Protocols for Probing Genome Architecture of Regulatory Networks in Hydrocarbon and Lipid Microorganisms

Costas Bouyioukos, Mohamed Elati, and François Képès

Abstract

Genome architecture and the regulation of gene expression are expected to be interdependent. Understanding this interdependence is key to successful genome engineering. Evidence for nonrandom arrangement of genes along genomes, defined as the relative positioning of cofunctional or co-regulated genes, stems from two main approaches. Firstly, the analysis of contiguous genome segments across species has highlighted the conservation of gene order (synteny) along chromosome regions. Secondly, the study of long-range regularities along chromosomes of one given species has emphasised periodic positioning of microbial genes that are either co-regulated, evolutionarily correlated, or highly codon biased. Software tools to detect, visualise, systematically analyse and exploit gene position regularities along genomes can facilitate the studies of such nonrandom genome layouts and the inference of transcription factor binding sites and potentially guide rational genome design. Here, a computational protocol is demonstrated for the analysis and exploitation of regular patterns in a set of genomic features of interest (e.g. cofunctional or co-regulated genes, chromatin immunoprecipitation results, etc.). This case study is conducted for genes involved in hydrocarbon metabolism of a marine petroleum-degrading bacterium *Alcanivorax borkumensis*.

Keywords: Gene regulation, Genome architecture, Genome organisation, Periodicity detection, Prediction of TFBSs

1 Introduction

In trying to understand and engineer microorganisms, it proved rewarding to consider at once the threefold relation between chromosome spatial conformation, genome expression, and genome layout. Genome layout is defined here as the respective positioning of cofunctional genes. 'Cofunctional genes' refer to three, not mutually exclusive, possibilities: genes that encode proteins from the same complex or from the same metabolic pathway or genes that are co-regulated by the same regulatory factor. Indeed, individual gene transcription is modulated by sequence-specific transcription factors (TF). A TF binds to its binding site (TFBS) in the regulatory region of its target gene(s) to activate or repress its transcription. Short-range genomic similarities in the one-dimensional (1-D)

T.J. McGenity et al. (eds.), *Hydrocarbon and Lipid Microbiology Protocols*, Springer Protocols Handbooks,
DOI 10.1007/8623_2015_92, © Springer-Verlag Berlin Heidelberg 2015

positioning, known as synteny [1], reveal valuable information regarding the physiology of microorganisms. However, it has been demonstrated and is an active area of investigation that yet another level of three-dimensional (3-D) organisation of the genome is realised in terms of the long-range periodic arrangements of genomic features along genomes [2, 3]. Studies involving co-regulated [4, 5], cofunctional [6] and evolutionary correlated [7] genes have all identified sets of periodic patterns of the organisation of genes along microbial chromosomes. As these regularities in genome organisation can serve as a means for genomes to accommodate a series of physiological constraints [8, 9], the systematic detection, analysis and visualisation of such periodic patterns can elucidate regulatory mechanisms at the genomic level and provide insights for rational genome design in microorganisms.

Here, we demonstrate the use of a computational approach which detects and analyses patterns of regular organisation of the positions of genomic features of interest (e.g. genes). This approach is part of a more general schema of using modelling of genome architecture as a tool for studying and engineering regulation on a global – genomic – level. This general computational schema is called Genome REgulatory and Architecture Tools (GREAT) and is under development in our team at the institute of Systems and Synthetic Biology (iSSB). In this chapter we give a detailed account about how to use, interpret and exploit the results of the SCAN suite of tools which comprises all the analytical capacities of the GREAT schema.

The chapter is organised as follows: Section 2 provides a brief introduction to the materials and requirements to perform analyses with the GREAT:SCAN suite as well as a quick description of each tool. Section 3 is the main section where the details of each tool are delineated and the steps for a successful analysis are described further. Finally, Sect. 4 deals with troubleshooting and provides a guide for the values of the most significant parameters of the analysis.

2 Materials

GREAT:SCAN is a computational protocol for the integrated analysis of regular patterns in genomes. Its requirements are merely computational/software based. No previous programming experience is required to perform a complete GREAT:SCAN analysis, and no installation is required as, at the moment, GREAT:SCAN is available as a web tool. The required computations are performed by the calculators of the abSYNTH platform of synthetic biology at the institute of Systems and Synthetic Biology (iSSB, www.issb.genopole.fr). All the files generated during the execution (plots, tables and raw output files) can be downloaded by the user as a single zipped file. The web interface to perform the analysis can be found at the address https://absynth.issb.genopole.fr/Bioinformatics/ by selecting the icon for GREAT. The software accepts a range of optional parameters to control the analysis steps; all the parameters are implemented in the online interface of the tool, and a technical overview is presented in Appendix 2. Every user has unlimited and free access to perform analyses with the tools, with a single requirement to complete a very simple registration process at the above-mentioned internet address.

