# NcDNAlign: Plausible Multiple Alignments of Non-Protein-Coding Genomic Sequences

Dominic Rose [a], Jana Hertel [a], Kristin Reiche [a], Peter F. Stadler [a,b,c], Jörg Hackermüller [d,*]

[a] *Bioinformatics Group, Department of Computer Science, University of Leipzig, Härtelstraße 16-18, D-04107 Leipzig, Germany*

[b] *Department of Theoretical Chemistry University of Vienna, Währingerstraße 17, A-1090 Wien, Austria*

[c] *Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM 87501, USA*

[d] *Fraunhofer Institute for Cell Therapy and Immunology — IZI Deutscher Platz 5e, D-04103 Leipzig, Germany*

**Abstract**

Genome-wide multiple sequence alignments (MSAs) are a necessary prerequisite for an increasingly diverse collection of comparative genomic approaches. Here we present a versatile method that focuses on generating high-quality MSAs for non-protein-coding sequences. The `NcDNAlign` pipeline combines pairwise `BLAST` alignments to initial MSAs which are then locally improved and trimmed. The program is optimized for speed and hence is well-suited in particular for pilot-studies. We demonstrate the practical use of `NcDNAlign` in three case studies: the search for ncRNAs in gammaproteobacteria and the analysis of conserved non-coding DNA in both nematodes and teleost fish, in the latter case focusing on the fate of duplicated ultra-conserved regions.

Compared to the currently widely used genome-wide alignment program `TBA` we achieve an up to 20 to 30-fold reduction of CPU-time necessary to generate gammaproteobacterial alignments. A showcase application of bacterial ncRNA prediction based on alignments of both algorithms results in similar sensitivity, false discovery rates and up to hundred putatively novel ncRNA structures. Similar findings hold for our application of `NcDNAlign` to the identification of ultra-conserved regions in nematodes and teleosts. Both approaches yield conserved sequences of unknown function, result in novel evolutionary insights expanding our knowledge of DNA conservation patterns among these genomes, and manifest benefits of an efficient and reliable genome-wide alignment package.

The software is available under the GNU Public License at `http://www.bioinf.uni-leipzig.de/Software/NcDNAlign/`.

*Key words:* Non-coding RNA, ncRNA, alignment, multiple sequence alignments, ultra-conserved elements, ultra-conserved regions, UCE, UCR, CNE, genome annotation, comparative genomics

## 1 Introduction

The construction of genome-scale alignments is the first crucial step in many comparative genomic applications. Certain questions can be treated based on pairwise alignments. However, an increasing number of analysis pipelines depends on multiple sequence alignments (MSAs), or at least profits profoundly from the additional information contained in MSAs. Examples include the analysis of evolutionary constraint [1], the discovery and assessment of functional, and in particular regulatory, sequences [2–5], the prediction of protein-coding genes [6], and *de novo* searches for non-coding structured RNAs [7,8].

Several software packages have been developed for generating genome-wide MSAs — a task that necessarily needs to be automated to a large extent because of the sheer amount of data: `CHAOS` [9], `Pecan` (`http://www.ebi.ac.uk/~bjp/pecan/`), `MAvid` [10], `MLagan` [11], `Mulan` [12], and the **T**hreaded **B**lockset **A**ligner (`TBA`) [13] are probably the most commonly used tools for this task. Each comes with its specific advantages and disadvantages, and most tools were designed with a specific set of applications in mind. Many of them, e.g. `Pecan` or `TBA`, rely on exonic anchors guiding the alignment process. Note, that we use the term `TBA` generically for all combinations of `blastz` and `MultiZ` variants. For recent reviews on the genomic multiple alignment problem we refer to [14,15].

As genomic sequence data become available at an ever-increasing rate (the ENCODE project, for example considers 22 vertebrate genomes [16]), the construction of genome-wide MSAs tends to become the computational bottleneck often requiring large computer clusters [17]. The necessary computational resource requirements are often prohibitive in practice, at least for exploratory studies and for repeating earlier analysis when updated or additional genomic sequences become available.

In this contribution, we describe the light-weight but flexible multiple analysis

* Corresponding author. Fax: +49 341 3550 855
*Email addresses:* `dominic@bioinf.uni-leipzig.de` (Dominic Rose), `jana@bioinf.uni-leipzig.de` (Jana Hertel), `kristin@bioinf.uni-leipzig.de` (Kristin Reiche), `stadler@bioinf.uni-leipzig.de` (Peter F. Stadler), `joerg.hackermueller@izi.fraunhofer.de` (Jörg Hackermüller).

pipeline NcDNAlign. Our approach is specifically geared towards constructing MSAs of non-repetitive non-protein-coding genomic DNA. We demonstrate the applicability of NcDNAlign to both prokaryotic and eukaryotic genomes. As examples, we compute alignments of several bacterial groups, of five nematode, and four teleost fish genomes.

Ultra-conserved elements (UCRs) are genomic regions which are shared between several species with 100 % sequence identity. UCRs particularly have been studied in vertebrates [18–20] and insects [21–23]. This unexpectedly [24] high level of sequence conservation implies that they are most likely a result of strongly constraining, stabilizing selection due to their functional importance [25]. In [26] an example is described in which multiple distinct functional constraints make the accumulation of substitutions impossible. In general, however, the detailed function of UCRs remains mysterious even though functional studies associated them primarily with binding sites for regulatory factors, RNA processing and the regulation of transcription and development [27]. As an exemplary application of NcDNAlign we investigate the fate of conserved non-coding DNA in general and UCRs in particular in the aftermath of the teleostean genome duplication.

Many non-protein-coding RNAs (ncRNAs), as well as certain regulatory features in mRNAs such as IRES or SECIS elements, require specific secondary structures for their function [28–34]. Hence, RNA secondary structure is conserved over evolutionary time-scales while the underlying sequences accumulate substitutions. These properties can be explored by computational methods such as QRNA [35], RNAz [36], or EvoFold [8] to identify regions with stabilizing selection on RNA structure within a sequence alignment. In all these studies it has become clear that the quality of the input alignments is a limiting factor for sensitivity and specificity of ncRNA detection.

The purpose of NcDNAlign is to provide a *user-friendly* and *efficient* method for generating MSAs of genomic DNA. Thereby, the term "user-friendly" unites usability, flexibility, and scalability. As a major design feature, NcDNAlign can *a priori* restrict the alignment process to a user-defined subset of annotation features, e.g. to introns, or exclude a subset of annotation features from the alignment process. This makes NcDNAlign particularly suitable for quick "pilot-studies" in numerous applications. The second design goal of NcDNAlign is to produce alignments that can be directly used as input data for a particular subsequent application, e.g. ncRNA gene finding. Other approaches like TBA provide alignments with a maximal coverage of the input genomes, which typically involves extensive post-processing of the alignments prior to e.g. an application of RNAz. NcDNAlign, in contrast, is geared towards calculating solely alignments which are sufficiently reliable – where the meaning of *reliable* can be defined in a configuration file – for a particular application. This implies, that compared to TBA, NcDNAlign can rely on less sensitive tools

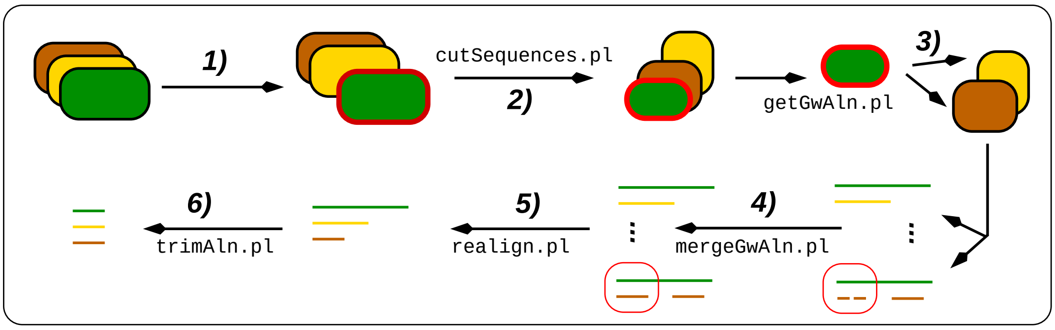


Fig. 1. Work flow of `NcDNAlign`.

1) One species out of all given genomic sequences has to be selected as reference. 2) Optionally, sequences are pruned of potentially interfering or uninteresting sequence stretches, reducing the data set to genomic sub-sequences. 3) All sub-sequences of the reference are compared to all sub-sequences of all other species and local alignments are calculated heuristically (`BLAST`). 4) Adjacent compatible hits are combined. 5) The best hits (`E-value`) of each organism for each sub-sequence of the reference are aligned (`DIALIGN`). 6) Finally, the alignments are pruned, poorly aligned sequences are removed and the remaining sequences are optionally realigned to obtain an optimal alignment.

for identifying homologous sequences and that the amount of data that has to be processed is significantly reduced, both resulting in considerable reduction of the computational effort. Thereby, multiple vertebrate genomes can be analyzed in principle on one single small computer, rather than a huge parallel computing cluster as used a few years ago for the first genome-wide human-mouse alignments [17].

