Research Diary Summary

|  |  |  |
| --- | --- | --- |
| Week beginning | Activity | Comments |
| July 24th | Online edX course: Data Analysis and Life Sciences – Week 1 (Statistics and R) | Course is very basic at the moment but useful to review, particularly writing in R. Focus on Central Limit Theorem |
| July 31st | Online edX course: Data Analysis and Life Sciences – Week 2 (Statistics and R)  Download data  Work through R example in data download  Meeting with Loic | Easy to download data and R example also easy to follow (also pleased to see it makes use of the dplyr library which I have just learned about in the edX course and the filter and select tools)  At first confused by the many different files available but Loic suggested I use the cojo files.  Re-read Luke's paper to refamiliarise myself with his data analysis. |
| August 7th | Online edX course: Data Analysis and Life Sciences – Week 3-4 (Statistics and R) | Read Yang et al to familiarise myself with cojo analysis.  Finished the introductory course on statistics and R – this has been really useful. |
| August 14th | Online edX course: Data Analysis and Life Sciences – Week 1 (Introduction to Linear Models and Matrix Algebra)  Write program in R: create sequence set for each eQTL locus | This edX course is a big leap up in difficulty and I'm not sure it's worth the effort to master the matrix algebra. It's beginning to seem not quite relevant and there are other things I should be doing.  The program in R turned out to be extremely simple, but it took me 2 days to get there, and on the way I divided all the data according to chromosomes and found sequences per chromosome. I left all these files sitting in a folder just in case I need them in the future. |
| August 21st | Online edX course: Data Analysis and Life Sciences – Week 2(Introduction to Linear Models and Matrix Algebra)  Begin literature review to assess best k-mer length for motif searching  Meeting with Luke and Loic | Decided to stop the edX course for the time being – the matrix algebra is becoming more complex and it's taking quite a while to complete the required exercises. I might go back to it later.  I made an initial list of readings (using the simple search term “kmer length” within UQ library) and began – I didn't get them finished. I need to start summarising them as well.  Meeting: planned future activities – finish this lit search, and begin another focussing on online programs to find motifs. Make sure the program can focus on motifs in blood (not on some other cell type that may limit its generalizability).  Then look for enrichment.  Need to keep a research diary. |
| August 28th | Conducted a more extensive literature search on motif finding algorithms.  Began to read and keep notes. | The most successful search terms were:  1. Within the UQ library site (a) “gwas” AND “kmer” (articles in last 5 years)  (b) “motif finding algorithm” (articles in last 2 years) – if added “eqtl” got zero results, even if extended to 5 years.  2. Within Google - “motif finding algorithm” and “eqtl” (since 2016) (482 results)  Initial observations from lit review:  Algorithms are either in vivo (based on eg ChIP-seq) and take into account biological conditions such as cell type; or in vitro and finding intrinsic binding preference based on sequences. So I'm looking for an in vitro algorithm.  Some algorithms seem to produce just one motif that has the highest EM score over all sequences – this won't work for me since I want all the motifs. |
| September 4th | Continued to read each article found through literature search and keep notes – finished  Created a “to do” list: see notes | Literature can be divided into motif search algorithms, reviews of motif search algorithms, more general reviews of characteristics of transcription factors.  Most articles focused on transcription factor motifs, although the de novo motif search algorithms are not restricted to TFBS.  Not sure if I've reached saturation point. The most popular well established motif search algorithms seem to be MEME, Weeder, Homer. At least these were the ones often used as benchmarks for comparison in articles presenting / reviewing new algorithms.  To do list:  1. Organise notes into an interim lit review  2. Follow up references in articles to add to lit review  3. Refresh use of Python programming language and Gibbs Sampling method in particular, since this forms basis of a number of motif finding algorithms.  4. Also revise Bayes theorem and Markov modelling |
| September 11 | Worked on lit review  Extended list of references  Focus on specific types of algorithms  Downloaded Anaconda3 software which includes the Jupyter notebook software which can run both Python and R. Started to familiarise myself with this. | Realised I needed more information about the different approaches to writing algorithms.  Did a google search: “types of motif discovery tools”. Yielded ~ 2 million results.  Refined this: used Google scholar (through UQ website) and tried “types of motif discovery tools review”, limited to articles since 2016 ~ 28,100 results  In searching for articles on graphical algorithms – Google search “motif finding graphical algorithms” which turned up some useful articles plus the website DNA MOTIF FINDING SOFTWARE TOOLS | GENOME ANNOTATION <https://omictools.com/motif-discovery2-category> and its associated article by Dassi and Quattrone which helps to pull together what I've read so far.  Mastered a couple of algorithms – I already understood the Gibbs sampling and expectation maximisation algorithms, now I've managed to understand the projection algorithm and also one type of graphical algorithm. Really feel like I'm making progress. |
