Pacific Symposium on Biocomputing 12:245-256(2007)

INTEGRATING NATURAL LANGUAGE PROCESSING WITH FLYBASE CURATION

NIKIFOROS KARAMANIS*†, IAN LEWIN*, RUTH SEAL†,
RACHEL DRYSDALE† AND EDWARD BRISCOE*
*Computer Laboratory** *and Department of Genetics*†
University of Cambridge
E-mail for correspondence: Nikiforos.Karamanis@cl.cam.ac.uk

Applying Natural Language Processing techniques to biomedical text as a potential aid to curation has become the focus of intensive research. However, developing integrated systems which address the curators' real-world needs has been studied less rigorously. This paper addresses this question and presents generic tools developed to assist FlyBase curators. We discuss how they have been integrated into the curation workflow and present initial evidence about their effectiveness.

1. Introduction

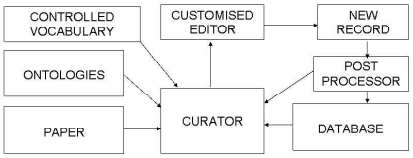
The number of papers published each year in fields such as biomedicine is increasing exponentially [1,2]. This growth in literature makes it hard for researchers to keep track of information so progress often relies on the work of professional curators. These are specialised scientists trained to identify and extract prespecified information from a paper to populate a database.

Although there is already a substantial literature on applying Natural Language Processing (NLP) techniques to the biomedical domain, how the output of an NLP system can be utilised by the intended user has not been studied as extensively [1]. This paper discusses an application developed under a user-centered approach which presents the curators with the output of several NLP processes to help them work more efficiently.

In the next section we discuss how observing curators at work motivates our basic design criteria. Then, we present the tool and provide an overview of the NLP processes behind it as well as of the customised curation editor we developed following the same principles. Finally, we discuss how these applications have been incorporated into the curation workflow and present a preliminary study on their effectiveness.

*William Gates Building, Cambridge, CB3 0FD, UK.
†Downing Site, Cambridge, CB2 3EH, UK.

Pacific Symposium on Biocomputing 12:245-256(2007)




```
! GENE PROFORMA                    Version 37:  5 Aug 2005
!
! G1a.  Gene symbol to use in database                *a :
! G1b.  Gene symbol used in paper (if different)      *i :
! G1c.  Database gene symbol to replace               *i :
! G1d.  Gene category (if gene is new to FlyBase) [CV]  *t :
! G2a.  Gene name to use in database                  *e :
....
! G15.  Internal notes  *W ::

! ALLELE PROFORMA                   Version 32: 5 Aug 2005
!
! GA1a. Allele symbol to use in database              *A :
! GA1b. Complete symbol for GA1a in paper (if different)  *i :
! GA1c. Database allele symbol to replace             *i :
! GA2a. Allele name to use in database                *e :
....
! GA14. Internal notes  *W ::
```

Figure 1. (A) Overview of the curation information flow. (B) Gene and allele proformae.

2. The FlyBase curation paradigm

The tools presented in this paper have been developed under an approach which actively involves the potential user and consists of iterative cycles of (a) design (b) system development (c) feedback and redesign [3].

The intended users of the system are the members of the FlyBase curation team in Cambridge (currently seven curators). FlyBase[a] is a widely used database of genomic research on the fruit fly. It has been updated with newly curated information since 1992 by teams located in Harvard, Indiana and Berkeley, as well as the Cambridge group. Although the curation paradigm followed by FlyBase is not the only one, it is based on practices developed through years of experience and has been adopted by other curation groups.

FlyBase curation is based on a watchlist of around 35 journals. Each curator routinely selects a journal from the list and inspects its latest issue to identify which papers to curate. Curation takes place on a paper-by-paper basis (as opposed to gene-by-gene or topic-by-topic).

