9600–9610 *Nucleic Acids Research, 2016, Vol. 44, No. 20* *Published online 26 September 2016*
doi: 10.1093/nar/gkw843

Finding approximate gene clusters with GECKO 3

Sascha Winter[1], Katharina Jahn[2,3,4], Stefanie Wehner[5,6], Leon Kuchenbecker[2,7], Manja Marz[5,8], Jens Stoye[2] and Sebastian Böcker[1,*]

[1]Chair for Bioinformatics, Institute for Computer Science, Friedrich-Schiller-University Jena, Jena, Germany, [2]Genome Informatics, Faculty of Technology and Center for Biotechnology (CeBiTec), Bielefeld University, Bielefeld, Germany, [3]Computational Biology Group, Department of Biosystems Science and Engineering, ETH Zurich, Basel, Switzerland, [4]SIB Swiss Institute of Bioinformatics, Basel, Switzerland, [5]RNA Bioinformatics and High Throughput Analysis, Institute for Computer Science, Friedrich-Schiller-University Jena, Jena, Germany, [6]Institute of Aquaculture, School of Natural Sciences, University of Stirling, Stirling, FK9LA, Scotland, UK, [7]Berlin-Brandenburg Center for Regenerative Therapies, Charité University Medicine Berlin, Berlin, Germany and [8]Leibniz Institute for Age Research–Fritz Lipmann Institute (FLI), Jena, Germany

Received December 10, 2015; Revised September 6, 2016; Accepted September 12, 2016

ABSTRACT

Gene-order-based comparison of multiple genomes provides signals for functional analysis of genes and the evolutionary process of genome organization. Gene clusters are regions of co-localized genes on genomes of different species. The rapid increase in sequenced genomes necessitates bioinformatics tools for finding gene clusters in hundreds of genomes. Existing tools are often restricted to few (in many cases, only two) genomes, and often make restrictive assumptions such as short perfect conservation, conserved gene order or monophyletic gene clusters. We present GECKO 3, an open-source software for finding gene clusters in hundreds of bacterial genomes, that comes with an easy-to-use graphical user interface. The underlying gene cluster model is intuitive, can cope with low degrees of conservation as well as misannotations and is complemented by a sound statistical evaluation. To evaluate the biological benefit of GECKO 3 and to exemplify our method, we search for gene clusters in a dataset of 678 bacterial genomes using *Synechocystis* sp. PCC 6803 as a reference. We confirm detected gene clusters reviewing the literature and comparing them to a database of operons; we detect two novel clusters, which were confirmed by publicly available experimental RNA-Seq data. The computational analysis is carried out on a laptop computer in <40 min.

INTRODUCTION

Genomes evolve not only on the level of single nucleotides but by large-scale alterations, such as gene deletion, duplication, inversion and transposition. Without selective pressure, gene order and content would randomize over time. In reality, we observe low overall conservation of gene order between species, but a large number of shared genome segments with up to 50 conserved genes (1). These *gene clusters* can provide signals for functional analysis (2,3): for example, pairwise gene proximity indicates co-regulation in prokaryotes, independent of relative gene orientation (4) and/or cotranscription of operons. Moreover, multiple occurrences of regions with conserved gene content are strong indicators for whole genome duplication (5). For a large number of genomes, identification of gene clusters can be a computationally challenging task, since conservation patterns may vary across species due to micro-rearrangements, gene insertions and losses or misannotations.

Numerous approaches and software tools have been developed for the detection of gene clusters (6–28), but many tools are limited to a pairwise comparison which cannot detect faint signals. Only a few approaches can handle multiple genomes, appearances in a subset of genomes (where the *quorum parameter* determines the minimum number of genomes) and inexact gene clusters that are allowed to contain errors (Supplementary Table S1). OTFQC3Part/Isofun (26) is limited to relatively few genomes. CYNTENATOR (25) generates a gene-based genome alignment. The method follows a guide tree, ignoring or splitting occurrences of a gene cluster not exclusively contained in a single phylogenetic clade. Running times of MCMuSeC (23) severely increase when the quorum parameter becomes large, and its statistical evaluation ignores the actual cluster or its conservation, taking into account only the genomes a particular cluster is found in. Finally, i-ADHoRe 3.0 (27) requires more than 128 GB of memory for more than 500 genomes; by design, it only detects collinear conserved regions in multiple genomes, corresponding to diagonal lines in the genome dot-plots. Both

*To whom correspondence should be addressed: Tel: +49 3641946451; Fax: +49 3641946451; Email: sebastian.boecker@uni-jena.de

i-ADHoRe 3.0 and CYNTENATOR were designed to find syntenic blocks in the genomes of higher eukaryotes, where collinearity is more common than in bacteria. Few tools offer a graphical user interface. See Supplementary Material Sections 1 and 2 for details.

We present GECKO 3 for finding approximate gene clusters. It improves upon its predecessor GECKO 2 by greatly reduced memory consumption and running times, a sound statistical evaluation and more flexible error parameters. GECKO 3 features essential and desirable properties (Supplementary Material Section 1) that were not previously combined in any approach for gene cluster detection: (i) the program uses an intuitive and potentially 'biologically realistic' model for approximate gene clusters; (ii) GECKO 3 is an exact method that is guaranteed to find all gene clusters within the specified parameters; (iii) GECKO 3 is swift in practice and processes more than 500 bacterial genomes on a laptop computer (2.3 GHz Intel Core i7 processor, 16 GB main memory) in less than an hour, using <8 GB RAM; (iv) the quorum parameter can be chosen without directly impacting running time; (v) GECKO 3 integrates a sound and accurate statistical evaluation (FDR-corrected *P*-values) of detected gene clusters which is also swift in practice; and, finally, (vi) it offers a swift, flexible and easy-to-use graphical interface for visualizing results.

To exemplify our method and to evaluate its biological profit, we search for gene clusters in *Synechocystis* sp. PCC 6803 against 677 other bacterial genomes. GECKO 3 detected 65 gene clusters and we successfully verified all but two gene clusters using the literature and a database of operons. Genes of operons often evolve as gene clusters, due to their common biological relevance; but this is not always the case. Finally, the two novel gene clusters were successfully confirmed by RNA-Seq data.

GECKO 3 is available online at http://bio.informatik.uni-jena.de/software/gecko3/.

MATERIALS AND METHODS

GECKO 3 in a nutshell

GECKO 3 provides easy access to our methods for gene cluster detection and statistical evaluation. The general workflow of GECKO 3 is depicted in Figure 1. GECKO 3 consists of a Java program that implements the reference gene cluster method, the computation of *P*-values and a graphical user interface for setting search parameters, visualizing and filtering results (Figure 2). The tool can also be run on the command line. GECKO 3 is distributed as a ZIP archive and requires Java 7 to run. In addition, we supply a python script to prepare full GenBank files for all-versus-all blasting (29), clustering the results with TransClust (30) (http://transclust.mmci.uni-saarland.de/) and transforming the output to the GECKO 3 input format.

Informally, we model genomes as strings of gene numbers (or, equivalently, gene names) where homologous genes from one family are represented by the same number. A 'reference gene cluster' as detected by GECKO 3 is a set of genes that have an exact occurrence in one of the genomes, and (possibly inexact) occurrences in a sufficient number of further genomes: to measure 'inexactness', we simply count the number of genes that have to be added or deleted from the gene cluster to match the occurrence in the genome, conceptually similar to insertions and deletions in a sequence alignment. For robustness, we ignore both multiplicities and ordering of genes within the occurrences. See Figure 3 below for an example.

GECKO 3 implements an exact algorithm for finding all reference gene clusters. The algorithm has polynomial running time, increasing quadratically with the number of genomes and the length of the genomes. In practice, running times are very swift, as we have made considerable effort to improve speed by means of algorithm engineering. To be able to rank and filter the gene clusters, GECKO 3 performs a statistical evaluation of all computed clusters, based on the null model of random gene order. The graphical user interface permits swift browsing through the detected gene clusters, a non-trivial task for hundreds of genomes and gene clusters. Filtering of overlapping gene clusters can be switched on/off in the user interface. See Supplementary Material Section 3 on how to use GECKO 3.

Reference gene clusters

Reference gene clusters were introduced in ref. (31) under the name 'cluster filters', and only later proposed as an alternative gene cluster model (32). Unlike 'median' and 'center gene clusters' (31), a reference gene cluster has an *exact occurrence* in one of the genomes. This drastically reduces the computational complexity of finding such gene clusters, allowing gene cluster detection in hundreds of genomes. Details can be found in Supplementary Material Section 4 and refs. (31–33). See Supplementary Figure S1 for an artificial reference gene cluster, highlighting advances of this gene cluster model in comparison to other models.

For gene clusters with incomplete conservation patterns, we quantify the differences in gene content of their approximate gene cluster occurrences, corresponding to the number of genes deleted plus the genes inserted into a cluster occurrence. To limit the fuzziness of deleted and inserted genes, we introduce the *distance threshold* $\delta \geq 0$ such that the sum of inserted plus deleted genes is at most δ. Our model ignores the order and multiplicity of genes in an occurrence, and the gene clusters are modeled as (simple) sets. For example, assume that one genome has a gene region ABACB, the second has gene region BCDDC. The corresponding gene cluster (ignoring order and multiplicity) are A,B,C for the first genome (say, the reference gene cluster) and B,C,D for the second genome (say, an occurrence); the distance is two (one insertion, one deletion). See also Figure 3.

