View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by INRIA a CCSD electronic archive server

CG-seq: a toolbox for automatic annotation of genomes by comparative analysis

Benjamin Grenier-Boley, Antoine de Monte, Helene Touzet

To cite this version:

Benjamin Grenier-Boley, Antoine de Monte, Helene Touzet. CG-seq: a toolbox for automatic annotation of genomes by comparative analysis. [Research Report] RR-7428, INRIA. 2010, pp.12. inria-00530507

HAL Id: inria-00530507
https://hal.inria.fr/inria-00530507
Submitted on 29 Oct 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

CG-seq: a toolbox for automatic annotation of genomes by comparative analysis

Benjamin Grenier-Boley — Antoine de Monte — Hélène Touzet

N° 7428

Octobre 2010

Domaine 5

Rapport de recherche

SN 0249-6399 ISRN INRIA/RR--7428--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
LILLE - NORD EUROPE

CG-seq: a toolbox for automatic annotation of genomes by comparative analysis

Benjamin Grenier-Boley, Antoine de Monte, Hélène Touzet

Domaine : STIC pour les sciences de la vie et de l'environnement
Équipe-Projet SEQUOIA2

Rapport de recherche n° 7428 — Octobre 2010 — 9 pages

Abstract: CG-seq is a software pipeline to identify functional regions such as noncoding RNAs or protein coding genes in a genomic sequence by comparative analysis and multispecies comparison. It takes as input a genomic sequence to annotate and a set of other sequences coming from a variety of species to be compared against the user sequence. The pipeline includes several external software components to perform sequence analysis tasks as well as some new features that were especially developed for the purpose. CG-seq is distributed under the GPL licence. It is available both for command line interface usage or with a Graphical User Interface. It can be downloaded from `http://bioinfo.lifl.fr/CGseq`. A web version can also be runned from this same website for input data of limited length.

Key-words: bioinformatics, computational biology, comparative genomics, genome annotation, gene prediction, noncoding RNAs

CG-seq: annotation automatique de gnomes par analyse comparative

Résumé : CG-seq est une suite logicielle qui permet l'identification de régions fonctionnelles, telles que les ARN non-codants ou les gènes codants, dans une séquence génomique en utilisant le principe de la génomique comparative et de la comparaison entre espèces. Il prend en entrée une séquence à annoter, ainsi que d'autres séquences de référence issues de différentes espèces, et retourne en sortie une liste de régions candidates, avec leur annotation. CG-seq intégre plusieurs logiciels d'analyse de séquences existants, ainsi que de nouveaux modules qui ont été développés spécifiquement pour ce travail.

CG-seq est distribué sous licence GPL, et téléchargeable à `http://bioinfo.lifl.fr/CGseq`. Il est disponible pour une utilisation en ligne de commande ou avec une interface graphique. Une version web est également proposée sur ce même site, qui permet de tester CG-seq sur des séquences de longueur raisonnable.

Mots-clés : bioinformatique, génomique comparative, annotation de génomes, prédiction de gènes, ARN non-codants

1 Introduction

More and more newly sequenced genomes are becoming available every week. In this context, sequence annotation is an essential step in understanding the genome and the transcriptome of a species [2]. Comparative genomics has proven to be a fruitful framework to address this problem. The rationale of this paradigm is that functional elements are under a positive selection pressure and therefore should be better conserved than other sequences. This gives a way to detect these elements by searching for sequences showing some degree of similarity across species. The function of these elements can be further investigated by inspection of mutation patterns. Furthermore, the completion of whole genome sequencing projects for species at the appropriate evolutionary distance makes this approach effective in practice.

Annotation by comparative analysis typically involves several computational steps that require some expertise: Aligning the sequences to identify conserved regions, combining conserved regions, analysing these regions to detect an evolutionary pattern that is representative of the selection pressure. In this paper, we present an automatic pipeline, called CG-seq, that allows the user to perform all these tasks in an integrated manner. It gathers several tools to allow easy and flexible annotation of genomes by comparative analysis.

2 Method

2.1 General overview

The pipeline takes as input a query sequence to be annotated and a set of other species that will be used for comparison. The query sequence is typically a chromosome or a contig. Several sequences and possibly several strains can be provided for the same species.

The output is a set of candidate regions on the query sequence that are likely to be protein-coding regions, or noncoding structured RNAs. To achieve this task, CG-seq proceeds in four steps.

