LUKE HEATON - NICK JONES - MARK FRICKER

FUNGAL T2M - SOFTWARE MANUAL

PLANT SCIENCES, OXFORD
FIRST EDITION

PUBLISHED BY PLANT SCIENCES, OXFORD
FIRST EDITION

First printing, November 2019

Current version, January 2020

Contents

Acknowledgements

This work was supported by a Visiting Fellowship at The Institute of Advanced Studies in Durham, The Leverhulme Trust (RPG-2015-437) and The Human Frontier Science Program (RGP0053/2012).

The export_fig package was written by Oliver Woodford and Yair Altman and downloaded from:
https://uk.mathworks.com/matlabcentral/fileexchange/23629-export-fig

The colorcet.m package was written by Peter Kovesi and downloaded from:
https://peterkovesi.com/projects/colourmaps/

Peter Kovesi. Good Colour Maps: How to Design Them. arXiv:1509.03700 [cs.GR] 2015
https://arxiv.org/abs/1509.03700

1
Installation

1.1 Overview

The code to run the simulation is provided as open source under a GNU General Public License v3.0 from:

`https://github.com/markfricker/Fungal_multicellularity`

The code is provided in three formats:

- A set of script files and functions that will run in MATLAB® on any platform;
- A MATLAB® app with a GUI interface that can be installed in MATLAB® on any platform (requires MATLAB® release 2019b or later, and a screen resolution of 1600 x 900 or greater);
- A compiled standalone version that can be installed and run on Windows 10, 64 bit platforms and requires a screen resolution of 1600 x 900 or greater;

On the home page of the github repository, click on the green **Clone or download** button (Figure 1.1) and then the blue **Download ZIP** button.

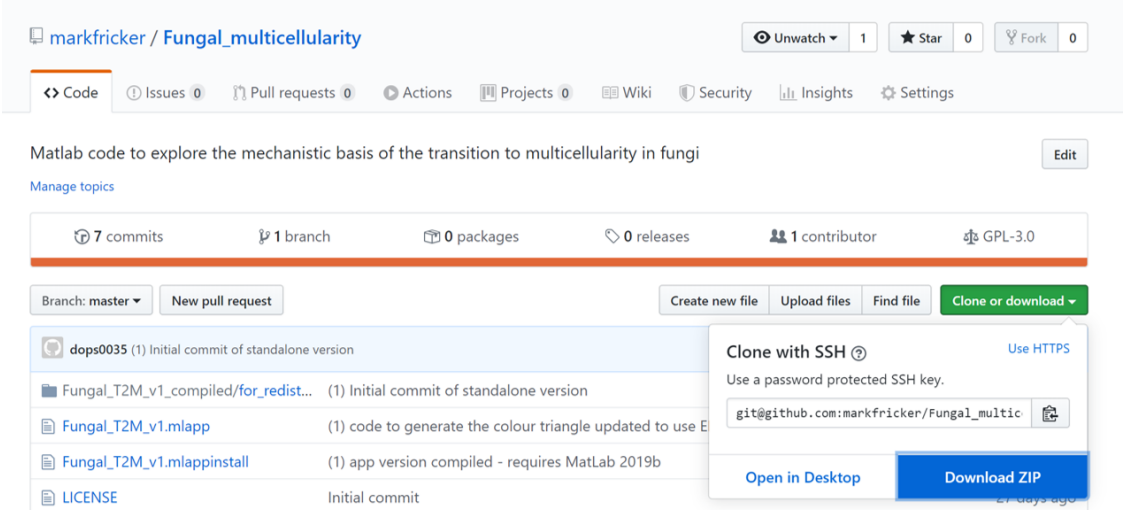


Figure 1.1: Download the software in a zip file from the github repository

1.2 *Download all files from the github repository*

Once the file has downloaded, extract all the files. If you intend to run the script versions, the extracted folder needs to be on the MATLAB path.

1.3 *Installation of the MATLAB app*

The MATLAB® app installer file, *Fungal_T2M_v1.mlappinstall*, contains everything necessary to install and run an app within the MATLAB® environment, including the source code, supporting data, information (such as product dependencies), and the app icon (Figure 1.2). The installation time is less than a minute.

Double-clicking on the *Fungal_T2M_v1.mlappinstall* file should launch MATLAB® if it is not already running, and install the app in the app toolbar (Figure 1.3). The program can be run, by clicking on the icon in the toolbar.

Figure 1.2: The program icon

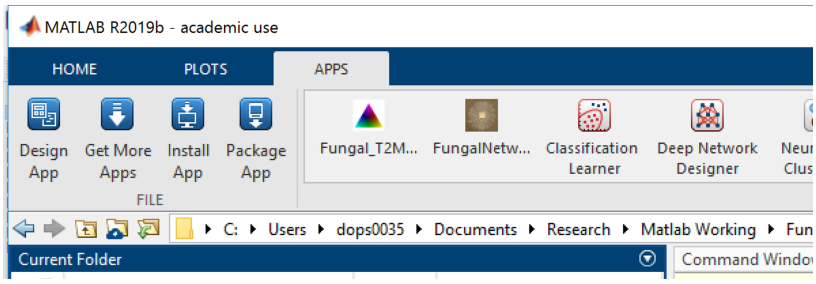


Figure 1.3: Location of the Fungal_T2M app in the APPS menu within MATLAB®

The app version of the program also requires the MATLAB® Image Processing Toolbox™ to be installed.

