

## **Msc Bioinformatics thesis**

# **Study of Division of Labor in Pseudomonas through single-cell RNA-seq**

Valentin Goupille

Master 2 in Bioinformatics

Academic Year: 2024-2025

Internship conducted at Ecobio UMR 6553 CNRS-University of Rennes



Ecobio UMR 6553 CNRS-University of Rennes

Campus de Beaulieu, 35042 Rennes Cedex, France

Under the supervision of:

Solène Mauger-Franklin, Postdoctoral Researcher

Philippe Vandenkoornhuyse, Professor

Presented on 2025-07-01



# Table of contents

Copyright notice	v
Declaration	vi
Abstract	vii
Acknowledgements	viii
List of Abbreviations	x
List of Figures	xi
List of Tables	xii
<b>1 Introduction</b>	<b>1</b>
1.1 Litterature review . . . . .	1
<b>2 Materials and Methods</b>	<b>3</b>
2.1 Materials . . . . .	3
<b>3 Results</b>	<b>4</b>
3.1 Overview . . . . .	4
3.2 Single-cell RNA-seq Analysis . . . . .	4
3.3 Functional Analysis . . . . .	4
3.4 Integration with Previous Studies . . . . .	4
3.5 Summary of Key Findings . . . . .	4
<b>4</b>	<b>5</b>
<b>5 Conclusion and Future Work</b>	<b>14</b>
5.1 Summary of Main Findings . . . . .	14
5.2 Impact on the Field . . . . .	14
5.3 Future Research Directions . . . . .	14
5.4 Final Remarks . . . . .	14
5.5 References . . . . .	14
<b>Bibliography</b>	<b>15</b>
<b>Appendices</b>	<b>16</b>
<b>A</b>	<b>16</b>

---

<b>B</b>	<b>Annexe B: erferfrefref</b>	<b>17</b>
<b>C</b>	<b>Annexe C: codcefe</b>	<b>18</b>

# Copyright notice

Produced on 11 April 2025.

© Valentin Goupille (2025).

# Declaration

## Statement of originality



I, the undersigned, **Valentin Goupille**, a student in the **Master's program in Bioinformatics**, hereby declare that I am fully aware that plagiarism of documents or parts of documents published on any type of medium, including the internet, constitutes a violation of copyright laws as well as an act of fraud.

As a result, I commit to citing all the sources I have used in the writing of this document.

Date : **01/04/2025**

Signature :

A handwritten signature in black ink, consisting of a stylized 'V' and 'G' enclosed within an oval.

## Reproducibility statement

This thesis is written using Quarto. All materials (including the data sets and source files) required to reproduce this document can be found at the Github repository [github.com/vgoupille/Internship\\_2025](https://github.com/vgoupille/Internship_2025).

This work is licensed under a [Attribution-NonCommercial-NoDerivatives 4.0 International License](https://creativecommons.org/licenses/by-nc-nd/4.0/).



# Abstract

## **Study of *Pseudomonas brassicacearum* gene expression variation in environmental constraints, towards the validation of Division Of Labor.**

Duis ornare ex ac iaculis pretium. Maecenas sagittis odio id erat pharetra, sit amet consectetur quam sollicitudin. Vivamus pharetra quam purus, nec sagittis risus pretium at. Nullam feugiat, turpis ac accumsan interdum, sem tellus blandit neque, id vulputate diam quam semper nisl. Donec sit amet enim at neque porttitor aliquet. Phasellus facilisis nulla eget placerat eleifend. Vestibulum non egestas eros, eget lobortis ipsum. Nulla rutrum massa eget enim aliquam, id porttitor erat luctus. Nunc sagittis quis eros eu sagittis. Pellentesque dictum, erat at pellentesque sollicitudin, justo augue pulvinar metus, quis rutrum est mi nec felis. Vestibulum efficitur mi lorem, at elementum purus tincidunt a. Aliquam finibus enim magna, vitae pellentesque erat faucibus at. Nulla mauris tellus, imperdiet id lobortis et, dignissim condimentum ipsum. Morbi nulla orci, varius at aliquet sed, facilisis id tortor. Donec ut urna nisi.

