Universitat Oberta de Catalunya

uoc.edu

Evolution and Transcriptomics in Proteobacteria

Alexandre Armillas Montornés
Màster en Bioinformàtica i Bioestadística
Área 2

Iván Erill Sagales
Carles Ventura Royo

24th June, 2020

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Evolution and Transcriptomics in Proteobacteria*
Nombre del autor:	*Alexandre Armillas Montornés*
Nombre del consultor/a:	*Iván Erill Sagales*
Nombre del PRA:	*Carles Ventura Royo*
Fecha de entrega (mm/aaaa):	06/2020
Titulación:	*Máster en Bioinformática y Bioestadística*
Área del Trabajo Final:	*Área 2*
Idioma del trabajo:	*Inglés*
Palabras clave	*Transcription, Proteobacteria, Python*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

En este trabajo se contempla estudiar una anomalía en el reconocimiento de motivos de unión del represor LexA en *Methylomonas koyamae* y ciertos géneros noveles de Gammaproteobacterias.

Tradicionalmente, se ha considerado que LexA es una proteína cuyo motivo de reconocimiento es monofilético (Alphaproteobacteria reconoce un motivo, mientras que Gammaproteobacteria reconoce un motivo completamente diferente), un caso inesperado, pues LexA se trata de un factor de transcripción involucrado en la respuesta SOS, responsable de coordinar el proceso de reparación de ADN.

La intención es determinar qué eventos, a nivel evolutivo, han resultado en esta situación, con el objetivo de entender mejor como una especie puede alterar completamente el motivo reconocido por un sistema crucial para la supervivencia celular, con posibles aplicaciones en la modificación o sobre como alterar los sistemas básicos de la célula para nuestro beneficio.

Debido al gran volumen de datos, el análisis manual de estos datos no resulta viable, por tanto, se ha utilizado Python, en combinación con el paquete Python CGB (Comparative genomics of transcriptional regulation in Bacteria) para automatizar el proceso de generación de datos, para determinar la prevalencia de tal anomalía en Gammaproteobacterias en taxonomías sospechosas, cercanas a *Methylomonas*, determinar que operones se encuentran regulados para cada motivo celular, y sobreponer estos datos sobre la taxonomía de la selección de Proteobacterias para formular una hipótesis al respecto.

Los resultados se componen por los resultados de CGB, una serie de taxonomías basadas en LexA y en 16S rRNA, y la interpretación de los datos.

Abstract (in English, 250 words or less):

The object of this thesis is to study an anomaly in the recognized motif by the transcriptional factor LexA in *Methylomonas koyamae* and certain novel genera of Gammaproteobacteria.

Traditionally, LexA binding motifs have been believed to be monophyletic, that is, Alphaproteobacteria recognizes one motif, while Gammaproteobacteria recognizes an unrelated motif. An unexpected case, for LexA is a transcription factor in the SOS response, which coordinates the process of DNA repair.

The aim is to hypothesize what events, in an evolutionary scale, have led to this situation, with the goal of better understanding how can the motif associated to such a crucial pathway can change over time, and possible applications when it comes to directing the regulation of the cells, or how to modify such pathways for our benefit.

Due to the large volume of data, manual analysis is not a viable direction, thus, Python will be used to automate the process, alongside CGB (Comparative genomics of transcriptional regulation in Bacteria), to evaluate how widespread the anomalous motif is in suspect Gammaproteobacteria genera, closely related to *Methylomonas,* identify which operons are regulated by each motif in every species, and overlay such data over an established taxonomy tree of Proteobacteria so as to formulate an hypothesis to explain it.

The results are made up by the results from CGB, a series of taxonomic trees for LexA and 16S rRNA, and an interpretation of the data.

Index

1. Introduction

1.1 Context

Transcriptional factor is the name given to any protein with the capacity to regulate the transcription of a number of genes by changing the rate of transcription in these genes, usually, as a response to a stimuli. Transcriptional regulators accomplish such a function by binding to key DNA regions, recognized by a motif, a string of recognized DNA nucleotides, and promoting or blocking the recruitment of DNA Polymerases, these changes may have further impacts as part of a regulatory network.

Binding motifs are normally represented by a sequence logo, a graphical representation of a sequence which represents the amount of information contained in each base (as the total height dimension) and the likelihood for each base to be found in every position of the sequence, two examples of a sequence logo are provided below.

There are a number of genes whose functionality, on a genetic level, has been conserved during the evolution of life but whose actual role may change as a result of alterations in their regulation. In the evolutionary sense, a species is considerably less likely to generate new features *de novo* than it is for existing features to mutate to fulfill a different function, transcriptional regulators are the driving force behind the latter.

Understanding the driving forces behind the evolution of key transcriptional regulators may provide insight on such pathways, and allow us to infer knowledge on how to bend such pathways to target products of interest in closely related taxa. However, such studies are made difficult by the scale of the data, as there exist thousands of described species, each of them with it's own genome.

In this thesis, the cellular SOS response has been chosen as a case study of the evolution of transcriptional regulators. The SOS response is an almost universal system of response to DNA damage in bacteria governed by a single transcriptional repressor, the LexA gene.

In its basal state, the LexA repressor is bound to a specific DNA region through a DNA-binding domain, in the so-called SOS Box, leading to its repressive nature, as the binding of LexA to the DNA prevents the transcription of key genes involved in DNA repair. In the event of stalled DNA replication (recognized by the presence of ssDNA fragments in the replication fork), the protein RecA will activate, changing into filamentous structures[1], and binding to LexA, which starts a process of autoproteolysis, unblocking the SOS Box and activating the expression of the associated genes. This process is regulated by a negative feedback loop, as once the DNA replication continues, the lack of ssDNA strands will allow LexA to take its place in the SOS Box, blocking again

the expression of the SOS Response genes, a schematic representation is provided in (Figure 1).

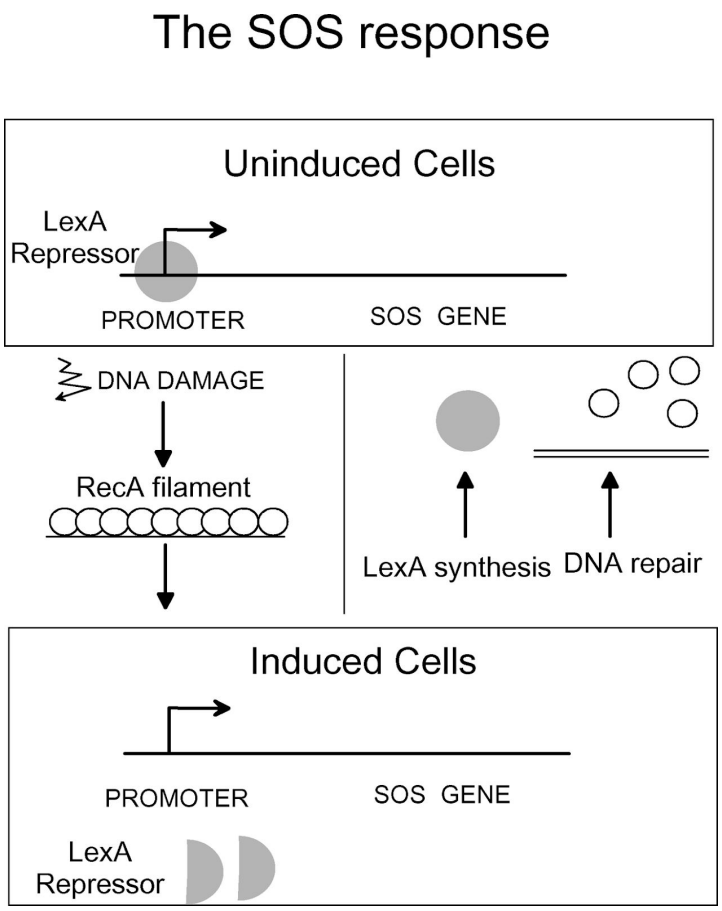


***Figure 1**: Schematic representation of the SOS Response; Michel B (2005) After 30 Years of Study, the Bacterial SOS Response Still Surprises Us. PLoS Biol 3(7): e255. https://doi.org/10.1371/journal.pbio.0030255*

In a single genome, there may exist multiple SOS Boxes of different affinity with LexA (their affinity being defined by their sequence specificity, how strong the binding between transcription factor and DNA is), and as a result, different SOS Boxes may be activated sequentially, such a process allows the escalation of the SOS Response, withholding potentially dangerous error-prone repair genes as a last resort to repair DNA damage, or in the event of long exposure to DNA damage, terminating the cell cycle altogether through apoptosis.

In contrast to most other transcriptional regulators, the sequence specificity (the sequence identity between the binding region and the motif) of the LexA repressor has changed dramatically through evolution, both in terms of what is recognized by each LexA monomer (LexA binds as a dimer), the space between both monomers and the relative orientation of the monomers when bound to DNA. This disparity is most likely due to the prevalence of events of LexA duplication, which allows one of the copies of LexA to mutate without

compromising the SOS response, and where the mutated copy may eventually take over the functions of the main LexA copy.

Traditionally, LexA has been thought to be monophyletic for any given group. That is, LexA recognizes one single, stable motif within large groups of bacteria. One such group is Gammaproteobacteria, which includes *Enterobacteriaceae* and their most famous representative (*Escherichia coli*). In *E. coli* and many other Gammaproteobacteria, LexA recognizes an inverted repeat (or palindrome, CTGT-n8-ACAG)[2], one such example is provided in Figure 2.

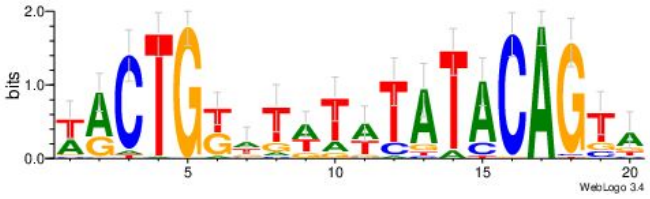


***Figure 2**: Sequence logo for the binding motif of Escherichia coli str. K-12 substr. MG1655; CollecTF database*

Although there have been localized descriptions of *lexA* gene duplication in the context of a mutagenesis cassette[3], such as in some Pseudomonas and Xanthomonas species, the overall consensus is that the CTGT-n8-ACAG is pretty much monophyletic for the class Gammaproteobacteria.

In the Alphaproteobacteria class, LexA recognizes a completely unrelated motif (GTTC-n7-GTTC), as can be seen in Figure 3, consisting of a direct repeat (not a palindrome):

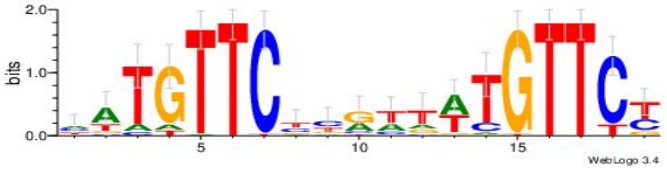


***Figure 3**: Sequence logo for the binding motif of Caulobacter crescentus CB15; CollecTF database*

Recently, novel genera of Gammaproteobacteria have been sequenced. Preliminary analysis of the promoter region of the LexA gene (which is normally self-regulated and therefore should contain an instance of the LexA-binding motif) shows that LexA in these Gammaproteobacteria (e.g. *Methylomonas koyamae* 45378) seemingly targets a motif closely related to the one described for Alphaproteobacteria (GTTC-n7-GTTC). In addition, it appears that some species may harbor two copies of LexA, each one recognizing a different motif.

1.2 Objectives

- Assess the phylogenetic distribution of these two LexA variants
- Identify the overall composition of their regulatory network across Alpha and Gammaproteobacteria.
- Abstract the process behind a Python-based framework due to the scale of the data involved.
- Identify the most likely evolutionary pathway which resulted in this event

1.3 Scope and methodology

For the purpose of automating the process of Comparative genomics, Python has been chosen for its development. Python provides a simple language, easy to understand, and with powerful plugins to connect with online databases or to include 3rd party programs into the pipeline. The following packages have been featured throughout the project:

- Biopython[4]: A suite of Python tools for computational molecular biology
- ETE Toolkit[5]: A package for managing phylogenetic/hierarchical trees, in this case, it has been used to visualize phylogenetic trees
- ProgressBar2: A package for visualizing the progress of certain, slow iterative processes.

To this end, we will use molecular phylogeny methods to reconstruct the most likely evolutionary scenario, and we will use comparative genomics tools to infer the core regulatory network of these two LexA proteins across the Gammaproteobacteria.

The main tool used will be CGB (Comparative genomics of transcriptional regulation in Bacteria) an open-source Python library for comparative genomics of transcriptional regulation in Bacteria. One of the key objectives is to automatically process and generate data into a format readable by CGB.

Other tools used are:

- T-Coffee[6]: T-Coffee is a multiple sequence alignment package. It's main advantage when it comes to the alignment of protein sequences is that it considers 3D structure patterns for the alignment, which is relevant to our case of study (as there should be structurally conserved regions). In this project, the T-Coffee server hosted by the Centre for Genomic Regulation (CRG) of Barcelona has been used, albeit binaries are available for MacOS/Linux operating systems.

– Gblocks[7] : Gblocks is a software which, given the result of an alignment, will filter out poorly aligned regions based on phylogenetic conservation and reducing noise. Resulting into a more compact alignment, built from phylogenetically significant data. The GBlocks server used is hosted by the Institut de Biologia Evolutiva (CSIC-UPF)

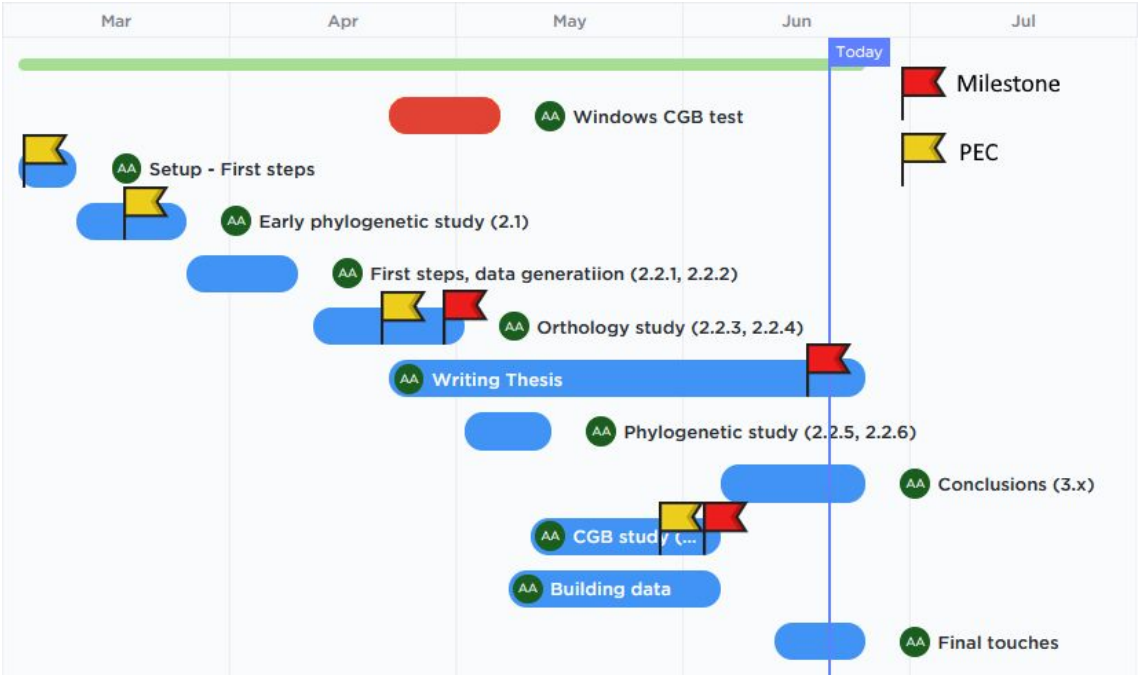
– MrBayes[8,9]: MrBayes is a program to perform Bayesian inference for the creation of phylogenetic models. MrBayes uses Markov chain Monte Carlo (MCMC) methods to estimate the posterior distribution of model parameters throughout a series of generations. The choice for MrBayes allows the phylogenetic trees to be displayed with a support score for each branch, to validate the statistical significance of each part of the tree.

