GePi - a batch tool for fully automatised selection of text based protein interaction reportings

Erik Fässler, Sascha Schäuble

JULIE Lab, FSU Jena

May 9, 2016

Abstract

PubMed / Medline (abstracts) and PMC (full texts) contain a substantial number of gene / protein interactions hitherto not reported / covered by public databases. Commonly researchers need to mine informationen manually by querying these databases and check abstracts / full texts for proper and strong reported relationships between genes / proteins. Here, we present an easy to use webservice, which streamlines / automates the search for a meaningful reported **ge**ne – **p**rotein **i**nteraction (gepi). Either one or two lists of valid identifiers are accepted as input. The output is automatically generated and provides a spreadsheet like summary of recognised reported gepis, including heavy use of web links and clear presentation of individual sentences, where the reported gepi is identified.

Hence, researchers are notably supported in their daily labor by automatically mining the literature for meaningful gepis and thus experierence an accelerated workflow by using the presented webservice.

1 Key steps

Summary of key steps. Detailed out later. This is not a strictly ordered list, e.g. frontend and backend are not necessarily in the right order.

1. evaluate GeNo
2. frontend
   - accept one or two arbitrary long input lists
   - one list: objective: find gepis between all given items
   - two lists: objective: find gepis between each item of list 1 and each item of list 2
     - as a start, accept only curated swissprot IDs
   - summarise output
     - spreadheet like result collection
     - basic graphics (bar, pie chart)
3. backend
   - process SwissProt IDs
   - use GeNo to find / tag genes and proteins
   - use BioSem to find / tag gepis

or find gepis between list elements and non-list-elements

2 Details

2.1 Evaluate GeNo

GeNo runs too conservatively due to a threshold. Either check whether its importance is less important with subsequently applying BioSem or change the way GeNo tags genes / proteins.

2.1.1 Performance in conjunction with BioSem

Originally BioCreative2 Train data, but "real world parameters" have been proposed whose exact origin and reasoning is unknown

GeNo is too conservative for finding / tagging genes and proteins and hence, misses a lot. A a priori set threshold is responsible for whether a hit is strong enough be report as actual gene / protein. Thus, precision is high, wheras recall is suboptimally low / average. The threshold is based / trained on available text training corpus (which?), but not for real world problems. It has never been evaluated so far in conjunction with BioSem. Consequently, evaluate GeNo for different threshold values F-score / precision / recall and apply subsequently BioSem. The rationale is that the likelihood of BioSem reporting a correct gepi is low when based on wrong genes / proteins tags provided by GeNo. For a low threshold the precision of GeNo may be low, but its recall is improved.

Hypothesis: The BioSem F-score should improve overall, as BioSem precision may not be harmed due to low probability of "gene – interaction – gene" structure in any given sentence, when one or two of the genes are false positives.

We also had another idea: We wanted to investigate if the GeNo performance raises when we use BioSem as an additional hint that a gene mention actually is a gene when it participates in an event

To do define a score for which a threshold may be used (current score is lucene where thresholds make no sense)

- Run GeNo for different thresholds
- log its performance
- log BioSem performance alongside

2.1.2 Alternate threshold determination

So far the score of tagged genes /proteins is only based on the Lucene index. The downside is that across different search queries the scores of GeNo are not comparable and thus, one constant threshold may be too conservative overall.

Alternatively, select for a given search query the best hits (by using quantiles and a sorted hit list) based on the Lucene index information.

2.2 Frontend

If no objections are present, service should run on server as web application / service. at least on the first version. A REST-like API might follow for programmatic access

2.2.1 Input lists

The batch tool should allow for two input modes: either one or two user defined lists. Only SwissProt IDs are allowed. Later on, conversion tools may be incorporated. Feedback about invalid and valid IDs will be provided. The formulars (e.g. entry boxes) for the two lists will contain text, describing briefly the two modes:

Box A: "Provide swissprot ~~genc~~ ID, one per line. If interactions among all these genes are sought, omit submitting IDs in list / Box B. If interactions partners for the given list items are sought, check the option below"

Box B: "Fill only with swissprot IDs, one per line, if interactions between items in A and B are sought."

Below these boxes the option to use both modes simultaneously (gepis among candidates of A and/or B and/or between the two, default: among A and between A and B (if B is given)) and an upload option should be available. Furthermore, an option to search for unknown interaction partners of list items A is available.

2.2.2 Output

The output should be twofold:

Firstly, a spreadsheet is given summarising the results.

Header: SwissProt ID; Name; Publication ID; Sentence

All entries shall be linked to respective sources (Swissprot, Semedico, PubMed, journals, etc.). An option to download the result as csv list is available.

Secondly, basic diagrams illustrating the results should be provided. A bar or pie chart showing the frequency of the best m hits. The count consists of genes / proteins that are most frequently found in recognised gepis.

2.3 Backend

The main question is, how often ressources are updated due to the heavy workload on server side.

2.3.1 process SwissProt IDs, provide Input for pipeline

The user provides an entry list of ideally swissprot IDs. If not, a "not valid" or "not found" warning message should be generated. If uniprot IDs are given, a warning should be generated as well ("m out of n are uniprot IDs and omitted").

The valid swissprot IDs should be furthermore enriched with homology information as follows:

1. Get gene / protein name for given swissprot ID
2. Search Swissprot with that name
3. Add all swissprot hits with the same name, but different swissprot ID to the input list.

For version 1.0 only swissprot is strictly allowed.

2.3.2 Pipeline GeNo - BioSem

A UIMA pipeline connecting GeNo and BioSem is necessary. Results are precompiled for the entire available corpus of swissprot IDs and the biomedical literature.

GeNo may be re-run upon new entries in the swissprot database. If the literature corpus hasn't changed, new items should be "just" added to the already computed tagged gene names.

TODO: How often an update? For each new entry? What is the frequency of new entries in swissprot over time?

How often should BioSem be updated? Indexing is costly, needs to be considered. Thus, if literature corpus has not changed, search only for interactions of new GeNo entries with everything else and add to database.

Tricky, needs much more thinking / discussion

If literature corpus is updated, use only update liturature for new gene / protein and gepi indexing.Already done

The most expensive operations are running GeNo and BioSem where GeNo >> BioSem. Indexing isn't that bad