



SOFTWARE ARTICLE

Open Access

Caipirini: using gene sets to rank literature

Theodoros G Soldatos^{1,6}, Seán I O'Donoghue^{1,7,8*}, Venkata P Satagopam¹, Adriano Barbosa-Silva^{1,2,3}, Georgios A Pavlopoulos^{1,5}, Ana Carolina Wanderley-Nogueira⁴, Nina Mota Soares-Cavalcanti⁴ and Reinhard Schneider^{1,9}

* Correspondence:

sean@odonoghuelab.org

¹Structural and Computational Biology Unit, European Molecular Biology Laboratory (EMBL), Heidelberg, Germany

Full list of author information is available at the end of the article

Abstract

Background: Keeping up-to-date with bioscience literature is becoming increasingly challenging. Several recent methods help meet this challenge by allowing literature search to be launched based on lists of abstracts that the user judges to be 'interesting'. Some methods go further by allowing the user to provide a second input set of 'uninteresting' abstracts; these two input sets are then used to search and rank literature by relevance. In this work we present the service 'Caipirini' (<http://caipirini.org>) that also allows two input sets, but takes the novel approach of allowing ranking of literature based on one or more sets of genes.

Results: To evaluate the usefulness of Caipirini, we used two test cases, one related to the human cell cycle, and a second related to disease defense mechanisms in *Arabidopsis thaliana*. In both cases, the new method achieved high precision in finding literature related to the biological mechanisms underlying the input data sets.

Conclusions: To our knowledge Caipirini is the first service enabling literature search directly based on biological relevance to gene sets; thus, Caipirini gives the research community a new way to unlock hidden knowledge from gene sets derived via high-throughput experiments.

Background

Keeping up-to-date with bioscience literature is becoming more challenging as the number of new papers appearing daily - currently over 2,000 - continues to increase. As a result, there is an increasing need for methods that can efficiently search this literature [1], and to this end a wide range of tools and services are now available [2,3]. Currently, most tools used for retrieving bioscience literature are based on keyword searches, although such approaches have limitations: firstly, it can be difficult for a researcher to find a set of keywords that exactly specify the biological functions she or he may be interested in; secondly, the ranking of results is usually not based on relevance to the biological functions of interest. Several recent methods have been proposed to address these limitations, e.g., ETBLAST [4] can launch literature searches based on a single text document such as an abstract; such methods allow searches to be defined implicitly, e.g., based on a text of interest, rather than having to explicitly define keywords. Several tools have extended this approach, allowing collections of abstracts as input, e.g., PubFinder [5] and MScanner [6].

A common problem with all literature search methods is that only a fraction of the literature retrieved is truly of interest or relevance for the end-user. Recently, a new tool, MedlineRanker [7], partly addresses this problem by allowing the end-user to define two input lists of abstracts - typically, one input list ('A') can be used for 'interesting' abstracts, and the second input list ('B') for abstracts that are 'not-interesting'. MedlineRanker then uses these two input sets to rank a third 'query' set of abstracts in order of interest, based on similarity to input sets A and B. A significant advantage of this approach is that any retrieved literature that is judged to be uninteresting can be added subsequently to the 'uninteresting' input set, and MedlineRanker can be re-run to iteratively improve the relevance of the results.

The work presented in this paper was motivated by our collaboration with a group of experimentalists interested in ranking literature corpora based on similarity to sets of genes known to be associated with specific phenotypes or conditions. For example, we were interested in ranking literature based on a set of genes known to be associated with disease resistance in the plant model organism *Arabidopsis thaliana*. In a second case, we wanted to find literature related to the S-phase of the human cell cycle, using as input one set of genes known to be involved in S-phase, and a second set of 'background' genes known to be involved with all other cell cycle phases, but not S-phase.

In searching for a tool that would suit our purposes, we surveyed existing tools for ranking and classifying biomedical literature. Many of the existing tools are based on the 'Support Vector Machine' (SVM) approach [8]; however most of these are 'hard-wired' to a particular topic, e.g., alternative transcription [9], human genetic associations [10], or clinical studies [11]. Of the generic SVM tools that can be trained with any topic of interest (e.g., BibGlimpse [12]), none allow literature ranking based on gene lists, although one does precisely the opposite, namely rank genes based on literature [13]. Of the generic non-SVM tools, some can accept lists of genes as input and produce a ranked list of literature as output (e.g., Kleio [14]). However, for our purposes the existing methods have some key limitations. Firstly, users can only specify what they want - they cannot in addition specify what they do *not* want. We were interested in a method that would allow users to specify a set of literature they find uninteresting, and would use this information to improve the relevance and ranking of subsequent literature searches. A second limitation of existing methods is that they can only find literature that explicitly mentions genes in the input set. We were interested in a method that can go beyond these limitations, learn patterns in the literature associated with the input genes, and potentially return literature that may mention none of the genes in the input lists, but that discusses biological processes and functions associated with the input genes.

To meet these needs, we developed a new service ('Caipirini') that, similar to MedlineRanker, allows two main input sets, but takes the novel approach of allowing one or both input sets to be a set of gene identifiers, hence allowing literature search to be launched based on sets of genes. Caipirini also differs from similar services in that comparison of abstracts is based on keywords taken from a large dictionary of nouns and verbs. In addition, ranking in Caipirini is done using a generic SVM strategy, re-trained for each input provided by the user. Table 1 summarizes key differences between Caipirini and other comparable literature services.

