The LAILAPS Search Engine: Relevance Ranking in Life Science Databases

Matthias Lange[1], Karl Spies[1], Joachim Bargsten[1], Gregor Haberhauer[3], Matthias Klapperstück[1], Michael Leps[4], Christian Weinel[3], Röbbe Wünschiers[2], Mandy Weißbach[1], Jens Stein[4] and Uwe Scholz[1]

[1]Research Group Bioinformatics and Information Technology, Leibniz Institute of Plant Genetics and Crop Plant Research (IPK), Gatersleben, Germany

[2]University of Applied Sciences, Mittweida, Germany

[3]BASF SE, Computational Chemistry and Biology, Ludwigshafen, Germany

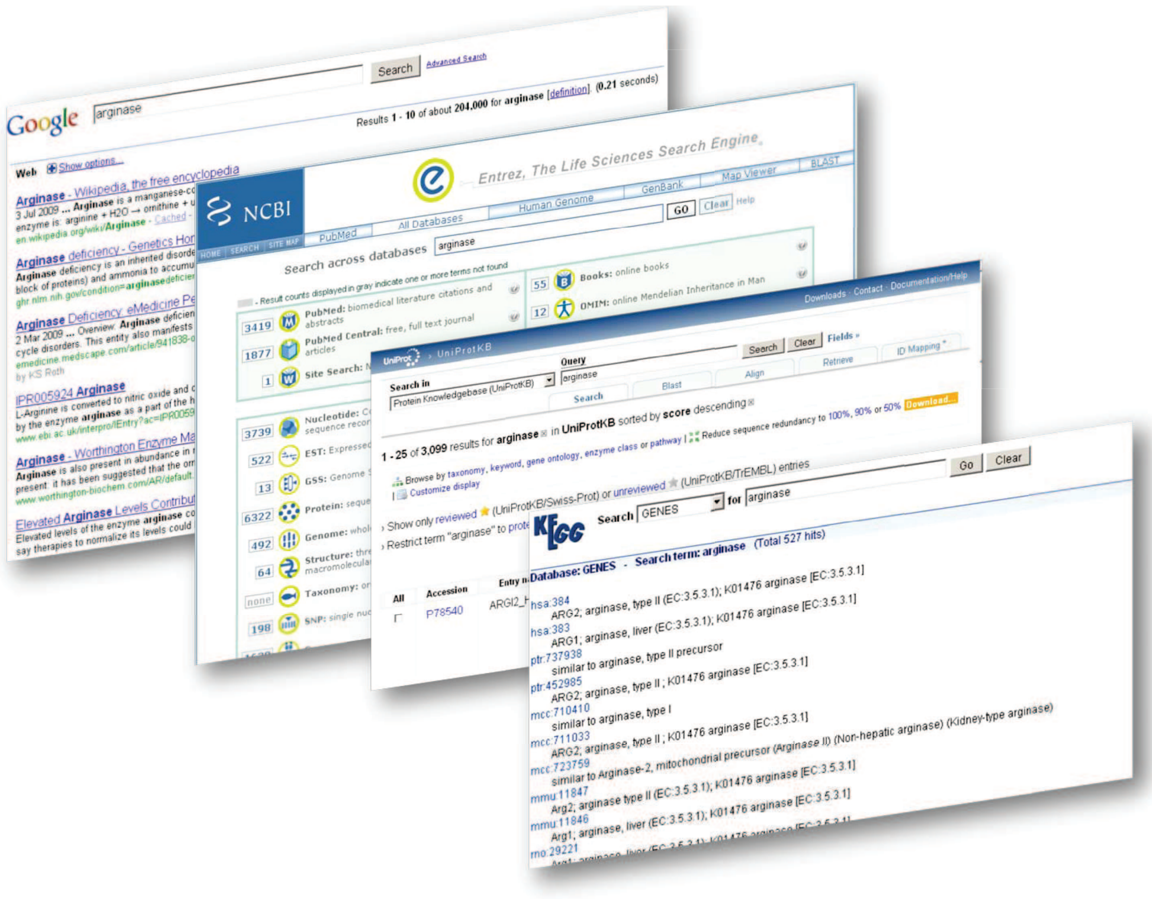
[4]SunGene GmbH, Gatersleben, Germany

Summary

Search engines and retrieval systems are popular tools at a life science desktop. The manual inspection of hundreds of database entries, that reflect a life science concept or fact, is a time intensive daily work. Hereby, not the number of query results matters, but the relevance does. In this paper, we present the LAILAPS search engine for life science databases. The concept is to combine a novel feature model for relevance ranking, a machine learning approach to model user relevance profiles, ranking improvement by user feedback tracking and an intuitive and slim web user interface, that estimates relevance rank by tracking user interactions. Queries are formulated as simple keyword lists and will be expanded by synonyms. Supporting a flexible text index and a simple data import format, LAILAPS can easily be used both as search engine for comprehensive integrated life science databases and for small in-house project databases.
With a set of features, extracted from each database hit in combination with user relevance preferences, a neural network predicts user specific relevance scores. Using expert knowledge as training data for a predefined neural network or using users own relevance training sets, a reliable relevance ranking of database hits has been implemented.
In this paper, we present the LAILAPS system, the concepts, benchmarks and use cases. LAILAPS is public available for SWISSPROT data at
`http://lailaps.ipk-gatersleben.de`

1 Introduction

”Finding information in the WWW is not much of a challenge. Just head for Google or Entrez and get the related web page or database entry.” This issue can be heard frequently talking to biologist, who search information about a certain biological object [1]. However, finding reliable information about the function of a protein or seeking the protein that is involved in a certain activity in the cell cycle, is much more challenging. One has the choice of about 1,100 life science databases and billions of database records [2]. Even if one reduces the number of



arginase

Results 1 - 10 of about 204,000 for arginase [definition]. (0.21 seconds)

Arginase - Wikipedia, the free encyclopedia
3 Jul 2009 ... Arginase is a manganese-co
enzyme is: arginine + H2O → ornithine + u
en.wikipedia.org/wiki/Arginase - Cached -

Arginase deficiency - Genetics Hom
Arginase deficiency is an inherited disorde
block of proteins) and ammonia to accumu
ghr.nlm.nih.gov/condition=arginasedeficier

Arginase Deficiency: eMedicine Pe
2 Mar 2009 ... Overview: Arginase deficien
cycle disorders. This entity also manifests
emedicine.medscape.com/article/941838-o
by KS Roth

IPR005924 Arginase
L-Arginine is converted to nitric oxide and c
by the enzyme arginase as a part of the h
www.ebi.ac.uk/interpro/IEntry?ac=IPR0059

Arginase - Worthington Enzyme Ma
Arginase is also present in abundance in
present: it has been suggested that the orn
www.worthington-biochem.com/AR/default.

Elevated Arginase Levels Contribut
Elevated levels of the enzyme arginase co
say therapies to normalize its levels could

Search across databases arginase

- Result counts displayed in gray indicate one or more terms not found

3419	PubMed: biomedical literature citations and abstracts
1877	PubMed Central: free, full text journal articles
1	Site Search: N
55	Books: online books
12	OMIM: online Mendelian Inheritance in Man
3739	Nucleotide: C sequence recor
522	EST: Expressed
13	GSS: Genome
6322	Protein: seque
492	Genome: whol
64	Structure: thre macromolecula
none	Taxonomy: or
198	SNP: single nuc

Search in: Protein Knowledgebase (UniProtKB) Query: arginase

1 - 25 of 3,099 results for arginase in UniProtKB sorted by score descending

Browse by taxonomy, keyword, gene ontology, enzyme class or pathway | Reduce sequence redundancy to 100%, 90% or 50%
| Customize display