1. *Preprocessing.* Sequences are preprocessed to mask CDSs (optional).
2. *Alignment.* The query sequence is compared to all other sequences to detect similar sequences across species. The result is a collection of local alignments between the query sequence and the other sequences.
3. *Conserved regions.* Pairwise alignments are combined into clusters of significantly conserved regions.
4. *Classification.* Each cluster of conserved regions is submitted to RNA structure inference program tools, that search for a consensus secondary structure, and to protein-coding prediction tools that search for a significantly conserved amino-acid sequence.

Figure 1 gives a flowchart of the method. We describe each step in further details in the remaining in this Section.

2.2 Preprocessing sequences

Before comparing all sequences, CG-seq allows the user to perform some optional preprocessing on his data.

Mask known CDSs/Mask known RNAs: This option is useful when one wants to focus on the discovery of functional elements in intragenic regions, or to eliminate usual noncoding RNAs such as tRNAs or rRNAs. It is possible to mask annotated elements either in the query sequence, or in some other sequences, or in all sequences. A GENBANK file should be specified for each masked sequence. Masked elements are CDSs or noncoding RNAs. For each genome, the CDSs

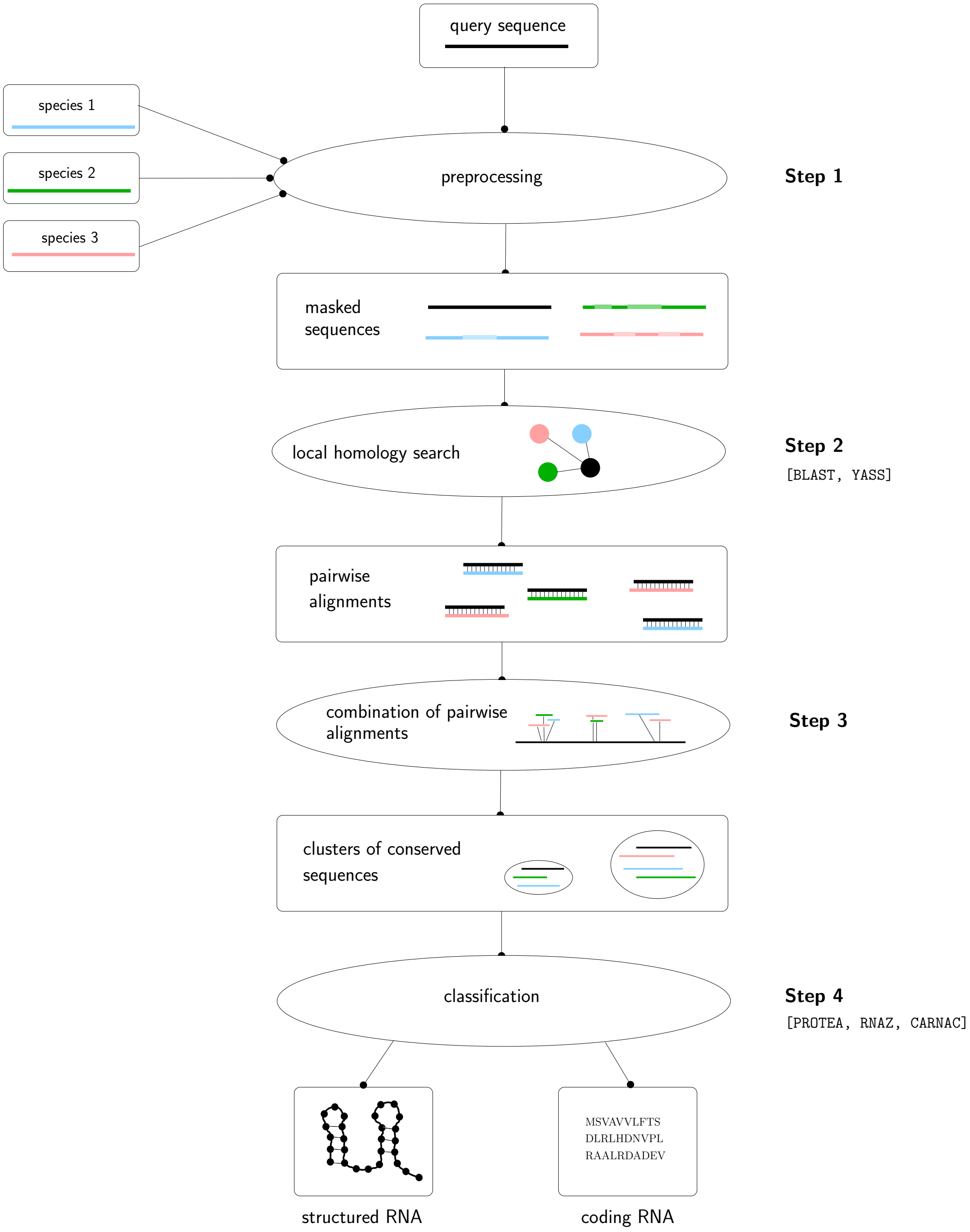

query sequence
species 1
species 2
species 3
preprocessing
Step 1
masked sequences
local homology search
Step 2
[BLAST, YASS]
pairwise alignments
combination of pairwise alignments
Step 3
clusters of conserved sequences
classification
Step 4
[PROTEA, RNAZ, CARNAC]
MSVAVVLFTS
DLRLHDNVPL
RAALRDADEV
structured RNA
coding RNA

Figure 1: Four main steps of CG-seq

correspond to features annotated as CDS in the GENBANK file, and noncoding RNAs correspond to features annotated as *tRNA*, *rRNA* and *misc_RNA* in the GENBANK file.

Clean up redundancy: It is possible to provide sequences from several strains for a same species. In this case, it is possible to eliminate redundant parts between all sequences within a same species. Redundancy is defined as two identical regions between two sequences that have an identity percentage greater than the selected threshold (default 98%) and a length greater than the minimal length (default 100 bp). For each species, only one copy of the redundant local sequence is kept. So only one copy of the redundant local sequence will be used int the classification step of the method (Section 2.5).

Note that we do not propose any option to mask low complexity regions. This can be done automatically at the following step by Blast or Yass with the appropriate parameter setting.

2.3 Pairwise alignments

In this step, the (possibly masqued) query sequence is compared against all other sequences. This is done by local similarity search. We propose two such tools: Blast [1] and Yass [5]. Both programs implement a heuristics for local alignment. The difference between Blast and Yass is that Blast is based on contigous k-mers, and Yass on subset spaced-seeds that are known to achieve higher sensitivity. Blast tends to be faster than Yass .

The result is a collection of pairwise alignments between the query sequences and other sequences.

2.4 Cluster algorithm

The set of pairwise alignments gives a rough picture of the similarity landscape for the query sequence. In this landscape, the idea is that conserved regions are supported by a high number of alignments. The number of alignments alone may not always be a satisfying criterium to infer conserved regions. The species involved in the alignments are also relevant. For example, one can wish that alignments coming from species with a high evolutionary distance to the query sequence are considered as more significant that alignments coming from species that have a small evolutionary distance to the query sequence. So the algorithm should give a higher weight to sequences with a poor global similarity. It is also advisable that the method shows flexibility and is able to select only clusters that contain sequences from given species, or on the contrary that do not contain a selection of given species.

The method we propose meets these requirements. It relies on a three-step algorithm. First, each position of the query sequence is assigned a *position specific score* that depends on the set of species having a matching alignment at that position. Then, we recover *conserved regions* on the query sequences as local regions having a high position specific score. Lastly, we construct *clusters of conserved regions* that are composed of a conserved region from the query sequences, and a set of similar sequences coming from other species.

Position specific score. For each position i of the query sequence, PPS_i is the score at that position. To compute PPS_i, we consider two models.

- Target model: this model describes the probability to observe an alignment at any given position between the query sequence and sequences from species s in conserved regions,
- Background model: this model describes the probability to observe an alignment at any given position between the query sequence and sequences from species s by chance.

PPS_i will be defined as the log of the ratio between the probability to belong to the target model, normalized by the probability to belong to the background model. So it will be positive when the probability of the position i to belong to the target model is higher than the probability of position i to belong to the background model. Otherwise, it will be negative.

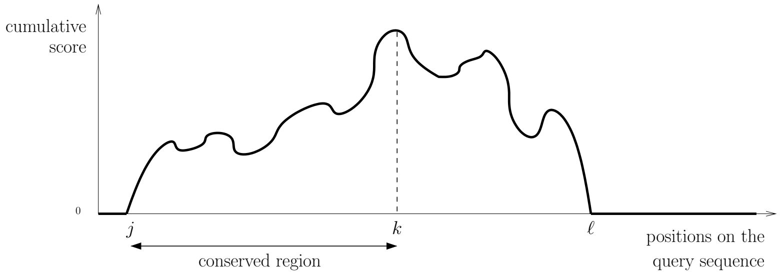


Figure 2: Utilization of the position specific score to identify conserved regions.

