Model of N-dependent competition between thermal tolerant and intolerant strains

Developed for Aranguren-Gassis et al. 2019 by CT Kremer and CA Klausmeier.

Initial reflections on parameter values and constraints:

Obtain & load EcoEvo package

Note: this Mathematica package was developed by CA Klausmeier. Newer versions are now available, with streamlined syntax and other improvements. However, the code below was developed using an older version of the package (0.9.3)

```
(* FIRST TIME ONLY: to install the relevant version of this package, you need to run the following line  *)
(*PacletInstall["https://github.com/cklausme/EcoEvo/releases/download/v0.9.3/EcoEvo-0.9.3.paclet"]*)
```

```
In[1]:= (* To load the package: *)
        << EcoEvo`
```

```
EcoEvo Package Version 0.9.3β (September 16, 2018)
```

Set up model

Define core equations

```
In[2]:= b[p_, r1_, r2_, T_] := b1 E^(b2 T) Min[v1 / (q0 + q1 p) r1 / (r1 + k1), v2 / q2 r2 / (r2 + k2)];
        d[p_, T_] := d1 E^(d2 T) / p;

        dn[i_] := (b[p_i, r1[t], r2[t], T] - d[p_i, T]) * n_i[t];

        dr1 := -Sum[b[p_i, r1[t], r2[t], T] * (q0 + q1 p_i) * n_i[t], {i, Nsp[1]}];
        dr2 := -Sum[b[p_i, r1[t], r2[t], T] * (q2) * n_i[t], {i, Nsp[1]}];
```

Define model for EcoEvo package

```
In[7]:= SetModel[{ModelType → "ContinuousTime", Period → τ,
           Aux[1] → {Variable → r1, Equation :→ dr1},
           Aux[2] → {Variable → r2, Equation :→ dr2},
           Guild[1] → {
              Component[1] → {Variable → n, Equation → dn},
              Trait[1] → {Variable → p, Range → Interval[{0, ∞}]}
            },
           WhenEvents → {WhenEvent[Mod[t, τ],
               {ntot = Sum[n_i[t], {i, Nsp[1]}], r1[t] → rin1, r2[t] → rin2, n_1[t] → n0 * n_1[t] / ntot, n_2[t] → n0 * n_2[t] / ntot}]}
         }];
```

Set up model parameters

See details in ***Supporting information: Appendix 1*** and R script ***TPC_fitting_for_mma_model_unweighted_final_041818.R***

```
In[8]:= (* estimated parameters *)
        α = 0.31673;
        δ1 = 1.9926 * 10 ^ - 8;
        δ2 = 7.6312 * 10 ^ - 9;
        b2 = 0.04854806212876125`;
        d2 = 0.5426802449468293`;

        (* assumptions *)
        b1 = 5 * 10 ^ - 1;
        p1g = 1; (* unprotected strain *)
        v1 = 15;
        q0 = 0.5;
        q1 = 5.25; (* controls cost of protection! *)
        q2 = 2;
        k1 = 6; k2 = 10;
        n0 = rin1 / 10;
        τ = 3;

        (* implications *)
        d1 = δ1;
        p2g = δ1 / δ2;(* protected strain *)
        v2 = α * q2 / b1;
```

```
In[25]:= (* checks *)
         α * q2 / b1
         {v2 / q2, v1 / (q0 + q1 * p1g), v1 / (q0 + q1 * p2g)}
         v2 / q2  ≤ v1 / (q0 + q1 * p1g)
         v2 / q2  ≤ v1 / (q0 + q1 * p2g)
         884. / (884 + k1)

Out[25]= 1.26692

Out[26]= {0.63346, 2.6087, 1.05571}

Out[27]= True

Out[28]= True

Out[29]= 0.993258
```