2.1 GREAT:SCAN: patterns

GREAT:SCAN:patterns is a tool written in R and is based on concepts and algorithms previously developed by the Képès team [2, 10]. The single requirement to perform the *pattern* analysis of the software suite is the format of the input file. Every input file of the analysis should contain two columns (any additional column will be ignored by the system). The first column should contain a unique identifier (e.g. a name) of the gene or the genomic feature of interest. The second column should contain the genomic coordinate (i.e. the position in the genome) of the gene or the genomic feature of interest. A single space is sufficient as a delimiter between the two columns although the system can accept any kind of conventional delimiter (tabs, semicolons). Appendix 1 contains an example of how the input file looks like. The source of the input data is totally arbitrary and is based on the motivation and the object of study of every researcher. GREAT:SCAN:patterns has been used to identify periodicities among co-regulated and co-evolved genes as well as among sites from ChIP-Seq or transcriptomics analyses. For this reason hereafter, when we describe a generic feature of GREAT:SCAN:patterns, we will be referring to the input as the 'set of genomic features of interest' (there be positions of co-regulated genes, ChIP-Seq peaks, transcriptomics peaks or any other set of interest as long as it obeys this simple rule of having a position in the genome and a unique identifier).

2.2 GREAT:SCAN: PreCisIon

GREAT:SCAN:PreCisIon is a tool written in R and based on concepts and algorithms previously developed by the team [11, 12].

PreCisIon is a general supervised method to infer new regulatory relationships between a known TF and genes in an organism. In its current form, it requires two types of data as inputs. Firstly, each gene in the organism must be characterised by some properties (views), here two views: its promoter sequence and its chromosomal position. While the former property has been used in all TFBS prediction studies so far, the latter has been developed by our team. The tool 'retrieve-seq' of the 'Regulatory Sequence Analysis Tools' http://rsat.ulb.ac.be/rsat/ [13] was used to retrieve upstream regulatory sequences ('promoters') defined here by the DNA sequence between position −400 and −1.

Secondly, for each TF, a list of its known target genes and, if possible, of its known nontargets is needed. Such lists can be constructed from publicly available databases of experimentally characterised regulations such as RegulonDB [14].

PreCisIon splits the problem of regulatory network inference into many binary classifications from disjoint views. For each view, PreCisIon trains a binary classifier to discriminate between genes known to be regulated and nonregulated by the TF. The final step is to combine all individual classifiers that have been trained on all (two here) disjoint views. All genes known to be regulated by this TF form a class of positive examples, and no prediction is needed for them. The remaining genes are split in three subsets of roughly equal size. In turn, each subset is taken apart, and PreCisIon is trained on all the positive examples, plus all genes in the two other subsets considered as negative examples. PreCisIon is then tested on the third subset, which has not been used during training. Rotating three times over the three subsets allows PreCisIon to attribute a prediction to each unlabelled gene by using an independent model.

2.3 Bacterial Genome

Alcanivorax borkumensis is a ubiquitous marine petroleum oil-degrading bacterium with an unusual physiology specialised for alkane metabolism. Its genome sequence and its genes involved in hydrocarbon metabolism were retrieved from the published freely available genome [15] through the UCSC Archaeal Genome Browser [16].

3 Methods

3.1 GREAT:SCAN: patterns

3.1.1 Periodicity Analysis

Every GREAT:SCAN study starts with a systematic and rigorous analysis and evaluation of all the periodic patterns that can be identified in the full genome of an organism. To this end, a pre-processing step is of paramount importance, the removal of proximity effects within the set of interest. Genomic features that are close to each other can 'contaminate' the calculation of probability values (*p*-values) for periods, as a few genes that are in proximity to each other can give a strong periodic signal with a single gene that is sufficiently far. Furthermore, as we study long-range regularities on bacterial chromosomes, we need to remove the sequential organisation of co-regulated and cofunctional genes into operons [17]. Thus, in the first step, all operons are reduced to a single position, that of their first cistron, because it is closest to the transcriptional start point. In the second step, a set of proximal genes is replaced by a single site located at their barycentre. The proximity criterion is defined by the user by specifying the average intergenic distance for the organism under study (by default two times this average intergenic distance).

The software then executes the periodicity detection algorithm as it is described in [10] exhaustively, that is, it looks for every possible period in the set of genomic features of interest and evaluates each one independently. The periods are evaluated according to their *p*-value, after applying a correction calculation to account for multiple testing. Indeed, for relatively short periods, many periods get tested, thus increasing the chances that a significant pattern will be detected. The *p*-values are weighted to take this fact into account.

At this level the user can specify a cut-off for the *p*-values (by default the significance level of 0.05 is applied) of the periods that will get displayed. The selected *p*-values are plotted in a typical plot that is used frequently in analyses of periodic phenomena and is called the periodogram (Fig. 1). A periodogram provides a quick overview of the most significant periods in terms of *p*-values (it depicts both the initial as well as the weighted *p*-value), and the researcher can readily identify which periods (if any) are the significant ones for the set of genomic features of interest.

3.1.2 An Example from Hydrocarbon Metabolism Genes

Here, we provide a test case of our analysis by performing a full GREAT:SCAN:patterns analysis on the genes from *A. borkumensis* which are involved in alkane degradation. We manually selected all the members of the two hydrocarbon degradation systems of *A. borkumensis* from [15] and found their respective translation start sites positions along the *A. borkumensis* genome. This information is enough to generate an input file for a GREAT:SCAN analysis. The only extra information that is required is the genome length as well as the average intergenic size of *A. borkumensis* which is used by the software in order to remove genome proximity effects. The output of the software consists of a set of tables where all the relevant information for each period is collected as well as a periodogram which is depicted in Fig. 1. The GREAT:SCAN analysis of the *A. borkumensis* genes that are involved in hydrocarbon metabolism found periods which approach the full genome size of the microorganism. This indicates that the major organisational principle of the hydrocarbon metabolism genes is 1-D genomic clustering, a result that could have been speculated from the neighbouring genomic coordinates of several genes in the set. Please note that this proximal trend is detected despite the prior removal of direct neighbourhood by the algorithm. However, a significant period of around 50 kbp was detected also, a finding which can raise some interesting insights (*see* Sect. 3.1.2 and further discussion in the legend of Fig. 1).

3.1.3 Clustering In-Phase Genes

Periodically arranged genes have a specific radial position on the modulo period coordinates of each individual significant period. Visualising their modulo coordinates is of key interest for biological researchers because this view might provide insights on whether the

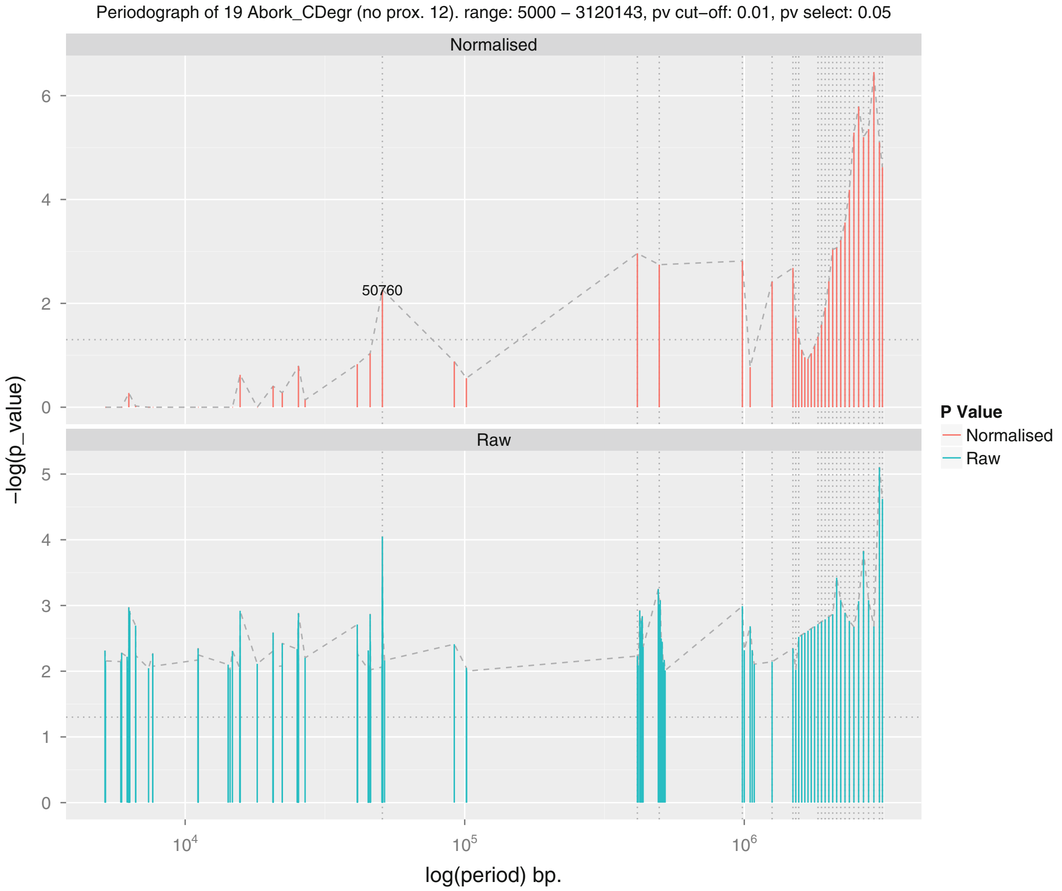

Fig. 1 Periodogram of genes involved in hydrocarbon metabolism of *A. borkumensis*. The height of the bars corresponds to the significance of the detected period (−log(*p*-value); thus if, e.g. the *p*-value equals 10^{-3}, the bar height is 3), the *dotted vertical lines* indicate highly significant periods (periods with *p*-value lower than the user specified *pvThres* parameter) and the *dashed line* connects the tips of the bars together to provide a view of regions with dense periodic signal detection. The *upper panel* depicts the same periods where the *p*-values have been normalised to correct for multiple testing and ordered by their size. The *lower panel* depicts the raw non-normalised *p*-values. Numerous significant periods are found to be close to the size of the whole *A. borkumensis* chromosome; this finding indicates that the proximal genomic arrangement of hydrocarbon metabolism genes is significant (*see* also Note 4.3 in the text). However, in the lower end of the spectrum, a few bars are also significant, which indicates a periodic pattern and suggests a potential 3-D solenoid arrangement of genes. Notably, a significant peak is detected for period 50,760 bp. A further step (Sect. 3.1.2) of the analysis with GREAT:SCAN:patterns can provide more information about that finding

set of the genes of interest can be found to be co-localised in the 3-D folding of the chromosome and take advantage of any proximity or local concentration effect for their transcriptional activity. We employ a simple density clustering approach based on an algorithm known in data sciences as DBSCAN [18]. DBSCAN is an unsupervised clustering algorithm that requires two parameters to find clusters, the minimum size of the cluster (which is set to two genes by default) and the minimum distance between points (which is set as the ratio of the average intergenic size to the period; this ratio is normalised by a single parameter called the clustering exponent, set to 0.5 by default). This density-based clustering technique is applied to the modulo period coordinates of the genomic features of interest for all of the significant periods that have resulted from the previous periodicity analysis step (Sect. 3.1.1). For each of the significant periods, the user obtains a table of the clustered genes including their position information score as well as a unique plot for each of the significant periods that we call a clustergram. A clustergram visualises the formation of the clusters of the genomic features of interests after plotting their modulo coordinate (phase) on the *x*-axis and their phase ranking on the *y*-axis. The clustergram automatically colours the clustered genes according to the cluster they belong to. However, a viewer can also identify clusters by looking for vertical alignment of genomic features of interest in the plot (Figs. 2 and 3). Genes belonging to a cluster will appear to be perfectly aligned on a vertical line in a clustergram plot.

3.1.4 An Example from Hydrocarbon Metabolism Genes

Continuing the analysis of the genes involved in hydrocarbon metabolism of *A. borkumensis*, GREAT:SCAN:patterns computed the clustergrams of all the significant periods from the previous analysis step (*see* Sect. 3.1.1). The results from the periodogram analysis indicate that most of the significant periods are similar to the genome length thus implying a 1-D genomic proximity arrangement of the hydrocarbon metabolic genes. The clustering analysis corroborates that further, as genes are clustered for the period of 3,043,845 bp as it is demonstrated in the clustergram of Fig. 2. However, a much shorter period of 50,760 bp was also found to be significant, and the hydrocarbon-involved genes appeared to cluster well (Fig. 3), suggesting a potential 3-D clustering of hydrocarbon metabolism genes.

3.1.5 Chromosome Mapping

So far, we considered periods that span the full length of the genome. Each period analysed and studied up to this section refers to the full set of the genes (or the genomic features) of interest as it is positioned on the whole genome. However, there might be cases where only a certain chromosomal region displays periodic arrangement. This section of the protocol will provide the tools and techniques to analyse these cases too.

To address this requirement, an additional feature of the algorithm was developed: the periodicity analysis can be performed in a sliding window. We have developed a 'mapping' algorithm where a sliding window approach is scrounging the whole genome on multiple scales in order to identify periodic regions. This section (and the following) will provide the steps to perform chromosome mapping analysis and interpret the results.

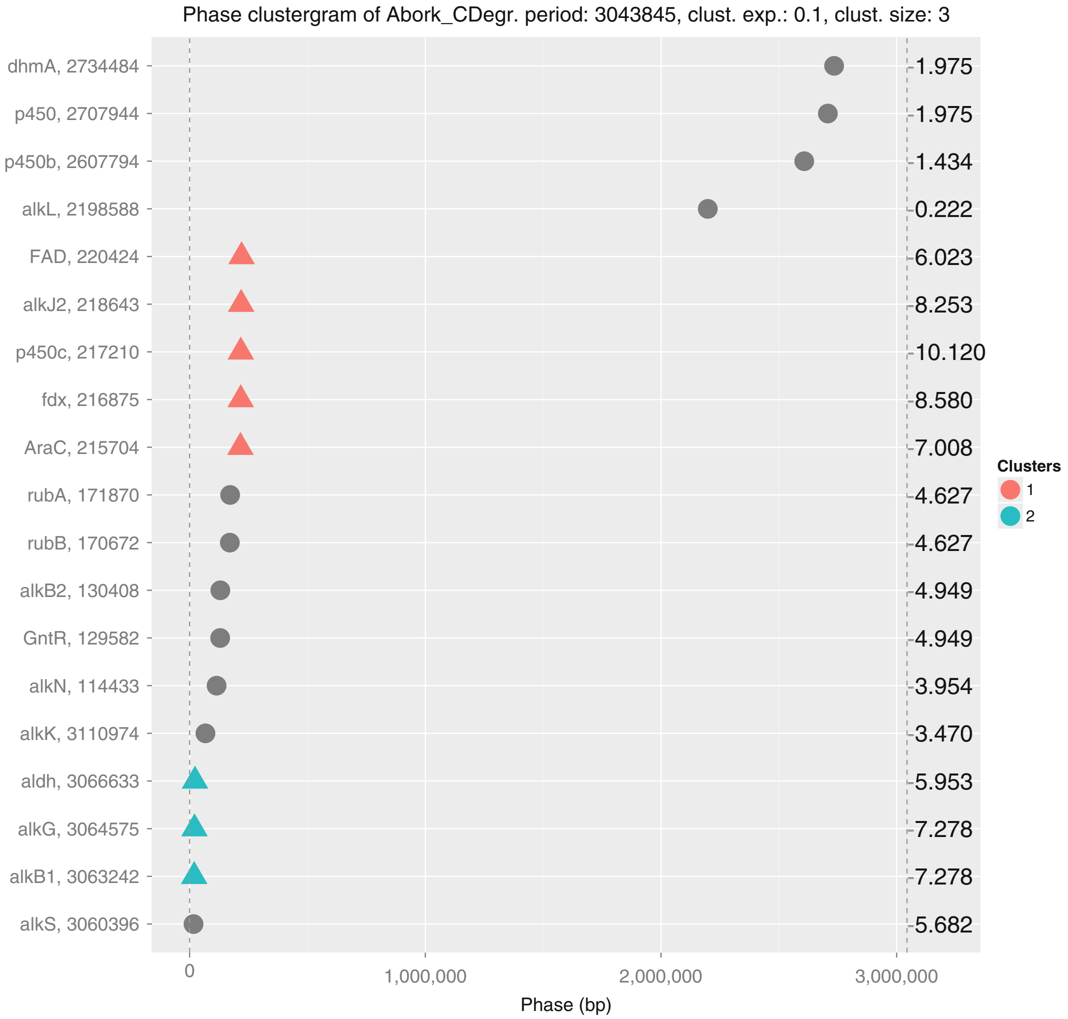

Fig. 2 Clustergram of the hydrocarbon metabolism genes of *A. borkumensis* for a period close to full genome length. The *x*-axis represents the length of one whole period, and each location corresponds to the phase (modulo coordinate) of each genomic feature of interest. Thus, any vertical quasi-alignment of the points denotes a gene cluster. The *left y-axis* shows the gene name and its genomic position; the *right y-axis* shows the positional information score of each gene (a score which corresponds to the individual contribution of each genomic feature to the clustering for this particular period). Cases like this, where the period approaches the genomic length, capture 1-D proximity, because proximity is detected by going around the *full circle* of the genome and falling back in the same neighbourhood

The period-scanning algorithm that is described in [10] and is used in section 3.1.1 to detect periods in the whole genome is adapted with a sliding window approach so that it operates in segments of the genome. The size of the window is specified by the user; however, a default value of 10,000 bp that grows incrementally to the whole genome length is used and provides the right

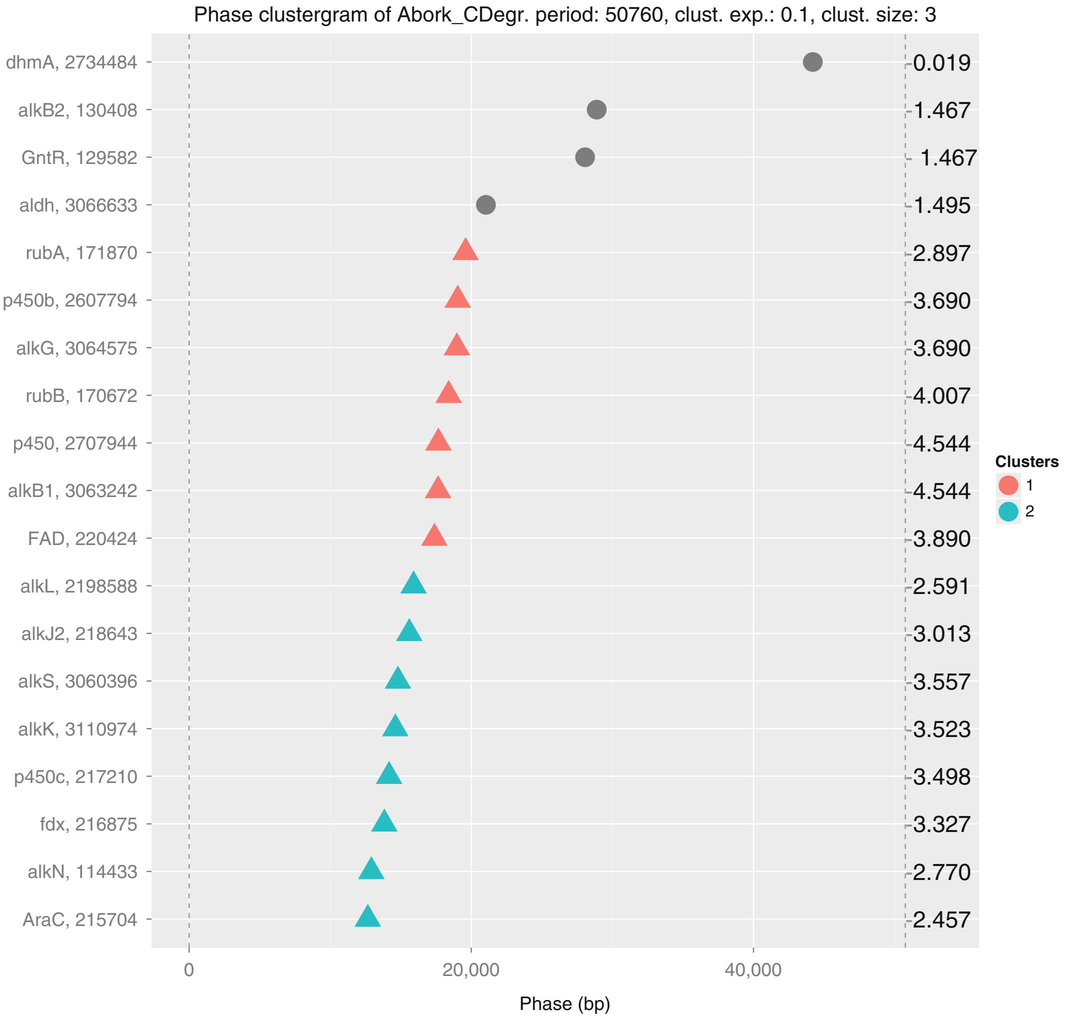

Fig. 3 Clustergram of the hydrocarbon metabolism genes of *A. borkumensis* for a period of 50,760 bp. Cases, like the one illustrated where the period is much lower than genome length, may be interpreted as 1-D periodicity, suggestive of 3-D arrangement and clustering of genes [2, 6]. Genes, or genomic features of interest, with a high position information score, are the top candidates for further investigation of potential 3-D co-localisation. The caption of Fig. 2 describes the details regarding the graph

results in any occasion. The user can also specify a *p*-value cut-off for the periods that are selected for plotting.

3.1.6 An Example from Hydrocarbon Metabolism Genes

We continue the analysis of the genes from *A. borkumensis* which play a central role in hydrocarbon metabolism by analysing their organisation in a finer scale using the 'sliding window' version of the periodicity detection algorithm. This allows the user to obtain a graph of the whole genome of the organism of interest where the detected periods on each particular segment are immediately observable together with information about the number of

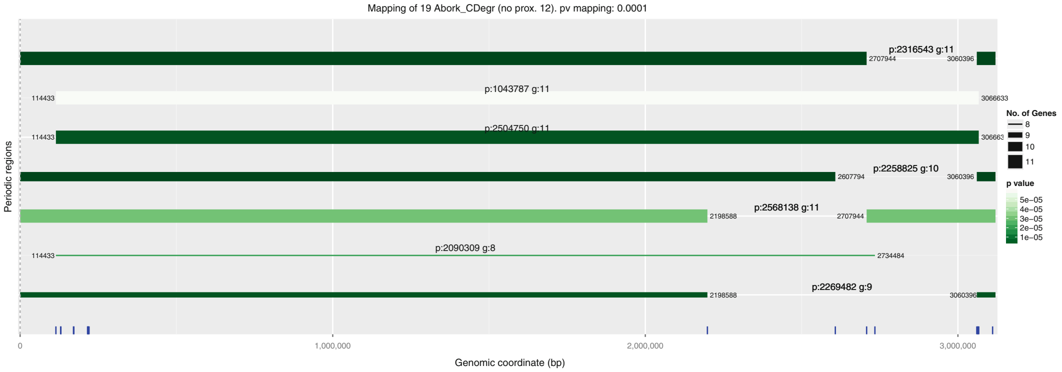

Fig. 4 The period mapping graph (or 'chromogram') of the hydrocarbon metabolism genes of *A. borkumensis*. Here, segments of the genome that contain periodic genes or genomic features of interest are depicted. The *x*-axis displays the genomic coordinates for the full genome length. The *y*-axis is used only to order the segments according to the segment size. The *thickness of the segment* denotes the number of genes that belong to this segment. The *colour code* corresponds to the *p*-value for this period. *Thickness* and *colour scales* self-adjust to the data and chosen parameters and are indicated on the *right side of the plot*. For each significant segment, its end coordinates, period value (p:) and number of genes (g:) appear in the text just above the middle of the segment. The *blue ticks* on the horizontal axis demark the genomic position of the input data. Additionally (not shown), the user can specify some genomic landmarks of interest from the parameters of the program. NOTE: all the above-mentioned plots were obtained by running the GREAT:SCAN:patterns program and using the following parameters: avgGene: 1000, clustExp: 0.1, clustSize: 3, infile: alcanivoraxHC_Metabolism.txt, length: 3120143, perRange: 5000, 3120143, pvSelect: 0.01, pvThres: 0.05, mapSelect: 0.001

involved genes, the *p*-value of each period and genomic locations of interest which can be superimposed on the graph. We call this plot a 'period mapping plot' or 'chromogram', and the result for the hydrocarbon metabolism genes is illustrated in Fig. 4.

3.2 GREAT:SCAN: PreCisIon

Current methods for the identification of cis-regulatory elements are marginally successful in their ability to discriminate between many alternative variants of the possible TFBSs. While the data on the consensus sequences for the corresponding regulatory sites are available, it often contains motifs with very low sequence conservation (like TCRNNNNNNACG, where N can be any nucleotide). Such degenerate consensus sequences lead to high false-negative and false-positive rates. The difficulty lies in the specific nature of DNA-protein interactions. Our method PreCisIon addresses this issue by taking into account both views: (a) local binding sequence readout and (b) global genome layout readout. The underlying rationale is based on the observation that co-regulated genes tend to be positioned at periodic intervals along the chromosome (*see* Sect. 3.1). The combined classifier is then obtained with an iterative weight update scheme, using a modified version of the AdaBoost algorithm. PreCisIon consistently improves methods based on consensus-binding sequence information only. This is shown by implementing a cross-validation analysis of the 20 major transcription factors from two phylogenetically remote model organisms. For *Bacillus subtilis* and *Escherichia coli*, respectively, PreCisIon achieves on average an AUC (area under the ROC curve) of 70% and 60%, a sensitivity of 80% and 70% and a specificity of 60% and 56% [11, 12].

4 Notes

GREAT:SCAN analyses might not always detect significant periods or clusters, and might not return some (or any) plots. Even though the software tries to prevent the most common mistakes a user can make (i.e. wrong parameter choices) and suggest the most common solution, there are some cases where the plots and the results are not easily interpretable. Here, we collect a couple of these cases and give some explanations of why it happened as well as how to solve the problem.

4.1 Significant Periods Not Detected

There might be cases where significant periods will not be reported. There are two reasons why this might happen. Firstly, a genuine reason is that the input data do not contain any genomic features that are periodically arranged in the genome. Please note however that periodicity of cofunctional genes has been detected in all eubacterial phyla [6] and in baker's yeast [4]. A second reason is that the parameters for reporting the periods to plots (and tables) are very stringent, and thus none of the periods passed the thresholds. This is often the case with the region mapping algorithm. As the chromosome periodical mapping (Sect. 3.1.3) zooms on segments of the chromosome with a small portion of the whole datapoints, the *p*-value of these periods is generally much lower than the *p*-value of periods that refer to the whole genome. Therefore, the default value for the plotting of these mapped periods is much lower than the level of significance of 0.05 (set to 0.001 by default in the web server). If an analysis fails to return any periods in the chromosome mapping plot, then try to increase this threshold *p*-value.

4.2 Clusters of In-phase Features Not Detected

The clusters reported in the clustergram analysis of Sect. 3.1.2 are calculated by a local density cluster approach. The algorithm that is used is called DBSCAN, and it requires two parameters: the cluster size (by default 3) and the minimum distance between members of the cluster (specified by the clustering exponent). The clustering exponent is applied on the ratio between the period and the genome length which specifies the minimum distance parameter for clustering. The exponent ranges between 0 and 1; the closer to 1, the lower the effect of the length of the period towards clustering sensitivity is, therefore clustering becomes more sensitive for a given period. For values 0 or close to it, the minimum distance for clustering becomes the largest possible, and thus clustering becomes less sensitive. As a rule of thumb, when no cluster of in-phase genes has been detected, it is advisable to lower the clustering exponent (the default is set to 0.5).

4.3 Period Nearly Equals Genome Length

GREAT:SCAN:patterns may return periods which equal the total genome length of the organism of interest or total genome length divided by a small integer. This was for instance the case with *A. borkumensis* (Sect. 3.1.1). Such very long periods denote significant proximity (1-D clustering) patterns. Indeed, it is known from the genome sequence of *A. borkumensis* [15] that there are several gene clusters where the hydrocarbon degradation genes are organised. This fact was evident from the periodicity analysis with GREAT: SCAN:patterns in Sect. 3.1.1, when the *patterns* procedure detects periods close to the genome length.

In sum, one interesting feature of the *pattern* algorithm is that it detects 1-D proximity and 3-D periodicity patterns in a single pass and provides *p*-values for both features that can be directly compared [10].

Acknowledgments

The authors thank the MEGA team members at iSSB for excellent discussions. This work was supported by Genopole, by the OSEO/ BPI-France 'BioIntelligence' consortium and by the EU FP7 KBBE project 'ST-FLOW'.

Appendix 1: Input File Format for a GREAT:SCAN:patterns Analysis (This Example Contains the Genes from *A. borkumensis* Involved in Hydrocarbon Degradation)

```
dhmA, 2734484
alkB1, 3063242
alkB2, 130408
aldh, 3066633
alkK, 3110974
alkL, 2198588
alkN, 114433
rubB, 170672
rubA, 171870
GntR, 129582
p450, 2707944
p450b, 2607794
p450c, 217210
fdx, 216875
alkJ2, 218643
FAD, 220424
AraC, 215704
alkG, 3064575
alkS, 3060396
```

Appendix 2: Usage Message of GREAT:SCAN:patterns

```
usage: patterns.R [-h] -t [<title> [<title> ...]]
                  [-l <genome_in_bp>]
                  [-a <avgGene_in_bp>]
                  [-r [<per_bounds> [<per_bounds> ...]]]
                  [-p <pvalue_thres>]
                  [-s <pvalue_select>]
                  [-d [<set_coords> [<set_coords> ...]]]
                  [-k [<set_ticks> [<set_ticks> ...]]]
                  [-c <clust_exponent>]
                  [-z <cluster_size>]
                  [-m <pvalue_mapping>]
                  [-i [<a_uniq_ID>]] [-v <path>]
                  [-o <output_path>]
                  <file_name>
Systematically analyse, cluster and visualise results from
a complete GREAT:SCAN analysis. Full global_spectrum (-DOM
and -CIRC analysis) followed by a DBSCAN clustering to identify
the in-phase genes and a solenoid_map (sliding window)
analysis and visualisation of the spread of all the possible
periods.

positional arguments:
  <file_name>     The input file consisting of two columns of
                  data formatted like this: <entity_ID>
                  <entity_position>


optional arguments:
  -h, –help       show this help message and exit
  -t [<title> [<title> ...]], –title [<title> [<title> ...]]

                  A substring to specify a title for the
                  experiment
                  (default: None)

  -l <genome_in_bp>, –chrom_length <genome_in_bp>

                  The length in bp of the organism
                  chromosome
                  (default: 4639675)

  -a <avgGene_in_bp>, –avg_gene <avgGene_in_bp>

                  The average gene length of the organism
                  genes
                  (default: 1000)

  -r [<per_bounds> [<per_bounds> ...]], –period_range
[<per_bounds> [<per_bounds> ...]]

                 The range (min. – max.) within which periods
                 will be considered for further analysis
                 (default: 5000)
```

-p <pvalue_thres>, -pvalue_thres <pvalue_thres>

The unweighted *p*-value threshold for considering a period for further analysis (default: 0.05)

-s <pvalue_select>, -pvalue_select <pvalue_select>

The weighted *p*-value threshold for selecting which periods will be printed (default: 0.05)

-d [<set_coords> [<set_coords> ...]], -plot_coords [<set_coords> [<set_coords> ...]]

Specifies a set of genomic coordinates to be printed as significant genome marks in the mapping plot (the *E.coli* macrodomains are defaults:
[46396, 603158, 1206296, 2180612, 2876552, 3758076])

-k [<set_ticks> [<set_ticks> ...]], -plot_ticks [<set_ticks> [<set_ticks> ...]]

