WaveQTL sims notes summary

Brendan Law

August 8, 2019

Key steps listed are:

1. Overview
2. Get 578 dsQTL data (raw data and genotype); interrogate, clean data
3. Estimate effect size using BAYES.THR in wavethresh package; wavelet denoising by wavelet thresholding
4. Estimate effect size using multiseq and WaveQTL; wavelet denoising and modelling using our packages
5. Check if we have all information from 578 dsQTL; validate signals found?
6. Perform simulation with effect size from BAYES.THR
7. Perform simulation with effect size from multiseq
8. Perform simulation with effect size from WaveQTL
9. check if there is an upper bound for performance of multiseq (by increasing library read depths)
10. Additional simulations (reduced library read depth)
11. Prepare output to make figures for a paper
12. Debugging

Steps 6-8 are fairly similar. Using the effect size found from the respective algorithms, we:

- Prepare simulated (both null and alt data) data with the given effect size
- Run multiseq, WaveQTL, DESeq2, edgeR
- Collect test stats (logLR) for WaveQTL, multiseq
- Plot test stats, ROC, AUC curves

Step 9 regards how effective multiseq is, at increased library read depths. The idea here is that multiseq is effective for small library read depths (it should be most effective against WaveQTL at small counts and therefore, small read depths). Does it have an 'upper bound' to its relative performance when we compare it to a situation where its theoretical benefit over WaveQTL is diminished? So then, this is similar to Step 7, but with increased read depth.

Step 10 seems quite relevant – reduced library read depth should be where we see WaveQTL, at least, do quite badly, and multiseq do quite well, relative to it. This part actually combines 6 - 8 into one big section, with a low read depth. Hence, good examples of how to do the ENTIRE workflow in one should be steps 9 and 10. (Although maybe i'm getting the small read depth confused with small sample size – where is multiseq meant to outperform waveQTL/waveQTL meant to be bad?)

In our case, we need to pick out some cases where we think WaveQTL_HMT will work better than WaveQTL – specifically, small, localised effects, over a broad range, rather than strong signals over a narrow (or braod) range.

1 Overview

Overview is summarised as such:

- Simulate null and alternative data sets from each of 544 dsQTL out of 578
- (Effects?) identified by both:
  - Wavelet based
  - 100bp window approach at FDR = 0.01 (what is this? False discovery rate, but what is it?)
- Simulation procedure similar to Shim and Stephens:
  - Here was their simulation procedure
- Consider three alternative models for simulating effect sizes:
  - WaveQTL
  - multiseq
  - BAYES.THR (BayesThresh method of Abramovich, Sapatinas, Silverman (1998) – which actually looks like some sort of parameterised, Bayesian thresholding rule, imposed on a prior like ours, and which, due to an l_1-norm, 'thresholds' out (either include or exclude) WCs by working in a similar way to how a LASSO 'thresholds' out variables in its optimisation problem, given particular parameterisations of λ)
- Compare two multiscale approaches (multiseq, WaveQTL) and one windowed approach (DESeq2 – 100, 300 and 1024 bin sizes (Are these analogous to the 100bps windows??)) on simulated data (544 null vs 544 alternative) to see how well it *retrieved* the effects.
  - Different sample sizes (Is this analogous to the 70 individuals in our play dataset?)
  - Different read depths (Would this affect the WC filtering? WC filtering on low WC counts, is this the same as read depth?)
- As expected, multiseq > WaveQTL, DESeq2 at all sample sizes
- As expected, at small sample sizes (6, 4) and with increased read depths, multiseq > DEseq2 >> WaveQTL
- What about at decreased read depths? Section 10...
- (only used major homozygotes and heterozygotes. WHAT DOES THIS MEAN???)

