The Fungi Decomposition Rate Model

Summary

Keywords: Python; Mathematical modelling; Fungi.

The Fungi Decomposition Rate Model

Miao Fangran, Wang Xizhi, Wang Zhuo

February 9, 2021

Summary

Keywords: Python; Mathematical modelling; Fungi.

Contents

1 Introduction

1.1 Problem Background

Carbon cycle is a significant process in biosphere. It is the material circulation chain between the inorganic environment and organic organisms that keeps the balance of carbon dioxide in the atmosphere. The breakdown of organic material is a key component of the carbon cycle. It describes the exchange of carbon throughout the geochemical cycle, where the microorganisms play an important role. Fungi is a typical microorganism which is vital to decompose the plant material and woody fiber. It is significant to do meticulous study to fungus decomposition to the environment.

1.2 Literature Review

Fungal ecology is believed to play a crucial role in shaping Earths ecosystems because of fungis ability to recycle nutrients in the system as decomposers[?], provide resources as symbionts of the majority of land plants[?], and impact the success and dynamics of other organisms as pathogens[?, ?] It is significant to do meticulous study to fungus decomposition to the environment. By means of understanding the influence factor to the decomposition rate of the fungus thoroughly, scholar can apply it into other microorganism, and it has positive effect on ecosystem research. Recent research development have revealed that there are a considerable number of traits of fungi related to decomposition rate and certain traits link with each other[?]. In that paper, several characters of fungi were discussed in order to determine the influence of wood decomposition. Researchers studied numerous traits associated with each fungus and tried to determine the role of these traits in wood block decomposition. In particular, the researchers found that are better able to adapt to different ranges of water conditions also tend to decompose wood more slowly. Fungi that grow faster and outperform other fungi tend to decompose wood faster[?].

1.3 Our work

This paper will concentrate on two traits, the growth rate of fungus and the fungus tolerance to moisture. Based on statistic data of three fungi species, we match this data with a same form of formula by interpolation and matching, which can describe the relationship between the fungus growth rate, the fungus tolerance to moisture and the decomposition rate accurately with a certain range. Then this paper combines different species fungus together observe the composite effect of the decomposition to the ground litter. It assigns a value to fungus called ranking. This is the value to describe the competitive capacity of fungus, and we research the interaction between different species of fungus. To research how the fungus and environment have influence with each other, This paper introduces more variables with respect to the environment and revise the formula concerned with the decomposition rate. The effect of different environmental conditions on the decomposition rate of fungi were described more specifically. And the analysis of different species combination to the environment change indicates the importance of the biodiversity. So this paper finally analyze the importance of the biodiversity to the whole environment when different degree of variations occur. It will make some sense in the field of ecology.

2   Preparation of the Models

2.1   Assumptions

Here are our assumptions:

For a single species of fungus[**?**]:

- Under an ideal environment, which means that the temperature is perfect and the food is sufficient, the decomposition rate of fungi is mainly related to the extension rate and the moisture trade-off.

- The decomposition rate has a linear relationship with extension rate.

- The logarithm of decomposition rate has a linear relationship with moisture trade-off.

- If the environment is not so perfect, we introduce one parameter to adjust the decomposition rate—temperature.

With such condition, we can conclude two partial differential equations:

$$\begin{cases} \dfrac{\partial f}{\partial x} & = C_1 \\[2mm] \dfrac{\partial (\log f)}{\partial y} & = C_2 \end{cases} \tag{1}$$

Where f represents the decomposition rate, x represents extension rate and y represents the moisture trade-off. C_1, C_2 represents the linear slope between the independent variables and dependent variables.

Solve the equation, we acquire that:

$$f = Ax \cdot e^{By} \tag{2}$$

where A and B are constants, which varies with the type of fungi.

To adjust the decomposition rate with temperature, we introduce the Temperature function, which is $u(T)$. Thus:

$$f = u(T) \cdot Ax \cdot e^{By} \tag{3}$$

When it comes to multiple fungi, we put forward our fifth assumption:

- The decomposition rate will change when the number of fungus changes.

Then, we also introduce another function of one specific fungus' number:

$$f = v(P) \cdot u(T) \cdot Ax \cdot e^{By} \tag{4}$$

The last problem is that how to measure the decomposition rate. Martin Witkamp[**?**] measured it by calculating the emission of carbon dioxide. It reminded us of different measure method. And what we need to do is to uniform the standard. Thus, the last assumption is:

- Every fungus decomposition rate can be measured by the amount of carbon dioxide released, and can also be translated into the mass loss of the raw material.

