Using JBrowse with large amounts of data

Mitchell Skinner
Ian Holmes Lab
UC Berkeley
“Large amounts of data”

= next-gen sequencing
JBrowse

- Moves work from web server to web browser
- Web browsers are limited
  - Have to be careful not to overload them
- Have to break up the data into digestible chunks
JBrowse

• Assumes that reads are much more common than writes
• Moves work from read-time to write-time
• Have to break up the data ahead of time
Results

- On one test data set:
  - 4.4 million features
  - 8 minutes to process
    - From 242 megabyte BAM file
    - Not paired-end
  - Used 400 megabytes of RAM
  - 330 megabytes on disk (without sequence)
  - Compresses down to 80 megabytes
How to break up the data?
JBrowse uses NCLLists
Lazy NCLLists?
“fake” features
Lazy Loading in Jbrowse
Other approaches to lazy loading

- Heng Li (SAMTools)
  - Binning, linear index
- Jim Kent (BigBed/BigWig)
  - R-Trees
- JBrowse javascript client can't use them directly
  - But Jbrowse could access them through a proxy
Thanks

- Ian Holmes
- Andrew Uzilov
- Chris Mungall
- Lincoln Stein
- GMOD
- NHGRI

More:
http://jbrowse.org