| September 18 | Worked on lit review  Extended list of references through focused searches (eg “expectation maximisation motif search” in order to learn more about algorithms | Came to grips with more algorithms – found that I didn't really follow how the EM algorithm is applied to motif finding – still not quite sure how the initialisation avoids enumeration. Learned about suffix trees, clustering, and the genetic algorithm.  Feel like there's lots more to do with this lit review, but pulled it all together to submit the first draft. |
| September 25 | Began “Pipeline” document to keep track of processes  Tried DRIMust (designed for ChIP-seq data) – found 8 motifs  Tried the MEME algorithm – failed – see notes  Started using the OMICtools website [https://omictools.com](https://omictools.com/)  Entered search term: motif discovery – returned 233 results  <https://omictools.com/search?q=motif+discovery>  Ran DMINDA2 This found 30 motifs which seem to be mostly GC motifs. DMINDA is meant to be run on promoter sequences  Decided to try HOMER – this needs a .bed file, so wrote a program to convert sequence data to a .bed file.  Ran RSAT, which found 18 motifs. | DRIMust needed sequences in fasta format. I couldn't find an easy way to do this in R, but I found a website (bugaco.com) which took data organised into 2 tab delimited columns (SNP site plus sequence) and turned these into a fasta document (ID plus sequence).  Entered data in fasta format into DRIMust website  <http://drimust.technion.ac.il/>  First attempt used default parameters (statistical significance threshold = 10-6) took 3 minutes to run but found no significant motifs.  Second attempt reduced the statistical significance threshold to 10-2. Also raised maximum number of motifs to all motifs. Named the job “drimustSearch” and asked for link to results by email. Job took 3 minutes to run and found 8 motifs.  The MEME algorithm failed to run . Problems:  ✘ Sequence identifier duplicated - identifiers must be unique   * Found 1065 duplicated sequence identifiers on lines 1691, 6011, 6051, 6561, 7031, 7056, 8031, 8406, 8921, 9286, ...   ✘Too many sequences Found 14995 sequences but this only accepts up to 1000  ✘Combined sequence length exceeds maximum Found sequences with lengths totaling 3013995 but this only accepts a total length up to 60000  Creating the bed file – the added advantage of this is that now I can view the target sequences in an IGV viewer. Homer ran easily and found 40 motifs  An advantage of running RSAT is that it gives the option of identifying motifs using standard data bases, which include Homer. This helps with comparing with Homer search. |
| October 2 | Wrote up project plan and submitted it (late).  As part of my plan, I decided I needed to take a systematic approach to identifying appropriate motif search tools. I created a spreadsheet and began to work through the 233 tools listed in the OMICs result, listing all tools that could be used in this project. | I omitted tools that were only suitable for data such as ChIP-seq or RNA-seq.  My categories included: the reference publication; the algorithm used; the journal ranking of the reference publication; the number of citations; and any reviews.  One useful sounding tool is some software called BliC, which is designed to process multiple results from multiple algorithms in order to find common motifs. Not motif finding in itself, but it might be very useful to process all my results. It also indicates that there might be other programs out there that do similar things. |
| October 9 | Worked on motif finding software. Finished list of possible tools and used this as a guide to trial different tools. Many tools wouldn't download successfully, others were missing (perhaps no longer maintained); others wouldn't run successfully (too much data).  Luke suggested learning LaTex – downloaded LaTex editor and worked through some tutorials. | Failed with SMILE – couldn't read my fasta file  Mobydick – server wouldn't connect  AlignAce – Error: file size less than 999 bites (??)  Improbizer – too many sequences – maximum is 4096  Glam2 – Combined sequence length exceeds maximum: found sequences totaling 2799930 but this only accepts total length up to 60000  glam2 n – produced one different motif per sequence  STEME – currently working; taking many hours |