A simplified view of the curation information flow is shown in Figure 1A. A standard UNIX editor with some customised functions is used to produce a record for each paper. The record consists of several proformae (Figure 1B), one for each significant gene or allele discussed in the paper. Each proforma is made of 33 fields (not all of which are always filled): some fields require rephrasing, paraphrasing and/or summarisation while others record very specific facts using terms from ontologies or a controlled vocabulary. In addition to interacting with the paper, typically viewed in printed form or loaded into a PDF viewer, the curator also needs to access the database

[a] www.flybase.org

Pacific Symposium on Biocomputing 12:245-256(2007)

to fill in some fields. This is done via several task-specific scripts which search the database e.g. for a gene-name or a citation identifier. After the record has been completed, it is post-processed automatically to check for inconsistencies and technical errors. Once these have been corrected, it is uploaded to the database.

Given that extant information retrieval systems such as MedMiner [4] or Textpresso [5] are devised to support the topic-by-topic curation model in other domains, FlyBase curators are in need of additional technology tailored to their curation paradigm and domain.

In order to identify users' requirements more precisely, several observations of curation took place focussing on the various ways in which the curators interact with the paper: some curators skim through the whole paper first (often highlightling certain phrases with their marker) and then re-read it more thoroughly. Others start curation from a specific section (not necessarily the abstract or the introduction) and then move to another section in search of additional information about a specific concept. The "find function" of the PDF viewer is often used to search for multiple occurrences of the same term. Irrespective of the adopted heuristics, all curators agreed that identifying the sections of the text which contain information relevant to the proforma fields is laborious and time-consuming.

Current NLP technology identifies domain-specific names of genes and alleles as well as relations between them relatively reliably. However, providing the curator simply with the typical output of several NLP modules is not going to be particularly helpful [1]. Hence, one of our primary aims is to design and implement a system which will not only utilise the underlying NLP processes but also enable the curators to interact with the text efficiently to accurately access segments which contain potentially useful information. Crucially, this is different from providing them with automatically filled information extraction templates and asking them to go back to the text and confirm their validity. This would shift their responsibility to verifying the quality of the NLP output. Instead, we want to develop a system in which the curators maintain the initiative following their preferred style but are usefully assisted by software adapted to their work practices.

Records are highly structured documents so additionally we aimed to develop, using the same design principles, an enhanced editing tool sensitive to this structure in order to speed up navigation within a record too. This paper presents the tools we developed based on these premises. We anticipate that our work will be of interest to other curation groups following the paper-by-paper curation paradigm.

Pacific Symposium on Biocomputing 12:245-256(2007)

3. PaperBrowser

PaperBrowser[b] presents the curator with an enhanced display of the text in which words automatically recognised as gene names are highlighted in a coloured font (Figure 4A). It enables the curators to quickly scan the whole text by scrolling up and down while their attention is directed to the highlighted names.

PaperBrowser is equipped with two navigation panes, called PaperView and EntitiesView, that are organised in terms of the document structure and possible relations between noun phrases, both of which are useful cues for curation [2]. PaperView lists gene names such as "zen" in the order in which they appear in each section (Figure 4B). EntitiesView (Figure 4C) lists groups of words (noun phrases) automatically recognised as referring to the same gene or to a biologically related entity such as "the zen cDNA". The panes are meant not only to provide the curator with an overview of the gene names and the related noun phrases in the paper but also to support focused extraction of information, e.g. when the curator is looking for a gene name in a specific section or tries to locate a noun phrase referring to a certain gene product.

Clicking on a node in either PaperView or EntitiesView redirects the text window to the paragraph that contains the corresponding gene name or noun phrase, which is now highlighted in a different colour. The same colour is used to highlight the other noun phrases listed together with the clicked node in EntitiesView. In this way the selected node and all related noun phrases become more visible in the text.

The interface allows the curators to mark a text segment as "read" by crossing it out (which is useful when they want to distinguish between the text they have read and what they still need to curate). A "find" function supporting case sensitive and wrapped search is implemented too.

The "Tokens to verify" tab is used to collect feedback about the gene name recogniser in a non-intrusive manner. This tab presents the curator with a short list of words (currently just 10 per paper) for which the recogniser is uncertain whether they are gene names or not. Each name in the list is hyperlinked to the text allowing the curator to examine it in its context and decide whether it should be marked as a gene or not (by clicking on the corresponding button). Active learning [6] is then used to improve the recogniser's performance on the basis of the collected data.