1.4 *Installation of the stand-alone program*

The software has been tested on Windows 10, and requires a minimum screen resolution of 1600 x 900. In addition, an appropriate version of the MATLAB® Compiler Runtime (MCR) is required to install the set of shared libraries that enable execution of the compiled MATLAB application. The MCR should automatically download from the MathWorks website when the program is installed for the first time. Alternatively MCR can be downloaded from:

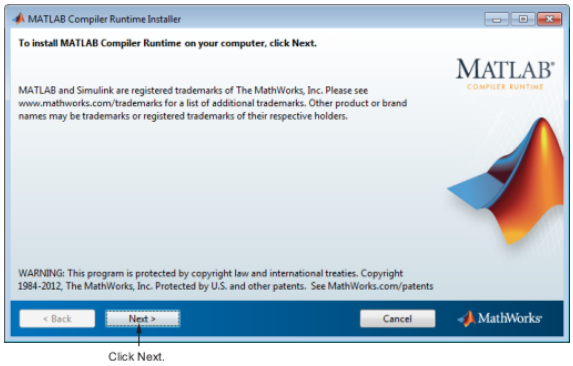
`http://www.mathworks.com/products/compiler/mcr`.

To install the MCR and standalone package, double-click the compiled MATLAB self-extracting *Fungal_T2M_v1.exe* file located in the subfolder:

`\Fungal_T2M_v1_compiled\for_redistribution`

This extracts the MATLAB® Runtime Installer from the archive, along with all the files that make up the deployed MATLAB en-

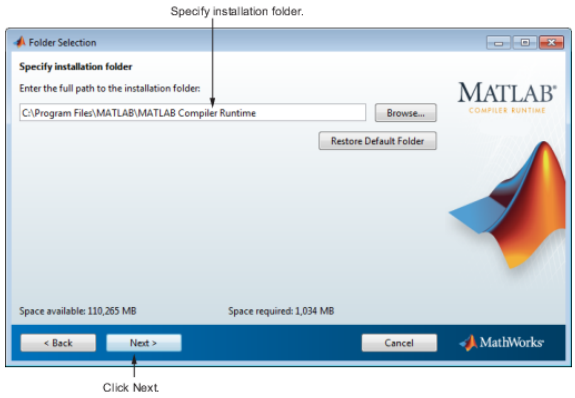
vironment. Once all the files have been extracted, the MATLAB® Runtime Installer starts automatically. When the MATLAB® Runtime Installer starts, it displays the following dialog box:



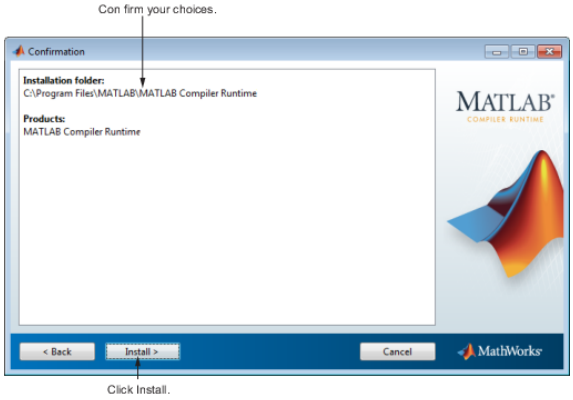
Read the information and then click **Next** to proceed with the installation.

Specify the folder in which you want to install the MATLAB® runtime in the Folder Selection dialog box and click **Next**.[1] It is recommended to keep the default settings as this ensures the path to other program files is set automatically.

[1] On Windows systems, you can have multiple versions of the MATLAB® runtime on your computer, but only one installation for any particular version. If you already have an existing installation, the MATLAB® runtime Installer does not display the Folder Selection dialog box because you can only overwrite the existing installation in the same folder.



Confirm your choices and click **Install**. The MATLAB® Runtime Installer starts copying files into the installation folder. Installation takes about 10 minutes.



Click **Finish** to exit the installer.

MATLAB® Runtime Installer Readme File: A readme.txt file is included with the MATLAB Runtime Installer. This file, visible when the MATLAB Runtime Installer is expanded, provides more detailed information about the installer and the switches that can be used with it.

1.5 Installation of additional program files needed

A number of additional files needed to run the full suite of programs may also be installed at the same time as the main program. The latest version of Java needs to be installed, and is available from:

`http://www.java.com/en/`

Output of images at full resolution uses *export_fig.m* originally written by Oliver Woodford (2008-2014) and now maintained by Yair Altman (2015-). This is included in the installation. However, when exporting to vector format (PDF or EPS) this function requires that ghostscript is installed on your system. Ghostscript can be downloaded from:

`http://www.ghostscript.com`.