How to compute each probability value? For the background model, we first compute for each species s the proportion of positions of the query sequence that are involved in an alignment between the query sequence and a sequence from s: p_s $(0 \leq p_s \leq 1)$. This value gives a measure of similarity between the query sequence and sequences for species s. It corresponds to the expected probability to observe an alignment between the species s and the query sequence at a given position.

For the target model, we assume that we are given a parameter t_s $(0 \leq t_s \leq 1)$ that describes the expected proportion of positions of the query sequence that are involved in an alignment with a sequence from species s in conserved regions of the query sequence.

We make one more assumption: Observations coming from different species are mutually independent. This allows us to state the following formula. For any position i on the query sequence, we define PPS_i as

$$PPS_i = \sum_{s \text{ matches at position } i} \log\left(\frac{t_s}{p_s}\right) + \sum_{s \text{ does not match at position } i} \log\left(\frac{1-t_s}{1-p_s}\right)$$

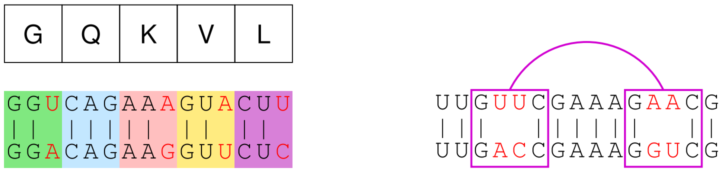
It is obvious that t_s is a critical parameter of the method, since it describes the target models. It can be tuned by the user. By default, t_s is set to $\max\{0.8, 0.5 + p_s/2\}$. It can be higher or lower depending on the level of conservation with species s expected in clusters. It can also be used to give diversified weights to species. For example, setting $t_s = 1$ will induce that only conserved regions showing at least one alignment with species s will be reported (mandatory species). Analogously, setting $t_s = 0$ will induce that conserved regions should not exhibit any alignment with species s (prohibited species).

One last point worth mentioning is that the score does not take into account the number of alignments matching at a given position coming from a same species. It depends only on the existence of at least one alignment. This guarantees that repeated sequences are not overrepresented in the process of formation of conserved regions. This also guarantees that several strains can be provided for a same species without creating a bias in the inference of conserved regions.

Identification of the conserved regions on the query sequence. Conserved regions are obtained as local regions on the query sequence that exhibit a high position specific score. To this end, we define the cumulative score S_i

$$\begin{cases} S_0 &= 0 \\ S_i &= \max\left(0, S_{i-1} + PSS_i\right) \end{cases}$$

where i is a position on the query sequence. A conserved region on the query sequence is defined as a pair of positions (j, k) on the query sequence such that $j < k$, $S_j = 0$ and S_k is the maximal value on the interval $[j..\ell]$ where ℓ is the smallest position after j such that $S_\ell = 0$ (see Figure 2).



protein-coding sequence

conserved RNA secondary structure

Figure 3: Classification of clusters of sequences according to their mutational schemes. The alignment on the left shows mutations that are typical of protein-coding regions: All mutations are silent mutations and the amino-acid sequence is perfectly conserved. The alignment on the right shows mutations that are typical of a conserved secondary structure: The mutations do not break the base pairings.

Construction of clusters. Finally, for each conserved region (j, k) we construct the associated *cluster of conserved regions*. Formally, a cluster is composed by a conserved region on the query sequence and a set of similar regions coming from other genomes. To do that, we consider all pairwise alignments involving at least one position of the conserved regions. For each such alignment, we realign the conserved region properly against the other sequences to refine the bounds of matching sequences. We retain only sequences above a given length. It is also possible to specify a minimal number of sequences, a maximal number of sequences, a maximal identity percentage, a minimal identity percentage, a maximal number of sequences per species, a minimal number of different species, a conserved secondary structure between the user sequence and any other sequence. Doing this, we obtain a high-quality cluster of multi-species similar sequences for each conserved region of the query sequence.

