Advanced *R* Programming: Course Introduction

Martin Morgan

Fred Hutchinson Cancer Research Center

17-18 February, 2011

Advanced *R* Programming

1. Efficient *R*
2. Interfacing with common resources: SQL, netCDF
3. Using and writing S4 classes and methods
4. 'Foreign language' interface: `.C`, `.Fortran`, `.Call`
5. Integrating *R*, resources, S4, `.C`

Throughout: Packages

Bioconductor: Analysis and Comprehension of High Throughput Genomic Data

Hallmarks of effective computational software

1. Extensive: data, annotation
2. Statistical: volume, technology, experimental design
3. Reproducible: long-term, multi-participant science
4. Leading edge: novel, technology-driven
5. Accessible: affordable, transparent, usable

Course Structure: *StudentGWAS*

Develop a package to manage a genome-wide association study.

- **Thousands of samples** with measured covariates including case / control disease status
- Millions of SNPs assayed with commercial microarrays
- Goal: identify SNPs associated with disease status; integrate with existing body of knowledge

Id	Status	...	Array
1	Case	...	A
2	Case	...	B
...			
M	Control	...	Z

Array	Date	Facility	...
A	12/1/09	FHCRC	...
...			
Z	2/16/11	JHU	...

Course Structure: *StudentGWAS*

Develop a package to manage a genome-wide association study.

- ▶ Thousands of samples with measured covariates including case / control disease status

- ▶ **Millions of SNPs** assayed with commercial microarrays

- ▶ Goal: identify SNPs associated with disease status; integrate with existing body of knowledge

Id	snp1	snp2	...	snpN
1	AA	CC	...	AA
2	AA	CC	...	AT
...				
M	AC	CC	...	AA

Course Structure: *StudentGWAS*

Develop a package to manage a genome-wide association study.

- ▶ Thousands of samples with measured covariates including case / control disease status
- ▶ Millions of SNPs assayed with commercial microarrays
- ▶ **Goal**: identify SNPs associated with disease status; integrate with existing body of knowledge
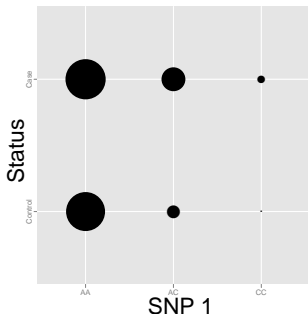
Logistics

Each portion of course:

- Presentation to motivate and orient
- Collaboration to implement specific capabilities
- Review common challenges / concepts
- Update *StudentGWAS* to reflect development