After that, we re-write f in the form of dm/dt, where m representsLustenhouwer11551 mass and t represents time.

2.2 Data Collection

It is really hard to collect data for there are a lot of literature with different measurement indexes. Thanks to the paper from Daniel et al.[**?**], we get lots of useful data. The relationship between fungi's decomposition rate and moisture niche width and the relationship between the decomposition rate and temperature. Then uniform the standard—Express the rate of decomposition with the rate of mass loss.

Repeat such process with 5-6 papers, we finally get three fungi's data.

- For fungus 1, $A = 2.123, B = -0.4249$

- For fungus 2, $A = 1.877, B = 0.5152$

- For fungus 3, $A = 3.553, B = 0.7636$

For the correction factor, temperature and fungi's number, please check out in Appendix B for more information. We created correction factors for each species.

3 The Models

To build up the model, we figure out that it is easy to use object-oriented programming.

We firstly set different objects for fungi. The fungi itself have many attributes: the inherent coefficients A and B, the extension rate, the decomposition rate, the number of fungi.

3.1 Model 1

In this model, we try to figure out the decomposition rate of one specific fungus against time under proper temperature.

3.2 Model 2

In this model, we try to figure out the interactions between the fungi.

4 Strengths and weaknesses

4.1 Strengths

In this model, we set up two mutually related models, one is a single species, and the other is the community combination of multiple species. Not only can we clearly see the change in the rate of decomposition of individual species. There is also competition and dependence among multiple species, as well as changes in the rate of decomposition under community conditions. We can also

clearly see that the decomposition rate of bacteria increases steadily at the beginning, and then after the density of species is saturated, the decomposition rate begins to decline rapidly.

At the same time, we found many complex and effective and correct data, obtained our final mathematical formula through partial differential equations, and obtained models and images with tools such as MATLAB and Python, which were scientific and effective.

4.2 Weaknesses

- We can't consider the death rate of each fungus, which means they grow in an almost ideal environment.

- The environment has lots of factors that can influence the decomposition rate of fungi, we just consider the moisture and the temperature. For instance, the light intensity can influence.

- We were only looking at fungi that break down carbon, however, there are also fungi can break down nitrogen.

- The shape of the colony may also affect the decomposition rate.

