LUKE HEATON - NICK JONES - MARK FRICKER

FUNGAL MULTICELLULARITY

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 by Oliver Woodford and Yair Altman:
https://uk.mathworks.com/matlabcentral/fileexchange/23629-export-fig

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 release 2019b or later);
- A standalone version that can be installed and run on Windows 64bit platforms;

1.2 Download all files from the github repository

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.

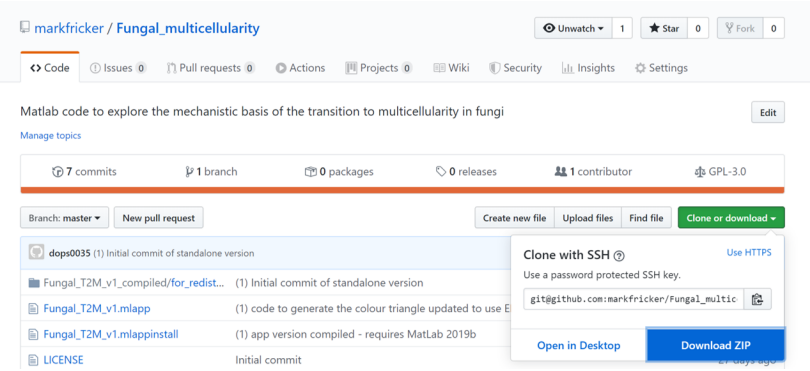


Figure 1.1: Download the software in a zip file from the github repository

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

1.3 Installation of the MATLAB app

The MATLAB® app installer file, .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).

Double-clicking on the .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.

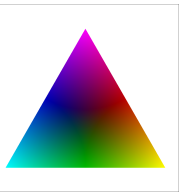


Figure 1.2: The program icon

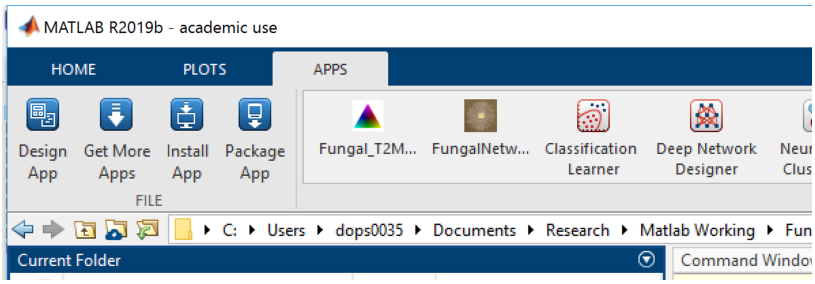


Figure 1.3: Location of the Fungal_T2M app in the APPS menu within MATLAB

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 enables 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 the MathWorks Website:

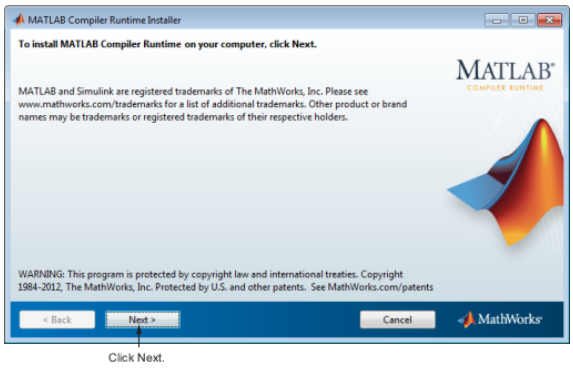
`http://www.mathworks.com/products/compiler/mcr`.

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

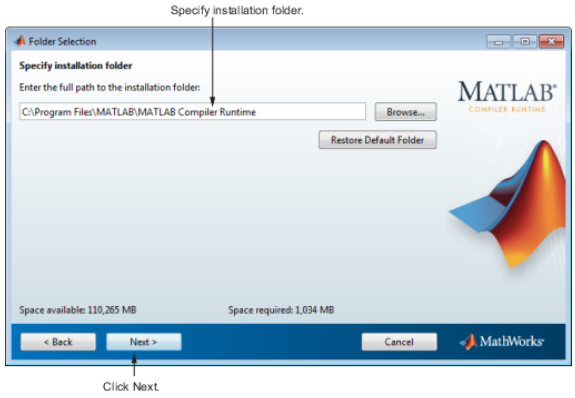
`\Fungal_T2M_v1_compiled\for_redistribution`

This extracts the MATLAB Runtime Installer from the archive, along with all the files that make up the deployed MATLAB environment. 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.