| October 16 | While waiting for STEME to complete - Took a look at the sequences via the bed file on the igv. Noticed that there seems to be a mix of locations – both within genes and outside of genes. Considered the possibility of dividing sequences according to their location, reasoning that intragenic snps are likely to regulate gene expression in different ways to intergenic snps. Investigated how to annotate each sequence according to its genomic location.  Followed the Mitch Beritsky instructions to get bedfiles of intron, exon, promoter (+1000), UTR'3, UTR'5 regions using the UCSC Table Browser.  Combined intron, exon and UTR bedfiles to get “genic” bedfile.  Used bedtools “intersect” tool with the list of sequences to get sequences that overlapped with introns (7834 sequences), exons (2131 sequences), promoters (1231 sequences) and intergenic (use the -v option) (5132 sequences) regions.  Ran motif finding tools on the promoter sequences – STEME, DREME, Thicweed, Drimust and Homer  Downstream analysis: Searched for a tool that would analyse the motifs found by these tools – found Stamp, which will cluster similar motifs within a tree formation and also name the motifs according to a designated library.  The motifs need to be formatted and entered into an input file: this succeeded with STEME, Thicweed and Homer, but not with DREME (surprising – needs investigation) or Drimust. For this trial run I went with STEME, Thicweed and Homer.  The algorithm ran successfully and produced a tree of clustered motifs, as well as a list with possible motif identities. | Found these instructions on <https://www.biostars.org/p/94823/>  from a user called Mitch Beritsky  If you want to get annotations for every exon/intron/UTR in a reference genome, you can use the UCSC Table Browser.  Here's how to get it done:   1. Pick your reference genome under clade/genome/assembly 2. Make sure the group is "Genes and Gene Predictions" 3. Choose your preferred track (I like to rely on RefSeq and CCDS) 4. Choose the table that gives gene information (e.g. for RefSeq, the table you want is refGene) 5. Select your region or the entire genome to get coordinates for 6. Select BED format as your output format 7. Name your output file 8. Click "get output"   On the next page, you will get the option to get coordinates only for all exons, coding exons, introns, 5' UTRs, or 3' UTRs (plus flanking sequence if you want). You can download these coordinates however you'd like (I prefer having one file for each genomic feature type), then overlap your mapped sequences to the genomic features using [bedtools](https://github.com/arq5x/bedtools2)' intersect.  To find intergenic regions, you can create a merged BED file of all exons, introns and UTR sequences and look for mapped sequences that overlap NONE of those features using [bedtools](https://github.com/arq5x/bedtools2) intersect with the -v option.  Checked the sequence by region files on IGV to make sure the regions tallied with the refseq genes – they do! Many of the intron sequences were also exon sequences as they overlapped both intron and exon regions.  Some of these motif finding algorithms want a background region (DREME, STEME, Homer) – for these trial runs I've been allowing the algorithm to create the background by scrambling the input sequences – but it would be good to enter actual sequences as background (e.g. intergenic sequences as background to both exon and intron sequences)  To open the .tree file I used TreeViewX software |
| October 23 | Ran a few more trials to settle on my final five algorithms:  1.DREME (part of the well-known and much cited MEME suite but unlike MEME can handle larger data bases). Not as fast as Homer but still good.  2.HOMER: Also much cited and very user friendly with plenty of instructional webpages. Also has its own directory of TF's for comparison. Fast.  3.STEME: copies the MEME algorithm but adds an initial suffix tree to manage large data. Still quite slow.  4.Multi\_motif\_sampler (or RPMCMC): uses the Gibbs sampling method.  5.ThiCweed: dated 2017, uses clustering, very fast.  Dreme, Homer and Steme all need background sequences. Created background sequences for each group of sequences (see notes). Used IGV to eyball whether any background sequence overlapped with any eQTL sequence – no overlaps.  Ran each algorithm, with the background sequences if required, for each group of sequences. | Creating background sequences: Used intron files from whole genome to make background pool for both promoter and intergenic sequences. Used intergenic files from whole genome for both intron and exon sequences. To create intergenic files, used the USCS table browser to create a bedfile of 10,000 bases upstream of each TSS.  Ensured that sequences in the background pool did not overlap with any eQTL sequences (exon, intron, intergenic or promoter). Wrote an R program to remove X and Y chromosomes, remove any sequences less than 201 base pairs, and trim sequence length and number of rows to match the relevant group.  Each intergenic background sequence started 10000 bases upstream of TSS and finished 201 basepairs later.  After running algorithms, had five folders (one for each algorithm), each containing results for each of the four groups of sequences. |