## 2 Results

### *NcDNAlign*

`NcDNAlign` is implemented as a pipeline that connects external programs and several custom tools (implemented in `Perl`). The overall layout is summarized in Figure 1. For algorithmic details we refer to section 4 at the end of this paper. Beside various command line parameters, the pipeline is controlled by a configuration file that can be flexibly adjusted to very different analysis projects. One of the input genomes has to be declared as the *reference* for the `NcDNAlign` pipeline. In the first step, subsets of the genomic input sequences are compiled, based on user-defined rules which allows to in- or exclude certain annotated genomic features from the analysis. This yields sequence fragments and all stretches of the reference genome are aligned to all other genomic sequences using `blastn` [37]. This is in contrast to `TBA` that uses the more sensitive `blastz` to compare all versus all sequences. Consistent adjacent hits are combined into an MSA using a maximum clique approach [38], which is derived from the `tracker` program [39]. At this stage, regions that are considered to align validly correspond with the term "blocks" of the `TBA` vocabulary. If desired, flanking sequences to each of the initial `BLAST` hits can be incorporated into the construction of this MSA. Empirically, we found that these first-stage MSAs tend to include many gap-rich regions or individual non-related sequences. We therefore developed heuristics to trim the alignments and to discard poorly aligned sequences. Finally, these sequences can be re-aligned using `ClustalW` to obtain an optimal global alignment.

In the remainder of this section we briefly outline three example-applications of `NcDNAlign`. Using a single computer (standard hardware, Intel Xeon 2.80 GHz CPU, 1 GB RAM), obviously depending on actual genome numbers, genome sizes and applied parameters, `NcDNAlign` allows to generate some hundred bacterial alignments out of 24 Mb genomic sequence data in a few minutes, ∼49 300 nematode alignments in approximately 2-3 days processing 753 Mb sequence data, and ∼63 500 teleostean alignments based on 3.4 Gb input in 2-3 weeks. Applying `NcDNAlign` to the genomes of nine nematode species (data not shown) comprising an input data set of 6.7 Gb (due to several unfinished genomes) shows that the pipeline is also capable of handling mammalian sized problems – the computation takes 2-3 weeks depending on parameters. Figure 2 illustrates the CPU-time necessary to align these three exemplary sets. Compared to `TBA` `NcDNAlign` is 20-30 fold faster in bacteria. In the teleost and the nine nematode examples, `TBA` has been terminated after exceeding the total run-time of `NcDNAlign` two-fold, in both cases the `blastz` phase was still ongoing.

*Estimates of alignment quality*

As outlined above, `NcDNAlign` employs several strategies to confine the produced alignments to a set with sufficient quality for specific downstream applications. We therefore monitor the efficiency and trade-off of `NcDNAlign`'s alignment beautification heuristics by comparing quality and coverage of alignments produced by `TBA` and `NcDNAlign`. We use alignments of bacterial genomes which are aimed at ncRNA gene finding. Unfortunately, there is no well established benchmark for the performance of genomic alignment tools. Approaches like `BAliBASE` [40] or `BRAliBase II` [41] are of limited use as they only reflect the MSA step. In addition to coverage and the fraction of gaps in the alignments, which is important for ncRNA gene finding, we use a simple Sum-of-Pairs score, to asses alignment quality:

$$\sigma = \frac{1}{n-1} \sum_{x,y \in \mathbb{A}} \frac{1}{\ell(\mathbb{A})} \sum_{i=1}^{\ell(\mathbb{A})} \delta(x_i, y_i) \qquad (1)$$

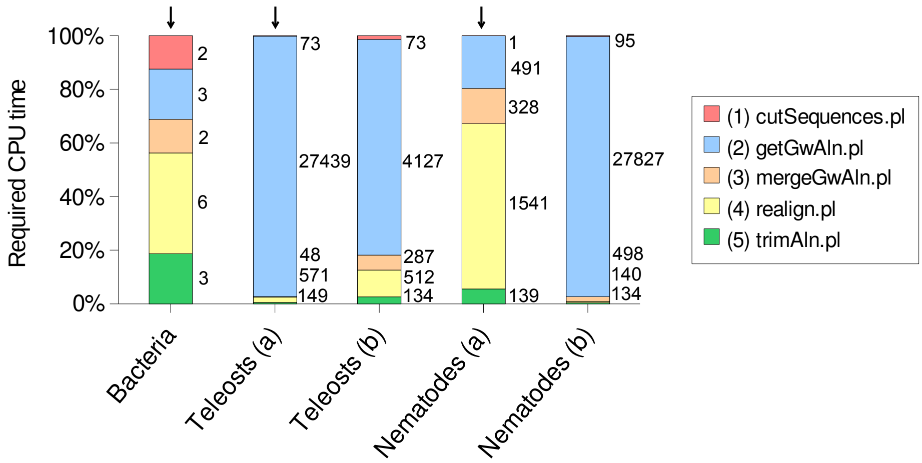


Fig. 2. Benchmark of `NcDNAlign`.

The figure illustrates the CPU-time in percentages respectively minutes necessary to align five different sets of bacterial, teleostean, and nematode genomic sequences for each step of the `NcDNAlign` pipeline using a single CPU. The bacterial screen comprises six genomes respectively 24 Mb sequence data. In case of teleosts we (a) aligned all five teleosts (3.4 Gb) and (b) all teleosts (1.9 Gb) except Danio, the largest and most distantly related genome. In the latter case protein-coding regions and repeats have moreover been excluded from the input sequences. In case of nematodes we (a) aligned the five nematodes (753 Mb) available at the `UCSC Genome Browser` and (b) a mixture of nine nematodes (6.7 Gb), inclusive several draft assemblies provided by the `NCBI TraceDB` (`ftp://ftp.ncbi.nih.gov/pub/TraceDB/`). The **`blastn`** searches (*blue*, **`getGwAln.pl`**) are obviously the most time-consuming step for huge, repeat-interspersed data sets (e.g. unfinished draft assemblies). Contrary, **`blastn`** is less predominant while processing well formatted (e.g. repeat-masked), high-quality assemblies and the computation of MSAs (*yellow*, **`realign.pl`**), especially for closely related and hence well aligning organisms, might get time-consuming. However, measured run-times are subject to individually chosen program-parameters and particular environmental conditions and therefore have to be addressed carefully (e.g. system load, network traffic, . . . , may strikingly vary in a multi-user environment). Screens labeled with arrows are discussed in more detail in the main text.

where $\delta(p, q)$ is Kronecker's delta and $\ell(\mathbb{A})$ denotes the length of the alignment $\mathbb{A}$. For the entire genomic alignment we measure the quality as the length-weighted average (further referred to as WSoP = **W**eighted **S**um-of-**P**airs) of all individual local multiple sequence alignments. As expected, we find a reduced coverage of the reference genome in `NcDNAlign` compared with `TBA`-alignments (Table 2), however, at the benefit of significantly longer alignments with a reduced fraction of gaps and overall slightly improved WSoP score (Table 1).

*Prediction of novel bacterial structured non-coding RNAs*

Both, `EvoFold` and `RNAz`, two widely used approaches to the prediction of ncRNAs in regions conserved on sequence level, rely on high-quality MSAs

Table 1
Comparison of bacterial alignments produced by `NcDNAlign` and `TBA`.
Results of `NcDNAlign` using four different sets of parameters are shown; the detailed modified `BLAST` parameters are given in section 4. `TBA` aligns more nucleotides in total than `NcDNAlign`, however the aligned reference sequence contains significantly more gaps. WSoP scores are largely comparable for both programs with a slight advantage for `NcDNAlign`. `NcDNAlign` is more efficient in terms of run-time and significantly outperforms `TBA` independent of the chosen parameter set. Overall, `NcDNAlign` efficiently generates long and gap-reduced alignments, when taking `TBA` as a reference.

| | `NcDNAlign` default Blast | | `NcDNAlign` modified Blast | | TBA |
|---|---|---|---|---|---|
| Flanking regions | + | - | + | - | n.a. |
| | (1) | (2) | (3) | (4) | (5) |
| # alignments | 169 | 169 | 542 | 499 | 1347 |
| aln.-length | 35 576 | 29 784 | 132 761 | 116 484 | 247 056 |
| mean length | 211 | 176 | 245 | 233 | 183 |
| mean # seqs | 4.11 | 3.85 | 3.65 | 3.56 | 3.27 |
| # gaps | 425 | 191 | 2 867 | 2024 | 10 562 |
| # gaps / column | 0.0119 | 0.0064 | 0.0216 | 0.0174 | 0.0427 |
| WSoP | 3.1089 | 3.1416 | 2.7082 | 2.7719 | 2.6464 |
| Elapsed CPU-time [min] | 15.68 | 14.35 | 29.27 | 27.52 | 548.36 |