When exporting images to eps and pdf formats, *export_fig* additionally requires pdftops, from the Xpdf suite of functions. This is included in the xpdf tools package and can be downloaded from:

`https://www.xpdfreader.com/download.html`

2

Running the simulation using the GUI

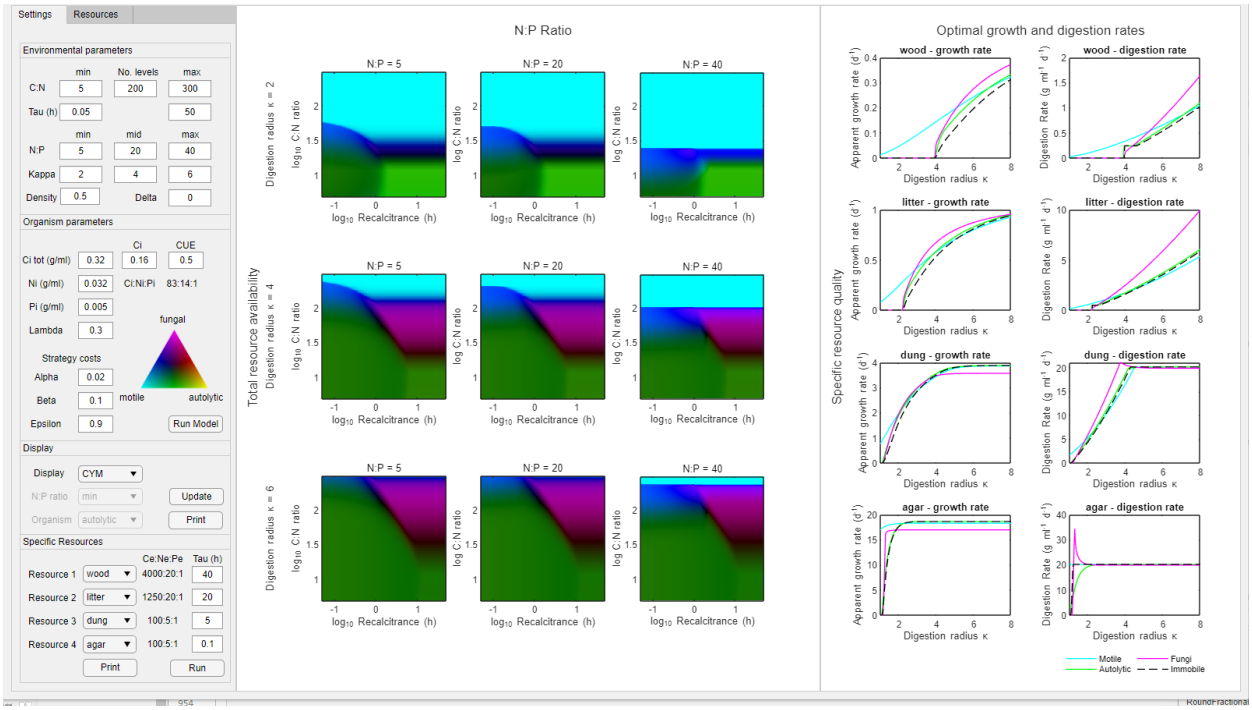


Figure 2.1: The GUI interface for the fungal multicellularity simulation

2.1 *Introduction*

The MATLAB app version and the standalone version both provide the same interface to run the simulation (Figure 2.1). The set of left-hand panels allow the user to change the parameters controlling the external resource environment and the internal organism parameters, including the cost terms for each of the different strategies. The definition of the parameters is given in Table 2.1.

Table 2.1: Definition and values of modelling parameters

Tunable Environmental Parameters	**Symbol and Units**	**Interpretation**
Supply of C, N and P	C_E, N_E, P_E	Grams of C, N and P per millilitre of substrate, set by the C:N and N:P ratio of the resource, and the resource density in g ml^{-1}
Recalcitrance	τ hours	Time required for an exoenzyme to supply a mass of C, N or P equal to the total mass required to synthesise the exoenzyme in question
Relative digestion radius	κ	Cylindrical cells and hyphae digest resource over a distance times the cell radius
Resource accessibility	δ	Ratio of C that has to be digested to release each N, to reflect that N is embedded within C-rich polymers

Tuneable Organism Parameters	**Symbol and Units**	**Interpretation**
Core demand for C, N and P	$C_I = 0.33$ g ml^{-1} $N_I = 0.032$ g ml^{-1} $P_I = 0.005$ g ml^{-1}	Core demand for C, N and P per unit volume of any organism. C_I comprises the C needed for growth and respiration, with the ratio determined by the Carbon Use Efficiency (CUE).
Motility cost	$\alpha = 0.02$	Mass of C and N required to synthesise motile apparatus, relative to core demand for C and N
Vesicle transport cost	$\beta = 0.1$	Mass of C and N in vesicles, relative to the total C and N in exoenzymes and the fungal core
Recycling efficiency	$\epsilon = 0.5$	Fraction of C, N and P that is recouped by autolysis
Maximal rate of resource use	$\lambda = 0.3$ g ml^{-1} $hour^{-1}$	Maximal mass of resource any cell can use per unit time and volume

Model Variables	**Symbol and Units**	**Interpretation**
Relative density of exoenzymes	x	Mass of C and N used for exoenzymes relative to the core demand for C and N
Relative density of C, N or P	x_C, x_N, x_P	Mass of C and N used for C, N or P digesting exoenzymes relative to the core demand for C and N

Functions of x	**Symbol and Units**	**Interpretation**
Time to exhaust local resource	T hours	Time until one of C, N or P is locally exhausted
Time to exhaust C, N or P	T_C, T_N, T_P hours	Time until relevant element is locally exhausted
Specific growth rate of growing cell	η h^{-1}	Rate of synthesis of cell biomass, per unit of cell biomass, in the growing cells
Apparent growth rate	μ h^{-1}	Volumetric rate of colonisation, per unit volume colonised
Total limiting supply	Ω	Maximum number of daughter cells and their exoenzymes that can be synthesised, given the local supply of resource
Rate of resource use	Γ g ml^{-1} h^{-1}	Total mass of resource used per unit time and volume, in the growing cells

2.2 *Environmental parameters*

The C:N ratio of the resource ranges from the minimum value (*min*), typically around 5, to the maximum value (*max*), with the number of intermediate levels set by *levels* (Figure 2.2).