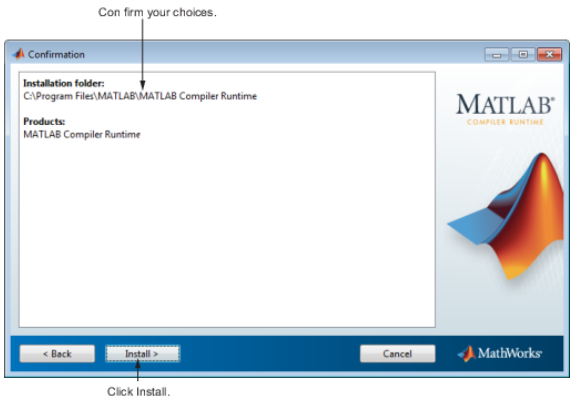
Specify the folder in which you want to install the MATLAB runtime in the Folder Selection dialog box and click **Next**. It is recommended to keep the default settings as this ensures the path to other program files is set automatically.



Note: 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



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-). 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

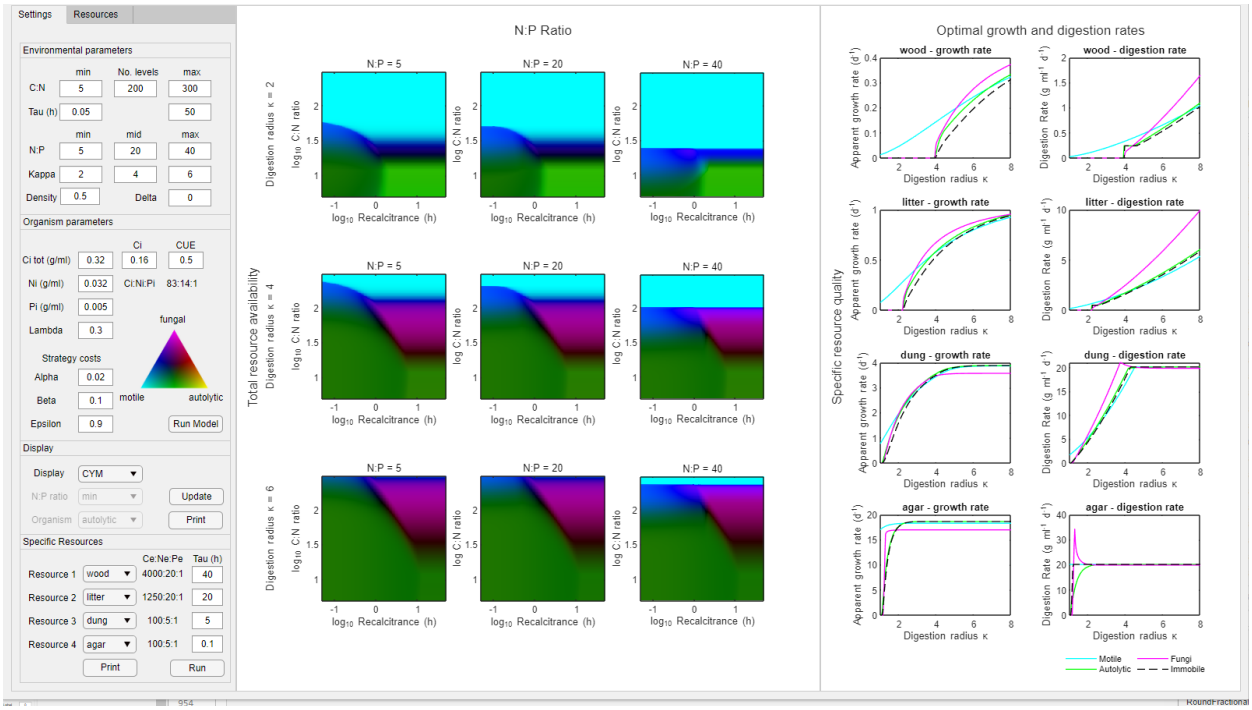


Figure 2.1: The GUI interface for the fungal multicellularity simulation

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.

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

Organism Parameters		
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
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		
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		
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 $hour^{-1}$	Rate of synthesis of cell biomass, per unit of cell biomass, in the growing cells
Apparent growth rate	μ $hour^{-1}$	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} $hour^{-1}$	Total mass of resource used per unit time and volume, in the growing cells

Table 2.1: Definition and values of modelling parameters

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* to *max* with the same number of intermediate levels. Results are plotted for three values of the N:P ratio, set by *min*, *mid* and *max* and three values for the overall resource availability, given by the radius of digestion (κ), again set by *min*, *mid* and *max*.