We demand that any cluster occurrence should have an overlap of at least two genes to the gene cluster. The *cluster size threshold* s is the minimum number of genes that a gene cluster must contain; for the reference A,B,C above, the size is three. The *quorum parameter* k' defines the minimum number of genomes that have an occurrence of the gene cluster. Formally, a *reference gene cluster* of k genomes with parameters s, δ, k' is a set of genes C with $|C| \geq s$ such that C has an exact occurrence in one of the genomes and C has δ-locations in at least $k' - 1$ other genomes, where a δ-location is an interval with distance at most δ to C. See the Supplementary Material for the complete definitions.

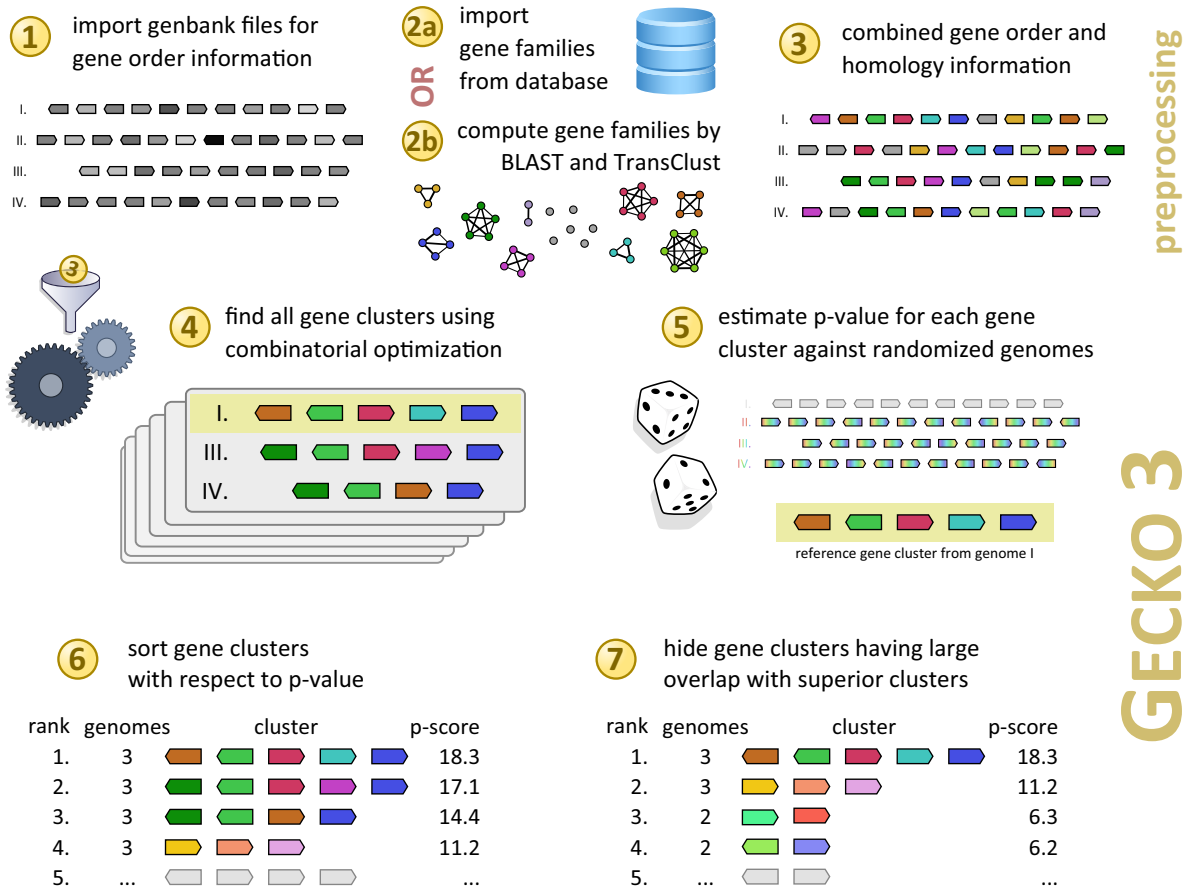

Figure 1. Workflow proposed for the analysis of gene order data using GECKO 3. (**1**) Gene order information imported from `GenBank` files. Homologous gene families are (**2a**) imported from a database such as STRING (36) or (**2b**) computed using all-against-all BLAST of the gene sequences, then applying a tool for finding gene families such as TransClust (30). We supply Python scripts for this step of the analysis pipeline. (**3**) The combination of gene order information and homology classification is imported into GECKO 3. (**4**) GECKO 3 finds *all* (hypothetical) gene clusters that are within the parameters given by the user. (**5**) Each gene cluster is evaluated by its *P*-value (significance), estimating the probability to encounter a gene cluster of this quality in randomized genomes. (**6**) Gene clusters are sorted by *P*-value, and (**7**) those showing a large overlap with a better gene cluster can be hidden in the user interface.

Algorithms

A high-level description of the algorithm behind GECKO 3 is as follows (Figure 1, steps 4–7):

- For given parameters s, δ, k', find all reference gene clusters in the genomes $S_1, \ldots, S_k$; alternatively, find all reference gene clusters in one genome selected by the user. For each reference gene cluster, output all optimal δ-locations in the other genomes.
- For each reference gene cluster and each combination of optimal δ-locations in the other genomes, compute its *P*-value as described below; discard all but the optimal combination of δ-locations.
- Filter overlapping gene clusters, reporting only the one with best *P*-value to the user. Filtering can be switched off in the user interface.

The efficient computation of reference gene clusters in uni-chromosomal genomes is described in ref. (31,32). As indicated above, it is straightforward to generalize this approach to multi-chromosomal genomes. Reference gene cluster computation in k genomes can be accomplished in $O(k^2n^2\delta^2 + k^2n^2)$ time using $O(kn^2)$ space (31,32), where n is the length of the largest genome. All optimal δ-locations of the reference gene clusters can be detected under the same time and space complexity (32). The algorithm is *exact*, meaning that it is *guaranteed* to find all reference gene clusters and their optimal occurrences as specified by the search parameters. Extending the algorithm to multi-chromosomal genomes, we reach the same time and space bounds, where n is the length of the largest genome after concatenation of chromosomes. See Supplementary Material Section 4.2 for details, and Supplementary Figure S2 for the pseudocode for finding reference gene clusters in two genomes.

Our algorithm outputs every gene cluster that cannot be extended. But this results in a large number of overlapping gene clusters: For example, a cluster may be found containing more genes, but conserved in less genomes, or with a higher distance to the reference. GECKO 3 offers the option to filter these cluster, keeping only those with the best *P*-value (see below). For this, we apply a simple greedy procedure that tries to add each cluster to a filtered list, processing clusters sorted by their *P*-value: we compare the new cluster

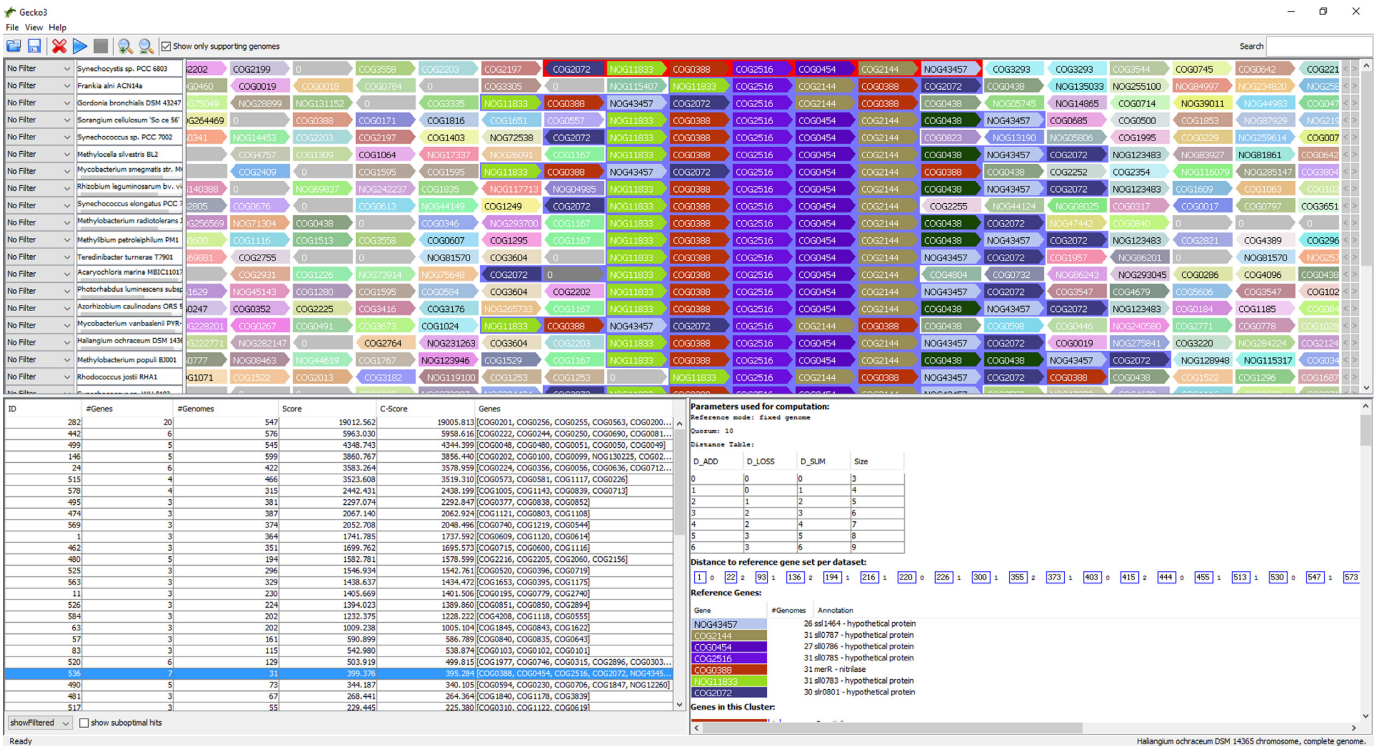

Figure 2. The GECKO 3 user interface after a gene cluster search has finished and one of the clusters has been selected for closer observation. 'Score' and 'C-Score' are negative logarithms (base 10) of the estimated *P*-value (uncorrected and FDR-corrected, respectively); for example, C-Score 395.284 corresponds to corrected *P*-value $10^{-395.284} = 5.20 \times 10^{-396}$.

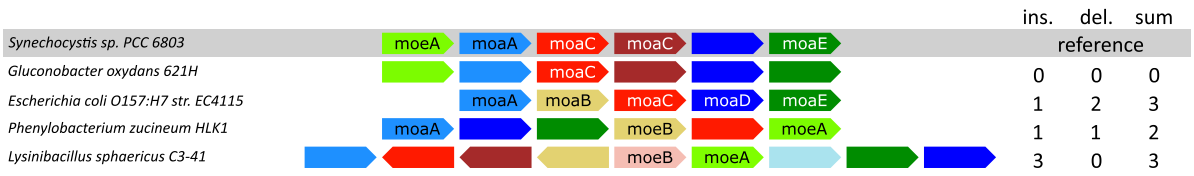

Figure 3. Selected occurrences of the gene cluster with ID 520 in Table 1 found in the STRING dataset. All genes of identical color belong to the same gene family; gene annotations are taken from RefSeq notes if available. In total, the cluster is found in 129 genomes. In the reference genome *Synechocystis* sp. PCC 6803, the cluster has five protein coding sequences, but one gene with locus_tag slr0902, named *moaC* has two different functional units (annotated COG0315, MoaC and COG0746, MobA in the RefSeq notes), here depicted in red and brown. For *Gluconobacter oxydans*, two other functional units (COG0746, MobA and COG1977, MoaD) are located on the same coding sequence, illustrated in brown and dark blue. Apart from that, the cluster is perfectly conserved between this two genomes. The gene order in *Escherichia coli* is well conserved, but two genes are missing (*moeA* and *mobA*) and *moaB* is inserted. In *Phenylobacterium zucineum* we find all but one (*mobA*) gene families of the reference genes, and again an inserted *moeB* gene, but with deviating gene order. *Lysinibacillus sphaericus* contains all genes from *Synechocystis* sp. PCC 6803, but we find three additional genes in that region and *moeA* is at a different position. The orientation of genes varies. Entries 'ins.', 'del.' and 'sum' give the number of additional, missing and sum of genes of occurrence versus reference gene cluster.

to all clusters already on the filtered list. If a cluster with a overlap in any genome is found in the list, we do nothing; otherwise, the cluster is added to the filtered list. We then proceed to the next cluster. Filtering is done solely for presenting the gene clusters in the graphical user interface, and the user can also inspect those gene clusters which have been filtered out.

In practice, many parameter combinations s, δ do not make sense: If $\delta \geq s - 2$, a δ-location is given by all occurrences of the two outer genes of the reference interval with at most $\delta - s + 2$ intermediate genes. So, parameters s, δ should be chosen as a pair. To search for gene clusters of different sizes at once, we modify the algorithm from Supplementary Material Section 4.2: Instead of a single parameter pair s, δ, the algorithm accepts pairs s_i, δ_i such that $s_i < s_j$ and $\delta_i \leq \delta_j$ holds for $i < j$. The algorithm guarantees to find all gene clusters for parameters $s = s_i$, $\delta = \delta_i$, for *one* of the pairs s_i, δ_i. Modifications of the algorithm are straightforward.

Searching for approximate occurrences of the conserved reference interval, the reference gene cluster model tends to overuse deletions: if we want to find approximate gene clusters with many insertions, we have to set a large distance parameter δ. Unfortunately, this usually results in the detection of many occurrences where a large number of genes from the reference interval have been deleted. To account for this problem, we modify the algorithm to accept three parameters δ^{add}, δ^{loss} and δ^{sum} instead of the sin-

gle distance bound δ. Here, δ^{add} is the maximum number of insertions, δ^{loss} is the maximum number of deletions, and δ^{sum} is the maximum sum of deletions and insertions (previously parameter δ). As the algorithm computes the number of deleted genes separately from the number of inserted genes, this modification is also straightforward to integrate into the algorithm. Again, we allow the user to provide not only one but many such parameter quadruples $s_i, \delta_i^{add}, \delta_i^{loss}, \delta_i^{sum}$: that is, for any (minimum) cluster size, we can choose the maximum allowed number of additional genes, missing genes and sum of additional plus missing genes. For example, the gene cluster and occurrences in Figure 3 can be detected for any parameters $\delta^{add} \geq 3$, $\delta^{loss} \geq 2$ and $\delta^{sum} \geq 3$ for $s = 6$.

It is understood that finding gene clusters strongly depends on a correct assignment of homology groups. Our flexible gene cluster model can deal with a certain amount of wrong assignments, but this is clearly limited. When homology assignment quality drops, higher distance parameters are required, which will result in higher running times and potentially worse *P*-values.

Algorithm engineering

Practical running times were strongly decreased by extensive algorithm engineering. We leave out the details, and just mention a single example: when processing 100+ genomes on a laptop computer, the memory requirements of the above method become the limiting factor. Here, the quadratic dependency of space on the genome length is no longer acceptable. To this end, we came up with a modification of the algorithm that only uses linear space but, in the worst case, no longer guarantees the theoretical running times mentioned above. In practice, the linear-space variant of the algorithm is often faster than the quadratic-space variant, and usually not significantly slower. See ref. (31–33) for more algorithm engineering tricks and evaluations thereof.

Statistical evaluation of gene clusters

Evidence of gene cluster conservation is typically explained as remnant ancestral gene order that was preserved up to present either for lack of divergence time, or due to selective constraints. However, it may as well be the case that seemingly conserved patterns occur merely by chance. The statistical evaluation integrated into GECKO 3 calculates *P*-values to measure the likeliness of such events. The approach used in GECKO 3 follows closely the framework introduced in ref. (34). In the following, we give a synopsis of this approach. Further details can be found in the original publication.

We use random gene order as the background model. For each genome, we draw a random string *S* of the same length where each character represents a gene family from the genome and has probability proportional to its frequency in the genome. We then estimate *P*-values, that is, the probability that a gene cluster of the observed quality can be found in the random genomes by chance.

Since we draw the random genomes independently, we can proceed as follows: for each genome, we compute the likelihood of a gene cluster occurring by chance in the corresponding random genome (Supplementary Section 5.1). These are the individual *P*-values for each genome. Next, we combine *P*-values from the individual genomes into one *P*-value for the gene cluster, taking into account the quorum parameter (Supplementary Section 5.2). Finally, we consider the problem of multiple testing using a false discovery rate (FDR) correction following Benjamini and Hochberg (35) (Supplementary Section 5.3).

We stress that our computations are deterministic, implying that the same cluster will always be assigned the same *P*-value if we run GECKO 3 multiple times on the same dataset. Evaluations indicate that reported *P*-values are highly accurate under the considered statistical model (34). Finally, note that GECKO 3 will sometimes report *P*-values which may appear to be unrealistically small: for example, the best gene cluster from Table 1 (ID 282) has *P*-value 1.5×10^{-19006}. But the gene cluster (corresponding to the 50S ribosomal protein L2) consists of 20 genes in the reference genome, and is found in 547 out of the 678 genomes with high degree of conservation. The probability that a gene cluster of this conservation could be *found by chance in random genomes*, is indeed vanishingly small. We stress that our statistical model does not take into account aspects such as the phylogenetic history of the organisms, so *P*-values should nevertheless be interpreted with care.

STRING dataset

To demonstrate the abilities of GECKO 3 to detect gene clusters in a large dataset, we analyze 678 bacterial genomes for which grouping of genes in gene families was available. The STRING database (36) (http://string-db.org/) clusters proteins into orthologous groups, namely manually curated 'clusters of orthologous groups' (COG) (37) and 'non-supervised orthologous groups' (NOG). We downloaded orthologous groups (COG and NOG) and combined the information with GenBank files downloaded from the RefSeq database (38) to generate the GECKO 3 input file. For species where multiple strains are present in the dataset, we keep only a single strain. This results in a dataset containing 678 genomes, see Supplementary Table S2 for a list of all contained genomes. The dataset is available in the GECKO 3 input format from http://bio.informatik.uni-jena.de/data/#gene_cluster.

RESULTS

We evaluate GECKO 3 by comparison to known operons of a model organism, and by comparison to other software tools. First, we evaluate gene clusters predicted by GECKO 3 for *Synechocystis* sp. PCC 6803. *Synechocystis* serves as model for fundamental and applied research in cyanobacteria (39), as it allows the analysis of reactions and metabolites of photosynthetic primary metabolism (39–41). Its biochemical similarities to plant chloroplasts are well-suited for research of molecular mechanisms underlying stress responses and stress adaptation in higher plants (42). Another hallmark of this unicellular cyanobacterium is its natural competence for DNA uptake (43,44). As *Synechocystis* sp. PCC 6803 is well-studied, we did not expect to identify

Table 1. Gene clusters of searching *Synechocystis* sp. PCC 6803 against 677 bacterial genomes using default distances (see Supplementary Table S3, left) and quorum parameter 10

						Synechocystis sp. PCC 6803		*Escherichia coli* O127:H6 str. E2348/69		
ID	p-score	G	GN	T	Collinearity	gene name	orientation	gene name	orientation	lit.
282	19005.81	20	547	–	546, 1	*adk-, rplBCDEFNO-PRVWX, rpmC, rpsCEHQS, secY*	all –	*rplBCDEFNOPRVWX, rpmCD, rpsCEHNQS, secY*	all –	(49)
442	5958.62	6	576	–	576	*nusG, rplAJKL, secE*	– – – – – –	*nusG, rplAJKL, secE*	+ + + + + +	(50)
499	4344.40	5	545	–	545	*fus-, rps7JL, tuf-*	– – – – –	*fusA, rpsGL, tufA*	– – – –	(51)
146	3856.44	5	599	–	599	*rpl36Q, rpoA, rps11M*	– – – – –	*rplQ, rpoA, rpsDKM*	– – – – –	(51)
24	3578.96	7	422	–	408, 14	*atpACDFGHI*	– – – – – – – –	*atpABEFGH*	– – – – – –	(52)
515	3519.31	6	466	–	460, 1, 4, 1	*pstABBCS, sphX*	– – – – – –	*pstABCS*	– – – –	(53)
578	2438.20	4	315	–	312, 3	*ndhAEGI*	– – – –	*nuoHIJK*	– – – –	(54)
495	2292.85	3	381	–	381	*ndhCJK*	+ + +	*nuoABC*	– – –	(55)
474	2062.92	3	387	–	315, 69, 3	*mntABC*	– – –	*	*	(56)
569	2048.50	3	374	–	374	*clpPX, tig-*	– – –	*clpPX, tig-*	+ + +	(57)
1	1737.59	4	364	–	50, 249, 65	*fecBCDE*	+ + + +	*fhuBCD*	+ + +	(58)
462	1695.57	5	351	–	349, 2	*nrtABCCD*	– – – – –	*	*	(59,60)
480	1578.60	5	194	–	179, 11, 3, 1	–, *kdpABCD*	+ + + + +	*kdpABCD*	– – – –	(61,62)
525	1542.76	4	296	–	295, 1	–, *nifS, ycf1624*	+ + + +	*sufBCDS*	– – – –	(63)
563	1434.47	3	329	–	300, 28, 1	–, –, –	+ + +	*ycjNOP*	+ + +	(64)
11	1401.51	3	230	–	230	–, –, *nusA*	+ + +	*	*	N
526	1389.86	3	224	–	223, 1	*minCDE*	– – –	*minCDE*	– – –	(65)
584	1228.22	3	202	–	200, 2	*cysATW*	+ + +	*cysAUW*	– – –	(47)
63	1005.10	3	202	–	202	*ctaCDE*	+ + +	*cyoABC*	– – –	(66)
57	586.79	3	161	–	76, 82, 3	–, –, *pilJ*	– – –	*cheAW, tar-*	– – –	(48)
83	538.87	3	115	–	114, 1	*rplM, rpsI, truA*	– – –	*	*	(67)
520	499.81	6	129	–	78, 6, 7, 12, 14, 6, 1, 1, 1, 3	–, *moaACCE, moeA*	+ + + + + +	*moaABCDE*	+ + + + +	(68)
536	395.28	7	31	–	7, 3, 3, 18	–, –, –, –, –, –, *merR*	– – – – – – +	*	*	(69)
490	340.10	5	73	–	71, 2	–, –, –, *rnpA, rpmH*	+ + + + +	*	*	(70)
481	264.36	3	67	–	46, 6, 15	–, –, –	– – –	*	*	N
517	225.38	3	55	–	52, 3	–, –, *cbiM*	– – –	*	*	(71)*
49	170.66	6	19	–	19	–, *psbEFJL, rub-*	+ + + + + +	*	*	(72)
488	162.14	5	23	–	23	–, –, *hoxFUY*	– – – – –	*	*	(73)
36	136.81	6	14	–	14	*ccmKKLMMN*	– – – – – –	*	*	(74)
552	134.59	7	26	–	25, 1	–, –, –, –, *nifJJJ*	– – – – – – –	*	*	N
38	114.94	3	34	–	30, 2, 2	*rfbCFG*	+ + +	*	*	(75)
60	107.51	6	13	–	13	–, –, –, –, *ndhDF*	+ + + + + +	*	*	(76)
472	94.19	3	17	–	17	*apcABC*	+ + +	*	*	(77)
585	86.08	3	18	–	18	–, –, –	+ + +	*	*	(78)
565	85.14	3	21	+	21	–, –, *lysA*	– – –	*	*	N
528	78.51	3	56	–	44, 12	–, –, –	– – +	*	*	(79)
506	77.33	3	15	–	15	–, –, –	+ + +	*	*	N
576	77.31	3	21	–	21	–, *asd-, dapA*	+ + +	*	*	(80)*
62	72.36	3	13	–	13	–, –, –	+ + +	*	*	(81)
509	71.83	3	14	–	14	–, –, –	+ + +	*	*	(82)
566	70.55	3	18	–	18	–, –, *dnaJ*	– – –	*	*	N
492	67.65	3	12	–	12	–, –, *pilM*	+ + +	*	*	(83)
512	61.96	3	16	–	16	–, *dgkA, trpG*	+ + +	*	*	N
549	59.76	3	11	–	11	–, –, *hypothetical protein*	– – –	*	*	N
67	54.28	3	17	–	17	*acpP, fabF, tktA*	– – –	*	*	N
434	49.52	3	15	–	15	–, *rpoB, rpsT*	– – –	*	*	N
455	47.40	3	11	–	11	–, *ndhD3F*	– – –	*	*	(76)
508	44.32	6	33	–	18, 7, 7, 1	–, –, –, –, –, –	+ + + + + +	*	*	N
463	41.58	3	10	–	10	–, –, –	– – –	*	*	N
511	41.18	3	13	–	13	–, *accB, efp-*	– + +	*	*	N
485	40.62	6	46	–	39, 4, 2, 1	–, –, –, –, –, –	+ + + + + –	*atoSS, rcsBCCDD*	+ + +– – + +	N
503	39.47	3	12	–	4, 3, 5	–, –, *arsC*	+ + +	*	*	(84)*
580	32.24	3	11	–	11	*gap2, murBC*	– + +	*	*	N
554	31.08	6	10	–	10	–, –, –, –, –, *tar-*	– – – – – –	*	*	N
42	31.02	6	16	–	16	–, –, –, –, *galE, rfbU*	+ + + + + +	*	*	N
586	30.26	3	11	–	4, 7	–, *rfbAB*	– – –	*	*	(75)*
456	29.57	4	11	–	11	–, –, –, *icsA*	– – – –	*	*	N
435	25.66	5	22	–	4, 6, 1, 8, 2, 1	–, *ETR111, cobN*	+ + + + +	*	*	(85)
510	23.64	3	11	–	1, 10	*glcP, secDF*	– + +	*	*	N
10	22.97	3	12	–	5, 7	–, *exbB, fhuA*	– – –	*	*	(86)
507	22.69	5	20	–	16, 3, 1	–, –, –, –, –	– – – – –	*	*	N
534	21.91	3	12	–	7, 5	–, –, –	– – –	*	*	(87)
9	6.24	3	11	–	5, 2, 4	–, *iutA, pchR*	– – –	*	*	N
486	5.55	5	15	–	15	–, –, –, –, *mtfB*	– – – – –	*	*	N
489	-1.48	5	11	–	1, 8, 1, 1	–, –, –, –, –	– – + + +	*	*	N

Clusters sorted by corrected *P*-values; 'p-score' is the negative logarithm (base 10) of the corrected *P*-value. 'G' is number of genes in the reference gene cluster; 'GN' is the number of genomes where the reference gene cluster is found. 'T' is '+' if the cluster is found in a monophyletic clade of the phylogenetic tree from STRING. 'Collinearity' describes how often the cluster is found with all genes in the same order. Gene names taken from the GenBank entries for *Synechocystis* sp. PCC 6803 and *Escherichia coli* O127:H6 str. E2348/69. Multiple gene names starting with the same first three letters are merged: for example, 'rplB' and 'rplC' become 'rplBC'. If the GenBank entry contains no gene name then we put '–'. If the cluster has no occurrence in *E. coli* then we put '*'. Column 'lit.' shows verification of the cluster in the literature, where 'N' means that no cluster description was found. *Cluster was described in other bacteria but not in Synechocystis sp. PCC 6803.

many gene clusters previously unknown; we aimed to evaluate GECKO 3's ability to recover known operons, evaluating it against the literature available for this cyanobacterium. Not all operons are evolutionarily organized as gene clusters, and of course, not all gene clusters are operons. But for the vast majority of gene clusters detected as part of this study under default parameters, we were able to verify these as operons.

For our computations, we use *Synechocystis* sp. PCC 6803 as the reference genome, comparing it against the other 677 bacteria from the STRING dataset (Supplementary Table S2). We use a quorum parameter of ten. We run GECKO 3 with two different parameter sets, namely, default parameters and relaxed parameters which allow for more insertions and deletions of genes in cluster occurrences (Supplementary Table S3).

Gene clusters in Synechocystis

Using default parameters, GECKO 3 predicts 587 gene clusters. After applying the filter option, we analyzed the remaining 65 gene clusters varying in size from 3 to 20 genes with an average of 4.28 genes, and FDR-corrected *P*-values ranging from $10^{-1.48}$ to 10^{-19005}, see Table 1 for the list of detected gene clusters, Figure 3 for an example gene cluster and Supplementary Figures S9 and 10 for the visualization of all gene clusters. Out of the predicted 65 gene clusters, 46 are not present in *Escherichia coli* O127:H6 (str. E2348/69); of these, 20 are cyanobacteria specific.

For the 65 gene clusters, we searched for laboratory confirmation in the literature, both for any organisms and also specifically for cyanobacteria and *Synechocystis* sp. PCC 6803. We find that 38 gene clusters were already described in *Synechocystis*, 4 within bacteria and for 23 gene clusters, no literature entry was found (Table 1).

Computations were performed on a Laptop Intel Core i5 M520, 2.4 GHz with 8 GB of RAM running Ubuntu Linux 14.04. For the default parameters, computations took 22 min for finding gene clusters plus 17 min for the statistical evaluation. Using relaxed parameters, computation required 33 min for finding gene clusters and 34 min for the statistical evaluation, resulting in 1179 clusters and 206 clusters after filtering. See Supplementary Table S5 for a list of gene clusters after filtering for relaxed parameters. See Supplementary Figures S3 and 4 for the influence of search parameters and different dataset sizes on running times.

Evaluation using DOOR 2.0 and Kopf *et al.*

We evaluate predicted gene clusters against the Database of prOkaryotic OpeRons DOOR 2.0 (45) and results of RNA-Seq studies reported by Kopf *et al.* (39). DOOR contains operons computationally predicted by genome-specific and general genomic information, such as promoter motifs. Kopf *et al.* (39) determined 4091 transcriptional units for *Synechocystis*, using RNAseg (46) on RNA-Seq data under 10 different environmentally relevant stimuli which provide information about operons. Although both methods are comparatively reliable, neither of the two sources provide experimental evidence of the predicted operons. Somewhat surprisingly, we detect a very strong agreement between operons and gene clusters, two non-related concepts.

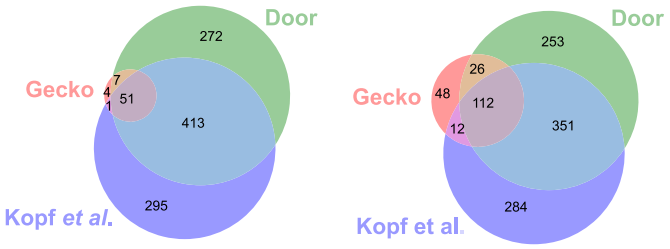

Figure 4. Venn-Diagrams comparing results of GECKO 3, DOOR 2.0 (45) and operons reported by Kopf *et al.* (39). GECKO 3 is run with default parameters (left) and parameters with reduced minimum size and increased maximum distance, increasing sensitivity but decreasing specificity of the method (right, see Supplementary Table S3). We combine clusters and operons in the three dataset based on connected components.

However, we find that gene clusters and operons are rarely *identical*, meaning that beginning and end perfectly agree for all three data sources. Hence, we rather test whether gene clusters and operons contain at least 50% identical genes or one is fully contained in the other; such cases are presumed to be identical. Now, one operon reported in DOOR 2.0 may overlap with two or more gene clusters detected by GECKO 3 and *vice versa*; to this end, we combine all operons and gene clusters that are contained in the same connected component, interpreting the similarity of operons and gene clusters defined above as the edges of an undirected graph. This case is rare in application: for default parameters, it results in 63 combined gene clusters instead of the original 65 for GECKO 3. For DOOR 2.0 we have 743 combined operons and 760 for RNA-Seq studies (39) (overlap of 464 operons). In Figure 4 we show the overlap of gene clusters and operons from GECKO 3, DOOR 2.0 and ref. (39). For default parameters, all but four gene clusters predicted by GECKO 3 can be verified using DOOR 2.0, results from ref. (39) or both. For relaxed parameters, the number of 'novel' gene clusters not reported in DOOR 2.0 or ref. (39) increases to 48.

Verification by RNA-Seq data

Using default parameters, four clusters (ID 9, 57, 485 and 584) are exclusively predicted by GECKO 3. These gene cluster have FDR-corrected *P*-values between 6.0×10^{-1229} (ID 584) and 5.8×10^{-7} (ID 9). Their appearance among all bacterial genomes is displayed in Supplementary Table S4. For two clusters (ID 57 and 584) we find laboratory confirmation in the literature (47,48). Cluster 485 with *P*-value 2.4×10^{-41} contains evolutionary conserved genes oriented antisense to each other, which are usually not detectable by approaches using RNA-Seq data. Cluster 9 consists of three proteins, namely IutA, a ferric aerobactin receptor, PchR, an AraC-like regulator of *fptA* gene expression and a protein with so far unknown function. All of these are encoded on the minus strand and show intergenic expression. We successfully verified these four clusters (ID 9, 57, 485 and 584) including functional operons using six RNA-Seq libraries from (39) (Figure 5).

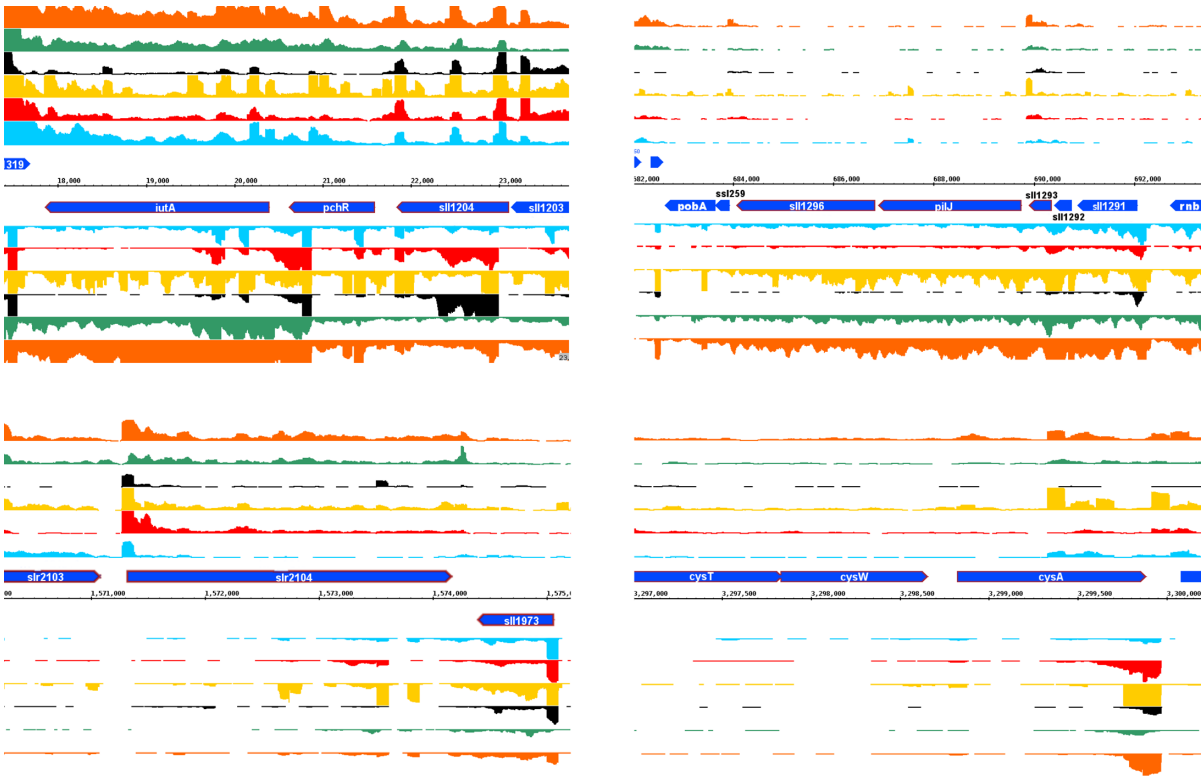

Figure 5. Confirmation of the four gene clusters 9, 57, 485 and 584 as operons using six RNA-Seq libraries from (39). Gene clusters ID 9 (*pchR* operon, left top), ID 57 (*pilJ* operon, right top), ID 485 (*rcs* operon, left bottom) and ID 584 (*cysW*, *cysT*, *cysA* operon, right bottom). Notably, *rcs* operon can be extended by another gene on the antisense strand, being part of the conserved cluster. In each subfigure, the upper half refers to the plus strand and the lower half to the minus strand. The y-axis is adjusted to 100 reads per RNA-Seq library. Orange and green—all untreated; black—dark, no light for 12 h; yellow—high light, 470 μmol quanta $m^{-2}s^{-1}$ for 30 min; red—heat stress, 42 °C for 30 min; light blue—cold stress, 15 °C for 30 min; blue—annotation. Continuous coverage of reads across several genes of a cluster are commonly interpreted as operons. Red boxed genes were detected to be part of the cluster.

Monophyly and collinearity

CYNTENATOR (25) is a progressive alignment method that repeatedly builds pairwise alignments of the gene sequences, following a guide tree. To this end, we test all clusters detected by GECKO 3 for monophyly. Clusters that are not found exclusively in one clade cannot be detected using CYNTENATOR, or it may detect multiple independent clusters for different clades. In both cases, the underlying idea of gene clusters being monophyletic is defective. Using the phylogenetic tree from the STRING database as our guide tree, we find that only one of 65 gene clusters from Table 1 is monophyletic. We also find that the four 'novel' gene clusters (ID 9, 57, 485 and 584) are widely distributed among different bacterial classes (Supplementary Table S4). We find similar results for the 206 gene clusters found by GECKO 3 using relaxed parameters (Supplementary Table S5).

Both i-ADHoRe (27) (in 'collinearity mode') and CYNTENATOR can only detect collinear gene clusters, forbidding micro-rearrangements inside the cluster. To this end, we check the 65 gene clusters from Table 1 for collinearity. To test whether two occurrences are collinear, we first remove all genes which are not present in the reference gene cluster found by GECKO 3. We then check if one occurrence is a subsequence of the other; this definition allows that gaps may be present in one of the clusters. For each gene cluster, we start with the reference as our first *collinearity group*, then try to add each cluster to one of the existing collinearity groups; in case this is not possible, we open up a new collinearity group. We find that for half of the gene clusters in Table 1, the collinearity assumption is violated, as there is more than one collinearity group. As an example, consider gene cluster ID 520 (Figure 3): The occurrence of the gene cluster in *Phenylobacterium zucineum* shows almost the same gene content as for the other genomes, but with a strongly deviating gene order. Again, we find similar results for the 206 gene clusters found by GECKO 3 using relaxed parameters (Supplementary Table S5).

Comparison with MCMuSeC

The three tools MCMuSeC (23), CYNTENATOR, i-ADHoRe 3.0 are highly advanced approaches which fulfill most criteria for a successful gene cluster detection method. Given that the monophyly assumption is violated for the vast majority of gene clusters detected by GECKO 3, we refrained from further evaluating CYNTENATOR. We were unable to process the dataset using i-ADHoRe 3.0 in 'collinearity mode', as it ran out of memory on a compute cluster with 128 GB RAM (multiplicon level 93). The 'cloud mode' of i-ADHoRe 3.0, which allows for micro-

rearrangements inside the gene cluster, does not support the detection of a gene clusters occurring in more than two genomes.

We perform a detailed comparison to MCMuSeC, which can find gene clusters in a large number of genomes and assumes neither monophyly nor collinearity. To allow for a fair comparison, we evaluate GECKO 3 on the dataset provided by Ling *et al.* (23), consisting of 133 genomes. We are able to rediscover all 18 gene clusters reported as novel in (23) using GECKO 3 with highly relaxed parameters (Supplementary Table S3). Further relaxing parameter δ_{add} is necessary as the max-gap gene clusters model behind MCMuSeC allows for a large number of inserted genes (Supplementary Figure S1). We observe that GECKO 3 detects many of these gene clusters in a more complete form, as gene clusters contain more genes and/or have occurrences in additional genomes; see Supplementary Figures S5–7 for nine examples. We also compare running times of MCMuSeC and GECKO 3 with respect to the quorum parameter (Supplementary Figure S8): whereas GECKO 3 computations get faster for larger quorum parameter, as fewer gene clusters are detected and statistically evaluated, MCMuSeC running times significantly increase with increasing quorum parameter and can become prohibitive for quorum parameter exceeding 15. Finally, we test the random sampling method used by MCMuSeC to estimated statistical significance: in 10 runs with identical input and identical parameters, about 7% of the reported gene clusters are considered significant (P-value ≤ 0.05) in one run, but insignificant in another run. See Supplementary Material Section 7 for details.

DISCUSSION

GECKO 3 is a tool for approximate gene cluster detection that, for the first time, includes all prerequisites needed for this type of analysis in times of next generation sequencing. Applying GECKO 3 to a newly sequenced bacterium plus a set of reference genomes can generate a set of high-quality operon candidates for the novel genome. Clearly, GECKO 3 can also be applied to eukaryotic genomes (5).

GECKO 3 does not require gene clusters to be collinear or monophyletic. We found that for the prokaryotic data analyzed here, the collinearity assumption is widely violated; this is even more so for the assumption of monophyly of gene clusters. GECKO 3 assigns P-values to all gene clusters which are based not only on the (number of) genomes a cluster is detected in, but also on the number of genes and the degree of conservation. Furthermore, computation of gene clusters is exact and P-value estimation is deterministic, meaning that repeated analysis of a dataset will result in exactly the same gene clusters with exactly the same P-values being detected and that no gene cluster within the given parameters will be missed. To avoid that all genes from the reference gene cluster can be deleted in an occurrence, GECKO 3 offers individual parameters for insertions, deletions, and sum of both that can be chosen depending on the cluster size. Finally, GECKO 3 offers a user-friendly graphical interface to view results, a feature missing from most other approaches for finding gene clusters.

Next to 63 gene clusters previously described in the literature or databases, we find two novel clusters in *Synechocystis* sp. PCC 6803 and confirmed these using RNA-Seq data, including an antisense clusters.

Currently, our random model does not take into account the phylogenetic relationship of genomes: to find a gene cluster that is conserved between two closely related species or strains, is much less surprising than to find this gene cluster for distantly related species. It is an interesting open question how to integrate such phylogenetic information into our null model.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

FUNDING

Deutsche Forschungsgemeinschaft (DFG) [910/8-1 to S.Wi., STO 431/5-1 to K.J., MA 5082/1 to S.We., in part]; Carl Zeiss Foundation (to M.M.). Funding for open access charge: DFG.
Conflict of interest statement. None declared.

REFERENCES

1. Tang,H., Bowers,J.E., Wang,X., Ming,R., Alam,M. and Paterson,A.H. (2008) Synteny and collinearity in plant genomes. *Science*, **320**, 486–488.
2. Overbeek,R., Fonstein,M., D'Souza,M., Pusch,G.D. and Maltsev,N. (1999) The use of gene clusters to infer functional coupling. *Proc. Natl. Acad. Sci. U.S.A.*, **96**, 2896–2901.
3. Wolf,Y.I., Rogozin,I.B., Kondraskov,A.S. and Koonin,E.V. (2001) Genome alignment, evolution of procaryotic genome organization, and prediction of gene function using genomic context. *Genome Res.*, **11**, 356–372.
4. Korbel,J.O., Jensen,L.J., von Mering,C. and Bork,P. (2004) Analysis of genomic context: prediction of functional associations from conserved bidirectionally transcribed gene pairs. *Nat. Biotechnol.*, **22**, 911–917.
5. Schwartze,V.U., Winter,S., Shelest,E., Marcet-Houben,M., Horn,F., Wehner,S., Linde,J., Valiante,V., Sammeth,M., Riege,K. *et al.* (2014) Gene expansion shapes genome architecture in the human pathogen *Lichtheimia corymbifera*: an evolutionary genomics analysis in the ancient terrestrial Mucorales (Mucoromycotina). *PLOS Genet.*, **10**, e1004496.
6. Uno,T. and Yagiura,M. (2000) Fast algorithms to enumerate all common intervals of two permutations. *Algorithmica*, **26**, 290–309.
7. Heber,S. and Stoye,J. (2001) Algorithms for finding gene clusters. In: *Proceedings of Workshop on Algorithms in Bioinformatics (WABI 2001)*. Springer, Berlin, Vol. **2149**, pp. 254–265.
8. Bergeron,A., Corteel,S. and Raffinot,M. (2002) The algorithmic of gene teams. In: *Proceedings of Workshop on Algorithms in Bioinformatics (WABI 2002)*. Springer, Berlin, Vol. **2452**, pp. 464–476.
9. Tesler,G. (2002) GRIMM: Genome Rearrangements Web Server. *Bioinformatics*, **18**, 492–493.
10. Vandepoele,K., Saeys,Y., Simillion,C., Raes,J. and Peer,Y.V.D. (2002) The automatic detection of homologous regions (ADHoRe) and its application to microcolinearity between *Arabidopsis* and rice. *Genome Res.*, **12**, 1792–1801.
11. Didier,G. (2003) Common intervals of two sequences. In: *Proceedings of Workshop on Algorithms in Bioinformatics (WABI 2003)*. Springer, Berlin, Vol. **2812**, pp. 17–24.
12. Calabrese,P.P., Chakravarty,S. and Vision,T.J. (2003) Fast identification and statistical evaluation of segmental homologies in comparative maps. *Bioinformatics*, **19**(Suppl. 1), i74–i80.
13. Haas,B.J., Delcher,A.L., Wortman,J.R. and Salzberg,S.L. (2004) DAGchainer: a tool for mining segmental genome duplications and synteny. *Bioinformatics*, **20**, 3643–3646.