Aenean placerat luctus tortor vitae molestie. Nulla at aliquet nulla. Sed efficitur tellus orci, sed fringilla lectus laoreet eget. Vivamus maximus quam sit amet arcu dignissim, sed accumsan massa ullamcorper. Sed iaculis tincidunt feugiat. Nulla in est at nunc ultricies dictum ut vitae nunc. Aenean convallis vel diam at malesuada. Suspendisse arcu libero, vehicula tempus ultrices a, placerat sit amet tortor. Sed dictum id nulla commodo mattis. Aliquam mollis, nunc eu tristique faucibus, purus lacus tincidunt nulla, ac pretium lorem nunc ut enim. Curabitur eget mattis nisl, vitae sodales augue. Nam felis massa, bibendum sit amet nulla vel, vulputate rutrum lacus. Aenean convallis odio pharetra nulla mattis consequat.

### **Keywords :**

Single-cell RNA-seq, *Pseudomonas brassicacearum*, Division Of Labor, (4-5 keywords) bacterial population, metabolism, specialization, root colonization

# Acknowledgements

I would like to thank ... Ecobio ANR Divide

In accordance with Chapter 7.1.4 of the research degrees handbook, if you have engaged the services of a professional editor, you must provide their name and a brief description of the service rendered. If the professional editor's current or former area of academic specialisation is similar your own, this too should be stated as it may suggest to examiners that the editor's advice to the student has extended beyond guidance on English expression to affect the substance and structure of the thesis.

If you have used generative artificial intelligence (AI) technologies, you must include a written acknowledgment of the use and its extent. Your acknowledgement should at a minimum specify which technology was used, include explicit description on how the information was generated, and explain how the output was used in your work. Below is a suggested format:

"I acknowledge the use of [insert AI system(s) and link] to [specific use of generative artificial intelligence]. The output from these was used to [explain use]."

Free text section for you to record your acknowledgment and gratitude for the more general academic input and support such as financial support from grants and scholarships and the non-academic support you have received during the course of your enrolment. If you are a recipient of the "Australian Government Research Training Program Scholarship", you are required to include the following statement:

"This research was supported by an Australian Government Research Training Program (RTP) Scholarship."

You may also wish to acknowledge significant and substantial contribution made by others to the research, work and writing represented and/or reported in the thesis. These could include significant contributions to: the conception and design of the project; non-routine



technical work; analysis and interpretation of research data; drafting significant parts of the work or critically revising it to contribute to the interpretation.

# List of Abbreviations

Abbreviation	Definition
AI	Artificial Intelligence
ANR	Agence Nationale de la Recherche
DNA	Deoxyribonucleic Acid
DOL	Division Of Labor
NGS	Next Generation Sequencing
RNA	Ribonucleic Acid
RNA-seq	RNA sequencing
scRNA-seq	single-cell RNA sequencing

# List of Figures

# List of Tables

# Chapter 1

## Introduction

### 1.1 Literature review

deddfefde<sup>1,2</sup> Internship description

The survival of organisms in evolving environments is driven by their fitness. The cost-benefit ratio of traits is constantly balanced and gives rise to different populational evolutionary strategies. To succeed, organisms will have to compete, cooperate and/or specialize as a result of how fit their traits are considering their biotic and abiotic environment. Bacteria are unicellular organisms with therefore little option to specialize and give up certain traits production to limit their metabolic costs, unlike multicellular organisms that present many different forms of specialized cells in one single organism. However, [[auxotrophic bacteria]] (i.e bacteria lacking genes coding for a molecule essential for their survival) have been studied (Morris et al., 2012).<sup>3</sup>

