

# Polyglot programming for single-cell analysis

Louise Deconinck Benjamin Rombaut Robrecht Cannoodt

2024-09-12

# Introduction

## **File formats**

# Calling Python from R and vice versa

or: in-memory interoperability

## Overview

reticulate:

1. Call Python in R
2. embed a Python session within your R session

basilisk allows managing Python environments within the BioConductor ecosystem

rpy2:

1. Call R in Python
2. run R in a Python process

## in-memory interoperability advantages

no need to write & read results

useful when you need a limited amount of functions in another language

## in-memory interoperability drawbacks

no access to classes

you need to extract necessary matrices & arrays for the method

ensure that the method accepts this

you need to be familiar with using & managing both environments

data duplication

## accessing R from Python

rpy2

Jupyter notebooks: - Use IPython magic interface - most useful for matrices & arrays

e.g. `%%R -i input -o output`

use `anndata2ri`: converts `anndata` objects to `SingleCellExperiment`



# accessing Python from R

reticulate basilisk

# **Package-based interoperability**

or: the question of reimplementation

# Workflows

# Takeaways