This forms the y-axis for each of the resultant plots (on a log_{10} scale). The x-axis is set by the recalcitrance (τ), and ranges from *min* (default 5) to *max* (default 300), with the same number of intermediate levels. Results are plotted for three values of the N:P ratio, set by *min*, *mid* and *max* (default 5, 20 and 40, respectively) and three values for the overall resource availability, given by the radius of digestion (κ), again set by *min*, *mid* and *max* (default 2, 4 and 6, respectively).

The overall resource density in g ml^{-1} is used to convert the C:N and C:P ratios into grams ml^{-1} of C_e, N_e, and P_e

The value of the additional accessibility parameter δ, reflects the fact that some fraction of the available C must be digested in order to access N, irrespective of the category of organism. For example, if the C:N ratio of the substrate is 200:1, imposing a value of $\delta = 0.1$ forces organisms to digest at least 20 C for every N they acquire.

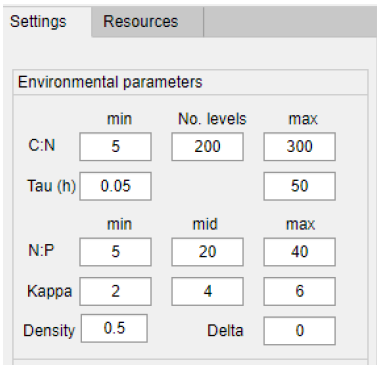


Figure 2.2: Controls for the environmental parameters that define the resource quantity and quality

2.3 *Organism parameters*

The internal C, N and P required by all organisms are set by C_i, N_i and P_i, respectively, in g ml^{-1}. These values are also displayed as the molar C:N:P ratio in the adjacent text box (Figure 2.3). The default values are typical of fungal and microbial cells, although there is considerable variation in both the absolute amounts and the relative ratios.

The total amount of C required (C_i *tot*) also includes that used in respiration. This is given as a proportion that depends on the carbon use efficiency (CUE), defined as the fraction of growth to total assimilation. The default value of CUE is set as 50%, but values can be much lower than this[1].

The maximum rate that resource that can be used (by any organism) is set by *Lambda*, with a default value of 0.3 g ml^{-1} h^{-1}. This would equate to a doubling time of just under 1h for the default resource density, and represents an upper limit unlikely to be achieved for organisms in the wild.

In addition, autolytic, motile and fungal cells have a single organism-specific parameter that helps to define their different strategies. Thus, *Alpha* is the additional cost of being motile, *Beta* is the cost associated with internal transport, whilst *Epsilon* is the fraction of an autolytic cell that can be recovered by recycling.

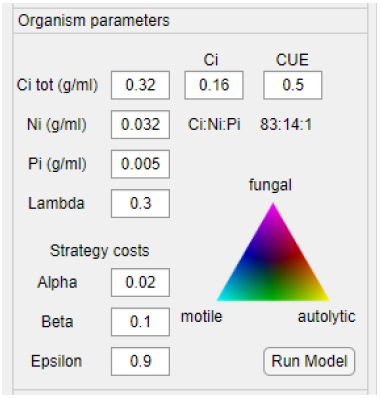


Figure 2.3: Parameter settings for the biological organisms

[1]

2.4 *Running the simulation*

The **Run Model** button will iterate through the external resource environmental parameters for each organism, and returns pseudo-

colour coded maps showing which class of organism is predicted to grow most rapidly under each set of conditions (Figure 2.4), according to the inset colour triangle (Figure 2.5). The default settings are matched to the data reported. The run time is about 160-180 min on a Windows 10 machine with an Intel® Core™ i7-8750H CPU running at 2.20GHz with 32GB RAM. Reduced run times can be achieved by decreasing the number of *levels* (the minimum suggested is around 20) as run time increases with $levels^2$.