Auxotroph bacteria can take advantage of leaky functions of helper's organisms to fulfill their needs in specific compounds (Morris et al., 2014, Estrela et al., 2016).<sup>4,5</sup> With a reduced genetic material, the beneficiary organism fitness is improved, at the risk of being dependent on the helpers presence in their environment. The conditions in which patterns of such [[division of labor (DOL)]] arise are still obscure, but its advantages for bacterial population are clear: DOL allows to diminish the cost associated to certain functions and the possibility of cohabitation of various mutants/specialized cells within the population to respond as a whole to environmental constraints, and thrive. New technologies allow us to access within-species diversity and study the possible metabolic specialization between cells. Single-cell -omics have been developed for this purpose in human health and are now applied to microbial systems. However, analyzing such datasets still requires custom pipelines to respond to the specificity of bacterial biology and technical challenges.

---

The goal of this internship is to explore [[scRNA-seq]] (single-cell RNA-seq) datasets of [[*Pseudomonas brassicacearum*]], a root colonizer. The student will analyse samples datasets from various nutritional conditions to determine if DOL can be detected within this species as a strategy for efficient root colonization. The intern will have to implement transcriptomic data analyses from ultra-high throughput sequence run(s). Thus the main aim of the intern will be to set up bioinformatic workflow(s) from existing tools to produce interpretable results.

## **Chapter 2**

# **Materials and Methods**

### **2.1 Materials**

# Chapter 3

## Results

- figure de % de type of RNA

### 3.1 Overview

This chapter presents the findings of our single-cell RNA-seq analysis of *Pseudomonas*, focusing on the division of labor within bacterial populations.

### 3.2 Single-cell RNA-seq Analysis

#### 3.2.1 Data Quality and Preprocessing

#### 3.2.2 Cell Type Identification

#### 3.2.3 Differential Expression Analysis

#### 3.2.4 Division of Labor Patterns

### 3.3 Functional Analysis

#### 3.3.1 Pathway Enrichment

#### 3.3.2 Gene Set Analysis

#### 3.3.3 Regulatory Network Analysis

### 3.4 Integration with Previous Studies

### 3.5 Summary of Key Findings



## Chapter 4

```
from plotly.basedatatypes import BaseTraceHierarchyType as _BaseTraceHierarchyType import copy
as _copy
```

```
class Line(_BaseTraceHierarchyType):
```

```
    # class properties
```

```
    # -----
```

```
    _parent_path_str = "violin.marker"
```

```
    _path_str = "violin.marker.line"
```

```
    _valid_props = {"color", "outliercolor", "outlierwidth", "width"}
```

```
    # color
```

```
    # -----
```

```
    @property
```

```
    def color(self):
```

```
        """
```

```
        Sets the marker.line color. It accepts either a specific color
        or an array of numbers that are mapped to the colorscale
        relative to the max and min values of the array or relative to
        `marker.line.cmin` and `marker.line.cmax` if set.
```

The 'color' property is a color and may be specified as:

- A hex string (e.g. '#ff0000')
- An rgb/rgba string (e.g. 'rgb(255,0,0)')