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.

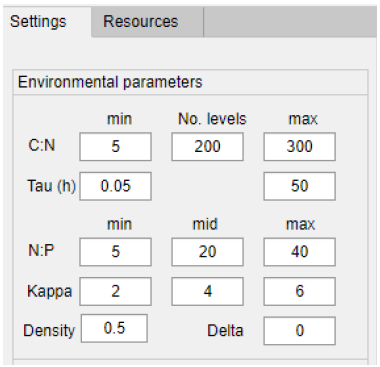


Figure 2.2: Controls for the environmental parameters that define the resource quantity and quality

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 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 recycled.

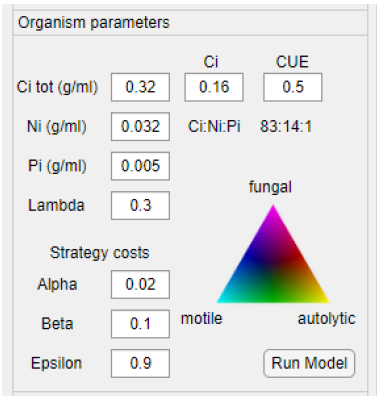


Figure 2.3: Parameter settings for the biological organisms

[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).

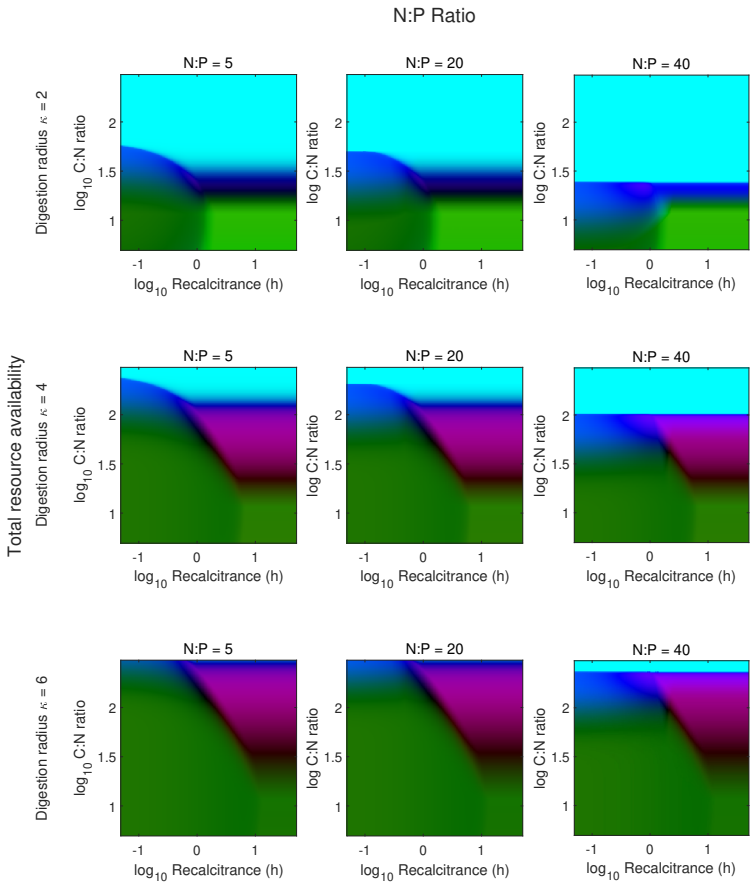


Figure 2.4: Colour-coded maps showing the relative performance of each type of organism across a wide set of resource quantity and quality.

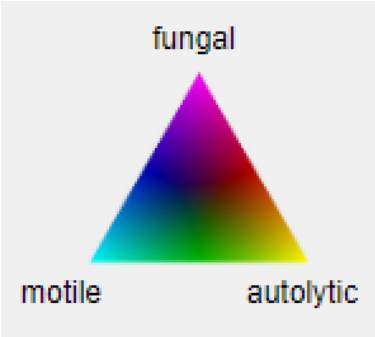


Figure 2.5: Legend colour code

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 tyically extensive regions where motile and fungal organisms both thrive (blue), or motile and autolytic are equivalent (yellow).

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 can be used to show the growth rates compared to the fastest growing cell by selecting the *growth* option (Figure **??**). 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.

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.

