BiocPy: Facilitate Bioconductor Workflows in Python

Jayaram Kancherla

2026-01-14

Table of contents

1 Welcome to BiocPy **3**
1.1 About this Book 3
1.2 Further Reading 4

2 BiocPy Packages **5**
2.1 Getting Started 5
2.1.1 Installation 5
2.2 Core Representations 5
2.3 Analysis Packages 6
2.4 R Interoperability 6
2.5 Developer Guide 8

1 Welcome to BiocPy

> Looking for the older website?
>
> A previous version of this book is published here and the Bioconductor workshop here (from 2024).

BiocPy is designed to build a bridge between the mature Bioconductor ecosystem and the Python landscape. Bioconductor is an open-source software project that provides tools for the analysis and comprehension of genomic data. One of the main advantages of Bioconductor is the availability of standard data representations and large number of analysis tools tailored for genomic experiments. These data structures allow researchers to seamlessly store, manipulate, and analyze data across multiple packages and workflows in R.

Inspired by Bioconductor, BiocPy aims to facilitate Bioconductor workflows in Python. To achieve this goal, we developed several core data structures that align closely to the Bioconductor implementations. By implementing these core Bioconductor data structures, BiocPy allows data to be easily interoperable between R and Python.

1.1 About this Book

This book is currently in ***active development*** and is organized into the following sections:

- **Foundations:** The core data structures that underpin the ecosystem.
  - GenomicRanges: Manipulation of genomic intervals.
  - Data Containers: Rich semantic data containers like `SummarizedExperiment`.
- **BioC Hubs:** Access to Bioconductor's cloud resources.
  - ExperimentHub: Discover and download datasets.
  - AnnotationHub: Interact with gene models (`TxDb`) and organism databases (`OrgDb`).
- **Interoperability:** Tools to bridge R and Python.
  - R Interoperability: Read R data files (`RDS`) directly in Python.
  - ArtifactDB: Language-agnostic storage for genomic data in R and Python.
- **Workflows:** Real-world usage examples.

  - Single-Cell Analysis: Multi-modal single-cell analysis with `scranpy`.
  - Annotation: Automated cell type annotation with `singler`.

1.2 Further Reading

Many online resources offer detailed information on Bioconductor data structures, namely:

- CompGenomR Book
- Orchestrating Single-Cell Analysis with Bioconductor
- Orchestrating Multi-Omics Analysis with Bioconductor
- EPIC-BiocIntro

2 BiocPy Packages

2.1 Getting Started

The BiocPy ecosystem is modular. You can install the collection of core packages via PyPI.

2.1.1 Installation

To get the core data structures and utilities:

```
pip install biocutils genomicranges summarizedexperiment singlecellexperiment multiassayexpe
```

To install interoperability tools:

```
pip install rds2py txdb orgdb experimenthub
```

2.2 Core Representations

These structures serve as essential and foundational data structures, acting as the building blocks for extensive and complex representations.

- **BiocFrame**: A Bioconductor-like data frame class.
- **GenomicRanges**: Aids in representing genomic regions and facilitating analysis.

Container classes represent single or multi-omic experimental data and metadata:

- **SummarizedExperiment**: Container class to represent genomic experiments.
- **SingleCellExperiment**: Container class to represent single-cell experiments.
- **MultiAssayExperiment**: Container class to represent multiple experiments and assays performed over a set of samples.

Moreover, BiocPy introduces a diverse range of data type classes designed to support the representation of atomic entities, including float, string, int lists, and named lists. These generics and utilities are provided through BiocUtils package, and the delayed and file-backed array operations in the DelayedArray and their derivatives (HDF5Array, TileDbArray).

2.3 Analysis Packages

BiocPy provides bindings to libscran and various other single-cell analysis methods incorporated into the **scranpy** package to support analysis of multi-modal single-cell datasets. It also features integration with the **singler** algorithm to annotate cell types by matching cells to known references based on their expression profiles.

2.4 R Interoperability

The **rds2py** package provides bindings to the rds2cpp library. This enables direct reading of RDS files in Python, eliminating the requirement for additional data conversion tools or intermediate formats. The package's functionality streamlines the transition between Python and R, facilitating seamless analysis.

The following table serves as a directory of the core packages in the **BiocPy** ecosystem. All packages within the BiocPy ecosystem are published to Python's Package Index (PyPI).

Package	Description	Latest Version	Links
BiocFrame	Flexible dataframe representation to support nested structures.	`0.7.2`	GitHub
DelayedArray	Delayed array operations from Bioconductor	`0.6.2`	GitHub
GenomicRanges	Container class to represent and operate over genomic regions and annotations.	`0.8.3`	GitHub
IRanges	Python implementation of the **IRanges** Bioconductor package.	`0.7.1`	GitHub
MultiAssayExperiment	Container class for representing and managing multi-omics genomic experiments	`0.6.0`	GitHub
SingleCellExperiment	Container class for single-cell experiments	`0.6.2`	GitHub

Package	Description	Latest Version	Links
SpatialExperiment	Container class for storing data from spatial-omics experiments	`0.1.0`	GitHub
SpatialFeatureExperiment	Container class for storing data from spatial feature experiments	`0.0.5`	GitHub
SummarizedExperiment	Container to represent data from genomic experiments	`0.6.5`	GitHub
biocsetup	"A CLI interface to quickly scaffold new BiocPy Python packages"	`0.3.3`	GitHub
biocutils	Utilities to use across the biocpy packages.	`0.3.3`	GitHub
biostrings	Efficient manipulation of genomic sequences	`0.0.2`	GitHub
celldex	Index of Reference Cell Type Datasets	`0.3.0`	GitHub
compressedlists	Memory-efficient container for list-like objects	`0.4.4`	GitHub
experimenthub	Access Bioconductors experimenthub resources	`0.0.1`	GitHub
hdf5array	HDF5-backed objects for array and matrix like data	`0.5.0`	GitHub
mopsy	Matrix operations for numpy and scipy	`0.3.0`	GitHub
orgdb	Access OrgDB annotations	`0.0.1`	GitHub
pyBiocFileCache	File based cache for resources and metadata	`0.7.0`	GitHub
rds2py	Parse and construct Python representations for datasets stored in RDS files	`0.8.0`	GitHub
scranpy	Analyze multi-modal single-cell data!	`0.3.0`	GitHub
scrnaseq	Collection of Public Single-Cell RNA-Seq Datasets	`0.3.1`	GitHub

Package	Description	Latest Version	Links
singler	Python bindings to the singleR algorithm to annotate cell types from known references.	`0.5.0`	GitHub
tiledbarray	TileDb backed objects for array and matrix like data	`0.2.0`	GitHub
txdb	Python interface to access and manipulate genome annotations in TxDB SQLite format.	`0.0.4`	GitHub

---

2.5 Developer Guide

If you are interested in developing new Python packages, check out the developer guide on the phisolophy and tools we employ to ensure code quality and consistency within and across all the packages. A more detailed Python package management process is documented in the biocsetup package.