as input data. We have selected the prediction of bacterial ncRNAs using `RNAz` in alignments of bacterial genomes as a showcase application to compare `TBA` and `NcDNAlign` for two reasons: ($i$) Alignments of bacterial genomes pose a sufficiently compact problem, allowing to experiment with different alignment parameter sets and an easy comparison with the computationally demanding `TBA` approach. ($ii$) Computational studies of bacterial ncRNA prediction are of wide interest [42–46]. While `RNAz` has been extensively applied to eukaryotic organisms, to vertebrates [7], drosophilids [47], nematodes [38], and urochordates [48], there is only one published application to bacteria [49]. Hence, we explored several bacterial families with qualitatively similar results in comparison to `TBA`. Here we display a screen in gammaproteobacteria, since their reference organism *Escherichia coli* is well annotated, allowing to reliably estimate sensitivity values of the compared methods. `NcDNAlign` and `TBA` have been applied to the genomic sequences of *E. coli* (NC_007146, 4.7 Mb), *Haemophilus influenzae* (NC_000913, 1.9 Mb), *Legionella pneumophila* (NC_002942, 3.3 Mb), *Salmonella typhimurium* (NC_003197, 4.9 Mb), *Shigella flexneri* (NC_004741, 4.6 Mb), and *Yersinia pestis* (NC_004088, 4.6 Mb). Configured to remove annotated coding and repetitive sequences, NcDNAlign yields between 169 (standard BLAST parameters) and 542 alignments (using flanking regions and adapted BLAST parameters), corresponding to 0.6 resp. 2.7 % of the aligned reference genome (Table 2-b). In contrast, while processing exactly the same input sequences (whole genome minus coding sequences and repeats), TBA yields a considerably larger number of 1 437 alignments corresponding to 5.1 % of the *E. coli* genome. For comparability reasons we only consider TBA alignments that match the NcDNAlign requirements of a minimal alignment length of 40 nt, at least three sequences per alignment, and the reference organism must be part of the alignment. The vast majority of alignments produced by NcDNAlign is also included in the TBA-alignments, albeit with different boundaries (e.g. 479 out of 542 NcDNAlign-alignments overlap with TBA-alignments for at least 70 % of the nucleotides in the shorter alignment). To generate these alignments TBA requires 20 to 30-fold more CPU-time than NcDNAlign utilizes (548 versus 16, respectively 29 minutes), which corresponds to a performance of 451 nt per minute for TBA and 4 578 nt per minute for NcDNAlign (Table2-a).

In line with the number of retrieved alignments, TBA-alignments yield 658 (488) RNAz hits at a prediction score of $p > 0.5$ ($p > 0.9$) threshold as opposed to 339 (280) for NcDNAlign (Table 2-c). A BLAST search (E-value <=1e-5) reveals that 123 of our and only 118 of the TBA hits match Rfam entries. Furthermore, 25 (NcDNAlign), respectively 24 (TBA), RNAz hits align with Noncode entries and 28 (NcDNAlign), respectively 31 (TBA), align with ncRNAdb sequences. Saetrom *et al.* [43] published 156 *E. coli* ncRNAs, of which we recover 74 % by RNAz on NcDNAlign-alignments (114/154 ncRNAs by 178 RNAz hits). Contrary, only 56 % of Saetrom's ncRNAs are identified in TBA-alignments (88/156 ncRNAs by 152 RNAz hits). False discovery rates, measured as the number of positive RNAz predictions in column-wise shuffled alignments over positive predictions in the normal data set, are comparable for both approaches (0.25 for TBA versus 0.24 – 0.33 for NcDNAlign). Irrespective of the number of RNAz hits, the overall sensitivity for detecting known RNA genes of both methods is almost equal – 0.60 for NcDNAlign versus 0.59 for TBA (Table 2-d).

*Non-coding ultra-conserved regions in nematode genomes*

In vertebrates and insects ultra-conserved regions have been studied in great detail in multi-way alignments [18–23]. In a first pilot-study Siepel *et al.* analysed UCRs in nematodes based on pairwise alignments only [21]. Recently, Vavouri *et al.* [50] retrieved 990 conserved non-coding elements (CNEs) from consecutive pairwise alignments of three nematodes computed by Megablast. The UCSC Genome Browser provides access to TBA/MultiZ-generated MSAs of five nematodes (http://hgdownload.cse.ucsc.edu/goldenPath/ce4/multiz5way/): *C. elegans* (ce4), *C. brenneri* (caePb1), *C. remanei* (caeRem2), *C. briggsae*

Table 2
`NcDNAlign` and `TBA` applied to the same genomic sequences of gammaproteobacteria.
Consuming approximately the 20 to 30-fold amount of CPU-time **(a)**, `TBA` produces roughly 2.5 fold more alignments than `NcDNAlign` **(b)**. `RNAz` applied to these alignments yields half the number of hits in `NcDNAlign` as compared to `TBA` at comparable false discovery rates **(c)**. Despite the discrepancy in the number of alignments, the sensitivity of the `RNAz`-based detection of annotated ncRNAs is equal for both approaches **(d)**. Numbers in brackets at (c) and (d) denote the fraction of recovered over total elements. Numbers of overlapping alignments, resp. overlapping `RNAz` predictions, refer (from the left to the right) to pairwise comparisons of the screens: (1) versus (2), (3) versus (4), (3) versus (5). Both programs processed 588,542 bp of *E.coli*'s genome.

| | NcDNAlign | | NcDNAlign | | TBA |
|---|---|---|---|---|---|
| | default BLAST | | modified BLAST | | |
| Flanking regions | + | - | + | - | n.a. |
| | (1) | (2) | (3) | (4) | (5) |
| (a) | **CPU-time** | | | | |
| Total [min] | 15.68 | 14.35 | 29.27 | 27.52 | 548.36 |
| (b) | **Alignments** | | | | |
| Nr. alignments | 169 | 169 | 542 | 499 | 1347 |
| Nr. overlapping aln. | 155 | | 483 | | 479 |
| Total align. nucleotides | 35 153 | 29 596 | 125 907 | 113 829 | 235 986 |
| % of *E. coli* genome | 0.76 % | 0.64 % | 2.71 % | 2.45 % | 5.09 % |
| (c) | ***RNAz*** | | | | |
| Nr. hits | 126 | 122 | 339 | 300 | 658 |
| Overlap | 99 | | 280 | | 260 |
| Overall length of hits | 25 100 | 20 618 | 94 995 | 80 680 | 92 888 |
| Mean length of hits | 199 | 169 | 280 | 269 | 141 |
| FDR | 0.32 | 0.33 | 0.25 | 0.24 | 0.25 |
| Nr. annotatable hits | 102 (.80) | 98 (.80) | 212 (.62) | 189 (.63) | 469 (.71) |
| Nr. non-annot. hits | 24 (.19) | 24 (.19) | 127 (.37) | 111 (.37) | 189 (.29) |
| (d) | **Sensitivity** | | | | |
| rRNA, 22 annot. | 10 (.46) | 9 (.41) | 14 (.64) | 14 (.64) | 14 (.64) |
| tRNA, 86 annot. | 55 (.64) | 61 (.71) | 62 (.72) | 62 (.72) | 56 (.65) |
| Misc. RNA, 49 annot. | 5 (.10) | 6 (.12) | 18 (.37) | 16 (.33) | 23 (.47) |
| Overall, 157 annot. | 70 (.46) | 76 (.50) | 94 (.60) | 92 (.59) | 93 (.59) |

(cb3), and *P. pacificus* (priPac1). We applied `NcDNAlign` to set up MSAs of these nematodes using the corresponding repeat-masked genome versions currently available at `UCSC`. Thereby, we find ∼49 300 local alignments (>= 3-way) which cover ∼9 Mb of the *C. elegans* genome, whereas the given `TBA` counterpart comprises more than 49 Mb.

We analyze ultra-conserved sub-sequences of different lengths ($\geq$ 50, 100, 200 nt) that display 100 % sequence identity in at least three, four, or five species. Note that this definition differs from the one used by Bejerano *et al.* [18] where only 3-way alignments and UCRs of at least 200 nt have been required. Table 3 illustrates the number of observed UCRs for the `NcDNAlign` resp. the `TBA` algorithm at different length thresholds.

Although we have nominally used the same genomes and genome versions as used for the alignments at `UCSC` we find different repeat masked regions in our and the `TBA`-alignments. Hence, we have discarded repeat-annotated UCRs in both sets. We obtain 530 UCRs in the `NcDNAlign`-alignments, which sum up to ∼37 kb and vary from 50 to 232 nt in length (average UCR length: 66 nt). Contrary, we identify 333 UCRs in >=3-way `TBA`-alignments of which 321 loci (96 %) are also present in the `NcDNAlign`-alignments. Both sets have 267 (81 %) UCRs in common. In 60 cases the competing algorithms choose different subject paralogs to build the MSA, preventing the alignment to display 100 % sequence identity. Of these 60 cases 37 are due to inconsistent repeat annotation where 'N'-masked nucleotides appear in our genome version preventing `NcDNAlign` to produce an alignment, while the `TBA` blocks contain the unmasked stretch. Several UCRs are separated by only a few less conserved sites, indicating that these ultra-conserved sequences belong to larger elements. A comparison of predicted *C. elegans* ultra-conserved loci with the current `WormBase` annotation and some other publicly accessible databases is summarized in Tab. 4 and 5. It shows that nearly 25 % of the UCRs lack any annotation and are of unknown function. 119 UCRs are shared with the CNE set of Vavouri *et al.* and 89 are homologous to *Drosophila melanogaster* UCRs [22]. However, there is no sequence similarity between our nematode UCRs and the vertebrate set of Bejerano *et al.* (`blastn E-value` <= 1e-3). Computing CNEs as a less stringent set of conserved elements, featuring a 100 % conserved stretch of $\geq$ 30 nt in all possible 3- to 5-way alignments, yields 4 479 items. 1 170 of them have `blastn` hits (`E-value` <= 1e-3) with Vavouris' CNEs, 238 with Glazov's, and five show sequence similarity with Bejerano's UCRs. Vavouris' and our set of conserved elements are quite similar but obviously not identical because different genome assemblies have been applied, available annotation tracks have changed, and, contrary to us, Vavouri *et al.* removed a plenty of annotatable elements (e.g. protein-coding regions or RNA genes) from their released CNE sequences.