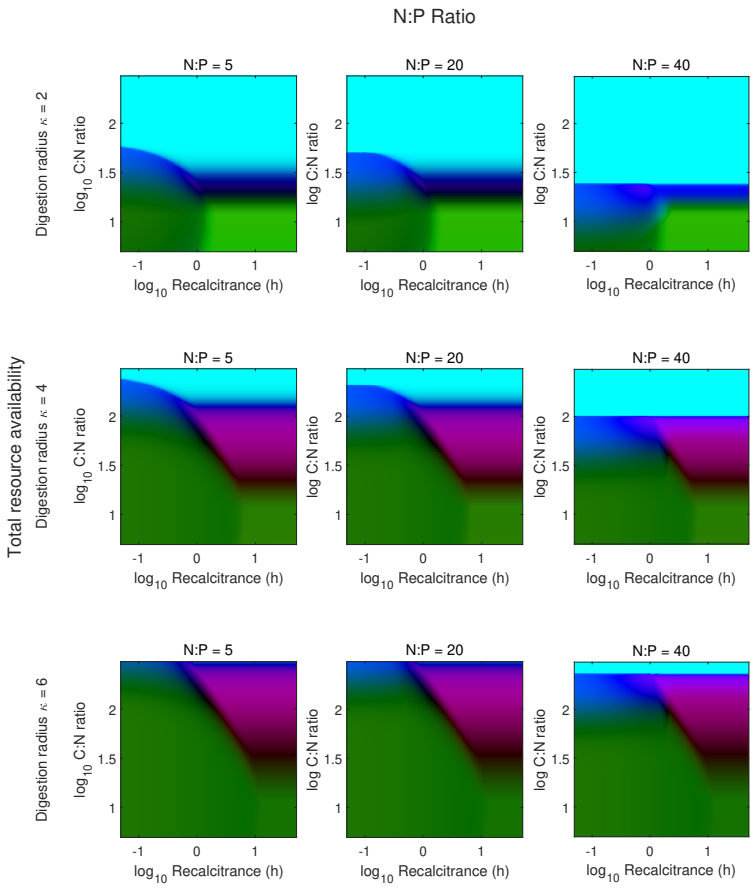


Figure 2.4: Colour-coded maps showing the relative performance of each type of organism across a wide range of resource quantity and quality.

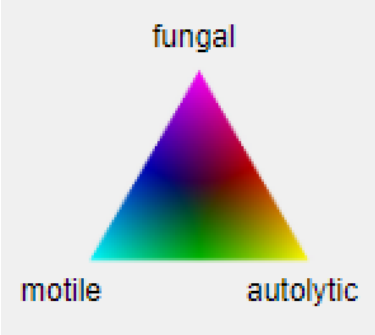


Figure 2.5: Legend colour code

Thus regions of parameter space where motile cells dominate are coloured cyan, regions where fungi dominate are magenta and regions where autolytic cells dominate are yellow. In many regions, two or more organisms may fare equally well, and there are typically extensive regions where motile and fungal organisms both thrive (blue), or motile and autolytic are equivalent (green).

2.5 *Display*

The cyan-magenta-yellow maps show the relative performance of each class of organism compared to the others.

In the *Display* panel (Figure 2.6), the *Display* drop down menu

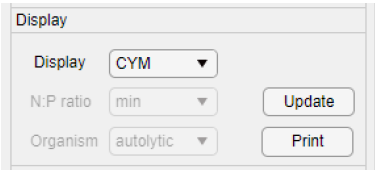


Figure 2.6: Display options

can be used to show how the individual growth rates compare to the fastest growing cell with no resource restriction, by selecting the *growth* option (Figure 2.7). This provides plots for each organism for each digestion radius, whilst the *N:P ratio* has to be selected from the corresponding drop down menu.

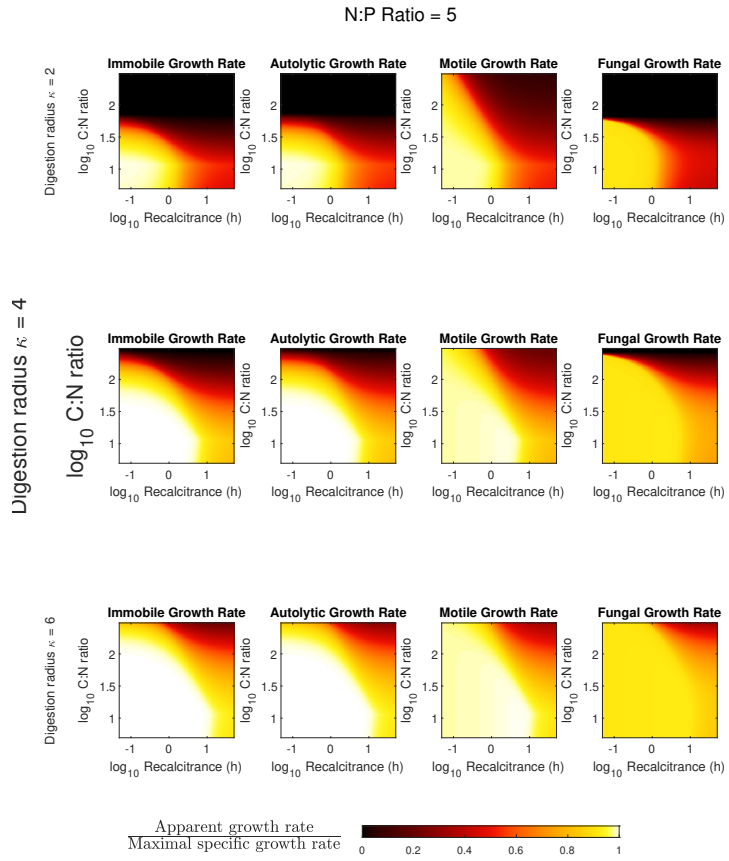


Figure 2.7: Colour-coded maps showing the relative growth of each type of organism compared to the maximum growth of any type of organism.

Alternatively, the relative growth rate of a fungal organism can be compared with one of the other classes of organism to compare directly the magnitude of the hyphal advantage (Figure 2.8). The **Print** button saves a copy of the display as a png or pdf file.