› Show only reviewed (UniProtKB/Swiss-Prot) or unreviewed (UniProtKB/TrEMBL) entries
› Restrict term "arginase" to prote

All	Accession	Entry n
☐	P78540	ARGI2_H

Search GENES for arginase

Database: GENES - Search term: arginase (Total 527 hits)

hsa:384
ARG2; arginase, type II (EC:3.5.3.1); K01476 arginase [EC:3.5.3.1]
hsa:383
ARG1; arginase, liver (EC:3.5.3.1); K01476 arginase [EC:3.5.3.1]
ptr:737938
similar to arginase, type II precursor
ptr:452985
ARG2; arginase, type II ; K01476 arginase [EC:3.5.3.1]
mcc:710410
similar to arginase, type I
mcc:711033
ARG2; arginase, type II ; K01476 arginase [EC:3.5.3.1]
mcc:723759
similar to Arginase-2, mitochondrial precursor (Arginase II) (Non-hepatic arginase) (Kidney-type arginase)
mmu:11847
Arg2; arginase type II (EC:3.5.3.1); K01476 arginase [EC:3.5.3.1]
mmu:11846
Arg1; arginase, liver (EC:3.5.3.1); K01476 arginase [EC:3.5.3.1]
rno:29221

Figure 1: Search Engines in Life Science – the screenshots show examples of popular search engines, to investigate protein function: Google, Entrez, UniProt and KEGG.

databases and records using database integration systems and powerful query systems, there are still too many results for a simple query like ”`arginase`” - an enzyme involved in the urea cycle. As shown in Figure 1, one gets 6322 hits in NCBI Entrez Protein databases, 3099 in Uniprot, and 527 in KEGG GENES (data from October 2009).

Intuitively, the first choice are web search engines. Web site ranking techniques order query hits by relevance. But, trying to apply ranking methods that were developed to rank natural language text or WWW-sites to life science content and databases is questionable [3]. For example, the top ranked Google hit for ”`arginase`” is a Wikipedia page. This is because the page is referenced by a high number of web-pages or Google assigned a manual defined priority rank. Here, the hypothesis is: *A high hyperlink in-degree of a page means high popularity and high popularity means high relevance*.

In order to find scientific relevant database entries, scientists need strong scientific evidence in relation to the specific research field. A dentist has other relevance criteria than a plant biologist or a patent agent. The intuitive and commonly used way at the scientist’s desktop is query refinement. Criteria like who published, in which journal, for which organism, evidence scores, surrounding keywords etc. matter. Even complete search guides, e.g. for dentists were published [4].

Other ranking algorithms use Term Frequency - Inverse Document Frequency (TF-IDF) as ranking criteria. Apache-Lucene[1] is a popular implementation of this concept and is frequently used in bioinformatics, like LuceGene from the GMOD project [5], which is used for the EBI 'google' like search frontend EBeye. The TF-IDF approach works well, but misses the semantic context between the database entries and the query.

Another approach is the probabilistic relevancy ranking [6], whereas probabilistic values for the relevance of database fields and word combinations have to be predefined. In combination with a user feedback system, the probabilistic approach shows promising ranking performance [7].

Semantic search engines use methods from natural language processing and dictionaries to predict the semantic most similar database entries. Such conceptual search strategies, implemented in GoPubMed [8] or ProMiner [9], are frequently used algorithms in text mining projects.

The combination and abstraction of the mentioned relevance indicators motivated the development of a feature model for life science databases, a user feedback system and a machine learning ranking engine. The feature model was abstracted from dynamic (query dependent) and static (database dependent) relevance properties. These properties were derived from a study of human ranking behavior in life science labs. Furthermore, a feedback module collects user implicit and explicit relevance ratings as input for an incremental collection of training data. Finally, user specific neural networks are trained and used for scoring the relevance of database entries in the context of the authenticated user or user group. These concepts are implemented in form of the LAILAPS search engine.

