Large genomic data set manipulation

Why are large datasets a problem?

- More difficult to manage
- Potential parsing issues if data not derived from automated data acquisition sources
- Computational constraints *memory / disk*
- Potential visualization, statistical summary interpretation issues
- Scalable?

Next generation sequencing context

- Many thousands of records containing ***short*** sequencing reads
- From a single experiment/sample/lane
- Amount of NGS sequence information is growing exponentially over time

Information growth rate

- Exponential increase
- Doubling time since 2004 is four to six months
- Contrast with Moores law
  - The number of transistors that can be placed inexpensively on an integrated circuit has doubled approximately every two years

"The cost of genome sequencing is now decreasing several times faster than the cost of storage, promising that at some time in the not too distant future it will cost less to sequence a base of DNA than to store it on a hard disk." *Lincoln Stein*

Need automated analysis methods to handle the growth rate

General approach

Ideally use a Collaborative analysis framework

Independent reproducibility/validation of results

Transparency of workflow

Work in a way to promote input form one to many researchers

Prototyping

- In order to work with large datasets a good approach is to break down the amount of information into a manageable amount for testing
- Sequentially sample ~1% of the data using head/tail (available in unix and in R)
- Randomly sample records for prototyping
- Batch processing can then run in

  seconds/minutes/hours

Processes taking > 1 day is too long – loose your train of thought

Sampling example using ShortRead

Code snippet in R;

```
library(ShortRead)

## Documentation
help(FastqSampler)

sp  <- SolexaPath(system.file('extdata', package='ShortRead'))
fl  <- file.path(analysisPath(sp), "s_1_sequence.txt")

f   <- FastqSampler(fl, 50)
rfq <- yield(f)    # sample of size n=50

sread(rfq)
 A DNAStringSet instance of length 50
     width seq
 [1]    36 GATTTTATTGGTATCAGGGTTAATCGTGCCAAGAAA
 [2]    36 GATTTCTTACCTATTAGTGGTTGAACAGCATCGGAC
 [3]    36 GTATGCCGCATGACCTTTCCCATCTTGGCTTTCTTG
 [4]    36 GGTAAAAATTTTAATTTTTGCCGCTGAGGGGTTGAC
 [5]    36 GGTTATTAAAGAGATTATTTGTCTCCAGCCACTTAA
 [6]    36 GTTGAAATGGTAATAAGACGACCAATCTGACCAGCC
 [7]    36 GTACGCTGGACTTTGTAGGATACCCTCGCTTTCCTT
 [8]    36 GACATTATGGGTCTGCAAGCTGCTTATGCTAATTTT
 [9]    36 GTTCTGGCGCTCGCCCTGGTCGTCCGCAGCCGTTGG
 ...   ... ...
```

Eliminate any potential spatial effects

Sanity checking

- Assumptions in workflow can propagate errors
- Think of an independent validation every few steps of programming to sanity check your work

e.g. Sanity check if file with .sam extension exists

```
$  ls -l | grep \\.sam$
```