Computational and Structural Biotechnology Journal 23 (2024) 783–790

Contents lists available at ScienceDirect

Computational and Structural Biotechnology Journal

journal homepage: www.elsevier.com/locate/csbj

Augusta: From RNA-Seq to gene regulatory networks and Boolean models

Jana Musilova [a,b], Zdenek Vafek [b,c], Bhanwar Lal Puniya [b], Ralf Zimmer [d], Tomas Helikar [b], Karel Sedlar [a,d,*]

[a] Department of Biomedical Engineering, Faculty of Electrical Engineering and Communication, Brno University of Technology, Brno 61600, Czech Republic
[b] Department of Biochemistry, University of Nebraska-Lincoln, Lincoln 68588, NE, USA
[c] Institute of Forensic Engineering, Brno University of Technology, Brno 61200, Czech Republic
[d] Department of Informatics, Ludwig-Maximilians-Universität München, Munich 80539, Germany

ARTICLE INFO

Keywords:
Python package
Gene interactions
Mutual information
Transcription factor binding motifs
Databases

ABSTRACT

Computational models of gene regulations help to understand regulatory mechanisms and are extensively used in a wide range of areas, e.g., biotechnology or medicine, with significant benefits. Unfortunately, there are only a few computational gene regulatory models of whole genomes allowing static and dynamic analysis due to the lack of sophisticated tools for their reconstruction. Here, we describe Augusta, an open-source Python package for Gene Regulatory Network (GRN) and Boolean Network (BN) inference from the high-throughput gene expression data. Augusta can reconstruct genome-wide models suitable for static and dynamic analyses. Augusta uses a unique approach where the first estimation of a GRN inferred from expression data is further refined by predicting transcription factor binding motifs in promoters of regulated genes and by incorporating verified interactions obtained from databases. Moreover, a refined GRN is transformed into a draft BN by searching in the curated model database and setting logical rules to incoming edges of target genes, which can be further manually edited as the model is provided in the SBML file format. The approach is applicable even if information about the organism under study is not available in the databases, which is typically the case for non-model organisms including most microbes. Augusta can be operated from the command line and, thus, is easy to use for automated prediction of models for various genomes. The Augusta package is freely available at github.com/JanaMus/Augusta. Documentation and tutorials are available at augusta.readthedocs.io.

1. Introduction

A Gene Regulatory Network (GRN), a static map of regulatory mechanisms, is defined as a graph where nodes represent genes and edges correspond to their interactions. The edges are typically directed, so the relation between pairs of genes clearly states the regulator, usually the transcription factor (TF), and the target gene (TG). As the GRN decodes one of the most crucial biological processes, regulatory interactions, there are multiple approaches to infer GRNs [31]. In general, the approaches can be divided into two main types based on the input data. The first one relies on gathering already acquired knowledge from literature and databases. Although there are several databases of gene interactions [10,17,27,41], they currently contain only limited information. Therefore, laboratory experiments remain irreplaceable in revealing new, unpublished interactions. The use of experimental data is typical for the second type of GRN inference methods, which are most typically designed to process gene expression measurements using high-throughput sequencing technologies. Five basic approaches using different computational techniques can be distinguished: Bayesian, Boolean, neural networks, regression-based, and information theory [3]. Among them, the information theory-based approach is an ideal solution for large-scale data, as it can study the global properties of networks with a high number of genes and is simple to use [3]. Several tools based on the information theory approach, specifically mutual information (MI) calculation, already exist, e.g. ARACNE [30], PREMER [42], or minet [32]. However, these algorithms are insufficient to process high-throughput genome-wide transcriptomic data and mostly do not distinguish whether the interactions of gene pairs are positive, i.e., a TF activates a TG, or negative, i.e., a TF inhibits a TG. Although other high-performance tools exist, e.g., Inferelator 3.0 [39], GRNBoost2, and Arboreto [33], they are typically designed for the use of single-cell RNA-Seq data which are mostly not available for non-model, poorly

* Corresponding author at: Department of Biomedical Engineering, Faculty of Electrical Engineering and Communication, Brno University of Technology, Brno 61600, Czech Republic.
E-mail address: sedlar@vut.cz (K. Sedlar).

https://doi.org/10.1016/j.csbj.2024.01.013
Received 26 October 2023; Received in revised form 17 January 2024; Accepted 19 January 2024
Available online 20 January 2024

studied, organisms due to their complicated cultivation. Therefore, inferring a large-scale, genome-wide network is still a tremendous challenge, and computational tools for genome-wide GRN inference are not widely adopted. This lack of available tools capable of processing standard RNA-Seq data motivated us to develop a new approach combining experimental data processing with database mining for GRN inference implemented in the introduced Augusta package.

While a static biological network like GRN serves to study many properties [13], the ability to observe behavior that a network expresses over a time-course simulation takes knowledge much further. A Boolean Network (BN), also referred to as a Boolean Model, is a qualitative biological network serving to study the regulatory mechanisms by identifying gene-regulatory logic [25]. Only a few tools or databases deal with large-scale BNs. The examples include CellNOpt [15] used for converting networks to predictive logic models from perturbation signaling data, or SQUAD [5], which enables network simulations in a GUI environment. In terms of databases (DBs), Cell Collective (CC) [18] is a collaborative modeling software that contains a freely available database of curated BNs. Augusta addresses this lack of tools, and in addition to GRN inference offers a possibility to further combine the information obtained from a static GRN with Boolean logic while inferring a BN. Moreover, it leverages CC to adjust Augusta-inferred