Table 1 Tools for ranking biomedical literature using document sets

Tool	Input		Dictionary	Method
	Sets	Genes		
Kleio [14]	1	Yes	~2 million entities	VSM score
PubFinder [5]	1	No	~100,000 words from Medline	Likelihood
MScanner [6]	1	No	~25,000 MeSH terms	Bayesian
MedlineRanker [7]	1 or 2	No	Detects nouns via syntax	Bayesian
Caipirini	2	Yes	~4 million entities	SVM

Methods

Overview of User Workflow

On Caipirini's home page [15] the user is prompted to provide one list of input identifiers (set A). These identifiers can correspond to Entrez genes [16], Ensembl genes [17], or Medline abstracts [18] - alternatively, the user may provide a PubMed query directly as input. On the advanced version (under 'More Options') the user is able to additionally specify a second input list (set B). Sets A and B are typically used to define an 'interesting' and an 'uninteresting' (or 'background') set. When the user provides only an 'interesting' input set (set A), set B will be automatically filled with the same number of abstracts as in set A, but randomly chosen from PubMed, excluding abstracts within set A (see text below for details on how abstracts are associated with the Set A input when the user specifies genes). The third input set (set C) may contain a PubMed query or a list of PubMed identifiers - the goal of Caipirini is to rank the abstracts matching set C in order of similarity to set A (highest rank) and set B (lowest rank). When the user does not provide a set C, Caipirini will rank all abstracts in Medline (indexed by AKS2), and return those with Caipirini score over 85% (see below for details).

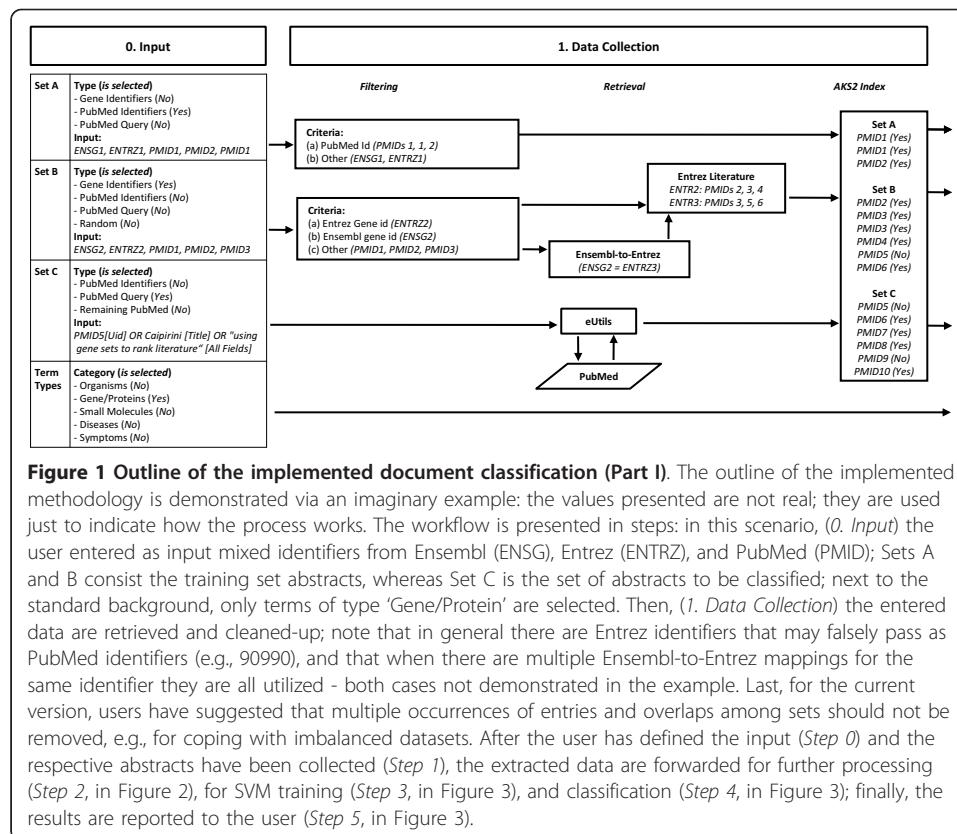
The classification and ranking is based on keywords extracted from abstracts associated with sets A and B. By default, Caipirini uses the following keyword types: diseases, symptoms, small molecules, genes/proteins, organisms, and other 'general biological terms'. The user can choose to deselect some or all of these types, except for the so called 'bio-terms' and 'bio-actions' (i.e., the two groups that comprise the 'general biological terms'), which are always used.

The results comprise mainly of two lists of abstracts from set C: those found with high certainty to be more similar to set A ('Top Set A Results') and those more similar to set B ('Top Set B Results'). Nevertheless, the user has access to Caipirini's ranking scores for all abstracts. A generic outline of Caipirini's workflow is presented in Figures 1, 2 and 3, where the overview is organized in discrete steps around an imaginary example-scenario.

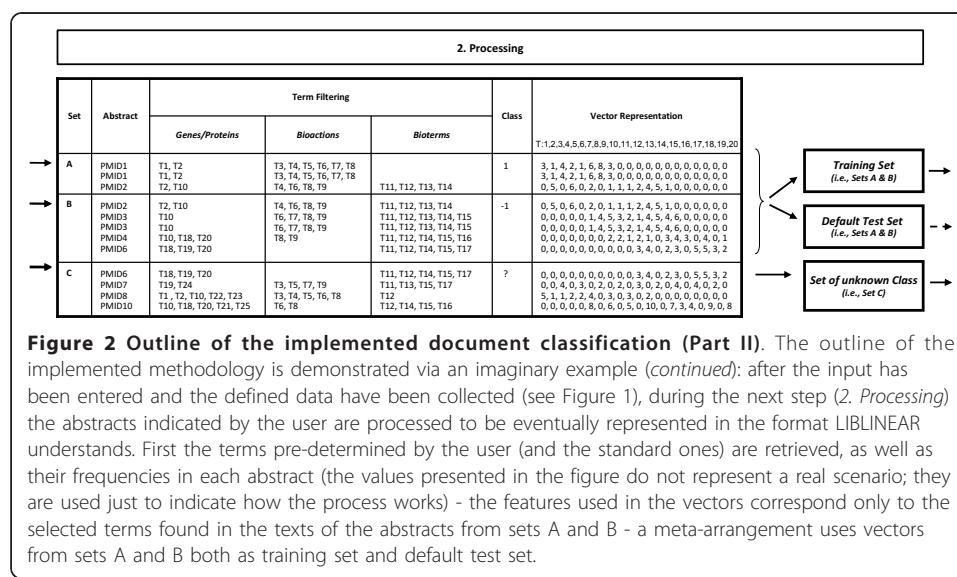
Input data and initial processing

For the initial processing of input sets in Caipirini we re-used parts of the pipeline we developed for Martini [19]. However, after initial processing, the similarity with Martini ends, and Caipirini uses a very different analysis method (SVM) to achieve a very different goal (ranking of literature).