The example shows that `NcDNAlign` can be used to quickly (2-3 days on a

Table 3
The table illustrates the number (a) and respective lengths (b) of nematode UCRs identified in NcDNAlign- and TBA-alignments at three different length thresholds. We furthermore present the amount of UCRs common to both alignment methods.

| minimal UCR length [nt] | 50 | 100 | 200 |
|---|---|---|---|
| **(a) Overall number of UCRs** | | | |
| NcDNAlign | 530 | 4 | 2 |
| TBA | 337 | 2 | 1 |
| Nr. common UCRs | 267 | 2 | 1 |
| % common UCRs | 79 | 100 | 100 |
| **(b) Overall length of UCRs** | | | |
| NcDNAlign | 32 908 | 655 | 438 |
| TBA | 20 509 | 308 | 206 |
| Common nucleotides | 16 379 | 308 | 206 |
| % common nucleotides | 80 | 100 | 100 |

Table 4
Annotation of nematode UCRs.
The table shows the number of nematode UCRs that have blastn hits (E-value < 1e-5) with three major public ncRNA databases, respectively the number of UCRs that overlap WormBase annotation. The different screens (A), (B), and (C) are further described at Tab. 3. Details of the WormBase annotated UCRs are given at Tab. 5.

| *screen* | Rfam | Noncode | ncRNAdb | WormBase180 |
|---|---|---|---|---|
| (A) | 107 | 60 | 31 | 405 |
| (B) | 40 | 22 | 12 | 240 |
| (C) | 27 | 17 | 9 | 184 |

single machine, particularly faster if distributed) and reliably produce large-scale genomic alignments. However, novel evolutionary insights of nematode conservation patterns are obtained, since our approach identifies UCRs that are not present in already existing alignments. Thereby, one has to keep in mind that, contrary to [17], where a computer cluster of 1 024 nodes (866 MHz) had been applied for 481 days to generate pairwise alignments, our whole analysis can be done on one single workstation (Intel Xeon 2.80 GHz CPU, 1 GB RAM) and hence the term *quick* indeed is appropriately used in this context.

Table 5
Detailed overview of `WormBase180`-annotated nematode UCRs.
Generally, ncDNA evolves quickly by accumulating mutations. Nevertheless, we are able to identify numerous ultra-conserved non-coding regions in our alignments, of which a significant fraction covers already annotated ncRNAs. The table lists the number of UCRs that can be annotated using `WormBase` data. The screen identifiers (A), (B), and (C) are introduced at Tab. 3.

| | Wormbase 180 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| *screen* | CDS | intron | 5'UTR | 3'UTR | ncRNA | rRNA | tRNA | miRNA |
| (A) | 207 | 195 | 38 | 9 | 140 | 1 | 64 | 2 |
| (B) | 109 | 132 | 22 | 6 | 81 | 0 | 30 | 2 |
| (C) | 87 | 103 | 17 | 2 | 56 | 0 | 15 | 2 |

*Non-coding ultra-conserved regions in teleostean genomes*

The evolution of conserved non-coding elements in vertebrates has been discussed in detail in several studies, see e.g. [51–54]. A system to study the effect of a whole-genome duplication (WGD) on non-coding DNA are teleost fishes, which have undergone an additional genome duplication relative to the ancestral gnathostome at least 300 million years ago [55–58]. Generally, duplications are believed to have provided raw genetic material for selection to act upon. Compared to other vertebrates, such as mammals or zebrafish, the fugu and tetraodon genomes have a significantly reduced fraction of duplications due to transposon activity [59]. In addition to the large-scale duplication(s), a considerable fraction of multi-copied vertebrate CNEs owes its existence to the activity of transposable elements [60].

In contrast to prior studies of teleostean CNEs [61] which used fairly loose requirements for sequence conservation ($\geq$ 65 %), we focus here on ultra-conserved elements, which we define as having 100 % sequence identity among aligned sub-sequences with a length $\geq$ 50 nt. Using `NcDNAlign`, we aligned non-coding regions of the five teleosts *Fugu rubripes* (FUGU 4.0, 393 Mb), *Tetraodon nigroviridis* (TETRAODON 7, 342 Mb), *Gasterosteus aculeatus* (BROAD S1, 447 Mb), *Oryzias latipes* (HdrR, 700 Mb), and *Danio rerio* (Zv7, 1,527 Mb) (all genomes are available at the `Ensembl` database: `http://www.ensembl.org`) and parsed resulting alignments for the existence of UCRs. We found ∼66 400 alignments containing at least three euteleost species. These alignments comprise 10.6 Mb of fugu's genomic sequence having a mean pairwise identity of 78 %. Among them, we identified 2 377 UCRs covering 158 kb of the fugu genome with an average length $l$ of 66 nt ($l_{max}$=236 nt).

A `BLAST` search of the teleostean UCRs against several vertebrate and invertebrate genomes (see Figure 3) confirms prior findings that vertebrate CNEs are largely absent in invertebrates [53], and reveals 1 173 teleost-specific UCRs.

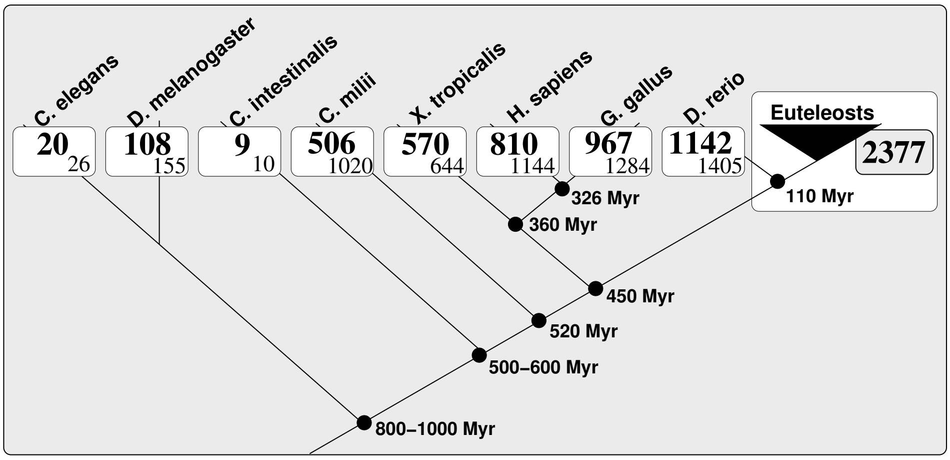


Fig. 3. Evolutionary distribution of teleostean UCRs.

We performed `BLAST` searches of our 2 377 teleostean UCRs against several other species. UCRs are well conserved throughout vertebrates but are largely absent at invertebrate lineages. Overall, 1 204 (51 %) of them are found in non-teleostean outgroups and thus 1 173 (49 %) UCRs seem to be teleostean-specific. We did not require 100 % sequence identity outside the euteleosts, just a significant `BLAST` HSP (`E-value` $< 1e-3$). Large-written numbers indicate the amount of conserved UCRs, small (sub-scripted) numbers denote the number of `BLAST` hits. Divergence times are taken from [62–64].

`BLAST` searches (`E-value` $< 1e-3$) against the sets of orthologous CNEs of the CONDOR database (`http://condor.fugu.biology.qmul.ac.uk/`) confirm our finding that a considerable fraction of UCRs within the data set is teleost-specific - see Table 3 of the supplement for details. 810 UCRs are conserved between the genomes of fugu and human. 91 of our UCRs correspond to the vertebrate UCR data set of Bejerano *et al.* (481 elements) and 26 match Drosophila UCRs that are associated with Homothorax mRNA splicing [22]. To resolve putative duplication events we searched for paralogous sequences of our UCRs in the fugu genome using `BLAST`. Thereby, all significant `BLAST` HSPs are taken into account (`E-value` $< 1e-6$). In principle it is allegeable to obtain up to eight copies ($2^3$) for an ancient sub-sequence since two rounds of whole-genome duplications are known at the root of the vertebrate lineage and a third within the teleosts [55,65–70]. 152 UCRs appear more than once in fugu, see Tab. 6. Many of them are still present in danio (101), and in the non-duplicated out-groups: shark (60), frog (73), chicken (80) and human (83).

## 3 Discussion

With `NcDNAlign` we present a pipeline for the computation of multiple sequence alignments of non-coding regions based on genomic input sequences.

