NCBI Datamining

By Carsten Johansen, s092981

Preface

For this project, I have used Oracle Virtual Machine with a Ubuntu OS and Microsoft Windows 7 with a 64-bit Strawberry Perl installation, Sublime Text 2 and gedit as text editors and Cytoscape version 2.8.3 32-bit for rendering the network. The project has been written in LaTeX.

Preproces

Homo sapiens tax-id 9606

I found the Homo sapiens (human) tax-id, by going to uniprot.org and searched "human" under taxonomy. It showed 6,356 result, the first being "Homo sapiens (human)". There I could find a description on among others lineage, etc. I also found the taxon identifier to be 9606.

Preparing the files

In order to have the main program run more efficient, I downloaded the files gene2pubmed (235,906 kb) and gene_info (1,256,506 kb) and the README file. Since the 'gene_info' file contained a lot of information, not needed for this project, I wrote a little script to extract all the Homo sapiens related entries.

```
#!/usr/bin/perl -w
use strict;

open (IN, '<', 'gene_info') or die "Couldn't open file $!";
open (OUT, '>', 'gene_info2') or die "$!";
while (defined (my $line = <IN>)) {
    print OUT $line if $line =~ m/^9606\s+\d+/g;
}
#Everytime a line matches, it will be printed to the file 'gene2'
close IN;
close OUT;
```

The output file 'gene_info2' is 9,817 kb and contains only the human entries and took only 55.4 seconds to run. With a little modification, I could use the same script for the 'gene2pubmed', for the same purpose. It took only 45.8 seconds to run and reduced the filesize to 16,835 kb. This is a clear gain in my perspective. The script was based on the information about the two files in the README file. Since the only file really necessary for this project is the 'gene2pubmed'. The format is a tab delimited file, with first position being the Tax ID, second the GeneID and the third and last is the

PubMed ID. Since we are handling only the human entries in the file, the first position is irrelevant.

```
#!/usr/bin/perl -w
use strict;

my @array = '';
open (IN, '<', 'gene2pubmed') or die "Couldn't open file $!";
while (defined (my $line = <IN>)) {
    my @tmp = split ("\t", $line);
    my $first = shift (@tmp);
    if ($first =~ m/^9606$/) {
        my $list = "$tmp[0]\t$tmp[1]";
        push (@array, $list) if $first =~ m/^9606$/;
    }
}
close IN;

open (OUT, '>', 'gene2pubmedHumanPro2') or die "$!";
print OUT @array;
close OUT;
```

This script takes the 'gene2pubmed' file, extracts all the human entries and discards the first column, so only the GeneID and the PubMed ID is left. I choose to structure the script, as such it would collect it all in array, before writing it to an file, instead of writing it directly into the file which takes time for an Hard disk drive.

Program and output

Output file format

The output file, will be in the format of

```
Gene1   Weight  Gene2
1       23      3
2       12      4
3       4       4
```

This is just a mock up, not actual data. It is a tab delimited file, with the weight of the network between the two genes.

Calculation of weight

The weight of the edge, can be calculated from a few variables. One of them is the overall number of nodes (unique genes) of which there are 31411 human in this project, with 854735 articles mentioning. The second is the number of co-mentioning articles (sum of edges). From these two variables, the weight of the edge, can be defined as the sum of edges per overall nodes, Weight = SumEdge/Nodes.

The Program

Unfortunately I never got the program to work, but my my plan was to make a hash of arrays, where the hash-key was the GeneID and the value a array-reference to the array of PubMed articels. I got the HoA "creator/packer" to partly work. It is displayed below.

