Hypothesis Generation in Large-Scale Event Networks

Kai Hakala[1], **Farrokh Mehryary**[1], **Suwisa Kaewphan**[1,2], **and Filip Ginter**[1]
[1]University of Turku, Turku, Finland
[2]Turku Centre for Computer Science (TUCS), Turku, Finland
first.last@utu.fi

Abstract

Hypothesis generation from literature is among the most prominent goals of the BioNLP research community. The existence of EVEX, a large-scale event network mined from the entire available biomedical literature, opens the possibility to cast this task in a supervised machine learning setting, defining it as the prediction of edges in this network, based on features from their network context.

In this paper, we study the task from two perspectives. First, we build a machine learning system which predicts novel pairwise relationships in the EVEX network and evaluate its performance using both the standard measures as well as through a manual inspection on a subset of the output. And second, we analyze and discuss the issues in evaluation arising from cross-validation in densely connected graphs with uneven edge distribution.

We find that the task is learnable, achieving performance clearly above baseline. Further, a manual inspection of predictions not found in the EVEX network showed several candidate pairs, whose interaction could be verified in the literature. These pairs hint at the possibility that true novel interacting pairs were identified by the system as well, even though further work is necessary to confirm whether that is indeed the case.

1 Introduction

Hypothesis generation based on literature mining is among the most prominent goals of the BioNLP research community. Already over 20 years ago, the legendary ARROWSMITH system (Swanson, 1988) identified novel association candidates by combining the information from entity pairs frequently co-occuring in the literature (Bekhuis, 2006). The work of Swanson, and many others, was based on the statistics of term co-occurrence in text. To increase the recall of low-frequency associations, subsequent work has focused on a more detailed extraction of pairwise interactions of (mainly) genes and proteins from individual sentences (Pyysalo et al., 2008; Tikk et al., 2010). Such extraction of interacting pairs has the advantage that even single assertions can be extracted, without the need for sufficiently high co-occurrence. These methods are, however, often largely restricted to the extraction of untyped, undirected pairs, i.e. an association is postulated, but no additional knowledge regarding its type is given. Finally, methods for the extraction of detailed *events* have been introduced, mainly as the outcome of the BioNLP Shared Tasks on Event Extraction (Kim et al., 2009; Kim et al., 2011; Nédellec et al., 2013). The events are detailed, recursive structures that provide a more faithful representation of the semantics of the underlying text. Event extraction systems have subsequently been applied on a large scale to the collection of PubMed abstracts and the open-access section of PubMed Central full-text articles (Björne et al., 2010; Gerner et al., 2012). EVEX (Van Landeghem et al., 2013), presently the only publicly available large-scale event collection, serves as the basis of this study and is discussed in more detail in Section 2.

The availability of EVEX as a large-scale network, with genes and gene products (GGPs) as the nodes and their relationships as the edges, allows us to study the problem of hypothesis generation at a large scale and in a machine learning setting. Rather than relying on a set of pre-defined patterns, such as the triangular pattern used by Swanson which postulated the hypothetical association

A–C given the identified associations *A–X* and *X–C*, we define a number of features extracted from the network context and train a supervised classifier. This allows us to incorporate more information into the classification process.

Given a candidate pair of nodes not already connected by an edge in the network, the task is to predict the existence of a potential edge, or edges, between the two nodes and possibly also the nature (type) of the predicted relationship. Features for this prediction task are extracted from the existing network neighborhood of the candidate pair, in particular from short paths in the network that connect the two nodes. Edges already existing in the network are then used as positive examples in training. In this paper, we will explore both the simpler task of predicting whether an edge exists or not, as well as the more complex multi-label task of predicting also the type of the newly predicted edges.