2.5 Classification of clusters

In the last step of the method, each cluster is inspected individually to see whether the sequences show an evolutionary pattern indicative of protein-coding regions or structured regions. See Figure 3

Protein coding regions. We use PROTEA that implements an evolutionary model for protein-coding sequences [3]. The idea of PROTEA is that the selection pressure tends to preserve the encoded amino acid sequence, and it is possible to identifies coding sequences by looking for a global conservation of common reading frames. The method first identifies best potential reading frames from each pair of sequences, and then incorporates this information into a frame graph from which a coding significancy score is calculated. By doing so, it also predicts the associated reading frame for each sequence, and the asociated amino-acid sequence.

Structured RNA. The underlying principle to identify RNAs is that mutations observed between homologous structured RNA sequences should be consistent with the formation of a conserved consensus secondary structure. CARNAC [6] or RNAz [7] can be selected for this task. RNAz is known to be well-suited to process clusters of highly similar sequences, and relies on precomputed multiple sequence alignments (built with ClustalW here). CARNAC shows a better specificity when sequences are hard to align accurately [4].

2.6 Parameter settings

CG-seq allows to perform multiple tasks and involves several modules. Each of these tasks and modules is parameterizable. CG-seq comes with default parameter values for each of them. It is also possible for the user to set its own parameter values using advanced parameter setting.

3 Implementation and Availability

CG-seq is available in three versions: Command line interface, graphical user interface and web form. It can be found at `http://bioinfo.lifl.fr/CGseq`

3.1 Command Line Interface

The CG-seq archive contains the following tools: CGseqcore for the creation of cluster of conserved sequences, YASS for local homology search, CARNAC for secondary structure prediction, protea for protein coding gene identification. You also need to install the ClustalW software for multiple sequence alignment (not provided in the archive). Optionnaly, CG-seq can be interfaced with several external tools: BLAST to be used complementarily or alternatively to YASS for local homology search, RNAz to be used complementarily or alternatively to CARNAC for secondary structure prediction, RNAPlot for producing 2D drawings of RNA secondary structures. Each software tools is coupled with a configuration file, that allows the user to define its own parameter setting.

Once the computation is completed, result is available both as a HTML page and as a CSV file (that can easily parsed, or opened and modified with any spreadsheet, such as Excel). Result is a list of putative functional regions: Protein-coding regions or RNA structured regions. For each prediction, the following information is available: position, strand, sequence, position and names of flanking genes (if GENBANK files were provided), clusters of related similar sequences, predicted amino-acid sequence or consensus secondary structure. All intermediate files needed for the computation are also kept for the user.

3.2 Graphical User Interface

It allows to run CG-seq through a user-friendly interface and to enter parameters step by step. Results file are identical as those obtained with the command line interface.

3.3 Web interface

It offers the same facilities as the GUI. It is well-suited for occasional use and for small sequences, such as bacterial genomes.

3.4 Requirements

Perl is required both for the command line and graphical user interfaces. CG-seq has been tested under version >= 5.8. You also need a C compiler to build CG-seq, as well as flex and bison libraries. JAVA is required for the graphical user interface. It is recommended to use a version >= 1.6 update 10 to enjoy all functionalities.

The web form interface does not require any prior installation. It is W3C compliant.

References

[1] S.F. Altschul, T.L. Madden, A.A. Schffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, 25:3389–3402, 1997.

[2] M.E. Dinger, K.C. Pang, T.R. Mercer, and J.S. Mattick. Differentiating protein-coding and noncoding RNA: challenges and ambiguities. *PLos Computational biology*, 4(11), 2008.

[3] A. Fontaine and H. Touzet. Computational identification of protein-coding sequences by comparative analysis. *International Journal of Data Mining and Bioinformatics*, 3(2):160–176, 2009.

[4] P. Gardner and R. Giegerich. A comprehensive comparison of comparative RNA structure prediction approaches. *BMC Bioinformatics*, 5(140), 2004.

[5] L. Noé and G. Kucherov. YASS: enhancing the sensitivity of DNA similarity search. *Nucleic Acid Research*, 33:W540–W543, 2005.

[6] H. Touzet and O. Perriquet. CARNAC: folding families of related RNAs. *Nucleic Acids Research*, 32 (Supplement 2):142–145, 2004.

[7] S. Washietl, I.L. Hofacker, and P.F. Stadler. Fast and reliable prediction of noncoding RNAs. *Proceedings of the National Academy of Sciences of the United States of America*, 102(7):2454–2459, 2005.

Centre de recherche INRIA Lille – Nord Europe
Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399