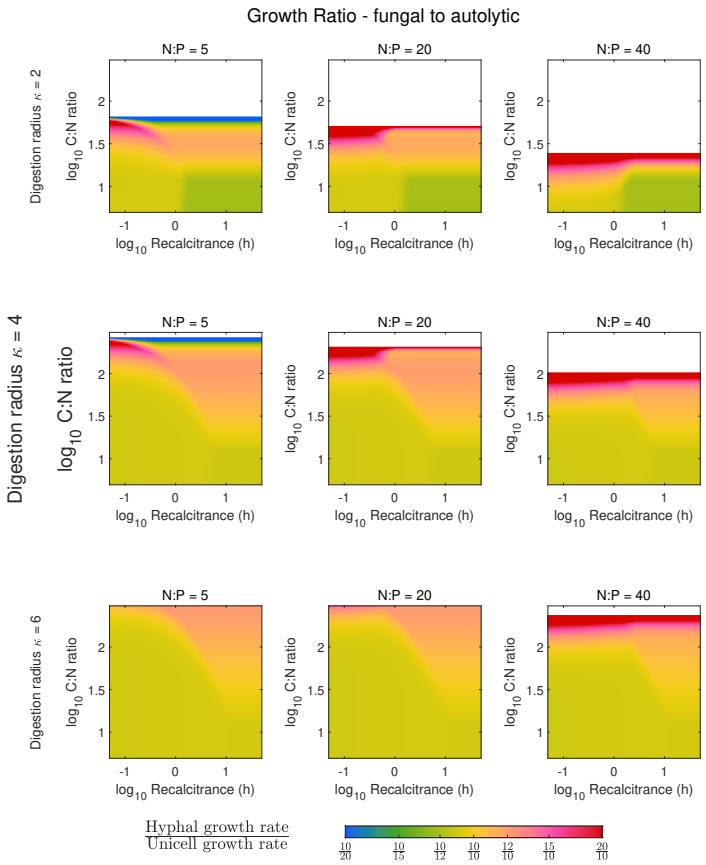


Figure 2.8: Colour-coded maps showing the relative growth of a hyphal organism compared to a specific class of unicellular organism.

2.6 *Performance of different classes of organism on specific resources*

The colour-coded maps display results for a very extensive range of resource environments. The *Specific resources* panel (Figure 2.9) provides options to plot the performance for four specific resources, with the defaults set to wood (C:N:P 4000:20:1, τ = 40h), leaf litter (C:N:P 1250:20:1, τ = 20), dung (C:N:P 100:5:1, τ = 5h) and agar (C:N:P 100:5:1, τ = 0.1h).

The **Run** button runs the simulation for these specific resource configurations using the other parameters defined for the full model (Figure 2.10)

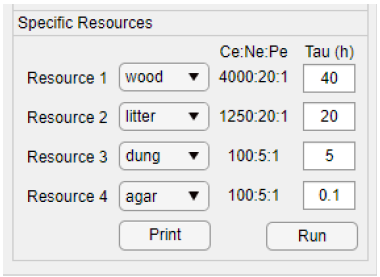


Figure 2.9: Display options

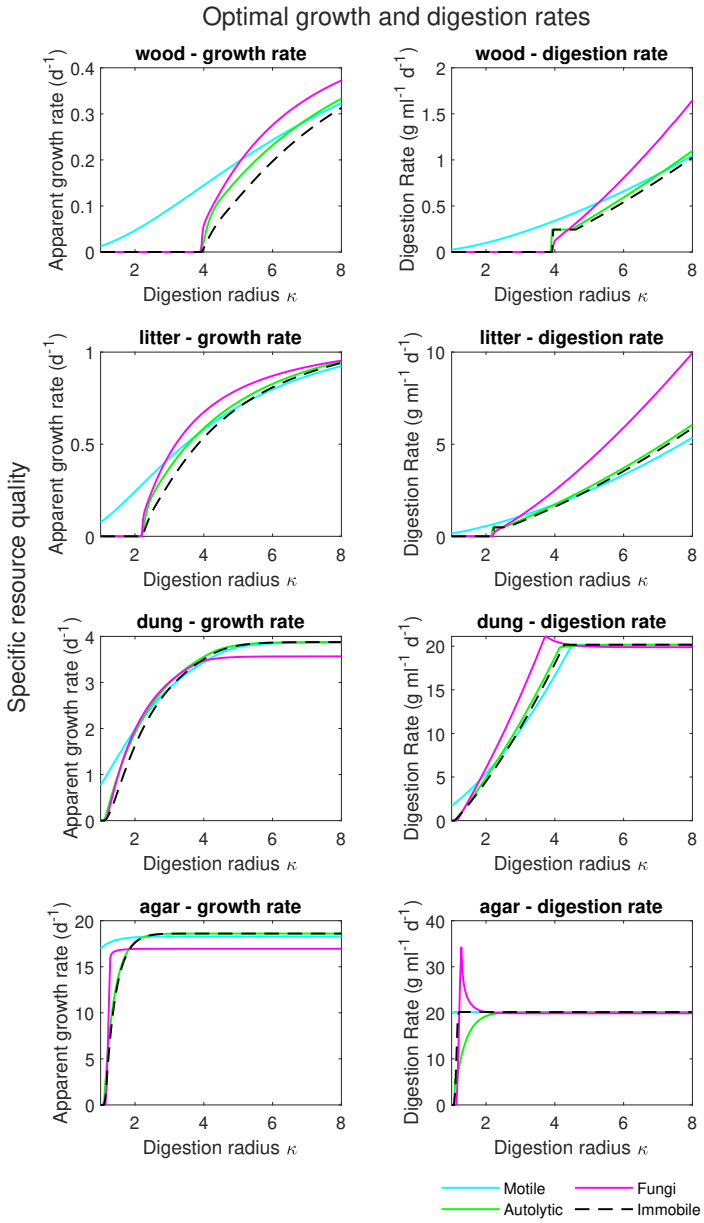


Figure 2.10: Colour-coded maps showing the relative growth of a hyphal organism to a specific class of unicellular organism.