2 Data

The data we use is extracted from EVEX, a large-scale literature mining resource built on top of the set of events extracted from all PubMed abstracts and PubMed Central Open Access full-text articles, using the TEES system (Van Landeghem et al., 2013; Björne et al., 2012). A feature of EVEX particularly important for this current study is that it provides a *network view*, where GGPs are normalized to their respective Entrez Gene identifiers using the GenNorm system (Wei et al., 2012), and the complex recursive events are reduced into pairwise relationships with the coarse-grained types of *Regulation, Binding*, and *Indirect regulation* and 29 fine-grained types such as *Regulation of phosphorylation* and *Indirect catalysis of hydroxylation*. This network view thus abstracts away some of the complexity of the recursive events and allows modeling the problem as a simple edge prediction in directed graphs. Figure 1 illustrates a tiny part of the human gene regulatory network extracted from the EVEX resource.

In the EVEX *network view*, the individual event occurrences extracted from text are aggregated, i.e. a single edge in the network stands for all individual events that represent this relationship anywhere in the literature. This is possible because the GGP symbols are normalized into Entrez Gene identifiers and all edges of the same type and direction between the same Entrez Gene identifier pair can be merged. This has the major advantage of allowing the use of features from all the available literature when predicting new relationships, not restricting ourselves to a single sentence, or a single article.

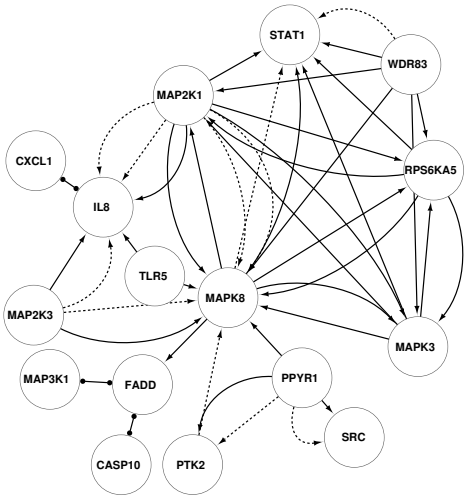

Figure 1: A tiny part of the highly connected network extracted from EVEX for human gene/protein interactions. Circle-terminated connections indicate binding and arrows indicate regulation. Indirect regulations are presented with dashed lines while direct regulations are presented with solid ones.

The complete EVEX network consists of 819,348 unique edges among 48,061 unique GGPs from a large number of different organisms. To deal with a smaller, yet biologically motivated problem for this initial study, we selected the subnetwork formed by all human genes (judged by their Entrez Gene identifier) and only consider the three coarse-grained types, rather than the 29 fine-grained types available in EVEX. This human gene network consists of 13,418 nodes and 265,738 directed edges. As illustrated in Table 1, the network is densely connected, with 97.6% of nodes belonging to a single large connected component. To simplify processing, we remove the 317 nodes that belong to connected components with less than 8 nodes, and the 76 edges among these nodes. The 212 connected components with only a single node reflect the self-interacting genes with no known interactions with other genes in the EVEX database.

# nodes	# components
1	212
2	38
3	6
4	2
6	1
7	1
13,091	1

Table 1: The distribution of connected components in the network, showing that essentially the entire network is spanned by a single connected component with 13,091 nodes.

3 Methods

Casting the task in a straightforward supervised machine learning setting, we need to specify what our positives and negatives are. A positive example is a pair of nodes in the network which is connected with an edge. In the classification, we will use features extracted from paths two or three edges long that connect the two nodes in the network.

As with many similar problems, there is no apriori given set of negative examples. Instead, any pair of nodes that is not directly connected in the network can be technically considered as a negative example. This would, however, have two unwanted consequences: First, the number of such negative examples would be enormous in comparison to the number of positive examples, and second, most arbitrary node pairs are distant in the network and obviously unrelated. The classification problem would thus become trivial if trained and evaluated on such negative examples, and its performance would not be very informative. Rather, we thus restrict the selection of negative examples to the "interesting" node pairs that are not connected by a direct edge in the graph, but are connected by at least one path of at most three edges. In this way, we focus on the more realistic problem of predicting novel relationships for node pairs that are closely connected in the network.