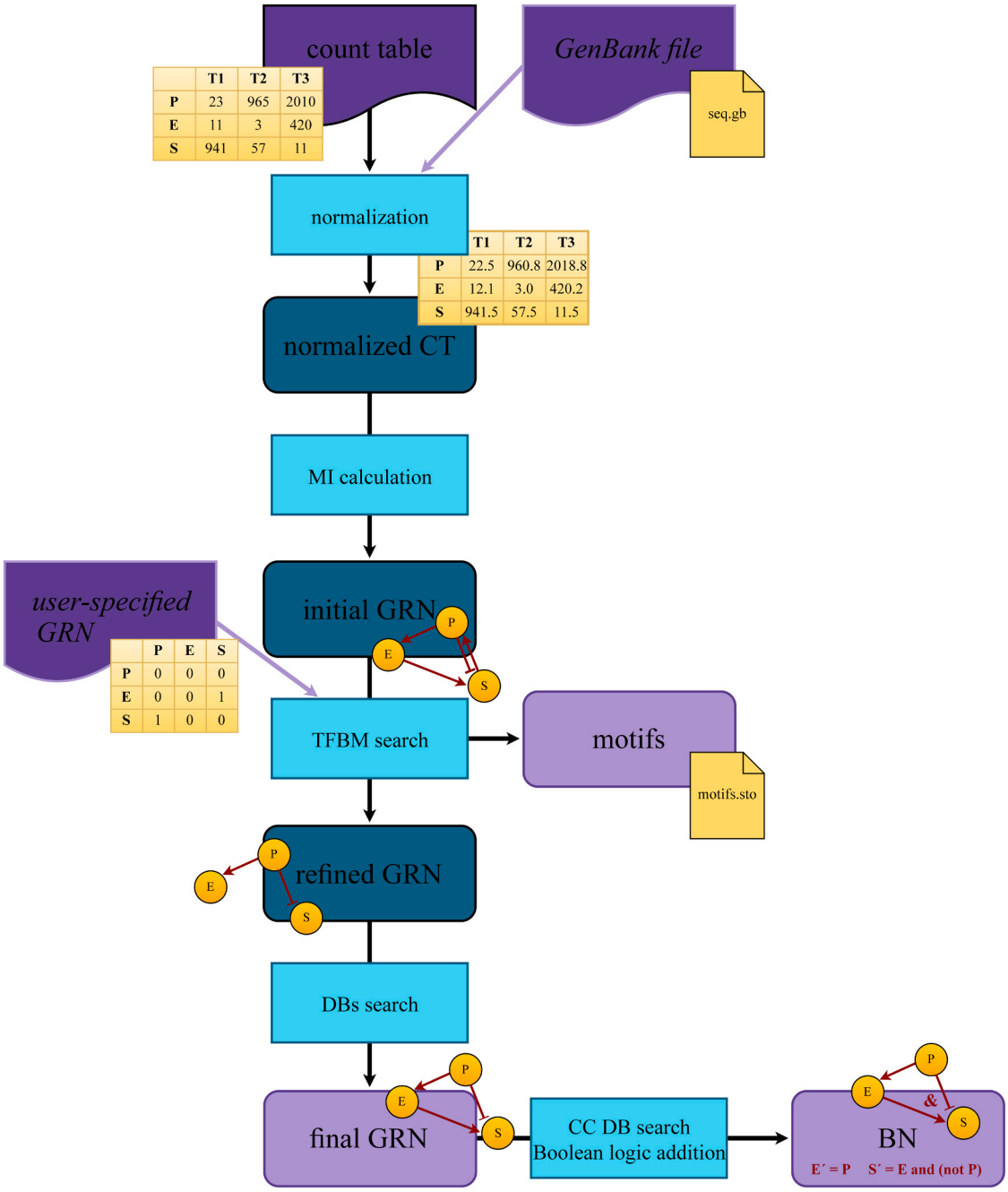

Fig. 1. Augusta pipeline: input files, a count table, and a GenBank file are used to optionally normalize the count table. Next, an initial Gene Regulatory Network (initial GRN) is inferred using a mutual information (MI) calculation. The initial GRN or a user-specified GRN is subsequently refined by searching for transcription factor binding motifs (TFBM) and databases (DBs). Moreover, the final GRN is converted into a Boolean Network (BN) by the Cell Collective (CC) database search and the addition of Boolean logic rules. As a result, the discovered motifs, the GRN in the form of an adjacency matrix, and the BN as an SBML-qual are exported. If the GenBank file is not provided, normalization, GRN refinement (TFBM and DBs search), and CC search are skipped.

logical rules and therefore presents a new alternative to other tools used for BN inference.

In this paper, we present Augusta, a Python package named after the famous mathematician and the first programmer, Augusta Ada King, Countess of Lovelace. Augusta combines experimental data with database searching and further refines inferred networks by *de novo* transcription factor binding motifs (TFBM) prediction. In addition to GRN inference, it offers the possibility to further transform this network into a BN. Augusta is a command-line interface (CLI) tool written in Python and is freely available under the MIT license as a source code and in a binary form from https://github.com/JanaMus/Augusta, as well as in binary form from the Python Package Index (PyPI) repository for easy installation. It is designed to be OS-independent, and thanks to CLI, it can be easily implemented into various shell pipelines. Moreover, the pipeline is designed to reconstruct whole-genome networks with low time complexity, and the most demanding computations are performed on publicly available servers. Therefore, a model for an average bacterium can be reconstructed in a reasonable time with a standard office PC or laptop.

2. Materials and methods

2.1. Overview

An initial Gene Regulatory Network is inferred from time-course gene expression datasets. Specifically, a set of gene expression data in the form of a time series of counts (a count table) is a required input that is used for the mutual information calculation. A GenBank file is an optional, yet highly recommended input providing additional information necessary for data normalization and a two-step initial GRN refinement by searching for transcription factor binding motifs, followed by searches in databases of TFs. Possibly, initial GRN inference can be skipped and a user can input his own GRN inferred by any tool to continue with the two-step refinement. The inferred final GRN is the main output of Augusta. Nevertheless, this GRN can be further converted into a Boolean Network. Finally, both of these networks can be exported in the form of an adjacency matrix and an SBML-qual file [7], respectively. Moreover, TFBM discovered *de novo* during GRN refinement are exported as well. Individual steps are shown in Fig. 1 and further described in the following subsections.

2.2. Gene expression data preprocessing

The study of gene expression provides insight into the cell's physiology at the transcriptomic level. The expression level of a gene is measured by a number of sequencing reads mapping to the particular gene. To achieve meaningful data analysis, normalization of the obtained raw counts is a crucial initial step. Normalization methods aim to eliminate systematic experimental bias and correct technical variations such as sequencing depth, library size, or gene length [28].

Several methods have been developed to normalize raw read count datasets. Since each method has its own strengths and limitations, the choice of normalization method depends on the research question and the nature of the data [12,14,24,28]. Therefore, Augusta offers several options for input data normalization. If a GenBank flat file, an optional input, is provided, the count table can be normalized using one of the provided methods [43]: Counts per Million (CPM), Reads per Kilobase Million (RPKM), and Transcripts per Million (TPM). Otherwise, an already normalized count table can be provided and a normalization step can be skipped.

2.3. Network inference with information theory approach

Information theory (IT) is a mathematical framework for quantifying and analyzing information. The main concept is based on a measurement of uncertainty, also known as entropy. Specifically, for a discrete random variable X, i.e., gene X in the input count table, the entropy $H(X)$ is calculated as follows:

$$H(X) = -\sum_{X\in x} P(x)\log_b P(x)\# \tag{1}$$

In the Eq. 1, $P(x)$ is the probability of event x, which corresponds to the expression level of the gene X. The base of the logarithm b represents the unit of entropy. Here, b is equal to the Euler number e representing the natural unit of information (nats).

The entropy of two discrete random variables, i.e., genes X and Y, the so-called joint entropy $H(X, Y)$, is the entropy of the joint probability distribution defined as:

$$H(X,Y) = -\sum_{X\in x}\sum_{Y\in y} P(x,y)\log_b P(x,y)\# \tag{2}$$

Mutual information (MI), a subset of information theory, determines the nonlinear dependency between a pair of variables, which is very common in nature. Therefore, it is an attractive approach for studying the communication of biological systems, such as interaction/relationship between gene pairs. In detail, the MI is a measure of the additional information known about one variable when given another:

$$MI(X;Y) = \sum_{X\in x}\sum_{Y\in y} P(x,y)\log_b \frac{P(x,y)}{P(x)P(y)} = H(X) - H(X|Y) = H(Y) - H(Y|X) \\ = H(X) + H(Y) - H(X,Y) \quad \# \tag{3}$$

To determine the relationship between two genes, MI can be calculated from the gene expression dataset. However, the normalized count table contains continuous values of expression levels. Therefore, binning, or the transformation of continuous values to discrete ones, is necessary. The optimal number of bins has been identified as 10 for most cases [4,22]. However, a too large number of bins relative to the number of genes in the input dataset may lead to estimation errors for the joint distribution. Augusta, therefore, uses an approach for small datasets that simulates adaptive distributions [6]. Specifically, the number of bins D is set according to the number of genes n to be 10 or less using the equation as follows:

$$D = \min\left(\lfloor\sqrt{n/5}\rfloor, 10\right)\# \tag{4}$$

Mutual information itself only reveals a relationship between two genes, i.e., proposes an edge between two nodes. The type of relationship, i.e., TF-TG or edge direction, as well as the regulation type, i.e., positive or negative, can be subsequently evaluated based on the transcriptional time lag [45]. The lag can be understood as the time needed for the TF gene to be translated into its protein product and the transcription of the TG to be affected by this regulatory protein. Thus, the evaluation is performed by identifying the most significant difference (MSD) among all adjacent pairs of time points in the measured time series of particular genes. The position of the MSD serves for the edge direction determination:

	T1	T2	T3	T4
P	40	960	980	999
E	25	60	280	272
S	322	321	348	12

	T2-T1	T3-T2	T4-T3
P	920	20	19
E	35	220	-8
S	-1	27	-336

Fig. 2. Principle of determining a type of relationship between pairs of genes, i.e., the definition of the TF-TG and positive-negative type of regulation.

$$e = \begin{cases} (v_1, v_2) & if \quad i < j \\ (v_2, v_1) & if \quad i > j \end{cases}, i = \underset{x \in (1,n)}{\operatorname{argmax}}\left(\left|DM_{1,x}\right|\right), j = \underset{x \in (1,n)}{\operatorname{argmax}}\left(\left|DM_{2,x}\right|\right) \quad \# \tag{5}$$

where e is an edge between nodes v_1 and v_2; $DM_{m,n}$ is a matrix of differences between adjacent time points of the count table of m genes and $n + 1$ time points. Furthermore, the sign of the MSD corresponds to the type of regulation. As the example in Fig. 2 shows, the P gene has the MSD in T2-T1 and the change is a positive value, which corresponds to the gene activation. Regulation of the gene E is lagged in comparison to P, as its MSD is captured in T3-T2. Therefore, P is considered to be the TF, and E is considered to be the TG in their relationship. As the captured change is also positive, P is considered to be an activator, i.e., the regulation is positive. Similarly, the highest regulation of S is lagged in comparison to both P and E. Therefore, both genes are considered to be TFs for S. Moreover, both of these TFs are repressors of S because the captured difference is negative.

2.4. Network refinement: motifs and databases search

While networks of living systems inferred solely by mathematical computations can provide useful predictions, accuracy is dependent on the quality and completeness of the input data. The networks only approximate reality from measured time series of genome-wide expression profiles whose sampling may not be sufficient to perfectly capture the complexity and interactions of the whole system. Moreover, these profiles summarize transcription in many cells present in the examined culture and are therefore biased. By incorporating additional knowledge on gene regulation, networks can better reflect the underlying mechanisms and interactions of biological systems, leading to more accurate predictions. Therefore, the topology of a GRN inferred by MI using Augusta can be further refined by *de novo* defining and searching for transcription factor binding motifs and by supplementing additional knowledge stored in curated databases associated with the organism under study.

A transcription factor binding motif, a short genomic sequence located in the promoter region of the gene, plays a crucial role in gene regulation. In order to initiate transcription, the TF binds to the promoter of the TG based on a specific motif pattern. A single TF most typically regulates several TGs that form its regulon. Therefore, the TFBM in gene promoters present in the same regulon are the same or very similar. The first step in a GRN refinement is based on *de novo* TFBM discovery, as shown in Fig. 3. Upstream sequences of genes in particular regulons are used to define new motifs recognized by TFs with the MEME Suite Docker image [2]. The default length of promoters, i.e., the upstream sequence to be analyzed, is set to 1000 bp [9,23,40]; however, the value can be adjusted by the user. In addition, only regulons that contain five or more genes are used to define new TFBMs. Subsequently, inferred motifs are searched in promoters of all other genes present in the network to reveal additional TF-TG relationships. The presence of the TFBM itself does not provide a type of regulation. Therefore, the transcriptional time lag is used again to determine whether the particular TF activates or inhibits the TG. In addition to the network refinement, TFBMs, which are unknown for most transcription factors, especially in non-model organisms, are also uncovered and reported as an auxiliary output.

The second step in GRN refinement is secured by database (DB) search. Manually curated DBs containing interactions of genes or proteins, particularly OmniPath [41], Signor [27], SignaLink [10], and TRRUST [17] are searched for the interactions corresponding to the organism under study. The organism name and the gene names, extracted from the input GenBank file, are further extended by other corresponding scientific and common names using the EcoNameTranslator Python package [11] to reveal all DB information. All regulatory relationships available in the DBs for a given organism are then added to the GRN. In the case a mismatch occurs between an edge type in the inferred GRN and in the DBs, information from the DBs is preferred as only curated databases are searched and thus considered reliable. In case of a mismatch between individual DBs, the type with the larger number of references is incorporated into the network; if the number of references is equal, no information is retrieved from the database. As a result, the final GRN presenting the main output of the Augusta package is constructed and exported in the form of an adjacency matrix in a.csv file format.

2.5. Boolean network inference

A Boolean Network, also referred to as a Boolean model, is a computational model used in the field of systems biology to analyze complex biological systems over time. The model serves to understand the dynamics of the regulatory mechanism as it gains insight into how different elements of the network influence each other and how the system responds to changes in input signals. Network inference involves assigning a Boolean function to each of the interacting network elements based on its regulatory interactions with other elements. In particular, the function for each target node is defined in terms of the logical operators AND, OR, NOT, and their combinations.

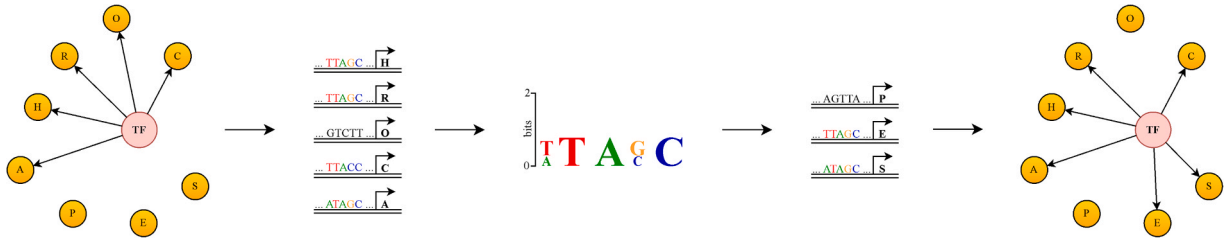

Fig. 3. GRN refinement by searching for transcription factor binding motifs (TFBMs). Initially, the upstream sequences of regulons defined by the initial GRN are utilized to define new motifs. The inferred motifs are then searched in the promoters of all other genes within the network. While some of the edges present in the initial GRN are filtered out due to the missing sequence motif in the promoter of a given regulon (TF-O edge in the example), other edges might be introduced in the refined GRN when newly defined motifs are found in the promoters of genes outside the given regulon, such as TF-E and TF-S pairs.

Augusta uses Boolean logic to transform the final GRN into a BN by specifying a Boolean function for every node in the network. The function is acquired in two ways depending on the availability of the data in the Cell Collective (CC) database: by using already published Boolean functions in the CC or by creating new generic functions. Regarding the former way, the process starts with the selection of available models. Since the transformation is organism-specific, only models for a studied organism are extracted from the CC database. Unlike GRN inference, where analysis is TF-oriented, BN inference is rather TG-oriented, as Boolean functions define input functions of particular regulated genes. For a given node, i.e., a TG, a Boolean function is transferred from the CC to the network being inferred only when a given TG and all its TFs in the final GRN are present in the extracted CC model. In the case the extracted model contains only partial information, the function for the TG cannot be transferred and the latter way of the function acquisition, i.e., creating a new generic function, needs to be employed. The generic function added to the TG node is created regarding the most commonly observed regulation processes: the logical OR operator is applied if only negative/positive interactions influence the TG (e.g. $A = B$ *or* C; $D =$ *not (E or F)*). On the contrary, the logical AND operator is applied if both negative and positive edges influence the TG to represent the dominance of the negative regulation (e.g. $G =$ *(not H) and I*).

Finally, the inferred BN is outputted in the SBML-qual file format. The format, an extension of the XML-based SBML, is a standardized format in biological modeling [19] and was designed specially to store qualitative models to which the Boolean networks belong [7]. The file is generated using the CellNOpt Python package (Cokelaer and Saez-Rodriguez, 2014), which has been extended for compatibility with the Augusta package by conversion into Python 3, and by adding a function for preserving nodes that are not connected to the rest of the network. As there are currently only a few reference BNs in the CC database, the BN inference in the Augusta pipeline is intended as a starting point for further development of various models utilizing primarily generic functions. These models can be manually refined, enhanced, and transferred to compatible software platforms, such as Cell Collective [18], BoolNet [34], or CellNOpt [15], which offer the possibility of running additional simulations.

2.6. Performance assessment

Benchmarking of Augusta was performed on several datasets and the results were compared to several tools for GRN inference: GENIE3 [20], TDAracne [44], and KBoost [21]. All analyses were performed on the same desktop PC with AMD Ryzen 3 3100 4-Core Processor with 3.59 GHz, 4 cores and 8 logical cores, and 32 GB of RAM. All tools were tested with default parameters. The required input parameter of TDAracne specifying a number of bins in percentile normalization or in rank normalization was set to 10 [22]. Performance was evaluated by measuring execution time, sensitivity, specificity, and accuracy.

First of all, we used the dataset obtained from the DREAM 4 In Silico Network Challenge [29] to evaluate Augusta's performance using an artificial gold standard network designed specifically for the comparison of GRN inference algorithms. Specifically, the first experiment of the time-series dataset consisting of 100 genes in 21 samples was used. To utilize all of Augusta's features, we downloaded the complete annotated genome sequence of *Escherichia coli* BW25113 [16] in the GenBank file format, as the DREAM 4 dataset was created based on this bacterium. Due to the fact that the gold standard dataset network does not contain a type of gene interactions, i.e., the GRN consists of directed edges but activating/inhibiting type of edges is not specified, the network inferred by Augusta had to be simplified by not considering the interaction types and all negative edges were converted to positive ones. Subsequently, binary classification was applied, in which a confusion matrix was computed to compare the predicted values of the benchmarking tools with the actual/true values provided by the gold standard. Using the confusion matrix, sensitivity is calculated as the proportion of correctly inferred interactions (true positives), while specificity measures the proportion of correct true negatives. Finally, accuracy is obtained as the overall proportion of correct predictions made by the benchmarked tool. Furthermore, we performed benchmarking of the count table normalization methods available in the Augusta pipeline, such as CPM, RPKM, and TPM.

Second, the *Bacillus subtilis* 168 RNA-Seq time-series dataset [35] consisting of 3997 genes in seven samples already normalized by the RPKM method, genome sequence (GenBank ID: AL009126.3) [26], and the gold standard GRN [1] containing both activating (positive) and inhibiting (negative) interactions were used for the evaluation based on the whole-genome data. Although the performance formula remained unchanged from that used for the DREAM 4 dataset, the confusion matrix was expanded to a 3×3 grid as the gold standard network contains an additional type of edge representing negative regulation (minus one). Therefore, multiclass classification was applied. Additionally, sensitivity and specificity metrics were calculated separately for each class, i.e., the type of the edge, and the resulting values were averaged. We also used the dataset to evaluate the time required for individual steps in the Augusta pipeline.

Finally, to demonstrate Augusta's ability to reconstruct networks for non-model organisms, we used a time series dataset from RNA-Seq that covers six time-points of *Clostridium beijerinckii* NRRL B-598 [38] consisting of 5442 genes and its complete genome sequence [37]. Since the GRN of *C. beijerinckii* is unknown, the parameters describing Augusta's performance could not have been calculated in this case. However, we compared the predicted GRN to the available gene ontology annotation of *C. beijerinckii* [36] to summarize Augusta's capabilities.

3. Results

Augusta's primary function, Gene Regulatory and Boolean networks inference, results in four outputs: (i) a GRN (adjacency matrix in.csv file format); (ii) identified motifs (in Stockholm.sto file format); (iii) all interactions found in the databases related to the input organism (list in.csv file format); (iv) a BN (in SBML-qual file format). Furthermore, the package offers secondary functions such as inferring only a GRN from a count table or inferring a BN from an existing GRN. Examples, test datasets, tutorials, and instructions are summarized in the documentation available from augusta.readthedocs.io or directly from GitHub github.com/JanaMus/Augusta.

To assess Augusta's performance, we conducted several evaluations. Initially, we benchmarked Augusta against existing tools for GRN inference using datasets containing gold standard GRNs. In addition, we tested Augusta in two different ways. Firstly, GRNs were inferred using a complete pipeline, i.e., an initial GRN was inferred by MI calculation and refined by TFBM and DBs search (labeled as Augusta in Table 1 and Table 2). Secondly, GRNs were inferred solely by MI calculation (labeled as Augusta no refinement in Table 1 and Table 3) to compare the performance with other tools, as their algorithms also rely on the inference without any refinement using sequence or database information. The RNA-Seq data normalization step was not performed as the remaining tools do not provide such an option. In addition to the benchmarking, we tested the functionality of the complete pipeline, i.e., both GRN and BN

Table 1
Comparison of different tools using the DREAM 4 challenge dataset. Augusta was tested both by refining the network by TFBM and DBs search (column labeled Augusta) and without providing GenBank file, i.e., GRN was inferred solely by MI calculation (column labeled Augusta no refinement).

	Augusta	Augusta no refinement	GENIE3	TDAracne	KBoost
Time [s]	714.18	3.07	4.62	2894.92	0.30
Sensitivity	0.06	0.38	1.00	0.15	1.00
Specificity	0.91	0.54	0.01	0.96	0.01
Accuracy	0.89	0.54	0.03	0.95	0.03

Table 2
Comparison of the performance achieved without a normalized dataset (column None) and datasets normalized using individual methods available in the Augusta pipeline: Counts per Million (CPM), Reads per Kilobase Million (RPKM), and Transcripts per Million (TPM). The DREAM 4 challenge dataset was utilized for the benchmarking.

	None	CPM	RPKM	TPM
Sensitivity	0.06	0.02	0.02	0.09
Specificity	0.91	0.93	0.93	0.91
Accuracy	0.89	0.91	0.91	0.90

inference, using two whole-genome bacterial datasets.

The first benchmarking was performed using the DREAM 4 challenge dataset for *E. coli*. The performance of the Augusta was compared with the existing tools for GRN inference, as shown in Table 1. Overall, the benchmarking results demonstrate that Augusta achieved the second best performance after TDAracne in terms of specificity and accuracy, but extremely better results in computational time. On the contrary, GENIE3 and KBoost achieved the lowest time and perfect sensitivity. However, their specificity and accuracy values were notably low due to the presence of edges connecting all gene pairs in the inferred GRNs.

Additionally, we performed the benchmarking of individual normalization methods using the DREAM 4 dataset. We compared the network's performance using the count table without normalization and the count table normalized by methods available in the Augusta, i.e., Counts per Million (CPM), Reads per Kilobase Million (RPKM), and Transcripts per Million (TPM). As shown in Table 2, CPM and RPKM techniques slightly improved specificity and accuracy of the GRN inference while considerably reducing sensitivity in comparison to the non-normalized data. On the other hand, TPM preserved high specificity and accuracy while improving sensitivity. This is not surprising, as initial MI calculation requires a comparison of adjacent time points, i.e., samples. Although CPM and RPKM are widely used techniques to normalize RNA-Seq-based count tables, they may not be optimal for comparison between different samples. On the contrary, TPM allows the most precise comparison of samples, i.e., the most precise MI calculation. Nevertheless, as results showed, while TPM can be generally recommended as the first choice, there is no versatile technique and particular results show tradeoff between sensitivity and specificity.

To demonstrate the performance on a real, whole-genome dataset, we performed the second benchmarking using the *B. subtilis* dataset, which contains 3997 genes. The results are provided in Table 3. Similarly to the previous results, TDAracne achieved the best specificity and accuracy, but the computations took almost six days, which is much longer than the other tools. Conversely, Augusta with no refinement by TFBM and DBs search achieved the best sensitivity, and in comparison to GENIE3 and KBoost also considerably higher accuracy. Moreover, the accuracy of Augusta became even higher by involving refinement steps. Overall, Augusta appears to be a promising tool for GRN inference, offering a good trade-off between accuracy and computational efficiency, even on a challenging whole-genome dataset. Computational complexity of Augusta's core algorithm is $O(mn^2D^2)$. Here, m corresponds to the number of time points in the input count table, expected to be in the order of units to tens at maximum, and n represents the number of genes. D equals the number of bins and is a constant of value 10 according to Eq. (4) if n is equal or higher than 500.

Table 3
Comparison of different tools using the *B. subtilis* dataset. Augusta was tested both by validating the network by TFBM and DBs search (column labeled Augusta) and without providing GenBank file, i.e., GRN was inferred solely by MI calculation (column labeled Augusta no refinement).

	Augusta	Augusta no refinement	GENIE3	TDAracne	KBoost
Time [h]	69.38	1.09	0.08	134.30	3.00
Sensitivity	0.33	0.35	0.34	0.33	0.33
Specificity	0.66	0.60	0.62	0.66	0.65
Accuracy	0.95	0.60	0.42	0.99	0.00

We also evaluated the time required for Augusta to perform individual steps in GRN and BN inference using the *B. subtilis* dataset. We show the time consumption on a whole-genome dataset consisting of 3997 genes, as well as on smaller sub-datasets. The results are provided in Fig. 4. While the initial GRN inference step consisting of data import and MI calculation is done in seconds to minutes, the subsequent GRN refinement consumes the highest amount of time. In particular, the TFBM search is the most time-consuming step. In addition to extracting promoter sequences from the genome and resulting data from the files, the sequences are sent to the MEME Suite web server for custom analysis, where the job is often queued. In return, validation fundamentally enriches the network, not only by obtaining the actual motifs for a given TF but also by improving its performance, as demonstrated by the comparison of inferred networks with gold standards (see Table 1 and Table 3).

Finally, to demonstrate the functionality of the Augusta pipeline on a whole-genome dataset of an exotic organism, we utilized the dataset of non-model bacterium *C. beijerinckii*. The final network comprises 2864 transcription factors (TFs) and 1880 target genes (TGs). This refined network outperformed the initial GRN obtained solely by the calculation of mutual information (MI), which consisted of 4684 TFs and 4529 TGs. The significant improvement achieved through network refinement highlights its importance. To further validate the biological accuracy of the network, we conducted a Gene Ontology (GO) enrichment analysis. Although it is evident that the number of predicted TFs is overestimated probably almost 10 times, GO terms related to transcription factor activity of the molecular function (MF) and biological process (BP) categories were within the top 100 terms enriched among predicted TFs, particularly MF term GO:0008134: transcription factor binding, top 43, p-value 0.14 (Fischer's exact test) and the BP term GO:0006355 (regulation of DNA-templated transcription), top 57, p-value 0.06 (Fischer's exact test). Although the statistical significance of these enriched terms is questionable, it is necessary to take into account that the whole network was constructed from six time points only, as time-series transcription data for non-model organisms are usually very sparse. Despite that, some of the believed to be TFs were still predicted.

4. Conclusions

We present the Augusta tool, a simple but effective approach suitable for observing the structural and dynamical properties of regulating mechanisms within the whole genomes. The main purpose of the Augusta is to infer a Gene Regulatory Network (GRN) enriched by the edges type (activating/inhibiting) and a draft Boolean Network (BN). Moreover, networks are refined by adding additional information based on transcription factor binding motifs (TFBM) and curated databases (DBs) search to increase network prediction performance. The tool is available as a Python package, so it can be used either standalone or incorporated into custom scripts. We believe Augusta can be of great use for biologists and biotechnologists who deals primarily with non-model bacteria that are difficult to work with using current lab protocols. Especially the fact that Augusta presumes the use of bulk RNA-Seq opens the possibility of GRN inference for a wide range of organisms. Besides GRN inference and definition of regulons for particular TFs, Augusta explicitly provides sequence motifs in inducible promoters, which is utilizable in genome engineering and synthetic biology in general. At last, not least, BN inference can be appealing not only for lab scientists looking for time course simulation of phenotype manifestation but also for computational systems biologists who propose new algorithms for BN analyses as this field remains underdeveloped due to the low amount of available models. Nevertheless, BNs provided by Augusta should be treated with caution as they rather present first drafts than final BNs.

Although benchmarking was done on prokaryotic datasets as we

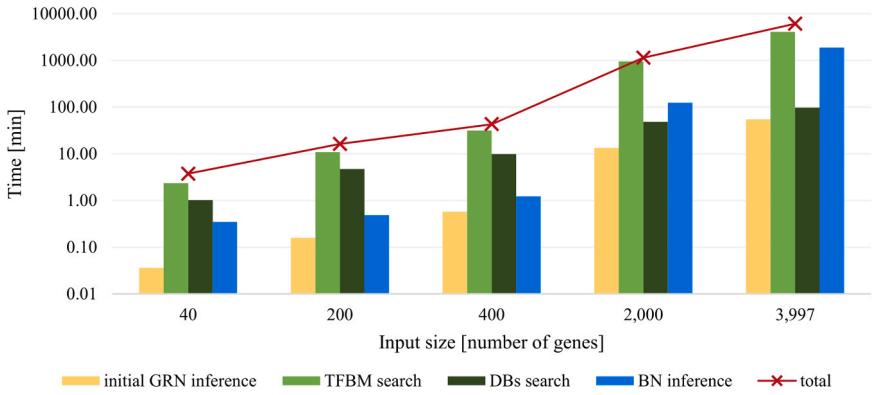

Fig. 4. Execution time of Augusta with respect to the input size using the *B. subtilis* dataset. The contribution of each Augusta step, as well as the total time, are highlighted.

aimed Augusta to be utilizable primarily on non-model bacteria, its potential extends to other organisms, as we verified during alpha testing on several other organisms. Although the refinement of an initial GRN may be computationally very demanding, it is computed on remote server and there is no restriction on the size of an input data. As we showed, a GRN for typical bacterial genome can be inferred in very reasonable time. Moreover, the refinement step can be omitted which leads to rapid inference of an initial GRN thanks to the low time and computational complexity of the core algorithm. Therefore, the inference is possible also for much more complex organisms than bacteria. Other advantage is that networks are inferred solely from a dataset of bulk time-series RNA sequencing, straightforward and commonly performed technique without high experimental complexity compared to other methods for measuring expression profiles, such as single-cell RNA-Seq [8]. Although refining steps cannot be performed without additional knowledge of a genome, Augusta can be used for the first approximation of networks even without the complete genome sequence and lack of data in databases for the particular organism, which is very common for non-model organisms known for the absence of available information.

Augusta has demonstrated promising performance. However, the inferred networks should be considered as approximations and not as perfectly accurate representations of biological systems regulations, particularly BNs, which are intended as drafts for further development. Although performing laboratory and computational experiments brings constantly expanding information, gaps disabling inferring perfect networks still exist. For example, inferring autoregulations remains a significant challenge not only in the field of network inference. Besides focusing on the listed limitations, Augusta's future directions involve processing multiple RNA-Seq experiments obtained by measuring gene expression during various environmental conditions during a single computation in tandem with incorporating the parallelization and thus reducing the computational time. In addition, future work includes the capability to enrich networks with additional information, such as known operon structures, as well as employing enhancements to the network inference process, especially in terms of BNs.

Funding

This work is a part of the project that has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 101023766. The work was also supported by grant project GACR GA22-10845S. TH and BPL were supported by NIH grant #R35GM119770.

CRediT authorship contribution statement

Jana Musilova: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Writing - review & editing, Visualization, Supervision, Project administration. **Zdenek Vafek**: SOFTWARE, Validation. **Bhanwar Lal Puniya**: Conceptualization, Methodology. **Ralf Zimmer**: Validation, Formal analysis, Resources. **Tomas Helikar**: Conceptualization, Methodology, Resources, Supervision. **Karel Sedlar**: Conceptualization, Methodology, Software, Validation, Formal analysis, Resources, Writing – review & editing, Visualization, Supervision, Funding acquisition.

Declaration of Competing Interest

None.