– Oracle VM VirtualBox: CGB requires to be run in an UNIX environment, for this purpose, a lightweight Linux virtual machine has been configured with the Debian distribution to run CGB.

– PyCharm / Spyder: For the purpose of developing the necessary Python scripts, these two integrated development environments have been used. PyCharm has been used for Windows-based Python 3 scripts, while Spyder has been used inside the Linux virtual machine with a Python 2 Conda environment.

– BLAST+ / Clustal Omega: While the use of these tools will not be detailed anywhere throughout the thesis, they are strict dependencies of CGB and need to be installed prior to running CGB. Both tools have been installed inside the Linux Virtual Machine. BLAST+ is used by CGB to find regions of favorable binding with the derived motif in the various genomes provided, while Clustal Omega is the tool used internally by CGB to perform Multiple Sequence Alignment.

1.4 Work Plan



1.5 Expected results

When this project is finished, the expected results will be:

- A phylogenetic tree seeded from the studied Transcription Factor
- A reference phylogenetic tree obtained from a Housekeeping protein
- The results from CGB for the Alphaproteobacteria and Gammaproteobacteria motifs, most importantly, the Heatmaps and operons
- A discussion drawn from the above results, alongside an hypothesis about the ancestral state of Proteobacteria
- The Python script(s) featured throughout the project

1.6 Additional Chapters

1.6.1: Section 2: Development

Section 2 covers the process of development of the script(s) used to generate the results. An emphasis is placed on documenting and justifying design choices and providing a general overview of the process.

This section is split into 9 smaller subsections, each one detailing a single step of the process, from defining the problem, the chosen approach to address the problem, and highlighting information about the process.

Do note that the split is done from an outside perspective, and is not meant to represent the amount of work involved in each step, but to mark the transformation of the data and/or the creation of a certain result. In reality, most of those steps are intertwined in a single script.

1.6.2: Section 3: Conclusions

Section 3 details a series of Conclusions, drawn from the data obtained by the end of section 2, and provides a discussion about them. Additionally, possible improvements over the project have been included for future references, alongside limitations over the scope of this project.

Key highlights from CGB are provided, to give a frame of reference while not compromising the presentation of the data, and some discussion is performed over the data.

Two hypotheses are detailed in section 3.2, to provide explanations to the results obtained, and to highlight possible evolutionary scenarios that have led to such results.

Section 3.3 instead, provides a personal explanation of the project, discussing some prevalent issues and part of the arisen problems, and discussing alternative solutions. Some references about how to continue this project are also discussed, alongside paths that could provide further information to accept or reject the proposed hypotheses.

1.6.3: Section 4: Glossary

While this thesis does not prevalently feature abbreviations, a glossary of terms has been included, where some key concepts are defined.

1.6.4: Section 5: Bibliography

The bibliography that has been referenced throughout the process, alongside citations to reference the Software employed throughout the process.

Citations have been automated through the use of Paperpile.

1.6.5: Section 6: Appendix

The appendix includes information or data whose characteristics make it’s inclusion into the main body of the thesis difficult. This data is referenced at certain points throughout the thesis.

2. Development

Before delving into the bioinformatic approach, a flowchart has been created *a posteriori* to provide a general overview of the process in Figure 5.

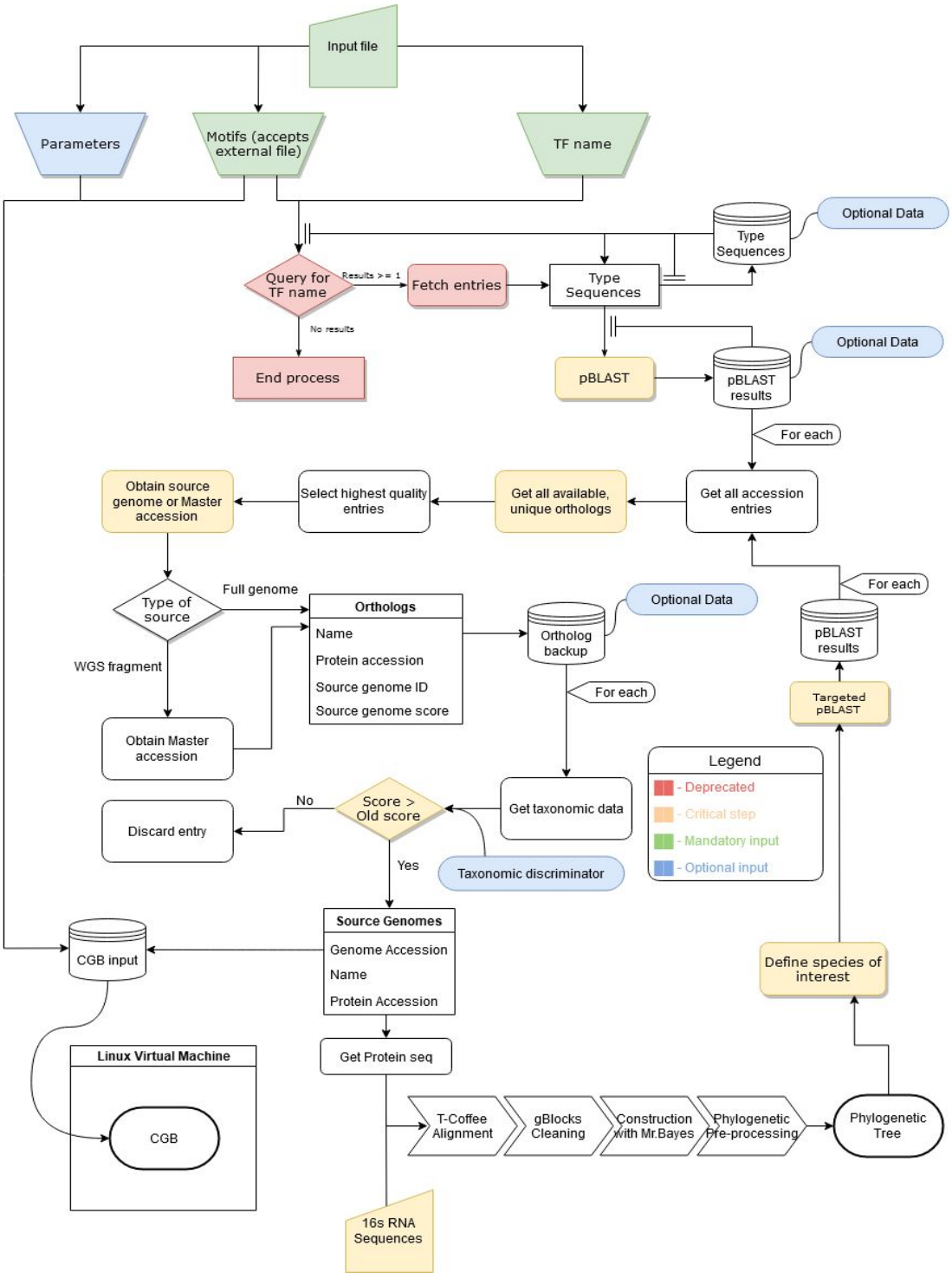


***Figure 5**: Flowchart for the process*

As can be seen in the flow chart, there's one main input (in .json format), with a series of optional inputs:

- Input file: This is a .json file and is mandatory for the purpose of defining the parameters of the study, the following parameters are required:
    - "TF": The Transcription Factor to be studied, no default is provided. "LexA" has been chosen as the Transcription Factor to study.
    - "motifs": A list object. If the list is empty, it will be populated with a set of motifs provided externally through a text file. If a list has been defined as part of the input file, that one will be used instead.
    - "genomes": An empty list object. The "genomes" will be populated with entries for each genome where a valid ortholog has been found.
    - This spot includes all parameters used by CGB, and will be passed as-is to the resulting input file. There parameters will be used in the script where relevant (eg. the "sleep" parameter, which defines the timeout between ENTREZ queries, will be used for the same purpose in the script)
    - Some overloaded parameters are given, which are not used by CGB, but are necessary to obtain the parallel phylogeny to validate the results. The necessary parameters are the following:
        - "housekeeping": The name of the Housekeeping gene used
        - "housekeeping_seq": The type sequence of the Housekeeping gene used. Entries will be generated by BLASTing this sequence against the generated "genomes"
        - "taxselector": A parameter which allows the user to customize the taxonomic discriminator of the study (one type sample is defined for each unique member of the taxonomic discriminator). Due to changes in the script, this parameter is now only used as a default taxonomic selector.
        -
- Motif file: A plain .txt file can be used as the source from which to populate "motifs" in the .json file, this file should contain the following information:
    - Name of the species
    - Accession number of the associated genome
    - All motifs
    - Text blocks are separated by the hidden \n character (newline). Data blocks should be separated by a blank line (\n\n character)

- Type Sequences: The algorithm requires pre-selected motifs sequences from the selected TF. This data should be provided as a .txt file with a name, accession of the genome and all valid motifs. Entries should be separated by an empty line.

- BLAST results: BLAST takes a long time to complete, and offloads the computation to the NCBI servers. By caching the results of BLAST, unnecessary server load and long waiting times can be avoided in repeated analysis. If no files containing BLAST results are found, they will be automatically generated for future use. If BLAST results are found, they will be used instead.

- Ortholog backup: Due to the long time it takes to calculate the full list of ortholog genes (and their associated genomes) a cache is automatically generated. If this cached file is detected in the environment, it is loaded rather than calculating all orthologs again.

The early input file (not to be confused with the output, which is itself an input for CGB) has been included as an annex (Appendix 6.1).

2.1: Obtaining the type sequences

Due to progress and potential issues, this part of the project has been deprecated, it works and provides functional results, but it has not been validated and the chosen approach has some glaring flaws. This section has been included for legacy purposes, and has not been updated alongside the rest of the script.

As a starting point, a series of motifs to study are necessary as a starting point, to populate the 'motifs' list in the final output file, these motifs are tagged with the name of the species and their source, a genome accession ID.

The first step is obtaining the accession ID of the Transcription factor in the associated genome, in theory, this accession ID encodes for the protein containing the motif. This is done by issuing three Entrez queries, a first one towards the 'taxonomy' database, to obtain the taxid parameter for the species name, and a second targeted query to the 'protein' database, with the restriction:

Transcription Factor + "[Gene] AND " + *Species* + "[Organism]"

Thus, we can obtain the accession number of the transcription factor in the source organism, this parameter is annotated as part of the resulting output, despite being seemingly unused.

This leads to one problem, there's documented instances of LexA duplication in the context of a mutagenesis cassette, like in some members of the *Pseudomonas* and *Xanthomonas* genuses, in the event that Entrez returns more than one result, the search is repeated, but the number one is appended to the Transcription factor in the search to obtain the first instance of LexA. This only works in this case, and should not be applied to other Transcription factors unless it's known that the same convention is followed.

A third query matches the accession ID against the 'protein' database to obtain the sequence belonging to the accession ID. These are stored in individual FASTA files, in the input folder, which can be used to access the species name (through the filename), the accession ID in the FASTA header, and the protein sequence.

In the final project, all FASTA sequences have been written manually by selecting reviewed sequences from the LexA transcription factor in various species, and the process detailed in this section has been disabled by default and deprecated.

2.2: Building a repository through BLAST

BLAST (Basic Local Alignment Search Tool) is a well established tool in bioinformatics for the purpose of finding regions of similarity between biological sequences. In this case, protein BLAST (pBLAST) will be used, as aminoacid sequences will be compared against documented proteins.

In our case of study, BLAST provides potential candidates for homology, as most favourable results of a local alignment as HSPs (High-scoring Segment Pairs), these HSPs are then tested against a series of randomized amino acid sequences, determined by a table of residue-by-residue background substitution matrix (eg. BLOSUM[10]), which takes into consideration how likely a residue is to randomly mutate into another. By evaluating how these random sequences align with the original input, it is possible to mathematically define how likely it is for the alignment to be due to chance. This is the e-value of a BLAST result, and will be used as the main discriminator of our search.

By submitting to BLAST our previously defined type sequences, a large library of potential homologous proteins has been built, however, this approach presents the following issue, BLAST is an inherently biased tool. The best results from BLAST (ie. sequences with very high identity and score) are most likely to belong to close relatives of the source of the original sequences. By BLASTing the transcription factor from *E. coli* K12 MG1655, it can be expected for most of the best results to come from the plethora of different *E. coli* strains, this floods the results with useless data, as the purpose is to obtain data from novel genera of Proteobacteria, rather than flood the results with redundant data. To generate a large library of data, the number of requested alignments has been set to 500, additionally, an e-value threshold has been set to $>10^{-10}$ to remove potentially poor homologous proteins from the data.

To better target our BLAST towards species of interest, the only restriction applied is a Class-wide discriminator in the results. If the source species belongs to Gammaproteobacteria, all BLAST results will be searched in other Gammaproteobacteria (taxid[1236]), likewise for Alphaproteobacteria (taxid[28211]). The class to which the source species belongs has been determined through Entrez, by querying the ‘taxonomy’ database. In the event that it is not possible to determine to which class a species belongs, a fail-safe has been implemented, BLAST queries whose taxonomy has not been defined default to the following:

txid1236[ORGN] OR txid28211[ORGN]

To restrict the results to Gammaproteobacteria and Alphaproteobacteria. This fail-safe has yet to be used, as all the type sequences have associated taxonomic data.

This part of the process has been deemed Critical in the flowchart, due to the long processing time, and the impact that querying an incorrect sequence would

have on the entire process, and is the main reason why the source sequences have been manually defined, rather than automatically generated.

Due to the long processing time, all results from BLAST are procedurally cached in the output (under ~/output/'*TF name'*/'*Number of blast hits*'/'*name of the source organism*'.xml, relative to the workspace), this way, it is not necessary to repeat the lengthy process of querying BLAST. And, in case that BLAST results are found in the above mentioned path, the entire script will load the BLAST results and start from the next section.

2.3: Building a library of Orthologs

Starting from a series of BLAST objects, it is necessary to extract all the data contained within these. To do so, all results of the BLAST alignments have been iterated, and the unique accession numbers associated with the alignment have been stored for further analysis. Additionally an identity test has been introduced (for the purpose of discarding alignments that match too well), in order to consider an entry as an ortholog, instead of a redundant sequence, it requires less than 95% identity match. This filter should prevent the introduction of repeated/redundant sequences into the library.

It should be noted that results with a poor alignment should never be discarded, as they provide valuable information on potentially distant relatives (and during the following steps, those results should be phased out if better options have been found)

From each entry in this early list of accession IDs corresponding to orthologous genes, the end goal is to obtain a dictionary, with unique entries for each accession ID, consisting of:

- The accession ID of the orthologous gene
- The accession ID of the genome containing the orthologous gene (or a list of genomes, in the event of a WGS submission)
- The position of the ortholog (start, end and DNA strand) within the genome
- The score associated with the genome
- The name of the genome

This is done by querying the IPG (Identical Protein Groups) database through ENTREZ, IPG provides single entries for every protein record (obtained from the BLAST alignments) which are annotated with the Accession ID of the genome where they have been found in, alongside the location of the protein in the genome.

WGS (Whole Genome Shotgun) submissions have been mentioned as a special case. These submissions are available in a fragmented state, where each individual genome contains a part of the whole, and they are coordinated from a so-called ‘Master Accession’, a genome accession which points out to all associated fragments. All WGS submissions follow the same pattern:

Prefix + Unique ID + Number

Where the ‘Prefix’ (eg. NC_) contains information about the Database and the type of the submission, the ‘Unique ID’ is unique for each species submitted, and the ‘Number’ relates to the number of the sequence within the WGS, in sequential order. The ‘Master Accession’ is always the sequence with number zero, so we can access the Master Accession from any of the WGS fragments,

by fetching the accession where all numbers are zero (and thus, access the complete list of WGS fragments).

WGS fragment → Master Accession → Complete list of WGS records

In the event that our sequence is part of a WGS project, the complete list of WGS records is used as the parameter ‘genome’.

Scores have also been assigned to each genome, these are based on their prefix, and provide a rough indicator of the type of submission and how many reviews it has undertaken. The genomes have been scored in the following order: Complete RefSeq genomes > complete GenBank genomes > RefSeq WGS records > GenBank WGS records > direct GenBank submissions.

This step has been deemed critical due to how any issue in the process will cascade into the entire project and the amount of time necessary to compute the complete dictionary of orthologs, as a combination of the large amount of data and the requisite of a timeout between ENTREZ queries, this process has been benchmarked at ~8 hours, by using the current BLAST results. Thus, once the list of orthologs has been calculated, it is dumped into a .json file for future use, to avoid having to repeat the process. It is possible to recalculate all orthologs by deleting the cached .json file, however.

A progress bar has been implemented to visualize the scope of the process, and the list of orthologs has been built in Debug Mode to ensure that, in the event of an http error, the progress is not completely lost.

2.4: Filtering with a taxonomic discriminator

The current volume of data is too large to be realistically analyzed, so it is necessary to filter the data into a more manageable state (without compromising the amount of information), for this purpose, data will be organized according to their taxonomic data, depending on their class. Gammaproteobacteria are filtered by unique taxonomic families, while Alphaproteobacteria are filtered by unique taxonomic orders.

Additionally, to ensure the representation of novel groups lacking taxonomic data at the desired level, 'order' will be used in case that a unique 'order' for a Gammaproteobacteria without any defined family appears. This presents a fallback in the event that an unclassified, unique Order, without an assigned family is featured in the results.

The first step in this process is to remove all entries whose class is neither Gammaproteobacteria nor Alphaproteobacteria, this may appear to be redundant, since the original library of BLAST results has already been restricted to, at least, one of the two classes, however, due to the prevalence of "Multispecies" records in the library of Orthologs, some species outside the scope of the study have been observed to make it in by sharing a "Multispecies" record with a valid species, thus, it is necessary to again remove all entries belonging to these species.

This process results into a list of dictionaries, named after their desired taxonomic classification ('order' for Alphaproteobacteria, 'family' for Gammaproteobacteria), represented by a single genome (or a single WGS scaffold, which may contain many entries, if no better genome has been found).

Orders/Families have been chosen because they present a reliable record of the evolution of Proteobacteria and the smaller subset of results should be notably less expensive to compute while maintaining a good enough resolution. However, due to this choice, we are considering all members inside a family equal, which may not be a correct assumption. The choice makes it easy to distinguish between Alpha and Gammaproteobacteria, through the -aceae or -ales suffix in the entries.

It should also be noted that this is a very reductionist approach, and the selected type genomes are not selected based on their status, just the quality of the entry. This has not been deemed a problem, since it is out of the scope of the project to create a perfectly representative tree of life, the current goals are to generate a series of reference points to compare the DNA binding motifs for the transcription factor LexA and determine possible sister taxa for *Methylomonas koyamae*, to add to the 'groups of interest'.

At this point in the process, it is deemed 'finished', all data is outputted as a correctly formatted input file for CGB, and the following steps may be safely skipped.

Since by using family/order name we forgo a lot of information about the source, the following table cross-references the families and orders with the source species of their genome (accession numbers for the genomes can be cross-referenced from the CGB_input.json file; in Annex 6.2):

```
Rhizobiales:        Agrobacterium fabrum str. C58
Caulobacterales:    Caulobacter vibrioides CB15
Rhodobacterales:    Hirschia baltica ATCC 49814
Parvularculales:    Amphiplicatus metriothermophilus strain CGMCC
Enterobacteriaceae: Escherichia coli str. K-12 substr MG1655
Yersiniaceae:       Yersinia pestis CO92
Thiotrichales:      Thiotrichales bacterium
Pasteurellaceae:    Haemophilus_influenzae_Rd_KW20
Vibrionaceae:Vibrio cholerae O1 biovar El Tor str. N16961
Pectobacteriaceae:  Dickeya zeae Ech586
Idiomarinaceae:     Idiomarina sediminum DSM 21906 G535
Alteromonadaceae:   Paraglaciecola arctica BSs20135
Orbaceae:           Gilliamella apicola strain wkB7
Shewanellaceae:     Shewanella sediminis HAW EB3
Aeromonadaceae:     Aeromonas salmonicida
Budviciaceae:Budvicia aquatica DSM 5075
Morganellaceae:     Proteus columbae strain T60
Magnetococcales:    Magnetococcus marinus MC 1
Holosporales:Holospora undulata HU1
Rhodospirillales:   Rhodospirillaceae bacterium SYSU D60006
Parvularculales:    Amphiplicatus metriothermophilus strain CGMCC
Sphingomonadales:   Zymomonas mobilis subsp. mobilis str. CP4
Emcibacterales:     Emcibacter nanhaiensis strain MCCC 1A06723
Methylococcaceae:   Methylomonas koyamae strain LM6
Chromatiaceae:      Candidatus Tenderia electrophaga isolate NRL1
Pseudomonadaceae:   Pseudomonas putida W619
Erwiniaceae:        Erwinia tasmaniensis Et1/99
Wenzhouxiangellaceae: Wenzhouxiangella sp. XN24
Cellvibrionaceae:   Cellvibrio japonicus Ueda107
Psychromonadaceae:  Corallincola sp. C4
Ferrimonadaceae:    Ferrimonas balearica DSM 9799
Pseudoalteromonadaceae: Pseudoalteromonas atlantica T6c
Colwelliaceae:      Colwellia sp. Arc7 635
Xanthomonadaceae:   Xanthomonas campestris str. ATCC 33913
Rhodanobacteraceae: Dokdonella koreensis DS 123
```

2.5: Building a phylogenetic tree

This step has two main objectives, first, to determine closely related taxa to *Methylococcaceae* (the family in which *Methylomonas koyamae* belongs), which may present the same abnormality in their LexA binding motif, and second, to validate the representativity of the genomes. If a taxonomic tree built from the sequences fails to properly classify the entries, the data can be considered to be unreliable, and so would be the results.

During the entire process, some data has been dragged throughout the process, for example, the relationship between a protein accession ID (from BLAST) and each accession genome containing the protein, this data has been re-used in this step with the purpose of, parting from a list of genomes, obtain the Accession ID from which the genome has been obtained, and, having the Accession ID, extract the sequence of the transcription factor that has been found originally by BLAST. This list of transcription factor sequences will be used to seed a phylogenetic tree, but before that, it is necessary to apply some transformations:

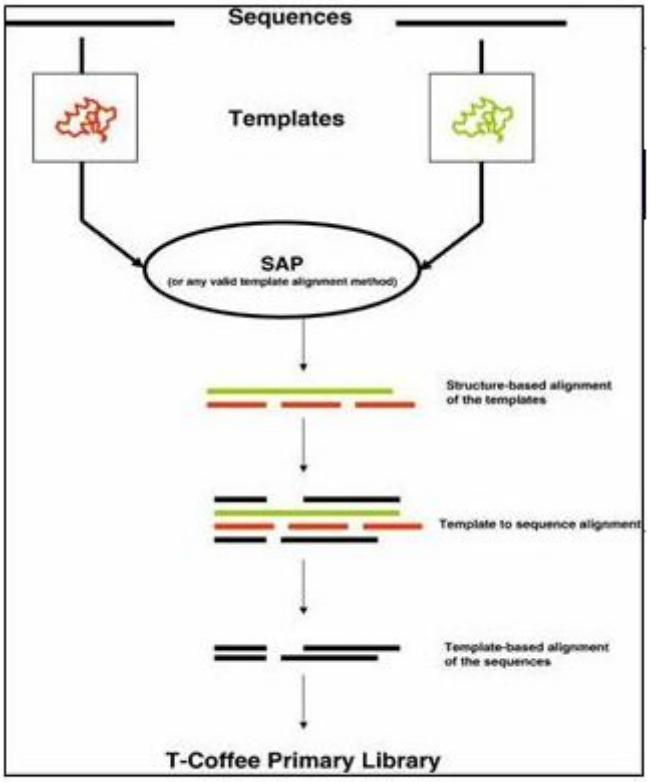


Figure 6: *How does T-Coffee align sequences?; The T-Coffee project: tcoffee.org*

The list of sequences has been aligned through a Multiple Sequence Alignment with T-Coffee[6] (server hosted by the Centre for Genomic Regulation (CRG) of Barcelona), the general process by which T-Coffee aligns sequences is shown in Figure 6. T-Coffee supports multiple sequence alignments of uneven length through the Expresso Structural Alignment[7].

T-Coffee has been chosen to perform the alignments due to its Structural Alignment option, which takes into consideration the 3D structure of the proteins.

The main difference between normal T-Coffee alignment and the Expresso option (used in this project) is that Expresso obtains the Templates to align through a BLAST search.

This process results in a FASTA file containing the resulting alignment between the library of LexA sequences.

It should be noted that, in this instance, a server has been used to perform the alignment, but T-Coffee also provides downloadable binaries for UNIX and Mac Operating Systems (Windows binaries can be manually compiled, but the process is noticeably more difficult, and provides little benefit compared to the online server). It should be possible to integrate T-Coffee into the pipeline without relying on an online service provided the computer is using an UNIX system. While future steps will make use of a UNIX virtual machine, the nature of a virtual machine limits processing power, so it would not be viable to move the entire process inside the Virtual Machine.

Next, it is necessary to extract all phylogenetically significant data from poorly aligned regions, this is done by submitting the alignment to GBlocks[8] 0.91b with default parameters.

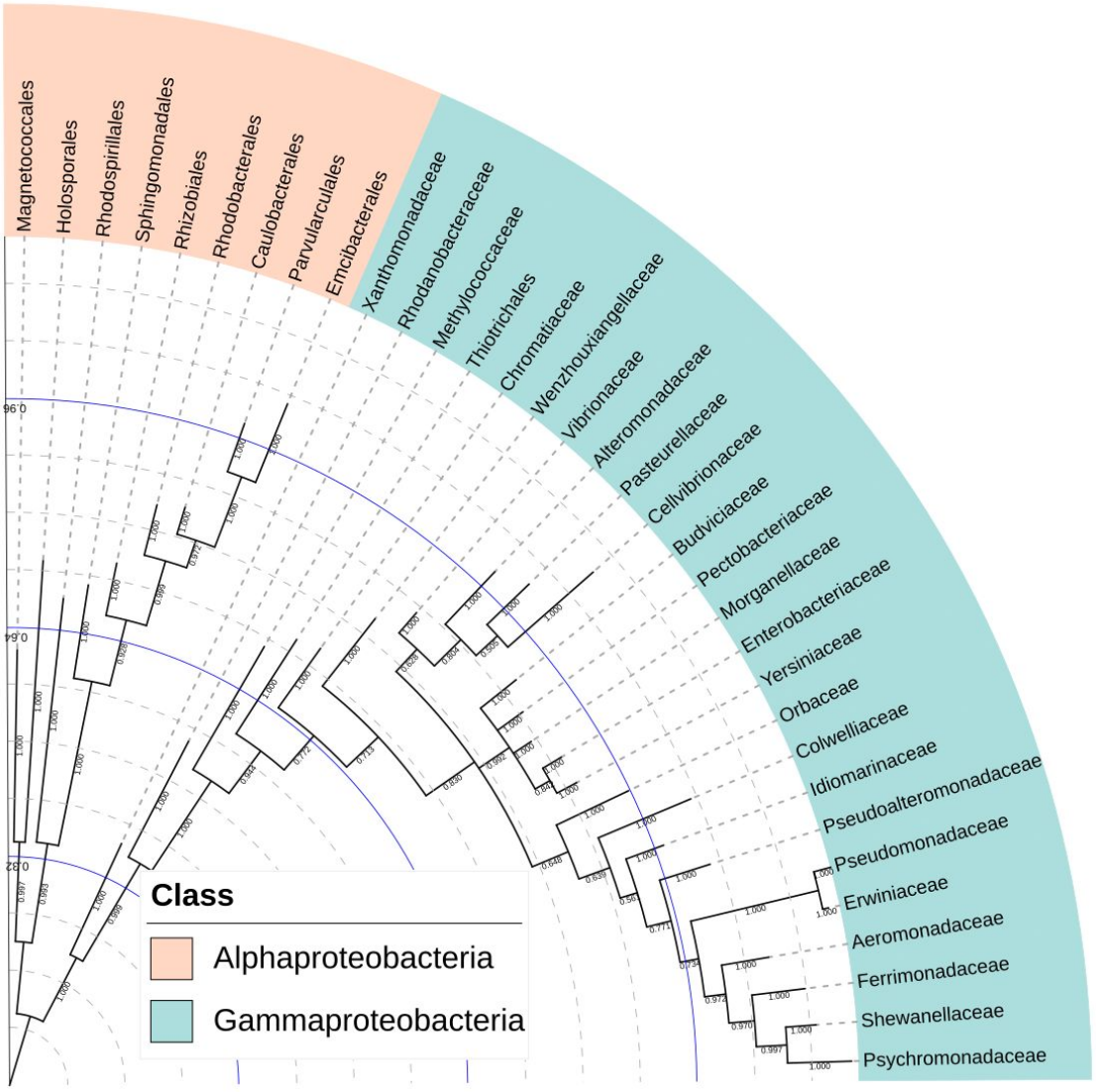
> *"Gblocks is a computer program written in ANSI C language that eliminates poorly aligned positions and divergent regions of an alignment of DNA or protein sequences. These positions may not be homologous or may have been saturated by multiple substitutions and it is convenient to eliminate them prior to phylogenetic analysis. Gblocks selects blocks in a similar way as it is usually done by hand but following a reproducible set of conditions. The selected blocks must fulfill certain requirements with respect to the lack of large segments of contiguous nonconserved positions, lack or low density of gap positions and high conservation of flanking positions, making the final alignment more suitable for phylogenetic analysis. Gblocks outputs several files to visualize the selected blocks. The use of a program such as Gblocks reduces the necessity of manually editing multiple alignments, makes the automation of phylogenetic analysis of large data sets feasible and, finally, facilitates the reproduction of the alignments and subsequent phylogenetic analysis by other researchers."*

-Castresana, J. from the official GBlocks documentation

This results in a smaller alignment of phylogenetically significant 'blocks', from which to finally seed a phylogenetic tree for the transcription factor. The phylogenetic tree has been built through Bayesian Inference with MrBayes. MrBayes has been left to run until the average standard deviation of split frequencies is less than 0.01, or until the analysis reaches 1500000 iterations. MrBayes has been chosen for providing alongside the resulting Tree, a set of support parameters stating the certainty of the branching, for this purpose, any probability under 0.8 should be considered suspect.

To finalize the process, the presentation of the phylogenetic tree has been improved with the Interactive Tree of Life[11] (iTOL).

The resulting tree for the LexA proteins is shown in Phylogeny 1.



Phylogeny 1: *First LexA phylogeny. The average standard deviation of split frequencies is at 0.02 after 1500000 generations. Midpoint rooting has been performed to aid visualization.*

None of the branches and nodes present statistically dubious probabilities, of less than 0.5, and the tree has correctly separated Alphaproteobacteria from Gammaproteobacteria. However, this section will be followed by a phylogenetic study of the housekeeping protein 16S rRNA so as to validate these results.

While the tree presents a degree of uncertainty when it comes to the placement of certain groups of Gammaproteobacteria, there's no indication of very poor predictions (less than 0.5), and the tree is not supposed to provide an accurate representation of the evolutionary history of LexA, this tree is meant to define the taxa containing the closest relatives to the *Methylococcaceae* protein, with an anomalous motif, as potential candidates for harboring the same motif.

As it can be observed, there's evidence to consider the order *Thiotrichales* and the families *Chromatiaceae* (and *Wenzhouxiangellaceae* due to sharing a taxonomic order with *Chromatiaceae*) as potential species of interest.

2.6: Cross-Validation with a Housekeeping Protein

It is important to note that the phylogenetic tree obtained from the protein lexA is not necessarily representative of the evolutionary history of alphaproteobacteria and gammaproteobacteria, since it's not currently known how the built phylogeny fits the known evolutionary history. So as to identify inconsistencies, it is necessary to contrast the lexA phylogenetic tree with a validated frame of reference, to do that, we need to define a Housekeeping gene.

Housekeeping gene is the name given to a group of genes with a very low rate of mutation, which allows them to be used as a frame of reference for the study of evolution. Usually, such genes have key roles in regulating basic cellular functions, thus, mutations in these genes usually lead to negative selection, which makes them relatively stable throughout evolution. Some proposed candidates for bacterial housekeeping genes are the following:

-*16sRNA:* This gene encodes the protein responsible for shaping the small subunit of the Ribosome

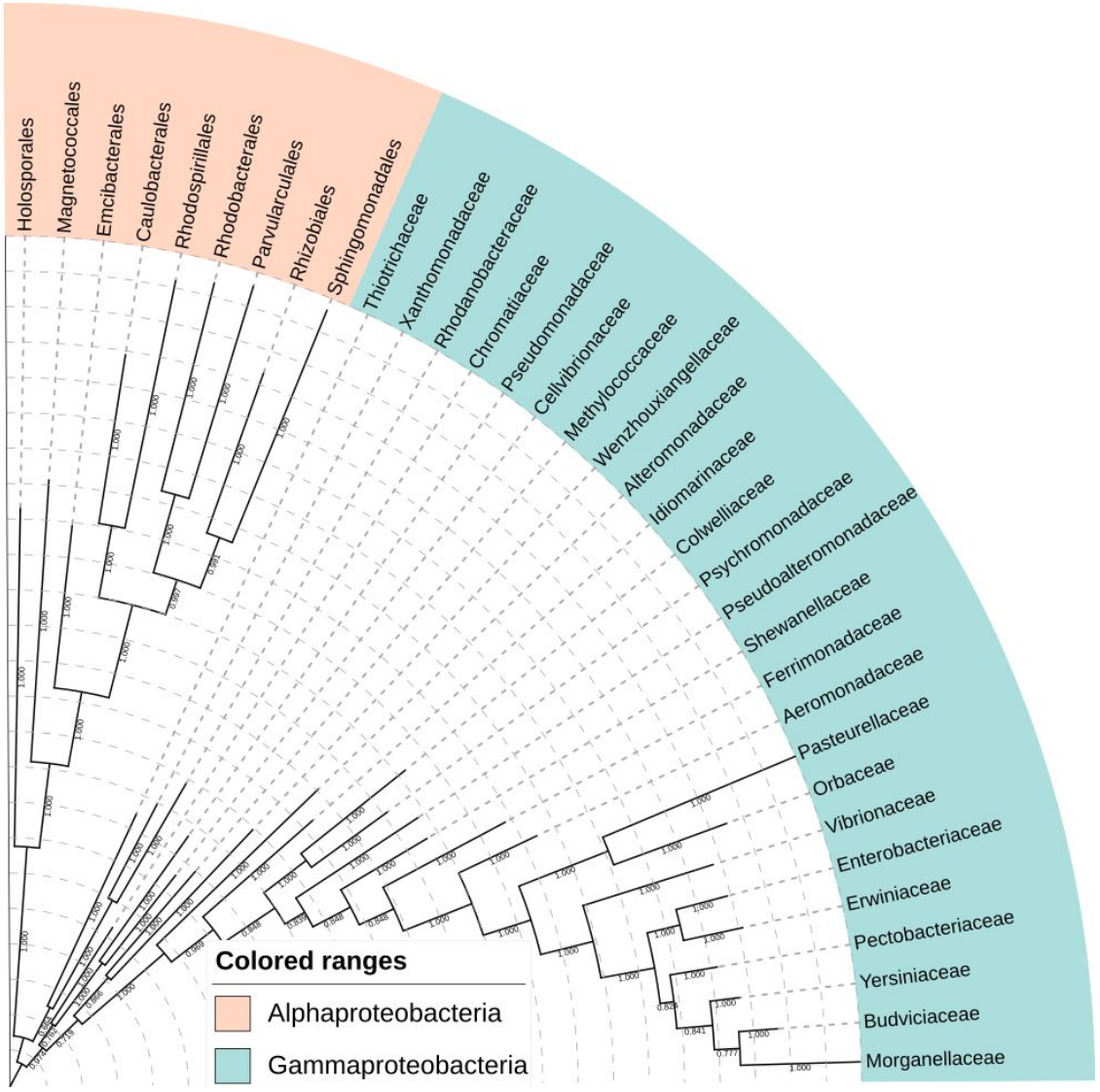
-*RecA:* Like lexA, the RecA gene is involved in the bacterial SOS Response as the catalyst, by recognizing stalled DNA replication through ssDNA and leading to the autoproteolysis of lexA

-*dnaE:* Responsible for encoding for the subunit alpha of DNA polymerase III

In this case, RecA has been discarded, one thing to keep in mind when selecting a housekeeping gene, is that, ideally, such a gene should not be involved nor interact with the gene/pathway being studied, as their close relationship may lead to a shared evolutionary history, and such a comparison may not be representative.

dnaE has too been discarded due to the existence of multiple paralogous genes, while the main dnaE sequence is a valid Housekeeping gene, this gene has suffered multiple events of duplication throughout evolution, and these events present an inconvenience for the phylogenetic analysis with the developed methodology.

It is necessary to rethink our approach to validating phylogeny, so 16S rRNA will be used instead, all 16S rRNA data has been obtained manually from EzBioCloud[12]. Due to the 16S rRNA sequences being given as DNA sequences, the T-Coffee alignment has been done as an M-Coffee alignment, specific to align DNA and with support for misaligned sequences, and the resulting alignment has been cleaned with GBlocks through the DNA option. Again, MrBayes has been used to infer the phylogeny behind the Multiple Sequence Alignment and the result has been rendered and annotated with the Interactive Tree of Life.



Phylogeny 2: *16S rRNA reference tree, the average standard deviation of split frequencies has reached the threshold of <0.01 after 500000 generations. Midpoint rooting has been performed to aid visualization.*

The phylogeny deduced from the 16S rRNA, shown in Phylogeny 2, presents a similar pattern to represent the evolutionary history of Alphaproteobacteria and Gammaproteobacteria. While the shape of the tree is not too dissimilar to the phylogeny from LexA, it does not showcase the same degree of uncertainty that the support data for the tree belonging to LexA has shown when it comes to the classification of certain Gammaproteobacteria.

Having validated the overall distribution, the species of interest have been chosen as the closest relatives to *Methylococcaceae*, as presented in Phylogeny 1. The LexA phylogeny has been chosen

2.7: A species-level zoom on potential candidates

One of the main issues presented by the use of BLAST is how the first alignments have been consistently sourced from close relatives of the source species. This is an issue, since we are most interested in novel genera of Proteobacteria, which BLAST does not present reliably provide unless the number of hits is set to >1000 (which increases computation time in all other parts of the process), thus, it is necessary to consider an approach to introduce potentially suspicious taxa without drastically increasing the data to compute, as the current sample size of potentially anomalous taxa is limited.

To better identify abnormalities in the suspicious taxa, a smaller taxonomic classification will be used to locally increase the resolution of the data. As we do know that *Methylomonas koyamae* presents such an anomaly, all other members of it's family, *Methylococcaceae*, are prime candidates.

Evidenced by the LexA phylogeny, the families *Chromatiaceae* and the order *Thiotrichales* have been found to be close relatives to *Methylococcaceae* so all species found belonging to these taxonomic orders have also been included in the study, however, a problem is presented, both of those taxonomic orders have one single representative sequence throughout the BLAST library (in fact, the single *Thiotrichales* entry doesn't have any annotated family or species, leading to the inclusion of an exception just to prevent it from being discarded), so it is necessary to pad the input file with additional genomes belonging to these taxonomic orders of interest.

It should be noted that only the LexA phylogeny has been used, and not the more representative 16S rRNA phylogeny, this has been done this way because it has been considered that protein similarity may be a better indicator to identify potential abnormalities in the recognized motifs, instead of using phylogenetic evidence. By using LexA instead of 16S it's possible, the additional focus will be spread throughout a wider range of Gammaproteobacteria, allowing the indirect inference of the state of genera, albeit both approaches have their own merits.

For this purpose, three new BLAST queries have been issued, all three with the lexA sequence from *Methylomonas koyamae*, the inferred closest relative to the three taxa, targeting the first 100 alignments for *Methylococcales* (taxid:135618), *Chromatiales* (taxid:135613) and *Thiotrichales* (taxid:72273). Orders have been chosen to maintain consistency with *Thiotrichales*, since there's no families assigned this order in the list of orthologs, and to limit the amount of data, since *Chromatiales* contains two families of interest.

The LexA sequence for *Methylomonas koyamae* is the following:

```
>LexA_Methylomonas_koyamae_WP_064040081
MKPLTHRQQQILDFIEHTLAREGFPPTIAEIAAAFGMGSGNAIRGHLQALAKKGAIQLTPGASR
GIRLLHPGTDQGLPLIGRVAAGQPILAEQHIEGYCQIGPELFQQRADYLLRVHGLSMRDAGILD
```

```
GDLLAVQRRPDARSGQLVVARIGDEATVKRLRLDGDIAYLEPANPDFTTIRIDLRRDALAIEGI
VVGVIRRIAL
```

The resulting BLAST alignments have been subjected to the same process, but instead of families, they will be classified by their unique source species. The source species can be defined with the “Species” parameter, found when the “taxonomy” database was previously queried to determine the taxonomy of the source of a genome, or, in case that it has not been defined, as the combination of the two first words in the NCBI submission, for example, the submission "Agrobacterium_fabrum_str__C58_chromosome_circular" would be included as “Agrobacterium_fabrum” if there’s no species information in the taxonomy associated to that genome. This less restrictive approach has allowed the inclusion of much more data than would have been possible otherwise, as most submissions are missing the “Species” tag, especially when it comes to the study of obscure taxa.

The result of this process is a new CGB input file, but only containing a repository of genomes representative of species belonging to *Methylococcales*, *Chromatiales* and *Thiotrichales*, all of which are plausible suspects of harboring anomalous LexA binding motifs. The ‘genomes’ determined in this input file have been merged with the existing CGB_input.json into one single input file featuring all entries from both files, while repeating the entire process, while accounting for the new data is perfectly possible, it would require a large time investment.

The resulting input file has had it’s “genomes” cleaned by another small script, so as to remove repeated entries (entries sharing the same name) and redundant entries (entries sharing the exact same genome accessions)

2.8: Running CGB

As described in the introduction, CGB (Comparative genomics of transcriptional regulation in Bacteria) is an open-source Python library for comparative genomics of transcriptional regulation in Bacteria.

Before proceeding, a Linux distribution presents the easiest way to run CGB, thus, a virtual machine has been set up with a Debian distribution for the sole purpose of running CGB. And a shared folder has been set up to provide a means to interact with the host computer (mostly, to pass the input files and retrieve the output).

Additionally, BLAST+ and Clustal Omega have been installed inside the Virtual Machine, as CGB depends on both of these applications.

CGB will, from the starting list of motifs, reconstruct the regulatory network (regulon) in which the derived motif (built as a consensus sequence between all inputted motifs) is involved. This is done by downloading the genomes introduced as the input as a local cache, finding regions of high affinity, where the derived motif could reasonably bind (which are determined through the use of Position Specific Weight Matrices in downstream regions of the gene through a local BLAST), fulfilling its purpose as a transcription factor, and then determining the operon regulated downstream from the selected regions.

The following information about the outputs has been taken from the public CGB repository, all outputs will be saved in the working directory where CGB has been called from, in a new ‘output’ folder:

- user_PSWM/ contains the user-provided binding motifs in JASPAR format.
- derived_PSWM/ contains binding motifs in JASPAR format, tailored for each target genome combining all the evidence from each reference motif.
- identified_sites/ contains identified binding sites and information such as their genomic locations, downstream regulated genes and their functions. Predicted binding site data is saved into CSV files, one for each target genome.
- operons/ contains the operon predictions of each target genome, saved as CSV files.
- orthologs.csv contains the groups of orthologous genes and their probabilities of regulation.
- phylogeny.png is plot of the phylogenetic tree.
- ancestral_states.csv has the reconstructed state of each gene in all ancestral clades. For each target species and ancestral clades, the states are
    - P(1), the probability of TF *binding*
    - P(0), the probability of TF *not binding*
    - P(A), the probability of *absence* of the gene.

- plots/ folder contains the visualization of the results.

-I. Erill, from the official CGB repository; https://github.com/ErillLab/cgb

It is known that there's two different binding motifs, one traditionally assigned to Alphaproteobacteria and another assigned to Gammaproteobacteria.

Since the motifs included in the input file contain both motifs, they should not be run together, thus, two copies of the input file will be run sequentially; in the first copy (named CGB_input_alpha.json), all motifs belonging to Gammaproteobacteria have been deleted, leaving only Alphaproteobacteria motifs. This will serve as validation, since it is known that the motif associated with Alphaproteobacteria is a direct repeat, GTTC-n7-GTTC, and CGB should predict such a motif as the derived motifs.

The same process will be applied to Gammaproteobacteria, deleting all motifs associated with Alphaproteobacteria species, which can be expected to deduce the palindromic motif CTGT-n8-ACAG.

Do note that no species suspicious of presenting abnormal motifs have been included in the motifs, as their inclusion presents a new, unknown factor which would influence in the process of deriving a consensus motif. Instead, suspect species have been only featured as 'Genomes', in which the motif will be searched, alongside the repository of Proteobacterial genomes.

Once CGB has finished running, two heatmaps should be obtained (alongside the regulatory networks from which they have been built, as a .csv file), one for the Alphaproteobacteria (GTTC-n7-GTTC) motif, and another for the Gammaproteobacteria (CTGT-n8-ACAG) motif, highlighting each instance of binding in the defined genomes. From these two, the heatmap belonging to the Alphaproteobacteria motif should be most important, as there's a strong chance that binding sites will be detected in the suspicious genomes (of species belonging to the *Methylococcaceae*) family

However, it should be noted that some of the genomes selected by the script presented serious problems that halt any attempt of analyzing with CGB, this has been expected, and has been the main reason to work with large sets of data, as some of them will end up removed from the study, in most cases, due to missing the Transcription Factor or due to a malformed genome alias file.

Due to the time issues when it comes to processing all the data, it has been deemed necessary to compromise the volume of data, thus, any sample that is not annotated to an existing species has been manually purged from the input file. This includes *sp.* entries, "Unclassified" entries and uninformative names. The raw, unfiltered list of genomes is included as part of the annex and should be ready to run (Appendix 6.2)

Here, all the problematic genomes have been described, and the course of action taken has been documented:

- *Enterobacteriaceae:* The determined input, NC_000913.3, causes the process to crash without throwing any kind of exception, thus, the cause of the error remains unknown and has not been possible to diagnose (albeit it's likely due to a local error, as it is the genome of reference for *E. coli* strain K12). Due to the importance of *Enterobacteriaceae,* it's associated genome has been replaced with NC_002695.2 (*E. coli* strain Sakai)

- *Budviciaceae, Idiomarinaceae, Pararheinheimera texasensis, Rheinheimera perlucida, Thioalkalibacteraceae, Rheinheimera tuosuensis, Fangia hongkongensis, Fastidiosibacteraceae, Thiofilum flexile, Thiofilaceae, Methylobacter luteus, Thiothrix sp*: This collection of entries have been removed from the study, as no TF instances could be found in their genomes. Observing the entries, it has been noted that this lack of TF instances stems from an incomplete genomic record, as all these entries are comprised of WGS scaffolds, and these scaffolds do not follow the expected sequential pattern of a WGS scaffold, it's likely that the genome region containing the Transcription Factor is not part of the WGS scaffolds found.

- *Methylocaldum szegediense, Methylobacter marinus*: The genome associated with this species of interest has been removed from the study, as the downloaded genome appears to be missing key information to allow it's study. Including these species causes CGB to crash.

- *Methylococcaceae, Piscirickettsiaceae, Thiolinaceae, Thiotrichaceae, Chromatiaceae:* All these families present redundant data, since they share a genome accession with one of their species (evidenced by a length-zero distance in an early taxonomic analysis in CGB)

After cleaning the CGB_input file of malformed data, CGB has been called from a Python script, which will provide the results from which a series of conclusions will be deduced (in section 3.)

This step of the process has been deemed critical due to being the source of the final results, thus, the validity of the results needs to be ensured. Also, the process of downloading all the input genomes into a local cache takes ~6 hours (and requires a stable internet connection) while running CGB over all the genomes has been benchmarked at ~7 hours per run. These limitations are thought to arise from the use of a Virtual Machine on a laptop device, a more powerful computer may present better performance.

2.9: Reading the results from CGB

CGB outputs a multitude of files, so this section will document the data used in the generation of the discussion. During this study, only the following files will be used:

- plots/heatmap_light.svg: The more compact version of heatmap.svg, trades the direct visualization of the accession id for each cell for a more compact showcase of the plot

The heatmap plots displays information about the phylogeny for the Transcription Factor (in this case, LexA) against a series of proteins which present positive downstream binding for the derived motif in at least one of the input genomes. The proteins are ordered by their mean probability of regulation in all genomes.

A green square indicates positive regulation from the transcription factor to the downstream region of the gene. Shades between red and green display the probability of the transcription factor binding, the more green, the stronger evidence for binding.

A red square indicates that no evidence of regulation has been found for the transcription factor, however, the gene itself is present in the genome.

A blue square indicates that the gene in question has not been found in the genome.

- plots/binding_motif.svg: This plot provides the derived consensus motif, built from the input motifs. In this case, the purpose of this plot is to validate the motif, since it should be expected for the motif derived from Gammaproteobacteria to be the inverted version of CTGT-n8-ACAG, while the derived motif for Alphaproteobacteria should be the inversion of GTTC-n7-GTTC.

- orthologs.csv: This file contains all the information relating to orthology, as comma-separated values, it represents the main source of information for the heatmap plot, each row provides information for a single ortholog group (group of homologous gene found in different species related by linear descent) and contains the following information

- average_probability: The average probability of binding only in the genomes where an ortholog to the transcription factor has been found

- average_probability_all: The average probability of binding in all genomes

- ortholog_group_size: Displays in how many of the genomes an ortholog for the target protein has been found

And, for each input genome:

- probability: The probability of downstream binding for the transcription factor in the ortholog found in the species.

- locus_tag: An identifier to locate the ortholog within the genome

- protein_id: The Accession ID corresponding to the ortholog protein

- product: The description corresponding to the Accession ID, indicative of the ortholog function

- operon_id: Orthologs are grouped into an operon id. If two different ortholog groups share the same operon_id, it's evidence that both proteins are part of the same operon

- paralogs: If there's any, displays information about paralogous genes found. A paralogous gene is the name given to a gene originating from an event of duplication, but which eventually radiated to fill a different function than it's original copy.

3. Results

3.1: Results

Due to the length of the plots resulting from CGB, the results have had to be included as part of Appendix 6.3, available inside of the attached appendix file. Instead, in this section, the most significant aspects of the results are showcased.

It should be also noted that the heat maps belonging to Alphaproteobacteria and Gammaproteobacteria do not result in the same phylogenetic tree, this will be explained below due to the presence of LexA paralogous genes, as each CGB study may pick a different LexA to build the phylogenetic tree with (in the event that there's two equivalent choices). However, both trees have very similar branch structures, splitting Alphaproteobacteria from Gammaproteobacteria as the first branching, so the results have been split into 4 plots, two plots for the Gammaproteobacteria motif (one showing the branch containing Gammaproteobacteria and another showing the branch containing Alphaproteobacteria) and two more plots for the Alphaproteobacteria motif.

3.1.1: Plots

Gammaproteobacteria motif:

The plots relating to the Gammaproteobacteria motif provide a control sample to evaluate against. The top results from the data have been split between Figure 7 and Figure 8.

Figure 7 shows Gammaproteobacteria alongside the sequences that have been found to be phylogenetically related, all placed together in a single branch.

Figure 8 shows the other half of the data, showing the branch containing Alphaproteobacteria.

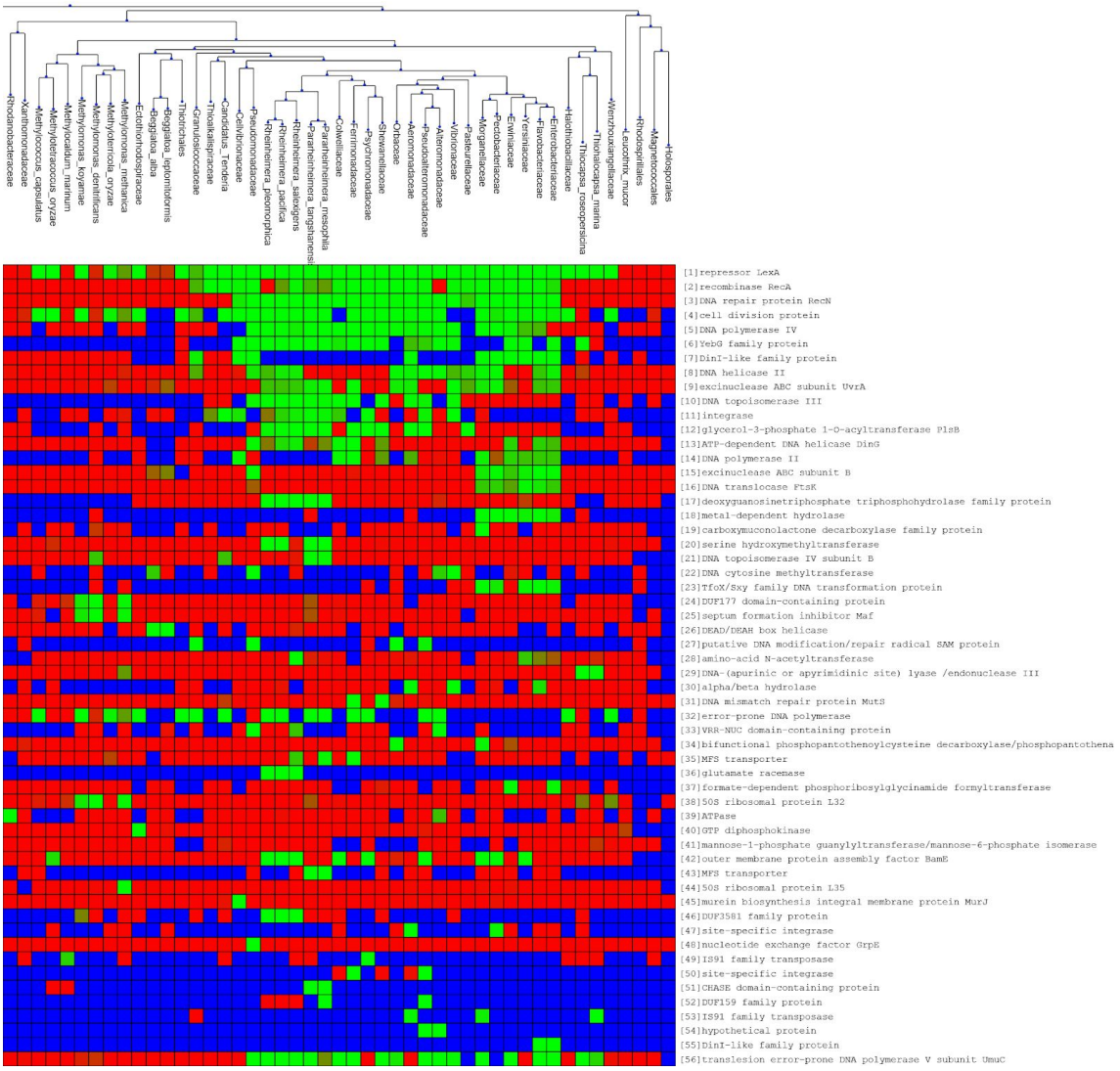


***Figure 7**: Fragment of the heatmap_light obtained from the Gammaproteobacteria motif, showing Gammaproteobacteria and their closest relatives only the first 56 orthologous groups are displayed. The plot displays the branch containing Gammaproteobacteria.*

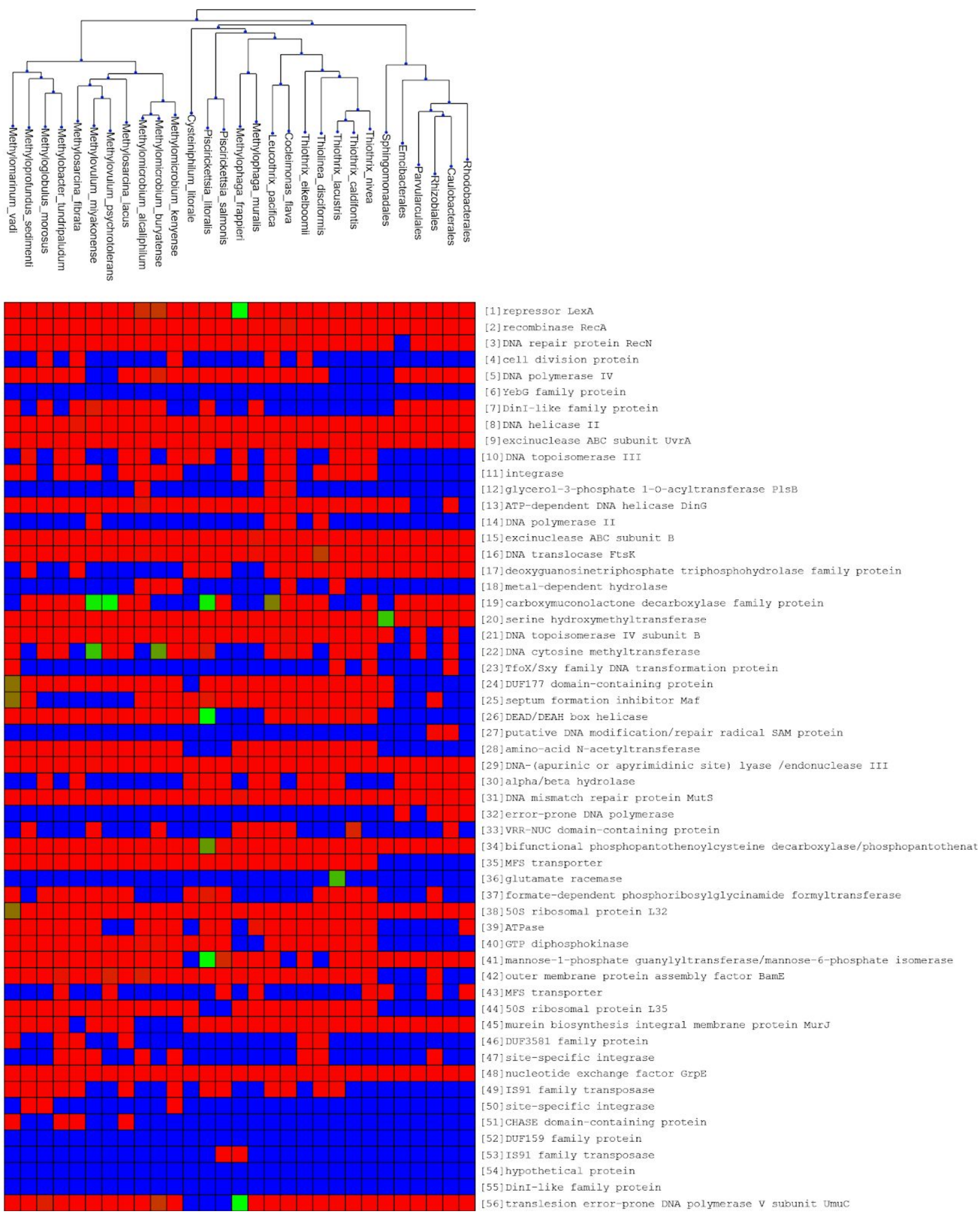


***Figure 8**: Fragment of the heatmap_light obtained from the Gammaproteobacteria motif, showing Gammaproteobacteria and their closest relatives only the first 56 orthologous groups are displayed. The plot displays the branch containing Alphaproteobacteria.*

Alphaproteobacteria motif:

The plot belonging to the Alphaproteobacteria motif presents the most interesting information. Two fragments of the plot are shown below, the first one showcasing the selected Alphaproteobacteria representatives and their closest relatives, and the second one targeting the closest relatives to Gammaproteobacteria.

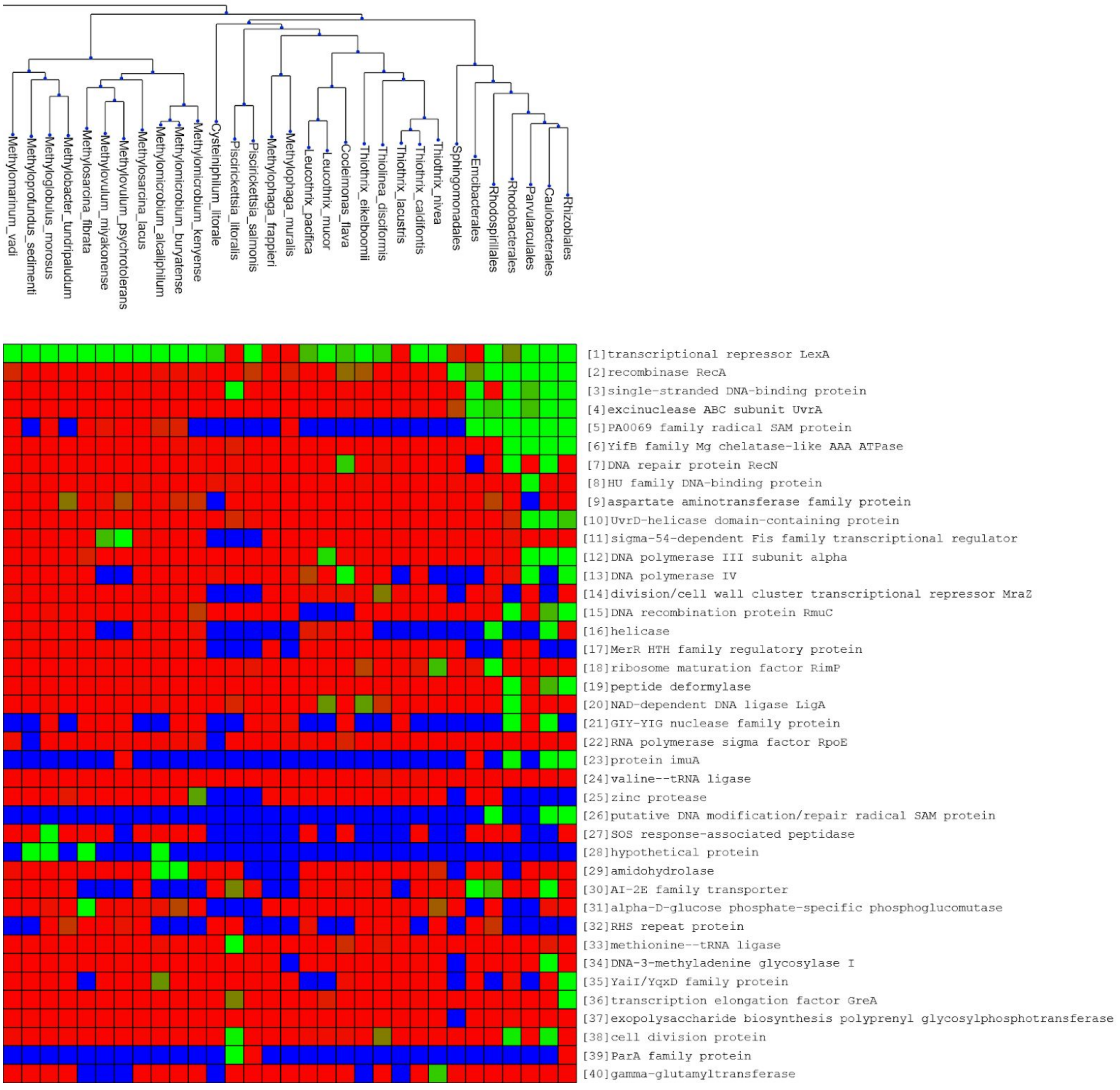


***Figure 9**: Fragment of the heatmap_light obtained from the Alphaproteobacteria motif, only the first 40 orthologous groups are displayed. The plot displays the branch containing Alphaproteobacteria.*

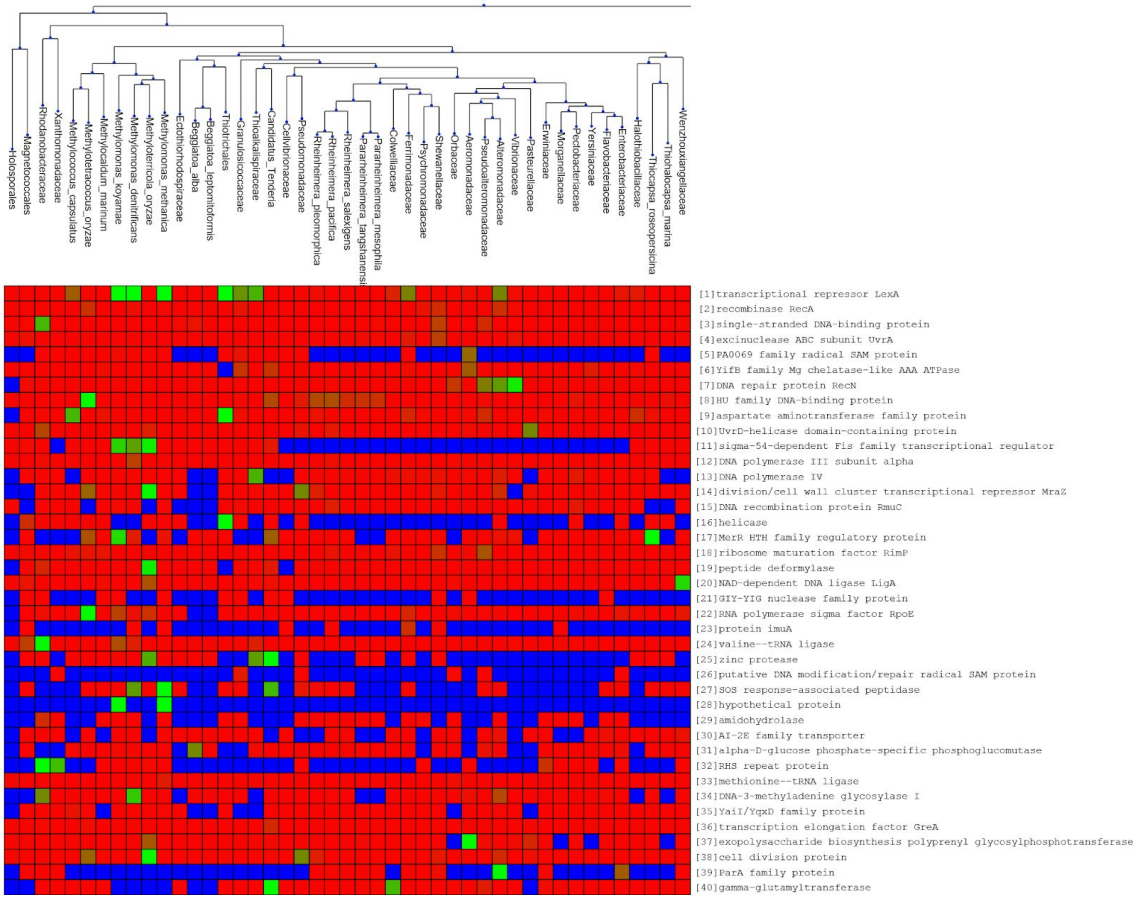


***Figure 10**: Fragment of the heatmap_light obtained from the Alphaproteobacteria motif, only the first 40 orthologous groups are displayed. The plot displays the branch containing Gammaproteobacteria.*

As it can be observed, there's a group of Gammaproteobacteria genera whose LexA sequence is closely related to that of Alphaproteobacteria, and most of those genera seemingly target the Alphaproteobacteria motif, while not displaying regulation for the Gammaproteobacteria motif (with the exception of *Methylophaga frappieri*).

Regulon size (the total number of homologous groups presenting a high likelihood of binding for a species) also presents interesting results, as while Gammaproteobacteria-recognizing species seem to show similar regulon size to the reference Gammaproteobacteria genera, this is not the case for Gammaproteobacteria species which recognize the Alphaproteobacteria motif, whose Regulon is usually limited to only LexA itself, unlike the true Alphaproteobacteria members, which have a much longer regulon. However, some species, such as *Methyloterricola oryzae* present fully fledged Operons, albeit such events are not shared with any other species.

3.1.2: Group-by-group discussion:

Methylococcales:

-*Methylococcaceae* has been split in two clearly separated groups, one group showing similarity to the LexA obtained in Alphaproteobacteria representatives (genera *Methylomarinum, Methyloprofundus, Methyloglobulus, Methylobacter, Methylosarcina, Methylomicrobium*) which shows clear evidence of recognizing almost exclusively the Alphaproteobacteria motif, whereas the second group has been placed as closer relatives to the rest of Gammaproteobacteria representatives (featuring the genera *Methylococcus, Methylomonas, Methylocaldum, Methyloterricola* and *Methylotetracoccus*), this groups presents evidence of binding to the Gammaproteobacteria motif, the Alphaproteobacteria motif, both motifs (evidence of LexA duplication, with each copy targeting a different motif) or neither

-*Methylocaldum marinum* presents no evidence for recognizing either motif throughout the operon. No bibliography attributes to *M. marinum* a parasitic lifestyle that could justify the loss of the SOS response, and a targeted BLAST search for *M. marinum* (using the LexA sequence from *Methylomonas koyamae* as the input) yields a highly significant ortholog, already marked as LexA.

Thiotrichales:

The *Beggiatoa* genus (part of the *Thiotrichaceae family*) has been found to ignore both expected motifs, albeit weak evidence is presented for the Gammaproteobacteria motif, which may hint towards a degenerated motif. It should be noted that the two *Beggiatoa* entries correspond to the only two BLAST results for the genus, and both show high likelihood of homology.

However, members of the *Leucothrix, Cocleimona, Thiolinea* and *Thiothrix* have been found to obey exclusively the Alphaproteobacteria motif, while ignoring the Gammaproteobacteria motif. With one exception, *Thiothrix lacustris,* which doesn't follow any of the two motifs. Again, BLAST shows that *Thiothrix lacustris* features highly significant LexA homologs in it's genome.

Grouped with the rest of *Thiotrichaceae* is the type sequence from *Thiotrichales*, however, it's source is a *Thiotrichales bacterium* that has yet to be properly classified. This unclassified genome responds to both motifs, thus, there's strong evidence of LexA duplication in this species, albeit no further information can be drawn due to its unclassified state.

Thiotrichales also includes the family *Piscirickettsiaceae* with two representative genera, *Piscirickettsia* and *Methylophaga. Piscirickettsia salmonis* follows the trend of recognizing the Alphaproteobacteria motif while not responding to the Gammaproteobacteria motif, whereas *P. litoralis* does not respond to either motif for the LexA transcription factor (however, it shows affinity for the Alphaproteobacteria motif for the recognition of ssDNA). While BLAST returns a LexA alignment for *P. litoralis*, it's worth noting that the single result has neither

the identity nor the e-score of the other cases drops from ~60% identity to 30%, and e-value <10^-30 -> >10^-20. *Methylophaga frappieri* obeys the Gammaproteobacteria motif and *Methylophaga muralis* follows neither motif, while not showing affinity to any SOS-related protein (BLAST shows evidence for the existence of a LexA protein in the species).

Last but not least, *Cysteiniphilum litorale* is the sole representative of *Fastidiosibacteraceae*, and it follows the Alphaproteobacteria motif

Chromatiales (purple sulfur bacteria)*:*

The *Ectothiorhodospiraceae, Chromatiaceae* (including *Rheinheimera, Pararheinheimera, Thiohalocapsa* and *Thiocapsa* genera), *Halothiobacillaceae, Wenzhouxiangellaceae* all follow exclusively the Gammaproteobacteria motif, while fully ignoring the Alphaproteobacteria motif

The *Granulosicoccaceae* and *Thioalkalispiraceae* families present evidence of binding to both motifs, however, the probability of binding is weaker than other similar samples, so it can not be stated that there has been LexA duplication.

Others:

Holosporales and *Magnetococcales*, two orders from Alphaproteobacteria, do not present evidence of regulation from neither motif. *Magnetococcales* has been studied and it is known that it's LexA binds to a third, different motif.

Something to be noted, has been the inclusion of *Flavobacteriaceae* in the study, this family does not belong to neither class of interest, and it's inclusion into the study has been an oversight (it belongs to Flavobacteria, not Proteobacteria). However, it presents evidence of binding towards the Gammaproteobacteria motif, while not responding to the Alphaproteobacteria motif.

Summary:

There is conclusive evidence for LexA duplication throughout the studied taxa, these events of duplication seemingly have arisen at random points throughout evolution, due to how widespread they are, and there is further experimental evidence of LexA duplication in certain *Xanthomonadaceae* and *Pseudomonadaceae* species[2]. This evidence suggests that events of LexA duplication are not restricted to the above genera and may be more widely distributed than previously thought.

There is also a recurring pattern of certain organisms not displaying high regulation likelihood for either motif, a pattern that is only matched by the selected genomes for *Magnetococcales* and *Holosporales*, two basal Alphaproteobacteria taxa, and in the event of *Magnetococcus marinus* (chosen representative for *Magnetococcales*) it is known that LexA recognizes an alternative, different motif, not matched by Alphaproteobacteria nor

Gammaproteobacteria, albeit the lack of resolution in these taxa make this a bold assumption. Such events are common throughout the phylogeny, and may just be instances of motif degeneration, where the recognized motif has suffered changes alongside it's binding motifs.

Also, while there are some recurring patterns of intracellular parasitism, anaerobicity and endosymbiosis between the taxa of interest, there's no clearly defined patterns in their life cycles that could be reasonably linked to the distribution of LexA motifs.

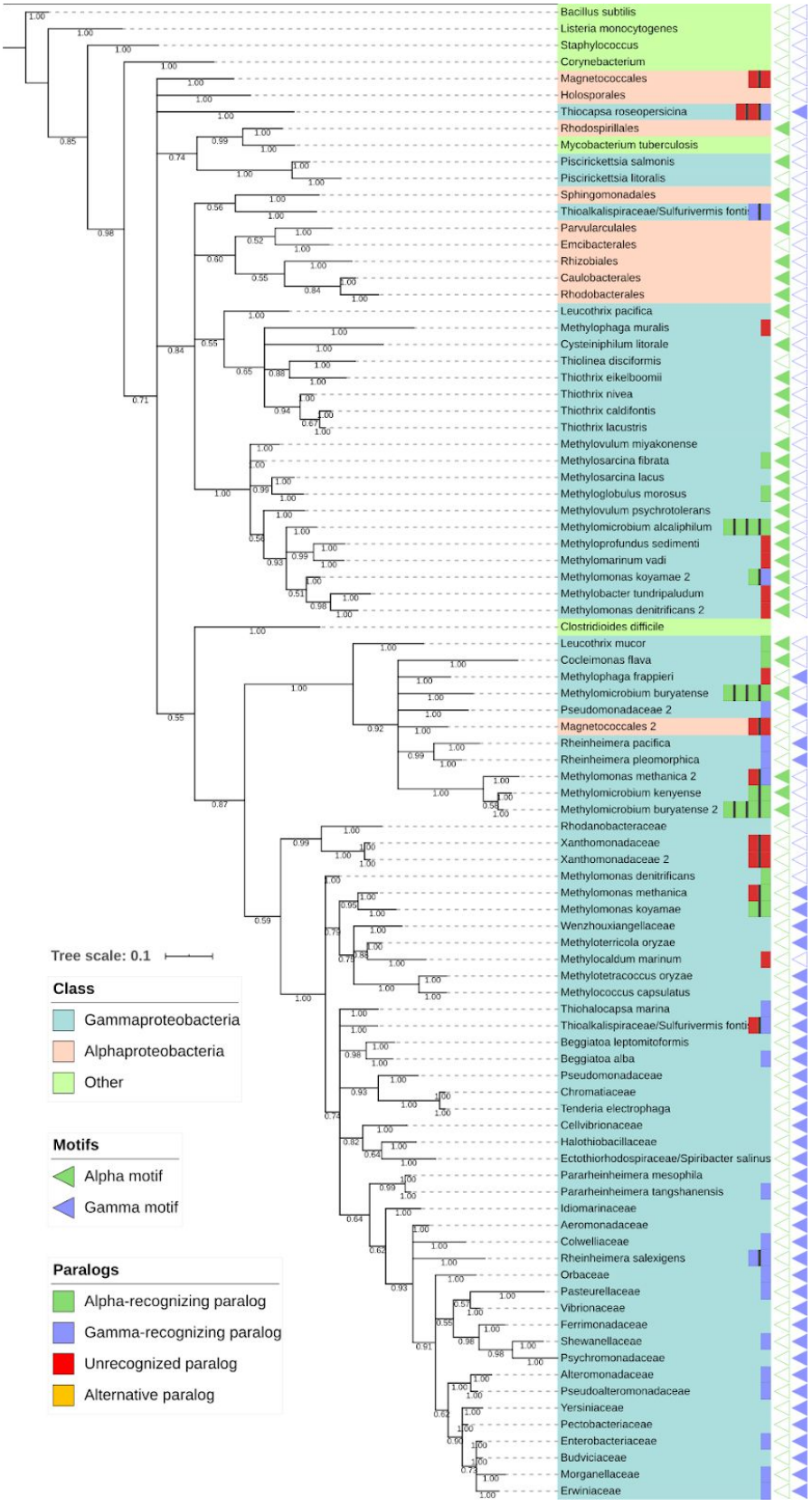
Mapping motif/regulon data on reference phylogenies

In order to evaluate the distribution of the motif between species, to identify patterns that could lead towards reconstructing the history of these motifs, all studied species (defined by all the genomes provided to CGB as an input) have been mapped against the phylogenetic trees defined in 2.1.5, and 2.1.6. Additionally, in the event of suspected LexA duplication (evidenced by the presence of different LexA genes in the orthologs.csv files belonging to both motifs), both copies will be included, with one of them having a "_2" suffix added to its name to signify duplication.

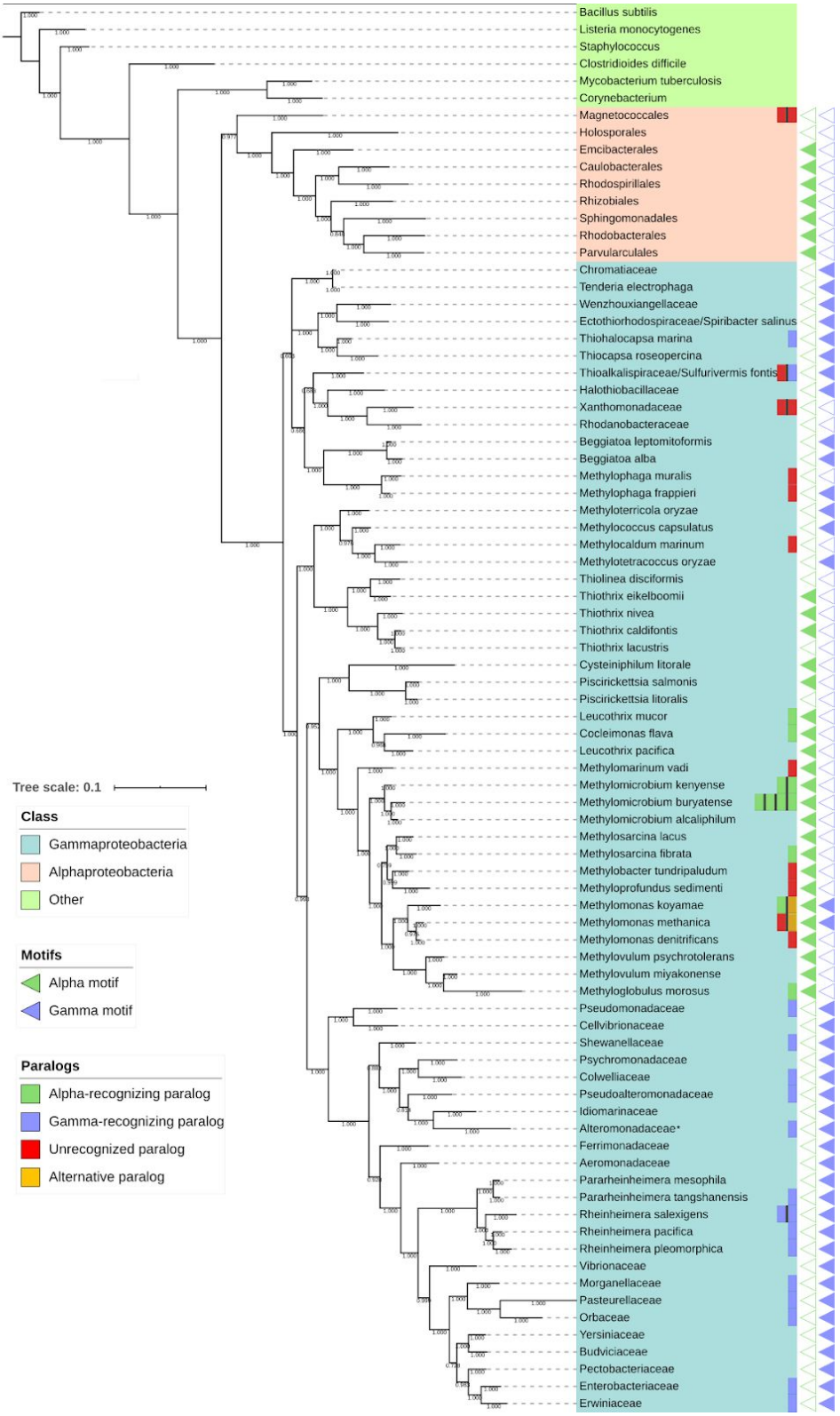
Before the process of Multiple Sequence Alignment, validated data from 6 additional non-Alphaproteobacteria and non-Gammaproteobacteria species (annotated with the color green) have been added to serve as a frame of reference for interpreting the tree, since due to the use of MrBayes for the purpose of performing Bayesian Inference, the resulting tree is not rooted. In order to improve the visualization of the tree, it has been rooted to *Bacillus subtilis*, and it should be expected for all the new species to be assigned to their own branch. These 6 species will not be annotated, since inside the scope of this project, it is not known what LexA motif they feature.

For the graphic displayed in Phylogeny 3, the 'main' copy of LexA has been annotated with its recognized motif with left-facing arrows. A hollow arrow indicates that no evidence of binding to the motif has been found, a full arrow indicates evidence of binding. For the purpose of defining 'evidence of binding' a threshold of a probability > 0.9 has been set to try to guarantee the rejection of false positives.

Additionally, potential paralogous genes to LexA have been annotated as colored rectangles to the left of the motif recognition annotation, a Red rectangle indicates that the paralogous gene does not react to either motif, a Blue/Green rectangle indicates that the paralogous gene reacts to the Gammaproteobacteria motif or to the Alphaproteobacteria motif, respectively.



Phylogeny 3: *Phylogenetic tree for the LexA protein, with motifs annotated for species of interest. The average standard deviation of split frequencies is at 0.014 after 1500000 generations. A hollow triangle represents lack of regulation, no annotation indicates that the species has not been studied, and is only included as a reference point.*



Phylogeny 4: *Phylogenetic tree for the 16S rRNA housekeeping sequence, with motifs annotated for species of interest. The average standard deviation of split frequencies has been left at 0.05 after 1500000 generations, this tree presents high uncertainty when it comes to the classification of Gammaproteobacteria outside the species of interest.*

Due to the fact that Phylogeny 4 groups together species with duplication, where each copy recognizes a different motif, a new, golden-colored marker has been introduced to mark a wildcard paralog. In such a case, consider the golden paralog to obey the motif opposite of the main LexA gene (represented by the left-facing arrows)

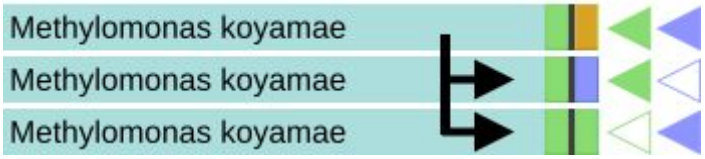


Figure 11: *Reading the wildcard paralog, the golden-colored paralog can be interchangeably interpreted in any of the displayed ways*

Furthermore, in Phylogeny 4, there's a very clear separation in the motifs recognized by various groups of Gammaproteobacteria, which can all be classified in three separate groups, two of which present binding exclusively to the Gammaproteobacteria motif (as would be expected) while the third group presents binding for the Alphaproteobacteria motif, with a few species in the latter recognizing both motifs, the implications of such a separation have been used to build two hypotheses, both detailed in section 3.3, however, here it will be noted that this group of Gammaproteobacteria which present positive binding towards the Alphaproteobacteria motif all present a tiny regulon (see Figure 9), which in most cases is limited to only LexA. This event could explain how this group presents so many events of duplication/deletion or degeneration between the motifs recognized by LexA, the tiny regulon would imply that a different transcriptional factor may have taken over the SOS response, leaving LexA obsolete, and allowing LexA to mutate without disturbing such an essential response (a similar case has been observed, where LexA lost it's functionality as activator of the SOS response in *Streptococcus thermophilus*[13])

One last comment in this section relates to *Alteromonadaceae,* marked with an asterisk (*) in Phylogeny 4, the gene identified as a LexA ortholog, which presents positive binding to the Gammaproteobacteria motif, does not contain an associated protein ID. Since there's a region of the genome associated to this ortholog, and there is a gene in this region, we may be looking at a pseudogene, a functional gene which is not expressed.

3.2: Hypothesis

It should be noted that evolution is not an exact science, and it's unlikely that the real evolutionary path can be neatly explained by a single event, so it's possible that both presented hypotheses have played a role in shaping the evolution of proteobacteria.

Both hypotheses are aimed at explaining how a group of Gammaproteobacteria ended up binding to the motif associated with Alphaproteobacteria. In order to aid the process of drawing hypotheses, it has been assumed that the smaller regulon associated with the Alphaproteobacteria motif is due to the higher uncertainty associated with having two different copies of LexA at some point in the past.

3.2.1: Ancestral State hypothesis

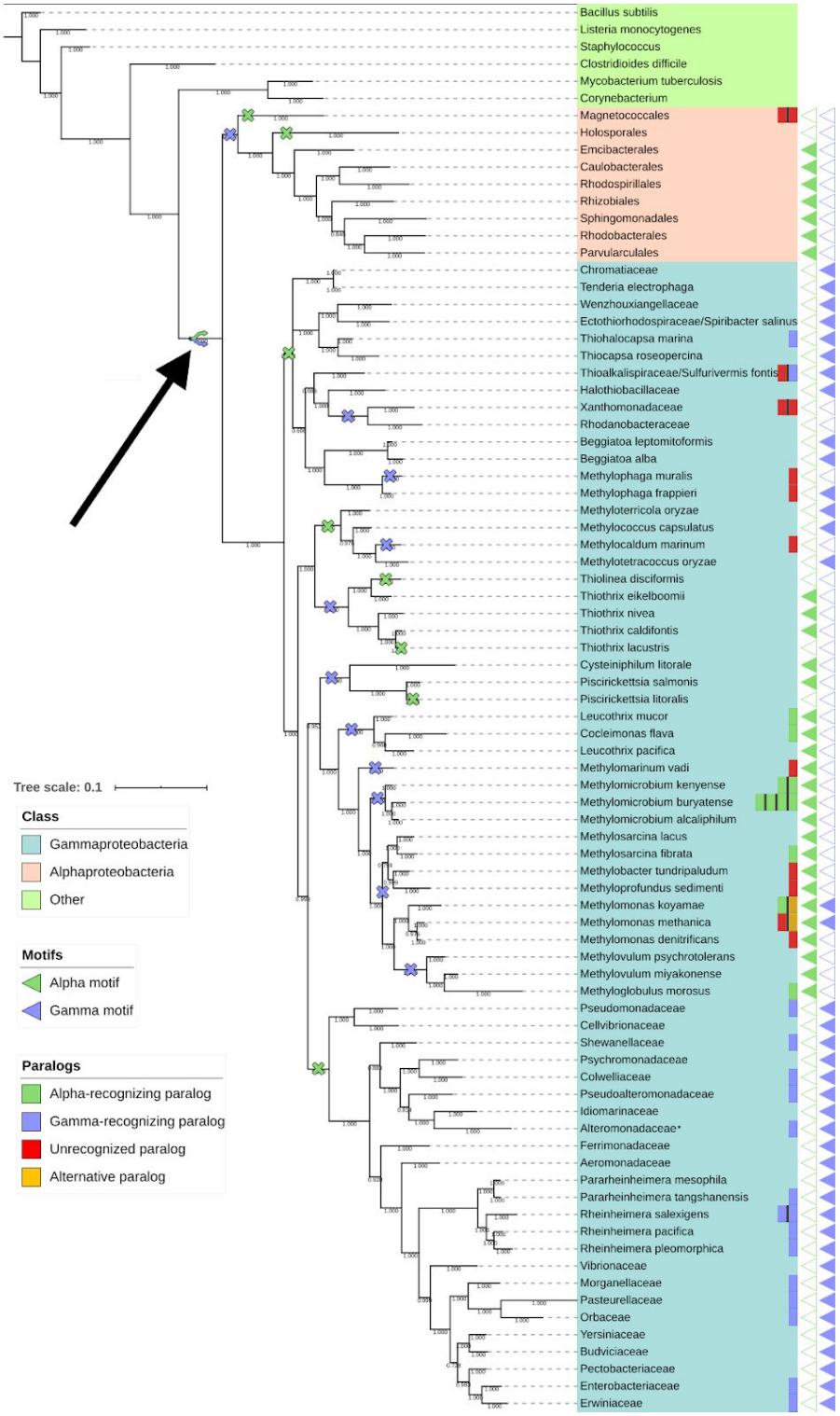
The main hypothesis presented is the ancestral hypothesis, where the last common ancestor between Alphaproteobacteria and Gammaproteobacteria suffered one (or multiple) events of LexA duplication, and would eventually radiate into the Alphaproteobacteria and Gammaproteobacteria of today.

LexA, unlike most other transcription factors, has been observed to change substantially throughout evolution, to the point that it's motifs have been thought to be monophyletic. This would normally be a strange event, as changes in the motifs recognized by a transcription factor would temporarily alter it's regulon, which would lead to cell death. However, this can be explained by events of LexA duplication (such as in a mutagenesis cassette), as this allows one of the LexA copies to mutate without altering the regulation of the SOS response, as only a single functional copy of LexA is required, this LexA copy would eventually mutate and recognize what would eventually become the CTGT-n8-ACAG associated with Gammaproteobacteria.

This hypothesis would explain why the LexA phylogeny, Phylogeny 3, presents a clear split in two groups of Gammaproteobacteria, one whose LexA sequence resembles that of Alphaproteobacteria, and another group whose LexA sequence closely matches the remaining Gammaproteobacteria, while the 16S rRNA phylogeny, Phylogeny 4, places both groups together.

Species featuring LexA duplication, where both paralogs present affinity for different motifs, provide strong evidence for this hypothesis, such as *Methylomonas koyamae* and *Methylomonas denitrificans* both indicate that one of their LexA copies has been phylogenetically placed closer to Alphaproteobacteria, whereas another copy has been closely linked to the rest of Gammaproteobacteria. Furthermore, the type species for *Magnetococcales, Magnetococcus marinus*, presents this same split, with one copy being placed close to Gammaproteobacteria, and another close to Alphaproteobacteria, however, *Magnetococcus marinus* recognizes a different motif.

There's also lots of evidence of LexA duplication (evidenced by the widespread distribution of paralogous genes) throughout all groups studied in the phylogeny, which is a possible indicator that LexA duplication may be an ancestral trait, originating from before Alphaproteobacteria and Gammaproteobacteria diverged from each other.



Phylogeny 5: *Phylogenetic tree for the 16S rRNA housekeeping sequence, with motifs annotated for species of interest. Additionally, the suggested event of ancestral LexA duplication has been annotated with the arrow, with scattered events of LexA motif degeneration/deletion annotated as colored crosses.*

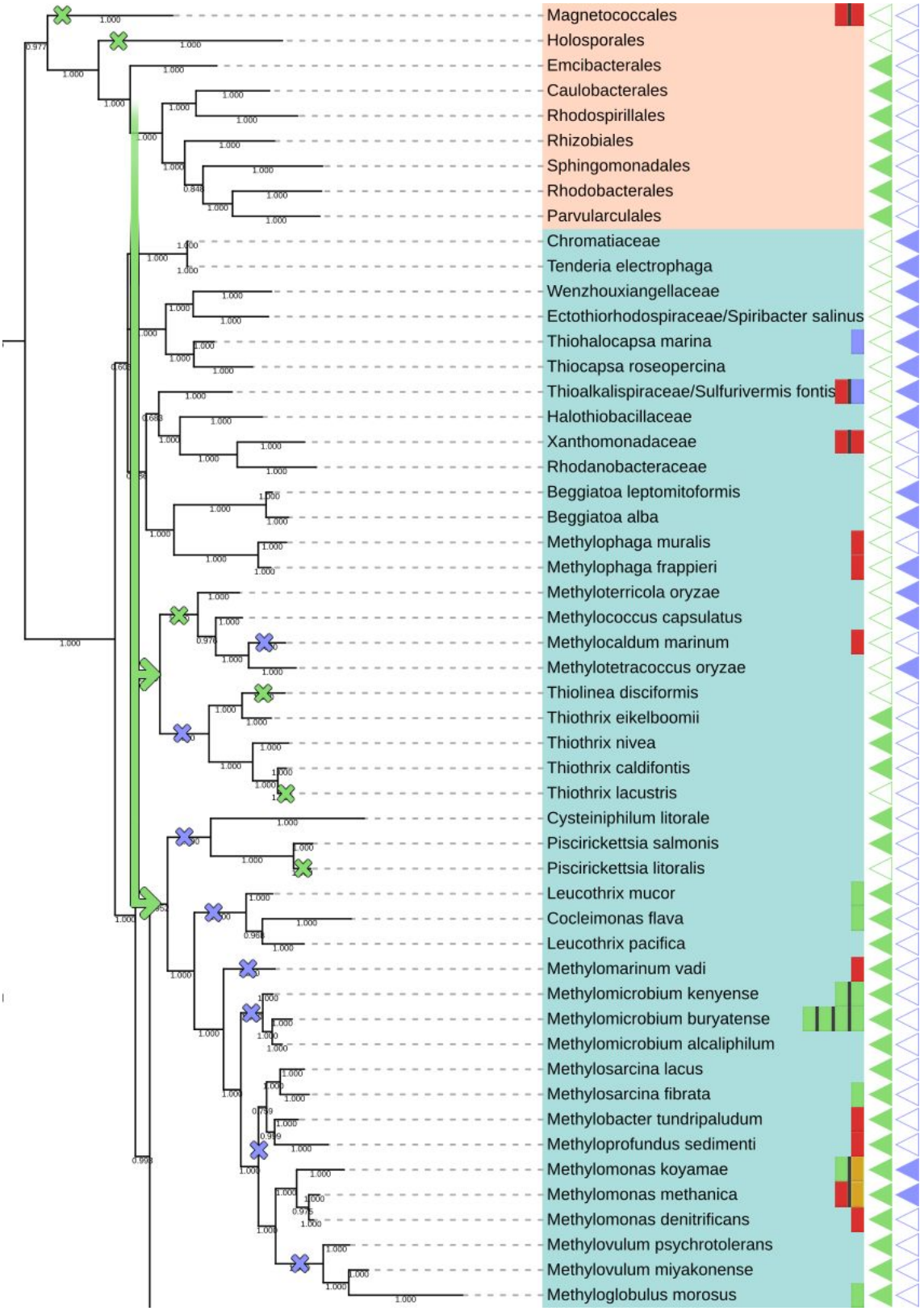
3.2.2: Horizontal gene transfer hypothesis

An alternative hypothesis would involve horizontal gene transfer, where Gammaproteobacteria and Alphaproteobacteria both diverged into their own, unique LexA genes from one single shared ancestral LexA, eventually recognizing the two studied motifs in Gammaproteobacteria and Alphaproteobacteria

In this hypothesis, the last common ancestor of *Methyloccocales, Chromatiales* and *Thiotrichales* would have received an additional copy of LexA from an unknown member of Alphaproteobacteria, and this new LexA copy would either take over the regulation of the SOS response, coexist as a duplicate of the original Gammaproteobacteria LexA, get jettisoned through gene deletion or degenerate into a different motif.

Thus, Gammaproteobacteria where the horizontally transferred LexA had taken over the regulon or coexisted with the original motif, would respond positively to the Alphaproteobacteria motif, despite being phylogenetically close to Gammaproteobacteria.

This hypothesis can be justified with two events of horizontal gene transfer, and a series of events of deletion/degeneration, with a model proposed in Phylogeny 6:



Phylogeny 6: *Snapshot of the phylogenetic tree for the 16S rRNA housekeeping sequence, with motifs annotated for species of interest. Additionally, a suggested event of horizontal gene transfer of the Alphaproteobacteria LexA has been represented with an arrow, whereas events of LexA motif degeneration/deletion have been annotated as colored crosses.*

3.3: Future and improvements

3.3.1: Personal limitations and self-critique

One of the key hindrances of this project has been my own personal lack of familiarity when it comes to the study of taxonomy, in that regard, this project has started from scratch and has only been made possible due to the guidance of Dr. Ivan Erill. This has presented a significant delay in the generation of results and has led to having to discard many early attempts and approaches, which is not something that can be properly reflected into this final thesis.

3.3.2: Software

This project has been hindered by the choice of operating system. Due to using a Windows operating system, CGB has not been compatible with it (an attempt to port CGB to Windows has been made, but it has been unsuccessful), the problem has been further exacerbated by the introduction of a Virtual Machine in the pipeline, as the performance of the Virtual Machine has been suffered by having to share the resources of a limited laptop, which has hurt performance. This has been solved by leaving the long computations to compute overnight, but that's not an ideal solution. No bottleneck has been defined, but it's likely that a more powerful computer may be able to reasonably compute more extensive data in a reasonable timespan.

Originally, the inclusion of the Virtual Machine into the pipeline was dismissed due to computer resource scarcity, so most of the code has been written and run in Windows, but the entire project could be reasonably migrated into a Linux environment with minimal changes, which would allow the abstraction of key parts of the process into an offline environment. Furthermore, both T-Coffee and GBlocks distribute pre-compiled binaries, but T-Coffee does not provide precompiled binaries for Windows. Since T-Coffee currently leads to a breach in the pipeline, so the GBlocks server has been kept in use. CGB could potentially be attached to the script at the end of the pipeline.

Additionally, the process features multiple time sinks, steps in the process whose computation takes a long time, they have been the following:

- BLAST: project has been running BLAST from Biopython, and results have taken between a few minutes to an hour (contrary to manually BLASTing from the NCBI page), a guess is that NCBI throttles some BLAST queries, as the speed of BLAST has been inconsistent, in retrospect, this could have been solved by downloading the "nr" database (the most preeminent database employed in this project) and running BLAST locally. This realization only came once BLAST+ was installed in the Virtual Machine, by which time it was too late to make a meaningful impact, as most lengthy BLAST alignments were already cached.

- Ortholog manipulation: During steps 2.3 and 2.4, large libraries of orthologs have been handled, during the last run of the process, this value is 5000 unique orthologs detected, each of which has an associated number of genomes, and each of which requires multiple ENTREZ queries to properly classify its taxonomy, its quality and its source. The problem arises once the ENTREZ guidelines are considered, there is a maximum of 3 queries per second, or 10 queries per second, if the user provides a valid key. These numbers add up, putting this step at 6-8 hours required. Steps to mitigated the length of this step had been taken, like requesting an API key to reduce the cooldown between queries to 0.15 seconds (due to inconsistent performance, a choice has been made not to reduce this value to the supposed minimum of 0.1), limiting the number of BLAST hits requested, to reduce the volume of source data (since the taxa of interest have been included through their own targeted BLAST results) and catching errors in all ENTREZ queries to prevent the entire process from being lost if the local internet connection momentarily fails. Additionally, once each step has finished, the results have been cached locally so as to avoid the computation time.

- CGB: The first step performed by CGB is downloading all the input genomes (defined as genome accession IDs) to a local cache, with which to create a local BLAST database in which the derived motif will be searched for, understandably, this is a slow process, but it pales in comparison to the rest of CGB. CGB has been observed to be computationally expensive, since it searches in all the genomes potential orthologous genes, paralogous genes, and looks into their downstream region looking for evidence of binding for the derived motif through the use of position specific weight matrices. Directly altering CGB has not been done (while possible, it may prevent the reproducibility of the data) so the actions taken have been to allocate most available memory to the Virtual Environment, and leaving the computation to run overnight. This last step has been benchmarked at 6-8 hours (and requires to be run twice to yield the desired results).

As a last mention, a display bug in CGB itself has prevented the correct assignment of accession IDs to paralogous genes. While it has been possible to circumvent this issue by manually cross referencing the rest of the paralogous gene data (which does not suffer from this issue), the manual process is slow and tedious. As of presenting this thesis, this issue has been fixed.

3.3.3: The results

A glaring flaw that has been noticed too late, has been the inclusion of the genera *Flavobacteriaceae*, which has wormed its way into the study, despite being neither an Alphaproteobacteria nor a Gammaproteobacteria. This issue has arisen from the process of determining orthologous proteins, as a bug in the algorithm made entries that should be discarded due to their Class be featured as families instead. All other offenders were quickly identified and removed from

the study, but *Flavobacteriaceae* avoided detection due to high homology with *Enterobacteriaceae* its way. However, it should be noted that it's inclusion, while not being enough to be representative, has provided evidence that certain species outside of Proteobacteria may feature the same motif as Gammaproteobacteria. Since the inclusion of *Flavobacteriaceae* has been a mistake, however, it has not been included in the final representations. It is also possible that such an entry has been mislabelled, so further exploration outside of Proteobacteria may be warranted. The inclusion of *Flavobacteriaceae* has not negatively affected the results, however.

Another issue worth noting is the process of building a phylogenetic tree, in Phylogeny 3, it can be observed that some branches are poorly supported (probability ~0.5), additionally, *Clostridium difficile* and *Mycobacterium tuberculosis* have been mixed with Gammaproteobacteria and Alphaproteobacteria, instead of being grouped close to the rest of reference LexA sequences, this is not correct, and represents evidence that conclusions drawn from Phylogeny 3 should be taken with a grain of salt. A possible explanation is in the use of GBlocks, as a number of highly conserved regions will be requested from a series of LexA sequences known to be unrelated to Alphaproteobacteria and Gammaproteobacteria. It should be possible to validate such an hypothesis by repeating the process of building a LexA phylogeny, while omitting the inclusion of GBlocks in the pipeline.

One last arguable choice is the process of selection of representative genomes, in this regard, the design of choice of 1 genome for 1 family is less than ideal, and its use has been a compromise, rather than a choice, by picking one single representative, all other members of the group will be excluded from the study. In this regard, one outlier genome could taint an entire taxonomic order, and while the results make sense, they can't be deemed representative with certainty, especially when it comes to groups composed exclusively by incomplete WGS records or direct submissions. A possible alternative approach would be selecting two or three genomes, from different species, to represent each group, just like the taxa of interest, but with a capped number of representatives.

3.3.4: Future

As per the future of this project, my personal suggestion would be to further expand the scope within Proteobacteria, by including more taxa which could provide further results to validate the presented hypotheses.

Xanthomonadaceae/Rhodobacteraceae/Cellvibrionaceae are the first candidates for inclusion, as they all appear to be closely related to the species of interest, considering that the 16S rRNA reference phylogeny has placed both those families between *Thiotrichales* and *Methylococcaceae*, both of which have been chosen as species of interest and both of which feature the Alphaproteobacteria motif and strong evidence of duplication. Such a study could further validate the results obtained in this thesis, albeit without truly expanding the scope, by determining if any members of these families may too

display such patterns. Do note that such an expansion should increase computation times.

On a similar note, the results associated with the second LexA copy of *Magnetococcales* are interesting, as that second copy has been placed in the same group as Gammaproteobacteria. It's hard to dismiss such a placement as randomness, since the branch is very strongly supported by Bayesian Inference, and while this second copy of LexA in *Magnetococcus marinus* does not recognize either motif, there's a chance that novel *Magnetococcales* species may provide further insight into the evolutionary pathways of LexA. If there's an Alphaproteobacteria that may prove the common ancestry hypothesis by presenting the motif associated with Gammaproteobacteria, it's most likely under the order *Magnetococcales.*

One of the procedures that has not been explored is ancestral reconstruction. While the provided trees feature annotations of paralogy and duplication, such annotations have been manually included in a time-consuming process, as it involves cross-referencing data in large datasets and annotating the results. Further hindered by the CGB bug in the labelling of paralogous genes. An ancestral reconstruction provides an automated, reproducible and scientifically validated way to provide data with which to infer the most likely evolutionary scenario that has resulted in such an event.

And last but not least, it's clear that there is a division in LexA sourced from Gammaproteobacteria species, but it is not clear why such genomes present such a tiny regulon, mostly featuring LexA as the only gene with positive regulation. A goal would be to further study the environment, life cycle and source of energy for these bacteria, and try to draw conclusions as to how they ended up with such a small regulon. It has been observed that methanotrophy is a pretty common trait in this abnormal group, but it's likely coincidence due to the inclusion of *Methylococcaceae*, rather than a source of causation.

3.3.5: Closing words

To close this thesis, this project has proven that there is a clear split in Gammaproteobacteria, if grouped by which motif LexA presents binding to, in that regard, all objectives have been fulfilled (besides automating the process, as manual input is still required). While these results do not represent a breakthrough, it provides a first glimpse into the transcriptional regulation in Proteobacteria, and provides evidence that a further study may be worth pursuing, and possible paths to explore when doing so.

Thank you for your attention.

4. Glossary

- **BLAST**: Basic Local Alignment Search Tool, algorithm and program for comparing primary biological sequence information, and identify potentially homologous sequences in a target genome.

- **CGB:** Comparative genomics of transcriptional regulation in Bacteria, a Python library for comparative genomics to analyze the operon of a transcription factor.

- **TF:** Transcription Factor, the name given to any protein with the capacity to regulate the transcription of a number of genes by changing the rate of transcription in these genes, usually, as a response to a stimuli

- **Motif**: A motif is a short, recurring DNA sequence, presumed to have an biological function by being recognized as a binding site.

- **Alphaproteobacteria motif:** The motif which LexA recognizes and binds to in order to repress the transcription of genes involved in the SOS pathway. This motif has been traditionally assigned to Alphaproteobacteria and follows a non-palindromic GTTC-n7-GTTC distribution, where n7 is a combination of 7 amino acids (usually, a sequence of Thymine and Adenine)

- **Gammaproteobacteria motif:** The motif which LexA recognizes and binds to in order to repress the transcription of genes involved in the SOS pathway. This motif has been traditionally assigned to Gammaproteobacteria and follows a palindromic CTGT-n8-ACAG distribution, where n8 is a combination of 8 amino acids (usually, a sequence of Thymine and Adenine)

- **WGS**: Whole Genome Shotgun, an approach to genome sequencing based on breaking multiple copies of a genome into fragments for the purpose of sequencing. The smaller fragments are easier to sequence and the Whole Genome can be reconstructed through homologous regions in the sequenced fragments. WGS may be available as multiple records (referenced through a Master Record)

5. Bibliography

1. Kidane, D. & Graumann, P. L. Intracellular protein and DNA dynamics in competent Bacillus subtilis cells. *Cell* **122**, 73–84 (2005).
2. Erill, I., Campoy, S. & Barbé, J. Aeons of distress: an evolutionary perspective on the bacterial SOS response. *FEMS Microbiol. Rev.* **31**, 637–656 (2007).
3. Erill, I., Campoy, S., Mazon, G. & Barbé, J. Dispersal and regulation of an adaptive mutagenesis cassette in the bacteria domain. *Nucleic Acids Res.* **34**, 66–77 (2006).
4. Cock, P. J. A. *et al.* Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25**, 1422–1423 (2009).
5. Huerta-Cepas, J., Serra, F. & Bork, P. ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data. *Mol. Biol. Evol.* **33**, 1635–1638 (2016).
6. Di Tommaso, P. *et al.* T-Coffee: a web server for the multiple sequence alignment of protein and RNA sequences using structural information and homology extension. *Nucleic Acids Res.* **39**, W13–7 (2011).
7. Armougom, F. *et al.* Expresso: automatic incorporation of structural information in multiple sequence alignments using 3D-Coffee. *Nucleic Acids Res.* **34**, W604–8 (2006).
8. Huelsenbeck, J. P. & Ronquist, F. MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics* vol. 17 754–755 (2001).
9. Ronquist, F. & Huelsenbeck, J. P. MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics* vol. 19 1572–1574 (2003).
10. Henikoff, S. & Henikoff, J. G. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. U. S. A.* **89**, 10915–10919 (1992).
11. Letunic, I. & Bork, P. Interactive Tree Of Life (iTOL) v4: recent updates and new developments. *Nucleic Acids Res.* **47**, W256–W259 (2019).
12. Yoon, S.-H. *et al.* Introducing EzBioCloud: a taxonomically united database of 16S rRNA gene sequences and whole-genome assemblies. *Int. J. Syst. Evol. Microbiol.* **67**, 1613–1617 (2017).
13. Boutry, C., Delplace, B., Clippe, A., Fontaine, L. & Hols, P. SOS response activation and competence development are antagonistic mechanisms in Streptococcus thermophilus. *J. Bacteriol.* **195**, 696–707 (2013).

6. Appendix:

In this section, a series of concepts worked on throughout the thesis are provided in detail. This information has been deemed to be too dense and/or uninformative to justify it's inclusion in the main body of the thesis, the files are made available as part of the attached "Appendix" folder, in .zip format, and are organized as follows:

6.1: Inputs folder

The inputs folder contains two files, 'input.json' and 'motifs.txt', 'input.json' is the starting point of the project, and is a .json file featuring the name of the transcription factor and the parameters to be used by CGB.

'motifs.txt' contains the recognized motifs by LexA, sourced from CollecTF, and provides the list of motifs that will eventually populate the 'motifs' list in the input file.

6.2: CGB_results folder

This folder contains all data associated with CGB, staring from the CGB input file (in format .json) to the split input files, to the complete results of CGB.

- CGB_input_full_data: Contains the resulting file from section 2.7, with the complete, unfiltered, unsplit list of selected genomes. Source species can be determined by cross-relating this file with the list of sources in section 2.4.

- CGB_input_alpha/gamma: These two files are the source of the CGB results, the only changes respect to CGB_input_full are the removal of redundant genomes, removal of uninformative genomes and the fact that motifs have been split, depending on the file (CGB_input_alpha only contains motifs from Alphaproteobacteria, while CGB_input_gamma only contains motifs from Gammaproteobacteria), this has been done to coerce CGB into deriving the known motif for each class, which serves as an additional form of validation.

- output_alpha/gamma: The full results from CGB, data of especial importance is described in the next segment.

6.3: results folder

This folder contains the most important results from CGB, it includes the two heat maps for the two motifs, and the associated orthologs.csv files from which the heat maps have been created (alongside additional information). It also includes full image versions of the annotated phylogenetic trees.

6.4: Python_scripts folder

This folder contains all the Python scripts that have been created during the development of this thesis, it should be noted that these scripts will not run in a vacuum, since they determine data relative to their path, so additional setup would be required to properly replicate the project. The scripts are also not the cleanest, and functionality has been prioritized over presentation.

phyloblast.py is the main script used, and all the processes detailed in section 2 is performed in there (outside of the multiple sequence alignment, the GBlocks treatment and the creation of phylogenetic trees)

The other scripts have been used as testing frameworks and/or to automate menial tasks, like determining the presence of Paralogous genes throughout all the sampled genomes. Some of these scripts have been derived from ‘phyloblast.py’ and/or have been integrated inside of ‘phyloblast.py’.