Current state-of-the-art event extraction systems perform in the range of 40–50% in terms of recall. Due to this fairly low retrieval rate some of the examples labeled as negatives in training are in fact false negatives in the underlying EVEX network, and are bound to add noise to the training and evaluation data. To diminish their effect, we further refine the data by excluding negative gene pairs that co-occur in at least one sentence. Since, as was shown for example in the Genia Shared Task data, statements of interactions rarely cross the sentence boundary, this filtering step will remove most of the EVEX false negatives. The final set of negatives used in training and evaluation is thus constituted by pairs that are connected in the network by at least one path of length at most three edges, and that have not co-occurred in a sentence.

Comparing the average number of paths in the network that connect candidates in the final set of positives (32,077), the final set of negatives (427), and the (currently discarded) set of negatives where the candidate GGPs co-occur in a sentence (8,602), reveals large differences, in particular further confirming that the currently discarded negatives probably contain a non-trivial proportion of actual existing interactions that the EVEX text mining system failed to extract. Even though these examples are excluded in the current evaluation so as to avoid the added noise in the data, future work should focus on assessing their importance in hypothesis generation as well as in improving the recall of the EVEX resource.

4 Features and Classification

To solve the binary classification problem of predicting the existence of an edge, we train a linear support vector machine using the SVM-light library (Joachims, 1999). The features used are based on the paths between the nodes, limiting to only the paths of length two and three. Two feature types are used:

1. For every unique path type, defined as the concatenation of edge types and directions along the path, the number of paths of this type connecting the pair of GGPs is given.

2. For every unique path type of length two edges, the maximum of EVEX confidence scores of the edges in the path. The confidence scores given in EVEX for the individual edges reflect the reliability of the underlying events being correctly extracted from the text.

The first set of features is purely based on the structure of the graph and could be used with various graphs constructed from different data sources. The second set, however, is unique to the underlying text mining resource, providing information that cannot be acquired from other type of

gene regulatory networks. The performance gain of these feature types is discussed in Section 5.3. It is worth noting that neither of these feature types encode information about the intermediary nodes in the paths nor the textual context where the interactions have been seen. As will be discussed in detail in Section 5.2, this is particularly important in the cross-validation setting, where it is difficult to avoid paths crossing between training and testing sets without substantially changing the characteristics of the data.

The optimization of the classifier C parameter was done with a grid search against a development set. As the natural distribution of positive and negative examples is very tilted, we oversample positive examples to create training data with a 1:10 proportion between positive and negative examples. No such oversampling is done for the development and test sets, naturally.

The more complex problem definition, where event types are also predicted, can be formalized as a multi-label classification task. In this case we use a one-vs-all classification approach implemented with the scikit-learn library (Pedregosa et al., 2011) and a linear support vector machine. The same features are used in both tasks.

5 Results

5.1 Baseline

Even though we select the negatives to be connected with a path of at most three edges, there is still a clear difference in the density — i.e. the number of paths in the network that connect the the two nodes — between the positive and the negative examples. The positive examples have on average a notably higher number of connecting paths. The distributions of the path counts are illustrated in detail in Figure 2. The histograms show that the distribution of the negative examples resembles an exponential distribution whereas the positive examples show a heavy-tailed distribution. This is naturally a difference which a classifier can learn to exploit. To test how predictive the path count is of the classification outcome, we train a baseline classifier which is only given the total number of connecting paths.

5.2 Test Protocol

All experiments are carried out using the 10-fold cross-validation protocol, whereby the network is split into ten sub-networks, of which eight are used for training, one for parameter optimization, and one for testing. 20,000 pairs with at least one connecting path of at most three edges are randomly sampled from each partition to form test sets with a natural distribution of positive and negative examples. The results on the ten sets are then averaged. Unlike in most machine learning problems where individual instances are largely independent, the densely connected event network complicates the 10-fold split substantially. The obvious approach of splitting the nodes randomly is not practical because for any given node, 90% of its neighbors will be assigned to a different set than the node itself, while for feature generation and testing it would be desirable for the node as well as its neighbors to be assigned to the same set. Rather than splitting the nodes into sets randomly, we apply the METIS toolchain for graph partitioning (Lasalle and Karypis, 2013), which heuristically splits the network into roughly equally-sized parts while minimizing the number of edges crossing among the parts.