14. Simillion,C., Vandepoele,K., Saeys,Y. and de Peer,Y.V. (2004) Building genomic profiles for uncovering segmental homology in the twilight zone. *Genome Res.*, **14**, 1095–1106.
15. He,X. and Goldwasser,M.H. (2005) Identifying conserved gene clusters in the presence of homology families. *J. Comp. Biol.*, **12**, 638–656.
16. Pasek,S., Bergeron,A., Risler,J., Louis,A., Ollivier,E. and Raffinot,M. (2005) Identification of genomic features using microsyntenies of domains: domain teams. *Genome Res.*, **15**, 867.
17. Kim,S., Choi,J.-H. and Yang,J. (2005) Gene teams with relaxed proximity constraint. In: *Proceedings of IEEE Computational Systems Bioinformatics Conference (CSB 2005)*. pp. 44–55.
18. Boyer,F., Morgat,A., Labarre,L., Pothier,J. and Viari,A. (2005) Syntons, metabolons and interactons: an exact graph-theoretical approach for exploring neighbourhood between genomic and functional data. *Bioinformatics*, **21**, 4209–4215.
19. Wang,X., Shi,X., Li,Z., Zhu,Q., Kong,L., Tang,W., Ge,S. and Luo,J. (2006) Statistical inference of chromosomal homology based on gene colinearity and applications to *Arabidopsis* and rice. *BMC Bioinformatics*, **7**, 447.
20. Sinha,A.U. and Meller,J. (2007) Cinteny: flexible analysis and visualization of synteny and genome rearrangements in multiple organisms. *BMC Bioinformatics*, **8**, 82.
21. Schmidt,T. and Stoye,J. (2007) Gecko and GhostFam—rigorous and efficient gene cluster detection in prokaryotic genomes. In: Bergman,N (ed). *Comparative Genomics*. Humana Press, Vol. **2**, pp. 165–182.
22. Ling,X., He,X., Xin,D., Han,J. and Han,J. (2008) Efficiently identifying max-gap clusters in pairwise genome comparison. *J. Comput. Biol.*, **15**, 593–609.
23. Ling,X., He,X. and Xin,D. (2009) Detecting gene clusters under evolutionary constraint in a large number of genomes. *Bioinformatics*, **25**, 571.
24. Rödelsperger,C. and Dieterich,C. (2008) Syntenator: multiple gene order alignments with a gene-specific scoring function. *Algorithms Mol. Biol.*, **3**, 14.
25. Rödelsperger,C. and Dieterich,C. (2010) CYNTENATOR: progressive gene order alignment of 17 vertebrate genomes. *PLoS One*, **5**, e8861.
26. Denielou,Y.-P., Sagot,M.-F., Boyer,F. and Viari,A. (2011) Bacterial syntenies: an exact approach with gene quorum. *BMC Bioinformatics*, **12**, 193.
27. Proost,S., Fostier,J., Witte,D.D., Dhoedt,B., Demeester,P., de Peer,Y.V. and Vandepoele,K. (2012) i-ADHoRe 3.0–fast and sensitive detection of genomic homology in extremely large data sets. *Nucleic Acids Res.*, **40**, e11.
28. Doerr,D., Stoye,J., Böcker,S. and Jahn,K. (2014) Identifying gene clusters by discovering common intervals in indeterminate strings. *BMC Genomics*, **15**(Suppl. 6), S2.
29. Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic Local Alignment Search Tool. *J. Mol. Biol.*, **215**, 403–410.
30. Wittkop,T., Emig,D., Lange,S., Rahmann,S., Albrecht,M., Morris,J.H., Böcker,S., Stoye,J. and Baumbach,J. (2010) Partitioning biological data with transitivity clustering. *Nat. Methods*, **7**, 419–420.
31. Böcker,S., Jahn,K., Mixtacki,J. and Stoye,J. (2009) Computation of median gene clusters. *J. Comput. Biol.*, **16**, 1085–1099.
32. Jahn,K. (2011) Efficient computation of approximate gene clusters based on reference occurrences. *J. Comput. Biol.*, **18**, 1255–1274.
33. Jahn,K. (2010) *Approximate Common Intervals Based Gene Cluster Models PhD thesis Technical Faculty*. Bielefeld University, Bielefeld.
34. Jahn,K., Winter,S., Stoye,J. and Böcker,S. (2013) Statistics for approximate gene clusters. *BMC Bioinformatics*, **14**(Suppl. 15), S14.
35. Benjamini,Y. and Hochberg,Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. B*, **57**, 289–300.
36. Franceschini,A., Szklarczyk,D., Frankild,S., Kuhn,M., Simonovic,M., Roth,A., Lin,J., Minguez,P., Bork,P., von Mering,C. *et al.* (2013) STRING v9.1: protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Res.*, **41**, D808–D815.
37. Tatusov,R., Fedorova,N., Jackson,J., Jacobs,A., Kiryutin,B., Koonin,E., Krylov,D., Mazumder,R., Mekhedov,S., Nikolskaya,A. *et al.* (2003) The COG database: an updated version includes eukaryotes. *BMC Bioinformatics*, **4**, 41.
38. Pruitt,K.D., Brown,G.R., Hiatt,S.M., Thibaud-Nissen,F., Astashyn,A., Ermolaeva,O., Farrell,C.M., Hart,J., Landrum,M.J., McGarvey,K.M. *et al.* (2014) RefSeq: an update on mammalian reference sequences. *Nucleic Acids Res.*, **42**, D756–D763.
39. Kopf,M., Klähn,S., Scholz,I., Matthiessen,J.K.F., Hess,W.R. and Voß,B. (2014) Comparative analysis of the primary transcriptome of *Synechocystis* sp. PCC 6803. *DNA Res.*, **21**, 527–539.
40. Knoop,H., Zilliges,Y., Lockau,W. and Steuer,R. (2010) The metabolic network of *Synechocystis* sp. PCC 6803: systemic properties of autotrophic growth. *Plant Physiol.*, **154**, 410–422.
41. Knoop,H., Gründel,M., Zilliges,Y., Lehmann,R., Hoffmann,S., Lockau,W. and Steuer,R. (2013) Flux balance analysis of cyanobacterial metabolism: the metabolic network of *Synechocystis* sp. PCC 6803. *PLoS Comput. Biol.*, **9**, e1003081.
42. Los,D.A., Zorina,A., Sinetova,M., Kryazhov,S., Mironov,K. and Zinchenko,V.V. (2010) Stress sensors and signal transducers in cyanobacteria. *Sensors*, **10**, 2386–2415.
43. Grigorieva,G. and Shestakov,S. (1982) Transformation in the cyanobacterium *Synechocystis* sp. 6803. *FEMS Microbiol. Lett.*, **13**, 367–370.
44. Zang,X., Liu,B., Liu,S., Arunakumara,K.K. and Zhang,X. (2007) Optimum conditions for transformation of *Synechocystis* sp. PCC 6803. *J. Microbiol.*, **45**, 241–245.
45. Mao,X., Ma,Q., Zhou,C., Chen,X., Zhang,H., Yang,J., Mao,F., Lai,W. and Xu,Y. (2014) DOOR 2.0: presenting operons and their functions through dynamic and integrated views. *Nucleic Acids Res.*, **42**, D654–D659.
46. Bischler,T., Kopf,M. and Voss,B. (2014) Transcript mapping based on dRNA-seq data. *BMC Bioinformatics*, **15**, 122.
47. Lyubetsky,V.A., Seliverstov,A.V. and Zverkov,O.A. (2013) Transcription regulation of plastid genes involved in sulfate transport in Viridiplantae. *Biomed. Res. Int.*, **2013**, 413450.
48. Yoshihara,S., Geng,X. and Ikeuchi,M. (2002) pilG Gene cluster and split pilL genes involved in pilus biogenesis, motility and genetic transformation in the cyanobacterium *Synechocystis* sp. PCC 6803. *Plant Cell Physiol.*, **43**, 513–521.
49. Sugita,M., Sugishita,H., Fujishiro,T., Tsuboi,M., Sugita,C., Endo,T. and Sugiura,M. (1997) Organization of a large gene cluster encoding ribosomal proteins in the cyanobacterium *Synechococcus* sp. strain PCC 6301: comparison of gene clusters among cyanobacteria, eubacteria and chloroplast genomes. *Gene*, **195**, 73–79.
50. Teixeira,D.C., Eveillard,S., Sirand-Pugnet,P., Wulff,A., Saillard,C., Ayres,A.J. and Bové,J.M. (2008) The tufB-secE-nusG-rplKAJL-rpoB gene cluster of the liberibacters: sequence comparisons, phylogeny and speciation. *Int. J. Syst. Evol. Microbiol.*, **58**, 1414–1421.
51. Koonin,E.V. and Galperin,M.Y. (1997) Prokaryotic genomes: the emerging paradigm of genome-based microbiology. *Curr. Opin. Genet. Dev.*, **7**, 757–763.
52. Grossman,A.R., Karpowicz,S.J., Heinnickel,M., Dewez,D., Hamel,B., Dent,R., Niyogi,K.K., Johnson,X., Alric,J., Wollman,F.-A. *et al.* (2010) Phylogenomic analysis of the Chlamydomonas genome unmasks proteins potentially involved in photosynthetic function and regulation. *Photosynth. Res.*, **106**, 3–17.
53. Pitt,F.D., Mazard,S., Humphreys,L. and Scanlan,D.J. (2010) Functional characterization of *Synechocystis* sp. strain PCC 6803 pst1 and pst2 gene clusters reveals a novel strategy for phosphate uptake in a freshwater cyanobacterium. *J. Bacteriol.*, **192**, 3512–3523.
54. Ellersiek,U. and Steinmüller,K. (1992) Cloning and transcription analysis of the ndh(A-I-G-E) gene cluster and the ndhD gene of the cyanobacterium *Synechocystis* sp. PCC6803. *Plant Mol. Biol.*, **20**, 1097–1110.
55. Berger,S., Ellersiek,U. and Steinmüller,K. (1991) Cyanobacteria contain a mitochondrial complex I-homologous NADH-dehydrogenase. *FEBS Lett.*, **286**, 129–132.
56. Bartsevich,V.V. and Pakrasi,H.B. (1999) Membrane topology of MntB, the transmembrane protein component of an ABC transporter system for manganese in the cyanobacterium *Synechocystis* sp. strain PCC 6803. *J. Bacteriol.*, **181**, 3591–3593.
57. Osterås,M., Stotz,A., Schmid Nuoffer,S. and Jenal,U. (1999) Identification and transcriptional control of the genes encoding the Caulobacter crescentus ClpXP protease. *J. Bacteriol.*, **181**, 3039–3050.