[b]PaperBrowser is a "rich content" browser built on top of the Mozilla Gecko engine and JREX (see www.mozilla.org for more details).

Pacific Symposium on Biocomputing 12:245-256(2007)

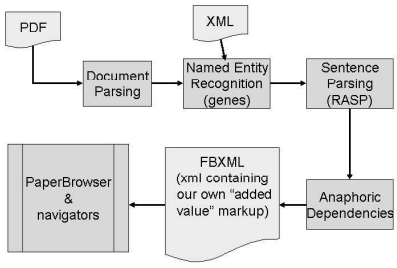


Figure 2. Paper processing pipeline

4. Paper Processing Pipeline

In this section we discuss the technology used to produce the XML-based format which is displayed by PaperBrowser. This a non-trivial task requiring the integration of several components, each addressing different but often inter-related problems, into a unified system. The pipeline in Figure 2 was implemented since it was unclear whether integrating these modules could be readily done within an existing platform such as GATE [7].

The input to the pipeline is the paper in PDF, which is currently the only "standard electronic format" in which all relevant papers are available. This needs to be translated to a format that can be utilised by the deployed NLP modules but since current PDF-to-text processors are not aware of the typesetting of each journal, text in two columns, footnotes, headers and figure captions tends to be dispersed and mixed up during the conversion. This problem is addressed by the Document Parsing module which is based on existing software for optical character recognition (OCR) enhanced by templates for deriving the structure of the document [8]. Its output is in a general XML format defined to represent scientific papers. By contrast to standard PDF-to-text processors, the module preserves significant formating information such as characters in italics and superscripts that may indicate the mention of a gene or an allele respectively.

The initial XML is then fed to a module that implements a machine-learning paradigm extending the approach in [9] to identify gene names in the text [10], a task known as Named Entity Recognition (NER).[c] Then, the RASP parser [11] is employed to identify the boundaries of the noun phrase (NP) around each gene name and its grammatical relations with other NPs in the text. This information is combined with features derived

[c]The NER module may also be fed with papers in XML available from certain publishers.

Pacific Symposium on Biocomputing 12:245-256(2007)

Table 1. Performance of the modules for Document Parsing, Named Entity Recognition and Anaphora Resolution.

	Recall	Precision	F-score
Named Entity Recognition	82.2%	83.4%	82.8%
Anaphora resolution	75.6%	77.5%	76.5%
Document Parsing	96.2%	97.5%	96.8%

from an ontology to resolve the anaphoric dependencies between NPs [12]. For instance, in the following excerpt:

> *... is encoded by the gene male specific lethal-1 ... the MSL-1 protein localizes to several sites ... male animals die when they are mutant for msl-1 ...*

the NER system recognises "male specific lethal-1" as a gene-name. Additionally, the anaphora resolution module identifies the NP "the gene male specific lethal-1" as referring to the same entity as the NP "msl-1" and as being related to the NP "the MSL-1 protein".

A version of the paper in FBXML (i.e. our customised XML format) is the result of the whole process that is displayed by PaperBrowser. The PaperView navigation pane makes use of the output of the NER system and information about the structure of the paper, while EntitiesView utilises the output of the anaphora resolution module as well.

Images, which are very hard to handle by most text processing systems [2] but are particularly important to curators (see next section), are displayed in an extra window (together with their captions which are displayed in the text too) since trying to incorporate them into the running text was too complex given the information preserved in the OCR output.

Following the standard evaluation methodology in NLP, we used collections of texts annotated by domain experts to assess the performance of the NER [10] and the anaphora resolution [12] modules in terms of Recall (correct system responses divided by all human-annotated responses), Precision (correct system responses divided by all system responses) and their harmonic mean (F-score). Both modules achieve state-of-the-art results compared to semi-supervised approaches with similar architectures.[d] The same measures were used to evaluate the document parsing module on an appropriately annotated corpus [8]. Table 1 summarises the results of these evaluations.