| October 30 | Wrote a python program to format each algorithm result so the Stamp program would return files with the motifs marked with the relevant algorithm.  For each of the four groups, ran the Stamp formatting program for each formatted algorithm file to create the input program for the Stamp algorithm.  Ran the Stamp algorithm: the most useful result at the moment seems to be the file that lists each motif and provides a JASPAR reference.  Wrote programs to format this file then created a table of JASPAR references with frequencies | Wrote a number of scripts to automate these processes, incorporating both python and R programs |
| November 6 | Ran all algorithms and programs to create file of possible motifs |  |
| November 13 | Began Report – writing in LaTex  Checked out null eQTLs – created sequences, did initial search for motifs. Difficult to compare with target sequences given the number of motifs. Decided to use as background sequences for target sequences.  Found problem: not all my algorithms use background sequences. The two algorithms not using a discriminative approach (Thicweed and RPMCMC) found very different results from the other three algorithms. Need to replace these algorithms with other discriminative algorithms | Null sequences seem ideal for a control group: divided into promoter, exon, intron and intergenic as target sequences, so similar – but eQTLs have no effect on gene expression. What is the difference between them? |
| November 20 | Checked out new algorithms, settled on three new ones: Bamm!motif, DECOD, and MotifRG.  Ran all algorithms for exon group. Wrote programs to include Homer stats in overall table – ratio of target to background sequences and e-value. | Homer may not be the best algorithm in that the e-values seem inflated, but it has excellent coverage of motifs – it finds them all – and it is the only algorithm to include the ratio of target to background sequences. |
| November 27 | Decided I should include E-values for all algorithms, so found where they were located in the algorithm reports and wrote scripts to extract them and a program to include them in the tables.  Continued with report.  Ran algorithms for intron, intergenic, promoter sequences. | All scripts and programs working well after a few glitches – but keep deciding on new tables to include and then need to write a program to create the new table! |
| December 4 | Report completed |  |
| December 11 | Meeting with Luke and Loic: 2 fairly specific jobs. First: ran Homer and Bamm using the null sequences as the target to see if the transcription factors found in the eQTL sequences are also found in these sequences. They are – with the exception of 4 TF's. The motifs are not the same, but they bind to the same TF's.  Second job: Need to create long sequences (200001 nt) and search for overlaps with long background sequences. Decided to do this in steps, trying out increasingly long sequences: 2001 nt; 10001nt; 40001 nt and finally the 200001 nt. Created all these sequences. |  |
| December 18 | Wrote programs to check for overlaps, then create histogram showing the proportion of eQTL sequences in each chromosome that overlaps with null sequences. | The program to check for overlaps was written in Python because it involved two “for loops”: for each target sequence (1st loop), check for overlap with any null sequence (2nd loop). The two loops meant that it took a long time to run (irrespective of length of sequence). I couldn't think of any way to do it without the two loops. The main problem with the running time was that it took an equally long time for bugs to show up and be resolved.  Creating the histogram was written in R. |
| December 25 | Finally the overlap program was running without bugs, so ran it for each of the different length sequences and created lots of histograms. Wrote 2 reports (in latex) to describe results for both the repeating TF's and all the overlaps. |  |
| Break for Philippines holiday | | |
| January 22 | Meeting with Luke and Loic: developed a plan for the next steps.  1. Use 1 kb sequences each side of eQTL (otherwise overlap is too big) – run Homer and Bamm  2. Set up git hub account and project and add documents.  3. Try out ways to match background null sequences and eQTL sequences – through LD scores, no. of genes in +-1MB region null SNP, GC content and no. of SNPs around null non-eQTL. This might help identify possible confounders.  Completed No. 2 (set up github account and project) and added this diary. | Homer runs without any difficulty for the 2001 bp sequences, but Bamm had a memory problem and couldn't run.  Increased virtual memory (called “swap size” in linux) to 32 GB (did this in increments until Bamm would run). After this, Bamm ran without any problems, though it took a few days. |
| Break for Tasmanian holiday | | |
| February 26 | After the long break I needed to refresh my memory with respect to scripts, programs etc. Also decided that it would be useful to check the reliability of Homer and Bamm. So ran programs 3 times each and then ran the Stamp program twice – on the three versions of each algorithm. The cladogram showed that each version of the same algorithm produced exactly the same motifs |  |