Bioconductor serialization best practices

Lori Shepherd, Martin Morgan
19 September 2019

Scenarios

Personal

-   Check-pointing (e.g., for fast computation) or reproducibility

Package

-   Demonstration data, e.g., for vignettes & examples
-   Package-specific resource, e.g., reference data
-   Project-wide benefit, e.g., EnsDb

Where

< 1 Mb

- Package-specific resource? in the package

1 - 100 Mb

- AnnotationHub or ExperimentHub

> 100 Mb

- Hmm, time to reconsider

How?

R formats

-   'Rda' files containing `data.frame`, `GRanges`, `SummarizedExperiment`, ...

Pros

-   Fast and easy to load

Cons

-   Only useful in *R*
-   Complex (e.g., S4) objects: updated class definitions require methods to update the objects

Community-standard formats

-   csv, bed, hdf5, …

Pros

-   Constant format, so consistent import
-   Useful outside *R*

Cons

-   Cost of importing or constructing complex objects 'on the fly'

Unserializing *R* objects

`readRDS()` (better than `data()` / load())

- Reads the object into R
- (S4) attaches the necessary package(s)
- No automatic validation or updating

`updateObject()`

- Convention
- `BiocGenerics::updateObject` generic
- Object-specific methods defined by the developer, e.g.,
  `selectMethod("updateObject", "GRanges")`

Best practices

Where?

- <1 Mb, useful in a single package or package hierarchy: package
- >1 Mb, or useful across packages: AnnotationHub or ExperimentHub

How?

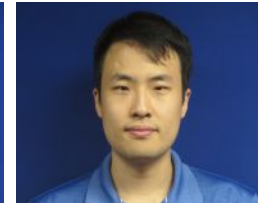- Community standard formats, unless ingestion into R is (time or space) expensive

Conclusions and acknowledgements

Bioconductor core team & close collaborators

- Funded by US National Institutes of
  Health, European Union,
  Chan-Zuckerberg Initiative ...

World-wide community of users & developers

Technical and scientific advisory boards

Acknowledgements