Simulating kinetics of protocell type particles

Abhishek Sharma
BioMIP,
Ruhr Universität, Bochum
Germany

@ Truce Summer School, 2014

References

1) Minimal Replicator Theory I: Parabolic Versus Exponential Growth
Günter von Kiedrowski,
Bioorganic Chemistry Frontiers Volume 3, 1993, pp 113-146

2) PACE project report, http://www.istpace.org/Web_Final_Report/scientific_meetings_at_eclt/workshops/first_year_nov_2004_-_march/protocells_experiments_ethi.html

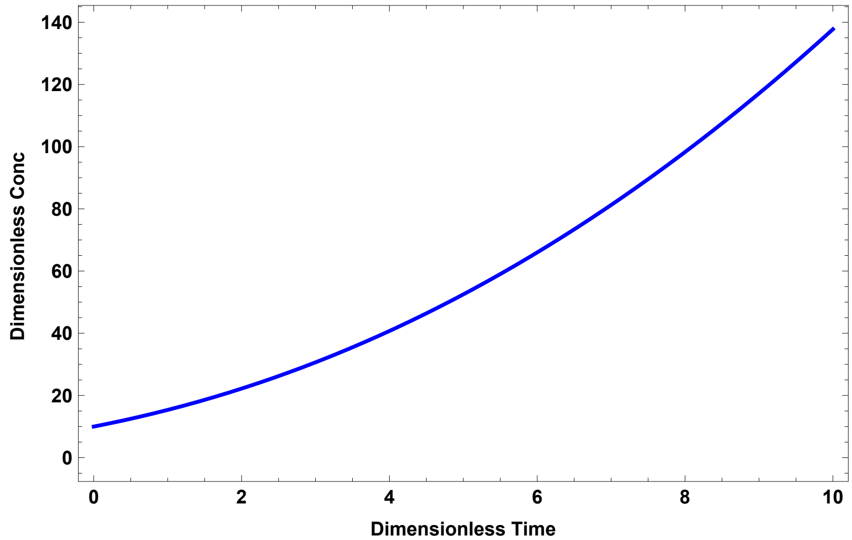
Self replicating chemistry using single reactant

In[1]:= $\psi[u\_] := \frac{2}{u}\left(\sqrt{1+u} - 1\right)$

```
In[2]:= eq[α_] := c'[t] == α c[t] ψ[c[t]]

In[3]:= ic = c[0] == 10;

In[4]:= sol = NDSolve[{eq[1], ic}, c[t], {t, 0, 10}]

Out[4]= {{c[t] → InterpolatingFunction[ Domain: {{0., 10.}} Output: scalar ][t]}}
```

```
In[5]:= Plot[c[t] /. sol, {t, 0, 10}, PlotStyle → {Blue, Thick}, Frame → True,
         FrameLabel → {"Dimensionless Time", "Dimensionless Conc"},
         BaseStyle → {FontWeight → "Bold", FontSize → 10}]
```

Out[5]=



Self replicating chemistry for protocells

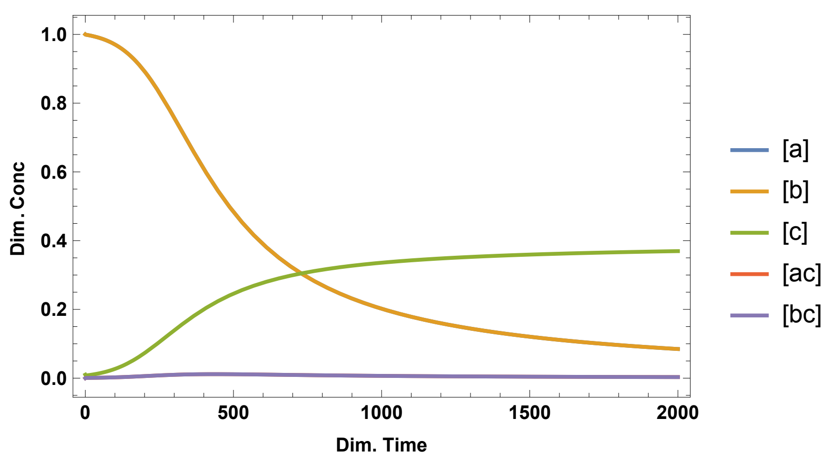
```
In[6]:= eq1[{Ka1_, Ka2_, Kb1_, Kb2_, Kd1_, Kd2_}] :=
         c'[t] == Ka1 ac[t] - Ka2 a[t] c[t] + Kb1 bc[t] - Kb2 b[t] c[t] + 2 Kd2 c2[t] - 2 Kd1 (c[t])^2

In[7]:= eq2[Ka1_, Ka2_, ap_] := ac'[t] == Ka2 a[t] c[t] - Ka1 ac[t] - ap ac[t] b[t];

In[8]:= eq3[Kb2_, Kb1_, app_] := bc'[t] == Kb2 b[t] c[t] - Kb1 bc[t] - app bc[t] a[t];

In[9]:= eq4[f1_, f2_, ap_, app_] :=
          c2s'[t] == f1 c2[t] - f2 c2s[t] + ap ac[t] b[t] + app bc[t] a[t];

In[10]:= eq5[f1_, f2_, Kd1_, Kd2_] := c2'[t] == f2 c2s[t] - f1 c2[t] - Kd2 c2[t] + Kd1 c[t]^2

In[11]:= eq6[Ka1_, Ka2_, app_] := a'[t] == Ka1 ac[t] - Ka2 a[t] c[t] - app bc[t] a[t];

In[12]:= eq7[Kb1_, Kb2_, app_] := b'[t] == Kb1 bc[t] - Kb2 b[t] c[t] - app ac[t] b[t];

In[13]:= eqns[{Ka1_, Ka2_, Kb1_, Kb2_, Kd1_, Kd2_, f1_, f2_, ap_, app_}] :=
          {eq1[{Ka1, Ka2, Kb1, Kb2, Kd1, Kd2}], eq2[Ka1, Ka2, ap], eq3[Kb2, Kb1, app],
           eq4[f1, f2, ap, app], eq5[f1, f2, Kd1, Kd2], eq6[Ka1, Ka2, app], eq7[Kb1, Kb2, app]}

In[14]:= iCons :=
           {a[0] == 1, b[0] == 1, c[0] == 0.01, c2[0] == 0, c2s[0] == 0, ac[0] == 0, bc[0] == 0};

In[15]:= totEqns[{Ka1_, Ka2_, Kb1_, Kb2_, Kd1_, Kd2_, f1_, f2_, ap_, app_}] :=
          {eqns[{Ka1, Ka2, Kb1, Kb2, Kd1, Kd2, f1, f2, ap, app}], iCons}
```