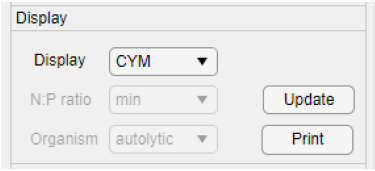


Figure 2.6: Display options

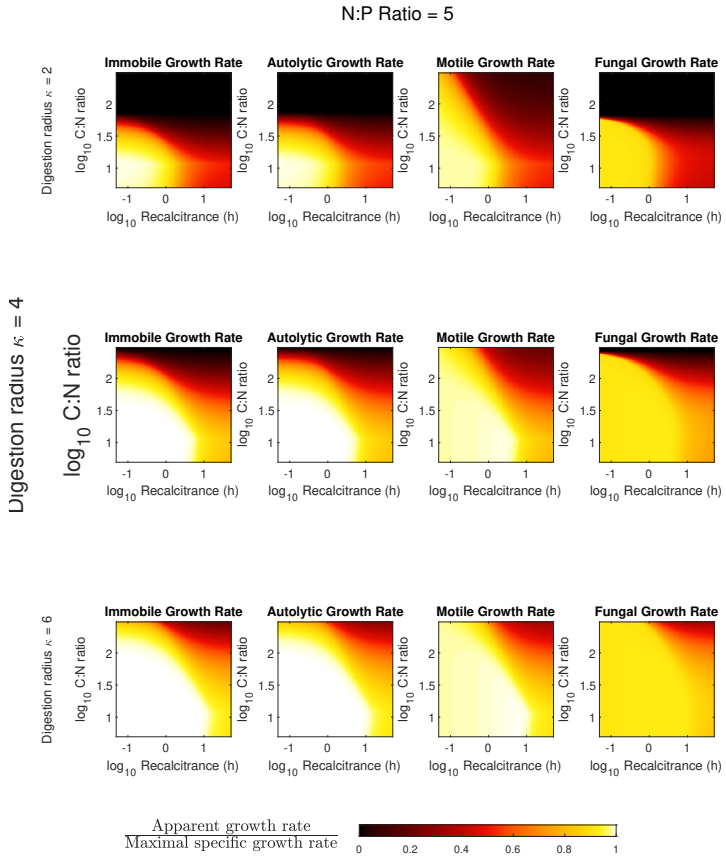


Figure 2.7: Colour-coded maps showing the relative growth of each type of organism compared to the maximum growth of any type of organism.

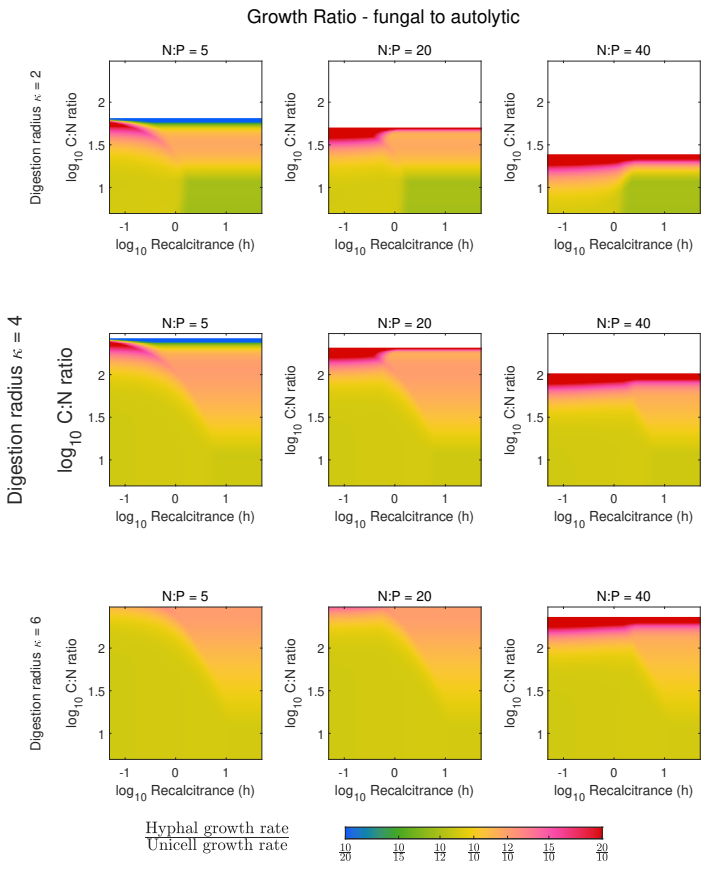


Figure 2.8: Colour-coded maps showing the relative growth of a hyphal organism to a specific class of unicellular organism.