[d]Earlier versions of the NER and anaphora resolution modules are discussed in [13].

Pacific Symposium on Biocomputing 12:245-256(2007)

5. ProformaEditor

In order to further support the curation process, we implemented an editing tool called ProformaEditor (Figure 4D). ProformaEditor supports all general and customised functionalities of the editor that it is meant to replace such as: (a) copying text between fields and from/to other applications such as PaperBrowser, (b) finding and replacing text (enabling case-sensitive search and a replace-all option), (c) inserting an empty proforma, the fields of which can then be completed by the curator, and (d) introducing predefined text (corresponding to FlyBase's controlled vocabulary) to certain fields by choosing from the "ShortCuts" menu.

Additionally, ProformaEditor visualises the structure of the record as a tree enabling the curator to navigate to a proforma by clicking on the corresponding node. Moreover, the fields of subsequent proformae are displayed in different colours to be distinguished more easily.

Since the curators do not store pointers to a passage that supports a field entry, finding evidence for that entry in the paper based on what has been recorded in the field is extremely difficult [2]. We address this problem by logging the curator's pasting actions to collect information which will enable us to further enhance the underlying NLP technology such as: (a) where the pasted text is located in the paper, (b) which field it is pasted to, (c) whether it contains words recognised as gene names or related NPs, and (d) to what extent it is subsequently post-edited by the curator. This data collection also takes place without interfering with curation.

6. Integrating the tools into FlyBase's workflow

After some in-house testing, a curator was asked to produce records for 12 papers from two journals using a prototype version of the tools to which she was exposed for the first time (Curation01). Curation01 initiated our attempt to integrate the tools into FlyBase's workflow. This integration requires substantial effort and often needs to address low-level software engineering issues [14]. Thus, our aims were quite modest: (a) recording potential usability problems and (b) ensuring that the tools do not impede the curator from completing a record in the way that she had been used to.

ProformaEditor was judged to be valuable although a few enhancements were identified such as the introduction of the "find and replace" function and the "ShortCuts" menu that the curators had in their old editor. Compared to that editor, the curator regarded the visualisation of the record structure as a very useful additional feature.

PaperBrowser was tested less extensively during Curation01 due to the

Pacific Symposium on Biocomputing 12:245-256(2007)

loss of the images during the PDF-to-XML process which was felt by the curator to be a significant impediment. Although the focus of the project is on text processing, the pipeline and PaperBrowser were adjusted accordingly to display this information.

A second curation exercise (Curation02) followed, in which the same curator produced records for 9 additional papers using the revised tools. This time the curator was asked to base the curation entirely on the text as displayed in the PaperBrowser and advise the developers of any problems. Soon after Curation02, the curator also produced records for 28 other papers from several journals (Curation03) using ProformaEditor but not PaperBrowser since these papers had not been processed by the pipeline.

Like every other record produced by FlyBase curators, the outputs of all three exercises were successfully post-processed and used to populate the database. Overall, the curator did not consider that the tools have a negative impact on task completion. ProformaEditor became the curator's editor of choice after Curation03 and has been used almost daily since then. The feedback on PaperBrowser included several cases in which identifying passages that provide information about certain genes as well as their variants, products and phenotypes using PaperView and/or EntitiesView was considered to be more helpful than looking at the PDF viewer or a printout.

Since the prototype tools were found to be deployable within FlyBase's workflow, we concluded that the aims of this phase had been met. However, the development effort has not been completed since the curator also noticed that the displayed text carries over errors made by the pipeline modules and pointed out a number of usability problems on the basis of which a list of prioritised enhancements was compiled.

The shortlisted improvements of PaperBrowser include: (a) making tables and captions more easily identifiable, (b) flagging clicked nodes in the navigation panes, and (c) saving text marked-as-read before exiting. We also intend to boost the performance of the pipeline modules using the curator's feedback and equip ProformaEditor with new pasting functionalities which will incorporate FlyBase's term normalisation conventions.

