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

- Documentation -

Dominic Rose, Jana Hertel, Kristin Reiche, Peter F. Stadler, and Jörg Hackermüller

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

{dominic, jana, kristin, stadler, joerg}@bioinf.uni-leipzig.de

June 26, 2007

Contents

1 Overview **3**

2 Requirements **3**

3 Installing `NcDNAlign` **4**

4 The central `NcDNAlign` configuration file **4**

5 Setting up a new screen **5**

6 Aligning nitrobacterial genomes - an examplary screen **6**

7 Step 1 - `cutSequences.pl` **6**

8 Step 2 - `getGenomewideAln.pl` **7**

9 Step 3 - `mergeGenomewideAln.pl` **7**

10 Step 4 - `realign.pl` **7**

11 Step 5 - `trimAln.pl` **7**

12 Open problems and known bugs **8**

A Manual Pages **9**
A.1 `cutSequences.pl` 9
A.2 `getGenomewideAln.pl` 11
A.3 `mergeGenomewideAln.pl` 12
A.4 `realign.pl` 14
A.5 `trimAln.pl` 15

1 Overview

The NcDNAlign package is a consecution of the following Perl scripts:

- cutSequences.pl
  Extracts (sub)sequences out of data files (GBK, FASTA) and standardizes input files (e.g. FASTA headers).
- getGenomewideAln.pl
  Creates initial local alignments between non-coding regions of a target organism and its related organisms using the BLAST algorithm.
- mergeGenomewideAln.pl
  Filters and merges neighboured compatible BLAST hits.
- realign.pl
  Creates alignments from given BLAST hits using DIALIGN and optionally realigns them using ClustalW.
- trimAln.pl
  Beautifies the alignments by trimming flanking gaps, incorporates a length maximization algorithm and excludes derogating sequences.

They should be called in the above order as one script is based on its forerunner.

2 Requirements

The NcDNAlign package is implemented in Perl. It is tested with Perl version 5.8.8. under Fedora.

The following prerequisite Perl modules have to be installed correctly:

- strict
- Getopt::Long
- Pod::Usage
- Switch
- Math::BigFloat
- Math::Round
- File::Find
- Data::Dumper

Moreover, NcDNAlign uses some additional programs to fullfill its purposes. The following programs have to be installed in advance to run the package properly:

- BLAST, including formatdb and fastacmd.
- ClustalW.
- DIALIGN (we use dialign2-2).
- BioPerl[1].

NcDNAlign uses Sampo Niskanens' and Patric Östergårds' cliquer during the validation of BLAST hits[2]. The cliquer binary is created during the installation process of the NcDNAlign package.

[1] http://www.bioperl.org
[2] see http://users.tkk.fi/~pat/cliquer.html

3 Installing NcDNAlign

Just download the current version of NcDNAlign at http://www.bioinf.uni-leipzig.de/Software/NcDNAlign/. You only have to extract the tar.gz archiv and run the *Makefile.PL* script. The following example assumes the package is downloaded and saved at your home directory ∼/:

1. **Extract the package**
   `$ tar -zxvf NcDNAlign.tar.gz`
2. **Change into the NcDNAlign directory**
   `$ cd ∼/NcDNAlign`
3. **Create a makefile**
   `$ perl Makefile.PL [PREFIX=YourDir]`
   You can optionally specifiy a directory denoting where to install the package via the PREFIX argument. If you do not provide a target destination, NcDNAlign will be installed at /usr/local.
4. **Use the power of make to install the package**
   `$ make`
   `$ make install`

This procedure installs NcDNAlign by moving necessary files to the destination directory. If you explicitly wish to remove the whole package, please remove the NcDNAlign directory manually. Distributed files are listed in the .packlist file (e.g. /usr/local/lib/perl5/5.8.8/i686-linux/auto/NcDNAlign/.packlist).

4 The central NcDNAlign configuration file

Each screen uses a config file where global parameters and specific information of the analyzed organsims are declared. An exemplary configuration is given at utils/template.cfg. Necessary configuration parameters of an entire screen are common general values starting with 'C_' and organism specific information starting with 'ORG_'. Currently there are 12 options specifying common information and two parameters for each occuring organism. The number of organisms per screen is not restricted, NcDNAlign handles up to n organisms (resulting in n-way alignments). They just have to occur in the config file. Beside the template.cfg, you can use the utils/cfgCreator.pl to set up own config files.