Table 6
Evolutionary conservation of duplicated teleostean UCRs.
The table illustrates the distribution of teleostean UCRs according to their copy number in fugu and tetrapods (human, chicken, frog). For example, 2 225 UCRs reside as a single element in fugu, but only 377 of them are singly present at tetrapods. 1 218 teleost-specific UCRs do not have an obvious counterpart in tetrapods, but 1 066 still exist as single element in either human, chicken, or frog.

| | | UCR-copy number in fugu | | | | |
|---|---|---|---|---|---|---|
| | all | 1x | 2x | 3x | 4x | >4x |
| teleostean UCRs | 2 377 | 2 225 | 118 | 27 | 4 | 3 |
| absent in tetrapods | 1 218 | 1 159 | 46 | 12 | 0 | 1 |
| present in tetrapods | 1 159 | 1 066 | 72 | 15 | 4 | 2 |
| 1x | 386 | 377 | 8 | 1 | 0 | 0 |
| 2x | 208 | 196 | 11 | 1 | 0 | 0 |
| 3x | 340 | 311 | 21 | 6 | 0 | 2 |
| 4x | 116 | 108 | 8 | 0 | 0 | 0 |
| >4x | 109 | 74 | 24 | 7 | 4 | 0 |

Different applications in comparative genomics make highly different demands on MSAs. Using e.g. existing genome-wide alignments as input data for RNA gene finding typically requires extensive filtering retaining only a fraction of the original data. When no ready-made alignments are available and MSAs are computed *de-novo* for a specific application it is therefore desirable to compute only those alignments that can be directly used in the downstream analysis. This potentially reduces computational effort for MSA generation and obviously saves effort for intermediate processing of the alignments as well as storage space for data sets of which only fractions would eventually contribute to the analysis. `NcDNAlign` has therefore been built to be easily configurable. Beside command-line parameters, a configuration file in key–value format can be edited to tune the software e.g. to produce less stringent long alignments (RNA gene finding) or highly conserved short blocks (UCR analysis).

`NcDNAlign` is completely implemented in `Perl` and uses standard bioinformatic software – `BLAST`, `DIALIGN` and `ClustalW` – thereby making it applicable on many different systems, e.g. all Unix derivatives. Its modular implementation can easily be adapted to include future algorithms for MSA generation or other useful features, like threading to support parallel computation.

In its basic paradigm, `NcDNAlign` has parallels with `TBA`. Both use an initial `BLAST` step to generate candidate pairwise alignments which are subsequently combined into MSAs. `TBA`-generated alignments have proven to be highly valuable in various studies on non-coding sequences. Recently, Wang *et al.* reported that alignments of genomic regions coding for RNA genes generated by `TBA` are in some cases erroneous, but that overall `TBA` performs well [71]. `TBA` has, however, a major drawback when ready-made alignments are not available: as it is geared towards maximal coverage of the input genomes, its computational effort is enormous, demonstrated e.g. in [17]. This is limiting, not only when computational resources are limited, but also as the constant increase in newly sequenced genomes and improved genomic data encourages the frequent re-computation of comparative genomic analyse. Using large genomes, the most demanding step in `TBA` is the computation of pairwise alignments. `NcDNAlign` therefore uses the less sensitive but faster `blastn` instead of `blastz` and does reference-genome-versus-all instead of all-versus-all pairwise alignments. At the cost of reduced sensitivity and the need to define a reference organism, we thereby achieve a significant increase in efficiency. `NcDNAlign` in contrast to `TBA` does not require the provision of phylogenetic data, as this cannot always be obtained easily. Also, we do not use exonic anchors to guide the alignment process as `TBA` and other alignment programs, like `Pecan` do. While this may lead to a loss of very short conserved intronic regions, it enables the alignment of mobile non-coding elements that are intronic in only a part of the aligned genomes. `NcDNAlign` in contrast to other genomic alignment pipelines allows to easily select sub-regions of the genomes based on annotation for the alignment process and provides an alignment polishing procedure, that avoids subsequent filtering of the MSAs prior to feeding them to downstream applications.

Non-coding genes or elements in general constitute a highly diverse set of sequences. Therefore, `NcDNAlign` does not make any assumptions on the type of sequence and will process sequences whether they are of intergenic, intronic, or exonic origin. There are, however, significantly different sequence constraints between non-protein-coding and protein-coding sequences. Alignments of the latter benefit from models of coding sequence evolution and more sophisticated approaches exist for that purpose. As performance has been an important design goal for `NcDNAlign` and coding sequences are not in its major focus, we have not included any specialized treatment of these sequences. For similar reasons, we do not explicitly distinguish between orthologs and paralogs. For the type of applications `NcDNAlign` has been designed, like ncRNA gene search, motif discovery, and analysis of ultra-conserved regions, the inclusion of a paralog instead of the true ortholog does not make a significant difference in those cases where orthologs and paralogs cannot be readily distinguished based on sequence conservation. When creating the MSA from pairwise alignments we therefore include the best hit (according to the blast score) of the reference sequence in each other organism.

We exemplarily applied `NcDNAlign` to three different sets of genomic sequences (bacteria, nematodes, teleosts) and generated alignments as input data for two different aspects of comparative genomics: ncRNA gene finding and UCR analysis. Thereby, we showed that our program scales encouragingly from bacterial to vertebrate sized alignment problems.

As expected, TBA generates significantly more alignments containing significantly more nucleotides than NcDNAlign which is caused by the speed-versus-sensitivity trade-offs we have made compared to TBA and the alignment polishing procedure. The efficiency of the latter is also demonstrated by an overall reduced fraction of gaps in alignments of NcDNAlign and a better Weighted-Sum-of-Pairs score compared to TBA. Despite the discrepancy in the number of alignments, the sensitivity for identifying annotated known ncRNA genes using RNAz is comparable between both approaches, or even better using NcDNAlign-alignments when considering putative ncRNAs annotated by Saetrom and colleagues [43]. Also, the false discovery rates are comparable for both approaches. Consequently, NcDNAlign can competitively be applied to ncRNA gene finding problems, while allowing a significant speed-up of the computation compared to TBA.

Applying NcDNAlign to detect non-coding ultra-conserved regions in nematodes yields similarly encouraging results. Almost all UCRs identified in TBA/MultiZ-alignments provided by UCSC are contained in NcDNAlign-alignments. However, not all of them are strictly UCRs in NcDNAlign-alignments. In the majority of cases this is caused by diverging repeat-masked regions in nominally identical genome versions. Similarly, NcDNAlign leads to the identification of a number of UCRs that are not contained in the TBA UCR set. Our approach of finding ultra-conserved sub-sequences restrictively focuses on 100 % sequence identity within nematode respectively teleostean genomes for demonstration purposes only. Further and more sophisticated analyses of conserved regions may apply more precise attempts, e.g. considering initial seed sub-sequences of 100 % identity which might be enlarged by less conserved flanking regions, to ameliorate capturing connected evolutionary preserved regions. Our results on UCRs and CNEs in teleosts are in line with prior findings that vertebrate CNEs are largely absent in invertebrates and that a significant fraction of UCRs is teleost specific. We identify about 150 potentially duplicated UCRs, which are partly conserved in non-duplicated outgroups. Our study of teleostean duplicated UCRs deliberately misses the distinction of copied genes that originate from local or small respectively whole-genome duplication events in great detail since teleosts contain plenty of mobile and pseudo-genic elements and there are only a few and mostly unreliable measures, e.g. locality, to perform such distinctions.

In summary, NcDNAlign is a pragmatic and performant approach to the generation of MSAs of non-coding regions from genomic sequences. Although NcDNAlign is optimized for performance rather than sensitivity and aligns significantly fewer nucleotides than TBA it appears to be as sensitive and specific as TBA when applied to ncRNA gene finding and UCR analysis. For applications where alignments of distant homologs are beneficial, TBA is certainly the more viable option. Overall, TBA is the more general approach, however, with a dramatically higher computational effort and the necessity to post-process alignments to satisfy specific needs. NcDNAlign in contrast is particularly useful when only limited computational resources are available, for pilot studies or studies that are routinely repeated as soon as new genomic data becomes available or when alignments should be taylor-made to a specific downstream application.

## 4 Methods - Insights into the NcDNAlign work flow

NcDNAlign is a pipeline that consists of the following five Perl programs

(1) ncDNAlign.1.cutSequences.pl
(2) ncDNAlign.2.getGwAln.pl
(3) ncDNAlign.3.mergeGwAln.pl
(4) ncDNAlign.4.realign.pl
(5) ncDNAlign.5.trimAln.pl

which are described below. In addition it calls the external alignment programs BLAST, DIALIGN or (optionally) ClustalW. Future implementations substituting currently incorporated algorithms by the user's favorites are conceivable.

Regarding worst case scenarios the big-O asymptotic time complexity of these five scripts is given as a function of the sequence length $l$ and the number of sequences $n$. ncDNAlign.1.cutSequences.pl processes each genome exactly once and therefore runs in $O(n{\cdot}l)$. ncDNAlign.2.getGwAln.pl performs $n{-}1$ pairwise BLAST searches and therefore belongs to $O(n-1\cdot l^2)$. ncDNAlign.3.mergeGwAln.pl uses cliquer [72] to solve the NP-hard problem of finding maximal cliques with branch-and-bound strategies. Generally, maximal cliques can be found most efficiently using Union-Find-Algorithms which belong to $O(k + l\log l)$, where $k$ is the number of Union-Find operations on $l$ elements. ncDNAlign.4.realign.pl applies DIALIGN which originally required $O(n^3)$, but current implementations tend to $O(n^2)$, and some additional filtering steps of $O(n{\cdot}l)$. ncDNAlign.5.trimAln.pl requires $O(n{\cdot}l)$ for beautification filters and the optional ClustalW step needs $O(n^4 + l^2)$, where creating distance matrices needs $O(n^2{\cdot}l^2)$, neighbor-joining is $O(n^4)$, and progressive alignment takes $O(n^3 + n{\cdot}l^2)$. Due to the big-O notation the most expensive step of the NcDNAlign pipeline seems to be ClustalW, but benchmarking showed that BLAST is the most time-consuming step and might be used as an upper bound regarding time complexity, cp. Figure 2. Keep in mind that the input data set for ClustalW is only a minor part of the original genomic sequences, while BLAST, contrary, has to process the complete genomic sequences of all applied organisms.