By default Caipirini assumes that sets 'A' and 'B' are lists of Entrez gene identifiers [16], in which case Caipirini retrieves, for each gene, all PubMed identifiers [18] that are referred to in the Entrez entry, including the GeneRIFs and interaction records.



Caipirini also allows Ensembl gene identifiers to be used [17] - in this case, Caipirini first translates them to Entrez gene identifiers, and then to the corresponding PubMed identifiers as above; see Figure 1. The mapping between Ensembl gene, Entrez genes, and Medline abstracts is retrieved offline using SRS [20] and stored in RAM (random access memory) to enable fast access while processing jobs.



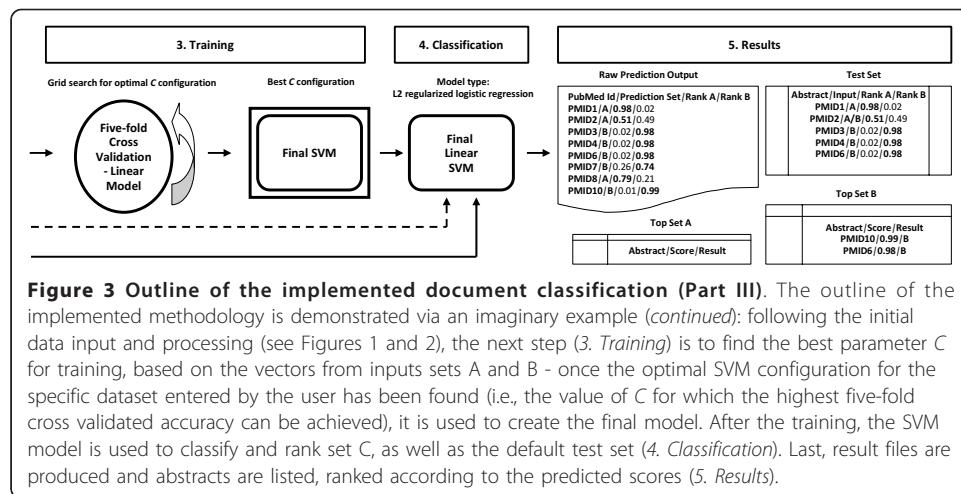


Figure 3 Outline of the implemented document classification (Part III). The outline of the implemented methodology is demonstrated via an imaginary example (*continued*): following the initial data input and processing (see Figures 1 and 2), the next step (3. *Training*) is to find the best parameter C for training, based on the vectors from inputs sets A and B - once the optimal SVM configuration for the specific dataset entered by the user has been found (i.e., the value of C for which the highest five-fold cross validated accuracy can be achieved), it is used to create the final model. After the training, the SVM model is used to classify and rank set C, as well as the default test set (4. *Classification*). Last, result files are produced and abstracts are listed, ranked according to the predicted scores (5. *Results*).

Alternatively, in the input sets 'A' and 'B', the end-user can specify either a PubMed query or a list of PubMed identifiers. For the third input set ('C'), the only formats that can be used are either a PubMed query, or PubMed identifiers. For PubMed queries, Caipirini uses Entrez Programming Utilities [21] to retrieve matching PubMed identifiers; see Figure 1. When only set 'A' is provided, a second list of PubMed identifiers (equal in size to the number of abstracts extracted from set 'A', but different from those) are randomly selected to populate set B, whereas set 'C' is considered to be all PubMed abstracts in Caipirini indexed with keywords (including abstracts from sets A and B).

The next step is to convert each PubMed identifier into a list of keywords. For this, we used an instance of AKS2, a literature analysis tool that was the engine behind the biomedical search service Novoseek. AKS2 is based on a keyword dictionary of ~3.7 million entries covering ~1.8 million genes and proteins, ~1.6 million chemicals, ~30,000 diseases, ~7,000 drugs, ~5,000 bio-actions, ~2,000 symptoms, and ~120,000 other biologically-relevant keywords. The size and breadth of the AKS2 dictionary compares favorably with similar resources such as Biothesaurus and Unified Medical Language System (UMLS), both of which have only ~2 million entries. However, our decision to use AKS2 was not based on its dictionary but on the convenience of one of its services in which the most recent half of all Medline abstracts have been pre-tagged with its dictionary, resulting in an average of 32 keywords per abstract. In addition, in previous work we found that AKS2 gave good results for the related task of keyword enrichment [19]. In the present work, we used AKS2 to construct a hash table in RAM in which each of the ~10 million Medline abstracts are linked to associated AKS2 keywords. By default, Caipirini uses all keyword types (genes, chemical, diseases, etc.) for subsequent analysis; however the web interface allows the user to exclude some types; see Figures 1 and 2.

Caipirini has a number of technical limitations: first, not all Entrez gene records have associated abstracts; second, some PubMed entries contain only titles, i.e., they have no abstract text; third, since AKS2 only indexed the most recent half of PubMed, older abstracts are not used. Finally, to reduce server load and processing time, we have limited each input field of Caipirini to a maximum number of 25,000 entries, either genes or abstracts. A job with more than this number of entries specified in any input set will not run, and the user will be asked to reduce the corresponding list's size.

Classification of abstracts

Training Set

If the user has specified genes in either set A or B, these genes are first converted to a list of abstracts. Next, Caipirini creates one feature vector for each abstract: the vector has a number of dimensions equal to the total number of unique keywords found in all abstracts of sets A and B, and a length in each dimension equal to the number of occurrences of that keyword in the abstract. The same process is followed to create the vectors for set C with the difference that only terms mentioned in abstracts from the training set are taken into account; see Figure 2.

SVM Training

In the next step, we search for the best penalty factor C that gives the linear SVM model with the highest accuracy for the entered dataset - the SVM library used is LIBLINEAR [22] with the ‘L2-regularized logistic regression (dual)’ solver and with the tolerance termination criterion ϵ set to 0.1 by default. The accuracy is assessed by LIBLINEAR using five-fold cross-validation with the training data; see Figure 3. Following the notation from [23], Caipirini does the C parameter search in two steps. Firstly, it evaluates the accuracy at eight points, for $c = 2^x$, where x in [-3, 4] increased by 1.0 per step. Then, if the training set is small (i.e., smaller than 5,000 vectors), or if it is of moderate size (i.e., training vectors are less than 10,000) and the best of the tested points gives a cross-validated accuracy of less than 80%, Caipirini runs a further grid search in the neighborhood of the best C , see Table 2.

Classification

The best configuration found during SVM training is used with all input data from set A and B to construct the final linear SVM model. The solver used allows assigning a ‘Caipirini score’ to each abstract, corresponding to the probability that the abstract belongs to either set A or B. The vectors from set C are passed into the trained model so as that each abstract is classified to belong to set A if the SVM assigns a probability of belonging to set A greater than 0.5. Vectors are assigned to set B using the opposite criteria. Finally, the abstracts from set C are listed together with their assignments, and ranked according to the Caipirini scores they received; see Figure 3.

Performance

Many other services based on SVM are trained using pre-defined data, hence ‘hard-wired’ to a particular topic, with only the classification done on-the-fly (e.g., GAPscranner [10]). Caipirini differs greatly in that the SVM is re-trained for each new input specified by the user. This approach has the advantage that it is likely to be more

Table 2 Training the SVM: grid search for best penalty parameter c

X	$C = 2^X$	Further neighborhood search (C tested also for)
-3.0	0.125	0.05, 0.1, 0.15
-2.0	0.25	0.15, 0.2, 0.3, 0.35
-1.0	0.5	0.3, 0.4, 0.6, 0.7
0.0	1.0	0.75, 1.25, 1.5
1.0	2.0	1.5, 2.5, 3.0
2.0	4.0	3.0, 3.5, 4.5, 5.0, 6.0
3.0	8.0	6.0, 7.0, 9.0, 10.0, 11.0
4.0	16.0	11.0, 12.0, 13.0, 14.0, 15.0

accurate; however a significant penalty is paid in the time taken to train. The exact time will depend on the size of the input sets, as well as the number of Medline abstracts per gene. As a guide to performance speed, using random sets of 25,000 abstracts in each set A and B takes about two hours to classify all PubMed abstracts indexed in Caipirini; however, running time can be highly variable, e.g., depending on unpredictable factors such as how many abstracts link to each gene, the quality of the input set, the number of keywords retrieved, the training search, and the server load.

The user can assess the reliability of Caipirini's performance in two ways. First, the Caipirini report shows a 'test set accuracy', indicating Caipirini's ability to distinguish or separate the input sets A and B. In addition, Caipirini reports a measure of the 'Cross Validation Accuracy', which is an approximate estimation of the accuracy of classification for set C; however, this estimation is based on inputs A and B, and the true classification accuracy will likely be lower, depending on what the user provides for set C. It would be possible to provide more precise estimations of the classification accuracy and significance using further re-sampling techniques, such as bootstrapping, jackknifing, or other permutation tests. However, such approaches have been developed primarily for quantitative analysis of experiment data, whereas Caipirini is more qualitative, designed to suggest which papers are interesting. Furthermore, such additional methods can be computationally expensive (e.g., [24]) and we chose not to use them, since each Caipirini analysis is already quite slow, primarily due to our use of five-fold cross-validation.

Datasets

Cell-cycle datasets

To test Caipirini we used a dataset in which 158 human gene identifiers were associated with S-phase of the cell cycle, and 456 with the other three phases [19,25]. To estimate precision and accuracy, we constructed three sets of Medline abstracts using the queries specified in Table 3. The first set ('S-phase') contains abstracts that have been pre-assigned via MeSH terms [26] to S-phase specifically, and not to any of the other cell-cycle phases. This set was used to calculate true positives and false negatives. The second set ('Not S-phase') consists of Medline abstracts that have been assigned to any of the other cell cycle phases, but not to S-phase. This set was used to calculate false positives and true negatives. The final set ('Unknown phase') consists of abstracts

Table 3 PubMed queries used to calculate precision and recall

Dataset	PubMed Query	Abstracts
S-phase	"humans"[MeSH Terms] AND ("S Phase"[MeSH Terms] OR "DNA Replication"[MeSH Terms]) NOT ("G1 Phase"[MeSH Terms] OR "G2 Phase"[MeSH Terms] OR "Prophase"[MeSH Terms] OR "Prometaphase"[MeSH Terms] OR "Metaphase"[MeSH Terms] OR "Anaphase"[MeSH Terms] OR "Telophase"[MeSH Terms] OR "Cytokinesis"[MeSH Terms]) AND ("2000/01/01"[PDAT]: "2008/06/31"[PDAT])	4,240
Not S-phase	"humans"[MeSH Terms] AND ("G1 Phase"[MeSH Terms] OR "G2 Phase"[MeSH Terms] OR "Prophase"[MeSH Terms] OR "Prometaphase"[MeSH Terms] OR "Metaphase"[MeSH Terms] OR "Anaphase"[MeSH Terms] OR "Telophase"[MeSH Terms] OR "Cytokinesis"[MeSH Terms]) NOT ("S Phase"[MeSH Terms] OR "DNA Replication"[MeSH Terms]) AND ("2000/01/01"[PDAT]: "2008/06/31"[PDAT])	4,329
Unknown phase	"Cell cycle" AND "humans"[MeSH Terms] NOT ("G1 Phase"[MeSH Terms] OR "G2 Phase"[MeSH Terms] OR "Prophase"[MeSH Terms] OR "Prometaphase"[MeSH Terms] OR "Metaphase"[MeSH Terms] OR "Anaphase"[MeSH Terms] OR "Telophase"[MeSH Terms] OR "Cytokinesis"[MeSH Terms] OR "S Phase"[MeSH Terms] OR "DNA Replication"[MeSH Terms]) AND ("2008/01/01"[PDAT]: "2008/06/31"[PDAT])	2,989

that were related to the human cell cycle, but have not been classified by MeSH terms to any specific phase. This set was used to demonstrate the ability of Caipirini to find abstracts related to S-phase that could not be identified using MeSH terms.