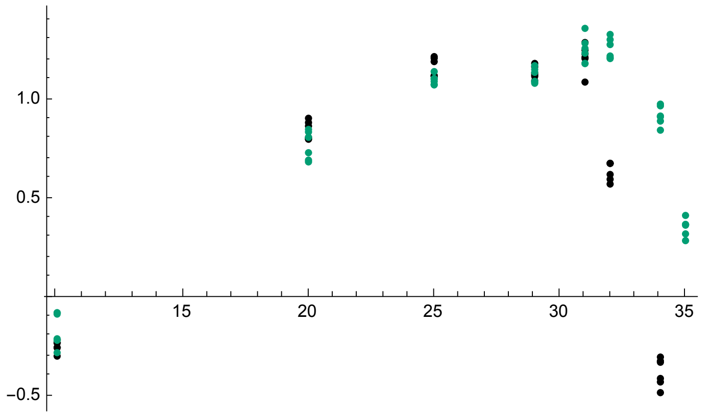
Import empirical data used to parameterize thermal performance curves of tolerant and intolerant strains:

```
In[30]:= dat = Import["/Users/colin/Dropbox/EvolutionPaper_SharedFolder/Data/data_for_tpc_fits_to_parameterize_model.csv"];

In[31]:= dat = Transpose[dat];

In[32]:= rvec = dat[[2, 2 ;;]];
         tvec = dat[[4, 2 ;;]];
         casevec = dat[[6, 2 ;;]];

In[35]:= rvecLOW = dat[[2, 2 ;; 43]];
         tvecLOW = dat[[4, 2 ;; 43]];
         casevecLOW = dat[[6, 2 ;; 43]];
         rvecHIGH = dat[[2, 44 ;;]];
         tvecHIGH = dat[[4, 44 ;;]];
         casevecHIGH = dat[[6, 44 ;;]];
```

Here’s what it looks like:

```
In[41]:= ListPlot[{Transpose[{tvecLOW, rvecLOW}], Transpose[{tvecHIGH, rvecHIGH}]}, PlotStyle → {Black, RGBColor["#009E73"]}]
```

Out[41]=

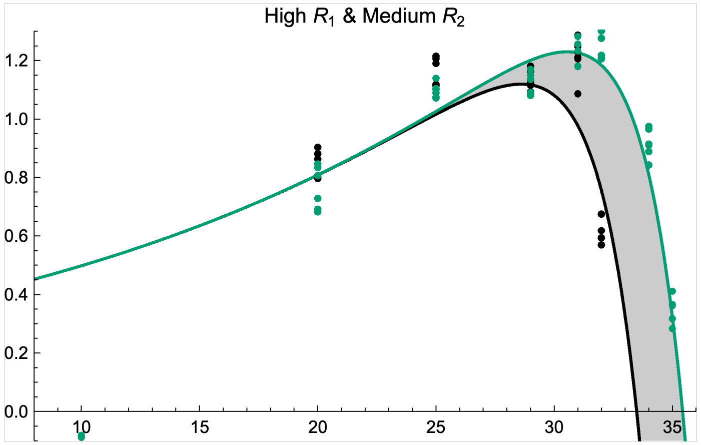


Explore model

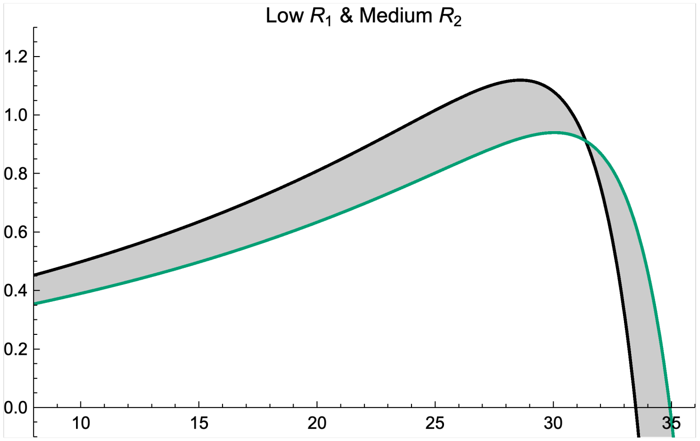
Consider effects of N supply on thermal performance curves (TPCs)

```
In[42]:= Clear[r1, r2]

In[309]:= r1tmpA = 884; r1tmpB = 5; r2tmp = 300;

Show[Plot[{b[p1g, r1tmpA, r2tmp, T] - d[p1g, T], b[p2g, r1tmpA, r2tmp, T] - d[p2g, T]},
  {T, 0, 36}, PlotRange → {{8, 36}, {-0.1, 1.3}}, Filling → {1 → {2}},
  PlotLabel → "High R1 & Medium R2", PlotStyle → {Black, RGBColor["#009E73"]}], ListPlot[
  {Transpose[{tvecLOW, rvecLOW}], Transpose[{tvecHIGH, rvecHIGH}]}, PlotStyle → {Black, RGBColor["#009E73"]}]]

Plot[{b[p1g, r1tmpB, r2tmp, T] - d[p1g, T], b[p2g, r1tmpB, r2tmp, T] - d[p2g, T]},
 {T, 0, 36}, PlotRange → {{8, 36}, {-0.1, 1.3}}, Filling → {1 → {2}},
 PlotLabel → "Low R1 & Medium R2", PlotStyle → {Black, RGBColor["#009E73"]}]
```