Appendices

Appendix A   First appendix

Appendix B   Second appendix

The class of **Fungi** and **environment** are as below:

```python
import math


class Fungi1:  # Middle temperature most fast decomposition rate
    c1 = 2.123
    c2 = -0.4249

    def __init__(self, extension_rate, environment):
        self.extension_rate = extension_rate
        self.moisture = environment.moisture
        self.dr = 0.0  # decomposition rate
        self.number = 1.0  # The total number

    def set_number(self, time_interval):
        self.number = self.number + self.extension_rate * time_interval

    def set_dr(self, environment):
        self.dr = self.c1 * self.extension_rate * math.exp(self.c2 * self.moisture)
```

```python
        self.dr = self.dr * math.log2(self.number)

        if 40 > environment.temperature > 25:
            self.dr = self.dr * 0.8
        elif 15 < environment.temperature <= 25:
            self.dr = self.dr * 1.0
        elif 10 <= environment.temperature < 15:
            self.dr = self.dr * 0.6
        elif 0 < environment.temperature < 10:
            self.dr = self.dr * 0.35
        elif 0 >= environment.temperature:
            self.dr = self.dr * 0.1
        elif environment.temperature > 40:
            self.dr = self.dr * 0.4

        if environment.wood_number < (self.number * 2):
            self.dr = self.dr * (environment.wood_number / (self.number * 2))


class Fungi2:  # High temperature most fast decomposition rate
    c1 = 1.877
    c2 = 0.5152

    def __init__(self, extension_rate, environment):
        self.extension_rate = extension_rate
        self.moisture = environment.moisture
        self.dr = 0.0  # decomposition rate
        self.number = 1.0  # The total number

    def set_number(self, time_interval):
        self.number = self.number + self.extension_rate * time_interval

    def set_dr(self, environment):
        self.dr = self.c1 * self.extension_rate * math.exp(self.c2 * self.moisture)

        self.dr = self.dr * math.log2(self.number)

        if 35 > environment.temperature > 25:
            self.dr = self.dr * 1.0
        elif 15 < environment.temperature <= 25:
            self.dr = self.dr * 0.85
        elif 10 <= environment.temperature < 15:
            self.dr = self.dr * 0.6
        elif 0 < environment.temperature < 10:
            self.dr = self.dr * 0.3
        elif 0 >= environment.temperature:
            self.dr = self.dr * 0.1
        elif environment.temperature > 35:
            self.dr = self.dr * 0.8

        if environment.wood_number < (self.number * 2):
            self.dr = self.dr * (environment.wood_number / (self.number * 2))
```

```python
class Fungi3:  # Low temperature most fast decomposition rate
    c1 = 3.553
    c2 = 0.7636

    def __init__(self, extension_rate, environment):
        self.extension_rate = extension_rate
        self.moisture = environment.moisture
        self.dr = 0.0  # decomposition rate
        self.number = 1.0  # The total number

    def set_number(self, time_interval):
        self.number = self.number + self.extension_rate * time_interval

    def set_dr(self, environment):
        self.dr = self.c1 * self.extension_rate * math.exp(self.c2 * self.moisture)

        self.dr = self.dr * math.log2(self.number)

        if 40 > environment.temperature > 25:
            self.dr = self.dr * 0.65
        elif 15 < environment.temperature <= 25:
            self.dr = self.dr * 0.85
        elif 10 <= environment.temperature < 15:
            self.dr = self.dr * 1.0
        elif 0 < environment.temperature < 10:
            self.dr = self.dr * 0.8
        elif 0 >= environment.temperature:
            self.dr = self.dr * 0.3
        elif environment.temperature > 40:
            self.dr = self.dr * 0.4

        if environment.wood_number < (self.number * 2):
            self.dr = self.dr * (environment.wood_number / (self.number * 2))


class Environment:
    def __init__(self, moisture, temperature, wood_number):
        self.moisture = moisture
        self.temperature = temperature
        self.wood_number = wood_number
        self.fungi1_density = self.fungi2_density = self.fungi3_density = 1 / 3

    def set_moisture(self, new_moisture):
        self.moisture = new_moisture

    def set_temperature(self, new_temperature):
        self.temperature = new_temperature

    def set_fungi_density(self, fungi1, fungi2, fungi3):
        total = fungi1.number + fungi2.number + fungi3.number
        self.fungi1_density = fungi1.number / total
        self.fungi2_density = fungi2.number / total
        self.fungi3_density = fungi3.number / total
```

```
    def set_wood_number(self, fungi1, fungi2, fungi3, time_interval):
        self.wood_number = self.wood_number - fungi1.dr * time_interval - fungi2.dr * time_
```

The **main code** is as below:

```python
from Fungi import Fungi1, Fungi2, Fungi3, Environment
import sys
import matplotlib.pyplot as plt
import numpy as np

if __name__ == '__main__':
    environment = Environment(0.61, 26, 100)
    fungi1 = Fungi1(4.71, environment)
    fungi2 = Fungi2(4.11, environment)
    fungi3 = Fungi3(3.77, environment)
    a = []
    b = []
    c = []
    d = []
    e = []
    f = []
    g = []
    t = 0
    for i in range(sys.maxsize ** 10):
        fungi1.set_dr(environment)
        fungi1.set_number(0.001)
        fungi2.set_dr(environment)
        fungi2.set_number(0.001)
        fungi3.set_dr(environment)
        fungi3.set_number(0.001)
        environment.set_fungi_density(fungi1, fungi2, fungi3)
        environment.set_wood_number(fungi1, fungi2, fungi3, 0.001)
        if environment.wood_number <= 1:
            break
        t = t + 1
        a.append(fungi1.dr)
        b.append(environment.fungi1_density)
        c.append(fungi2.dr)
        d.append(environment.fungi2_density)
        e.append(fungi3.dr)
        f.append(environment.fungi3_density)
        g.append(environment.wood_number)
    print(a)
    print(b)
    print(c)
    print(d)
    print(e)
    print(f)
    print(g)
    print(t)
    x = np.arange(0, t * 0.01-0.005, 0.01)
    plt.subplot(2, 1, 1)
    plt.plot(x, a)
    plt.plot(x, c)
```

```
plt.plot(x, e)
plt.ylabel('Decomposition rate')
plt.title('Fungi')
plt.legend(['Fungi1', 'Fungi2', 'Fungi3'])
plt.subplot(2, 1, 2)
plt.plot(x, b)
plt.plot(x, d)
plt.plot(x, f)
plt.xlabel('time')
plt.ylabel('Fungi-density')
plt.legend(['Fungi1', 'Fungi2', 'Fungi3'])
plt.show()
```