**Single Organism Context Analysis (CDMAP-SOA)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**CDMAP\_SingleOrganism\_Analysis.R:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Purpose**

This is the Main executable script for the CDMAP Package that runs the Single Organism Analysis, referred to as CDMAP-SOA. In this script, CDMAP first checks for, and if necessary, installs all required packages. Then CDMAP prompts the user for relevant input information such as the cleaned VCF file, GBK file, and reference FASTA file and then dynamically creates all the relevant child directories needed to store all output files. Then CDMAP proceeds to calculate the counts and rates for all 64 site-specific contexts, codon usage, both with respect to the leading and lagging strand.

*(User Note: CDMAP grants the user the option to use customized directories for where they wish to store their output if they so choose. For most users we recommend using the default options unless highly experienced with using CDMAP in an extensive manner.)*

**Required Packages**

These packages are the required tools currently to implement the CDMAP Single Organism Pipeline (CDMAP-SOP). Each run will ask you either if you need to install and/or update the required packages, and how you wish to install them.

*(User Note: If this is the first time you are running CDMAP, Installing and loading all the required packages and dependencies may take some time)*

* **Seqinr** – Multi-use bioinformatics sequence analysis package for parsing and searching FASTA files, contains **ORILOC** for ORI/TERM determination
* **BiocManager** – multi-use bioinformatics package containing **genbankr**, a package for parsing and manipulation of GBK files
* **Pracma –** Mathematics package for R containing a library of numerical analysis functions
* **Lattice** - Lightweight graphics visualization package used for generation of heatmaps
* **Beepr** – Notification package that sends audio notifications during certain steps of the analysis process.

**Matrix Object, Variables & Functions Legend:**

Listed below are a list of the various variables and data objects instantiated when running CDMAP-SOP.

* **Organism** – String variable containing the name of the organism being analyzed.
* **DirCheck** – String variable that acts as a flag parameter to check if you are working from home/work/ or as a guest (depreciated).
* **Pathwd** – Working directory in which the executable scripts are contained.
* **Path\_output** – File path of where the user desires output data generated**.**
* **Path\_RefFile** – Reference FASTA sequence file path.
* **Path\_GBFile** – Reference genbank GBK file path.
* **Path\_correlate\_repo** – Path of output generated multi-organism correlation repository.
* **generations** – Number of mutation accumulation generations carried out during the MA experiment
* **param\_flag** – Visualization output scaling flag.
* **RefFile** – Raw reference FASTA file parsed by seqinr
* **RefSeq\_arr** – Munged and wrangled version of **RefFile**, which has been converted to an uppercase character array.
* **len\_refseq –** Integer value for that references the length of **RefSeq\_arr.**
* **MutBaseCalls** – Data frame that holds the position (numeric), the reference (character), and mutant (character) nucleotide values.
* **ori\_ref** - Generates the oriloc data object, this is a test variable.
* **ori\_pos –** Numeric that finds the position of the origin of replication (in KB)
* **ori\_bp** – Converted version of **ori\_pos** that has been adjusted for relative nucleotide position length
* **ori\_value** – Skew value of the ORI position
* **term\_pos -** Numeric that finds the position of the terminus of replication (in KB)
* **term\_value –** Skew value of the term position
* **term\_bp -** Converted version of **term\_pos** that has been adjusted for relative nucleotide length
* **flag\_3mer** – flag to store the base nucleotide triplet information for diagnostic purposes during mass generation of mutation rates for coding regions, upstream, and downstream output.
* **Chisq\_Input –** output storage matrix that converts the nucleootide triplet single organism output into a tableau ready format for downstream analysis.

**Common Variables (used in multiple scripts)**

**cols/rows** – character object arrays used for storing column/row name objects to be applied to matrix objects

**flagcheck** – Boolean variable used in multiple scripts to check the truth value of a set of conditions (example: whether a gene is located on the left or right replichore)

**Bool1, Bool2, … -** a set of boolean checkers typically used to represent a true/false expression for a series of checks on a singular object.

**increment** – numeric object which accesses a specific entry in a matrix, increments the value by 1, and then updates the entry.

**comp\_increment** – numeric object which accesses a specific entry in a reverse compliment matrix, increments the value by 1, and then updates the entry.

**Comp –** flag to denote reverse compliment orientation i.e the lagging strand of replication direction for a given chromosome or replichore.

**LeftNuc** – the left neighbor nucleotide in a nucleotide triplet. This is the index-1 of the middle nucleotide.

**MiddleNuc –** the center nucleotide in a nucleotide triplet.

**RightNuc** – the right neighbor nucleotide in a nucleotide triplet. This is the index+1 of the middle nucleotide.

**rowiter/coliter/rowswitch/colswitch** – temporary variable object used for accession, incrementation, or assignment of objects pertaining to the determination of rows and column accession.

**GC\_counter** – counter object used to manually calculate genomic g+c nucleotide content in a chromosome.

**position** – storage numeric variable used to reference a nucleotide position currently being operated upon.

(Note: the **\*** symbol used in conjunction with variable, matrix, or script name is used to refer to all objects that belong to the set of objects and/or variables using that set of name flags. Example: Mut\_Matrix\_\* refers to all Mut\_Matrix Matrices.)

**Common Matrix Name Flags:**

**GC3C** – represents the base nucleotide triplet storage matrix that stores nucleotide triplets within a coding region.

**GC4C** – represents a GC4C matrix object that stores a nucleotide triplet in a coding region with respect to a given **up** or **down** stream nucleotide.

**A/C/G/T** – name flag representing an object with respect to a specific nucleotide (example: representing a given upstream or downstream nucleotide triplet)

**Left/Lcore/Right/Rcore** – represents whether the matrix contains Leading (right) or Lagging (left) replichore nucleotide triplet objects.

**Chrome** – name flag for matrix objects that track and store datasets with respect to a given chromosome.

**output** –name flag for character variable used for generating file output names

**up/upstream** – name flag for an object representing an object with respect to an upstream nucleotide object.

**down/downstream** – name flag for an object representing an object with respect to an upstream nucleotide object.

**path** – name flag for directory character variables

**Base/Mut\_Matrix** – represents a 16x12 matrix representation of mutation counts/rates with respect to the chromosome, each replichore, and replication direction.

**Sung** – represents the 16x4 matrix matrix representation of mutation counts/rates with respect to the chromosome, each replichore, and replication direction.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**createDirectories.R:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Purpose**

The primary objective of createDirectories is to generate the required ordered output directories dynamically for a given organisms triplet counts and context dependent mutation rates with respect to the leading and lagging strand. For reach directory, CDMAP checks if the directory exists, then proceeds to create it if it does not exist.

*(User note: these directories will be generated in either the default directories included with the program, or will be generated in the specified output directory by the end user in prior steps)*

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**generateContextCounts.R:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Purpose**

The objective of generateContextCounts is to generate the context dependent mutation counts for a given chromosome of a single organism. This file parses the MutBaseCalls data frame generated from a stripped down VCF file and then for each mutation generates the following information:

* + - Nucleotide Mutation 5-mer frame:
      * Nucleotide Triplet Mutation Frame: the 3mer comprised of the point mutation, and its left and rightmost nucleotide neighbors.
      * 4-mer Upstream Mutation Frame: the 4mer comprised of the point mutation, and its two left-side, and immediate right-side nucleotide neighbors.
      * 4-mer Downstream Mutation Frame: the 4mer comprised of the point mutation, and its immediate left-side, and two right-side nucleotide neighbors.
    - Spatiotemporal Mutation Distance:
      * Distance in which the point mutation occurred with respect to the ORI and TERM location.

\*\*insert important loops, control structures and variables here\*\*

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**partitionReplichores.R:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Purpose:**

The purpose of partitionReplichores is to take the output generated from generateContextCounts and split the context dependent mutation counts generated into the organisms left and right replichore for each mutation frame.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**GWTC.R and Rev\_GWTC.R:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

The following Set of scripts are used to generate the counts of the observed nucleotide triplet, and 4mer frequency of given nucleotide 3mers and 4mers throughout a given chromosome of an organism with respect to its given replichore:

* + - * **GWTC.R** sequentially calculates each nucleotide triplet (with upstream and downstream nucleotides) on the leading strand (5’ to 3’) and increments its corresponding entry into the requisite matrices for calculation in subsequent steps.
      * **Rev\_GWTC.R** operates in a similar manner to GWTC.r and sequentially counts the observed frequency of nucleotide triplets (with upstream and downstream nucleotides) on the lagging strand (3’ to 5’) and increments the corresponding matrices for calculation in subsequent steps.

**Matrix Object, Variables & Functions Legend:**

**GWTC** – Matrix object used to access and increment nucleotide triplets for the genome wide triplet count with respect to the leading strand.

**Rev\_GWTC** – Matrix object used to access and increment nucleotide triplets for the genome wide triplet count with respect to the lagging strand.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**GC4C.R:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

GC4C is an augmented version of the GWTC and Rev\_GWTC scripts. For a given chromosome, GC4C tabulates and sorts the nucleotide triplets in coding regions along with their respective upstream and downstream nucleotides for 4mer analysis. This is performed using the supplied Genbank (.gbk) file and for each gene coding region cataloged performs triplet tabulation on a per-gene basis. Due to Genbank files listing genes in both the forward (+) and the reverse (-) orientation, two helper scripts (**GC4C\_recorder\_Forward.r** and **GC4C\_recorder\_Reverse.r**) were implemented with an appropriate control structure to accurately annotate coding region triplets with respect to the leading and lagging strand.

**Child Scripts of GC4C.r**

* **GC4C\_recorder\_Forward.r** – child script that implements similar control structures to GWTC.r to determine, increment, and record Nucleotide triplets occurring on the leading strand.
* **GC4C\_recorder\_Reverse.r** – child script that implements similar control structures to GWTC.r to determine, increment, and record Nucleotide triplets occurring on the lagging strand.
* **CUB\_output.r** – child script responsible for text output of the codon usage with respect to each amino acid in a given chromosome.

**Matrix Object, Variables & Functions Legend:**

**upstream\_matrix** and **downstream\_matrix** – temporary object matrices that are used for accessing and incrementing upstream and downstream nucleotide matrices.

**geneFeatureMatrix** – output matrix where each row contains the start, end and length of a gene feature in a given gbk file for downstream processing.

**featstart** – this represents the start nucleotide of a gene feature object.

**featstartindex** – index array of all the featstart positions that occur within the gbk file.

**featend**  - represents the end nucleotide position of a gene feature object.

**featendindex** – index array of all the featend positions that occur within the gbk file.

**featlength** – length of the coding region feature.

**featlengthindex** – index array of all the featlength values that occur within the gbk file.

**featuretype** – storage variable used to determine if a feature is a ‘gene’ object

**dataobj/matobj** – matrices used for data coercion and storage of each gene feature in the gbk file.

**upNuc** – storage variable for the upstream nucleotide

**downNuc** – storage variable for the downstream nucleotide

**gene\_arr** – storage variable for the nucleotide character array for GC3C/GC4C analysis

**input\_matrix** – temporary storage object for output matrices to process output

**GC3C\_doesNotEqual** – function used at the end of the script to quality check and ensure that the sum of the set of each 4mer upstream and downstream matrix is equal to the sum of the number of triplets in the GC3C matrix object.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**4fold\_mutation\_Record.R:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

4fold\_mutation\_Record.r is the primary script where we take the output from generateContextCounts.r and tabluates each nucleotide triplet, and its respective 4mer upstream and downstream frames to calculate the context dependent mutation rates for all 64 nucleotide in subsequent steps.

**Matrix Object, Variables & Functions Legend:**

**codeRegion** – boolean checker object to detect if a mutation is within a coding region.

**Mut\_Matrix**  - a 16x12 matrix representing the nucleotide triplet matrix N[X → Y]N where N represents a nucleotide (A,T,C,G), and X and Y represent a nucleotide N where X ≠ Y and X mutates to a specific nucleotide Y. Each row represents all possible combinations of neighbor nucleotides N, and each column represents a specific X → Y nucleotide mutation.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Sung\_Matrix\_Left.r , Sung\_Matrix\_Right.r, and Sung\_Matrix\_Chromosome.r:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

The purpose of the Sung\_Matrix\_\*.r scripts are to concatenate the 16x12 Mut\_Matrix matrices into a standardized 16x4 **Sung\_Matrix** nucleotide matrix format N[X → Z]N where N represents a nucleotide (A,T,C,G), and X and Z represent a nucleotide N where X ≠ Z, and X mutates to any other nucleotide. Each row represents all possible combinations of neighbor nucleotides N, and each column represents a specific X → Y nucleotide mutation.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**MutationRate.R and RevCompliment\_MutationRate.R:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

The purpose of MutationRate.r and RevCompliment\_MutationRate.r is to compute the context dependent mutation rates for each nucleotide triple for all 64 possible triplet sites with respect to the chromosome, each replichore, and each direction of replication. This rate is calculated in the following manner:

Where represents the chromosome and represents the replichore specific base substitution rates for a given nucleotide triplet site for each generation. represents the total number of mutations observed at a given site. In the case of replichore specific mutation rates, only comprises of mutations accumulated on the specific replichore as opposed to the genomic total number of mutations. , and represent the site specific GWTC and RWTC for a given triplet. T represents the number of generations of replication for an organism-specific MA experiment, and K represents the number of separate mutation accumulation lines from an ancestral progenitor in a MA study.

**Matrix Object, Variables & Functions Legend:**

**MutationRatio\_\*** - A set of matrices that keeps track of the Mutation ratio of each nucleotide triplet site in an organism. The mutation ratio of a nucleotide triplet site is defined as

Where is the mutation ratio with respect to the chromosome and is the mutation ratio with respect to a specific replichore

**LeftRep & RightRep** – name flags to denote the calculation of

**GWTC\_count** – accessor storage variable used for grabbing the GWTC of a specific nucleotide triplet.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Context\_Data\_Visualization.R:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

The primary purpose for Context\_Data\_Visualization is to serve as the main script for processing visual output for the counts and context dependent mutation rates generated in prior steps. The user will be asked how they wish to scale the visual output for ease of viewing. The user has the option to scale by default parameters (listed below), the average mean of counts and rates, or to use custom scalars if the user desires.

**Matrix Object, Variables & Functions Legend:**

**mut\_count** – scalar for mutation count data heatmaps. Default is set to 1

**mutrate\_count** – scalar for mutation rate data heatmaps. Default is set to 1E-8.

**Lattice\_Visualizer.r**  is a child script of context data visualization that performs the generation and saving of high quality output images using input data.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Config\_Dump.R:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Config\_Dump is a small text output script that occurs at the end of the single organism pipeline analysis that dumps relevant information such as sequence length, replication origin, terminus, and number of mutations on each replichore in the chromosome.

**Multi-Organism Correlation Analysis (CDMAP-MOA)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**CDMAP\_MultiOrganism\_Analysis.R:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

This script is the main executable script for CDMAP’s Multi Organism Analysis (CDMAP-MOA). MOA’s primary function is performing the one-to-many context dependent mutation chromosome analysis of all 64 nucleotide triplet sites among organisms analyzed with CDMAP. The multi organism analysis script utilizes the correlation output directory generated by the single organism pipeline and indiscriminately analyzes all chromosomes analyzed with CDMAP with respect on a given user’s computer. This portion of the pipeline is flexible with respect to the number of organisms analyzed, as there is no upper limit to the number of comparisons you can make i.e. whether you have 5 or 500 chromosomes, the multi-organism analysis will scale accordingly.

(Developer note: In order to run this type of analysis, you must have run the single organism analysis on two or more organisms prior to this analysis.)

(Developer Note 2: Visualization constraints when looking at more than 30 chromosomes)

**Required Packages**

These packages are the required tools currently to implement the CDMAP Single Organism Pipeline (CDMAP-SOP). Each run will ask you either if you need to install and/or update the required packages, and how you wish to install them.

*(User Note: If this is the first time you are running CDMAP, Installing and loading all the required packages and dependencies may take some time)*

* **BiocManager** – multi-use bioinformatics package containing **genbankr**, a package for parsing and manipulation of GBK files
* **Pracma –** Mathematics package for R containing a library of numerical analysis functions
* **Lattice** - Lightweight graphics visualization package used for generation of heatmaps
* **Beepr** – Notification package that sends audio notifications during certain steps of the analysis process.

**Child Scripts:**

**GCcontent.r** – small helper script that parses the reference FASTA directory that computes the genomic GC content of each chromosome in the directory, then sorts them by GC content for downstream ordering of correlation cells in the output visualization.

**GC\_Analysis\_InitDir\*\*.r** – The primary function of these scripts are used to initialize the directories of a given nucleotide triplet configuration for analysis. Listed below are the flags needed to understand each script:

* **Triplet** – The first script initialized when analyzing the one-to-many correlations between organisms. This loads all of the base nucleotide triplet outputs and executes the one to many analysis in the subsequent scripts.
* **Up/Down** – Directory flag that corresponds to analyzing the upstream or downstream triplet with respect to a given nucleotide triplet.
* **A/C/T/G** – Directory flag that corresponds to the specific upstream or downstream nucleotide being analyzed.

**GC\_Analysis.r** – This helper script initializes and then directs CDMAP-MOA for a given nucleotide triplet configuration in the specified helper script above (GC\_analysis\_InitDir\*\*.r), directs the program to conduct CDMAP-MOA with respect to the each individual chromosome and replichore, in both 5-3’ and 3-5’ direction of replication. Below is a brief summary of name flags used for executing analysis:

* **Left/Right** – Used to represent analysis of the left or right replichore
* **Chromosome –** Represents the standard chromosome analysis with respect to the 5-3’ leading strand of replication.
* **Counterclockwise/RevComp** – Used to represent the analysis of a chromosome or a replichore with respect to the 3-5’ lagging strand of replication.
* **Rep/RepSpecific** – These keywords denote correlation analysis of a specific replichore (in either its 5-3’ or 3-5’ configuration) with respect to replichore specific mutation rates.

**Common Variables (used in multiple scripts):**

**MainDir** – The default installation directory path of CDMAP where primary executable scripts are stored

**LibDir** – The default installation directory path of CDMAP helper scripts is installed and accessed from

**UserOutput** – Variable used to dynamically generate an output directory path based on the user’s name

**Path\_to\_Fasta** – The default path of output reference fasta files used when analyzing a given organism using CDMAP-SOP

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Correlation\_Script.R:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

This is the primary backbone script of CDMAP-MOA that executes the one-to-many analysis for all chromosomes within a given directory. For each directory, MOA parses the directory and for each entry, dynamically loads in each output matrix, and indexes each matrix and the name of the corresponding organism. Next, CDMAP-MOA runs a verification check to ensure there are no null matrices for a given chromosome or replichore, i.e. a matrix consisting of zeroes. After verification is completed, each 16x4 matrix is correlated using Pearson’s product moment correlation. Upon calculation of each comparison, the Correlation coefficient, p-value, and the corresponding test statistic are stored for downstream visualization and inspection. In the final step, each entry is then manually sorted by MOA row-wise, and column-wise by genomic GC content for every comparison made within the directory and then initializes the child script for visualization.

**Child Script:**

**Correlation Visualizer** – This is a lightweight visualization script that uses Lattice to take the resulting one-to-many-analysis output matrix and generate a high quality heatmap for easy visualization of individual comparisons within the output matrix.