# **28/10/22**

Started work on adapting previous IGR\_presence\_absence reader to convert PIGGY IGR\_presence\_absence.csv to GFF3 format for each genome (I.e for each genome, list all IGRs in GFF3 format)

# 04/11/22

Finished script and processed genomes/igrs

# 061122

Failed to make meme tool work – idea was to have meme identify all motifs(sequences that seem to occur repeatedly in genome in statistically significant manner) and see if any identified IGRs overlapped at one of the motif occurrences, if so it suggests the other occurrences of that motif may bind the same proteins as the IGR that overlapped the motif.

# 071122

Forgot to save, part of diary last entry erased.

Made python code to collate sequences for all occurrences of a given cluster across all genomes, stored as multifasta(1 per cluster)

Found I was using meme incorrectly – im meant to put the IGR sequences in, rather than the genomes; If a common motif is found between two different clusters, then the clusters could bind the same proteins as the conserved motif may be the regulatory binding site. Therefore if one cluster is known to bind a TF according to the collecTF data, then the other clusters that share the motif found in the original TF binding cluster may also bind the same TF.

There is over 169000 clusters – need to find a mode of automation.

# 081122

Further misunderstanding – instead of collating all sequences of a given IGR in a single text file, I should make text files per *genome* and collate the sequence of all IGRs that occur in that genome only, then use this with meme

Given the above, I changed the code slightly to produce multifastas as detailed above, and have made a snakemake file to automate genome processing on my computer vs manually submitting to website.

MEME parameters: meme -o {output.outdir} -dna -mod anr -nmotifs 50 {input}

Snakemake parameters: snakemake -s memeautomator.snakefile –max-threads 8 –cores 4 –latency-wait 20

If my computer explodes, expect no further entries

Application

Description automatically generated with low confidence

The snake file test failed, ill try fix it with Leighton tomorrow. Ive begun manually inputting the files to meme online and running one on my laptop, but meme limits me to 4 per hr online.

The initial test of meme on GCA\_007525(smth like that) for 3 motifs worked, ive moved to 50 motif scans as the internship yielded binding data for 50 TFs. I don’t know how to interpret the results however – bring this up next meeting.

Provided I can figure out how to interpret the results, the next step would to be to look at the motifs *within* genomes and see if a single motif occurs within two different IGR clusters – if one of those clusters was previously identified as a TF binder, the other cluster sharing the motif may also bind that TF, but this isn’t necessarily true.

After that, I don’t know. I could try looking for alternative TF binding site data than collecTF but im not sure where to begin.

The graphing scripts will need updating in order to accept the new sources of IGR binding data.

Furthermore, for the TFs ive already identified binding sites for I could try looking into whether their binding is positive or negative.

# 091122

Met with Leighton, snakemake is now functional. Turns out the MAST output is more useful than meme, but requires meme as an input. Will automatically run files through meme over the next few days, then will do mast after – mast is very fast so wont hinder much.

According to Leighton, when looking at the MAST results I should look at clusters with very similar motif structures, and for motifs with a palindromic structure – palindromes often bind TFs.

Diary likely wont be updated for a day or so until meme is done so cya.

# 191122

Meme finally done. Worked on code and studied in mean time. Need to quickly run meme.html through mast and develop code to read output mast.txt

Mast automator complete – running with command line: snakemake -s mastautomator.snakefile –max-threads 8 –cores 4 –latency-wait 20

Mast files done, moving to work on mast.txt analyser – not sure how to do this

-scoring system? One that demerits based on motif not matching? MAYBE

-look at percent similarity between overall IGR sequence? NO

# 201122

Decided to make barebones motif comparator, will only say IGRs are similar if they contain the exact same motifs in the same order(or exactly inverse order, for if the motifs are on the reverse strand).

This will end up missing any IGRs that are similar but differ in one motif – will have to correct in the future but for now this will do.

After im done this the graphing itself should be easy, I just need to find a way to group IGR nodes without destroying their edges

# 211122

Motif comparator is nearly finished, it reads mast.txt fine and can split up the motif diagram for each IGR into its constituents, the only issue is actually deciding how I want the strands to be compared.

Ill probably go with a point deduction mechanism – if by the end of all deductions the score is still above a threshold then the IGRs are similar enough

Could compare len of constituent sections – i.e each motif diagram is broken down into just the numbers (with square brackets around the numbers that are defined as motifs, other numbers are just the interspacing regions) – closer match equals less deducted

Following that, compare the actual motifs present – if motif composisiotn is similar then less deducted. Do similar for interspersing regions but weight the deductions less than the motif deductions.

Then compare the *order* of motifs – incorrect order means deduction, but how much do I deduct?

IDEA:

In the program there will be two sequences compared at any given time – the constant sequence, and the variable sequence that we’re comparing the constant to. Each iteration of the comparison loop compares the same constant sequence to a new variable sequence. After all variable sequences are cycled through, the constant sequence is removed from the original list and a new constant sequence is selected to be compared to all the variable sequences again.

Initial score = 1000

Threshold = 500

Number of sections deduction(NOS\_deduction) = initialscore \* shorterlen/higherlen (so the greater the disparity, the greater the deduction?

* issue – say IGR1 has motif sequence(letters=motif,number=intersperse) X-4-Y-9-Z-2 whilst IGR2 has A-4-F-6-E-4-X-4-Y-9-Z-2-R-3-S-6-O, the entirety of IGR1 is in IGR2 and is in the correct order so IGR1 may be the entire regulator whilst IGR2 is the regulator + random DNA that was included since PIGGY only differentiates gene from non gene – the IGRs are similar but this system classes them dissimilar.

Motif deduction(MoD) = NOS\_deduction \* NO\_Motifs\_Shared(by variable seq)/NO\_motifs\_total(in constant sequence)

Order deduction(OrD) = MoD \* (1-(NO\_shared\_motifs\_correct\_position/NO\_motifs\_shared))

* Order deduction is more complex as the motifs from constant seq can be in the same order RELATIVE TO EACHOTHER in the variable seq but the variable seq may have additional motifs between the shared motifs. For now I will ignore this and simply say that if theyre all in the same order theres no deduction
* ALTERNATIVE – MoD and OrD COMBINED Len omitted?

ALT2: Max score 1000, sum of 4 factors – Length disparity, Motifs Shared, Motif relative orders, interspersing regions – Score divided 200:325:325:150 respectively

Lengthcontrib = 200 \* (shorterlength/longerlen) #See above for len definition and issues, If needed will be omitted and contribution divided among remaining sections

Sharecontrib = 325 \* (Motifs\_Shared\_Variable/Motifs\_total\_constant) #If all motifs in Constant seq shared by variable seq, contrib is maxed

Ordercontrib = 325 \* (Motifs\_Ordered\_Variable/Motifs\_total\_constant) #If relative motif order in variable same as constant, contrib is maxed – NOTE IF REV COMP MATCHES ORIGINAL FORWARD COMP, CONTRIB WILL STILL BE MAXED

Interspersecontrib = 150 \* 1/(Unshared\_intersperse + Unordered Intersperse)

# 231122

Leighton meeting notes:

1. After graphs, look into interpretations

# 251122

The comparator is done, the total score formula was altered slightly from above

Totalscore = lengthcontrib+ordercontrib+sharedcontrib+interspersedcontrib

Lengthcontrib = 200\*(lessdiagramsections/morediagramsections)

Ordercontrib = 300\*(numbersharedmotifsinorder/totalnumbersharedmotifs) (considers order in either direction, picks direction that yields greatest sharedmotifsinorder value)

Interspersedcontrib = 200\*((1/unsharedinterspersingregionstotal+unorderedsharedinterspersingregions)

Ive run it for all genomes, theres an option to review each comparation manually but theres too many datapoints to review. I’d estimate that 90% of the identified similarities are accurate if not more, though there may be some issues when you compare IGRs via a proxy- i.e IGR1 is similar to IGR2, and IGR2 is similar to IGR3, therefore IGR1 may be similar to IGR3 but not necessarily, my code assumes IGR1 and IGR3 are similar.

Im going to make a script to collate all the gene names from the gbk files and associate them to their locus tag. For the genes that seem common ill look for papers online detailing their function and see if this correlates to the IGR its commonly associated with.

Beyond that im going to look for more transcription factor binding site data as my graphs are relatively incomplete.

# 271122

Contrary to last entry, I ended up making a script that for each genome it scans the gbk file, gets each locus\_tag(and if present, the gene name) and scans the operon prediction and IGR predictions associated with that locustag and operon(respectively) so that for each locus tag you can easily check associated cluster, operon and genename if known.

Im now going to make a script that scans this file to check if any known genes commonly coincide with one of the clusters, and make note of where deviations occur. This will also incorporate the motif comparator merges.

Whatever output I find interesting from the above, ill look into the genes involved and see if theyre of significant importance to cell function or perhaps environmental adaptations from the pangenome.

# 021222

Done lots of different things:

* Made powerpoint presentation for project
* Made script to check cluster to named gene association differences across genomes
* Made updated versions of motif comparator
  + V2 simply corrects a bug found in V1, enabling two IGRs that share a mutual similarity to be merged, despite not necessarily having a similarity occurrence due to how my code works
  + V3 merges clusters based on similarity found across ALL mast files, difficult to word what I mean – basically V1/V2 read only the mast.txt appropriate for that genome, then only merged the clusters in that genome according to the similarities found in that mast.txt. V3 reads the mast.txt file then finds all similarities within that file, but then instead of merging clusters, it moves onto the next mast.txt file until all are read and the similarities WITHIN the files recorded then for each genome, clusters are merged based on similarities found across ALL mast.txt files
    - Note clusters from 2 mast.txt files CANNOT be compared as the motif numbering system isn’t necessarily the same. Instead, the comparison is made if Cluster A from 1 mast.txt is similar to Cluster B in the same file, and Cluster B from 2 mast.txt was found similar to Cluster C in the 2 Mast.txt file
  + The output was a mess and probably useless but kept It incase

Currently looking into the function of some of the downstream genes of clusters that are associated with that gene in >70% of genomes as I want to find out why the remaining 30% associate differently. The percentage chosen was arbitrary.

# 101222

Late update

Got more IGR association analysis tools made, contained as part of the overall cluster-gene assoc analysis script, just has more outputs:

Gene\_to\_IGR\_assoc analysis lists every gene per genome and the associated cluster for that occurrence, alongside relevant statistics for the number of times that gene occurs across all genomes and what percent of those are associated to the same cluster and what percentage association would be expected if each cluster were equally likely

IGR\_TF\_Binding\_analysis lists every TF binding site(according to collecTF) per genome for which the genome also encodes the TF, the same but for IGRs that have a TF the genome DOESN’T encode, then the same for the TFs from collecTF that don’t appear to have a binding site in the genome, then all remaining IGRs for which no TF binding profile is known

Collated\_IGR\_to\_gene\_assocs lists every TF binding IGR across ALL genomes and then the typical composition of the downstream operon: i.e IGR1 gene1 %ofoperons, IGR1 gene2 %ofoperons… Essentially for each gene, it lists the probability of it occurring in an operon downstream of the given IGR

# 111222

Plan for manual IGR association analysis:

1. Select an IGR which is a binding site for a TF I’ve got binding data for
2. Check typical operon composition of operons downstream of IGR
3. For each gene found in operon except unknown genes, for all occurrences of the gene what percentage are associated to this IGR
4. For gene identified in the above step, see if any of the other IGRs the genes found assoc to are binding sites for a *different* tf. If found at another IGR that binds the same TF note for later but don’t look further for now.
   1. If no TF-binding IGR found for any gene found in original IGR operon, start again from step 1 and select diff igr
5. If diff TF binding IGR found, then look into the TF that regulates it online. See what affects expression/activation of that TF vs the original one. Try to make hypothesis on the effect this has on the cell response
   1. E.g the gene for green is associated to IGR1 which binds TF activated by pyruvate, but in another genome its associated to IGR2 which binds TF activated by lactate: In the first genome its likely then that the cell only appears green under aerobic conditions and in the 2nd only under anaerobic conditions, as these are what lead to pyruvate/lactate production and therefore activate the respective TF.

# 121222

Made a script that reads the collated bsIGR2gene /gene2IGR files and outputs each gene found to be regulated by more than one TF across all genomes (i.e two different IGRs I found assoc to it were known to bind different TFs). Collates the gene stats across genomes aswell for each cluster occ – i.e gene count, percent assoc to whatever igr, expected percent.

Now to look into function of identified genes and how the TFs that regulate them are in turn regulated to see if the differing regulation can result in differing phenotype

Maybe if I get time after looking into a lot, see if theres any that coincide – i.e GENEA – IGR1, GENEA – IGR2, GENEB – IGR3, GENEB – IGR4 : Maybe GeneA-IGR1 and GeneB-IGR3 tend to occur together more often than not, and if one is associated differently, the other is likely to be aswell.

# 171222

Presentation done, got some tips to get more tf binding data

Searching Sequence Read Archive on NCBI

# 181222

Sra toolkit installed and tested, seems viable but slow, going to download FASTQ for as much TF as I can, limiting file size to 10MB at most due to ridiculous file sizes.

# 191222

Only found viable BS data for Ompr and ATF-7. But also found fairly complete reg network for P.Aeruginosa – TF binding sites unlisted but TF to gene associations *are*. Thinking of drawing edges from operons encoding TF to IGRs associated to an operon containing the relevant gene

These new graphs will be kept separate from the main project due to inaccuracy: P.Putida will have different TF-gene associations that aeruginosa, and there will be no validation of the TF to IGR binding.