

Integrated-cell

February 5, 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)

```

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

```
Stacktrace:
 [1] macro expansion
   @ ./loading.jl:2296 [inlined]
 [2] macro expansion
   @ ./lock.jl:273 [inlined]
 [3] __require(into::Module, mod::Symbol)
   @ Base ./loading.jl:2271
 [4] #invoke_in_world#3
   @ ./essentials.jl:1089 [inlined]
 [5] invoke_in_world
   @ ./essentials.jl:1086 [inlined]
 [6] require(into::Module, mod::Symbol)
   @ Base ./loading.jl:2260
```

```
[3]: 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")
```

```
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:1:5
```

```
SDEProblem(::SciMLBase.AbstractSDEFunction, ::Any, ::Any; ...)  
  @ SciMLBase ~/.julia/packages/SciMLBase/8XHkk/src/problems/sde_problems.jl:1:1  
...
```

Stacktrace:

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

[]: