Knowledge-Based Access to the Bio-Medical Literature

Ontologically-Grounded Experiments for the TREC 2003 Genomics Track

Richard Tong
Tarragon Consulting Corporation
1563 Solano Avenue, #350
Berkeley, CA 94707
rtong@tgncorp.com

John Quackenbush
The Institute for Genomic Research
9712 Medical Center Drive
Rockville, MD 20850
johnq@tigr.org

Mark Snuffin
DataNaut, Inc.
11 Dudley Court, Suite 100
Bethesda, MD 20814
snuffin@datanaut.com

Abstract

The Tarragon Consulting team participated in the primary task of the TREC 2003 Genomics Track. We used a combination of knowledge-engineering and corpus analysis to construct semantic models of the interactions between genes/proteins and other biological entities in the organism, and then used automatic methods to convert these models into evidential queries that could be executed by the K2 search engine from Verity, Inc. The primary goal of our participation in the Genomics Track was to establish a performance baseline using ontologically-grounded techniques that are scalable and implementable using current commercial retrieval technology. The results from both our official submissions and subsequent experiments demonstrate that good performance can be achieved using our approach.

Overall Approach

In our approach we focused on "function" as opposed to the other aspects of the basic biology of the gene and its protein products. That is, we interpreted the primary task to be one that requires us to identify the ways in which the gene/protein is involved in the organism's behavior, as opposed to one that simply requires us to identify that some property of the gene/protein is being discussed.

The framework for constructing our semantic models is an ontology that makes a set of core distinctions between: (a) the gene/protein subsystem; (b) the organism; (c) the interactions of the gene/protein subsystem with the organism; and, (d) the documents that report on the biological entities and processes. (See Figure 1 at the end of the paper for a high-level, and much simplified, view of the ontology as a UML static structure.)

We then interpret function to be synonymous with interaction, and thus make the retrieval task one of finding documents that use language that describe, or report on, these interactions.

We used the PubMed corpus and the GeneRIFs from the training set to construct a "lexicon of interaction" (LoI) which was hand-edited for consistency and validity, and then used as input to the automatic, evidential query generation process.

To run the experiments, we created a minimal XML variant of the PubMed records (see Figure 2) and indexed them using Verity's K2 search engine.[1] We used the output of the evidential query generation process as input to the K2 engine, and ran the fifty queries (i.e., one for each gene in the test set) against the indexed collection. The results from the K2 engine were then converted into the standard TREC format for submission to NIST.

Official Submissions

Our experimental strategy prior to the official submissions explored two main issues: (a) detection and recognition of genes/proteins; and, (b) effective ways of exploiting the LoI. Based on our experiments with the training data, we settled on a single strategy for name detection and recognition, and on two strategies for exploiting the LoI. Our official runs (*tgnBaseline* and *tgnVariant1*) reflect this two-fold strategy.

In both our official submissions, we modeled gene/protein names by focusing on the symbols (both the `OFFICIAL_SYMBOL` and the `ALIAS_SYMBOL`) and then creating a set of regular expression variants based on treating all punctuation as optional and also allowing for potential whitespace or punctuation when the symbol character sequence changes from

[1] See: http//www.verity.com/ for basic information about the K2 family of products.

alphabetic characters to numerical character, and vice versa.

So for example, in Topic 2 ("E2F transcription factor 1"), the symbols get mapped to the regular expressions:

```
E2F1     =>    E_2_F_1
RBP3     =>    RBP_3
E2F-1    =>    E_2_F_1
RBBP3    =>    RBBP_3
```

where "_" denotes an optional single space or punctuation character. Note that here, as with many other symbol sets, the transformation produces equivalent regular expressions. We remove any such duplicates to create a final set of expressions for each gene.

How best to exploit the gene/protein names in our models, is more problematic. Based on our experiments with the training data, we decided to use all of the `OFFICIAL_GENE_NAME`, the `PREFERRED_PRODUCT`, the `PRODUCT` and the `ALIAS_PROT` (if they are different) as part of the model. In many cases, however, we hand-edited the names to remove "annotation" or to extract alternates. So, for example in Topic 7, the official name:

```
syndecan 4 (amphiglycan, ryudocna)
```

becomes a three-fold set of names:

```
syndecan 4
amphiglycan
ryudocna
```

Once the name data is processed into our standard form, we automatically generate a sequence of K2 topic fragments such as:

```
tgn_geneName_2 <Or>
* 1.00 <Many><Phrase>
** 'E2F'
** 'transcription'
** 'factor'
** '1'
* 0.90 <Many><Phrase>
** 'retinoblastoma'
** 'associated'
** 'protein'
** '1'
```

which defines the gene name specification for Topic 2 and where notation like `<Or>` denotes an operator in the Verity Query Language (VQL).

The function component of our model leverages the LoI by creating three sub-modules that capture verbs, verb phrases, and general vocabulary that are related to function. As noted earlier, this initial LoI was created using a mix of corpus analysis techniques and knowledge engineering methodologies, and was developed to give us a basis for exploring a more formal linguistic analysis derived from our semantic models.

The LoI contains verbs such as "upregulate" and "phosphorlyate", verb phrases such as "localize in" and "mechanism for", as well a small set of mostly nouns, such as "pathway" and "antagonist", that relate to biological entities typically involved with functional behavior. In VQL this part of the model becomes (somewhat simplified for presentation purposes):

```
tgn_function_lexicon <Accrue>
* 0.60 tgn_function_vs
* 0.80 tgn_function_vps
* 0.20 tgn_domain_lex
```

Finally, we modeled the species constraint as a test for the presence of the corresponding MeSH keyword. So, for example, to be a candidate for retrieval for a Homo Sapiens gene, we check to see if the keyword "human" appears anywhere in the MeSH tags. In VQL this test becomes:

```
"Human" <In> /zonespec = "MH"
```

The overall query model for a gene topic is then essentially just a conjunct of the gene name and symbols, the function model, and the species test. In VQL this becomes (again somewhat simplified):

```
tgn_trecgenQuery_1 <And>
* 1.00 _isHuman
* 1.00 <Sum>
** 0.80 <And>
*** 1.00 tgn_geneModel_1
*** 1.00 tgn_function_lexicon
** 0.20 _queryProximity_1
```

where we also show a component of the model (here `_queryProximity_1`) that tests for the proximity of the gene model (here `tgn_geneModel_1`) and the function model (here `tgn_function_lexicon`). In training we found that this improved our overall scores, as measured using the `trec_eval` scoring program.

Using this core model and the two variant function models, our official scores for the two submissions (i.e., *tgnBaseline* and *tgnVariant1*) are shown in Table 1. Note that these gave just about the same overall performance, with *tgnBaseline* doing slightly better on Average Precision, and *tgnVariant1* doing slightly better on R-Precision.

Both runs, though, were better than the overall median scores reported (i.e., 0.2117 for Average Precision), with 37 and 36 individual topics (respectively) getting Average Precision scores greater than the median individual topic score.

Failure Analysis

Our preliminary failure analysis of the official runs showed that there were two main causes of poor performance (relative to the median published scores).

First of all, we had some significant recall failures. Overall we only retrieved 463 of the possible 566 relevant documents, and while in most cases we missed just one or two, we did have more serious failures. For example, for Topic 7 we missed all 7 relevant documents, and for Topic 37 we missed 37 of the 61 relevant documents.

We analyzed each failure and identified whether it was due to either: (a) a failure to detect the gene/protein; (b) a failure to identify the "function" being discussed; or, (c) a failure of the species test. In a few cases there were multiple causes of the failure. Overall though, of the 103 non-retrieved document, we attributed 81 failures to name recognition, 16 to function, and 13 to the species test.[2]

The second major performance failure was ranking failure. That is, our inability to get the relevant documents sufficiently high in the retrieval ranking. At this point in our investigation, we are less concerned with this issue since we believe, as do other groups[3], that the GeneRIFs we have been using as the "ground truth" in this exercise significantly under-represent the amount of "RIF-able" material in the collection.

Alternate Experiments

Given the failure analysis, we experimented with a number of alternate name detection/recognition algorithms. These were all variants of what we might call "token n-gram" methods in which, instead of attempting to match the exact name as a phrase (e.g., "ETF transcription factor 1"), we explored various forms of sub-string matching that involve both ordered and un-ordered matching of tokens in the name.

The alternate run labeled *ptVar01* in Table 1 uses a combination of ordered bi-grams and un-ordered k-grams (where k+1 is the number of tokens in the name). Note that we now find 523 of the 566 relevant documents and also retrieve many more documents than before—30,605 as compared to 7,758. Both Average Precision and R-Precision are better than either of the official submissions, although not by very much.

The other alternate run reported here, labeled *ptVar02* in Table 1, is the first is a series aimed at seeing if exploiting document structure can improve performance. This run simply adds an additional test to the model used in *ptVar01* to check if the full name, or one of the symbol variants, of the gene/protein appears in the title of the document (i.e., in the <TI> field), increasing the retrieval score of the document if it does.

Note that this simple extension produces a significant jump in performance over the *ptVar01* model—a 13.13% increase in the Average Precision, and a 19.66% increase in the R-Precision.

This is a surprising result given the fact that the documents are all abstracts, and suggests to us that the selection of the set of GeneRIFs used as the ground truth was heavily influenced by the titles of the original documents.

Ground Truth and Other Observations

A key element of the approach we adopted for he TREC 20003 Genomics main task was to model the concept of "function" directly. Yet as the results presented at the TREC meeting show, it was not necessary to model function in order to do well on the task. In fact, non of the top three best performing systems used any explicit representation of function.

This suggests to us that, in the context of MedLine abstracts, almost any mention of the gene/protein is likely to be relevant to function under the very broad definition we were working with (i.e., "MEDLINE references that focus on the basic biology of the gene or its protein products from the designated organism. Basic biology includes isolation, structure, genetics and function of genes/proteins in normal and disease states.").

Coupled with the fact that the GeneRIFs we have relied on for test and evaluation obviously under-report the relevant material, we think it is safe to say that we cannot draw too many significant conclusions from this initial venture into the genomics arena.

Nevertheless, this was definitely a worthwhile exercise and helped us validate our basic approach. At the same time, it made us aware of the many special issues associated with this field (e.g., name variation).

We look forward to TREC 2004 in which we can address a problem that is better motivated by the needs of practicing biologists, and for which we have a more defensible evaluation framework.

---

[2] Of these 13 species failures, our analysis suggests that at least 11 are due to incorrectly labeled GeneRIFs with respect to species.

[3] For example, the initial analysis reported by Bill Hersh, Sarah Corley, Ravi Teja Bhupatiraju in "Relevance Analysis for Primary Task of TREC Genomics Track" distributed via the TREC Genomics mailing list on 2003.10.13 shows that over 40% of the documents they retrieved were "RIF-able" but not labeled by the ground truth.