Here's the Shim and Stephens simulation methodology:

- Degner:
  - Do I need to understand the details? Pretty much just taking non-overlapping 100bp windows, summing counts and sequencing counts, then using this to do the association analysis and effect size estimation (as expected).
  - Differences are in which SNPs (neighbourhood/locality of the SNPs) in which they tested against.
- WaveQTL:
  - WaveQTL on 1024 bp site
  - Can also compare to analogous windowing method – $9 \times 100bp + 1 \times 124bp$
- WaveQTL identifies all the ones from the window, plus more. Fig2 is one picked up by both (strong, similar in length to the 100bp window, hence easy to identify in window). Fig3a is picked up by WaveQTL, but not windows (strong, but very narrow – lot shorter than window, not strong enough in 100bp to be picked up). Fig3b is the opposite – effect is weaker, over a wider region. Fig4 good example of effect split between two windows (neither window strong enough), and another which shows effects in two directions in one window – cancelling out across the window (again, not strong enough)
- Tried using shifting windows instead

There's a whole heap about 'design' on page 5 and 6, specifying:

- Calculating effect sizes from each of the three methods
- Something about simulating with p parameter, determining a binomial distribution
- Then apply all three multiscale methods (DEseq2 - 100, 300, 1024 bin sizes) and evaluate performance with ROC

2 Section 2 – data checking

- Check for missing data
- Recreate some missing 'geno' files

3 Section 3 – estimate effect size using BAYES.THR

- I don't have access to a lot of these R scripts I think...
- The same can be said for section 4 (using multiseq and WaveQTL) – all driven by R scripts I can't see, and bash script commands.
- Sample sizes 70, 30, 10, 6 and 4
- What are the read depth ratios???
- Runs:
  - Over-disp: 1/70/70/10
  - Null:

ReadDepthRatio	70	30	10	6	4
4				Y	Y
2				Y	Y
1	Y	Y	Y	Y	Y
0.5	Y	Y	Y		

  - Alt:

ReadDepthRatio	70	30	10	6	4
4				Y	Y
2				Y	Y
1	Y	Y	Y	Y	Y
0.5	Y	Y	Y		

- No alt and null distinction for DESeq2 and edgeR (as opposed to multiseq and WaveQTL)
- Investigate Section 6.10 the most – has the most transparent code (plotting results):
  - pROC library for AUC curves
  - Example of ROC curve code:

```
# multiseq
stat.null = ms.null
stat.alt = ms.alt
dec.order = TRUE
fpr.list = tpr.list = vector("list", length(case.name))
for(cc in 1:length(case.name)){
    ##cc = 1
    total.test = length(stat.null[[cc]])
    stat = as.numeric(c(stat.null[[cc]], stat.alt[[cc]]))
    wh = which(stat < 0)
    if(length(wh) > 0){
        stat[wh] = 0
    }
    disc = c(rep(0,total.test), rep(1,total.test))
    rnk = order(stat, decreasing = dec.order)
    stat.order = stat[rnk]
    disc.order = disc[rnk]
    sig = NULL
    tpr = NULL
    fpr = NULL
    uni.stat.order = unique(stat.order)
    for(i in 1:length(uni.stat.order)){
        if(dec.order){
            wh = which(stat.order >= uni.stat.order[i])
        }else{
            wh = which(stat.order <= uni.stat.order[i])
        }
        sig[i] = length(wh)
        tpr[i] = sum(disc.order[wh])/total.test
        fpr[i] = (length(wh) - sum(disc.order[wh]))/total.test
    }
    fpr.list[[cc]] = fpr
    tpr.list[[cc]] = tpr
}
fpr.ms = fpr.list
tpr.ms = tpr.list
```

  - Images on page 91 – which ones of these would we want?

4 Section 9 – increase read depth

So, according to wikipedia, read depth is ‘number of unique reads that include a given nucleotide in the reconstructed sequence’ (in order to reduce chance of one sequencing reflecting an error, rather than many reflecting an actual SNP).

My guess is the read depth here is to show how good our algorithms go on sequencing of various details. So for read depth 4x, for example, how do we simulate the results from the four reads? Is that what the binomial parameter, p, is about? We do n trials from a binomial with probability p, with differing p depending on whether effect or not at that location?

- Sample size 6 and read depth 10, 50, 100, 500
- Run for WaveQTL null and alt, multiseq null and alt, DESeq2 null and alt
- Combine simulated data – what does this mean?

5 Section 10 – additional sims

- Sample size: 70, 30, 10
- Effect sizes: multiseq, WaveQTL, BAYES.THR
- lib depths: 0.1, 0.05, 0.01 (really small)
- Combine simulated data from the three effect size generation methods
- Run each of multiseq, WaveQTL and DESeq2 on the combined data (Why didn’t we do this before?) And why do we only seemingly combine data when running the WaveQTL and DESeq2 algos, but not when running multiseq?