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)

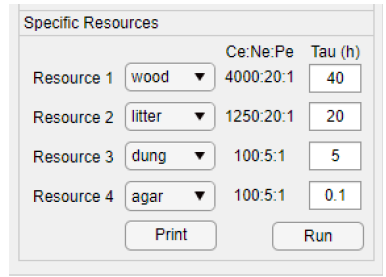


Figure 2.9: Display options

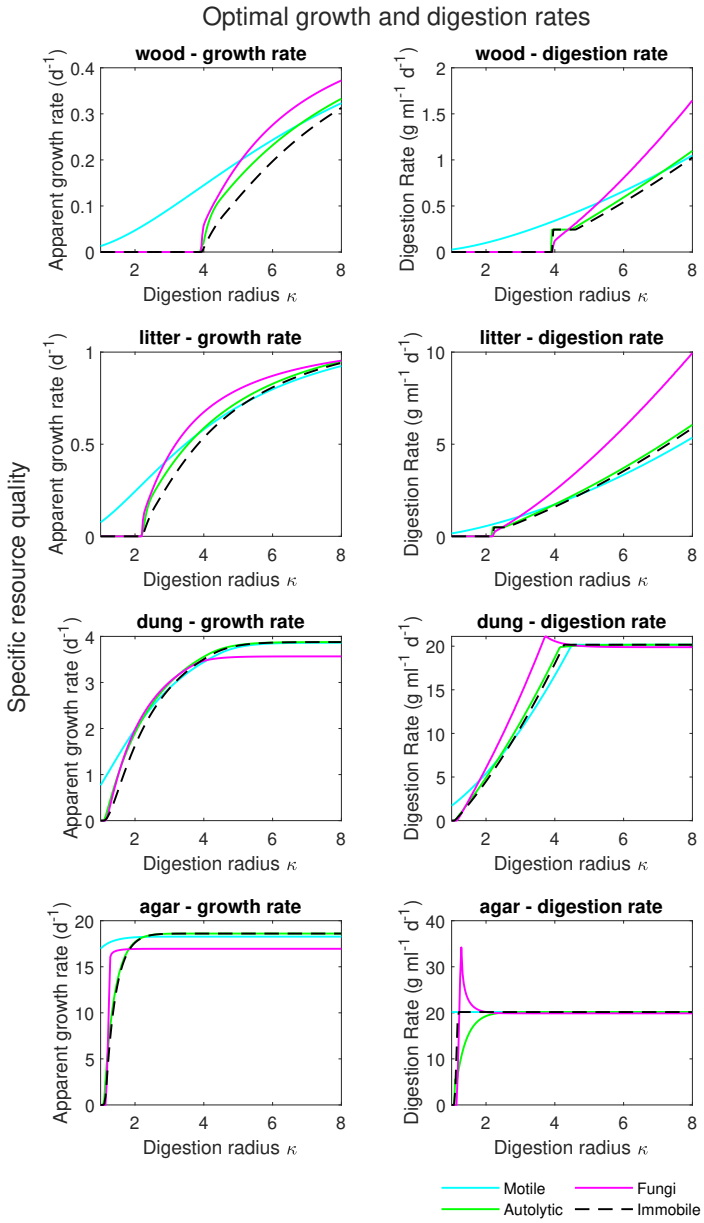


Figure 2.10: Colour-coded maps showing the relative growth of a hyphal organism to a specific class of unicellular organism.

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. However, the only effect in practice is to reduce the overall growth rate for all classes of organism, without changing their relative performance.

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.

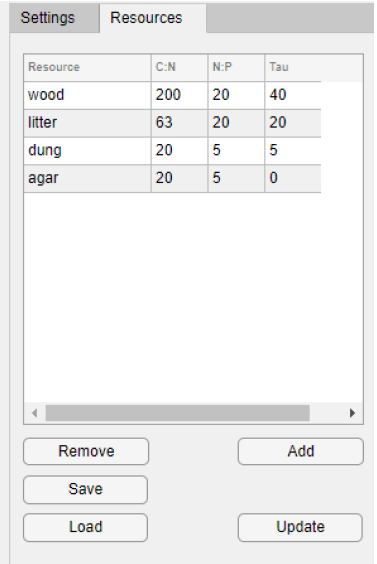


Figure 2.11: Resource table to allow addition of specific resources with user-defined C:N:P ratios and recalcitrance (Tau)