Arabidopsis datasets

We manually created a set of 90 abstracts we judged to be related to pathogen resistance in *Arabidopsis*. We also created a second set of 90 abstracts that were also related to *Arabidopsis*, but did not specifically discuss pathogen resistance mechanisms. As a test set, we defined a third set of 216 *Arabidopsis*-related abstracts for which relevance, or not, to pathogen resistance was initially unknown; this was then tested using Caipirini and subsequently verified by three curators; see Additional file 1.

Results

Cell-Cycle Dataset

We evaluated the performance of Caipirini using a dataset of 158 gene identifiers associated with S-phase and 456 gene identifiers associated with the other three phases of the human cell cycle (G1, G2, and M). These gene sets were then used as input to Caipirini to rank the results of a PubMed query that specifies ~4,000 abstracts known to be related to S-phase. The same input gene sets were also used to rank the results of another query specifying ~4,000 abstracts known to be related to the other cell-cycle phases, but not S-phase. Caipirini obtained a moderate recall (i.e., 43%), but had high precision (i.e., 84%); see Figure 4. That is, of all the abstracts predicted by Caipirini to be related to S-phase, 84% were correctly assigned, and Caipirini found 43% of all the abstracts known to be S-phase related. To account for the fact that some abstracts belonged both to the training and to the classified set (which can be a significant issue

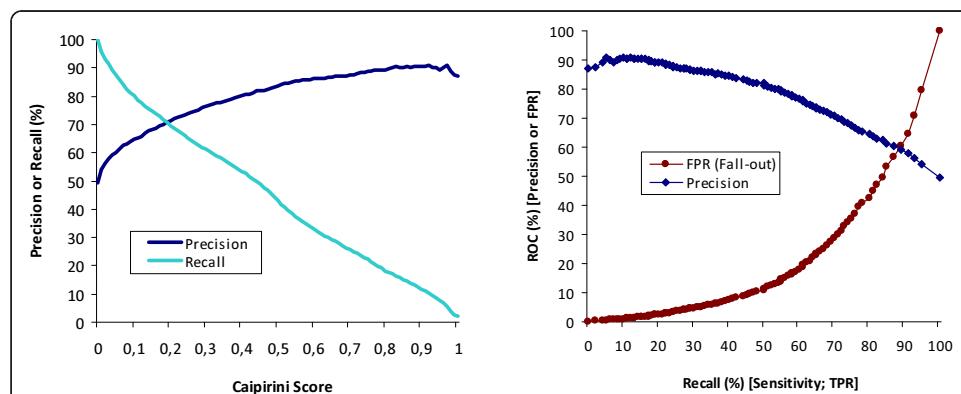


Figure 4 Precision and recall values vs. Caipirini score for cell cycle dataset. For each abstract matching the query in input set C, Caipirini calculates a score, where a value of > 0.5 means the abstract is more likely similar to set A than set B (see Methods for details). The graph on the left shows the relationship between precision, recall, and Caipirini score for the benchmark dataset, where set A are genes related to S-phase of the cell-cycle and set B are cell-cycle genes not related to S-phase. These sets were used to rank a set of abstracts already known to be related either to S-phase or to all other cell-cycle phases. With these data Caipirini achieved a moderate recall value (43% at the default Caipirini score of 0.5), indicating that just less than half of all abstracts truly related to set A were retrieved. However Caipirini achieved a good precision: 84% of abstracts with a score of > 0.5 are true positives, and higher scores give increasingly better precision. The graph on the right shows the performance of Caipirini via the relationship between precision, recall, and false positive rate for the same dataset when the threshold is set at different Caipirini scores. Note that while Caipirini's threshold increases from left to right in the Precision vs. Recall representation (left), in the ROC representations (right) the opposite is implied and Caipirini's score increases with direction from right to left.

because the classifier gets to ‘see’ documents of known class and this in turn can influence performance assessment), we also calculated precision and recall after having removed the overlap (i.e., after excluding from set C abstracts from the training set): there were only minor differences observed; the recall reduced to 41% and the precision remained the same, i.e., 84%.

Another notable characteristic feature of the dataset is the 50% precision at 100% recall (see Figure 4), which implies a perfectly balanced test data set. Indeed, the two queries result in abstract lists of comparable size (see Table 3). However, such balance is unrealistic in information retrieval where ‘uninteresting’ is far more prevalent than ‘interesting’ and we used the same input gene sets to classify the results of a final query that specified a set of ~2,200 abstracts known to be related to the cell-cycle, but where the exact cell-cycle phase was not annotated with MeSH terms. For this set, we manually checked the top 20 abstracts judged by Caipirini to be most likely related to the S-phase. We found that four of these abstracts explicitly mentioned terms related to the S-phase (e.g., ‘S-phase’, ‘DNA replication’, or ‘DNA repair’). The remaining sixteen abstracts did not explicitly mention processes associated with the S-phase, but they mentioned proteins and genes known to be related to the S-phase (e.g., VCP, p21, p16, Sp1, E2F, MLH1, and BRCA1) - several of them were included in the input set A.