Whilst the values for C:N:P are widely reported in the literature, the values for τ are not defined experimentally for most substrates.

Thus, there is the option to vary τ for each resource using the adjacent edit box. For direct comparison, the same resource can be selected in all four dropdown menus and the simulation run with different values of τ. Increasing the value of tau significantly decreases the growth rate for all organisms, but in general, changing tau only has a small effect on the relative performance of different classes of organisms.

2.7 *Inclusion of user defined resources*

Given that there are many different resource possibilities other than the four defaults used, there is an option to define more substrates using the **Resources** tab at the top of the control panels (Figure 2.11)

The Table includes the four default resource types. Additional resources can be included using the **Add** button, which adds an additional row to the table. Likewise the **Remove** button will delete the currently selected entry. The modified table can be saved using the **Save** button, and re-loaded using the **Load** button. The **Update** button adds the new set of resources to the **Resource** drop down menus in the **Specific Resource** panel in the main **Settings** tab. The effect of changing resources can then be explored using the **Run** button.

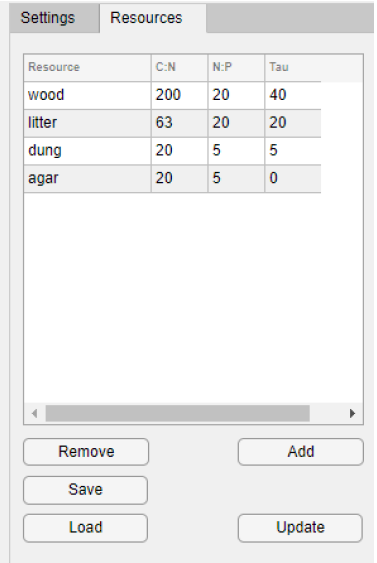


Figure 2.11: Resource table to allow addition of specific resources with user-defined C:N:P ratios and recalcitrance (Tau)

3
Command Line Version

3.1 Overview

The set of MATLAB® scripts that can be used to run the different components of the simulation are given in Table 3.1. These should work in any release of MATLAB® on any platform. They are the same functions as called by the GUI and standalone versions of the program.

Script Name	**Description**
vary_CNP_and_tau.m	The main script to run the simulation and generate colour-coded maps of relative performance
vary_kappa.m	Script to generate the growth rate and rate of digestion for each class of organism in a specific resource environment

Function Name	**Description**
find_best_cell.m	Function to determine numerically the best growth rate (μ) for a given investment in digestive enzymes (x) using the parameter values for a cell growing at the maximum possible rate without substrate depletion which implements equations 1-2 of the Supplementary Information.
find_best_immobile.m	Function to determine numerically the best growth rate (μ) for a given investment in digestive enzymes (x) using the parameter values for immobile cells. Calls the function *find_mu_given_x_immobile.m* which implements equations 1-6 of the Supplementary Information.
find_best_autolytic.m	Function to determine numerically the best growth rate (μ) for a given investment in digestive enzymes (x) using the parameter values for autolytic cells. Calls the function *find_mu_given_x_autolytic.m* which implements equations 7-10 of the Supplementary Information.
find_best_motile.m	Function to determine numerically the best growth rate (μ) for a given investment in digestive enzymes (x) using the parameter values for motile cells. Calls the function *find_mu_given_x_motile.m* which implements equations 11-13 of the Supplementary Information.

find_best_fungi.m	Function to determine numerically the best growth rate (μ) for a given investment in digestive enzymes (x) using the parameter values for fungal cells. Calls the function *find_mu_given_U_fungi.m* which implements equations 14-17 of the Supplementary Information.

Display functions	**Description**
generate_CYM_merge.m	Function to generate the Cyan-Yellow-Magenta maps of relative growth rate. If the fastest growing colony has grown by a factor of n, and a slower growing colony has grown by a factor of m, the channels for the colonies are coloured accordingly with intensity 1 and m/n respectively. Higher n increases the apparent difference in colour, making a lighter CYM merge.
heatmap_comparison.m	Generates colour-coded ratio heatmap of the relative growth of fungal cells and autolytic cells in the standard set of resource environments. Calls *colorcet.m* to define a perceptually uniform ratio colourmap ('R3') and *applycolourmap.m* to apply it to the ratio growth image. See SI Figure 3.
heatmap_growth_inhibition.m	Generates heatmap showing the relative growth of each class of organism compared to the maximum growth possible with unlimited resource. Calls *colorcet.m* to define a perceptually uniform ratio colourmap ('L3') and *applycolourmap.m* to apply it to the relative growth inhibition image. See SI Figure 4.

Table 3.1: MATLAB scripts and functions

3.2 *Running the simulation for varying resource quality and recalcitrance*

Open the *vary_CNP_and_tau.m* script in the MATLAB® Editor. The first part of the code sets the resource and organism parameters (Listing 3.1), which can be modified manually to explore different environmental resources or impose different costs on the strategy used by each organism. For initial trials, the *res1* parameter can be set to a much lower value (20) to reduce the computation time, whist still titrating the range of C:N and recalcitrance values (Figure 3.2).