2 LAILAPS Method for Relevance Ranking

Nearly every search engine, including LAILAPS, incorporates a scoring or ranking function to calculate a relevance for an entry. The central LAILAPS hypothesis is, that the relevance score is context-dependent and the absolute rank position can be determined by sorting the relative scores. We apply information theory and postulate, that the relevance of a database entry is dependent from two factors: its *content* and its *interpretation* by the search engine user. For the first factor we found that the relevance decision is based on a small number of core properties of the content. These core properties are used to deduce a feature model, that expresses all important properties of a database entry as feature vector. The factor user interpretation is realized by LAILAPS in form of a feedback system and hand curated reference sets of relevant rated database entries. Both factors are used for the scoring algorithm. The algorithm uses artificial neural networks as method to estimate the relevance score of a database entry based on these factors.

2.1 LAILAPS Feature Model

A relevance scoring function for life science databases is highly dependent on the underlying data. In contrast to full text data like PubMed or even traditional web sites, a database entry in a life science database is

[1]`http://lucene.apache.org`

1. structured and split into blocks (e.g. attributes, entities),
2. enriched by metadata (e.g. name of an attribute, quality information),
3. a compressed excerpt of a fact and
4. a mix of pure values and natural language data text (e.g. stoichiometric biochemical reaction vs. function description).

This properties lead to the conclusion, that classical ranking methods miss important properties and are suboptimal for life science database entries. Consequently, we had to define our own set of features, that combines traditional and life science database specific ones. Motivated by the observation of user behavior during search engine result inspection, we introduced a set of 9 features F, presented in table 1. These features are intuitively used by scientists, who briefly

	feature	description
F_1	attribute	in which attribute the query term was found
F_2	database	to which database the found entry is included
F_3	frequency	the frequency of all query terms in the entry and attribute
F_4	coocurence	express how close and in which order the query term were found
F_5	keyword	gives information, if good or bad keyword are present near to the query terms
F_6	organism	to which organism the databse entry relates to
F_7	sequence length	the length of the sequence described by the database entry
F_8	text position	which portion of the attribute is covered by the query term
F_9	synonym	gives information if the hit was produced by an automatic synonym expansion

Table 1: Overview of the LAILAPS feature set

screen database entries for potential relevance. The features are both sufficient to estimate the potential relevance, and computationaly efficiently determinable.

2.2 The Ranking Method

Based on these features, we defined a function for each feature

$$\sigma_f(T, D, A, \{P\}) \rightarrow \omega \mid f \in F_1, \ldots, F_9 \tag{1}$$

Where, T is the query term hit; D is the database hit; A is the attribute hit and $\{P\}$ is the set of matched positions for the query term in the database entry.

Hence, these functions compute for each database entry, where at least one query term or synonym matches, scalar values $\omega_1, \ldots, \omega_9$. The final step is a ranking function, which computes a relevance score τ from a vector $\vec{\omega}$ of the 9 feature values:

$$rank((\omega_1, \ldots, \omega_9)) \rightarrow \tau \tag{2}$$

These relevance scores can be ordered such as

$$\tau_1 < \tau_2 \Rightarrow \tau_1 \text{ is less relevant than } \tau_2 \tag{3}$$

The final relevance ranking is the order of all relevance scores:

$$(\tau_1, \ldots, \tau_n) \mid \forall \tau : \ \tau_{n-1} < \tau_n \tag{4}$$

The function $rank$ constitutes a regression problem. We have to map a vector of 9 features to a scalar, continuous relevance score. Such problems are solved by machine learning approaches. In particular, supervised method perform well for text mining systems [10]. We found that artificial neural netwoks show best performance for our regression problem [11]. Using a set of training data, we trained a feed-forward neural network with 9 neurons at the input and 7-4 in the hidden layer (see Figure 2).