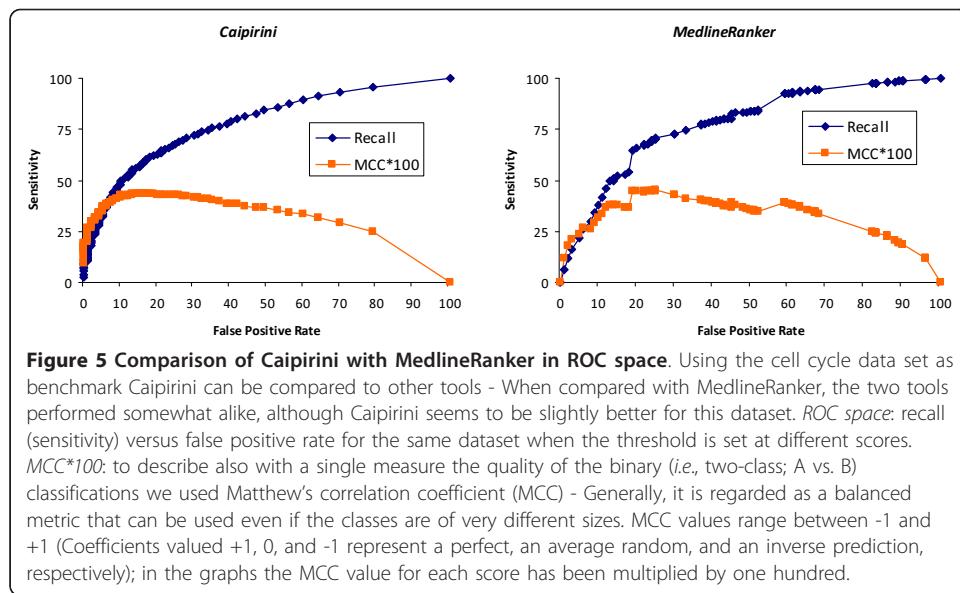
Unlike the other two cell-cycle queries, where the correct assessment of each abstract could be determined automatically, for the third case we did not assess all abstracts since this would have required manually checking the thousands of abstracts predicted to be interesting or uninteresting, which would be quite arduous while producing results of little significance. Thus, for this dataset we did not calculate true precision or recall scores. However, this scenario (as used for the third query) is more similar to how life scientists are likely to use Caipirini, i.e., many will use queries for set C matching large numbers of abstracts, while only a small fraction of the top ranked abstracts will usually be of interest.

The three cell cycle jobs lasted 43, 45 and 39 minutes, respectively.

Benchmarking

Clearly similar in style, Caipirini and MedlineRanker serve similar purposes but in very distinct ways. A first noticeable difference is that MedlineRanker does not facilitate expanding a list of gene identifiers into a list of linked PubMed identifiers. Although this may be considered trivial by some, Caipirini takes this burden off the user’s shoulders. Also, a user does not have to search for multiple synonyms of genes, or to disambiguate. To compare the performance of the two tools, we extracted from Caipirini the abstracts associated to each list of genes from the cell cycle dataset and used them as input to MedlineRanker (default settings applied) in order to rank the abstracts retrieved by the first two queries.

Regarding running time, MedlineRanker was a lot faster, performing the task in only seven seconds. We also compared the accuracy of these two tools using the cell cycle dataset, and found that both performed fairly similarly (e.g., for recall 41% MedlineRanker achieves a slightly lower precision of 81%), although Caipirini seems to be slightly more robust (Figure 5). Thus, since MedlineRanker is much faster with similar performance, it is probably preferable in many cases, except where the user wants to provide gene lists as input.



Arabidopsis Dataset

In addition to gene sets, Caipirini also allows sets of abstracts to be used as input. As part of an ongoing project related to disease resistance in plants [27], we tested Caipirini using sets of abstracts related to the model organism *Arabidopsis thaliana*. In this case, we used 90 abstracts known to be related to pathogen defense mechanisms in *Arabidopsis* (set A) and a further 90 abstracts also related to *Arabidopsis* but that do not discuss pathogen defense mechanisms (set B). We used Caipirini to classify 216 *Arabidopsis* abstracts (set C) that were afterwards independently checked by three of us (ABS, ACWN, and NMSC) and manually assigned as either related to pathogen defense mechanisms, or not (see Additional file 1). The criteria used included explicit mention of either disease-resistance proteins [28], of signaling hormones or other chemicals involved in disease resistance processes [29], or of processes or pathways involved in the hypersensitive response in plants [30].

Of the 216 abstracts, 122 were assigned to set A and 30 to set B by all curators, whereas for 64 there was disagreement. This spread of results underscores the importance of using several independent annotators in assessing the accuracy of a method: although the criteria for interesting vs. uninteresting abstracts may be relatively clear (as specified above), the interpretation of these criteria can vary considerably from annotator to annotator. Unlike the cell-cycle data, where the correct assessment of each abstract could be determined automatically, for the *Arabidopsis* data the 'correct' classification was considered to be that defined by the majority vote of the three independent assignments (*i.e.*, 166 for set A and 50 for set B).

When using both sets A and B as input, Caipirini achieved 87% precision (*i.e.*, found 188 correct assignments). Because set C is decomposed in two classes of very different sizes (dominated by set A-like abstracts), we also calculated the Matthew's correlation coefficient and found that Caipirini achieved a good prediction (with correlation equal to 0.6). Of the 24 false positives obtained, fourteen were already ambiguously categorized by manual examination (*i.e.*, one out of the three annotators disagreed; but not the same curator every time), and of the four false negatives, two were also 'ambiguous'

(i.e., one of the three annotators disagreed; the same one in both cases). When the curators later focused on the unambiguously annotated as false positive abstracts, they noted that some could be indeed considered as related to resistance, but ‘indirectly’ (Additional file 2).

In addition, we ran ten different jobs using the same set C but with only the 90 ‘interesting’ *Arabidopsis* abstracts for set A, and using the random option for set B. These ten runs gave poorer performance than above, with almost all set C abstracts classified as related to set A. This illustrates the benefit of explicitly defining a background set (set B), as it can give more focused results, in this case returning a set of abstracts that almost all relate specifically to pathogen defense mechanisms in *Arabidopsis*.

Regarding speed, all eleven jobs lasted between 8 and 12 minutes each.

Both the *Arabidopsis* and cell-cycle datasets used above are available at the Caipirini web site as examples [31].

