee-5b88-8671-b96e40350fa1].
    exception = Error when precompiling module, potentially
caused by a __precompile__(false) declaration in the module.
WARNING: method definition for #MassActionJump#8 at
/Users/kevin/.julia/packages/DiffEqJump/zMznO/src/jumps.jl:234 declares type
variable U but does not use it.
WARNING: method definition for #MassActionJump#8 at
/Users/kevin/.julia/packages/DiffEqJump/zMznO/src/jumps.jl:234 declares type
variable S but does not use it.
WARNING: method definition for #MassActionJump#9 at
/Users/kevin/.julia/packages/DiffEqJump/zMznO/src/jumps.jl:235 declares type
variable U but does not use it.
WARNING: method definition for #MassActionJump#9 at
/Users/kevin/.julia/packages/DiffEqJump/zMznO/src/jumps.jl:235 declares type
variable S but does not use it.
WARNING: method definition for #RDirectJumpAggregation#62 at
/Users/kevin/.julia/packages/DiffEqJump/zMznO/src/aggregators/rdirect.jl:23
declares type variable DEPGR but does not use it.
```

```
MethodError: no method matching SDEProblem(::typeof(stochastic_cell!), ::
 ↪Vector{Float64}, ::Tuple{Float64, Float64})
The type `SDEProblem` exists, but no method is defined for this combination of␣
 ↪argument types when trying to construct it.

Closest candidates are:
  SDEProblem(::Any, ::Any, ::Any, ::Any, ::Any; kwargs…)
   @ SciMLBase ~/.julia/packages/SciMLBase/8XHkk/src/problems/sde_problems.jl:1 5
  SDEProblem(::Any, ::Any, ::Any, ::Any; …)
```

```
   @ SciMLBase ~/.julia/packages/SciMLBase/8XHkk/src/problems/sde_problems.jl:15
  SDEProblem(::SciMLBase.AbstractSDEFunction, ::Any, ::Any; …)
   @ SciMLBase ~/.julia/packages/SciMLBase/8XHkk/src/problems/sde_problems.jl:11
  …


Stacktrace:
 [1] top-level scope
   @ In[3]:13
```

[ ]: