Dissertation – MSc Synthetic Biology

Deriving stress-specific biomarkers for *Bacillus subtilis* from the integration of RNA-seq and tiling data

Delia Casas Pastor[1]

[1]School of Computing Sciences, Newcastle University, UK

Supervisors: Prof. Anil Wipat, Dr. Goksel Misirli and Dr. Wendy Smith

Abstract

Motivation: Stress processes are the cause of loss of performance of industrial cultures of bacteria. Synthetic biology provides the tools to address this problem but the discovery of stress-specific biomarkers to identify and address the onset of a particular stress remains unsolved.
Results: This work describes a new algorithm for the retrieval of stress-specific biomarkers that applies two sequential feature selection algorithms to high-throughput gene expression data in *Bacillus subtilis*. Then, an inverse C-element circuit is designed using a black box approach. As an *in silico* proof of concept of this design, the regulatory sequences of the top two oxidative stress biomarkers are set as inputs of this circuit with the objective of easing the stress.
Contact: d.casas-pastor2@newcastle.ac.uk

1 Introduction

Synthetic biology consists of the application of engineering approaches to life science aiming at the design of novel biological systems. For doing so, it requires from the integration of several disciplines that, altogether, enable the coupling of biological parts, devices and circuits so as to make a target chassis able to fulfil a predefined specification. Synthetic biology has been applied to the design of cell factories to produce high-value compounds (Mahalik, et al., 2014). Currently, the main organism used for the production of heterologous proteins in industrial processes is *Escherichia coli* (Demain and Vaishnav, 2009). However, *Bacillus subtilis* is widely used for homologous expression of enzymes and it provides several advantages over *Escherichia coli* in the heterologous production, such as the lack of endotoxins and a high secretion yield (Demain and Vaishnav, 2009).

At present, there are several repositories that host functional information about biological parts of *E. coli* and *B. subtilis* (Misirli, et al., 2014), but there is still a need for more parts to expand the functionality of synthetic circuits. Moreover, the great complexity of the molecular interactions within the cells used as chassis and the lack of a host with minimum genome prevents the use of context-independent parts (Choe, et al., 2016). Therefore, the increase of the pool of available genetic parts for *B. subtilis* goes through the specific characterisation of its endogenous regulatory mechanism.

One of the areas were the application of synthetic biology would be advantageous is in the track and control of cellular stress. High-yield engineered bacteria often suffer from stress processes that activate feedback responses that diminish both cellular growth and recombinant protein production (Mahalik, et al., 2014). Some attempts have been made to overcome this stress response in *B. subtilis* (Carneiro, et al., 2013; Ceroni, et al., 2015); nevertheless, none of them managed to dynamically respond to specific changes in the host's metabolism.

The stress response is a natural mechanism of adaptation to changes in the environment that decrease the fitness of the organism (Sulmon, et al., 2015). The presence of an external stressor is a threaten to the survival of the cell as it causes metabolic imbalances that, eventually, can lead to death (Sulmon, et al., 2015). Nevertheless, cells are able to fight back activating intracellular signalling pathways so as to adapt to the new suboptimal growth conditions (de Nadal, et al., 2011).

The stress response can be divided into two categories: a generic response that provides cross-protection against several stressors and a specific adaptive response, in which cells specifically respond to one stressor (de Nadal, et al., 2011; Price, et al., 2001; Sulmon, et al., 2015). The generic response genes are typically involved in primary metabolism, transport and detoxification, protein homeostasis, intracellular signalling and DNA repair (de Nadal, et al., 2011). Although the same stress affects similar processes across the tree of life (Sulmon, et al., 2015), the stress adaptive response greatly depends of the organism, its life-cycle stage (de Nadal, et al., 2011) and the specific stressor.

After the stressor has been sensed and the signal has been transduced, the most immediate cellular responses are post-translational modifications (PTM), which provide a rapid defence against stress, whereas gene expression regulation provides long-term adaption to stress (de Nadal, et al., 2011). As a result, gene expression changes are a major mechanism in cells adaptive response to stress (de Nadal, et al., 2011).

1.1. *Bacillus subtilis*' stress response

B. subtilis responds to harsh conditions using a battery of mechanisms that include cell specialisation (genetic competence and sporulation), as well as stress-specific responses to protect, repair and detoxify the cell

(Zuber, 2009). Generally, σ^B is the sigma factor that recognises the promoters of genes related to stress protection (Hecker and Volker, 1998; Schumann, 2003; Zuber, 2009).

Among the adaptive stress response, oxidative and heat-shock responses are commonly activated in overproduction strains and they are one of the main agents responsible for their loss of productivity (Hoffmann and Rinas, 2004). Therefore, it would be desirable to wire these responses to the production of the overexpressed gene so as to switch off its transcription when the stress response is active.

1.1.1.Oxidative stress response in *B. subtilis*

Oxidative stress is the biological condition caused by the exposure of a cell to oxidising agents that are able to take electrons from biomolecules such as DNA and redox enzymes, damaging their structure, disrupting their functionality and leading to mutagenesis and cellular death (Zuber, 2009). Among these agents, Reactive Oxygen Species (ROS) such as O_2^- and H_2O_2 are normally generated as by-products of aerobic metabolism (Imlay, 2015), especially in strains with great energy expenses, such as industrial strains. *B. subtilis* contains enzymes to degrade ROS (superoxide dismutases, peroxidases and catalases) (Imlay, 2015). Nevertheless, oxidising agents can also have an external source, for example, the herbicide paraquat is able to trigger the production of O_2^- and H_2O_2, while diamide is able to directly oxidise thiol groups of proteins (Kashyap, et al., 2014).

ROS can disrupt cellular structures and metabolism through different targets, such as exposed $[4Fe\text{-}4S]^+$ clusters and thiols from cysteine residues in proteins (Zuber, 2009). $[4Fe\text{-}4S]^+$ clusters are typically found in the active centres of redox enzymes, where ROS are able to undercoordinate Fe^{2+}. Fe^{2+} is subsequently oxidised to Fe^{3+} as a consequence of the intracellular redox imbalance, which in turn leads to the production of hydroxyl radicals that have the potential to damage most biomolecules and to cause mutations (Herbig and Helmann, 2001; Imlay, 2015; Zuber, 2009). Furthermore, ROS are able to disrupt the oxidative metabolism of the cell due to their electron-scavenging activity (Imlay, 2015).

In *B. subtilis* the treatment with paraquat and H_2O_2 triggers the expression of the operons repressed by PerR, Fur, Spx, OhrR and CymR, among others (Helmann, et al., 2003; Tam, et al., 2006; Tanous, et al., 2008; Zuber, 2009). Fe^{2+} is normally sensed by Fur and PerR, two repressors of the expression of the iron uptake proteins (Zuber, 2009). Fur and PerR are not able to recognise Fe^{3+}, the main valence of iron after exposure to ROS, leading to an increase in the uptake of iron during oxidative stress, which promotes the disruption of more cellular structures (Lee and Helmann, 2006; Varghese, et al., 2007; Zuber, 2009).

1.1.2. Heat stress response in *B. subtilis*

The exposure to high temperature increases the likelihood of proteins to reach non-native conformations, not usually functional and with tendency to aggregation (Schumann, 2003). *B. subtilis* copes with the heat stress regime upregulating the expression of chaperones and proteases; chaperons prevent the denaturalisation of proteins, while proteases degrade proteins in their non-native conformation (Schumann, 2003).

B. subtilis' heat response cascade is induced above 48ºC and it is started by direct sensors, i.e. RNA and proteins that have a temperature-dependent conformation; and indirect sensors, i.e. chaperones that modulate the activity of transcription factors and are titrated by denatured proteins (Schumann, 2003).

1.2. Supervised machine learning for feature selection

The gene expression intensity under different conditions can be used to explore which genes (hereafter also called features) respond specifically to a particular stress and can be considered stress-specific biomarkers. The regulatory sequences of these genes could be used as inputs to rewire the stress response so as to improve cellular fitness.

Bacillus subtilis subsp. subtilis str. 168 has a total of 4,421 CDSs (Coding DNA Sequence) (NC_000964.3 NCBI) and most of them would either not be related to stress or be part of the generic stress response. Moreover, the increased complexity, the cross-talk between parts and the detrimental effects of stochastic processes in the wiring would diminish the efficiency of a circuit with more than 2 inputs. Consequently, a feature selection procedure is needed to decide which features are able to explain most of the changes between stress and control conditions.

Feature selection algorithms fall into four categories: filters, wrappers, embedded and hybrid methods, which combine different strategies (Bolon-Canedo, et al., 2014). Embedded methods are a trade-off between wrappers and filters: they have a closer interaction with the classifier than filters, while keeping a smaller computational cost than wrappers (Bolon-Canedo, et al., 2014). Recursive Feature Elimination (RFE) is an embedded method extensively applied to gene expression analysis due to its performance (Bolon-Canedo, et al., 2014). It consists of iteratively training a classifier and removing the feature with the lowest score on each iteration (Bolon-Canedo, et al., 2014).

Feature selection algorithms depend on classifiers to rank features according to their importance to distinguish groups of samples. The most popular classifier applied to RFE is Support Vector Machine (SVM), but other common classifiers such as Random Forest (RF) could also be utilised (Granitto, et al., 2006).

Random Forest is an ensemble classifier in which several decision trees are built from a training dataset, forming a forest. It is frequently used due to its simple theory, high speed, stability, robustness and small model overfitting (Chen, et al., 2013). It is a bagging method as each tree is built from a bootstrap sample drawn from the training set with replacement (Breiman, 2001). Inside each decision tree, each split is picked from a random subset of features using the *gini* impurity index to decide which feature is able to divide the bootstrap sample into purer subsets (Breiman, 2001; Pedregosa, et al., 2011). Once it is built, RF is used to classify a test set according to the mode of the prediction for each tree in the forest (Pedregosa, et al., 2011).

SVMs are commonly used for the analysis of high-throughput biological experiments as they possess a good classification accuracy keeping the computational cost low, although they tend to overfit models (Bolon-Canedo, et al., 2014; Fang, et al., 2012; Guyon, et al., 2002). SVMs project each sample in an n-dimensional space as an n-dimensional vector, where n is the number of features. Then, they draw the hyperplanes able to separate samples belonging to different groups. The selected hyperplane is the one with the maximum margin, this is, the greatest distance between the nearest training samples belonging to different groups, or support vectors. Consequently, the features that determine to the position of support vectors are the ones that contribute more to the classification (Granitto, et al., 2006; Guyon, et al., 2002; Pedregosa, et al., 2011; Scholkopf and Smola, 2001). SVMs use different kernels to compute the margins but linear kernels provide the best results in terms of speed and accuracy in tasks with a small ratio groups/features (Granitto, et al., 2006; Scholkopf and Smola, 2001). The problem arises when the training samples belonging to different groups are not linearly separable and some of them are misclassified. The soft-margin approach is used in this case, which employs the parameter C, or penalty of the error term, to decide which is the best trade-off between margin maximization and misclassification minimization (Scholkopf and Smola, 2001). Low values of C retrieve a greater margin, whereas high values tend to classify all training samples correctly (Scholkopf and Smola, 2001).

The heuristic RGIFE (Ranked Guided Iterative Feature Elimination) was used as a first approach for finding the minimum subset of features able to classify control and stress samples with the maximum accuracy. RGIFE is a feature selection algorithm that iteratively removes groups of features until the performance of a RF classifier does not improve (Swan, et al., 2015). It usually returns more than two features; hence, another feature selector needs to be sequentially applied to reduce the number of features in detriment of the classifier's performance. Here, two feature selection strategies will be tested, RF-RFE and SVM-RFE.

1.3. Parameter optimisation

Computational operations have a great dependence on parameters of unknown value. However, these parameters can be determined optimising the result of a fitness or objective function given a set of constrains. Analytic procedures are the methods of election when the exact optimal value of the parameter needs to be found and the fitness function is simple enough. Nonetheless, for more complex functions, each value of the parameter has to be evaluated using an exhaustive optimisation. Exhaustive methods are not always possible since the computational expense increases exponentially for multiparametric optimizations or when the fitness function is stochastic. In these cases, heuristic optimisation methods, which retrieve an approximation of the optimal value, are utilised.

Simulated annealing is a stochastic global optimisation heuristic that iterates over a range of values evaluating the fitness function. It accepts three constants: a maximum and a minimum temperature, the rate of decrease of the temperature per iteration and the search space of each parameter to be optimized. A new value of the target parameters, or state, will be accepted if the output of the fitness function is improved with respect to the previous accepted state, or reference state. Otherwise the new state could be still accepted with a probability proportional to the temperature and inversely proportional to the difference between the new and the reference value of the fitness function. The new state to be evaluated is selected among the neighbours of the reference state. In this way, the search is facilitated by high temperatures at the beginning to scape local optima, whereas at the end the temperature is lower and the heuristic turns to be greedier so as to converge to the global maximum (de Amorim, 2009). The search stops when the solution is considered good enough or after a pre-fixed number of steps.

RF and SVM largely depend on two parameters, the number of trees in the forest (T) and the penalty of the error term in the soft-tail approach of SVM (C). Different optimization strategies of these parameters will be tested before the most discriminative features are retrieved. Then, these features will be used for the design of a genetic circuit.

1.4. Genetic logic synthesis

Genetic circuits are gene regulatory networks (GRN) that modulate an output response according to a set of input signals. They are composed by a set of genes and the set of their interactions arranged in gates to perform a defined logic function, similarly as electric circuits. Genetic logic can be implemented at different levels but transcriptional level, in which the interactions between genes involve the induction or repression of the binding of RNA polymerase (RNAP) to a promoter, is the one that currently offers more advantages (Vaidyanathan, et al., 2015). The complexity of genetic logic circuits grows with the number of elements it contains (Chaouiya, et al., 2004), impeding the implementation of complex behaviours such as sequential logic.

Sequential logic circuits are characterised by their ability to set an internal state so that their output depends on both inputs signals and this internal memory (Lou, et al., 2010), similarly to a finite-state automaton. This behaviour allows sequential circuits to perform more sophisticated functions than combinatorial circuits, whose output only depends on the inputs received.

Muller C-element is a sequential logic function resistant to transient fluctuations in the input signals. In a genetic context, it is able to set the expression of an output CDS to ON, or 1, when both inputs are active and to OFF, or 0, when there is no input. The robustness of this system comes from its memory, which enables it to keep the previous set state when only one input signal is present (**Table 1**). Several versions of a genetic C-element have been designed and simulated (Nguyen, et al., 2010) but a black box implementation that could be coupled to any input is still pending. The design and digital simulation of this circuit using Petri nets would help in its logic synthesis.

Table 1. Truth table for a Muller C-element

Input A	Input B	Output
0	0	0
0	1	Hold
1	0	Hold
1	1	1

Manual design is currently the most effective technique for sequential circuits (Nielsen, et al., 2016). This is an error-prone process, especially for complex circuits. In this context, the application of Petri nets supposes a benefit in both the design and the testing steps of the genetic logic synthesis. Petri nets are place-transition automata composed by a set of places or states and a set of directed transitions between places. Each place can accept a fixed number of tokens that are able to trigger or impede the transition to another place (Chaouiya, et al., 2004). Petri nets provide an scalable and standardized platform for representing GRN, where genes are places, transitions are transcriptional interactions and tokens are transcription factors (Bonzanni, et al., 2014). Furthermore, they provide a flexible platform that can be used to model and simulate Boolean, continuous, hybrid and stochastic systems (Heiner and Gilbert, 2013). Currently, the main application of Petri nets in biology is the analysis of biological pathways (Bonzanni, et al., 2014; Chaouiya, et al., 2004), but their utilisation in the design, analysis and simulation of synthetic circuits is at a preliminary stage, with some examples such as a model of a repressilator (Heiner and Gilbert, 2013).

1.5. General aim

Recently, stress processes in industrial strains have gained attention of the scientific community as a way to improve productivity. *In vivo* monitors of metabolic stress have been implemented for *B. subtilis* (Smith, *et al*., unpublished) and *E. coli* (Ceroni, et al., 2015); however, they are only fluorescence-based sensors not able to wire cellular responses towards the relief of the stress. Consequently, the aim of this work is to create a reproducible algorithm to derive gene markers whose expression can be used to monitor a specific stress. Then, a genetic inverse C-element will be designed as a black box using an orthogonal system of transcriptional repressors so as to connect it to any transcriptional process. As a proof of concept of the pipeline, the oxidative stress biomarkers of *B. subtilis* will be retrieved and set as the input of the circuit so as to control the output of a burdensome protein.

2 Methods

2.1. Data sources

Six exponential cultures of *B. subtilis subsp. subtilis str. BSB1* (similar to *Bacillus subtilis subsp. subtilis str. 168* for the purpose of this study) grown at 37ºC, of which 3 samples had been exposed to 0.1mM H_2O_2 10min, had been subjected to mRNA sequencing using IonTorrent platform and its recommended protocol. The subsequent reads had been quality assessed and trimmed using FastQC (Andrews, 2010) with cut-offs between 19 and 249. Then, they had been aligned against *B. subtilis'* reference genome (AL009126.3) using Bowtie2 2.2.2 (Langmead, et al., 2009). The number of reads per CDS had been quantified using HTseq-count routine (Anders, et al., 2015) and they were the starting point of this analysis.

The 169 NimbleGen tiling microarray samples had been hybridized with cultures subjected to different experimental conditions, including anaerobic growth, glucose depletion and starvation, high and low phosphate concentration, high (3 samples at 48ºC and 3 samples at 51ºC) and low temperature, high NaCl osmolarity, presence of ethanol or mitomycin C and 18 samples subjected to oxidative stress induced by 0.5mM diamide 15min (6 samples), 0.6mM diamide 10min (3 samples), 0.4mM paraquat (3 samples) or 0.1mM H_2O_2 (3 samples) (Nicolas, et al., 2009).

BacillusRegNet dataset is a GRN that contains a total of 1264 regulatory interactions between 861 genes in *B. subtilis 168* (**Fig. 1**). Interactions between genes that do not encode proteins are not included (Misirli, et al., 2014).

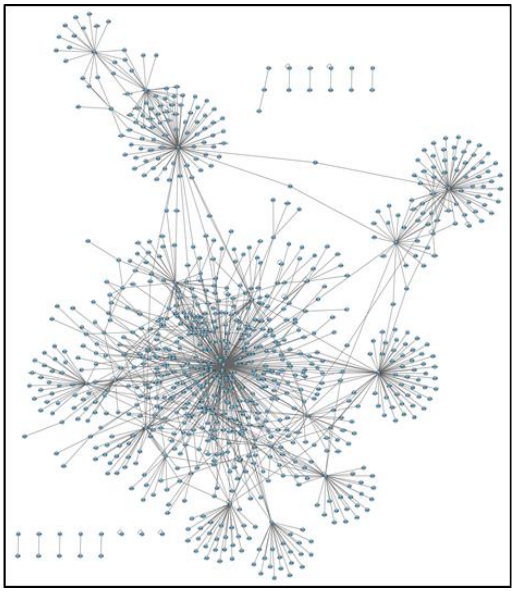

Fig. 1. **Global layout of BacillusRegNet data** displayed in Cytoscape.

2.2. Operating system, programming languages and software

Ubuntu 14.04.1 was executed in Windows 8 using the virtual machine VMware Workstation 12 with 1GB of RAM, 1 processor and 100GB of hard disk space. The programming languages used were R 3.3.0 written through the IDE (Integrated Development Environment) RStudio 0.98.1062, Python 2.7 through the IDE Spyder 2.3.9, and Java 8.0 in Eclipse Neon 4.6.0. Anaconda (Analytics, 2015) was used for installing new packages and as the platform for running Python 2.7. Cytoscape 3.3.0 was utilized to create the plots of the GRN and to execute jActiveModules, which enables the obtainment of subnetworks containing differentially expressed genes (Ideker, et al., 2002). Workcraft 3.1.0 was used for the design and modelling of the C-element circuit and its Petri net (Poliakov, et al., 2009).

Table 2. Packages used in this work

Package	**Language**	**Usage**	**Reference**
numpy 1.10.4	Python	Numerical computation	(van der Walt, et al., 2011)
scipy 0.17.1	Python	System specific parameters and functions	(van der Walt, et al., 2011)
random	Python	Pseudorandom numbers	Standard library
sys	Python	Access to system	Standard library
Classifiers_module	Python	RF and SVM	This work
os	Python	Operating system interface	Standard library
rpy2 2.7.8	Python	Run R in python	(Belopolsky, et al., 2014)
matplotlib 1.5.1	Python	Boxplots	(Hunter, 2007)
csv 1.0	Python	Read-write CSV files	Standard library
collections	Python	Container of datatypes	Standard library
sklearn 0.17.1	Python	Machine learning	(Pedregosa, et al., 2011)
simanneal2	Python	Simulated annealing	(Perry and Wagner, 2014)
re	Python	Regular expressions	Standard library
GOSemSim 1.30.2	R	Score GO terms	(Yu, et al., 2010)
preprocessCore 1.34	R	Quantile normalization	(Bolstad, 2016)
sva 3.20.0	R	ComBat for batch effects correction	(Leek, et al., 2016)

2.3. Biomarker retrieval algorithm

2.3.1. Integration of gene expression datasets

The entries corresponding to the 855 CDSs common to all datasets (RNA-seq, microarray and BacillusRegNet) were kept for the analysis. Firstly, the gene symbols, or human-friendly gene identifiers in NCBI database, were converted into locus tags, the identifier of the loci, using the database MicroScope for the reference genome of *Bacillus subtilis subsp. subtilis str. 168* as conversion key (www.genoscope.cns.fr). It was taken into consideration that symbol tags were contained in two different columns of the MicroScope's tab separated file and that some entries contained more than one gene symbol per tab slot. When the search was not successful, the locus tag was sought in BacillusRegNet table since it contains both locus tags and gene symbols. The gene symbols *ymfK, dnaE* and *rsfA* were manually assigned to their locus tag. As a result, entries with paralogous CDSs that use the same symbol tag were assigned to more than one locus tag. Finally, the entries of RNA-seq and microarray datasets were merged using locus tags as identifiers (Tilling_RNA_final_arg.R), resulting in two tab separated files: the RNA-seq and microarray expression profiles, from which the stress-specific biomarkers were derived, and the reduced version of BacillusRegNet dataset, which was used for plotting the genetic interactions of these biomarkers.

The gene expression data were pre-processed in order to make the values of the RNA-seq and microarray gene expression comparable. The different approaches tested contained one batch correction and at least one normalization step (Normalisation_trials folder). In the normalization, the process that homogenizes the scale of the expression data, two different methods were tested: minimum-maximum normalization (min-max) and quantile normalization (QN). Min-max normalization adjusts the distribution of each experiment to a scale of [0, 1] (1) (Chen, et al., 2013). It was selected as a simple and computationally affordable method of scaling data (Chen, et al., 2013). Alternatively or in conjunction with it, QN was applied since it is currently the best method to correctly cluster different microarray samples keeping the biological variation among genes' expression (Muller, et al., 2016). QN equalizes the expression intensity of the genes in the same quantile among samples, providing the same boxplots for all the samples (**Table 2**) (Muller, et al., 2016).

$$\frac{x-x_{min}}{x_{max}-x_{min}} \qquad (1)$$

Batch effects are technical artefacts not related to the system under evaluation but with the external conditions in which the experimentation is conducted (Muller, et al., 2016; Sun, et al., 2011). It has been probed that normalization is not enough to remove batch effects (Muller, et al., 2016; Sun, et al., 2011). RNA-seq and microarray experiments determine the expression intensity of each locus using radically different procedures: while RNA-seq sequences mRNA, microarrays hybridize it to probes on a chip. Therefore, the differences in the expression data arising from the different experimental settings do not have a biological origin and they can be associated to the variability between to two different batches. In this way, the differences between RNA-seq and microarray expression data were corrected using the Combat function from R sva package (**Table 2**). This function takes into account that the bias of a batch is common across all its samples to estimate a batch parameter utilized for correcting the batch effects with an empirical Bayesian method (Johnson, et al., 2007).

In order to check the degree of integration of RNA-seq and microarray samples during the different pre-processing tests, Ward's hierarchical agglomerative clustering algorithm with Euclidean distance was used (Ward, et al., 2001). Similarly, the equivalence of the distribution of the expression data among samples was checked using boxplots.

Once the best pre-processing technique was selected, technical replicates were averaged and stress samples were tagged as STRESS independently of the time point when the sample was drawn. In this way, the biomarkers retrieved are indicative of the exposure to the specific stressor and not of a temporary response (Scale_Normalise_batch.R).

According to the central limit theorem, the distribution of a random variable tends to a Gaussian distribution for a high number of samples; therefore, it was assumed that the expression values of each gene were normally distributed for CONTROL and STRESS groups and a two-tailed Student's t-test was applied to obtain the degree of differential expression of a feature in terms of p-values, this is, the probability of rejecting the null hypothesis (the expression does not change between stress and control samples) when it is true.

2.3.2. RGIFE heuristic

The gene expression matrix containing SAMPLE and CONTROL tags was transformed into .arff format (To_arff.R). The parameters set as input to RGIFE were selected for it to be highly restrictive: one repetition of a 10-fold distributed-balanced stratified cross-validation scheme, which assigns close-by samples to different folds so each fold contains representatives of every cluster (Zeng and Martinez, 2000), one misclassified sample to identify a soft tail, random forests with 3000 trees and a maximum depth of five and a misclassification cost of one (Lazzarini *et al.*, unpublished(Swan, et al., 2015). The metric used to evaluate the performance of the classifier was "robust_accuracy", which divides the overall number of correctly classified samples across folds by the total number of test samples. The biomarkers resulting from 10 executions of RGIFE were unified so as to obtain a broader range of biomarkers using the polices.py option of RGIFE (Lazzarini, et al., unpublished).

2.3.3. RF-RFE and SVM-RFE

RF-RFE and SVM-RFE were utilized in order to select the features whose expression is more distinctive of stress or control conditions. Both were executed 200 times so as to return the frequency of each feature being selected as the most discriminative biomarker.

For RF-RFE, RF classifier was executed using the function sklearn.ensemble.RandomForestClassifier (**Table 2**) with the recommended parameters (scikit-learn.org): size of the random subset of features checked for each node set to the square root of the total number of features and each tree spanned until pure leaves. The number of trees in the forest (T) was 10 when checking the performance of the pre-processing schemes, otherwise it was optimized since the documentation did not provide an adequate value for it. RFE was manually implemented with *gini* impurity index as the ranking criterion and one feature removed per iteration (Breiman, 2001; Chen, et al., 2013; Pedregosa, et al., 2011).

For SVM-RFE, SVM classifier was executed with a linear kernel using the function sklearn.svm.LinearSVC() with the default parameters. The penalty assigned to the error term (C) was one in the pre-processing schemes, otherwise it was subjected to optimization. The weight of each feature in the margin's location was used as scoring criterion by the RFE executed with the function sklearn.feature_selection.RFE() with one feature removed per iteration. The scripts to execute RF-RFE and SVM-RFE were stored as the Python's module Classifiers_module (**Table 2**), which is also able to directly run from shell.

2.4. Visualization of the GRN of biomarkers in Cytoscape

The number of entries of BacillusRegNet dataset was reduced to the subset of interactions in which the biomarkers resulting from RGIFE participated (RGIFE_to_cytoscape.R). This network was uploaded into Cytoscape using the instructions in Cline, et al., 2007. Nodes' key attribute was the gene symbol and edges tip distinguished between positive interactions (arrow tip), negative interactions (T tip) and sigma factor (straight line). The target binding sequence was included as an edges' attribute. The nodes corresponding to the genes retrieved by RGIFE were represented in a different color. From them, the genes also retrieve by the RF-RFE and SVM-RFE in more than 10% of the executions were highlighted. The housekeeping sigma factor σ^A was removed of some plots so as to improve the clarity. The Cytoscape's plugin jActiveModules was executed to retrieve highly significant subnetworks importing the p-values of the Student's t-test analysis as node's attribute (Ideker, et al., 2002).

2.5. GO scoring

GO defines standard terms to refer to the domains Molecular Function, Biological Process and Cellular Component, which reflect to the elemental function, biochemical process and subcellular location of the protein encoded by a gene (www.geneontology.org). Terms are organized as a directed acyclic graph, where each term can share parent-child relationships with others so that it is possible to calculate the distance between two GO terms (www.geneontology.org).

Therefore, the GO score used as fitness function of the optimization step was defined as the semantic proximity between the GO terms of the biomarkers retrieved by RF-RFE or SVM-RFE and the biotic and abiotic stress GO term GO:0006950 with its 33 child terms. This score was calculated using the function mgoSim from the R package GOSemSim (**Table 2**) with Wang's distance, which utilizes the topology of the GO graph to compute the distance between terms (Wang, et al., 2007). The only GO domain considered was Biological Process since it contains the stress terms. Then, the scores of all the GO terms of the same biomarker were combined using "max" method, which only keeps the maximum score. This method was selected as some biomarkers might be multifunctional, so the GO associated to non-stress functions would decrease their total score.

2.6. Optimization of the classifiers

As a first attempt, simulated annealing was utilized to optimize the parameters T of RF and C of SVM using the GO scoring as fitness function. The neighbors of each accepted state were the values ±10 positions away of it, the feature selection was executed 200 times and the most explanatory classifier on each execution was returned. The maximum temperature was set to 0.5, minimum temperature to 10^{-5} and the number of steps to 30. For T, the optimization was carried out between 5 and 50 trees (optimisation_rforest.py), whereas for C the range of values spanned from 0.1 to 4.6 in steps of 0.1 in order to discretize and equalize the search space of T (optimisation_SVM.py).

Then, a brute-force or exhaustive search algorithm was implemented and applied to obtain the best parameter using the same search spaces as in simulated annealing. This algorithm executes the 200 iterations of the feature selection routine three times per value of the target parameter and then it creates a boxplot showing the median and the two extreme values as the whiskers using the Python's package matplotlib (**Table 2**).

2.7. Genetic circuit modelling and synthesis

A genetic C-element was manually designed using the guidance of the majority gate circuit kindly provided Dr. Khomenko using Workcraft (Poliakov, et al., 2009) (**Fig. 2**). The translation of the majority gate into a genetic-implementable circuit was carried out at a transcriptional level harnessing the library of repressors orthogonal to TetR described to be functional in *E. coli* and mammal cells (Stanton, et al., 2014; Stanton, et al., 2014). This approach ensures that the C-element black box could be plugged to any input or output signal in *B. subtilis* as long as none of the repressors is present in the chassis. The repressors that had less toxicity and did not belong to *Bacillus* species were chosen.

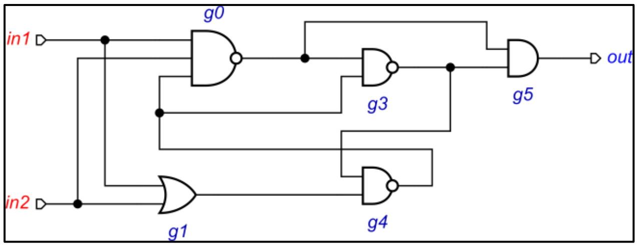

Fig. 2. Gate-level design of a majority gate C-element. Provided by Dr. Khomenko.

The majority gate had to be inverted in order to turn off the expression of the output as a consequence of the onset of both input signals. Then, the circuit was transformed into a combination of NAND, NOR and NOT gates so that only repressors are needed for its genetic implementation. NAND gates were designed using different transcriptional units with the same coding sequence but each of them regulated by one of the repressible promoters set as inputs of the gate. NOT gates were designed using a single transcription unit encoding a repressor and regulated by an input repressible promoter. Lastly, NOR gates utilised a single transcriptional unit in which the coding sequence was regulated by a constitutive promoter containing all the input repressible operators. CDSs that shared the same promoter were combined into a single transcriptional unit. The native Ribosome Binding Sequences (RBS) of each biomarker and the optimized RBS of the repressors (Stanton, et al., 2014) were used.

Once the abstract design of the genetic C-element was accomplished, it was converted into a Petri net and digitally simulated using Workcraft (Poliakov, et al., 2009). Finally, the genetic design was written using the Synthetic Biology Open Language (SBOL) 2.0 (Bartley, et al., 2015), a data standard developed to computationally exchange synthetic biology designs (celement folder with the Java script to generate SBOL celement.sbol). In this way, this black box implementation can be easily reusable and visualised using VisBOL (McLaughlin, et al., 2016), a platform for the graphical visualisation of genetic designs. VisBOL platform uses the set of glyphs defined by SBOL Visual (Quinn, et al., 2015) in order to standardise the representation of genetic circuit.

2.8. Experimental approach

Microarray and RNA-seq dataset contained samples corresponding to different stresses and growth condition. Samples not subjected to the stress under analysis were used as controls so as to ensure that the observed differences are specific to the target stress and are not part of the general stress response. For example, when the oxidative stress was the target stress, the samples treated with diamide, paraquat and H_2O_2 were tagged as STRESS samples, whereas the remaining 154 samples were considered controls.

3 Results

All the scrips, documentation and a tutorial of the biomarker retrieval algorithm are uploaded into https://deliacp@bitbucket.org/deliacp/scripts.git. The biomarker retrieval algorithm was implemented using different programming languages wrapped using bash script, the language that automates the execution of Linux shell commands (bash_file).

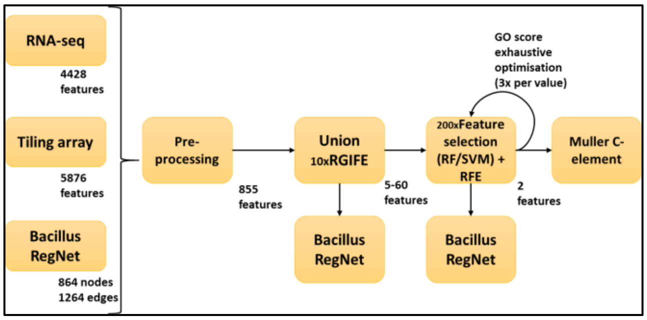

Fig. 3. Workflow of the stress-specific biomarker retrieval algorithm. A total of 855 features remained after the integration of the three original datasets. The expression data were pre-processed and set as input to 10 RGIFE runs. An exhaustive optimization of T and C was run before the execution of 200 repetitions of RF-RFE or SVM-RFE with the union of the features retrieved by RGIFE. The regulatory sequences of the top two features common to both routines were used as an input for a C-element circuit.

The general workflow of the developed tool (**Fig. 3** and **4**) integrates the entries of RNA-seq, microarray and BacillusRegNet data. As a result, the 855 features kept were contained in all the datasets. The integrated RNA-seq and microarray data were subjected to a pre-processing step so as to normalize and correct the batch effects. Then, 10 repetitions of RGIFE were executed with the expression data of these entries to obtain the set of features that are needed to distinguish between control and stress samples. To discern which feature was the most discriminative, a second feature selection was executed (RF-RFE or SVM-RFE) using the expression data corresponding to the features retrieved by RGIFE as input. On each step of RFE, the feature that had a smaller contribution to the result of the classifier is removed until only one was remaining. This feature was considered to be the one with the best discriminative power but the result may change for different executions of feature selection due to the stochasticity of the classifier. Consequently, SVM-RFE and RF-RFE were repeated 200 times to obtain the frequency of each feature being the most discriminative. This process was repeated three times per value of the target parameter during the exhaustive optimization. Finally, the regulatory sequences of the two features more frequently retrieved by both feature selection algorithms were used as inputs of a C-element able to ease the stress. During all the process, BacillusRegNet data was utilized to plot the genes retrieved by the different feature selection algorithms, highlighting regulatory cascades related to the target stress.

```
Merge RNA-seq, microarray and BacillusRegNet entries
Normalize, correct batch effects and label control and stress samples
for 1 to 10 {Run RGIFE}
Unite all the retrieved features
for T in 5:50/for C in 0.1:4.6 with steps of 0.1 {
    for 1 to 3 {
        repeat 200 times {(RF/SVM)-RFE with T/C}
        % times each biomarker is the most discriminative
        GO_score}}
Boxplot of the 3 GO scores per value of T/C
Select the parameter with the highest score and lowest variance
repeat 200 times {(RF/SVM)-RFE with the optimum T/C}
Use the two biomarkers with the greater % as inputs in the genetic circuit
```

Fig. 4. Sequence of eventes in the stress-specific biomarker retrieval algorithm. This process, except the last step, was executed through the attached bash_file file.

```
Convert gene symbols into locus tags using the key in MicroScope {
    Remove the entries with no locus tag
    Merge the entries with the same locus tag}
Remove entries of BacillusRegNet that are not in RNA-seq and microarray
Remove entries of RNA-seq that are not in BacillusRegNet and microarray
Remove entries of microarray that are not in RNA-seq and BacillusRegNet
Merge RNA-seq and microarray using locus tags as keys
```

Fig. 5. Workflow of the integration of RNA-seq, microarray and BacillusRegNet datasets. This step is contained in Tiling_RNA_final_arg.R

3.1. Integration of the RNA-seq, microarray and BacillusRegNet entries

3.1.1. Conversion gene symbols to locus tags

For this step it was taken into account that each locus can have several gene symbols (for example, the locus BSU0003 is named as both *rapA* and *yaaA*) and some gene symbols can refer to several CDSs in different loci (*ymfK* is encoded in both BSU16890 and BSU16900). Once the conversion was carried out, the locus tags were utilized to reduce the three datasets to the subset of 855 common entries. RNA-seq and microarray profiles were merged into the gene expression matrix and BacillusRegNet data was used to plot the retrieved biomarkers (**Fig. 5**).

3.2. Selection of the normalization method

Three different tests were carried out since the order in which normalization and batch correction are applied varies across literature (Sun, et al., 2011). It has to be noticed that the microarray dataset was composed by 169 samples, whereas there were only six RNA-seq samples. Therefore, the objective of this step was to select the approach able to integrate the RNA-seq samples across the microarray samples (this is, fewer clusters of RNA-seq-only samples after the application of Ward's clustering), able to return a more homogeneous distribution of the expression data within samples (this is, similar boxplots) and able to derive consistent biomarkers on both second feature selection routines using oxidative stress as the target stress. The approaches tested were:

(1) Min-max normalization, batch correction and QN (Batch_min-max_quantile.R)

The most sensible approach is to apply min-max normalization and then correct the batch effects using the rescaled expression values. Then, QN would ensure a similar distribution of the expression values. After this pre-processing scheme was applied all RNA-seq samples were clustered together, making it unsuccessful (data not shown). This was assumed to be related to the application of a normalization step prior to the batch correction, which masked the differences between RNA-seq and microarray samples. However, the boxplots were similarly distributed across samples: they had a normal distribution with a median of 0.5, as corresponds to the application of min-max and QN. The rest of the biomarker retrieval algorithm was executed for oxidative stress with this normalization routine and the biomarker retrieved in 100% of the SVM-RFE iterations was *trxA*, which was also retrieved in 4% of the executions of RF-RFE.

(2) Batch correction, min-max normalization and QN (Min-max_batch_quantile.R)

In order to improve the results of (1), the order of the batch correction and min-max normalization was inverted. As a result, there were only two clusters with only RNA-seq samples, one composed by two RNA-seq oxidative stress samples and another by two RNA-seq control samples. Even though these clusters did not contain any microarray sample, they did not grouped control and stress samples together. The boxplots obtained were equal for all the samples but they are skewed towards low expression values. This result confirmed that the integration performance is better when the batch correction is carried out before the normalization, although the homogeneity diminishes. This indicates that the order of application of the batch correction imposes a trade-off between the correction of the inter-sample variability and the homogenization of the intra-sample distribution of the expression data. When the rest of the biomarker retrieval algorithm was executed with this normalization routine, the biomarkers retrieved by RF-RFE and SVM-RFE did not match. SVM-RFE retrieved *gltA* in 100% of the iterations, whereas this gene was only returned in 0.5% of the executions of RF-RFE. This lack of consistency between feature selection algorithms was also observed in (1), which points out to the pre-processing masking the biological differences between genes' expression. As a result, small differences in the features selection algorithms can lead to entirely different features selected.

(3) Batch correction and min-max normalization (Batch_min-max.R)

A simpler version of the previous pre-processing schemes was applied to test if the lack of concordance between SVM-RFE and RF-RFE was due to an excessive modification of the expression data during the pre-processing. In this new trial the QN step was omitted. The hierarchical clustering returned two RNA-seq stress and one RNA-seq control samples clustered together. Furthermore, the boxplots showed that the data is skewed towards low values of expression and there were great differences in the distribution of the expression values between RNA-seq and microarray samples (data not shown). Even though the results of this pre-processing scheme did not seem promising, the biomarkers retrieved by SVM-RFE and RF-RFE for oxidative stress were similar: *manA* was returned in 100% of the iterations of SVM-RFE and in 29.5% of the iterations of RF-RFE. As a result, this last pre-processing scheme, consisting on batch-correction and min-max normalization, was selected due to the consistency of the biomarkers retrieved and its low computational cost.

We can measure the goodness of the pre-processing ranking from one to three the pre-processing schemes according to the degree of normalization and integration of samples, where one is the lowest score (**Table 3**). Taking into account the degree of concordance of RF-RFE and SVM-RFE, it was shown that the score for the pre-processing is inversely proportional to the classifiers' agreement. These results suggest a correlation between the degree of pre-processing and the concordance of the biomarkers retrieved by the two feature selection procedures: methods that are able to thoroughly integrate and normalize RNA-seq and microarray samples did not provide consistent biomarkers, probably because the data had been extremely processed, increasing the number of artefacts and leading to a lack of robustness in subsequent processes.

Table 3. Scoring of the pre-processing schemes according to the normality of their final boxplots, the integration of the RNA-seq samples and the concordance of the biomarkers retrieved by RF-RFE and SVM-RFE.

Pre-processing procedure	**min-max + batch + QN**	**batch + min-max + QN**	**batch + min-max**
Normality	3	2	1
Integration	1	3	2
TOTAL	4	5	3
Concordance	2	1	3

3.3. Feature selection algorithms

After the execution of 10 iterations of RGIFE, all biomarkers retrieved were subjected to one more step of feature selection so as to obtain the gene whose expression is a better predictor of stress. RGIFE used a 10-fold cross-validation scheme, where samples are divided into 10 subsamples, of which one is used as validation dataset and the remaining as training dataset for building a RF classifier. In this way, RGIFE can measure the accuracy of the classification using the validation subset so as to retrieve the minimum number of features able to train a maximum accuracy classifier. When the second feature selection algorithm is applied, the less discriminative features are removed one by one so that the accuracy of the classifier is always going to decrease. For this reason, a cross-validation scheme was not included in the second step of feature selection.

Among the multiple options of feature selection algorithms, RFE was selected as it is an embedded method specifically designed for the analysis of microarray experiments (Guyon, et al., 2002). RFE is typically coupled with SVM, whose performance has demonstrated to outperform other classifiers (Bolon-Canedo, et al., 2014; Guyon, et al., 2002). Moreover, RFE was executed with another classifier so as to be able to compare results. RF was selected as this second classifier as it outperformed SVM in terms of overfitting and accuracy in metabolic data analysis for biomarker selection (Chen, et al., 2013). The RFE algorithm implemented (**Fig. 6**) contained as ranking criterion *gini* impurity index for RF and the weight of each feature for selecting the support vectors in SVM.

```
Load expression matrix ($E^n$), where n ∈ [RGIFE-retrieved features]
run 200 times {
    i = n
    while length(i) ≠ 1 {
        Train a classifier with $E^i$                          |
        Rank(i)                                                |  RFE
        Remove the least important feature (i = i - 1)}        |
    Add i to the set of biomarkers}
return (frequency of biomarkers)
```

Fig. 6. Scheme followed by RF-RFE and SVM-RFE.

3.4. Optimization of the classifiers

The parameters T and C did not have recommended values in the reviewed literature even though they have a great impact in the result of their respective classifier; consequently, they were subjected to optimization using a heuristic and an exhaustive method. T determines the number of bootstrap samples or trees that are taken into account for building the RF. For high values of T, RF classifications would converge to the same solution (Breiman, 2001), but their computational cost impedes their usage. On the other hand, parameter C reflects the trade-off between the maximization of the margin and the error in the classification of the training sample in SVM. It is recommended to use a low C for noisy data as it returns more robust results, or to increase it for retrieving more highly weighted biomarkers (Pedregosa, et al., 2011).

3.4.1.Fitness function: GO score

The fitness function to be optimized was the GO score, i.e., the extent to which the retrieved biomarkers are related to a stress process. This score is provided by the semantic similarity of the biomarkers' GO terms to the stress term GO:0006950 and its child terms. A drawback of this fitness function is that it would prevent the optimized feature selection from returning uncharacterized genes.

$$Score_{RFE} = \sum_i GOscore_i \frac{frequency_i}{100} \tag{2}$$

Where RFE is either RF-RFE or SVM-RFE, i is each biomarker retrieved by 200 iterations of RF-RFE or SVM-RFE, GO score is the score of the biomarker i and frequency is the percentage of the executions in which the biomarker i is the most discriminative.

Firstly, the GO terms of the 4197 proteins in *Bacillus subtilis subsp. subtilis str. 168* reference proteome (UniProt proteome ID UP000001570) were retrieved (locus_go.py). The resulting tab separated file was used as a library for mapping the locus of each feature returned after the 200 iterations of the RFE routines to its GO terms. Then, the function mgoSim (Yu, et al., 2010) was used to calculate the semantic similarity of each biomarker and the stress GOs. The percentage of times each biomarker was the most discriminative was used to weight its score and, finally, obtain the GO score (2).

3.4.2.Simulated annealing

The parameter T takes integer values and both RF and SVM make use of pseudorandom number generators. Therefore, the solution of the fitness function changes in each execution for the same value of the parameter.

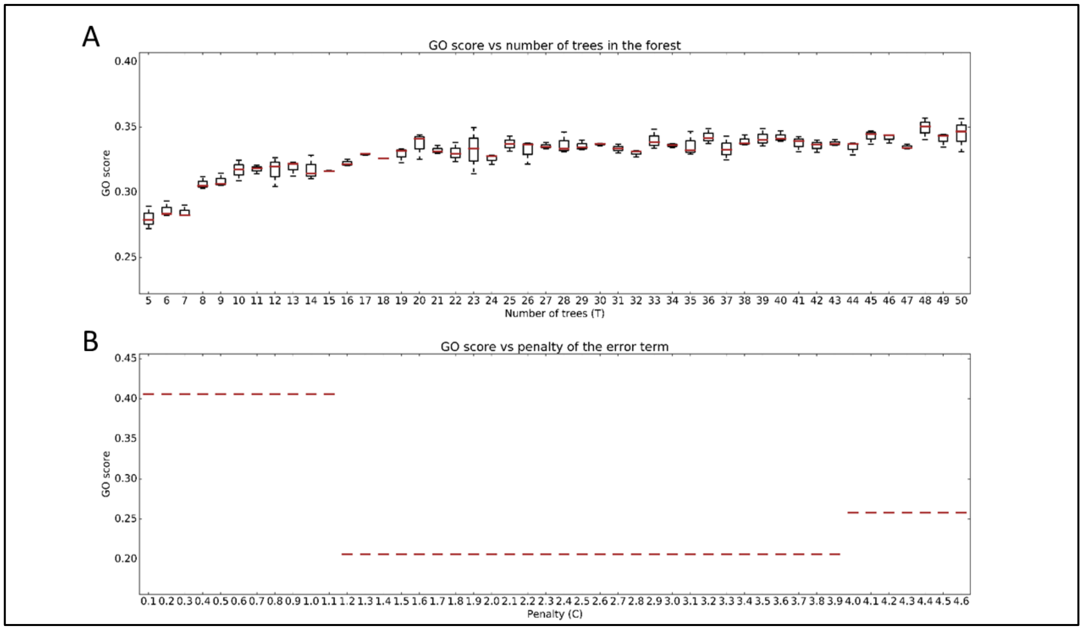

Fig. 7. Boxplot of the optimization of T in RF-RFE (A) and C in SVM-RFE (B) for oxidative stress. Y-axis contains the score obtained by the biomarkers in the three executions of the 200 iterations of the feature selection algorithm per value of the parameter, contained in x-axis.

As a result, a stochastic combinatorial optimization was needed, for which simulated annealing was selected as it is the most adequate heuristic for combinatorial optimization (Perry and Wagner, 2014). This heuristic, executed through simanneal2 package (**Table 2**), was used to obtain the value of T and C able to retrieve the oxidative stress biomarkers with a greater GO score (**Fig. 8**). The search space of C was discretized so that its search space is the same as T's.

```
P = [p1, p2, …, p46], where p is the value of the parameter; Tmax = 0.5; Tmin=$10^{-5}$; steps = 30
Initial conditions: new_state = random(P), T=Tmax
from 1 to steps{
    current_state = new_state
    classifier_RFE (200 iterations, parameter = current_state)
    GO_score (classifier_RFE)

    if not (GO_score < GO_score_accepted) AND
                ( $\frac{GO_{score\_accepted} - GO_{score}}{T}$ > random([0,1]))
        {accepted_state = current_state}
    $T = T_{max} * e^{\left(\frac{-\ln\left(\frac{T_{max}}{T_{min}}\right)*step}{steps}\right)}$

    new_state = accepted_state ±10}
return saved_state for maximum GO_score
```

Fig. 8. Scheme of simulated annealing optimization implemented using simanneal2 package (Perry and Wagner, 2014). Classifier referes to either RF or SVM and parameter is T in RF and C in SVM.

After 30 iterations of RF-RFE and SVM-RFE, simulated annealing was not able to converge to any good solution (data not shown). This failure occurred even with a higher maximum temperature and more iterations. A possible reason is that the heuristic failed to explore the search space of the parameters since it stayed in values close to the random initial state, even when the space for neighbors' selection was incremented. Another issue was the stochasticity of the classifiers, which prevented GO scores from being consistent. Accordingly, the heuristic strategy was discarded due to the lack of consistency of the resulting optimum parameter in different executions.

3.4.2.Exhaustive search

The variability in the classifier made it possible to apply descriptive statistics to the GO scores obtained after three executions of the feature selection with the same parameter (**Fig. 7**). Because of T being a discrete parameter and the size of search space of C being equalized to T's, the search space was limited to 46 values, which made it feasible to repeat the classifier several times per value. As a consequence, an exhaustive search optimization was applied to T and C (**Fig. 9**). The smaller value of the parameter able to compute a high GO score with a small variability was used to execute again 1 repetition of the 200 iterations of the classifier-RFE and obtain the ranking of features.

```
Initial conditions: P = [p1, p2, …, p46]
for p in P {
        repeat 3 times {
                Classifier-RFE (200 iterations, parameter = p)
                GO_score (classifier_RFE)}
        boxplot showing the median and standard deviation}
```

Fig. 9. Scheme for exhaustive optimisation, where P is the search space of either T or C and classifier is either RF or SVM.

3.5. Application of the biomarker retrieval algorithm

The biomarker retrieval algorithm was applied to oxidative and heat stresses. Since different oxidative stressors can cause different responses, subsequent analysis targeted the stress caused by each individual oxidative agent to check the dependence of the oxidative response upon the oxidative agent employed.

3.5.1.Oxidative stress

After the pre-processing, the 21 oxidative stress samples were tagged as STRESS and the remaining 154 samples as CONTROL. A Student's t-test was applied and the p-values associated to each gene's expression were obtained. These p-values were used to execute jActiveModules for BacillusRegNet entries.

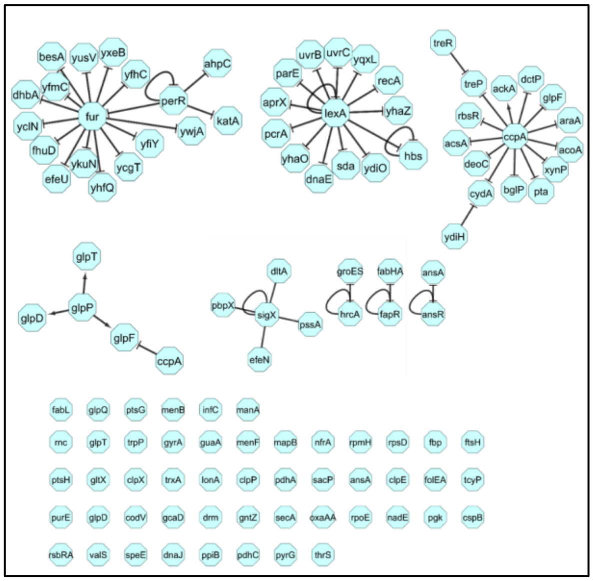

Fig. 10. **Differentially expressed regulatory modules in samples subjected to oxidative stress** as retrieved by jActiveNetworks. SigA has been removed for simplicity.

Table 4. Oxidative stress biomarkers after 200 iterations of RF-RFE (T=48) and SVM-RFE (C=0.1). It shows the percentage of times each biomarker is returned as the most explanatory, the p-values after a two-tailed Student's t-test and the difference between the average expression values of control and stress samples.

CDS	Description	RF-RFE (%)	SVM-RFE (%)	p-value	Control-stress
manA	Mannose 6-P isomerase	41.0	100	10^{-6}	-0.257
yxeB	Iron binding protein	23.5		10^{-4}	-0.090
gltA	Glutamate synthase	21.5		0.775	-0.017
treP	Trehalose transporter	12.5		10^{-5}	-0.238
clpP	Protease	1.0		10^{-2}	-0.136
fbp	Fructose 1,6-bisphosphatase	0.5		10^{-2}	-0.131

The differentially expressed subnetworks contained genes previously described as participants in the oxidative stress response (**Fig. 10**) (Helmann, 2016; Mols and Abee, 2011; Zuber, 2009). The main coordinators of these subnetworks were the transcription factors Fur, LexA, CcpA and GlpP. Fur represses the expression of proteins involved in the uptake of iron in the presence of this metal (Zuber, 2009) and LexA represses the genes that responds to DNA damage (Mols and Abee, 2011). CcpA and GlpP are regulators of the carbon metabolism in *B. subtilis*: CcpA is the main coordinator of the glucose-mediated catabolite repression (Wacker, et al., 2003), whereas GlpP is involved in the transcription of the genes responsible for the uptake and degradation of glycerol (Lewin, et al., 2009). The last hub was SigX, an extracytoplasmic function sigma factor involved in cell envelope homeostasis whose mutant versions cause hypersensitivity to oxidative and heat stresses (Helmann, 2016).

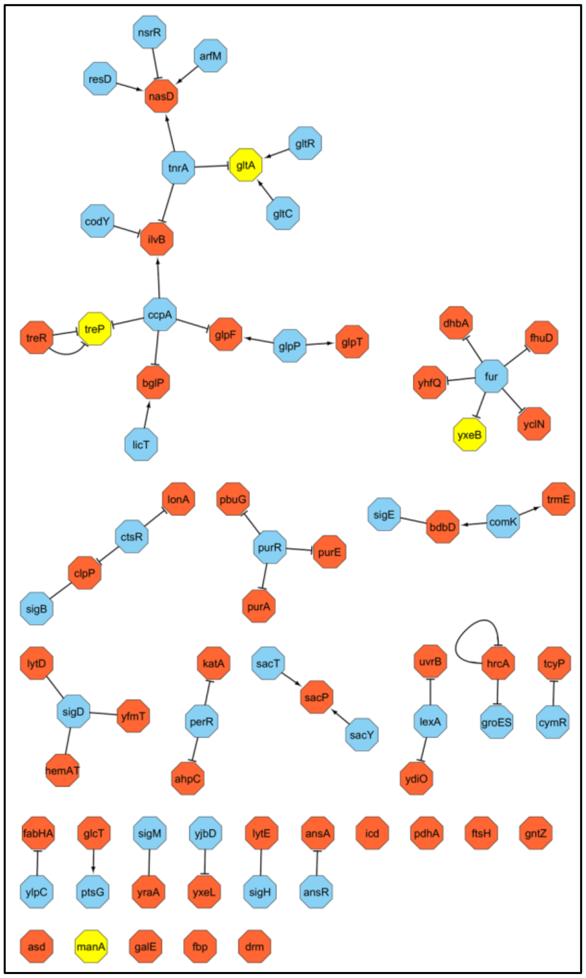

Fig. 11. First degree in and out interactions of the genes found by RGIFE (red and orange nodes) for oxidative stress as contained in BacillusRegNet data. The biomarkers obtained as the most explanatory in more than 10% of the executions of RF-RFE and SVM-RFE are represented as yellow nodes.

The 10 repetitions of RGIFE returned a total of 45 genes needed to distinguish between oxidative stress and control samples with the maximum accuracy. Plotting the interactions of these genes using BacillusRegNet data, a pattern of alternate RGIFE biomarker and non-biomarker was revealed (**Fig. 11**). Biomarkers are typically the target of the interaction, whereas non-biomarkers are the transcription factors that enable them to have a different expression profile in the stress conditions.

The values selected for T and C after the exhaustive optimization were 48 and 0.1, respectively (**Fig. 7**). These values were applied to RF-RFE and SVM-RFE resulting in *manA*, *yxeB*, *gltA* and *treP* being the genes with the largest discriminative power, all of them significantly upregulated except *gltA* ($p > 0.05$) (**Table 4)**. Of them, *yxeB*, *manA* and *treP* are also contained in a differentially expressed module (**Fig. 10**). Looking at their interactions, *gltA* and *treP* participate in the same regulatory network but they are regulated by different transcription factors, *yxeB* is regulated by Fur and *manA* does not have any known interaction with transcription factors (**Fig. 11**).

Regarding their functions, TreP is involved in the uptake of trehalose and it is repressed by CcpA and TreR, a repressor of the trehalose utilization operon in absence of this disaccharide (**Fig. 11**) (Rezacova, et al., 2007; Wacker, et al., 2003). GltA is part of the glutamate synthase complex, the only enzyme able to synthesize glutamate in *B. subtilis* (Wacker, et al., 2003). *yxeB* encodes a ferrioxamine-binding protein that participates in the uptake of this iron-coordinated complex (Ollinger, et al., 2006). Lastly, *manA* encodes a mannose 6-phosphate isomerase essential for the catabolism of mannose (Sun and Altenbuchner, 2010).

3.5.2.Diamide, H_2O_2 and paraquat biomarkers

The high number of biomarkers needed to distinguish oxidative stress samples pointed out to distinct responses to different stressors. To check this hypothesis, the same analysis was executed for each oxidizing agent individually. RGIFE retrieved a total of 10 features for diamide-treated samples (12 samples), 4 for H_2O_2 (6 samples) and 2 for paraquat (3 samples). These preliminary results show that the number of biomarkers retrieved by RGIFE increases with the number of stress samples, which may be due to the inter-sample variability not being totally corrected during the normalization.

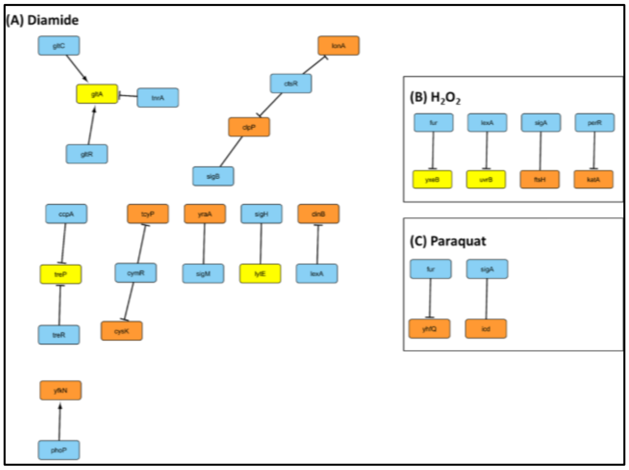

Fig. 12. First degree in and out interactions of the genes found by RGIFE (orange and yellow nodes) for diamide (A), H_2O_2 (B) and paraquat (C) stresses as contained in BacillusRegNet dataset. Biomarkers obtained as the most explanatory in more than 10% of the executions of RF-RFE and SVM-RFE are represented as yellow nodes.

After the application of RF-RFE and SVM-RFE with the optimum parameters, the genes whose expression is determinant of treatment with diamide were *gltA*, *lytE* and *treP* (**Table 5, Fig. 12A**). *gltA* and *treP* are common to the oxidative stress response but in the treatment with diamide *gltA* is downregulated, whereas *treP* upregulation is kept (**Table 5**). *lytE*, downregulated during treatment with diamide ($p<0.01$), encodes an enzyme for the degradation of the peptidoglycan (Kasahara, et al., 2016).

The biomarkers obtained for H_2O_2 were *yxeB* and *uvrB* (**Table 5**), both significantly upregulated ($p<0.05$). *yxeB* was also found as an overexpressed biomarker in the oxidative stress response (**Table 4**). *uvrB* was retrieved by both RF-RFE and SVM-RFE and it encodes the DNA damage recognition subunit of UvrABX complex.

The samples treated with paraquat are only characterized by one biomarker, *icd* (**Table 5**). *icd*, downregulated in stress samples ($p<0.01$), encodes the isocitrate dehydrogenase that catalyzes the oxidation of isocitrate to α-ketoglutarate coupled with the reduction of $NADP^+$ to NADPH (Kim and Colman, 2005).

Table 5. Diamide, H_2O_2 and paraquat stress biomarkers after 200 iterations of RF-RFE and SVM-RFE. It displays the percentage of times each biomarker is returned as the most explanatory, the p-values after a two-tailed Student's t-test and the difference between the average expression values of control and stress samples.

Diamide (T=5, C=0.1)					
Symbol	**Description**	**RF-RFE (%)**	**SVM-RFE (%)**	**p-value**	**Control-stress**
gltA	Glutamate synthase	49		10^{-27}	0.0819
lytE	Peptidoglycan endopeptidase	25.5		10^{-4}	0.0421
treP	Trehalose transporter subunit	12.5	100	10^{-13}	-0.1837
clpP	Proteolytic subunit	5		10^{-5}	-0.0649
dinB	Protein DinB	3.5		10^{-4}	0.0208
lonA	Lon protease 1	2.5		10^{-4}	-0.0574
yfkN	Nucleotide phosphoesterase	1		10^{-14}	0.0724
yrqA	Cysteine protease	0.5		10^{-3}	-0.0536
cysK	Cysteine synthase	0.5		10^{-3}	-0.0298
H_2O_2 (T=11, C=0.1)					
yxeB	Iron(3+) binding protein	50.5		0.0423	-0.382
uvrB	UvrABC system protein B	33.5	100	0.0369	-0.393
ftsH	Metalloprotease	9		0.0454	-0.246
katA	Catalase	7		0.0135	-0.449
Paraquat (T=50, C=0.1)					
icd	Isocitrate dehydrogenase	100	100	10^{-7}	0.109

3.5.3.Heat stress

The same analysis was conducted for heat stress so as to double-check the stress-specific biomarker retrieval algorithm. A total of six microarray samples cultured at either 48ºC or 51ºC were tagged as STRESS and RGIFE returned two features, the chaperone *htpG* and the benzoate dehydrogenase *dhbA*. Finally, the most predictive gene for heat stress was *htpG*, returned 100% of the times in 200 iterations of both RF-RFE (T=6) and SVM-RFE (C=0.3).

3.6. Genetic C-element

The abstract model of an inverted genetic C-element was implemented based on a majority gate (**Fig. 14**). The inputs of the circuit were the promoters repressed by BetI and LitR and the output was the CDS of the transcriptional repressor PsrA, which represses the expression of the target CDS (VioB in the example, **Fig. 14**).

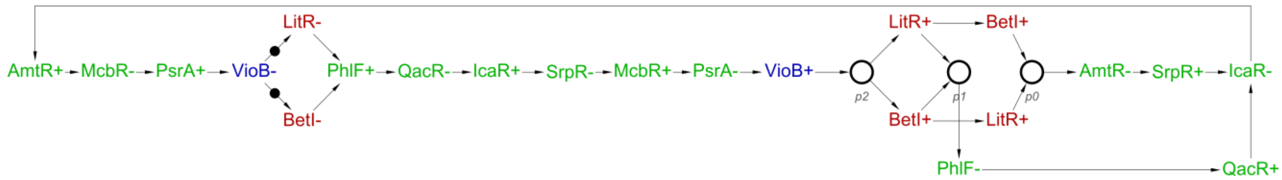

Fig. 13. Petri net model of the inverted C-element. The different places indicate the activation (+) or repression (-) of the expression of each CDS. The input signals (red), switch on or off the expression of the transcription factors used as internal signals (green), leading to the expression or inhibition of the target CDS (VioB). Transitions between places are displayed by arrows. A transition is active when a black dot is shown on it. A place triggered by two different transitions needs both active in order to progress towards it. When an active transition leads to two different states or when two places trigger a transition independently, the dot is enclosed in a circle.

This design can be utilised as a black box to be coupled to any two input signals that activates the expression of *betI* and *litR*. In the same way, the output of the circuit could be any CDS regulated by the promoter repressed by PsrA. This circuit was transformed into a Petri net (**Fig. 13**) and a digital simulation was carried out (**Fig. 15**). It could be observed that the expression of *vioB* is repressed when both BetI and LitR are present. Under this circumstances, the presence of BetI activates the expression of *qacR*, while, in turn, the presence of LitR activates the expression of *srpR*. QacR and SrpR are both needed to switch on *icaR* to activate *vioB*. On the other hand, the absence of both BetI and LitR is needed to sequentially inhibit *qacR*, activate *icaR* and inhibit *srpR*. Only after this sequence of events the absence of SrpR switches on the expression of *vioB*. It was confirmed that the state of *vioB* does not change when only one input is present (**Fig. 15**).

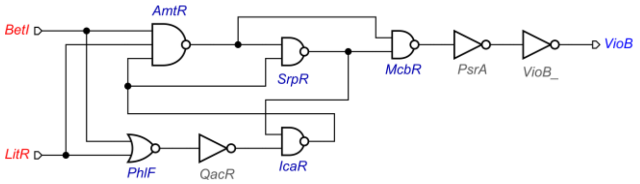

Fig. 14. Logic-level design of an inverted genetic C-element circuit. The inputs of the circuit are the promoters repressed by BetI and LitR, whereas the output is PsrA, controlling the expression of *vioB* in the example.

Once the mechanism of the inverted C-element was checked, it was utilized to cope with oxidative stress. The promoters of the two highest scoring oxidative stress biomarkers, *manA* and *yxeB*, were used as inputs of a C-element circuit that will result in the activation or repression of the expression *vioB* so as to ease the stress. *vioB* is a gene of 2997bp that encodes the enzyme VioB from *Chromobacterium violaceum* (Balibar and Walsh, 2006). VioB is not functional in *B. subtilis*, but it is able to subject cells to a high metabolic burden due to its large size (Smith, *et al*., unpublished). *manA* and *yxeB* are overexpresses under oxidative stress and, therefore, their promoters would induce the expression of *betI* and *litR*, respectively. However, for other stresses, if any of the biomarker genes is downregulated, another NOT gate is needed upstream of *betI* or *litR*.

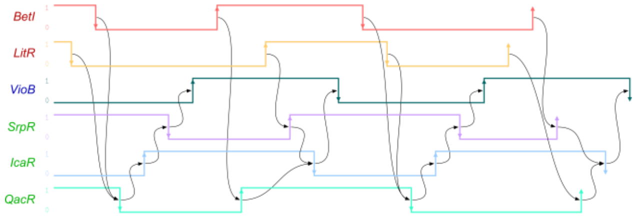

Fig. 15. Digital simulation of the inverted C-element. The transitions of the two inputs (BetI and LitR) trigger the sequence of transitions of the internal signals IcaR, SrpR and QuaR that switches on or off the expression of *vioB*.

This design was written in SBOL 2.0 using the sequences of the elements as `DNAcomponent` and defining the transcription repressions as `PromoterRepression` (**Fig. 16**). The promoter of *manA* was obtained from the promoter of the operon *manPA-yjdF*. Its sequence spans from the previous CDS (*manR*) to the last non-transcribed nucleotide, 42bp upstream of ATG (Sun and Altenbuchner, 2010). The RBS of *manA* was taken from its 5' UTR. The promoter of *yxeB* was selected as the 94bp from 184bp to 90bp upstream of the initiation of the translation. This region contains the binding site of Fur and a multiple sequence alignment with Clustal Omega against several prokaryotic promoters predicted that it contains the RNAP binding site. The remaining 90bp upstream *yxeB* were considered as *yxeB*'s RBS. The weak and constitutive promoter of the *liaG* gene was used for the implementation of the NOR gate. The fluorescent reporters *rfp*, *cfp* and *gfp*, whose emission and excitation wavelengths do not overlap, were fused to *betI*, *litR* and *vioB* to track the activation and repression of these transcriptional units.

4 Discussion

This work describes the implementation and usage of a stress specific biomarker retrieval algorithm. The folder of its source code contains a user guide with a description of its different options as well as a tutorial that ensures its reproducibility (README). The retrieved biomarkers were plugged to an inverted C-element. Although in this project the process is mainly applied to oxidative stress as a proof of concept, the objective will be to use this algorithm and the genetic circuit to address metabolic stress in synthetic strains. The overproduction of heterologous proteins lessens the amount of energy and biomolecules available for host's housekeeping functions, limiting the growth rate, which leads to metabolic stress and the subsequent loss of productivity (Carneiro, et al., 2013). Once the metabolic load samples are obtained from cultures expressing the burdensome protein VioB (Smith, *et al.*, unpublished), the same process described here will be followed and the resulting genetic circuit with the corresponding biomarkers will be tested *in vivo*.

A drawback of applying this approach to metabolic load is that the maintenance and expression of the elements of the circuit may cause as much metabolic load as *vioB*, so the expression of *vioB* would always be off. If this occurs the regulatory sequences of the biomarkers used as inputs of the circuit should be modify to raise the maximum intensity of the signal that keeps the system off, so the circuit will only respond to greater doses of stress. This task can be achieved by directed evolution of the input promoters. Nevertheless, the size of the circuit should not be a problem to the cells as the industrial overproducers can support large fragments of synthetic DNA, usually containing several copies of the heterologous CDS. Moreover, if the circuit is functional, the overproducer strains bearing the C-element will have an adaptive advantage over the cells not able to turn down the production of the burdensome gene and they will be more prone to survival and fast growth.

TetR repressor and its orthologous are inhibited by tetracycline, a drug commonly used for the screening of mutants and expression induction in recombinant bacteria. This means that the designed circuit can be switched off adding this compound, so that strains based on tetracycline are not compatible with the current implementation of the C-element. Nevertheless, it is important to notice that tetracycline is the only compound commonly used in genetic engineering that affects the functioning of the circuit and that the circuit does not need the addition of external, relatively costly substances such as IPTG or arabinose. During the digital simulation of the genetic circuit it was observed that the sequence of internal changes that leads to a change in the state of *vioB* differs between activation and inhibition as is characteristic of sequential logic, where different sequence of signals may trigger a different output (**Fig. 15**) (Lou, et al., 2010).

Focusing now on the pre-processing, the results suggest that the number of biomarkers increases with the heterogeneity of the samples set as STRESS. The more samples and treatments, the more biomarkers are retrieved, as corresponds to a more diverse cellular response to a broader range of threatens. This effect allows to uncover regulatory patterns that may not be seen when a smaller array of samples is screened.

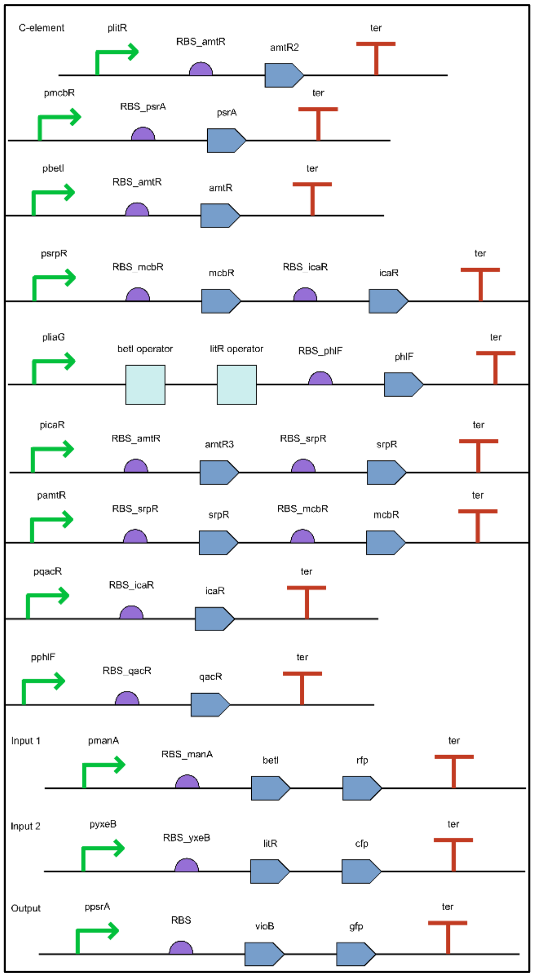

Fig. 16. Genetic design for an inverted C-element able to respond to oxidative stress using VisBOL (McLaughlin, et al., 2016). It modifies the expression of *vioB* (Output) as a function of the oxidative stress biomarkers *manA* and *yxeB* (Input 1 and 2). Arrows represent promoters, semicircles RBS, boxes operators, polygons CDSs and T terminators.

As far as the feature selection algorithms are concerned, SVM-RFE retrieved always the same biomarker in all the iterations, whereas RF-RFE had more varied results (**Table 4** and **5**). This is due to a smaller stochasticity of SVM as compared to RF since pseudorandom numbers are only applied for shuffling the features before selecting them for fitting the model. This characteristic makes SVM-RFE's GO scores larger, as there are not non-stress related biomarkers that diminish the score, but prevents this feature selection routine from showing the contribution of other features. This lack of variability added up to the overweighting of experimental artifacts that arises from SVM's overfitted models (Bolon-Canedo, et al., 2014), makes SVM prone to retrieve features that account for the noise and not for changes in the expression data. On the other hand, the stochasticity of RF that originates from the random selection of both the features per split and the bootstrap subset, makes it preferable due to its ability to escape from experimental artifacts. Actually, it has been described that RF-RFE overperforms SVM-RFE in finding small sets of discriminative features (Granitto, et al., 2006). In conclusion, RF-RFE is preferred as a second feature selection algorithm due to its ability find alternative biomarkers, which allows it to prevent overfitting.

Once the feature selection was applied, the GRN of the biomarkers showed that transcription factors were not identified as features able to distinguish stress samples from controls (**Fig. 11** and **12**). This result might be related to the differences in gene expression intensity having been set as distinction criterion. Transcription factors react to regulatory cascades with PTMs that modify their activity (Filtz, et al., 2014); hence, their expression intensity does not usually need to vary to respond to changes in the environment. They, in turn, modify the expression of the proteins that allow *B. subtilis* to adapt to the harsh conditions.

These genes were *treP*, *gltA*, *yxeB* and *manA* in oxidative stress (**Table 4**). The overexpression of *treP*, encoding a transporter of trehalose (**Table 4**), suggests that *B. subtilis* requires a higher rate of uptake of trehalose. This could be due to a greater demand of carbon for catabolism and anabolism in order to deal with ROS disabling cellular structures and scavenging electrons from the oxidative metabolism, decreasing its efficiency. Another possibility is a need to increment the pool of osmoprotectants in the cytosol as typically occurs during desiccation. The mechanism of overexpression of *treP* may be related to a possible inactivity of the catabolite repression towards trehalose regulated by CcpA (**Fig. 11**). This idea is compatible with the result of jActiveModules, which contained *treP* and other genes regulated by CcpA as part of one of the differentially expressed modules (**Fig. 10**), and it is supported by the fact that *E. coli* increases the metabolic flux through pathways alternative to glycolysis during oxidative stress (Rui, et al., 2010). An inactivation of TreP is discarded as the stress samples were not cultured in the presence of trehalose.

On the other hand, *gltA*, encoding a subunit of the glutamate synthase, is downregulated. Its transcription is repressed by TnrA in the absence of glutamine and ammonium (Wacker, et al., 2003). Glutamate synthase is the main link between carbon and nitrogen metabolism since it catalyzes the conversion of α-ketoglutarate, an intermediate of Krebs' cycle, and glutamine into glutamate (Wacker, et al., 2003). The expression of *gltA* is also induced by carbohydrates that increase the pool of α-ketoglutarate via GltC and GltR (**Fig. 11**) (Picossi, et al., 2007; Wacker, et al., 2003). Transcription regulators of the same family as GltC and GltR are involved in zinc homeostasis and oxidative stress in *Caulinobacter crescentus* (Braz, et al., 2010). The upregulation of *gltA* may indicate that *B. subtilis* is depleting carbon from Krebs cycle towards the synthesis of glutamate. However, the overexpression of *gltA* is not statistically significant so this biomarker could be a false positive.

yxeB, encoding a ferrioxamine chelant, is significantly overexpressed in oxidative stress samples, which points out to an inactivation of its repressor, Fur (**Fig. 10** and **11**) (Ollinger, et al., 2006). This is consistent with ROS oxidizing Fe^{2+} and preventing it from activating Fur (Zuber, 2009). In fact, both *fur* and *yxeB* are contained in a differentially expressed module (**Fig. 10**).

Lastly, *manA* is the most discriminative biomarker of oxidative stress since it is retrieved in most of the iterations in both RF-RGIFE and SVM-RFE. It is significantly overexpressed ($p < 0.05$) in the stress samples and it was also found by jActiveModules (**Table 4**, **Fig. 10**). *manA* encodes the enzyme that transforms mannose 6-phosphate into fructose 6-phosphate, an intermediate of the glycolysis also used as osmoprotectants for retaining water during desiccation stress. The overexpression of *manA* may indicate a possible need of more energy or building blocks for adapting to oxidative stress, but also a need for more osmoprotectants, as it may be the case of *treP*. The increase of ManA and TreP, both involved in boosting the pool of compatible solutes, may be explained by the control dataset not containing samples subjected to desiccation. The effectivity of the biomarker retrieval algorithm relies upon the range of conditions included in the set of samples so that false positive biomarkers are more likely to be avoided if a broader range of stresses is included as CONTROL samples. Whether *manA* and *treP* are oxidative stress specific biomarkers needs to be confirmed including desiccation samples in the analysis.

In contrast to the oxidative stress response, in the diamide analysis *gltA* is significantly downregulated ($p < 0.01$) (**Table 5**), indicating that cells are committing carbon to Krebs cycle in detriment of the synthesis of glutamate. Two possibilities arise regarding the destination of this extra carbon: its catabolism so as to obtain the energy and reductive power, or an increase in the pool of α-ketoglutarate. This last possibility has been previously described in *E. coli* and *Pseudomonas fluorescens,* where α-ketoglutarate is accumulated under oxidative stress conditions because of its consumption of H_2O_2 to form succinate in a non-enzymatic oxidative decarboxylation (Mailloux, et al., 2009; Rui, et al., 2010).

The second most discriminative biomarker for diamide stress was *lytE* (**Table 5**), which encodes the enzyme responsible for the degradation of the peptidoglycan of the cell wall during cell elongation (Kasahara, et al., 2016). It has been observed that certain bactericidal proteins, such as PGRPs (mammalian Peptidoglycan Recognition Protein), are able to bind disaccharide-pentapeptides in Gram+ bacteria so as to induce oxidative, thiol and metal stress responses (Kashyap, et al., 2014). The downregulation of *lytE* during oxidative and thiol stresses induced by diamide could be due to an endogenous cellular response triggered by the presence of oxidative and thiol damage. *B. subtilis* would interpret these damages as a possible attack targeting products of the degradation of peptidoglycan and it would decrease their presence downregulating *lytE.*

It is interesting to point out that the features *lonA* and *clpP*, overexpressed during diamide stress and retrieved with a low explanatory power (**Table 5**), are regulated by the transcriptional repressor CtsR (**Fig. 12A**), a central stress regulator that is targeted for degradation by McsB upon oxidative and heat stresses (Stannek, et al., 2015). ClpP is part of a protease complex that degrades misfolded proteins, while LonA is a conserved protease that targets proteins damaged by oxidation in the mitochondrial matrix of eukaryotes (Pinti, et al., 2015; Stannek, et al., 2015). As expected by a similar transcriptional regulation, they increase their production in the same order of magnitude (0.065 units for *clpP* and 0.057 for *lonA*) (**Table 5**).

All data provided suggests that the specific diamide response involves the expression of proteases that degrade misfolded proteins, the stabilization of the peptidoglycan in the cell wall and the redirection of the carbon metabolism towards the synthesis of the antioxidant α-ketoglutarate. This response is radically different to the one of H_2O_2 and paraquat. During H_2O_2 stress, the biomarkers *uvrB* and *yxeB* are overexpressed (**Table 5**). UvrABX is in charge of the NER (Nucleotide Excision Repair), which detects and replaces nucleotides in bulky DNA lesions (Waters, et al., 2006). Oxidative stress and specially H_2O_2 is known to cause DNA damage (Imlay, 2015; Zuber, 2009), which is recognized and targeted to repair by UvrB. On the other hand, *icd* is downregulated in the samples treated with paraquat, a mechanism previously observed in *E. coli*, where this herbicide induces the production of acetate. In this organism, acetate inactivates Icd and turns down the production of NADPH in favor of NADPH, diminish the amount of ROS produced in the electron transport chain (Rui, et al., 2010).

All in all, the biomarkers of oxidative stress depend on the stressors used to induce it. When all the stress samples are taken into consideration, *manA*, *yxeB*, *gltA* and *treP* are retrieved (**Table 4**). From them, *gltA* and *treP* are also found in diamide-treated samples, and *yxeB* is one of the H_2O_2 biomarkers (**Table 5**). Nevertheless, *lytE*, *uvrB* and *icd*, also found in diamide, H_2O_2 and paraquat stress responses, are not found as biomarkers of the general oxidative stress response and *manA* is not present in any stressor-specific analysis (**Table 4** and **5**). *manA* could be a false positive retrieved due to tagging samples subjected to various treatments as STRESS, increasing the variability of the expression data and making the processing more prone to be affected by artefacts.

The central regulators of the oxidative stress response LexA, Fur and PerR are present in the GRN of the retrieved biomarkers (**Fig. 11** and **12**). However, the only gene retrieved from the set of detoxifying enzymes previously identified as indicative of oxidative stress was the vegetative catalase *katA* for H_2O_2 (**Fig. 5**) (Imlay, 2015), which indicates either a bad performance of the biomarker retrieval algorithm or a lack of need of those detoxifying enzymes in the response to some oxidative stressors, probably in diamide's. Interestingly, diamide's deleterious mechanism does not involve the generation of ROS, the targets of catalase, superoxide dismutase and peroxidase (Kashyap, et al., 2014).

The application of the retrieval algorithm to heat stress retrieved *htpG* exclusively. *htpG* is the only member of the type IV heat-shock regulon in *B. subtilis* and encodes the chaperone HtpG (Schumann, 2003). The mechanism of induction of this regulon is still no clear but it is considered to be triggered by a membrane or extracellular sensor (Schumann, 2003). Chaperones have been found to participate in the specific adaptive response to heat stress so as to prevent protein misfolding and aggregation (de Nadal, et al., 2011). Moreover, it has been observed that HtpG associates to the ribosomal protein L2 during heat stress in *E. coli* (Motojima-Miyazaki, et al., 2010).

4.1. Limitations

Although generally the biomarkers found are consistent with the stress under analysis, it was not clear the extent to which the RNA-seq samples contributed to the result; consequently, the biomarker retrieval algorithm was tested executing it for oxidative stress using only microarray samples. In this analysis SVM-RFE returned *manA* in 100% of the iterations and RF-RFE returned *manA* (75%), *treP* (11.5%) and *gltA* (10.5%), similarly to the biomarkers obtained when RNA-seq samples were also included in the analysis (**Table 4**). This result indicates that the RNA-seq samples are being neglected, possibly due to their reduced number in comparison with microarray samples, preventing them from having a real impact in the result of the algorithm.

On the other hand, the raw number of reads per CDS in the RNA-seq samples does not provide a measurement able to compare the expression of different CDSs since the ones that are longer have a greater number of reads for the same level of expression. A scaling that would provide a better metric of the expression intensity would be RPKM (reads per kilobase per million of mapped reads). The RPKM scaling (3) was applied to the RNA-seq entries adding this step to the integration process (Scale_Normalise_batch_RPKM.R). After applying the rest of the biomarker retrieval algorithm targeting oxidative stress SVM-RFE (C=0.1) retrieved *manA* in 100% of the executions and RF-RFE (T=25) returned *ylaC* 32.5%, *fbp* 24.5%, *comZ* 22.5% and *lonA* 19.5%. These results are different from the ones obtained when the RPKM approach was not applied (**Table 4**), which indicates that RNA-seq samples influence the results only when the RPKM scaling is applied. Future work will check the biomarkers obtained for other stresses applying this new scaling step.

$$RPKM = \frac{raw\ reads}{CDS\ length} * \frac{10^6}{total\ number\ of\ reads} \qquad (3)$$

Another improvement will consider genes as part of operons instead of individual transcriptional units. In prokaryotes, CDSs that encode proteins participating in the same pathway are often organized in operons regulated by the same promoter so that they should have an equivalent level of expression. Consequently, the execution of this tool targeting the overall expression of operons instead of single CDSs would prevent false positives.

4.2. Future work

The different software used for the biomarker retrieval algorithm heavily depends on several parameters that are kept as default. In the current work the only ones tuned belong to the second feature selection step but it would be interesting to tune some of the parameters used by RGIFE or the pre-processing. The optimisation of more than one parameter requires the use a heuristic optimization scheme and, as shown in this report, this task is not straightforward when the differences in the fitness function are masked by the stochasticity of the algorithm. However, when more than one parameter is changed the evaluation of the fitness function may have a larger variation for different parameters and new optimum solutions may be found. The main issue when implementing this optimization for parameters used in early stages of the biomarker retrieval algorithm is the slow speed of the 10 executions of RGIFE, that would need to be distributed among different cores to make the optimization feasible.

The fitness function should also be modified since the GO terms may not be correctly assigned for poorly characterised genes. A new fitness function would take into account the topography of the GRN as a measurement of the proximity of the biomarker to predefined stress nodes such as *lexA*, *fur*, *perR*, *ohrR* and *cymR* for oxidative stress. Furthermore, the GRN can be enriched converting it into a PFIN (Probability Functional Integrative Network). PFINs weight the interaction between two genes according to the evidence gathered from several genetic, biochemical and computational experiments, providing a probabilistic measure of the likelihood of an interaction (Lee, et al., 2004). Although contrast of hypothesis and machine learning are striking different methodologies, as shown when comparing the results of Student's t-test and the ranking of biomarkers (**Table 4** and **5**), another improvement of the fitness function would include the p-values derived from a contrast of hypothesis as a measurement of the likelihood of a biomarker being a true positive.

As far as the genetic circuit is concerned, a continuous simulation is needed to check if the dynamic ranges of response of the TetR repressors are compatible with the functionality of the circuit. Different repressors trigger different levels of expression when they are on or off and it may occur that the off state of a repressor is enough to inhibit the expression of the next element in the circuit. The dynamic ranges of the TetR repressors had been maximised modifying the RBSs that control their expression, so that the difference of expression between on and off states is maximum (Stanton, et al., 2014). The same RBSs have been used in this work, and the response function of each repressor together with the expression profile of the biomarkers with and without stress should be used to find the best distribution of repressors among logic gates. Once the genetic circuit has been tried *in silico*, it has to be tested *in vivo*. For that purpose, it will be split in fragments that will be included in different commercial integrative plasmids to be synthesized using overlapping oligonucleotides.

4.3. Conclusions

This work describes a simple and versatile stress-specific biomarker retrieval algorithm. This pipeline can be applied to several biological quantitative measures in different organisms and non-stress scenarios such as differences in communities of bacteria using metagenomic experiments or biomarkers determinant of clinical conditions harnessing quantitative proteomics. Furthermore, to my knowledge, this work is pioneer in coupling machine learning to the design of a synthetic circuit able to dynamically adjust a cellular response.

Acknowledgements

Apart from the team of supervisor of this work, I would like to thank Dr. Victor Khomenko for his contribution to the design of the C-element circuit.

References

Analytics, C. 2015. Anaconda Software Distribution. Release 2-2.4.0.

Anders, S., Pyl, P. and Huber, W. HTSeq-a Python framework to work with high-throughput sequencing data. *Bioinformatics* 2015;31(2):166-169.

Andrews, S. 2010. A quality control tool for high throughput sequence data

Balibar, C.J. and Walsh, C.T. In vitro biosynthesis of violacein from L-tryptophan by the enzymes VioA-E from Chromobacterium violaceum. *Biochemistry* 2006;45(51):15444-15457.

Bartley, B., *et al.* Synthetic Biology Open Language (SBOL) Version 2.0.0. In, *J Integr Bioinform*. Germany; 2015. p. 272.

Belopolsky, A., *et al.* 2014. rpy2 2.3.9 package

Bolon-Canedo, V., *et al.* A review of microarray datasets and applied feature selection methods. *Inform Sciences* 2014;282:111-135.

Bolstad, B. 2016. preprocessCore: A collection of pre-processing functions. Release 1.34.0. https://github.com/bmbolstad/preprocessCore

Bonzanni, N., *et al.* Petri Nets Are a Biologist's Best Friend. *Formal Methods in Macro-Biology* 2014;8738:102-116.

Braz, V.S., *et al.* CztR, a LysR-Type Transcriptional Regulator Involved in Zinc Homeostasis and Oxidative Stress Defense in Caulobacter crescentus. *Journal of Bacteriology* 2010;192(20):5480-5488.

Breiman, L. Random forests. *Machine Learning* 2001;45(1):5-32.

Carneiro, S., *et al.* Metabolic responses to recombinant bioprocesses in Escherichia coli. *Journal of Biotechnology* 2013;164(3):396-408.

Ceroni, F., *et al.* Quantifying cellular capacity identifies gene expression designs with reduced burden. *Nature Methods* 2015;12(5):415-+.

Chaouiya, C., *et al.* Qualitative modelling of genetic networks: From logical regulatory graphs to standard Petri nets. *Applications and Theory of Petri Nets 2004, Proceedings* 2004;3099:137-156.

Chen, J.M., *et al.* Analysis and Construction of Genetic Network for Mice Brain Microarray Datasets. *Journal of Medical and Biological Engineering* 2013;33(4):400-405.

Chen, T., *et al.* Random forest in clinical metabolomics for phenotypic discrimination and biomarker selection. *Evidence-based complementary and alternative medicine : eCAM* 2013;2013:298183-298183.

Choe, D., *et al.* Minimal genome: Worthwhile or worthless efforts toward being smaller? *Biotechnology Journal* 2016;11(2):199-211.

Cline, M.S., *et al.* Integration of biological networks and gene expression data using Cytoscape. *Nature Protocols* 2007;2(10):2366-2382.

de Amorim, R.C. Computational Methods of Feature Selection. *Information Processing & Management* 2009;45(4):490-493.

de Nadal, E., Ammerer, G. and Posas, F. Controlling gene expression in response to stress. *Nature Reviews Genetics* 2011;12(12):833-845.

Demain, A.L. and Vaishnav, P. Production of recombinant proteins by microbes and higher organisms. *Biotechnology Advances* 2009;27(3):297-306.

Fang, W., *et al.* A machine learning framework of functional biomarker discovery for different microbial communities based on metagenomic data. In, *2012 IEEE 6th International Conference on Systems Biology (ISB)*. 2012. p. 106-112.

Filtz, T., Vogel, W. and Leid, M. Regulation of transcription factor activity by interconnected post-translational modifications. *Trends in Pharmacological Sciences* 2014;35(2):76-85.

Granitto, P., *et al.* Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products. *Chemometrics and Intelligent Laboratory Systems* 2006;83(2):83-90.

Guyon, I., *et al.* Gene selection for cancer classification using support vector machines. *Machine Learning* 2002;46(1-3):389-422.

Hecker, M. and Volker, U. Non-specific, general and multiple stress resistance of growth-restricted Bacillus subtilis cells by the expression of the sigma(B) regulon. *Molecular Microbiology* 1998;29(5):1129-1136.

Heiner, M. and Gilbert, D. BioModel engineering for multiscale Systems Biology. *Progress in Biophysics & Molecular Biology* 2013;111(2-3):119-128.

Helmann, J., *et al.* The global transcriptional response of Bacillus subtilis to peroxide stress is coordinated by three transcription factors. *Journal of Bacteriology* 2003;185(1):243-253.

Helmann, J.D. Bacillus subtilis extracytoplasmic sigma factors and defense of the cell envelope. *Current Opinion in Microbiology* 2016;30:122-132.

Herbig, A. and Helmann, J. Roles of metal ions and hydrogen peroxide in modulating the interaction of the Bacillus subtilis PerR peroxide regulon repressor with operator DNA. *Molecular Microbiology* 2001;41(4):849-859.

Hoffmann, F. and Rinas, U. Stress induced by recombinant protein production in Escherichia coli. *Adv Biochem Eng Biotechnol* 2004;89:73-92.

Hunter, J.D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering* 2007;9(3):90-95.

Ideker, T., *et al.* Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics* 2002;18 Suppl 1:S233-240.

Imlay, J.A. Diagnosing oxidative stress in bacteria: not as easy as you might think. *Current Opinion in Microbiology* 2015;24:124-131.

Johnson, W.E., *et al.* Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* 2007;8(1):118-127.

Kasahara, J., *et al.* Teichoic Acid Polymers Affect Expression and Localization of DL-Endopeptidase LytE Required for Lateral Cell Wall Hydrolysis in Bacillus subtilis. *Journal of Bacteriology* 2016;198(11):1585-1594.

Kashyap, D.R., *et al.* Peptidoglycan Recognition Proteins Kill Bacteria by Inducing Oxidative, Thiol, and Metal Stress. *Plos Pathogens* 2014;10(7).

Kim, T.K. and Colman, R.F. Ser(95), Asn(97), and Thr(78) are important for the catalytic function of porcine NADP-dependent isocitrate dehydrogenase. *Protein Science* 2005;14(1):140-147.

Langmead, B., *et al.* Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology* 2009;10(3).

Lee, I., *et al.* A probabilistic functional network of yeast genes. *Science* 2004;306(5701):1555-1558.

Lee, J.W. and Helmann, J.D. The PerR transcription factor senses H2O2 by metal-catalysed histidine oxidation. *Nature* 2006;440(7082):363-367.

Leek, J., *et al.* 2016. sva: Surrogate Variable Analysis. Release 3.20.0

Lewin, A., Su, X.D. and Hederstedt, L. Positively Regulated Glycerol/G3P-Dependent Bacillus subtilis Gene Expression System Based on Anti-Termination. *Journal of Molecular Microbiology and Biotechnology* 2009;17(2):61-70.

Lou, C., *et al.* Synthesizing a novel genetic sequential logic circuit: a push-on push-off switch. *Molecular Systems Biology* 2010;6.

Mahalik, S., *et al.* Genome engineering for improved recombinant protein expression in Escherichia coli. *Microbial Cell Factories* 2014;13.

Mailloux, R.J., *et al.* alpha-Ketoglutarate Dehydrogenase and Glutamate Dehydrogenase Work in Tandem To Modulate the Antioxidant alpha-Ketoglutarate during Oxidative Stress in Pseudomonas fluorescens. *Journal of Bacteriology* 2009;191(12):3804-3810.

McLaughlin, J.A., *et al.* VisBOL: Web-Based Tools for Synthetic Biology Design Visualization. *ACS Synthetic Biology* 2016.

Misirli, G., *et al.* BacillusRegNet: A transcriptional regulation database and analysis platform for Bacillus species. *Journal of Integrative Bioinformatics* 2014;11(2).

Misirli, G., Hallinan, J. and Wipat, A. Composable Modular Models for Synthetic Biology. *J. Emerg. Technol. Comput. Syst.* 2014;11(3):1-19.

Mols, M. and Abee, T. Primary and secondary oxidative stress in Bacillus. *Environmental Microbiology* 2011;13(6):1387-1394.

Motojima-Miyazaki, Y., Yoshida, M. and Motojima, F. Ribosomal protein L2 associates with E. coli HtpG and activates its ATPase activity. *Biochemical and Biophysical Research Communications* 2010;400(2):241-245.

Muller, C., *et al.* Removing Batch Effects from Longitudinal Gene Expression - Quantile Normalization Plus ComBat as Best Approach for Microarray Transcriptome Data. *Plos One* 2016;11(6).

Nguyen, N., *et al.* Design and analysis of a robust genetic Muller C-element. *Journal of Theoretical Biology* 2010;264(2):174-187.

Nicolas, P., *et al.* Transcriptional landscape estimation from tiling array data using a model of signal shift and drift. *Bioinformatics* 2009;25(18):2341-2347.

Nielsen, A.A.K., *et al.* Genetic circuit design automation. *Science* 2016;352(6281):53-+.

Ollinger, J., *et al.* Role of the fur regulon in iron transport in Bacillus subtilis. *Journal of Bacteriology* 2006;188(10):3664-3673.

Pedregosa, F., *et al.* Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 2011;12:2825-2830.

Perry, M. and Wagner, R. 2014. simanneal: python module for Simulated Annealing optimization. https://github.com/perrygeo/simanneal

Picossi, S., Belitsky, B.R. and Sonenshein, A.L. Molecular mechanism of the regulation of Bacillus subtilis gltAB expression by GltC. *Journal of Molecular Biology* 2007;365(5):1298-1313.

Pinti, M., *et al.* Lon protease at the crossroads of oxidative stress, ageing and cancer. *Cellular and Molecular Life Sciences* 2015;72(24):4807-4824.

Poliakov, I., *et al.* WORKCRAFT - A Framework for Interpreted Graph Models. *Applications and Theory of Petri Nets, Proceedings* 2009;5606:333-342.

Price, C., *et al.* Genome-wide analysis of the general stress response in Bacillus subtilis. *Molecular Microbiology* 2001;41(4):757-774.

Quinn, J.Y., *et al.* SBOL Visual: A Graphical Language for Genetic Designs. *Plos Biology* 2015;13(12).

Rezacova, P., *et al.* The crystal structure of the effector-binding domain of the trehalose repressor TreR from Bacillus subtilis 168 reveals a unique quarternary assembly. *Proteins-Structure Function and Bioinformatics* 2007;69(3):679-682.

Rui, B., *et al.* A systematic investigation of Escherichia coli central carbon metabolism in response to superoxide stress. *Bmc Systems Biology* 2010;4.

Scholkopf, B. and Smola, A.J. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press; 2001.

Schumann, W. The Bacillus subtilis heat shock stimulon. *Cell Stress & Chaperones* 2003;8(3):207-217.

Stannek, L., *et al.* Factors that mediate and prevent degradation of the inactive and unstable GudB protein in Bacillus subtilis. *Frontiers in Microbiology* 2015;5.

Stanton, B.C., *et al.* Genomic mining of prokaryotic repressors for orthogonal logic gates. *Nature Chemical Biology* 2014;10(2):99-105.

Stanton, B.C., *et al.* Systematic Transfer of Prokaryotic Sensors and Circuits to Mammalian Cells. *Acs Synthetic Biology* 2014;3(12):880-891.

Sulmon, C., *et al.* Abiotic stressors and stress responses: What commonalities appear between species across biological organization levels? *Environmental Pollution* 2015;202:66-77.

Sun, T. and Altenbuchner, J. Characterization of a Mannose Utilization System in Bacillus subtilis. *Journal of Bacteriology* 2010;192(8):2128-2139.

Sun, Z., *et al.* Batch effect correction for genome-wide methylation data with Illumina Infinium platform. *Bmc Medical Genomics* 2011;4.

Swan, A., *et al.* A machine learning heuristic to identify biologically relevant and minimal biomarker panels from omics data. *Bmc Genomics* 2015;16.

Tam, L., *et al.* Proteome signatures for stress and starvation in Bacillus subtilis as revealed by a 2-D gel image color coding approach. *Proteomics* 2006;6(16):4565-4585.

Tanous, C., *et al.* The CymR Regulator in Complex with the Enzyme CysK Controls Cysteine Metabolism in Bacillus subtilis. *Journal of Biological Chemistry* 2008;283(51):35551-35560.

Vaidyanathan, P., *et al.* A Framework for Genetic Logic Synthesis. *Proceedings of the Ieee* 2015;103(11):2196-2207.

van der Walt, S., Colbert, S.C. and Varoquaux, G. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering* 2011;13(2):22-30.

Varghese, S., *et al.* Submicromolar hydrogen peroxide disrupts the ability of Fur protein to control free-iron levels in Escherichia coli. *Molecular Microbiology* 2007;64(3):822-830.

Wacker, I., *et al.* The regulatory link between carbon and nitrogen metabolism in Bacillus subtilis: regulation of the gltAB operon by the catabolite control protein CcpA. *Microbiology-Sgm* 2003;149:3001-3009.

Ward, P.A.S., Taylor, D.J. and Ieee Computer, S. A hierarchical cluster algorithm for dynamic, centralized timestamps. In, *21st IEEE International Conference on Distributed Computing Systems*. Phoenix, AZ; 2001. p. 585-593.

Waters, T.R., *et al.* Damage detection by the UvrABC pathway. *Febs Letters* 2006;580(27):6423-6427.

Yu, G.C., *et al.* GOSemSim: an R package for measuring semantic similarity among GO terms and gene products. *Bioinformatics* 2010;26(7):976-978.

Zeng, X. and Martinez, T. Distribution-balanced stratified cross-validation for accuracy estimation. *Journal of Experimental & Theoretical Artificial Intelligence* 2000;12(1):1-12.

Zuber, P. Management of Oxidative Stress in Bacillus. In, *Annual Review of Microbiology*. Palo Alto: Annual Reviews; 2009. p. 575-597.