The syntax follows a key–value format: KEY = "VALUE";

Options concerning common parameters:

- C_P_DIR
  The projects home directory (absolute path).
- C_SOURCE_DIR
  The directory containing the source (sequences and annotation) files (absolute path).
- C_ALN_DIR
  The directory contains computed alignments (absolute path).
- C_BLAST_CMD
  The BLAST executing command (or binary). Standard value: 'blastall -p blastn'.
- C_ALN_CMD
  The alignment executing command (or binary). Standard value: 'dialign2-2'.

- C_CUTOFF_EVALUE
  The threshold for keeping BLAST results. Standard value: '1e-10'.
- C_MIN_ALN_LENGTH
  The minimal alignment length, sequences greater or equal to this value are kept. Standard value: '40'.
- C_FLANKING
  Flanking regions of this size are added to the BLAST hits.
- C_GAP_SCORE
  The gap-score is a parameter ensuring the alignment quality. It is a cutoff value, only trimmed sequences with $countGaps/length < C\_GAP\_SCORE$ make it into an alignment. The smaller the value, the better the alignment. Standard value: '0.3'.
- C_GBK_KEYS
  All GenBank keys[3] specified here in a comma separated list are neglected in the genomic sequence and are not used for BLAST searches. Particularly, this is useful for ncRNA detection because putative false positives are excluded early. Furthermore, the dataset shrinks what generally is of utmost interest for genome-wide analyses to speed up overall runtime.
  Standard GenBank keys for rejection are:
  - CDS
  - repeat_region
  - repeat_unit
  - LTR
  - satellite
  - sig_peptide
  - transit_peptide
  - mat_peptide
- C_TARGET_NAME
  Each screen needs a target organism, its name is stated here. Be careful, it has to be the same string like one of the organisms names.

Options concerning organism specific parameters:

- ORG_NAME
  The name of the organism used as unique identifier throughout the analyses. Organisms must be named with a 3-letter code (e.g. Escherichia coli = Eco). There are several programs that do not support very long names, thus, the compromise of a general 3-letter-code seems to be appropriate.
- ORG_NCREGIONS_FASTA
  The name of the Fasta file containing non-coding regions used as input for BLAST searches.

5 Setting up a new screen

This section exemplaryly demonstrates how to set up a new screen.

1. **The screens' home directory**
   First of all, create a new directory, its the screens' home directory. New files produced by NcDNAlign scripts are stored in there.
   `$ mkdir MyScreenDir`

[3] see 'http://www.ebi.ac.uk/embl/Documentation/FT_definitions/feature_table.html#7.3'

2. **Adjust the config file**
   Every screen needs a central configurations file. You can either use the utils/template.cfg or create your own config using the utils/cfgCreator.pl. Adapt all pathes and parameters of the screens config file matching your environment like it is described at 4. Now copy the config file to your screen directory.
   `$ cp myScreen.cfg myScreenDir`

3. **Nothing else?**
   Now just cd to your screen directory and your are ready to run the `NcDNAlign` pipeline.
   `$ cd myScreenDir` Necessary commands are exemplaryly listed at utils/exampleCall.sh

6 Aligning nitrobacterial genomes - an examplary screen

We demonstrate how to use `NcDNAlign` in an exemplary screen of chosen bacterial genomic sequences. All necessary files are available at the example directory of the `NcDNAlign` package.

1. **Change to the example directory.**
   `$ cd NcDNAlign/example`

2. **Adopt the C_P_DIR, C_SOURCE_DIR, C_ALN_DIR options**
   You have to adopt the three options of the example.cfg fitting to your environment. Be sure to use absolute paths.

3. **Start the runExample.sh script**
   `$ sh runExample.sh`

The script will produce genome-wide alignments of 5 nitrobacterial genomes. Just have a look at the example/dialign2-2 directory, it contains resulting alignments.

7 Step 1 - `cutSequences.pl`

The script cuts out unwanted features of `GenBank` files and standardizes `Fasta` headers.

Input

`cutSequences.pl` expects `GenBank` formatted input data. Obviously, not every sequencing project provides its data in this format, thus, we supply a set of exemplary converters for fast and easy rewriting of individual user input data to `GenBank` format (among others we suggest `fasta2gbk.pl` and `embl2gbk.pl`). It is straight forward to apply own converters to handle other file formats. Additionally, it supports a set of `Fasta` files as input where consequently no features are cutted out, but at least corresponding `Fasta` headers are standardized. If it is requested to get alignments of all possible regions in the genomes using `GenBank` files, just set the C_GBK_KEYS option to an empty string.