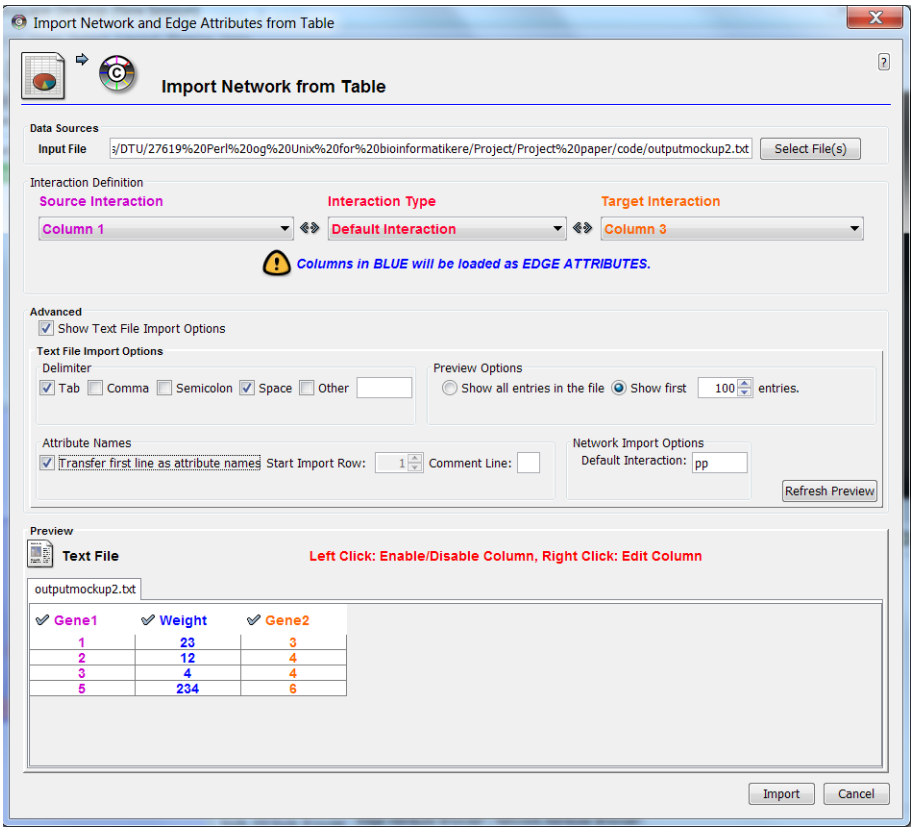
```
#!/usr/bin/perl -w
use strict;

my %HOA;
my $gene2 = 1;
my $array = [];
open (IN, '<', 'gene2pubmedHumanPro') or die "Couldn't open file $!";
while (defined (my $line = <IN>)) {
    chomp $line;
    my $flag = 0;
    my @tmp = split ("\t", $line);
    my $gene1 = $tmp[0];
    if ($gene1 eq $gene2) {
        push (@$array, $tmp[1]);
    }
    else {
        $array = [];
    }
    $HOA{$gene1} = $array; #FEJL FEJL FEJL first index is not there
    $gene2 = $tmp[0];
}
close IN;
```

Runtime for program

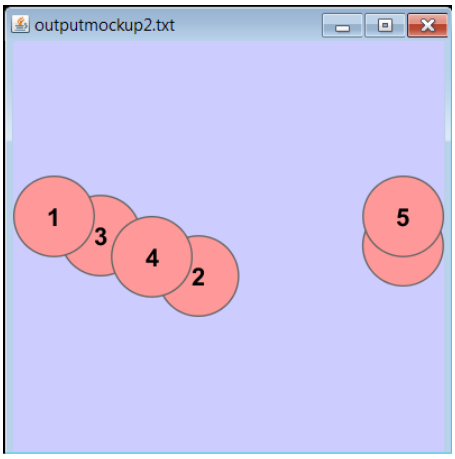
As the program didn't work, I can't make a runtime analysis, but I in the section about preparing the files, I have made some comments on the runtime and program structure.

Import into Cytoscape

In order to import it into Cytoscape, I opened the file via File/Import/Network from table (Text/MS Excel). Then I choose the file in the option screen and set the options (see picture below)



Then I clicked 'Import' and rendered the network. In order to see the weighted network I choose Layout/Cytoscape Layout/Edge-Weighted Force-Directed (Biolayout)/weighted. Below is a picture of the mock-up network



The larger the weight is, the more the edges of the notes overlap.