7. A pilot study on usability

This section presents an initial attempt to estimate the curator's performance in each exercise. To the best of our knowledge, although preliminary, this is the first study of this kind relating to scientific article curation.

Although the standard NLP metrics in Table 1 do not capture how useful a system actually is in the workplace [1], coming up with a quantitative

Pacific Symposium on Biocomputing 12:245-256(2007)

measure to assess the curator's performance is not straightforward either. At this stage we decided to use a gross measure by logging the time it took for the curator to complete a record during each curation exercise. This time was divided by the number of proformae in each record to produce an estimate of "curation time per proforma".

The data were analysed following the procedure in [15]. Two outliers were identified during the initial exploration of the data and excluded from subsequent analysis.[e] The average time per proforma for each curation exercise using the remaining datapoints is shown in Figure 3A.

A one-way ANOVA returned a relatively low probability ($F(2,44)=2.350$, $p=0.107$) and was followed by planned pairwise comparisons between the conditions using the independent-samples two-tailed t-test. Curation01 took approximately 3 minutes and 30 seconds longer than Curation02, which suggests that revising the tools increased the curator's efficiency. This difference is marginally significant ($t(44)=2.151$, $p=0.037$) providing preliminary evidence in favour of this hypothesis.

Comparing Curation03 with the other conditions suggests that the tools do not impede the curator's performance. In fact, Curation01 took on average about 2 minutes longer than Curation03 (the main difference between them being the use of the revised ProformaEditor during Curation03). The planned comparison shows a trend towards improving curation efficiency with the later version of the tool ($t(44)=1.442$, $p=0.156$) although it does not provide conclusive evidence in favour of this hypothesis.

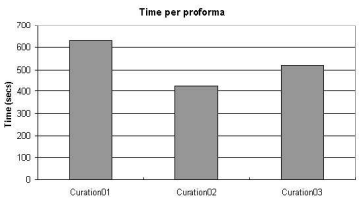
The main difference between Curation02 and Curation03 is viewing the paper exclusively on PaperBrowser in Curation02 (as opposed to no use of this tool at all in Curation03).[f] Completing a proforma using PaperBrowser is on average more than one minute and thirty seconds faster. Although the planned comparison shows that the difference is not significant ($t(44)=1.1712$, $p=0.248$), this result again indicates that the tool does not have a negative impact on curation.

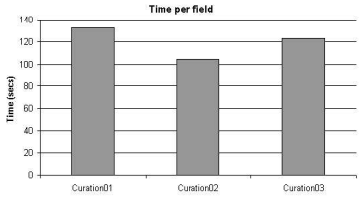
Additional analysis using a more fine-grained estimate of "curation time per completed field" (computed by dividing the total time per record

[e]The first outlier corresponds to the first record ever produced by the curator. This happened while a member of the development team was assisting her with the use of the tools and recording her comments (which arguably delayed the curation process significantly). The logfile for the second outlier which was part of Curation03 included long periods during which the curator did not interact with ProformaEditor.

[f]The version of ProformaEditor was the same in both cases but the curator was more familiar with it during Curation03.

Pacific Symposium on Biocomputing 12:245-256(2007)





	(A) Time per proforma		(B) Time per completed field		
	Average	St. dev.	Average	St. dev.	papers
Curation01	631.64s (10m 32s)	192.21s	132.90s (2m 13s)	33.50s	11
Curation02	424.21s (7m 04s)	157.04s	104.67s (1m 45s)	41.47s	9
Curation03	520.95s (8m 41s)	236.91s	123.20s (2m 03s)	52.35s	27

Figure 3. Results of pilot study on usability.

by the number of completed fields) showed the same trends (Figure 3B). However, the ANOVA suggested that the differences were not significant ($F(2,44)=0.925$, $p=0.404$), which is probably due to ignoring the time spent on non-editing actions by this measure.

Overall, this preliminary study provides some evidence that the current versions of ProformaEditor and PaperBrowser are more helpful than the initial prototypes and do not impede curation. These results concur with the curator's informal feedback. They also meet our main aim at this stage which was to integrate the tools within an the existing curation workflow.