In[16]:=
```
sol = NDSolve[totEqns[{1, 0.1, 1, 0.1, 1, 1, 1, 1, 0.1, 0.1}],
  {a[t], b[t], c[t], c2[t], c2s[t], ac[t], bc[t]}, {t, 0, 2000}]
```

Out[16]=
```
{{a[t] → InterpolatingFunction[ Domain: {{0., 2.00×10^3}} Output: scalar ][t],
  b[t] → InterpolatingFunction[ Domain: {{0., 2.00×10^3}} Output: scalar ][t],
  c[t] → InterpolatingFunction[ Domain: {{0., 2.00×10^3}} Output: scalar ][t],
  c2[t] → InterpolatingFunction[ Domain: {{0., 2.00×10^3}} Output: scalar ][t],
  c2s[t] → InterpolatingFunction[ Domain: {{0., 2.00×10^3}} Output: scalar ][t],
  ac[t] → InterpolatingFunction[ Domain: {{0., 2.00×10^3}} Output: scalar ][t],
  bc[t] → InterpolatingFunction[ Domain: {{0., 2.00×10^3}} Output: scalar ][t]}}
```

In[17]:=
```
Plot[Evaluate[{a[t], b[t], c[t], ac[t], bc[t]} /. sol], {t, 0, 2000},
 PlotStyle → Thick, PlotLegends → {"[a]", "[b]", "[c]", "[ac]", "[bc]"},
 Frame → True, PlotRange → Full, FrameLabel → {"Dim. Time", "Dim. Conc"},
 BaseStyle → {FontWeight → "Bold", FontSize → 10}]
```

Out[17]=



Coupling between two protocells

Consider two particles, which were initially spatially separated. After doing sopme random walks, they came close to each other and start interacting with each other. Due to the presence of two different compartments, the rate equations needs to be modified with a transport term, so and to be solved together so that we can keep track of the concentrations of various species in different compartments.

So, the governing equations in the first compartment is given by :
(The species present in this compartment are “a”, “b”)

```
In[18]:= eqC11[ϕ_] := aC1'[t] == -ϕ (aC1[t] - aC2[t])

In[19]:= eqC12[ϕ_] := bC1'[t] == -ϕ (bC1[t] - bC2[t])
```