Out[∘]=



Out[∘]=



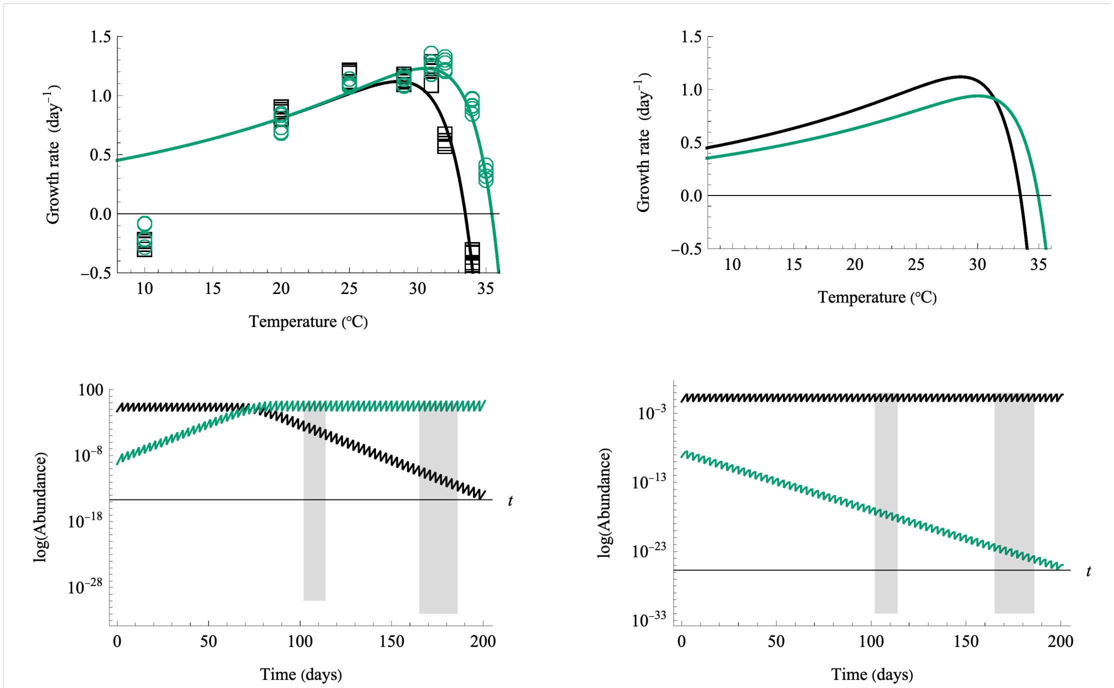
Run competition model under N-limiting and N-replete conditions

```
Clear[r1, r2];
tmax = 67 τ;

T = 31;
rin2 = 300;

(* N-limiting results *)
rin1 = 5;
n0 = rin1 / 100;
sollo = EcoSim[{p_1 → p1g, p_2 → p2g}, {r1 → 0, r2 → 0, n_1 → n0, n_2 → n0 * 10^-8}, tmax];

(* N-replete results *)
rin1 = 884;
solhi = EcoSim[{p_1 → p1g, p_2 → p2g}, {r1 → 0, r2 → 0, n_1 → n0, n_2 → n0 * 10^-8}, tmax];
```