References

[1] Arrieta-Ortiz ML, et al. An experimentally supported model of the Bacillus subtilis global transcriptional regulatory network. Mol Syst Biol 2015;11:839.
[2] Bailey TL, et al. The MEME Suite. Nucleic Acids Res 2015;43:W39–49.
[3] Barbosa S, et al. A guide to gene regulatory network inference for obtaining predictive solutions: underlying assumptions and fundamental biological and data constraints. Biosystems 2018;174:37–48.
[4] Boulle M. Optimal bin number for equal frequency discretizations in supervized learning. Intell Data Anal 2005;9:175–88.
[5] Di Cara A, et al. Dynamic simulation of regulatory networks using SQUAD. BMC Bioinforma 2007;8(1):10.
[6] Cellucci CJ, et al. Statistical validation of mutual information calculations: comparison of alternative numerical algorithms. Phys Rev E - Stat Nonlinear, Soft Matter Phys 2005;71:066208.
[7] Chaouiya C, et al. SBML qualitative models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools. BMC Syst Biol 2013;7(1):15.
[8] Chen G, et al. Single-cell RNA-Seq technologies and related computational data analysis. Front Genet 2019;10.
[9] Cooper SJ, et al. Comprehensive analysis of transcriptional promoter structure and function in 1% of the human genome. Genome Res 2006;16:1.
[10] Csabai L, et al. SignaLink3: a multi-layered resource to uncover tissue-specific signaling networks. Nucleic Acids Res 2022;50:D701–9.
[11] Daniel Davies , 2020 EcoNameTranslator.
[12] Dillies MA, et al. A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. Brief Bioinform 2013;14:671–83.
[13] Emmert-Streib F, et al. Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks. Front Cell Dev Biol 2014;2:38.
[14] Evans C, et al. Selecting between-sample RNA-Seq normalization methods from the perspective of their assumptions. Brief Bioinform 2018;19:776.
[15] Gjerga E, et al. Converting networks to predictive logic models from perturbation signalling data with CellNOpt. Bioinformatics 2020;36:4523–4.
[16] Grenier F, et al. Complete genome sequence of Escherichia coli BW25113. Genome Announc 2014;2:1038–52.
[17] Han H, et al. TRRUST v2: an expanded reference database of human and mouse transcriptional regulatory interactions. Nucleic Acids Res 2018;46:D380–6.
[18] Helikar T, et al. The cell collective: toward an open and collaborative approach to systems biology. BMC Syst Biol 2012;6:1–14.
[19] Hucka M, et al. The Systems Biology Markup Language (SBML): language specification for level 3 version 2 Core Release 2. J Integr Bioinform 2019;16.
[20] Huynh-Thu VA, et al. Inferring regulatory networks from expression data using tree-based methods. PLoS One 2010;5:e12776.

[21] Iglesias-Martinez LF, et al. KBoost: a new method to infer gene regulatory networks from gene expression data. Sci Rep 2021 111 2021;11:1–13.
[22] Jung S, et al. Evaluation of data discretization methods to derive platform independent isoform expression signatures for multi-class tumor subtyping. BMC Genom 2015;16:S3.
[23] Kanhere A, Bansal M. Structural properties of promoters: similarities and differences between prokaryotes and eukaryotes. Nucleic Acids Res 2005;33:3165.
[24] Khan Y, et al. Normalization of gene expression data revisited: the three viewpoints of the transcriptome in human skeletal muscle undergoing load-induced hypertrophy and why they matter. BMC Bioinforma 2022;23:1–9.
[25] Kitano H. Systems biology: a brief overview. Science 2002;295:1662–4.
[26] Kunst F, et al. The complete genome sequence of the gram-positive bacterium Bacillus subtilis. Nature 1997;390:249–56.
[27] Licata L, et al. SIGNOR 2.0, the SIGnaling network open resource 2.0: 2019 update. Nucleic Acids Res 2020;48:D504–10.
[28] Liu X, et al. Normalization methods for the analysis of unbalanced transcriptome data: a review. Front Bioeng Biotechnol 2019;7:358.
[29] Marbach D, et al. Generating realistic in silico gene networks for performance assessment of reverse engineering methods. Marb, Daniel; Schaffter, Thomas; Mattiussi, Claudio; Flore, Dario (2009) Gener Realis silico gene Netw Perform Assess Reverse Eng Methods J Comput Biol 2009;16(*2*):229–39. *229-39.*, **16**.
[30] Margolin AA, et al. ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinforma 2006;7:1–15.
[31] Mercatelli D, et al. Gene regulatory network inference resources: a practical overview. Biochim Biophys Acta - Gene Regul Mech 2020;1863:194430.
[32] Meyer PE, et al. Minet: A r/bioconductor package for inferring large transcriptional networks using mutual information. BMC Bioinforma 2008;9(1):10.
[33] Moerman T, et al. GRNBoost2 and Arboreto: efficient and scalable inference of gene regulatory networks. Bioinformatics 2019;35:2159–61.
[34] Müssel C, et al. BoolNet—an R package for generation, reconstruction and analysis of Boolean networks. Bioinformatics 2010;26:1378–80.
[35] Omony J, et al. Dynamic sporulation gene co-expression networks for Bacillus subtilis 168 and the food-borne isolate Bacillus amyloliquefaciens: a transcriptomic model. Microb Genom 2018;4:e000157.
[36] Sedlar K, et al. A transcriptional response of Clostridium beijerinckii NRRL B-598 to a butanol shock. Biotechnol Biofuels 2019;12:243.
[37] Sedlar K, et al. Complete genome sequence of Clostridium pasteurianum NRRL B-598, a non-type strain producing butanol. J Biotechnol 2015;214:113–4.
[38] Sedlar K, et al. Transcription profiling of butanol producer Clostridium beijerinckii NRRL B-598 using RNA-Seq. BMC Genom 2018;19(1):13.
[39] Skok Gibbs C, et al. High-performance single-cell gene regulatory network inference at scale: the Inferelator 3.0. Bioinformatics 2022;38:2519–28.
[40] Tabach Y, et al. Wide-Scale Analysis of Human Functional Transcription Factor Binding Reveals a Strong Bias towards the Transcription Start Site. Plos One 2007;2(8):e807.
[41] Türei D, et al. OmniPath: guidelines and gateway for literature-curated signaling pathway resources. Nat Methods 2016 1312 2016;13:966–7.
[42] Villaverde AF, et al. PREMER: a tool to infer biological networks. IEEE/ACM Trans Comput Biol Bioinforma 2018;15:1193–202.
[43] Zhao Y, et al. TPM, FPKM, or normalized counts? a comparative study of quantification measures for the analysis of RNA-seq data from the NCI patient-derived models repository. J Transl Med 2021;19:1–15.
[44] Zoppoli P, et al. TimeDelay-ARACNE: reverse engineering of gene networks from time-course data by an information theoretic approach. BMC Bioinforma 2010;11(1):15.
[45] Zou M, Conzen SD. A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. Bioinformatics 2005;21:71–9.