58. Katoh,H., Hagino,N., Grossman,A.R. and Ogawa,T. (2001) Genes essential to iron transport in the cyanobacterium *Synechocystis* sp. strain PCC 6803. *J. Bacteriol.*, **183**, 2779–2784.
59. Kobayashi,M., Takatani,N., Tanigawa,M. and Omata,T. (2005) Posttranslational regulation of nitrate assimilation in the cyanobacterium *Synechocystis* sp. strain PCC 6803. *J. Bacteriol.*, **187**, 498–506.
60. Aichi,M., Takatani,N. and Omata,T. (2001) Role of NtcB in activation of nitrate assimilation genes in the cyanobacterium *Synechocystis* sp. strain PCC 6803. *J. Bacteriol.*, **183**, 5840–5847.
61. Kopf,M., Klähn,S., Pade,N., Weingärtner,C., Hagemann,M., Voß,B. and Hess,W.R. (2014) Comparative genome analysis of the closely related *Synechocystis* strains PCC 6714 and PCC 6803. *DNA Res.*, **21**, 255–266.
62. Ballal,A., Basu,B. and Apte,S.K. (2007) The Kdp-ATPase system and its regulation. *J. Biosci.*, **32**, 559–568.
63. Wang,T., Shen,G., Balasubramanian,R., McIntosh,L., Bryant,D.A. and Golbeck,J.H. (2004) The sufR gene (sll0088 in *Synechocystis* sp. strain PCC 6803) functions as a repressor of the sufBCDS operon in iron-sulfur cluster biogenesis in cyanobacteria. *J. Bacteriol.*, **186**, 956–967.
64. Willis,L.B. and Walker,G.C. (1999) A novel Sinorhizobium meliloti operon encodes an alpha-glucosidase and a periplasmic-binding-protein-dependent transport system for alpha-glucosides. *J. Bacteriol.*, **181**, 4176–4184.
65. Miyagishima,S.-Y., Wolk,C.P. and Osteryoung,K.W. (2005) Identification of cyanobacterial cell division genes by comparative and mutational analyses. *Mol. Microbiol.*, **56**, 126–143.
66. Mrázek,J., Bhaya,D., Grossman,A.R. and Karlin,S. (2001) Highly expressed and alien genes of the *Synechocystis* genome. *Nucleic Acids Res.*, **29**, 1590–1601.
67. Zhang,Z., Pendse,N.D., Phillips,K.N., Cotner,J.B. and Khodursky,A. (2008) Gene expression patterns of sulfur starvation in *Synechocystis* sp. PCC 6803. *BMC Genomics*, **9**, 344.
68. Rubio,L.M., Flores,E. and Herrero,A. (1998) The narA locus of *Synechococcus* sp. strain PCC 7942 consists of a cluster of molybdopterin biosynthesis genes. *J. Bacteriol.*, **180**, 1200–1206.
69. Schlebusch,M. and Forchhammer,K. (2010) Requirement of the nitrogen starvation-induced protein Sll0783 for polyhydroxybutyrate accumulation in *Synechocystis* sp. strain PCC 6803. *Appl. Environ. Microbiol.*, **76**, 6101–6107.
70. Pascual,A. and Vioque,A. (1996) Cloning, purification and characterization of the protein subunit of ribonuclease P from the cyanobacterium *Synechocystis* sp. PCC 6803. *Eur. J. Biochem.*, **241**, 17–24.
71. Zhang,Y., Rodionov,D.A., Gelfand,M.S. and Gladyshev,V.N. (2009) Comparative genomic analyses of nickel, cobalt and vitamin B12 utilization. *BMC Genomics*, **10**, 78.
72. Calderon,R.H., García-Cerdán,J.G., Malnoë,A., Cook,R., Russell,J.J., Gaw,C., Dent,R.M., de Vitry,C. and Niyogi,K.K. (2013) A conserved rubredoxin is necessary for photosystem II accumulation in diverse oxygenic photoautotrophs. *J. Biol. Chem.*, **288**, 26688–26696.
73. Peschek,G., Löffelhardt,W. and Schmetterer,G. (1999) *The Phototrophic Prokaryotes*, Springer, NY.
74. Cannon,G.C., Heinhorst,S., Bradburne,C.E. and Shively,J.M. (2002) Carboxysome genomics: a status report. *Funct. Plant Biol.*, **29**, 175–182.
75. Fisher,M.L., Allen,R., Luo,Y. and Curtiss,R. 3rd (2013) Export of extracellular polysaccharides modulates adherence of the Cyanobacterium *Synechocystis*. *PLoS One*, **8**, e74514.
76. Shibata,M., Ohkawa,H., Kaneko,T., Fukuzawa,H., Tabata,S., Kaplan,A. and Ogawa,T. (2001) Distinct constitutive and low-CO2-induced CO2 uptake systems in cyanobacteria: genes involved and their phylogenetic relationship with homologous genes in other organisms. *Proc. Natl. Acad. Sci. U.S.A.*, **98**, 11789–11794.
77. Capuano,V., Braux,A.S., Tandeau de Marsac,N. and Houmard,J. (1991) The 'anchor polypeptide' of cyanobacterial phycobilisomes. Molecular characterization of the *Synechococcus* sp. PCC 6301 apce gene. *J. Biol. Chem.*, **266**, 7239–7247.
78. Wiegard,A., Dörrich,A.K., Deinzer,H.-T., Beck,C., Wilde,A., Holtzendorff,J. and Axmann,I.M. (2013) Biochemical analysis of three putative KaiC clock proteins from *Synechocystis* sp. PCC 6803 suggests their functional divergence. *Microbiology*, **159**, 948–958.
79. Hirt,H. and Shinozaki,K. (2004) *Plant Responses to Abiotic Stress, Topics in Current Genetics*. Springer, Berlin.
80. Nærdal,I., Netzer,R., Ellingsen,T.E. and Brautaset,T. (2011) Analysis and manipulation of aspartate pathway genes for L-lysine overproduction from methanol by Bacillus methanolicus. *Appl. Environ. Microbiol.*, **77**, 6020–6026.
81. Daley,S.M.E., Kappell,A.D., Carrick,M.J. and Burnap,R.L. (2012) Regulation of the cyanobacterial CO2-concentrating mechanism involves internal sensing of NADP+ and α-ketogutarate levels by transcription factor CcmR. *PLoS One*, **7**, e41286.
82. Singh,A.K. and Sherman,L.A. (2002) Characterization of a stress-responsive operon in the cyanobacterium *Synechocystis* sp. strain PCC 6803. *Gene*, **297**, 11–19.
83. Yoshihara,S., Geng,X., Okamoto,S., Yura,K., Murata,T., Go,M., Ohmori,M. and Ikeuchi,M. (2001) Mutational analysis of genes involved in pilus structure, motility and transformation competency in the unicellular motile cyanobacterium *Synechocystis* sp. PCC 6803. *Plant Cell Physiol.*, **42**, 63–73.
84. Li,R., Haile,J.D. and Kennelly,P.J. (2003) An arsenate reductase from *Synechocystis* sp. strain PCC 6803 exhibits a novel combination of catalytic characteristics. *J. Bacteriol.*, **185**, 6780–6789.
85. Song,J.-Y., Cho,H.S., Cho,J.-I., Jeon,J.-S., Lagarias,J.C. and Park,Y.-I. (2011) Near-UV cyanobacteriochrome signaling system elicits negative phototaxis in the cyanobacterium *Synechocystis* sp. PCC 6803. *Proc. Natl. Acad. Sci. U.S.A.*, **108**, 10780–10785.
86. Suzuki,I., Kanesaki,Y., Mikami,K., Kanehisa,M. and Murata,N. (2001) Cold-regulated genes under control of the cold sensor Hik33 in *Synechocystis*. *Mol. Microbiol.*, **40**, 235–244.
87. Giner-Lamia,J., López-Maury,L., Reyes,J.C. and Florencio,F.J. (2012) The CopRS two-component system is responsible for resistance to copper in the cyanobacterium *Synechocystis* sp. PCC 6803. *Plant Physiol.*, **159**, 1806–1818.