The equations for the second compartment are given by :
(The species present in this compartment are “a”, “b”, “c”, “c2”,“ac”, “bc”,”cs”)

```
In[20]:= eqC21[{Ka1_, Ka2_, Kb1_, Kb2_, Kd1_, Kd2_}] := c'[t] ==
           Ka1 ac[t] - Ka2 aC2[t] c[t] + Kb1 bc[t] - Kb2 bC2[t] c[t] + 2 Kd2 c2[t] - 2 Kd1 (c[t])^2

In[21]:= eqC22[Ka1_, Ka2_, ap_] := ac'[t] == Ka2 aC2[t] c[t] - Ka1 ac[t] - ap ac[t] bC2[t];

In[22]:= eqC23[Kb2_, Kb1_, app_] := bc'[t] == Kb2 bC2[t] c[t] - Kb1 bc[t] - app bc[t] aC2[t];

In[23]:= eqC24[f1_, f2_, ap_, app_] :=
           c2s'[t] == f1 c2[t] - f2 c2s[t] + ap ac[t] bC2[t] + app bc[t] aC2[t];

In[24]:= eqC25[f1_, f2_, Kd1_, Kd2_] := c2'[t] == f2 c2s[t] - f1 c2[t] - Kd2 c2[t] + Kd1 c[t]^2

In[25]:= eqC26[ϕ_, Ka1_, Ka2_, app_] :=
           aC2'[t] == Ka1 ac[t] - Ka2 aC2[t] c[t] - app bc[t] aC2[t] + ϕ (aC1[t] - aC2[t]);

In[26]:= eqC27[ϕ_, Kb1_, Kb2_, app_] :=
           bC2'[t] == Kb1 bc[t] - Kb2 bC2[t] c[t] - app ac[t] bC2[t] + ϕ (aC1[t] - aC2[t]);

In[27]:= eqns[{ϕ_, Ka1_, Ka2_, Kb1_, Kb2_, Kd1_, Kd2_, f1_, f2_, ap_, app_}] :=
          {eqC11[ϕ], eqC12[ϕ], eqC21[{Ka1, Ka2, Kb1, Kb2, Kd1, Kd2}],
           eqC22[Ka1, Ka2, ap], eqC23[Kb2, Kb1, app], eqC24[f1, f2, ap, app],
           eqC25[f1, f2, Kd1, Kd2], eqC26[ϕ, Ka1, Ka2, app], eqC27[ϕ, Kb1, Kb2, app]}

In[28]:= iCons := {aC1[0] == 1, bC1[0] == 1, aC2[0] == 0, bC2[0] == 0,
            c[0] == 0.01, c2[0] == 0, c2s[0] == 0, ac[0] == 0, bc[0] == 0};

In[29]:= totEqns[{ϕ_, Ka1_, Ka2_, Kb1_, Kb2_, Kd1_, Kd2_, f1_, f2_, ap_, app_}] :=
          {eqns[{ϕ, Ka1, Ka2, Kb1, Kb2, Kd1, Kd2, f1, f2, ap, app}], iCons}

In[30]:= totEqns[{ϕ, Ka1, Ka2, Kb1, Kb2, Kd1, Kd2, f1, f2, ap, app}] // Flatten // MatrixForm;
```

In[31]:=
```
sol = NDSolve[totEqns[{0.1, 1, 0.1, 1, 0.1, 1, 1, 1, 1, 0.1, 0.1}],
  {aC1[t], bC1[t], aC2[t], bC2[t], c[t], c2[t], c2s[t], ac[t], bc[t]}, {t, 0, 5000}]
```

Out[31]=
```
{{aC1[t] → InterpolatingFunction[ Domain: {{0., 5.00×10^3}} Output: scalar ][t],
  bC1[t] → InterpolatingFunction[ Domain: {{0., 5.00×10^3}} Output: scalar ][t],
  aC2[t] → InterpolatingFunction[ Domain: {{0., 5.00×10^3}} Output: scalar ][t],
  bC2[t] → InterpolatingFunction[ Domain: {{0., 5.00×10^3}} Output: scalar ][t],
  c[t] → InterpolatingFunction[ Domain: {{0., 5.00×10^3}} Output: scalar ][t],
  c2[t] → InterpolatingFunction[ Domain: {{0., 5.00×10^3}} Output: scalar ][t],
  c2s[t] → InterpolatingFunction[ Domain: {{0., 5.00×10^3}} Output: scalar ][t],
  ac[t] → InterpolatingFunction[ Domain: {{0., 5.00×10^3}} Output: scalar ][t],
  bc[t] → InterpolatingFunction[ Domain: {{0., 5.00×10^3}} Output: scalar ][t]}}
```

In[32]:=
```
p1 = Plot[Evaluate[{aC1[t], bC1[t]} /. sol], {t, 0, 5000},
   PlotStyle → Thick, PlotLegends → {"[a]", "[b]", "[c]", "[ac]", "[bc]"},
   Frame → True, PlotRange → Full, FrameLabel → {"Dim. Time", "Dim. Conc"},
   BaseStyle → {FontWeight → "Bold", FontSize → 10}, PlotLabel → "Compartment 1"];
```

In[33]:=
```
p2 = Plot[Evaluate[{aC2[t], bC2[t], c[t], ac[t], bc[t]} /. sol], {t, 0, 5000},
   PlotStyle → Thick, PlotLegends → {"[a]", "[b]", "[c]", "[ac]", "[bc]"},
   Frame → True, PlotRange → Full, FrameLabel → {"Dim. Time", "Dim. Conc"},
   BaseStyle → {FontWeight → "Bold", FontSize → 10}, PlotLabel -> "Compartment 2"];
```

In[34]:=
```
GraphicsRow[{p1, p2}, Frame → All]
```

Out[34]=