Discussion

One of our key goals in creating Caipirini was to address the needs of experimentalists interested in using gene lists to guide literature exploration, and to find biologically relevant abstracts even if these do not explicitly mention the input genes. To our knowledge, only Caipirini and MedlineRanker enable this functionality, and Caipirini is the only method that allows the input sets to be defined using lists of genes. For example, a scientist may want to rank literature based on the difference between a set of genes associated to a primary cancer versus those associated with the metastatic form of the same cancer.

Another related goal of Caipirini - not facilitated otherwise by other tools in this way - was to allow the exploration of distinct and disjoint literature sets, gaining insight into the similar concepts they may share. For example, it would be possible to use Caipirini with the *Arabidopsis* dataset to find abstracts that discuss resistance mechanisms in other plants (see Additional file 2), effectively using knowledge from one organism to learn more about equivalent functions in similar ones. Caipirini supports such usages, particularly via the advanced options, which allow different types of keywords to be enabled or disabled - we believe that the combination of features provided here is currently not available in any other service.

In principle, methods such as Caipirini that use two input sets in this way should be capable of high precision, i.e., we would expect Caipirini to be able to find sets of abstracts with high likelihood to be related to underlying phenomena of interest. This is indeed what we see for the cell-cycle benchmark presented in Figure 4, with a precision of 84% at a Caipirini score of 0.5, and with higher Caipirini scores giving progressively better precision and fewer false positives. Thus for the biologist interested in using gene sets to search literature, our results suggest that many of the abstracts found by Caipirini are likely to correctly reflect the underlying biological difference between the two gene sets, and hence to be of interest.

Nevertheless, the performance (i.e., precision and recall) can vary greatly with different input datasets, and using Caipirini for a task may find only a moderate fraction of all relevant literature, as in the case of the cell cycle dataset. On the other hand, as shown by the *Arabidopsis* example, even in cases where the input data consists of only

a small number of interesting and uninteresting cases, it can still be possible to obtain good results provided that the input is carefully selected to reflect the focused question that is asked.

Furthermore, using two input sets offers an interesting possibility to iteratively improve performance, e.g., by adding falsely classified abstracts to the ‘uninteresting’ input set, and re-running the classification. Such an interactive approach is currently limited slightly by the fact that Caipirini does not yet allow gene and abstract identifiers to be added to the same input set. In the near future we plan to enable such mixed input sets, and hence to further improve the ability to iteratively increase performance.

Clearly, a key limitation of Caipirini currently is its slow performance, especially for large input sets containing many well-studied genes (i.e., genes that are linked to many Medline abstracts). This is the cost paid for allowing users to train SVMs matching their particular input data every time. In the near future, we plan several changes that may improve the speed significantly. Other extensions include optional features, such as the users choosing whether they wish to be notified via e-mail when a task has finished, or whether they want statistical significance tests to be performed additionally. Furthermore, while Caipirini is currently best suited for two-set problems, we also plan to enable users to enter multiple data sets, or to save and re-use previous results in order to be able later to classify new sets of abstracts (e.g., the literature of each new month).

Conclusions

To our knowledge Caipirini is the only service that can search literature directly based on gene sets. Though it can be slow, Caipirini allows some quite complex, new operations for extracting biological insight from gene sets. Therefore, Caipirini gives the research community a new way to unlock hidden knowledge from gene sets derived via high-throughput experiments.

Availability and requirements

The Caipirini service is freely available at <http://caipirini.org>.

Additional material

Additional file 1: Caipirini results for the *Arabidopsis* Dataset. The file contains (a) the annotation of the 216 abstracts of Set C (incl. some comments and notes by the curators), (b) Caipirini results from the ten runs when Set A was compared with random Sets B as background, and (c) Caipirini results for the case when Set A was tested against the manually created Set B used as reference.

Additional file 2: Additional discussion on the *Arabidopsis* Dataset. Detailed explanations from the manual verification of Caipirini’s results for the *Arabidopsis* data-set; ‘PMID’ stands for PubMed identifier.

Acknowledgements and Funding

European Molecular Biology Laboratory, by the European Union Framework Programme 6 grant ‘TAMAHUD’ (LSHC-CT-2007-037472, in part).

Author details

¹Structural and Computational Biology Unit, European Molecular Biology Laboratory (EMBL), Heidelberg, Germany.

²Computational Biology and Data Mining Group, Max-Delbrück Center for Molecular Medicine, Berlin, Germany.

³Bioinformatics Graduate Program, Federal University of Paraná - UFPR (SEPT). Curitiba - PR, Brazil. ⁴Departamento de Genética, Laboratório de Genética e Biotecnologia Vegetal, Centro de Ciências Biológicas, Universidade Federal de Pernambuco, Recife, PE, Brasil. ⁵ESAT-SCD/IBBT-K.U. Leuven Future Health Department, Katholieke Universiteit Leuven,

Leuven, Belgium. ⁶LIFE Biosystems GmbH, Heidelberg, Germany. ⁷Garvan Institute of Medical Research, Sydney, Australia. ⁸Division of Mathematics, Informatics, and Statistics, CSIRO, Sydney, Australia. ⁹Luxembourg Center for Systems Biomedicine, University of Luxembourg, Luxembourg.

Authors' contributions

TGS participated in the concept creation and design of the study, was the principle participant in its implementation, and further contributed to preparing the manuscript and figures. SIOD participated in the design and coordination of the study, interpretation and analysis of the results, and in preparing and proofreading the manuscript. VPS helped in the acquisition of the data, and in the administration of the system. ABS conceived and used the results of the *Arabidopsis* dataset and together with GAP, ACWN and NMSC performed manual assignments of abstracts from results. RS participated in the initial concept of the study.

Competing interests

The authors declare that they have no competing interests.