Specifies a set of axis ticks to be printed as indicators of genome marks in the mapping plot (must be equal size with the coordinates).
(default: ['ori', 'right', 'R/ter', 'ter/L', 'left', 'ori'])

-c <clust_exponent>, -clust_exp <clust_exponent>

The clustering exponent. Assigns the minimum distance d between two points to be members of the same cluster. Specifies the exponent of the ratio between the length of the period and chromosome length (p/L). (default: 0.5)

-z <cluster_size>, -clust_size <cluster_size>

The minimum number of members for a group to be considered as a cluster (DBSCAN parameter)
(default: 2)

-m <pvalue_mapping>, -pvalue_map <pvalue_mapping>

The weighted *p*-value threshold for selecting which sliding window periods will be plotted (default: 0.001)

-i [<a_uniq_ID>], -uniq_ID [<a_uniq_ID>]

The unique ID for the generation of the results folder. (default: patternAnalysis_ xxxx_xx_xx)

```
-v <path>, -pv <path>

                The path to the 'pv' fit parameters file.
                (default:    <installation_of_cmdline_
                programs>)

-o <output_path>, -output_path <output_path>
                The absolute path for a directory (exist-
                ing one including the trailing slash '/')
                where the output will be kept, or omit for
                the current working directory. (just the
                path, the directory name itself is con-
                trolled by the -i option).
                (default: <current_working_dir>)
```

References

1. Huynen MA, van Nimwegen E (1998) The frequency distribution of gene family sizes in complete genomes. Mol Biol Evol 15 (5):583–589
2. Dorman CJ (2013) Genome architecture and global gene regulation in bacteria: making progress towards a unified model? Nat Rev Microbiol 11:349–355
3. Képès F, Vaillant C (2003) Transcription-based solenoidal model of chromosomes. ComPlexUs 1:171–180
4. Képès F (2004) Periodic transcriptional organization of the *E.coli* genome. J Mol Biol 340:957–964
5. Képès F (2003) Periodic epi-organization of the yeast genome revealed by the distribution of promoter sites. J Mol Biol 329:859–865
6. Junier I, Hérisson J, Képès F (2012) Genomic organization of evolutionarily correlated genes in bacteria: limits and strategies. J Mol Biol 419:369–386
7. Wright MA, Kharchenko P, Church GM, Segré D (2007) Chromosomal periodicity of evolutionarily conserved gene pairs. Proc Natl Acad Sci U S A 104:10559–10564
8. Ma Q, Ying X (2013) Global genomic arrangement of bacterial genes is closely tied with the total transcriptional efficiency. Genomics Proteomics Bioinformatics 11:66–71
9. Porcar M, Danchin A, de Lorenzo V (2014) Confidence, tolerance, and allowance in biological engineering: the nuts and bolts of living things. Bioessays 37:95–102
10. Junier I, Hérisson J, Képès F (2010) Periodic pattern detection in sparse boolean sequences. Algorithms Mol Biol 5:31
11. Elati M, Fekih R, Nicolle R, Junier I, Herisson J, Képès F (2011) Boosting binding sites prediction using gene's positions. In: Algorithms in bioinformatics (WABI'11), LNCS – 6833, pp 92–103
12. Elati M, Nicolle R, Junier I, Fernández D, Fekih R, Font J, Képès F (2013) PreCisIon: PREdiction of CIS-regulatory elements improved by gene's positION. Nucleic Acids Res 41(3):1406–1415
13. Turatsinze JV, Thomas-Chollier M, Defrance M, van Helden J (2008) Using RSAT to scan genome sequences for transcription factor binding sites and cis-regulatory modules. Nat Protoc 3(10):1578–1588
14. Salgado H, Peralta-Gil M, Gama-Castro S, Santos-Zavaleta A, Muñiz-Rascado L, García-Sotelo JS, Weiss V, Solano-Lira H, Martínez-Flores I, Medina-Rivera A, Salgado-Osorio G, Alquicira-Hernández S, Alquicira-Hernández K, López-Fuentes A, Porrón-Sotelo L, Huerta AM, Bonavides-Martínez C, Balderas-Martínez YI, Pannier L, Olvera M, Labastida A, Jiménez-Jacinto V, Vega-Alvarado L, Del Moral-Chávez V, Hernández-Alvarez A, Morett E, Collado-Vides J (2013) RegulonDB v8.0: omics data sets, evolutionary conservation, regulatory phrases, cross-validated gold standards and more. Nucleic Acids Res 41:D203–D213
15. Schneiker S, Martins dos Santos VAP, Bartels D, Bekel T, Brecht M, Buhrmester J, Chernikova TN, Denaro R, Ferrer M, Gertler C, Goesmann A, Golyshina OV, Kaminski F, Khachane AN, Lang S, Linke B, McHardy AC, Meyer F, Nechitaylo T, Pühler A, Regenhardt D, Rupp O, Sabirova JS, Selbitschka W, Yakimov MM, Timmis KN, Vorhölter F-J, Weidner S, Kaiser O, Golyshin PN (2006) Genome sequence of the ubiquitous hydrocarbon-degrading marine bacterium *Alcanivorax borkumensis.* Nat Biotechnol 24:997–1004

16. Schneider KL, Pollard KS, Baertsch R, Pohl A, Lowe TM (2006) The UCSC archaeal genome browser. Nucleic Acids Res 34:D407–D410

17. Salgado H, Moreno-Hagelsieb G, Smith TF, Collado-Vides J (2000) Operons in *Escherichia coli*: genomic analyses and predictions. Proc Natl Acad Sci U S A 97:6652–6657

18. Ester M, Kriegel H, Sander J, Xu X (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis E, Han J, Fayyad UM (eds) Proceedings of the second international conference on knowledge discovery and data mining (KDD-96), Portland. AAAI, pp 226–231