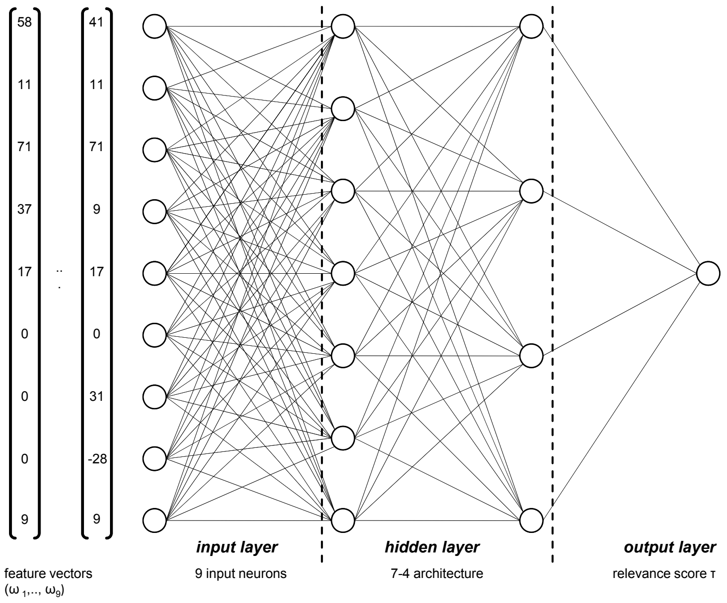


Figure 2: neural network for relevance ranking – the used neural network predict for a vector $\vec{\omega}$ of 9 feature values the relevance of the database entry.

Our industrial and academic partners provided a set of plant metabolic queries with 1089 manualy relevance ranked database records (see Table 2). This reference ranking list was separated into three confidence classes: high, medium and low. With these data, a neural network for plant metabolism was trained.

Query Text	Size	Category Split-Up (hi/me/lo)
industrial use case 1	20	6 / 4 / 10
”pinene synthase”	18	10 / 3 / 5
industrial use case 2	39	8 / 13 / 18
industrial use case 3	64	14 / 32 / 18
”gamma tocopherol methyltransferase”	38	21 / 9 / 8
”ent-kaurene synthase”	65	17 / 38 / 10
”chlorophyll synthase”	77	17 / 54 / 6
industrial use case 4	134	35 / 68 / 31
”cinnamyl-alcohol dehydrogenase”	214	45 / 36 / 133
industrial use case 5	17	3 / 4 / 10
”dihydrokaempferol 4-reductase”	65	9 / 29 / 27
”l-ascorbate peroxidase”	100	69 / 12 / 19
”morphine 6-dehydrogenase”	35	2 / 15 / 18
”zeaxanthin epoxidase”	51	21 / 2 / 28
”squalene monooxygenase”	84	24 / 30 / 30
”acetoacetyl-coa synthetase”	68	14 / 36 / 18

Table 2: Overview of the training data set.

To train the network, we split up the training data into 80% for training and 20% for testing and used 500 training epochs. These parameters were estimated by minimizing the mean squared error ϵ over the training set:

$$\epsilon = \frac{1}{n}\sum_{i=1}^{n}(\tau_i' - \tau_i)^2 \mid n = \text{size of training set}; \tau_i' = \text{manual score}; \tau_i = \text{predictes score} \quad (5)$$

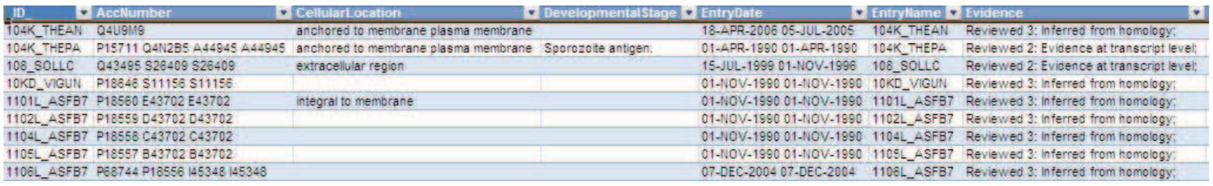
In each epoch the change of ϵ was checked. After 500 epochs, no significant decrease was found and the final mean square error was 0.33 [11].