- An hsl/hsla string (e.g. 'hsl(0,100%,50%)')
- An hsv/hsva string (e.g. 'hsv(0,100%,100%)')
- A named CSS color:
  - aliceblue, antiquewhite, aqua, aquamarine, azure,
  - beige, bisque, black, blanchedalmond, blue,
  - blueviolet, brown, burlywood, cadetblue,
  - chartreuse, chocolate, coral, cornflowerblue,
  - cornsilk, crimson, cyan, darkblue, darkcyan,
  - darkgoldenrod, darkgray, darkgrey, darkgreen,
  - darkkhaki, darkmagenta, darkolivegreen, darkorange,
  - darkorchid, darkred, darksalmon, darkseagreen,
  - darkslateblue, darkslategray, darkslategrey,
  - darkturquoise, darkviolet, deeppink, deepskyblue,
  - dimgray, dimgrey, dodgerblue, firebrick,
  - floralwhite, forestgreen, fuchsia, gainsboro,
  - ghostwhite, gold, goldenrod, gray, grey, green,
  - greenyellow, honeydew, hotpink, indianred, indigo,
  - ivory, khaki, lavender, lavenderblush, lawngreen,
  - lemonchiffon, lightblue, lightcoral, lightcyan,
  - lightgoldenrodyellow, lightgray, lightgrey,
  - lightgreen, lightpink, lightsalmon, lightseagreen,
  - lightskyblue, lightslategray, lightslategrey,
  - lightsteelblue, lightyellow, lime, limegreen,
  - linen, magenta, maroon, mediumaquamarine,
  - mediumblue, mediumorchid, mediumpurple,
  - mediumseagreen, mediumslateblue, mediumspringgreen,
  - mediumturquoise, mediumvioletred, midnightblue,
  - mintcream, mistyrose, moccasin, navajowhite, navy,
  - oldlace, olive, olivedrab, orange, orangered,
  - orchid, palegoldenrod, palegreen, paleturquoise,
  - palevioletred, papayawhip, peachpuff, peru, pink,
  - plum, powderblue, purple, red, rosybrown,
  - royalblue, rebeccapurple, saddlebrown, salmon,
  - sandybrown, seagreen, seashell, sienna, silver,

```
skyblue, slateblue, slategray, slategrey, snow,  
springgreen, steelblue, tan, teal, thistle, tomato,  
turquoise, violet, wheat, white, whitesmoke,  
yellow, yellowgreen
```

Returns

-----

str

"""

```
return self["color"]
```

@color.setter

```
def color(self, val):
```

```
    self["color"] = val
```

# outliercolor

# -----

@property

```
def outliercolor(self):
```

"""

Sets the border line color of the outlier sample points.

Defaults to marker.color

The 'outliercolor' property is a color and may be specified as:

- A hex string (e.g. '#ff0000')
- An rgb/rgba string (e.g. 'rgb(255,0,0)')
- An hsl/hsla string (e.g. 'hsl(0,100%,50%)')
- An hsv/hsva string (e.g. 'hsv(0,100%,100%)')
- A named CSS color:

```
aliceblue, antiquewhite, aqua, aquamarine, azure,  
beige, bisque, black, blanchedalmond, blue,  
blueviolet, brown, burlywood, cadetblue,  
chartreuse, chocolate, coral, cornflowerblue,  
cornsilk, crimson, cyan, darkblue, darkcyan,
```

darkgoldenrod, darkgray, darkgrey, darkgreen,  
darkkhaki, darkmagenta, darkolivegreen, darkorange,  
darkorchid, darkred, darksalmon, darkseagreen,  
darkslateblue, darkslategray, darkslategrey,  
darkturquoise, darkviolet, deeppink, deepskyblue,  
dimgray, dimgrey, dodgerblue, firebrick,  
floralwhite, forestgreen, fuchsia, gainsboro,  
ghostwhite, gold, goldenrod, gray, grey, green,  
greenyellow, honeydew, hotpink, indianred, indigo,  
ivory, khaki, lavender, lavenderblush, lawngreen,  
lemonchiffon, lightblue, lightcoral, lightcyan,  
lightgoldenrodyellow, lightgray, lightgrey,  
lightgreen, lightpink, lightsalmon, lightseagreen,  
lightskyblue, lightslategray, lightslategrey,  
lightsteelblue, lightyellow, lime, limegreen,  
linen, magenta, maroon, mediumaquamarine,  
mediumblue, mediumorchid, mediumpurple,  
mediumseagreen, mediumslateblue, mediumspringgreen,  
mediumturquoise, mediumvioletred, midnightblue,  
mintcream, mistyrose, moccasin, navajowhite, navy,  
oldlace, olive, olivedrab, orange, orangered,  
orchid, palegoldenrod, palegreen, paleturquoise,  
palevioletred, papayawhip, peachpuff, peru, pink,  
plum, powderblue, purple, red, rosybrown,  
royalblue, rebeccapurple, saddlebrown, salmon,  
sandybrown, seagreen, seashell, sienna, silver,  
skyblue, slateblue, slategray, slategrey, snow,  
springgreen, steelblue, tan, teal, thistle, tomato,  
turquoise, violet, wheat, white, whitesmoke,  
yellow, yellowgreen