Output

Automatically, annotation of both strands is considered and coordinates as well as strand information are retained in the header of the produced `Fasta` file:

```
>species:chr/contig:start:length:strand
sequence

species:    species, from which sequence is extracted
```

```
chr/contig: chromosome or contig of sequence
start:      start of sequence regarding chr/contig
length:     length of sequence
strand:     reading direction of sequence, {+,-}
sequence:   sequence itself
```

Important features of the GenBank files are:

DEFINITION defines the species' name and chromosome information

FEATURES defines the start of annotated subsequences

ORIGIN defines the start of the complete sequence

If there are no GenBank files but Fasta files available, it is necessary to leave the ORG_SOURCE_GBK tag in the central config file empty and directly write down the name of the Fasta file in the ORG_NCREGIONS_FASTA tag.

8 Step 2 - getGenomewideAln.pl

The script getGenomewideAln.pl first creates a subdirectory at the projects home directory to store the BLAST databases and the resulting hits. The BLAST databases are created with fastacmd (-p F -o). Afterwards a BLAST search of the reference organism against all other organisms is performed. Before the computation of local alignments the script writes upcoming system commands to STDOUT. This allows you to distribute the calculation manually and accelerate allover computation time (currently NcDNAlign does not support multiple threads).

9 Step 3 - mergeGenomewideAln.pl

The script mergeGenomewideAln.pl filters and combines neighboured BLAST hits beneath a threshold of 30 nt in order to ensure that a global alignment constitutes a complete ncRNA gene. Its input is tabular formatted BLAST output. The script calculates consistence graphes, their maximal cliques define valid sets of combinable alignments. For each input file *.merged files are produced listing the merged BLAST hits.

10 Step 4 - realign.pl

The script sets up a directory containing computed alignments at the project directory (both specified at the config file) and then uses DIALIGN and optionally ClustalW to set up alignments. It automatically uses input data from the 'blast' directory (*.merged). The resulting alignments are outputted in separate subdirectories containing up to 1000 files.

11 Step 5 - trimAln.pl

The script trimAln.pl removes flanking gaps of aligned sequences and, thereby, defines the minimal overlapping region. If the resulting alignment lengh is longer than a user defined threshold the alignemnt is accepted. If it is too short its length is optimized by rejecting sequences. The script produces beautified alignment files ending with *.aln.trim. Alignments that do not pass the filtering process are excluded from upcoming analyses by moving them to the folder 'bad'. Due to the fact that not every alignment format supports the annotation of coordinates, the file 'trimmedAln.bed' is created. It keeps coordinates of all aligned sequences.

12 Open problems and known bugs

- `BioPerl` **and** `cutSequences.pl`
  `BioPerl` seems to handle circular genomes incorrectly. Thus, `cutSequences.pl` produces incomplete output, if it is feed with genbank files containing coordinates of circular genomes, e.g. join(12345..123456, 1..100)
  You should split such entries into two separate ones and `BioPerl` problems are avoided.

A Manual Pages

A.1 `cutSequences.pl`

`cutSequences.pl` - part of the NcDNAlign alignment pipeline, step (1)
Extracts (sub)sequences out of sequence files and writes them out in NcDNAlign standardized FASTA format (supported input file formats: FASTA, GenBank).

SYNOPSIS

cutSequences.pl [options, mode 1] OR [options, mode 2]
There are two ways of running the program:
Mode 1: EITHER you specify the a configuration file and the program runs for a whole screen,
Mode 2: OR you specify a gbk file, the organisms name, the disturbing gbk feature keys and an outfile to cut one single gbk file.

OPTIONS

Mode 1:

-c, –conf FILE

Path to central NcDNAlign configuration file, inhibits -gbk, -name, -gbk_keys, -cut, -min_aln_len

Mode 2:

–gbk FILE

Path to a single gbk file.

-n, –name xyz

The 3-letter name of the organism that belongs to the gbk file

–gbk_keys "foo;bar"

Specify gbk feature keys that should be cutted out of the gbk file. List them comma separated! Reference of GenBank feature keys: `http://www.ncbi.nlm.nih.gov/collab/FT/#7.3`

–cut FILE

Path to a file taking the output in NcDNAlign customized fasta format. Cutted regions will be stored there (e.g. *.cut.fa).

–min_aln_len INT

Minimal required alignment length.

-v, –version

Prints version information and exits.

-h, –help

Prints a short help message and exits.

–man

Prints a detailed manual page and exits.

DESCRIPTION

cutSequences.pl optionally cuts out specific regions from genome files resulting in FASTA formatted files that either contain subsequences or the complete genomic sequence with appropriate headers.
If it is requested to get alignments of all possible regions of applied genomes and nothing should be cutted, just deliver an empty string for the option C_GBK_KEYS in the config file or on the -gbk_keys option at the command line. Getting rid of disturbing elements as earliest as possible directly speeds up subsequent BLAST searches and further alignment procedures due to shorter sequences.
However, one must be familiar with the GenBank feature keys documented at `http://www.ncbi.nlm.nih.gov/collab/FT/#7.3` to ensure that no desired loci are unintentionally cut out of the genomes (e.g. tRNAs can appear at exons, so dropping exons would cause to loose tRNAs).

EXAMPLES

```
Mode 1 (whole screen):
\$ cutSequences.pl -c config-file.cfg

Mode 2 (cutting single file):
\$ cutSequences.pl -gbk Eco.gbk -name Eco -gbk_keys "CDS;repeat_region" -cut Eco.cut.fa
```

AUTHORS

Dominic Rose (dominic@bioinf.uni-leipzig.de)
Jana Hertel (jana@bioinf.uni-leipzig.de)

AVAILABILITY

http://www.bioinf.uni-leipzig.de/Software/NcDNAlign/

A.2 `getGenomewideAln.pl`

`getGenomewideAln.pl` - part of the NcDNAlign alignment pipeline, step (2)
Wraps a pairwise BLAST search of the target genome against all other organisms listed in the configurations file.

SYNOPSIS

getGenomewideAln.pl [options]

OPTIONS

-c, –conf FILE

Path to central NcDNAlign configuration file

-s, –silent [0|1]

Silent mode, avoid printing to STDOUT 0=OFF (Default), 1=ON

-v, –version

Prints version information and exits.

-h, –help

Prints a short help message and exits.

–man

Prints a detailed manual page and exits.

DESCRIPTION

The script performs a pairwise nucleotide BLAST search using BLASTN to align the reference sequence heuristically against all other species. The best hit for each query subsequence is retained if the EVALUE of the BLAST search does not exceed the EVALUE-threshold and the query length is larger than a certain minimal length. Both thresholds are given in the config file. Results of the BLAST search are stored at the directory 'blast'.

EXAMPLES

$ getGenomewideAln.pl -c config-file.cfg

AUTHORS

Dominic Rose (dominic@bioinf.uni-leipzig.de)

AVAILABILITY

http://www.bioinf.uni-leipzig.de/Software/NcDNAlign/

A.3 `mergeGenomewideAln.pl`

`mergeGenomewideAln.pl` - part of the NcDNAlign alignment pipeline, step (3)
Filtering and merging of BLAST hits (required blast input format: tabular output by -m 8). Ensures that all compatible hits in same genomic region (distance <= –dist) are merged to span complete genes.

SYNOPSIS

mergeGenomewideAln.pl [options]

OPTIONS

-c, –conf FILE

Path to central NcDNAlign configuration file [REQUIRED]

-d, –dist INTEGER

Maximal nucleotide distance between BLAST HSPs to be considered as neighboured and, thus, potentially merged. Default: 30.

-o, –out [0|1]

Printing detailed results at STDOUT ON(1) or OFF(0). Default: 0.

-t, –test [0|1]

Should the results be verified? Enable consistency checks, e.g. ensure that each inputted HSP is also outputted (1=YES, 0=NO). Default: 0.

-s, –silent [0|1]

Silent mode, avoid printing to STDOUT 0=OFF (Default), 1=ON

-v, –version

Prints version information and exits.

-h, –help

Prints a short help message and exits.

–man

Prints a detailed manual page and exits.

DESCRIPTION

Prevalently, sequences are less conserved in regions without base pair interactions, which might prevent the aligning procedures in getGenomwideAln.pl from extending the sequence alignment into such regions. Due to rearrangement, deletion, and duplication events during evolution, not all single local alignments lead to consistent global alignments. Therefore, compatible adjacent BLAST HSPs with a maximal distance of 30 nt are combined to ensure that heuristic global alignments span complete genes. Only appropriate HSPs exceeding a certain EVALUE-cutoff are reported and make it into a consecutive merging process. This merging algorithm is about calculating consistence graphs. Their maximal cliques define valid sets of combinable alignments. Output is written to *.merged files. They contain the merged BLAST hits.

EXAMPLES

$ mergeGenomewideAln.pl -c config-file.cfg

AUTHORS

Dominic Rose (dominic@bioinf.uni-leipzig.de)
Kristin Reiche (kristin@bioinf.uni-leipzig.de)

AVAILABILITY

http://www.bioinf.uni-leipzig.de/Software/NcDNAlign/

A.4 realign.pl

`realign.pl` - part of the NcDNAlign alignment pipeline, step (4)
Realignment of prior BLAST hits.

SYNOPSIS

realign.pl [options]

OPTIONS

-c, –conf FILE

Path to central NcDNAlign configuration file [REQUIRED]

-o, –out [0|1]

Printing detailed results at STDOUT ON(1) or OFF(0). Default: 0.

-s, –silent [0|1]

Silent mode, avoid printing to STDOUT 0=OFF (Default), 1=ON

-v, –version

Prints version information and exits.

-h, –help

Prints a short help message and exits.

–man

Prints a detailed manual page and exits.

DESCRIPTION

Traditionally, MSAs are set up by pairwise alignments. We follow this strategy by grouping corresponding BLAST HSPs to one single MSA. Therefore, all HSPs are sorted according to the loci of the reference genome. Afterwards, one 'best' subject sequence is selected for each reference locus that overlaps the reference. It is ensured that every new subject sequence overlaps with all previous ones of the hitherto MSA. FThe minimal length for acceptance of an alignment is defineable. We recommend a value of 40 nt for ncRNA prediction and gene-finding on a genome-wide scale. Optionally, alignable subsequences can be enlarged by a certain user-defined flanking region. This compensates putative BLAST alignment errors, whenever initial seed alignments are not sufficiently extended to HSPs. Finally, MSAs of all accepted regions are computed using DIALIGN. Furthermore, the script creates a BED file 'gwAln.bed' keeping the coordinates of the alignment input respectively initial BLAST hits.
Be careful: A temporary file 't' must be createable in the screens' directory!

EXAMPLES

$ realign.pl -c config-file.cfg

AUTHORS

Dominic Rose (dominic@bioinf.uni-leipzig.de)

AVAILABILITY

http://www.bioinf.uni-leipzig.de/Software/NcDNAlign/

A.5 `trimAln.pl`

`trimAln.pl` - part of the NcDNAlign alignment pipeline, step (4)
Remove flanking gaps (trim leading and succeeding gaps and N's) of all *.aln files in the alignment directory. Beautify the alignments by length maximization.

SYNOPSIS

realign.pl [options]

OPTIONS

-c, –conf FILE

Path to central NcDNAlign configuration file [REQUIRED]

-o, –out [0|1]

Printing detailed results at STDOUT ON(1) or OFF(0). Default: 0.

-g, –maxGaps INT

The minimal number of connected gaps enforcing application of the beautification algorithm. The maximal allowed number of connected gaps in a sequence is maxGaps-1. Default is 120, an appropriate window size for third party programs (e.g. RNAz).

-z, –maxZeros INT

The minimal number of connected zeros enforcing a DIALIGN alignment to be splitted. The maximal allowed number of connected zeros in the DIALIGN alignment score-string (sum-of-weights indicates the degree of local similarity among sequences) is maxZeros-1. Default is 20, an appropriate value to split one alignment into two.

-d, –dropRef [0|1]

Is it allowed to drop the reference sequence out of an alignment for beautification purposes? 0=NO (default), 1=YES

-s, –silent [0|1]

Silent mode, avoid printing to STDOUT 0=OFF (Default), 1=ON

–stat [0|1]

Calculate statistics OFF(0, default) or ON(1). Statistics are printed to STDOUT and into the 'aln.statistics' file. Not fully implemented yet.

-r, –realign [0|1]

After all, should a realignment step using CLUSTALW be performed? 1=YES (default), 0=NO

-f, –format [CLUSTAL|FASTA|MAF]

Output alignment format. Default: CLUSTAL

-v, –version

Prints version information and exits.

-h, –help

Prints a short help message and exits.

–man

Prints a detailed manual page and exits.

DESCRIPTION

Regardless of the applied alignment algorithm, MSAs consisting of pairwise alignments may need beautification, conveniently, because of too large variations in sequence lengths. DIALIGN2-2 prints out a sum-of-weight-score indicating the degree of local similarity among sequences for each alignment column. However, we use this score to split one alignment into two if at least –maxZeros consecutive zeros are read in to separate valid local alignments from non-alignable regions ($x=20$ could be an appropriate default value). Moreover, we test the alignments (1) if their length exceeds the minimal length (Minimal length of the overlap is the same value as for retaining the local alignments in the configuration file) and (2) if they contain –maxGaps consecutive gaps (120 could be a valuable threshold for many alignment processing tools applying sliding windows). Obviously, it is useless to scan sequences that contain more consecutive gaps than the length of the sliding window. If (1) or (2) 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 improvement is achieved. Trimmed/Improved alignment files are outputted and a BED file 'trimmedAln.bed' for the coordinates of the trimmed alignments (not every alignment format handles coordinates) is created.

EXAMPLES

$ trimAln.pl -c config-file.cfg

AUTHORS

Dominic Rose (dominic@bioinf.uni-leipzig.de)

AVAILABILITY

http://www.bioinf.uni-leipzig.de/Software/NcDNAlign/