OVERLAP REGRESSION ANALYSIS MODULE

Ian Dew

1. Overview

The assembler overlap regression analysis module (AS_ORA) measures the effectiveness of the overlap detector module (AS_OVL) as applied to data generated by the sequence simulator and fragment generator (AS_SIM). Sequence data from other sources may also be used provided the fragment messages (FRG or SFG) contain properly formatted "truth" information in their source (src) fields. This field lists an ordered pair of start and stop indices of the fragment in the source sequence and possibly repeat annotation information. (Please refer to the *Overlap Detector Module: Specification and Preliminary Design* document and manpage-like documents on celsim and frag.)

The AS_ORA module performs a limited but detailed analysis of AS_OVL inputs and outputs. Since the fragment-generating program (frag) can introduce errors into the fragments, algorithms approaching the complexity of those in AS_OVL will be needed for a complete analysis. The program provides the following outputs:

- the number of:
  1. 'true' overlaps (as determined via fragment indices)
  2. overlaps 'found' by the overlap detector
  3. true overlaps that were also found
  4. false positives (found overlaps that are not true) with histogram by size, and
  5. false negatives (true overlaps that were not found) with histogram by size
- statistics on differences in the true versus found A_Hang and B_Hang fields with a histogram of the number of false negative overlaps of different sizes
- repeat-induced false positives using annotation data in the src field (this is crude – based on presence of repeat in fragment but not on interval within fragment)
- sequencing coverage gaps in number and bases
- likely chunk gaps (regions with read overlap sizes less than the overlapper can detect) in number and bases
- an option for generating an overlap file containing just true overlaps for testing the chunk graph software
- histograms output in format readable by celagram
- ~~The third version (in progress) will provide an option for generating an overlap file containing just 'true' overlaps for testing the chunk graph software. Additional versions will inspect the offset and delta fields and the fragment sequence strings.~~

2. Memory Usage

AS_ORA reads in all fragments and computes true overlaps. Each fragment takes up ~~53~~ 57 bytes. Each true or found overlap takes up 60~~48~~ bytes. Each annotation takes up 45 bytes.

3. Interface

AS_ORA is a command-line, stand-alone application and will use some AS_PER and AS_MSG components for reading the binary fragment store and binary or ASCII proto-IO overlap files. The program detects whether the input overlap file is binary or ASCII, reads it correctly, and creates an output overlap file in the same mode. The program is named overlap_regressor and its command-line interface is shown below.

```
overlap_regressor -s fragstore-name
                  -l min-valid-overlap-size
```

```
                    [-i input-overlap-filename |
                     -o output-overlap-filename]
```

The table below explains the command-line parameters. Either or both an input and output overlap filename must be specified.

Parameter	Description
-s fragstore-name	Name of binary fragment store directory
-l min-valid-overlap-size	Minimum overlap size to detect or compare (usually 40)
-i input-overlap-filename	File containing found overlaps to analyze
-o output-overlap-filename	Filename to create and populate with true overlaps

-Output will be text, dumped to `stdout`, which may be viewed directly or imported to a data display program. An example of the statistical output is shown below.

The output overlap file will list fragments in sorted order by start and stop position in the original sequence. Overlaps will be listed in arbitrary order.

```
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Comparison of true overlaps computed from fragment indices
 with found overlaps determined by the overlap detector.

-- Data sources
  Fragment Store: a004/a004
  Overlap file:   a004/a004.ovl
    Number of fragments:      70000
    Minimum overlap size of interest:   40
    Minimum abs( A-hang + B-hang ) difference for overlap match:   10

-- Sampling
   Number of sequencing gaps:            0
   Bases of sequencing gaps:             0
   Number of non-overlapable gaps:       0
   Bases of non-overlapable gaps:        0

-- Simple counts
   Found overlaps (based on overlapper output):      789746
   True overlaps (based on fragment indices):        637872
                                      ----------------------
                                     Difference:     151874

-- Composition of found overlaps
  True overlaps that were found at least once:       615477
  True overlaps that were found more than once:         572
  Near misses:                                            0
  False positives:                                   173697
                                                 -----------
                                                 Total:     789746

-- True overlaps that were found
  Statistical comparison of A-hang and B-hang values
    between matching true & found overlaps
             Mean          Max          Stdev
  A-hang:  0.0000e+00    400    0.0000e+00
  B-hang:  0.0000e+00    459    0.0000e+00

-- False positives (limited by annotations in src field)
```

```
  Possibly repeat-induced:            0
  Not repeat-induced             173697

-- Repeats
     ID       Fragments containing
     B                     129
     C                      38
```

(if repeat-induced overlaps were detected, the break out of these would be listed here)

```
-- False negatives
  False negatives:                                  22395
  Critical false negatives:                          1234
  Number of separate, non-contained overlap graphs:     1

-- Histogram of overlap sizes:
(False negatives are in celagram file a004.fn.cgm)
(Critical false negatives are in celagram file a004.cfn.cgm)
(False positives are in celagram file a004.fp.cgm)
                              Critical
  Overlap        False         False         False
   Sizes       Negatives     Negatives     Positives
     40            519           327           461
     41            325           288           400
     42            291           255           436
      Overlap size     number missed
            40               519
            41               325
            42               291
```

…(the histogram listing continues out to the longest missed overlap)…

4. Design

See ..\..\src\AS_ORA\AS_ORA_main.c

1. Count the number of found overlaps to allocate an initial heap of overlaps
2. Read the fragments into an array
3. Sort it by minimum start/stop index
4. Loop over the sorted fragments and determine & populate true overlaps. Associate each overlap with one of the fragments
5. If an input overlap file is specified, ~~R~~read the found overlap messages, one at a time. Search for each one among the true overlaps & do different things depending on whether it: A) isn't a true overlap, B) matches a true overlap and 1) has been encountered before among found overlaps, or 2) hasn't been encountered before among found overlaps. Loop over the true and found overlaps & accumulate statistics & print them
6. If an output overlap file is specified, create it based on true overlaps

5. Limitations

`AS_ORA` is designed to run in batch mode on entire fragment and overlap store files. It will need modifications to function in an incremental mode. The first version will use only fragment start and stop indices into the source sequence as "truth" information. Future versions will incorporate the offset and `delta` fields and the fragment sequence strings into the analysis.

6. Status

~~Version 2 is completed.~~ Completed. More work should be done on determining whether overlaps are repeat induced based on interval of repeat in fragments relative to the overlapping interval rather than on mere presence of repeat in fragments.

7. Component Architecture and Unit Dependencies

AS_ORA depends on the AS_PER and AS_MSG modules for reading binary fragment store and reading and writing binary and ASCII overlap files.

AUTHORS

Ian Dew:

Created December 10, '98

Revisions:

Ian Dew: January 29, '99	Version 3+ updates
Ian Dew: July 15, '99	Celagram-related text & minor reformatting