Received: 12 October 2010 Accepted: 1 February 2012 Published: 1 February 2012

References

1. Altman R, Bergman CM, Blake J, Blaschke C, Cohen A, Gannon F, Grivell L, Hahn U, Hersh W, Hirschman L, et al: **Text mining for biology - the way forward: opinions from leading scientists.** *Genome Biology* 2008, **9**:S7.
2. Rodriguez-Esteban R: **Biomedical text mining and its applications.** *PLoS Comput Biol* 2009, **5**e1000597.
3. Cohen AM, Hersh WR: **A survey of current work in biomedical text mining.** *Brief Bioinform* 2005, **6**:57-71.
4. Lewis J, Ossowski S, Hicks J, Errami M, Garner HR: **Text similarity: an alternative way to search MEDLINE.** *Bioinformatics* 2006, **22**:2298-2304.
5. Goetz T, von der Lieth C-W: **PubFinder: a tool for improving retrieval rate of relevant PubMed abstracts.** *Nucleic Acids Res* 2005, **33**:W774-W778.
6. Poulter GL, Rubin DL, Altman RB, Seoighe C: **MScanner: a classifier for retrieving Medline citations.** *BMC Bioinformatics* 2008, **9**:108.
7. Fontaine JF, Barbosa-Silva A, Schaefer M, Huska MR, Muro EM, Andrade-Navarro MA: **MedlineRanker: flexible ranking of biomedical literature.** *Nucleic Acids Res* 2009, **37**:W141-W146.
8. Polavarapu N, Navathe SB, Ramnarayanan R, ul Haque A, Sahay S, Liu Y: **Investigation into biomedical literature classification using support vector machines.** *IEEE Computational Systems Bioinformatics Conference; Stanford, USA* 2005.
9. Shah PK, Bork P: **LSAT: learning about alternative transcripts in MEDLINE.** *Bioinformatics* 2006, **22**:857-865.
10. Yu W, Clyne M, Dolan SM, Yesupriya A, Wulf A, Liu T, Khouri MJ, Gwinn M: **GAPscrrener: an automatic tool for screening human genetic association literature in PubMed using the support vector machine technique.** *BMC Bioinformatics* 2008, **9**:205.
11. Kilicoglu H, Demner-Fushman D, Rindflesch TC, Wilczynski NL, Haynes RB: **Towards automatic recognition of scientifically rigorous clinical research evidence.** *J Am Med Inform Assoc* 2009, **16**:25-31.
12. Tuchler T, Velez G, Graf A, Kreil DP: **BibGlimpse: the case for a light-weight reprint manager in distributed literature research.** *BMC Bioinformatics* 2008, **9**:406.
13. Yu S, Van Vooren S, Tranchevent LC, De Moor B, Moreau Y: **Comparison of vocabularies, representations and ranking algorithms for gene prioritization by text mining.** *Bioinformatics* 2008, **24**:i119-i125.
14. Nobata C, Cotter P, Okazaki N, Rea B, Sasak Y, Tsuruoka Y, Tsujii Ji, Ananiadou S: **Kleio: A Knowledge-enriched Information Retrieval System for Biology.** *31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; Singapore Association for Computing Machinery*; 2008, 787-788.
15. Caipirini home page. [<http://caipirini.org>].
16. Entrez gene database. [<http://www.ncbi.nlm.nih.gov/sites/entrez?db=gene>].
17. Ensembl. [<http://ensembl.org>].
18. PubMed. [<http://pubmed.org>].
19. Soldatos T, O'Donoghue SI, Satagopam VP, Brown NP, Jensen LJ, Schneider R: **Martini: using literature keywords to compare gene sets.** *Nucleic Acid Res* 2010, **38**:26-38.
20. Etzold T, Argos P: **SRS - an indexing and retrieval tool for flat file data libraries.** *Bioinformatics* 1993, **9**:49-57.
21. eUtils. [http://www.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html].
22. LIBLINEAR- A Library for Large Linear Classification. [<http://www.csie.ntu.edu.tw/~cjlin/liblinear/>].
23. Hsu Chih-Wei, Chang Chih-Chung, Lin Chih-Jen: **A Practical Guide to Support Vector Classification.** [<http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>].
24. Pattin KA, White BC, Barney N, Gui J, Nelson HH, Kelsey KT, Andrew AS, Karagas MR, Moore JH: **A computationally efficient hypothesis testing method for epistasis analysis using multifactor dimensionality reduction.** *Genet Epidemiol* 2009, **33**:87-94.
25. Jensen LJ, Jensen TS, de Lichtenberg U, Brunak S, Bork P: **Co-evolution of transcriptional and post-translational cell-cycle regulation.** *Nature* 2006, **443**:594-597.
26. Medical Subject Headings. [<http://www.nlm.nih.gov/mesh/>].
27. Barbosa-Silva A, Mudado M, Ortega JM: **Plant Defense Mechanisms Database (PDM): Building and Evaluation.** *Proceedings of the 1st International Conference of The Brazilian Association of Bioinformatics and Computational Biology (X-Meeting): 4-7 October 2005 Caxambu-MG;* 2005, 110.
28. Martin GB, Bogdanove AJ, Sessa G: **Understanding the functions of plant disease resistance proteins.** *Ann Rev Plant Biol* 2003, **54**:23-61.
29. Kessman H, Staub T, Hofmann C, Maetze T, Herzog J, Ward E, Uknes S, Ryals J: **Induction of Systemic Acquired Disease Resistance in Plants by Chemicals.** *Ann Rev Phytopathol* 1994, **32**:439-459.

30. Morel J-B, Dangl JL: The hypersensitive response and the induction of cell death in plants. *Cell Death & Differentiation* 1997, 4:671-683.
31. Caipirini examples. , <http://caipirini.org/caipiriniATexample.html>; <http://caipirini.org/caipiriniCellCycleExampleSphase.html>; <http://caipirini.org/caipiriniCellCycleExampleNotSphase.html>; <http://caipirini.org/caipiriniCellCycleExampleNotMESH.html>.

doi:10.1186/1756-0381-5-1

Cite this article as: Soldatos et al.: Caipirini: using gene sets to rank literature. *BioData Mining* 2012 5:1.

Submit your next manuscript to BioMed Central
and take full advantage of:

- Convenient online submission
- Thorough peer review
- No space constraints or color figure charges
- Immediate publication on acceptance
- Inclusion in PubMed, CAS, Scopus and Google Scholar
- Research which is freely available for redistribution

Submit your manuscript at
www.biomedcentral.com/submit