Returns

-----

str

```
"""
    return self["outliercolor"]

@outliercolor.setter
def outliercolor(self, val):
    self["outliercolor"] = val

# outlierwidth
# -----

@property
def outlierwidth(self):
    """
    Sets the border line width (in px) of the outlier sample
    points.

    The 'outlierwidth' property is a number and may be specified as:
        - An int or float in the interval [0, inf]

    Returns
    -----
    int|float
    """
    return self["outlierwidth"]

@outlierwidth.setter
def outlierwidth(self, val):
    self["outlierwidth"] = val

# width
# -----

@property
def width(self):
    """
    Sets the width (in px) of the lines bounding the marker points.
```

The 'width' property is a number and may be specified as:

- An int or float in the interval [0, inf]

Returns

-----

int|float

"""

return self["width"]

@width.setter

def width(self, val):

self["width"] = val

# Self properties description

# -----

@property

def \_prop\_descriptions(self):

return ""\n

color

Sets the marker.line color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `marker.line.cmin` and `marker.line.cmax` if set.

outliercolor

Sets the border line color of the outlier sample points. Defaults to marker.color

outlierwidth

Sets the border line width (in px) of the outlier sample points.

width

Sets the width (in px) of the lines bounding the marker points.

```
"""

def __init__(
    self,
    arg=None,
    color=None,
    outliercolor=None,
    outlierwidth=None,
    width=None,
    **kwargs,
):
    """
    Construct a new Line object

    Parameters
    -----
    arg
        dict of properties compatible with this constructor or
        an instance of
        :class:`plotly.graph_objs.violin.marker.Line`
    color
        Sets the marker.line color. It accepts either a
        specific color or an array of numbers that are mapped
        to the colorscale relative to the max and min values of
        the array or relative to `marker.line.cmin` and
        `marker.line.cmax` if set.
    outliercolor
        Sets the border line color of the outlier sample
        points. Defaults to marker.color
    outlierwidth
        Sets the border line width (in px) of the outlier
        sample points.
    width
        Sets the width (in px) of the lines bounding the marker
```

points.

Returns

-----

Line

"""

```
super(Line, self).__init__("line")
```

```
if "_parent" in kwargs:
```

```
    self._parent = kwargs["_parent"]
```

```
    return
```

```
# Validate arg
```

```
# -----
```

```
if arg is None:
```

```
    arg = {}
```

```
elif isinstance(arg, self.__class__):
```

```
    arg = arg.to_plotly_json()
```

```
elif isinstance(arg, dict):
```

```
    arg = _copy.copy(arg)
```

```
else:
```

```
    raise ValueError(
```

```
        """\
```

The first argument to the `plotly.graph_objs.violin.marker.Line` constructor must be a dict or an instance of `:class:plotly.graph_objs.violin.marker.Line`""")

```
# Handle skip_invalid
```

```
# -----
```

```
self._skip_invalid = kwargs.pop("skip_invalid", False)
```

```
self._validate = kwargs.pop("_validate", True)
```

```
# Populate data dict with properties
```

```
# -----
```

```
_v = arg.pop("color", None)
```

```
_v = color if color is not None else _v
```



```
if _v is not None:
    self["color"] = _v
_v = arg.pop("outliercolor", None)
_v = outliercolor if outliercolor is not None else _v
if _v is not None:
    self["outliercolor"] = _v
_v = arg.pop("outlierwidth", None)
_v = outlierwidth if outlierwidth is not None else _v
if _v is not None:
    self["outlierwidth"] = _v
_v = arg.pop("width", None)
_v = width if width is not None else _v
if _v is not None:
    self["width"] = _v

# Process unknown kwargs
# -----
self._process_kwargs(**dict(arg, **kwargs))

# Reset skip_invalid
# -----
self._skip_invalid = False
```

