Open-source software for bioinformatics: parallel computing and large datasets with R

Dr. Benilton S Carvalho
Computational Biology and Statistics
Department of Oncology
The Problem

R is a collaborative project with many contributors. Type 'contributors()' for more information and 'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or 'help.start()' for an HTML browser interface to help. Type 'q()' to quit R.

Bioconductor version 2.11 (BiocInstaller 1.8.3), ?biocLite for help
> x = rnorm(1e9)
Error in rnorm(1e+09) : cannot allocate vector of length 1000000000
> a = matrix(NA, 1500000, 60)
> a = matrix(NA, 2500000, 60)
> a = matrix(NA, 3500000, 60)
Error: cannot allocate vector of size 801.1 Mb
The Problem

- Array-based, but also present for sequencing;

- 10,000+ samples;

- 2,000,000 markers;

- Processing:
  - Normalization;
  - Genotype calls;
  - Copy number calls;
Available Tools

• Multiple machines;

• Lots of cores;

• RAM is rarely enough!

• Additionally:
  
  • ‘The cloud’;

  • GPUs
The product to deliver

• Easy to install and to use; Personal choices listed below:
  • Avoid approaches that require lots of efforts by the user;
  • External libraries (i.e., at OS-level) or anything that cannot be addressed completely with biocLite('myPackage');
  • User does not need to jump through hoops to achieve a certain task;
  • Ideally uses established tools to address already ‘solved’ problems;
Using disk to represent data in R

<table>
<thead>
<tr>
<th></th>
<th>Install</th>
<th>Multiple objects</th>
<th>Use file with 3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>ncdf</td>
<td><img src="yellow" alt="Install" /></td>
<td><img src="green" alt="Multiple objects" /></td>
<td><img src="green" alt="3rd" /></td>
</tr>
<tr>
<td>rhdf5</td>
<td><img src="green" alt="Install" /></td>
<td><img src="green" alt="Multiple objects" /></td>
<td><img src="green" alt="3rd" /></td>
</tr>
<tr>
<td>ff</td>
<td><img src="green" alt="Install" /></td>
<td><img src="red" alt="Multiple objects" /></td>
<td><img src="red" alt="3rd" /></td>
</tr>
<tr>
<td>bigmemory</td>
<td><img src="green" alt="Install" /></td>
<td><img src="red" alt="Multiple objects" /></td>
<td><img src="red" alt="3rd" /></td>
</tr>
</tbody>
</table>
Problems I’ve seen

• bigmemory:
  • uses boost library, which uses all RAM before to switching to disk. Machine becomes unavailable.

• ff:
  • cannot go beyond $2^{31}-1$;

• rhdf5:
  • some types are not yet implemented; can make it not suitable for some tasks;
What have I done?

• rhdf5utils (still for internal use), which implements:
  • data container;
  • helper to add arrays to the container;
  • $[, [<-], \text{dim}$
How does it help me?

- Create containers;
- Add placeholders;
- Process one sample in RAM;
- Save results in container;
- Load results back to R, if needed;
What if I decide to parallelise?

• Choose backend;

• Choose front-end:
  • parallel package
  • MPI;
  • Cluster queues?
What did I choose _________ to use?

• foreach:
  • nothing;
  • doParallel;
  • doMPI;
  • doSNOW;
  • doMC;

• It’s always the same foreach.
Basic usage of foreach

```r
foreach (i=1:10) %dopar% {
    Sys.seep(1)
}
```
Add-on to foreach

• Many options...

• Custom iterator:
  • splits matrix into chunks;
  • sets of (NR/NC) cameras;

• custom iterators:

  foreach (i=myiter) %dopar%{
    colsMeans(i)
  };
Problems and questions

- Send indices to the nodes;
- Really bad if multiple workers try to write on same wine;
- Should users have the choice?
- Should us identify a standardize model for large datasets and use it;
- Should the user have the power of choice?;