Generate plot of TPCs and temporal dynamics under limiting and replete nutrients

```
In[60]:= panelA = Show[Plot[{b[p1g, r1tmpA, r2tmp, T] - d[p1g, T], b[p2g, r1tmpA, r2tmp, T] - d[p2g, T]},
             {T, 0, 36}, PlotRange → {{8, 36}, {-0.5, 1.5}}, PlotLabel → " ",
             PlotStyle → {Directive[Black], Directive[ RGBColor["#009E73"]]}, Frame → {{True, False}, {True, False}},
             FrameLabel → {"Temperature (°C)", "Growth rate  (day^-1)"}, LabelStyle → {FontFamily → "Times New Roman", Black}],
           ListPlot[{Transpose[{tvecLOW, rvecLOW}], Transpose[{tvecHIGH, rvecHIGH}]}, PlotMarkers → {□, ○},
             PlotStyle → {Directive[PointSize → Small, Black], Directive[RGBColor["#009E73"], PointSize → Small]}]];
         panelB = Plot[{b[p1g, r1tmpB, r2tmp, T] - d[p1g, T], b[p2g, r1tmpB, r2tmp, T] - d[p2g, T]}, {T, 0, 36}, PlotRange →
             {{8, 36}, {-0.5, 1.5}}, PlotLabel → " ", PlotStyle → {Directive[Black], Directive[ RGBColor["#009E73"]]},
           Frame → {{True, False}, {True, False}}, FrameLabel → {"Temperature (°C) \n", "Growth rate  (day^-1)"},
           LabelStyle → {FontFamily → "Times New Roman", Black}];
         panelC = Show[PlotDynamics[solhi, {n}, PlotPoints → 200, Logged → True, PlotLabel → " ",
             PlotStyle → {Directive[Black, Thickness[0.005]], Directive[RGBColor["#009E73"], Thickness[0.005]]},
             Frame → {{True, False}, {True, False}}, FrameLabel → {"Time (days)", "log(Abundance)"}],
           ListLogPlot[{{102, 10^-30}, {114, 10^-30}}, Joined → True, Filling → Top,
             FillingStyle → GrayLevel[0.3, 0.2], PlotStyle → {GrayLevel[0.3, 0]}],
           ListLogPlot[{{165, 10^-32}, {186, 10^-32}}, Joined → True, Filling → Top, FillingStyle → GrayLevel[0.3, 0.2],
             PlotStyle → {GrayLevel[0.3, 0]}], LabelStyle → {FontFamily → "Times New Roman", Black}];
         panelD = Show[PlotDynamics[sollo, {n}, PlotPoints → 200, Logged → True, PlotLabel → " ",
             PlotStyle → {Directive[Black, Thickness[0.005]], Directive[RGBColor["#009E73"], Thickness[0.005]]},
             Frame → {{True, False}, {True, False}}, FrameLabel → {"Time (days)", "log(Abundance)"}],
           ListLogPlot[{{102, 10^-32}, {114, 10^-32}}, Joined → True, Filling → Top,
             FillingStyle → GrayLevel[0.3, 0.2], PlotStyle → {GrayLevel[0.3, 0]}],
           ListLogPlot[{{165, 10^-32}, {186, 10^-32}}, Joined → True, Filling → Top, FillingStyle → GrayLevel[0.3, 0.2],
             PlotStyle → {GrayLevel[0.3, 0]}], LabelStyle → {FontFamily → "Times New Roman", Black}];

         Fig4 = GraphicsGrid[{{panelA, panelB}, {panelC, panelD}}, ImageSize → Large]
```

Out[64]=



```
Export["Fig4.pdf", leg]
```

Legend for use in post-processing of graphic:

In[65]:=

```
leg = LineLegend[{Directive[Thick, Black], Directive[Thick, RGBColor["#009E73"]]},
  {"Intolerant strain", "Tolerant strain"}, LabelStyle → {FontFamily → "Times New Roman"}]
```

Out[65]=

▬ Intolerant strain

▬ Tolerant strain

```
Export["Fig4_legend.pdf", leg]
```

Comparing carrying capacity (evidence for 2 resource system)