Cgaln: **Fast and memory-efficient program for pairwise alignment of whole genomic sequences**

Ryuichiro Nakato[1] and Osamu Gotoh[2,3]

1 Institute of Molecular and Cellular Biosciences, The University of Tokyo
2 Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University
3 National Institute of Advanced Industrial Science and Technology, CBRC

1. Overview

`Cgaln` (Coarse grained alignment) is a program designed to align a pair of whole genomic sequences of not only bacteria but also entire chromosomes of vertebrates on a nominal desktop computer. `Cgaln` performs an alignment job in two steps, at the block level and then at the nucleotide level. The former "coarse-grained" alignment can explore genomic rearrangements and reduce the regions to be analyzed in the next step. The latter is devoted to detailed alignment within the limited regions found in the first stage. The output of `Cgaln` is “glocal” in the sense that rearrangements are taken into consideration while each alignable region is extended as long as possible. Thus, Cgaln is not only fast and memory-efficient, but also can filter noisy outputs without missing the most important homologous segment pairs.

2. Install

`Cgaln` and its associated program `maketable` are written in ANSI-C, and can be executed on a Linux OS. To build `Cgaln` from the source, first download the archive file from
http://www.genome.ist.i.kyoto-u.a.c.jp/~aln_user/cgaln/
and extract it with:

```
% tar -xvzf Cgaln_1_*_*.tar.gz
```

Then, go to the new directory and type:

```
% make
```

`Cgaln` requires `gcc` for compilation. (http://gcc.gnu.org/)

3. Usage

Before running `Cgaln`, each genomic sequence to be aligned should be preprocessed by `maketable`. `Cgaln` and `maketable` accept single- or multi-fasta files. If the input file is in multi-fasta format, `Cgaln` treats all the entry sequences individually, and outputs the unified results.

3.1 Repeat mask

The repetitive regions in input sequences should be soft masked by other repeat-masking tools such as `RepeatMasker`.

3.2 make tables

`Maketable` is used to convert each fasta sequence into binary files that are read by `Cgaln`.

```
% ./maketable [options] <sequence>
```

Four kinds of file, *.seedtable, *.blktable, *.poistable and *.txt, are made in the directory ./<output directory>/ (default: CgalnTable). The seedtable and blktable for reverse complement are also constructed at the same time. If the input file is in multi-fasta format, corresponding table of all fastas is merged into one table. It should be noted that different tables are required when the user wants to use Cgaln with different block size or k-mer sizes. The options for setting specific k-mer or block size are as follows.

Options:
-o specify output directory (default: CgalnTable)
-K# specifies k-mer size (11 as default)
-BS# specifies block size (10000 as default)

For example, the command

```
% ./maketable -K12 -BS20000 seq1.fasta
```

makes tables for seq1.fasta with 12-mer spaced seeds and block size of 20000.

Important note: The prefix of all tables generated is the input file name. If the user makes the tables of a file which has same filename of others (e.x. ./human/chrX.fa and ./mouse/chrX.fa), the tables are overwritten. Please use different filenames for the files to compare.

3.3 Cgaln

After making tables, you can invoke Cgaln as follows:

```
% ./Cgaln [options] sequence1 sequence2 -o <outputfile>
```

The results are output to <outputfile> with a format compatible with gnuplot, while other formats may be chosen with -optype option.

options: (default in parentheses)
-t: table directory (default: CgalnTable)
-r: both strands (forward strand only)
-b: output block level alignments (output nucleotide level alignments)
-nc: no chaining (with chaining)
-ia: do iterative alignment (off)
-pr: print outlines of regions for iterative alignment (off)
-fc: remove inconsistent colonies at the block level (off)
-cons: remove inconsistent HSPs at the HSP-chaining (off)
-noext: (when -ia is on) skip gapped extension of HSPs (don't skip)
-sr: examine short reverse complement (off)
-k#: seed weight/width; -k1: 11/18; -k2: 12/19; -k3: 13/20 (11/18)
-BS#: block size (10000)
-X#: X drop-off value at block-level (4000)
-R#: ratio between X drop-off score and gap penalty at block-level (1500)
-nl: don't insert delimiters between outputs for multifasta file (on)
-otype: output format; 0: gnuplot; 1: ensfile; 2: fasta (gnuplot format)

Important note: When "-otype2 (fasta format)" is selected, the size of output file may be quite large (about several times larger than input files). This option is not appropriate for whole-genome comparisons of large genomes such as mammals.

Examples:

- For standard use, we recommend the options

```
% ./Cgaln -r -ia seq1.fasta seq2.fasta -o result.dot
```

  which outputs the alignment result between seq1.fasta and seq2.fasta to `result.dot` in a format compatible with `gnuplot,` considering both strands and applying iterative step.

- If the input sequences are large (e.g. over 100 Mb), a larger *k*-mer is appropriate. For example,

```
% ./Cgaln -r -ia -k2 seq1.fasta seq2.fasta -o result.dot
```

- If you simply wants to take a global view of homology between two genomes, e.g. to infer evolutionary events of genome rearrangement, it is enough to output the block-level results obtained in the first stage, which requires very short computational time.

```
% ./Cgaln -r -b seq1.fasta seq2.fasta -o result.dot
```

- If the result of block level alignment looks too noisy, the value for `X` might be increased. For example,

```
% ./Cgaln -r -ia -X6000 seq1.fasta seq2.fasta -o result.dot
```

  Conversely, if the result of block level alignment looks too sparse, the value for `X` should be decreased.

- Repetitious outputs at the block-level caused by unmasked repetitive regions can be filtered out by `-fc` option. `Cgaln` with `-fc` option enforces the resultant alignments to be consistent with one another (projections of dot plots to neither axis overlap each other) at the block level.

- *Other examples:*

```
% ./Cgaln -r -nc seq1.fasta seq2.fasta -o result.dot
```

  The above command outputs the alignment results considering reverse strand as well and does not chain HSPs.

```
% ./Cgaln -R1000 seq1.fasta seq2.fasta -o result.dot
```

  The value for `R` indicates the gap penalty at the block-level alignment. The smaller the score is, the larger the gap penalty is.

```
% ./Cgaln -r -ia -noext -sr seq1.fasta seq2.fasta -o result.dot
```

  With `-sr` option, `Cgaln` tries to find short inversions at the NA level. In this mode, `Cgaln` does not perform gapped HSP extension, and examines both the forward and reverse strands within inter-HSP regions.

4. References

Nakato, R., and Gotoh, O. (2008). A novel method for reducing computational complexity of whole genome sequence alignment. In Proceedings of the 6th Asia-Pacific Bioinformatics Conference

(APBC2008), pages 101-110.

Nakato, R., and Gotoh, O. (2010). Cgaln: fast and space-efficient whole-genome alignment, BMC Bioinformatics, 11:224, 2010

Ryuichiro Nakato <rnakato@iam.u-tokyo.ac.jp>
Osamu Gotoh <o.gotoh@i.kyoto-u.ac.jp>
Last modified on 2012-01-06