*Extracting genomic sub-sequences* - `ncDNAlign.1.cutSequences.pl`

The pipeline simply starts with genomic sequence data of a group of related species either in GenBank or Fasta format. Regions that are not of interest for a particular application, e.g. CDS in the context of an ncRNA search, can be excised provided they are annotated in the input GenBank files. This results in a Fasta formatted file that either contains sub-sequences (GenBank) or the complete genomic sequence with adjusted header information. Removing uninteresting parts of the genomic DNA at this stage of course speeds up both the initial BLAST searches and all further alignment procedures due to shorter input sequences. It may also improve the final results as it reduces the number of spurious alignments.

*Genome-wide alignments* - `ncDNAlign.2.getGwAln.pl`

In the next step each sequence-block of the reference is pairwisely aligned against all other sequences of all other given species. Thereby, the processed sequences can be small chunks of DNA as a result of excising certain loci (see `ncDNAlign.1.cutSequences.pl`) or complete chromosomes, respectively genomes. Only those BLAST results are analysed where `E-value` and length of hit satisfy the conditions specified in the configuration file. Only the best hit in each species for each query is retained for further processing. For studies of non-coding DNA, we recommend to use a modified set of parameters for the BLAST search that is optimized for non-protein-coding sequences (`-r 5 -q -4 -G 10 -E 6`, `http://stevemount.outfoxing.com/Posting0004.html`). Effects of using these non-standard BLAST parameters are shown and discussed at Table 1 and 2.

*Combining adjacent neighbors* - `ncDNAlign.3.mergeGwAln.pl`

Structured RNA sequences are often less conserved in regions without base pair interactions. This may cause BLAST to identify two short hits rather than a single long one. Therefore, we try to combine these individual local alignments into a single larger one. Due to rearrangement, deletion and duplication events during evolution not all single local alignments lead to consistent global alignments, see [38] for a detailed description of the associated technical issues. We use here the algorithm described in [38] to combine adjacent hits with a

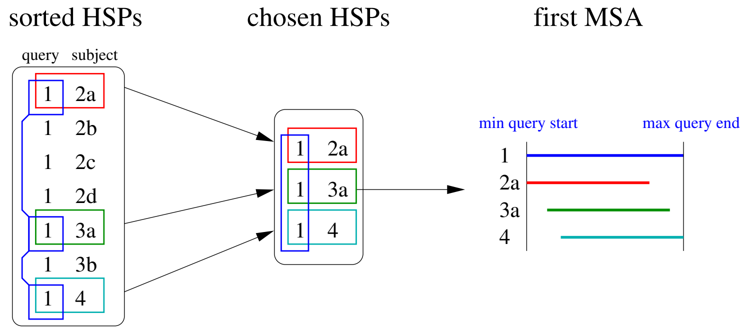


Fig. 4. MSAs are set up by grouping heuristic pairwise alignments.
Starting from tabular `BLAST` output, HSPs are sorted by query position and one "best" representative is chosen. All `BLAST` hits, resp. all subject sequences participating in an alignment, have to overlap in their query coordinates.

maximal distance of 30 nt in `BLAST` High Scoring Sequence Pairs (HSPs). In brief, the merging algorithm first computes a consistency graph whose nodes are the individual `BLAST`-alignments. The maximal cliques in this graph define sets of compatible pairwise alignments that can be combined. An illustratory example using artificial data is given in the electronic supplement.

*Initial multiple sequence alignments* - ***ncDNAlign.4.realign.pl***

Most algorithms for generating MSAs adhere to the progressive alignment paradigm, i.e. MSAs are built incrementally from pairwise alignments. We follow this strategy by grouping the corresponding `BLAST` HSPs. Thus, all HSPs are sorted by their loci in the reference genome. The "best" subject regarding the `E-value` is selected for each locus. Global alignment methods typically outperform local alignment approaches whenever the input sequences are related over their entire length [73]. Local methods, on the other hand, are superior in multiple domain cases where sequence identity is low and the sequences tend to share common motifs only [74]. Hence, we strongly recommend to use `DIALIGN` [75] for the realignment of HSPs to avoid a destruction of the alignment of pairwise grouped and mostly independent HSPs.

The `DIALIGN` algorithm can be applied to both, globally and locally related sequence sets, what indeed constitutes the missing link between locality of HSPs and an accurate alignment of globally similar sequences. In case sequences are only locally related, `DIALIGN` does not compute a global alignment and only aligns residues connected by selected diagonals [76]. Figure 4 illustrates the procedure of constructing the MSA out of the pairwise `BLAST` HSPs. In order to filter the results, the minimal number of sequences and minimal length of the alignment can be specified in the configuration file while the maximal

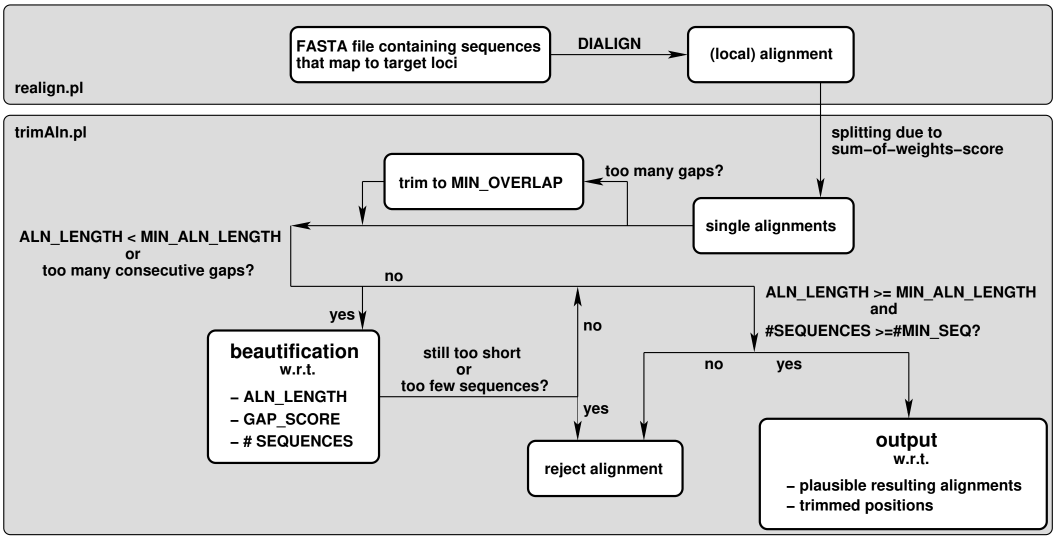


Fig. 5. Work flow of `ncDNAlign.5.trimAln.pl`.

number of sequences is, of course, the number of species in the screen. Optionally, `BLAST` results can be extended up- and downstream by a user-defined number of nucleotides to compensate for possible shortcomings in the original `BLAST` alignments.

*Beautifying multiple alignments - `ncDNAlign.5.trimAln.pl`*

Irrespective of the applied alignment algorithm, the initial MSAs are not of high quality with respect to large variations in sequence lengths due to the underlying pairwise alignments. Therefore, we developed a beautification procedure that trims the alignments to be in conformity with our definition of high-quality alignments. Figure 5 provides an overview of this work flow. `Dialign2-2` returns a Sum-of-Weight-score indicating the degree of local similarity among sequences for each alignment column. We use blocks with a minimal number of columns with score 0 to split raw `Dialign2-2` alignments into significantly aligned blocks. The minimum size $x$ of these separating insignificant blocks can be defined by the user. We then test the alignments ($i$) if their length exceeds the minimal length (the minimal length of the overlap is the same value as for retaining the local alignments in the configuration file) and ($ii$) if they contain $y$ consecutive gaps. Default values of $x = 20$ and $y = 120$ have been empirically determined as sensible thresholds for the trimming procedure. If ($i$) or ($ii$) is true, the "beautification algorithm" is applied to the alignment until the number of aligned sequences exceeds the minimal number of species in the screen or no further improvement is achieved.

The "beautification algorithm" addresses the following key issues:

(1) **The alignment may be elongated, however, at the cost of loosing individual short sub-sequences.**
Based on the alignment coordinates the specific sequence that can be dropped to enlarge the alignment needs to be selected. This is done by computing the two dropping candidates that define the borders of the central minimal overlap of all sequences (MIN_OVERLAP, *cp.* Figure 5 and Figure 2 in the supplement, can be interpreted as a seed alignment). The number of sequences having a valid base (not a gap) in the starting column of the left candidate sequence and in the ending column of the right candidate sequence are summed up. Depending on the higher number we drop the shortest sequence protruding at the left or right side of the minimal overlap. In case both sums are equal the sequence contributing less nucleotides to a putative longer alignment is chosen for dropping. If this is still not unique, the left one is chosen to enforce unambiguousness.

(2) **Considering the fraction of introduced gaps:**
Additionally to minimal length and minimal number of sequences we calculate a gap-score $g_s$ that is defined as:

$$g_s = \frac{\#gaps\ of\ trimmed\ sequence}{length\ of\ trimmed\ sequence} \tag{2}$$

A sequence is rejected if this value falls below a user specified threshold. The default value is set to 0.3, implying that at most 30% gap characters are allowed within a sequence.

(3) **Alignments that do not pass consistency filters are rejected:**
In all cases potential alignments do not pass above described filtering steps, the alignment is rejected (not returned) and beautification procedure finishes, to immediately save computation time.

An exemplary illustration of the alignment beautification procedure is given in Figure 2 in the supplement. In addition to the MSA, `NcDNAlign` also writes a file detailing the number of trimmed nucleotides for each sequence. Given as an optional command line argument, the entire MSA can optionally be realigned by applying `ClustalW` once more. The main reason for including this feature is that many analysis programs, including `RNAz`, are trained on alignments that were prepared with this program.

## 5 Supplement

Supplemental text, figures and tables are appended as separate pdf file. Further online supplemental material, including machine readable sequence and annotation files of UCRs/CNEs and `RNAz` predictions, is available at
`http://www.bioinf.uni-leipzig.de/Publications/SUPPLEMENTS/07-007/`

## 6 Availability

The `NcDNAlign` software is available under the GNU Public License from `http://www.bioinf.uni-leipzig.de/Software/NcDNAlign/`.

## 7 Acknowledgments

This work has been funded, in part, by the German DFG Bioinformatics Initiative BIZ-6/1-2 and by the 6th Framework Programme of the European Union (SYNLET).

## References

[1] G. M. Cooper, E. A. Stone, G. Asimenos, NISC Comparative Sequencing Program, E. D. G. Green, S. B. Batzoglou, A. Sidow, Distribution and intensity of constraint in mammalian genomic sequence., Genome Res 15 (2005) 901–913.

[2] D. Boffelli, J. McAuliffe, D. Ovcharenko, K. D. Lewis, I. Ovcharenko, L. Pachter, E. M. Rubin, Phylogenetic shadowing of primate sequences to find functional regions of the human genome., Science 299 (2003) 1391–1394.

[3] C. Dieterich, H. Wang, K. Rateitschak, H. Luz, M. Vingron, `CORG`: a database for COomparative Regulatory Genomics., Nucleic Acids Res. 31 (2003) 55–57.

[4] C. Dieterich, S. Grossmann, A. Tanzer, S. Ropcke, P. F. Arndt, P. F. Stadler, M. Vingron, Comparative promoter region analysis powered by `CORG`., BMC Genomics 6 (2005) 24 [10 pages].

[5] S. Prabhakar, F. Poulin, M. Shoukry, V. Afzal, E. M. Rubin, O. Couronne, L. A. Pennacchio, Close sequence comparisons are sufficient to identify human cis-regulatory elements., Genome Res 16 (2006) 855–863.

[6] S. S. Gross, M. R. Brent, Using multiple alignments to improve gene prediction., J Comput Biol 13 (2006) 379–393.

[7] S. Washietl, I. L. Hofacker, M. Lukasser, A. Hüttenhofer, P. F. Stadler, Mapping of conserved RNA secondary structures predicts thousands of functional noncoding RNAs in the human genome., Nat Biotechnol 23 (2005) 1383–1390.

[8] J. S. Pedersen, G. Bejerano, A. Siepel, K. Rosenbloom, K. Lindblad-Toh, E. S. Lander, J. Kent, W. Miller, D. Haussler, Identification and classification of conserved RNA secondary structures in the human genome., PLoS Comput Biol 2 (2006) e33.

[9] M. Brudno, M. Chapman, B. Göttgens, S. Batzoglou, B. Morgenstern, Fast and sensitive multiple alignment of large genomic sequences., BMC Bioinformatics 4 (2003) 66.

[10] N. Bray, L. Pachter, MAVID: constrained ancestral alignment of multiple sequences., Genome Res 14 (2004) 693–699.

[11] M. Brudno, C. B. Do, G. M. Cooper, M. F. Kim, E. Davydov, N. I. S. C. C. S. Program, E. D. Green, A. Sidow, S. Batzoglou, LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA., Genome Res 13 (2003) 721–731.

[12] I. Ovcharenko, G. G. Loots, B. M. Giardine, M. Hou, J. Ma, R. C. Hardison, L. Stubbs, W. Miller, Mulan: multiple-sequence local alignment and visualization for studying function and evolution., Genome Res 15 (2005) 184–194.

[13] M. Blanchette, W. J. Kent, C. Riemer, L. Elnitski, A. F. Smit, K. M. Roskin, R. Baertsch, K. Rosenbloom, H. Clawson, E. D. Green, D. Haussler, W. Miller, Aligning multiple genomic sequences with the threaded blockset aligner., Genome Res 14 (2004) 708–715.

[14] C. N. Dewey, L. Pachter, Evolution at the nucleotide level: the problem of multiple whole-genome alignment., Hum Mol Genet 15 Spec No 1 (2006) R51–R56.

[15] S. Kumar, A. Filipski, Multiple sequence alignment: in pursuit of homologous DNA positions., Genome Res 17 (2007) 127–135.

[16] ENCODE Project Consortium, The ENCODE (ENCyclopedia Of DNA Elements) Project., Science 306 (2004) 636–640.

[17] S. Schwartz, W. J. Kent, A. Smit, Z. Zhang, R. Baertsch, R. C. Hardison, D. Haussler, W. Miller, Human-mouse alignments with BLASTZ., Genome Res 13 (2003) 103–107.

[18] G. Bejerano, M. Pheasant, I. Makunin, S. Stephen, W. J. Kent, J. S. Mattick, D. Haussler, Ultraconserved elements in the human genome., Science 304 (2004) 1321–1325.

[19] A. Sandelin, P. Bailey, S. Bruce, P. G. Engström, J. M. Klos, W. W. Wasserman, J. Ericson, B. Lenhard, Arrays of ultraconserved non-coding regions span the loci of key developmental genes in vertebrate genomes., BMC Genomics 5 (2004) 99.

[20] E. J. Gardiner, L. Hirons, C. A. Hunter, P. Willett, Genomic data analysis using DNA structure: an analysis of conserved nongenic sequences and ultraconserved elements., J Chem Inf Model 46 (2006) 753–761.

[21] A. Siepel, G. Bejerano, J. S. Pedersen, A. S. Hinrichs, M. Hou, K. Rosenbloom, H. Clawson, J. Spieth, L. W. Hillier, S. Richards, G. M. Weinstock, R. K. Wilson, R. A. Gibbs, W. J. Kent, W. Miller, D. Haussler, Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes., Genome Res 15 (2005) 1034–1050.

[22] E. A. Glazov, M. Pheasant, E. A. McGraw, G. Bejerano, J. S. Mattick, Ultraconserved elements in insect genomes: a highly conserved intronic sequence implicated in the control of homothorax mRNA splicing., Genome Res 15 (2005) 800–808.

[23] T. Tran, P. Havlak, J. Miller, MicroRNA enrichment among short 'ultraconserved' sequences in insects., Nucleic Acids Res 34 (2006) e65.

[24] D. J. Gaffney, P. D. Keightley, Unexpected conserved non-coding DNA blocks in mammals., Trends Genet 20 (2004) 332–337.

[25] S. Katzman, A. D. Kern, G. Bejerano, G. Fewell, L. Fulton, R. K. Wilson, S. R. Salama, D. Haussler, Human genome ultraconserved elements are ultraselected., Science 317 (2007) 915.

[26] J. Feng, C. Bi, B. S. Clark, R. Mady, P. Shah, J. D. Kohtz, The Evf-2 noncoding RNA is transcribed from the Dlx-5/6 ultraconserved region and functions as a Dlx-2 transcriptional coactivator., Genes Dev 20 (2006) 1470–1484.

[27] A. Derti, F. P. Roth, G. M. Church, C. Wu, Mammalian ultraconserved elements are strongly depleted among segmental duplications and copy number variants., Nat Genet 38 (2006) 1216–1220.

[28] S. R. Eddy, Non-coding RNA genes and the modern RNA world., Nat Rev Genet 2 (2001) 919–929.

[29] G. Storz, An expanding universe of noncoding RNAs., Science 296 (2002) 1260–1263.

[30] J. S. Mattick, Challenging the dogma: the hidden layer of non-protein-coding RNAs in complex organisms., Bioessays 25 (2003) 930–939.

[31] A. Hüttenhofer, P. Schattner, N. Polacek, Non-coding RNAs: hope or hype?, Trends Genet 21 (2005) 289–297.

[32] F. F. Costa, Non-coding RNAs: new players in eukaryotic biology., Gene 357 (2005) 83–94.

[33] F. F. Costa, Non-coding RNAs: lost in translation?, Gene 386 (2007) 1–10.