An issue with splitting the graph into partitions roughly equally-sized with respect to the number of nodes is that a small number of extremely densely connected hub nodes causes large variations in edge density in the resulting sets and, as will be shown later, subsequent variation in the results. The METIS algorithm allows for weights to be given to the nodes, which affects the division to create graph partitions with roughly equal sum of node weights. Weighting the nodes by their degree, we can thus subdivide the graph into partitions with a roughly equal number of edges, thus balancing the edge density rather than node density. To illustrate the difference, in Table 2 we show the number of nodes and edges in two METIS-based 10-fold splits corresponding to the two aforementioned strategies. Note the particularly disturbing fold no. 1 in the unweighted strategy, which has an order of magnitude more edges than any of the other nine folds. This partition includes several well-studied genes such as TNF-alpha, IL-6 and insulin, all with hundreds of known interaction partners in the EVEX resource.

Another problem stems from the fact that, regardless of the strategy used to divide the nodes into subsets, there will be a number of edges spanning across these subsets. Of particular concern are edges spanning between the training and the test set in a given fold of the 10-fold protocol.

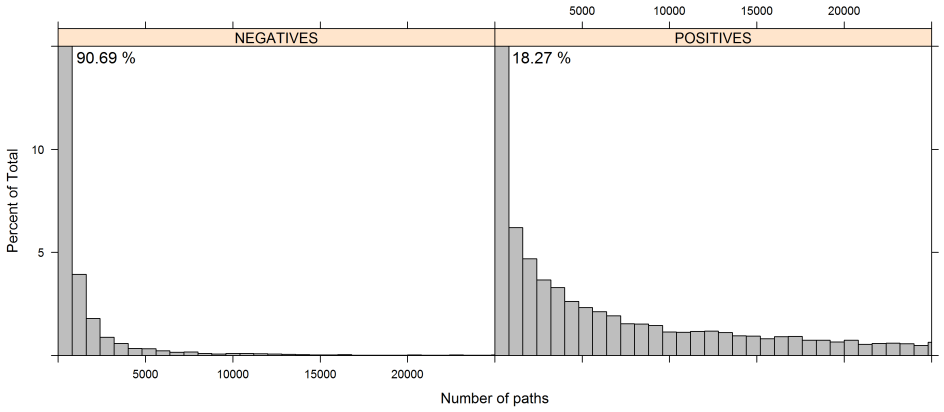

Figure 2: Distributions of positive and negative examples in terms of the connecting path counts. The y-axis has been limited to 15% and the actual heights of the bins exceeding this limit are denoted in the figure.

	Unweighted		Weighted	
Fold	Nodes	Edges	Nodes	Edges
0	1,335	3,713	904	7,664
1	1,348	79,005	1,287	9,678
2	1,320	4,263	1,892	7,677
3	1,348	9,959	1,369	8,326
4	1,302	3,198	1,256	11,616
5	1,278	1,738	1,140	8,668
6	1,289	2,129	921	8,533
7	1,296	2,273	1,792	8,333
8	1,283	3,116	1,180	6,786
9	1,292	2,221	1,350	8,421
Total	13,091	111,615	13,091	85,702

Table 2: The distribution in terms of the number of nodes and edges when splitting the network into 10 folds with roughly equal node count (unweighted) and roughly equal edge count (weighted) using the METIS algorithm.

While the obvious "safe" course of action would be to remove all edges that connect nodes between the training and test data, this has a notable impact on the data exactly because it is so densely interconnected. This is again illustrated in Table 2, which shows the edge counts for the 10 partitions when edges spanning across partitions are removed. In the unweighted set, 58% of edges are removed, and in the weighted set, full 68% of edges are removed. Removing edges spanning across the 10 folds thus clearly substantially affects the properties of the underlying data. Note that while only removing edges spanning between the training and test set in every iteration of the 10 fold evaluation strategy is also an option, this would result in substantially skewed distributions between the training and test data, and we thus do not consider this approach further.

To assess the impact of these choices, we carry out evaluation on all four combinations, i.e. splitting to balance the number of nodes versus number of edges, and preserving or removing the edges spanning between the folds in the 10-fold protocol. The four resulting divisions and their salient characteristics are summarized in Table 3.

5.3 Classification Results

In the evaluation, we compare classifiers with three different feature sets on the four network partitioning strategies introduced in Section 5.2. The baseline classifier uses only one feature which encodes the total number of paths connecting the candidate pair. A second classifier utilizes counts of unique path structures, and a third classifier introduces also features encoding confidences of the individual edges, as extracted from EVEX. Precision, recall, and F-score averaged over the 10 folds for the three classifiers are shown in Table 4.

Several observations can be made: To begin with, the performance of the baseline classifier is very poor in evaluation strategies with equal node count partitioning, achieving F-scores of 7.36% and 3.65%. This is most visible when edges spanning across partitions are retained in the data, where the baseline classifier obtains an F-score of 0.0 in four folds out of ten. This is likely because the baseline classifier can only rely on the number of paths, which substantially differs among

Dataset name	Positives Frequency (%)	Negatives Frequency (%)	Average paths count (Total)	Average paths count (Positives)	Average paths count (Negatives)	Sample STD (Total)	Sample STD (Positives)	Sample STD (Negatives)
unweighted/remove	3.86	96.14	1289.00	12590.00	446.20	10325.95	42184.00	3267.35
weighted/remove	1.72	98.28	195.40	3086.00	104.40	929.10	5134.79	331.40
unweighted/keep	3.86	96.14	1543.00	16480.00	943.10	11358.70	52216.45	3915.70
weighted/keep	1.72	98.28	1094.00	25050.00	673.60	7932.11	52306.50	2403.92

Table 3: The salient characteristics of the four ways to construct the 10-fold split of the data.

Classifier	Precision	Recall	F-score
unweighted/remove			
B	3.98	79.88	7.36
S	50.81	31.69	31.94
C	82.99	49.79	54.30
weighted/remove			
B	54.96	34.22	34.14
S	60.84	46.44	49.81
C	62.69	52.88	56.47
unweighted/keep			
B	1.89	60.00	3.65
S	58.08	38.05	41.61
C	78.64	28.92	41.20
weighted/keep			
B	59.31	29.43	33.50
S	67.72	46.26	53.76
C	60.93	49.98	53.33

Table 4: Averaged precision, recall and F-score over all test partitions for each evaluation strategy. B = baseline classifier, S = classifier with path structure features, C = classifier with confidence and structure features.

the 10 folds with equal node count partitioning (see Table 2). Especially with the dense fold no. 1, the network density and therefore path count differs substantially between the training and test set, leading to the poor classification performance. With partitions balanced by edge counts, on the other hand, the baseline classifier performance is much higher, with F-scores of over 30%.

Classifiers using structure and confidence features clearly outperform the baseline in all evaluation strategies, indicating that this problem indeed is learnable and that the paths themselves, not only their overall count, provide useful information to the classification. Interestingly, the confidence features decrease the performance in evaluation strategies where paths are allowed to span across folds. As these features provide clear improvement when the folds are completely independent, further work is required to examine whether it is the case that confidence features are beneficial only with sparser networks, leading to potential gains in networks for less studied organisms.

For the more complex task of predicting also the edge type and direction we select only one evaluation strategy: balanced edge counts with edges spanning over folds. This method is chosen as it provides a sensible baseline and low variation between the folds, yet it reflects the natural density of the graph well. As the edge types are not exclusive, multiple labels can be predicted for each example, reflecting cases where several relationships exist simultaneously between the candidate GGPs, for example both Binding and Regulation.

Results for the multi-label classification task are shown in Table 5. As can be expected, the performance for this task is lower than for the simple binary classification task. As with the binary task, the performance of the classifiers is substantially higher than for the baseline. An interesting difference can be observed between the performance of predicting binding versus regulation. As binding edges are symmetric and the most common out of these types, predicting them should be intuitively the easiest. However, the baseline classifier obtains higher scores for regulation events. On the other hand, the classifier with path structure features performs clearly better for binding edges, resulting in approximately 10pp higher F-score than for regulation edges.