Figure and Tables

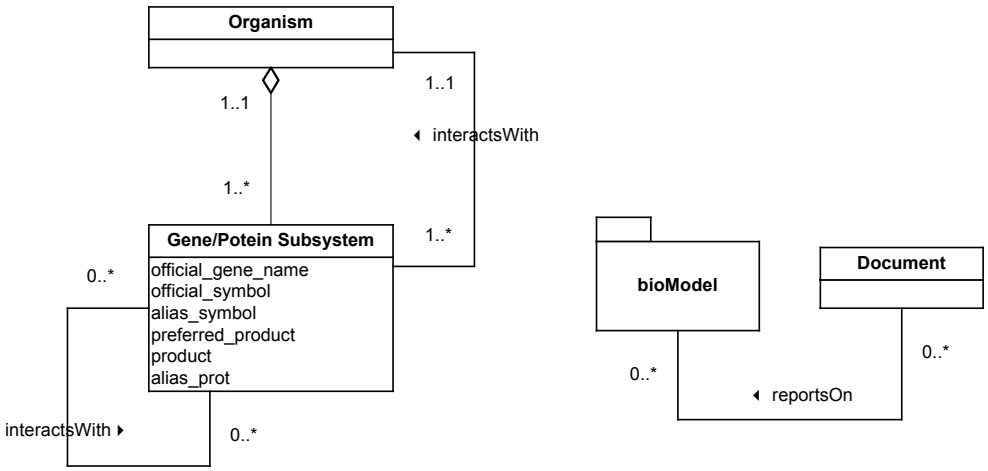

Figure 1: Simplified Ontology

```
<PubmedArticle>
<PMID>11727758</PMID>
<DCOM>20020520</DCOM>
<TI>
Opiates promote T cell apoptosis through JNK and caspase pathway.
</TI>
<AB>
Opiate addicts are prone to recurrent infections. In the present study we
evaluated the molecular mechanism of opiate-induced T cell apoptosis. Both
morphine and DAGO ([D-Ala2,N-Me-Phe4,Gly5-ol]enkephalin) enhanced T cell
apoptosis. Morphine as well as DAGO activated c-Jun NH2-terminal kinase (JNK) in T
cells. Moreover, opiates increased the expression of ATF-2. a specific substrate
for JNK and P38 mitogen activated kinases (MAPK). Furthermore, opiates attenuated
extracellular signal related kinase (ERK) in T cells. Both morphine and DAGO
cleaved pro-caspases 8, 9, and 10 and generated caspases 8, 9 and 10 (active
products). Morphine as well as DAGO also cleaved poly-(ADP-ribose) polymerase
(PARP) into 116 and 85 kD proteins indicating the activation of caspase-3. These
results suggest that opiate-induced T cell apoptosis may be mediated through the
JNKcascade and activation of caspases 8 and 3.
</AB>
<MH>Apoptosis/drug effects/physiology</MH>
<MH>Caspases/*metabolism</MH>
<MH>Enkephalin, Ala(2)-MePhe(4)-Gly(5)-/toxicity</MH>
<MH>Enzyme Activation/drug effects</MH>
<MH>Human</MH>
<MH>In Vitro</MH>
<MH>Jurkat Cells</MH>
<MH>Mitogen-Activated Protein Kinases/*metabolism</MH>
<MH>Morphine/toxicity</MH>
<MH>Narcotics/*toxicity</MH>
<MH>Support, U.S. Gov't, P.H.S.</MH>
<MH>T-Lymphocytes/*cytology/*drug effects/enzymology/immunology</MH>
</PubmedArticle>
```

Figure 2: Example `<PubmedArticle/>` XML Format

Table 1: Summary Results for Official Submissions and Selected Alternate Experiments

	tgnBaseline	***tgnVariant1***	***ptVar01***	***ptVar02***
# Docs Retrieved	7758	7758	30605	30634
# Docs Relevant	566	566	566	566
# Docs Rel_ret	463	463	523	532
Interpolated R-P:				
at 0.00	0.5721	0.5606	0.5799	0.5492
at 0.10	0.4975	0.4922	0.5398	0.5284
at 0.20	0.4179	0.4192	0.4952	0.5081
at 0.30	0.3707	0.3577	0.4222	0.4533
at 0.40	0.3298	0.3193	0.3348	0.4062
at 0.50	0.3150	0.3086	0.3060	0.3875
at 0.60	0.2622	0.2616	0.2519	0.3326
at 0.70	0.2067	0.2083	0.2041	0.2559
at 0.80	0.1614	0.1637	0.1646	0.2125
at 0.90	0.1220	0.1214	0.1095	0.1393
at 1.00	0.0992	0.0998	0.0922	0.1156
Average Precision	**0.2837**	**0.2791**	**0.2917**	**0.3300**
Precision:				
at 5 docs	0.2640	0.2720	0.3000	0.3120
at 10 docs	0.2180	0.2220	0.2280	0.2420
at 15 docs	0.2000	0.1973	0.1933	0.2253
at 20 docs	0.1760	0.1780	0.1760	0.1990
at 30 docs	0.1473	0.1480	0.1493	0.1713
at 100 docs	0.0752	0.0758	0.0754	0.0818
at 200 docs	0.0426	0.0422	0.0449	0.0472
at 500 docs	0.0184	0.0184	0.0205	0.0207
at 1000 docs	0.0093	0.0093	0.0105	0.0106
R-Precision	**0.2850**	**0.2852**	**0.2858**	**0.3420**