Project Ideas:

#1: Visualizing and Exploring SeqBias
- Develop a tool to automatically measure, quantify, display summaries of bias in specific RNAseq data sets and apply this to a variety of them
- How does bias vary from one data set to another? Is more modern data less biased? How does it impact down-stream analyses?
- Steps:
  - Learn SOA RNAseq Quality Control
  - Add SeqBias, starting with figures like those in Daniel’s paper
  - Other metrics?
  - Apply to variety of data?
  - HCI Issues in presenting data?
  - Can we implicate cause???
- Challenges:
  - File formats
  - Really understand RNAseq
  - HCI challenges
  - Some statistics involved

#2: Bias Distorts Allele Specific Expression Analysis
- Alleles differ in a small number of positions; bias is sensitive to sequence; so a change in bias at a few changed positions might falsely appear to be ASE or falsely mask ASE
- Explore the effect of SeqBias on ASE prediction. If deemed significant, develop a tool to automatically “correct” this.
- This can be really important
- Does bias compromise our ability to detect ASE? What can we do?

#3: Impact of bias in other RNAseq use cases
- RNA structure prediction (SHAPE)
  - Most RNA is single threaded (no double helix), but it can fold back on itself. How would you predict these structures?
  - If you have an enzyme that chews away single stranded RNA but leaved double stranded in tact – then you can detect double-stranded regions. Bias may be relevant here (missing pieces or seeing over representation)
- Ribosome foot-print
o Ribosome decodes RNA to create protein (3 bases to one amino acid)
o Moves at uniform rate through the RNA, mostly
o But going faster or slower could mean something
o Foot-printing involves freezing RNA with ribosome attached, digest away RNA (except bit protected by ribosome), now sequence RNA to get a snapshot of where the ribosomes were
o Again, bias has implications here (overrepresentation could mean a pause in ribosome, or could just be bias)

• Challenges:
o Understand protocol
o Understand SOA
o Bit of theoretical modeling
o Bias correction models need a background mode
  ▪ Need enough data that is ‘neutral’
  ▪ We need to know what’s artificially amplified and what isn’t
o So FIRST – figure out how to get a background model
  ▪ Guess – both protocols produce enough background RNA that if we can separate it out from signal, that’s enough

#4: Improved crossover detection

• Background
  o A position in your genome where Mom and Dad agree – homozygous (99.9%), where they disagree is heterozygous (e.g. A instead of T or something)
  o To find these – map your reads to genome, find where there are differences
  o (usually there are 2 common letters, very occasionally 3)
  o 30x coverage on any place in the genome is a good lower bound – you really want 100x or so

• Phasing problem
  o IF we have two heterozygous positions, which are on the same chromosome? (A/G ...... G/T) or (A/G......T/G)
  o We have 4 potential proteins, which 2 are really being produced?
You might have some reads that cover both (if they’re within 100 base pairs)

Depending on sequencing, you may have some sense of position (you control the size of the fragments)

**Crossover problem**

- What if we have two siblings with slightly different genomes – 
  \((A/G….G/T)\) v \((A/G.....T/G)\)
- It could just be different (genetic crossover/recombination, meiosis) or it could be a phasing error

**However, if you have multiple phasing calls in the same vicinity, can you increase/lower confidence phasing calls are correct?**

**There is an unpublished solution, but it’s not perfect**

- It finds more crossovers than it should

**Note: any even number of crossovers will show up as 0, any odd number will show up as 1**

**You could maybe set up a system of linear equations of non-overlapping intervals**

**What’s involved?**

- Understand tools for identifying phasing, heterozygous positions
- They usually give some sort of error rate estimate – understand it
- Decode output
- Construct system of equations and solve in MLE framework
- At a guess – this problem is NP hard
- We think this breaks the genome into small independent sub-problems
- Can you try all \(2^n\) assignments, then?
- Or not, maybe some good heuristics?