Indirect regulations are clearly the hardest types to predict. This might be due to their low number in the data sets or the fact that an indirect regulation edge always originates from a complex regulation event. The confidence features do not seem to have a significant influence on the results as also observed in the binary classification task. Further investigation is needed to clarify these evaluation numbers.

Edge type	Precision	Recall	F-score
B			
Binding	63.07	9.27	14.84
Reg. >	66.01	11.36	17.74
Reg. <	61.37	14.35	21.38
Ind. reg. >	36.00	5.46	9.18
Ind. reg. <	31.83	5.76	9.56
Micro-average	60.96	10.63	16.73
Macro-average	51.66	9.24	14.54
S			
Binding	66.46	37.79	46.85
Reg. >	65.14	27.80	37.47
Reg. <	64.14	27.65	36.49
Ind. reg. >	50.67	9.59	15.82
Ind. reg. <	32.94	8.99	13.87
Micro-average	65.11	30.85	40.52
Macro-average	55.87	22.36	30.10
C			
Binding	62.33	40.43	47.92
Reg. >	65.72	28.10	37.86
Reg. <	63.85	27.59	36.03
Ind. reg. >	59.38	10.74	17.73
Ind. reg. <	34.31	9.22	14.28
Micro-average	62.40	32.24	41.34
Macro-average	57.12	23.22	30.76

Table 5: Averaged precision, recall and F-score over all test partitions for each edge type. Binding is a symmetric interaction whereas regulation and indirect regulation are directed. The direction is denoted with > and <. B = baseline classifier, S = classifier with path structure features, C = classifier with confidence and structure features.

5.4 Manual Evaluation

The false positive predictions provide an extremely interesting research target from the hypothesis generation perspective. First, some of these predictions are evaluated as false positives only because the text mining system that was used to generate the underlying data has failed to extract these relationships from the text, even though they were present. And second, some of the predictions evaluated as false positives may in fact constitute existing undiscovered relationships, identification of which, after all, is the overall goal of this work.

If some proportion of the former can be found, it may at least hint at the possibility of the latter being present among the "false" positives as well. To assess whether some of the false positives can be attributed to extraction failures of the text mining system underlying the EVEX network, we manually evaluated from each partition the 10 false positive pairs with the highest number of connecting paths, 100 examples in total. This evaluation was carried out using the edge-balanced partitioning with edges spanning across partitions and the predictions were made with the classifier using the path structure features. In this evaluation we only determined whether an interaction exists between the genes and did not consider the interaction types.

The manual evaluation was carried out in two ways: First, we searched in the EVEX resource for occurrences of every false positive pair, but this time including also event occurrences among sequence homologs of the candidate genes. Given a false positive pair *geneA–geneB*, we thus inspect all pairs *geneX–geneY* such that *geneX* and *geneA* belong to one homologous family, and similarly also for *geneY* and *geneB*. This way, we are able to detect corresponding events that are reported to occur between similar genes in other organisms, instead of focusing only on the human gene regulatory network. EVEX contains several gene family definitions — we use HomoloGene (Sayers et al., 2012) and Ensembl (Flicek et al., 2013) for the evaluation, as these specifically focus on eukaryotes and include the human genome. For 15 of the 100 pairs, we found a corresponding event among HomoloGene families. Among families based on the Ensembl resource, the number of pairs was 34. Further examining these pairs, we found that 3 out of the 15 HomoloGene-based interactions (4 out of 34 with Ensembl) could be confirmed to hold among the exact human genes predicted, but the pair was not present in EVEX because of gene symbol normalization failure. These are thus cases of successful prediction of relationships not present in the EVEX network, which could be subsequently verified in the literature. The remaining 12 interactions were either reported to happen in other organisms or they were protein complex interactions and the exact subunits were not mentioned. For instance, predicted interacting genes PTK2B and NGF are found to belong to interacting families, with the sentence *"NGF induced the tyrosine phosphorylation of RAFTK..."* supporting this prediction. Even though the family assignment has grouped PTK2B together with RAFTK, the precise relation is that PTK2B is a subunit of RAFTK and no confirmed interaction is known

between PTK2B and NGF. Nonetheless, it is intriguing to observe that the system is able to predict a hypothetical interaction close to a known interaction of related protein complexes.

In the second manual evaluation, the 100 pairs were searched from the STRING database (Franceschini et al., 2013), which combines protein-protein interaction evidence from various sources, including text mining resources, experimental data and curated databases. In this evaluation all STRING evidence above the confidence value of 0.150 (i.e. the low confidence threshold on the STRING website) was considered as a possible interaction candidate. Out of the 100 pairs 31 were found to have some evidence in the STRING database.

These results indicate that the system is able to identify correct interactions not currently present in the EVEX network.

6 Conclusions and Future Work

In this paper, we have introduced a machine-learning hypothesis generation system, based on large-scale literature mining networks and supervised learning. We have shown that the problem is indeed learnable using features extracted from the network context of each candidate pair. The classification performance is far above the random baseline as well as the baseline classifier which only considers the number of paths connecting the candidate in the network. This indicates that not only the density but also the content in the network context is used by the classifier.

In addition to the aforementioned machine learning results, we have also explored some of the difficulties associated with machine learning in densely connected networks, where independence of the individual instances does not hold in many cases, causing problems in the application of the standard cross-validation procedure. Another problematic issue is the non-uniform density of the network where even few highly-connected hub nodes may cause large variance in experimental results.

There is a number of future directions for this work. First, the EVEX network offers aggregation of events not only by their Entrez Gene identifiers, but also by gene families defined through gene sequence homology and spanning across species. Incorporating events from different organisms would allow us to include the aspects of cross-species, homology based function prediction commonly used in genome annotation. Second, we currently only utilize features from the network, but not from the underlying text. It would be of interest to explore what other features from the texts, beyond the events themselves, can contribute to the classification.

Acknowledgments

We would like to thank Sofie Van Landeghem, Ghent University, for her valuable suggestions and comments, the Academy of Finland and Turku Centre for Computer Science (TUCS) for funding the study and CSC – IT Center for Science Ltd. for computational resources.

References

Tanja Bekhuis. 2006. Conceptual biology, hypothesis discovery, and text mining: Swanson's legacy. *Biomedical Digital Libraries*, 3(1):2.

Jari Björne, Filip Ginter, Sampo Pyysalo, Jun'ichi Tsujii, and Tapio Salakoski. 2010. Complex event extraction at PubMed scale. *Bioinformatics (ISMB'2010 proceedings volume)*, 26:i382–i390.

Jari Björne, Filip Ginter, and Tapio Salakoski. 2012. University of Turku in the BioNLP'11 Shared Task. *BMC Bioinformatics*, 13(Suppl 11):S4.

Paul Flicek, Ikhlak Ahmed, M. Ridwan Amode, Daniel Barrell, Kathryn Beal, Simon Brent, Denise Carvalho-Silva, Peter Clapham, Guy Coates, Susan Fairley, Stephen Fitzgerald, Laurent Gil, Carlos Garca-Girn, Leo Gordon, Thibaut Hourlier, Sarah Hunt, Thomas Juettemann, Andreas K. Khri, Stephen Keenan, Monika Komorowska, Eugene Kulesha, Ian Longden, Thomas Maurel, William M. McLaren, Matthieu Muffato, Rishi Nag, Bert Overduin, Miguel Pignatelli, Bethan Pritchard, Emily Pritchard, Harpreet Singh Riat, Graham R. S. Ritchie, Magali Ruffier, Michael Schuster, Daniel Sheppard, Daniel Sobral, Kieron Taylor, Anja Thormann, Stephen Trevanion, Simon White, Steven P. Wilder, Bronwen L. Aken, Ewan Birney, Fiona Cunningham, Ian Dunham, Jennifer Harrow, Javier Herrero, Tim J. P. Hubbard, Nathan Johnson, Rhoda Kinsella, Anne Parker, Giulietta Spudich, Andy Yates, Amonida Zadissa, and Stephen M. J. Searle. 2013. Ensembl 2013. *Nucleic Acids Research*, 41(D1):D48–D55.