[34] The Athanasius F Bompfünewerer Consortium: R. Backofen, S. H. Bernhart, C. Flamm, C. Fried, G. Fritzsch, J. Hackermüller, J. Hertel, I. L. Hofacker, K. Missal, A. Mosig, S. J. Prohaska, D. Rose, P. F. Stadler, A. Tanzer,

S. Washietl, S. Will, RNAs everywhere: genome-wide annotation of structured RNAs., J Exp Zoolog B Mol Dev Evol 308 (2007) 1–25.

[35] E. Rivas, S. R. Eddy, Noncoding RNA gene detection using comparative sequence analysis., BMC Bioinformatics 2 (2001) 8.

[36] S. Washietl, I. L. Hofacker, P. F. Stadler, Fast and reliable prediction of noncoding RNAs., Proc Natl Acad Sci USA 102 (2005) 2454–2459.

[37] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, D. J. Lipman, Basic local alignment search tool., J Mol Biol 215 (1990) 403–410.

[38] K. Missal, X. Zhu, D. Rose, W. Deng, G. Skogerbø, R. Chen, P. F. Stadler, Prediction of structured non-coding RNAs in the genomes of the nematodes Caenorhabditis elegans and Caenorhabditis briggsae., J Exp Zoolog B Mol Dev Evol.

[39] S. Prohaska, C. Fried, C. Flamm, G. Wagner, P. F. Stadler, Surveying phylogenetic footprints in large gene clusters: Applications to Hox cluster duplications., Mol Phyl Evol 31 (2004) 581–604.

[40] J. D. Thompson, F. Plewniak, O. Poch, BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs., Bioinformatics 15 (1999) 87–88.

[41] P. P. Gardner, A. Wilm, S. Washietl, A benchmark of multiple sequence alignment programs upon structural RNAs., Nucleic Acids Res 33 (2005) 2433–2439.

[42] E. Rivas, R. J. Klein, T. A. Jones, S. R. Eddy, Computational identification of noncoding RNAs in E. coli by comparative genomics., Curr Biol 11 (2001) 1369–1373.

[43] P. Saetrom, R. Sneve, K. I. Kristiansen, O. Snove, T. Grünfeld, T. Rognes, E. Seeberg, Predicting non-coding RNA genes in Escherichia coli with boosted genetic programming., Nucleic Acids Res 33 (2005) 3263–3270.

[44] I. M. Axmann, P. Kensche, J. Vogel, S. Kohl, H. Herzel, W. R. Hess, Identification of cyanobacterial non-coding RNAs by comparative genome analysis., Genome Biol 6 (2005) R73.

[45] C. Wang, C. Ding, R. F. Meraz, S. R. Holbrook, PSoL: a positive sample only learning algorithm for finding non-coding RNA genes., Bioinformatics 22 (2006) 2590–2596.

[46] N. Yachie, K. Numata, R. Saito, A. Kanai, M. Tomita, Prediction of non-coding and antisense RNA genes in Escherichia coli with Gapped Markov Model., Gene 372 (2006) 171–181.

[47] D. R. Rose, J. Hackermüller, S. Washietl, S. Findeiß, K. Reiche, J. Hertel, P. F. Stadler, S. J. Prohaska, Computational RNomics of drosophilids, BMC Genomics 8 (2007) 406, doi:10.1186/1471-2164-8-406.

[48] K. Missal, D. Rose, P. F. Stadler, Non-coding RNAs in Ciona intestinalis., Bioinformatics 21 Suppl 2 (2005) ii77–ii78.

[49] C. D. Val, E. Rivas, O. Torres-Quesada, N. Toro, J. I. Jimnez-Zurdo, Identification of differentially expressed small non-coding RNAs in the legume endosymbiont Sinorhizobium meliloti by comparative genomics., Mol Microbiol.

[50] T. Vavouri, K. Walter, W. Gilks, B. Lehner, G. Elgar, Parallel evolution of conserved non-coding elements that target a common set of developmental regulatory genes from worms to humans., Genome Biol 8 (2007) R15.

[51] E. T. Dermitzakis, A. Reymond, N. Scamuffa, C. Ucla, E. Kirkness, C. Rossier, S. E. Antonarakis, Evolutionary discrimination of mammalian conserved non-genic sequences (CNGs)., Science 302 (2003) 1033–1035.

[52] I. Ovcharenko, L. Stubbs, G. G. Loots, Interpreting mammalian evolution using Fugu genome comparisons., Genomics 84 (2004) 890–895.

[53] A. Woolfe, M. Goodson, D. K. Goode, P. Snell, G. K. McEwen, T. Vavouri, S. F. Smith, P. North, H. Callaway, K. Kelly, K. Walter, I. Abnizova, W. Gilks, Y. J. K. Edwards, J. E. Cooke, G. Elgar, Highly conserved non-coding sequences are associated with vertebrate development., PLoS Biol 3 (2005) e7.

[54] G. K. McEwen, A. Woolfe, D. Goode, T. Vavouri, H. Callaway, G. Elgar, Ancient duplicated conserved noncoding elements in vertebrates: a genomic and functional analysis., Genome Res 16 (2006) 451–465.

[55] A. Amores, A. Force, Y. L. Yan, L. Joly, C. Amemiya, A. Fritz, R. K. Ho, J. Langeland, V. Prince, Y. L. Wang, M. Westerfield, M. Ekker, J. H. Postlethwait, Zebrafish hox clusters and vertebrate genome evolution., Science 282 (1998) 1711–1714.

[56] J. Taylor, I. Braasch, T. Frickey, A. Meyer, Y. Van De Peer, Genome duplication, a trait shared by 22,000 species of ray-finned fish., Genome Res 13 (2003) 382–390.

[57] A. Meyer, Y. Van de Peer, From 2R to 3R: evidence for a fish-specific genome duplication (FSGD)., Bioessays 27 (2005) 937–945.

[58] K. D. Crow, P. F. Stadler, V. J. Lynch, C. T. Amemiya, G. P. Wagner, The fish specific Hox cluster duplication is coincident with the origin of teleosts., Mol Biol Evol 23 (2006) 121–136.

[59] X. Xie, M. Kamal, E. S. Lander, A family of conserved noncoding elements derived from an ancient transposable element., Proc Natl Acad Sci USA 103 (2006) 11659–11664.

[60] H. Nishihara, A. F. A. Smit, N. Okada, Functional noncoding sequences derived from SINEs in the mammalian genome., Genome Res 16 (2006) 864–874.

[61] A. Woolfe, D. K. Goode, J. Cooke, H. Callaway, S. Smith, P. Snell, G. K. McEwen, G. Elgar, CONDOR: a database resource of developmentally associated conserved non-coding elements., BMC Dev Biol 7 (2007) 100.

[62] S. Kumar, S. B. Hedges, A molecular timescale for vertebrate evolution., Nature 392 (1998) 917–920.

[63] V. E. Prince, F. B. Pickett, Splitting pairs: the diverging fates of duplicated genes., Nat Rev Genet 3 (2002) 827–837.

[64] J. E. Blair, S. B. Hedges, Molecular phylogeny and divergence times of deuterostome animals., Mol Biol Evol 22 (2005) 2275–2284.

[65] P. W. H. Holland, J. Garcia-Fernàndez, N. A. Williams, A. Sidow, Gene duplication and the origins of vertebrate development., Development (Suppl.) (1994) 125–133.

[66] J. Spring, Genome duplication strikes back., Nat Genet 31 (2002) 128–129.

[67] A. Christoffels, E. G. L. Koh, J. Chia, S. Brenner, S. Aparicio, B. Venkatesh, Fugu genome analysis provides evidence for a whole-genome duplication early during the evolution of ray-finned fishes., Mol Biol Evol 21 (2004) 1146–1151.

[68] S. Hoegg, H. Brinkmann, J. S. Taylor, A. Meyer, Phylogenetic timing of the fish-specific genome duplication correlates with the diversification of teleost fish., J Mol Evol 59 (2004) 190–203.

[69] P. Dehal, J. L. Boore, Two rounds of whole genome duplication in the ancestral vertebrate., PLoS Biol 3 (2005) e314.

[70] I. A. Hurley, R. L. Mueller, K. A. Dunn, E. J. Schmidt, M. Friedman, R. K. Ho, V. E. Prince, Z. Yang, M. G. Thomas, M. I. Coates, A new time-scale for ray-finned fish evolution., Proc Biol Sci 274 (2007) 489–498.

[71] A. X. Wang, W. L. Ruzzo, M. Tompa, How accurately is ncRNA aligned within whole-genome multiple alignments?, BMC Bioinformatics 8 (2007) 417.

[72] P. R. J. Östergård, A fast algorithm for the maximum clique problem, Discr. Appl. Math. 120 (2002) 195–205, software: `http://users.tkk.fi/~pat/cliquer.html`.

[73] J. D. Thompson, F. Plewniak, O. Poch, A comprehensive comparison of multiple sequence alignment programs., Nucleic Acids Res 27 (1999) 2682–2690.

[74] T. Lassmann, E. L. L. Sonnhammer, Quality assessment of multiple alignment programs., FEBS Lett 529 (2002) 126–130.

[75] B. Morgenstern, DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment., Bioinformatics 15 (1999) 211–218.

[76] B. Morgenstern, K. Frech, A. Dress, T. Werner, DIALIGN: finding local similarities by multiple sequence alignment., Bioinformatics 14 (1998) 290–294.