Clearly, more detailed and better controlled studies are necessary to assess the potential usefulness of the tools building on the encouraging trends revealed in this pilot. Devising these studies is part of our ongoing work, aiming to collect data from more than one curator. Similarly to the pilot, we will attempt to compare different versions of the tools which will be developed to address the compiled shortlist of usability issues. We are also interested in measuring variables other than efficiency such as accuracy and agreement between curators.

In our other work, we are currently exploiting the curator's feedback for the active learning experiments. We also intend to analyse the data collected in the logstore in order to build associations between proforma fields and larger text spans, aiming to be able to automatically identify and highlight such passages in subsequent versions of PaperBrowser.

Acknowledgments

This work takes place within the BBSRC-funded Flyslip project (grant No 38688).

We are grateful to Florian Wolf and Chihiro Yamada for their insights and contributions in earlier stages of the project. PaperBrowser and ProformaEditor are implemented in Java and will be available through the project's webpage at: www.cl.cam.ac.uk/users/av308/Project_Index/index.html

References

1. A. M. Cohen and W. R. Hersh. A survey of current work in biomedical text mining. *Briefings in Bioinformatics* **6**(1):57-71 (2005).
2. A. S. Yeh, L. Hirschman and A. A. Morgan (2003), Evaluation of text data mining for database curation: lessons learned from the KDD Challenge Cup. *Bioinformatics* **19** (suppl. 1): i331-i339.
3. J. Preece, Y. Rogers and H. Sharp. *Interaction design: beyond human-computer interaction*. John Wiley and Sons (2002).
4. L. Tanabe, U. Scherf, L. H. Smith, J. K. Lee, L. Hunter and J. N. Weinstein. MedMiner: an internet text-mining tool for biomedical information with application to gene expression profiling. *BioTechniques* 27(6):1210-1217 (1999).
5. H. M. Mueller, E. E. Kenny and P. W. Sternberg. Textpresso: An ontology-based information retrieval and extraction system for biological literature. *PLoS Biology* **2**(11):e309 (2004).
6. D. A. Cohn, Z. Ghahramani and M. I. Jordan. Active learning with statistical models. In G. Tesauro, D. Touretzky and J. Alspector (eds), *Advances in Neural Information Processing*, vol. 7, 707-712 (1995).
7. H. Cunningham, D. Maynard, K. Bontcheva and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. *Proceedings of ACL 2002*, 168-175 (2002).
8. B. Hollingsworth, I. Lewin and D. Tidhar. Retrieving hierarchical text structure from typeset scientific articles: A prerequisite for e-Science text mining. *Proceedings of the 4th UK e-science all hands meeting*, 267-273 (2005).
9. A. A. Morgan, L. Hirschman, M. Colosimo, A. S. Yeh and J. B. Colombe. Gene name identification and normalization using a model organism database. *J. of Biomedical Informatics* **37**(6):396-410 (2004).
10. A. Vlachos and C. Gasperin. Bootstrapping and evaluating NER in the biomedical domain. *Proceedings of BioNLP 2006*, 138-145 (2006).
11. E. Briscoe, J. Carroll and R. Watson. 'The second release of the RASP system', *Proceedings of ACL-COLING 2006*, 77-80 (2006).
12. C. Gasperin. Semi-supervised anaphora resolution in biomedical texts. *Proceedings of BioNLP 2006*, 96-103 (2006).
13. A. Vlachos, C. Gasperin, I. Lewin and E. J. Briscoe. Bootstrapping the recognition and anaphoric linking of named entities in Drosophila articles. *Proceedings of PSB 2006*, 100-111 (2006).
14. C. Barclay, S. Boisen, C. Hyde and R. Weischedel. The Hookah information extraction system. *Proceedings of Workshop on TIPSTER II*, 79-82 (1996).
15. D. S. Moore and G. S. McCabe. *Introduction to the practice of statistics*, 713-747. Freeman and Co (1989).

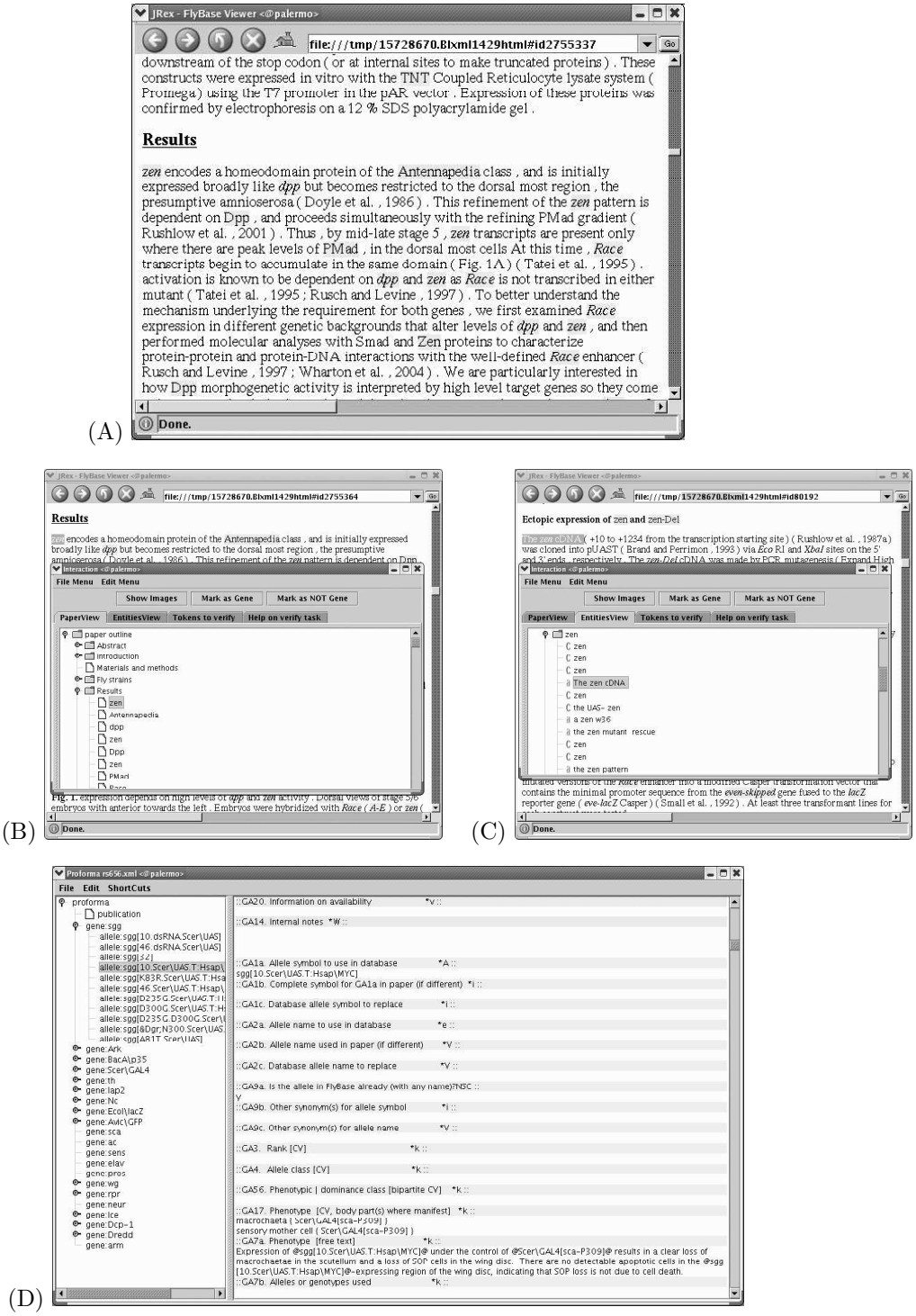


Figure 4. (A) Automatically recognised gene-names highlighted in PaperBrowser. Navigating through the paper using: (B) PaperView and (C) EntitiesView. (D) Editing a record with ProformaEditor.