Andrea Franceschini, Damian Szklarczyk, Sune Frankild, Michael Kuhn, Milan Simonovic, Alexander Roth, Jianyi Lin, Pablo Minguez, Peer Bork, Christian von Mering, and Lars Juhl Jensen. 2013. STRING v9.1: protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Research*, 41(Database-Issue):808–815.

Martin Gerner, Farzaneh Sarafraz, Casey M Bergman, and Goran Nenadic. 2012. BioContext: an integrated text mining system for large-scale extraction and contextualization of biomolecular events. *Bioinformatics*, 28(16):2154–2161.

Thorsten Joachims. 1999. Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of BioNLP'09 Shared Task on Event Extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 1–9, Boulder, Colorado, June. Association for Computational Linguistics.

Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, Ngan Nguyen, and Jun'ichi Tsujii. 2011. Overview of BioNLP Shared Task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 1–6, Portland, Oregon, USA, June. Association for Computational Linguistics.

Dominique Lasalle and George Karypis. 2013. Multi-threaded graph partitioning. *Parallel and Distributed Processing Symposium, International*, 0:225–236.

Claire Nédellec, Robert Bossy, Jin-Dong Kim, Jung-Jae Kim, Tomoko Ohta, Sampo Pyysalo, and Pierre Zweigenbaum. 2013. Overview of BioNLP Shared Task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 1–7, Sofia, Bulgaria, August. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Sampo Pyysalo, Antti Airola, Juho Heimonen, Jari Björne, Filip Ginter, and Tapio Salakoski. 2008. Comparative analysis of five protein-protein interaction corpora. *BMC bioinformatics*, 9(Suppl 3):S6.

Eric W. Sayers, Tanya Barrett, Dennis A. Benson, Evan Bolton, Stephen H. Bryant, Kathi Canese, Vyacheslav Chetvernin, Deanna M. Church, Michael DiCuccio, Scott Federhen, Michael Feolo, Ian M. Fingerman, Lewis Y. Geer, Wolfgang Helmberg, Yuri Kapustin, Sergey Krasnov, David Landsman, David J. Lipman, Zhiyong Lu, Thomas L. Madden, Tom Madej, Donna R. Maglott, Aron Marchler-Bauer, Vadim Miller, Ilene Karsch-Mizrachi, James Ostell, Anna Panchenko, Lon Phan, Kim D. Pruitt, Gregory D. Schuler, Edwin Sequeira, Stephen T. Sherry, Martin Shumway, Karl Sirotkin, Douglas Slotta, Alexandre Souvorov, Grigory Starchenko, Tatiana A. Tatusova, Lukas Wagner, Yanli Wang, W. John Wilbur, Eugene Yaschenko, and Jian Ye. 2012. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 40(D1):D13–D25.

Don R. Swanson. 1988. Migraine and magnesium: eleven neglected connections. *Perspectives in Biology and Medicine*, 31(4):526–557.

Domonkos Tikk, Philippe Thomas, Peter Palaga, Jörg Hakenberg, and Ulf Leser. 2010. A comprehensive benchmark of kernel methods to extract protein-protein interactions from literature. *PLoS Comput Biol*, 6(7):e1000837, 07.

Sofie Van Landeghem, Jari Björne, Chih-Hsuan Wei, Kai Hakala, Sampo Pyysalo, Sophia Ananiadou, Hung-Yu Kao, Zhiyong Lu, Tapio Salakoski, Yves Van de Peer, and Filip Ginter. 2013. Large-scale event extraction from literature with multi-level gene normalization. *PLoS ONE*, 8(4):e55814.

Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. 2012. SR4GN: A species recognition software tool for gene normalization. *PLoS ONE*, 7(6):e38460, 06.