Scalability in R

Kylie A. Bemis

Northeastern University
Khoury College of Computer Sciences

Northeastern University

Goals for this session

- Parallelization in R
- "Big" data backends

PARALLELIZATION

"Embarrassingly" parallel problems

- Independent tasks requiring no communication
- Data can be split into independent subsets
- E.g., could be performed with `lapply()`

BiocParallel

- Parallelization package on Bioconductor
- Provides `bplapply()` function
  - Analogous to the base `lapply()` function
  - Also provides `bpmapply()` and `bpvec()`
- Can `register()` different backends

Serial backend

- `SerialParam()` backend for BiocParallel
- Fallback for non-parallel execution
- Necessary for debugging code

SNOW backend

- `SnowParam()` backend for BiocParallel
- "Simple network of workstations"
- Cross-platform cluster using socket connections
- Starts new parallel R sessions
  - Data must be transferred to worker sessions

Multicore backend

- `MulticoreParam()` backend for BiocParallel
- Single-machine POSIX-only cluster using forking
- Clones the original R session
  - Worker sessions share same data as original session

Other backends

- BiocParallel supports additional backends
- `DoparParam()` backend
  - Supports backends registered through `foreach` package
- `BatchtoolsParam()` backend
  - Supports `batchtools` package for HPC clusters

"BIG" DATA

"Big" data in R

- R expects data to be loaded in memory
- Large datasets require different approach
- Need file-based data structures

Bioconductor packages for "big" data

- DelayedArray
  - Delays operations to avoid unnecessary computation
- HDF5Array
  - Backend for DelayedArray using HDF5 format
- matter
  - File-based data structures using custom binary formats

Using file-based data structures

- Avoid substantiating whole matrix
- Operate on small chunks of data
- Utilize parallelism where possible

Q&A