## **Chapter 5**

# **Conclusion and Future Work**

### **5.1 Summary of Main Findings**

#### **5.1.1 Key Discoveries**

#### **5.1.2 Methodological Contributions**

#### **5.1.3 Biological Insights**

### **5.2 Impact on the Field**

#### **5.2.1 Contribution to Single-cell RNA-seq Methodology**

#### **5.2.2 Contribution to Pseudomonas Research**

#### **5.2.3 Broader Implications for Microbial Ecology**

### **5.3 Future Research Directions**

#### **5.3.1 Technical Improvements**

#### **5.3.2 Biological Questions to Address**

#### **5.3.3 Potential Applications**

### **5.4 Final Remarks**

### **5.5 References**

# Bibliography

1. Kuchina, A. *et al.* [Microbial single-cell RNA sequencing by split-pool barcoding](#). *Science* **371**, eaba5257 (2021).
2. Gaisser, K. D. *et al.* [High-throughput single-cell transcriptomics of bacteria using combinatorial barcoding](#). *Nature Protocols* **19**, 3048–3084 (2024).
3. Morris, J. J., Lenski, R. E. & Zinser, E. R. [The black queen hypothesis: Evolution of dependencies through adaptive gene loss](#). *mBio* **3**, e00036–12 (2012).
4. Morris, E. K. *et al.* [Choosing and using diversity indices: insights for ecological applications from the German Biodiversity Exploratories](#). *Ecology and Evolution* **4**, 3514–3524 (2014).
5. Estrela, S., Kerr, B. & Morris, J. J. [Transitions in individuality through symbiosis](#). *Current Opinion in Microbiology* **31**, 191–198 (2016).

## **Appendix A**

## **Appendix B**

### **Annexe B: erferfrefref**

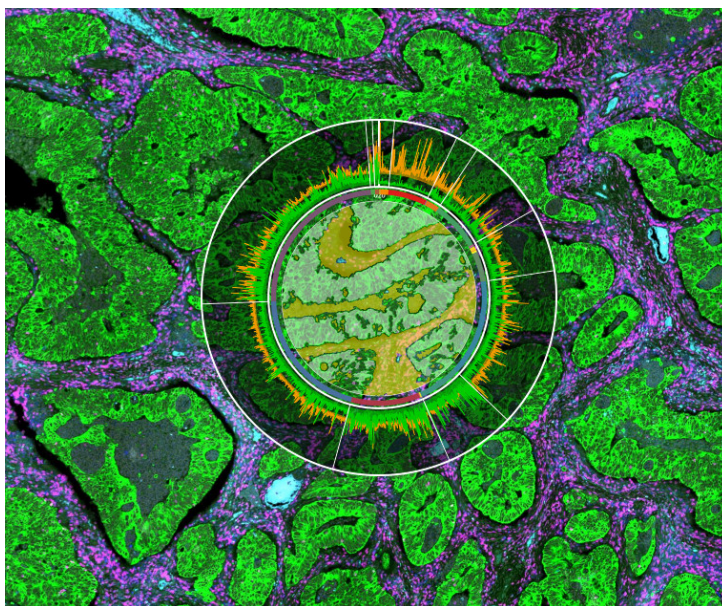
## **Appendix C**

### **Annexe C: codcefe**



# Master's Thesis in Bioinformatics

University of Rennes



*This thesis was conducted in the framework of the Master's program in Bioinformatics at the University of Rennes. The research presented here contributes to the field of computational biology and bioinformatics.*

© Valentin Goupille - ?meta:year

*All rights reserved*