Description of programmes and files

Programmes

coauthorAnalysis.py – this performs the co-author analysis. Within the programme you need to specify the research council being analysed, the type of analysis (individual or institution/university) and for the NERC data the subset of the data to be analysed (Phase I, II or III – these are three time periods when the number of NERC panels was consistent).
Requires:

- *Cleaned data*
- *yearsOfPublication_council.txt* (produced by *processScopusData.py*)
- *coauthorJSON_council.txt* (produced by *processScopusData.py*)

Produces:

- *council coauthor analysis – phase.txt* (the results for each coauthor analysis)

jointAnalysis.py – this performs the individual and institution analyses. Within the programme you need to specify the research council being analysed, the type of analysis (individual or institution/university) and for the NERC data the subset of the data to be analysed (Phase I, II or III – these are three time periods when the number of NERC panels was consistent).
Requires:

- *Cleaned data*
- *yearsOfPublication_council.txt* (for individual analysis)(produced by *processScopusData.py*)

Produces

- *council individual analysis – phase.txt or council university analysis – phase.txt*

processScopusData.py – takes the Scopus data and generates a list of all authors that have co-authored a paper with each panel member in each year and the years each panel member has published in.
Requires:

- *Scopus downloads - council*

Produces

- *coauthorJSON_council.txt*
- *yearsOfPublication_council.txt*

scopusAffiliationID.py – queries the Scopus database to obtain the affiliation ID
Requires:

- *University names list.txt*

Produces

- *Scopus affiliation IDs.txt*. This is then manually edited to fill in some missing gaps; the resultant file is *Scopus affiliation IDs – updated.txt*

scopusCrawler.py – programme to query the Scopus database using the affiliation name, last name of the author and first initial. It downloads the files and puts them in folders with hierarchy of council then individuals in the form – l*ast name_initial_institution.txt.* These files are then processed using *processScopusData.py*
Produces

- l*ast name_initial_institution.txt.*

simplifyUniversityName.py – a set of modules that simplifies the university name to remove the ambiguity between, for example, the University of Sussex and Sussex University; in this case it reduces the name to Sussex. It also contains a routine to convert the name into the unique Scopus Affiliation ID using the file *Scopus affiliation IDs - updated.txt* produced by *scopusAffiliationID.py*.
Requires:

- *Scopus affiliation IDs - updated.txt*

Test for difference.nb – Mathematica notebook which calculates the weighted mean difference between the rates on and off panel bootstrapping to yield the confidence interval; it reads each of the results files. It also includes routines to assess the power of the analysis.

universityNames.py – a programme which goes through all the panel and grant files collating the names of the universities and institutes. It simplifies the name and gives all the alternatives that map to the simplified name. It uses the module *simplifyUniversityName.py.* It generates the file *University names list.txt*
Requires:

- *Cleaned data*

Produces:

- *University names list.txt*

Data
Cleaned data – these give the membership of the panel and the grants awarded for each council.

coauthorJSON_council.txt - gives a list of all coauthors for each year for each individual panel member.

yearsOfPublication_council.txt - gives the years between which an individual was publishing.

University names list.txt – gives the simplified name of the institution and alternatives