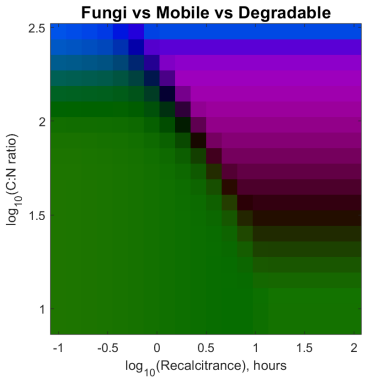


Figure 3.1: Coarse output for rapid prototyping

Listing 3.1: Setting parameter values at the start of the vary_CNP_and_tau code

```
%% set up parameters. These can be manually edited.
% res1 is the number of different values for the C:N ratio and recalcitrance
% that are tried
res1 = 200;

% xres determines the number of different values for x that are tried in finding
% the optimal solution. Final value found is accurate to xres^2.
xres = 200;

% sres is the number of iterations used in finding best solution for a given
% environment and a given rate of sythesis.
sres = 60;

% N_to_P set the N:P ratio of the resource.
N_to_P = 20;

% kappa is the radius of digestion, normalised to the radous of the cells/hypha,
% and sets the total amount of resource available.
kappa = 6;

% C_to_N_min and C_to_N_max define the limits of the C:N ratio in the resource.
% The number of intermediate steps is controlled by res1. The C:N ratio is
% plotted on the y-axis of the resultnt map.
C_to_N_min = 8;
C_to_N_max = 300;

% tau_min and tau_max define the limits of the resouce recalcitrance, where tau
% is the time in hours for hydrolases to digest their own mass . the number of
% intermediate steps is controlled by res1. The recalcitrance is plotted on the
% x-axis of the resultnt map.
tau_min = 0.1;
tau_max = 100;

% Ci, Ni and Pi are the mass of carbon, nitrogen and phosphorous, respectively,
% needed per unit volume of growth, in g per ml.
Ci = 0.33;
Ni = 0.032;
Pi = 0.005;

% dry weight in grams per ml of the substrate.
density = 0.5;

% epsilon is the efficiency of recycling for senscent autolytic cells.
epsilon = 0.5;

% alpha is the additional mass of machinery needed for cell mobility, relative
% to the mass of essential metabolic machinery.
alpha = 0.02;

% beta is the mass of material in vesicles, relative to the mass of the rest of
% the fungus, including hydrolases.
beta = 0.1;

% lambda is the maximum rate of resource use per unit volume, in g per ml per
% hour.
lambda = 0.3;

% ∆ is the ratio of C that has to be digested to release each N, to reflect
% that N is embedded within C-rich polymers.
∆ = 0;
```

The output is an array of the relative growth (Mu) and the amount of digestive enzymes released (x) for each class of organism. The relative growth can be compared using the three different display scripts:

- *generate_CYM_merge.m* gives a three-way comparison using a cyan-yellow-magenta colour-code for motile, autolytic and fungal cells, respectively.
- *heatmap_growth_inhibition.m* compares the relative growth of each organism against a cell with no limitation on resource availability such that the colony continues to grow at the maximum rate under the conditions selected.
- *heatmap_comparison.m* compares the rate of fungal growth explicitly with autolytic cells to provide a more quantitative visual indication of the hyphal advantage in different resource environments.

3.3 *Running the simulation for a specific resource quality in varying abundance*

The second main script (Listing 3.2) requires the user to define the resource quality (C:N:P ratios and recalcitrance), but then calculates the relative growth rate for each class of organism and the amount invested in digestive enzymes to give the optimal strategy. As with the *vary_CNP_and_tau* script, the first section has to be edited manually to set up different parameter configurations. The output is automatically generated as a plot of growth rate against resource abundance (varying kappa) and rate of digestion for the four classes of organism.

Listing 3.2: Setting parameter values at the start of the vary_kappa code

```
%% set up parameters. These can be manually edited.
% res1 is the number of different values for kappa that we try
res1 = 200;

% xres is the number of iterations used in finding best solution for
% a given environment and a given rate of sythesis
xres = 200;

% sres is the number of iterations used in finding best solution for
% a given environment and a given rate of sythesis
sres = 60;

% k_min and k_max are the minimum and maximum values for the resource
% acquisition length that we try
k_min = 1;
k_max = 8;

% Ci, Ni and Pi are the mass of carbon, nitrogen and phosphorous, respectively,
% needed per unit volume of growth, in g per ml. Ci includes the internal C,
% and the C required for respiration, with the ratio set by the Carbon Use
% Efficiency (CUE).
Ci = 0.33;
Ni = 0.032;
Pi = 0.005;

% C_to_P and N_to_P set the C:P and N:P ratio of the resource, respectively.
C_to_P = 2000;
N_to_P = 10;

```

```
% tau is the time in hours for hydrolases to digest their own mass and reflects
% the recalcitrance of the resource.
tau = 40;

% dry weight in grams per ml of the substrate
density = 0.5;

% ∆ is the minimal fraction of C that must be digested in order to
% digest the available N
∆ = 0;

% epsilon is the efficiency of recycling for autolytic cells
epsilon = 0.5;

% alpha is the mass of machinery needed for cell mobility, relative
% to the mass of essential machinery
alpha = 0.02;

% beta is the mass of material in vesicles, relative to the mass of
% the rest of the fungus, including hydrolases
beta = 0.1;

% lambda is the maximum rate of resource use per unit volume,
% in g per ml per hour
lambda = 0.3;
```

Figure 3.2: Growth rate and digestion rate for a specific resource quality and quantity