2.3 Implementation of LAILAPS

LAILAPS was developed as a 3-tier web application using Apache Tapestry[2] as web application-framework, ORACLE as database backend and a JAVA-implemented ranking logic featured by the Java Object Oriented Neural Engine (JOONE)[3]. The backend database stores the loaded life science databases in an entity-attribute-value (EAV) [12] adapted database schema. This flexible concept enables the import of RFC-compatible CSV-formatted[4] exports from life science databases, whereas each row comprise a database record and its columns the fields (see Figure 3).

For the imported databases, an inverse text index is computed and synonyms are loaded. In the public available system, we provide protein synonyms extracted from UNIPROT/SWISSPROT [13] and BRENDA [14]. The ranking logic

[2] `http://tapestry.apache.org`
[3] `http://www.jooneworld.com/`
[4] `http://tools.ietf.org/html/rfc4180`



ID	AccNumber	CellularLocation	DevelopmentalStage	EntryDate	EntryName	Evidence
104K_THEAN	Q4U9M9	anchored to membrane plasma membrane		18-APR-2006 05-JUL-2005	104K_THEAN	Reviewed 3: Inferred from homology;
104K_THEPA	P15711 Q4N2B5 A44945 A44945	anchored to membrane plasma membrane	Sporozoite antigen.	01-APR-1990 01-APR-1990	104K_THEPA	Reviewed 2: Evidence at transcript level;
108_SOLLC	Q43495 S26409 S26409	extracellular region		15-JUL-1999 01-NOV-1996	108_SOLLC	Reviewed 2: Evidence at transcript level;
10KD_VIGUN	P18646 S11156 S11156			01-NOV-1990 01-NOV-1990	10KD_VIGUN	Reviewed 3: Inferred from homology;
1101L_ASFB7	P18560 E43702 E43702	integral to membrane		01-NOV-1990 01-NOV-1990	1101L_ASFB7	Reviewed 3: Inferred from homology;
1102L_ASFB7	P18559 D43702 D43702			01-NOV-1990 01-NOV-1990	1102L_ASFB7	Reviewed 3: Inferred from homology;
1104L_ASFB7	P18558 C43702 C43702			01-NOV-1990 01-NOV-1990	1104L_ASFB7	Reviewed 3: Inferred from homology;
1105L_ASFB7	P18557 B43702 B43702			01-NOV-1990 01-NOV-1990	1105L_ASFB7	Reviewed 3: Inferred from homology;
1106L_ASFB7	P68744 P18556 I45348 I45348			07-DEC-2004 07-DEC-2004	1106L_ASFB7	Reviewed 3: Inferred from homology;

Figure 3: Database import format for LAILAPS – the Excel-screenshot shows a snapshot of a CSV-formatted SWISSPROT export. Each row represents one database entry. The columns are the fields. The first row contains all field names.

1. computes all matched positions per database entry,
2. extracts for each database entry an n-dimensional feature vector with 9 basis feature classes and
3. predicts a relevance probability using user specific neural networks, which maps the feature vectors to the users specific relevance score.

Since the ranking profile is computed in context of the authenticated user, a valid user login is recommended. Otherwise, a default context with a pre-trained neural network is used.

The relevance ordered query hits are assigned to the ranking profile and rendered by the middleware into a number of web pages (see Figure 4). The embedded feedback system provides a tool for the user to rate the relevance of a particular database entry. The ratings are stored in the database backend and used to accumulate user training data. The LAILAPS feedback system is transparently embedded into the result browser. By opening the database detail browser, AJAX[5] code is injected into the original data HTML presentation, which is for example, provided by the SRS@EBI data retrieval system [15]. This code collects the user rating of the database entry and trackes user interactions. This feedback is used to enrich the original training data in the related ranking profile.

3 Results and Discussion

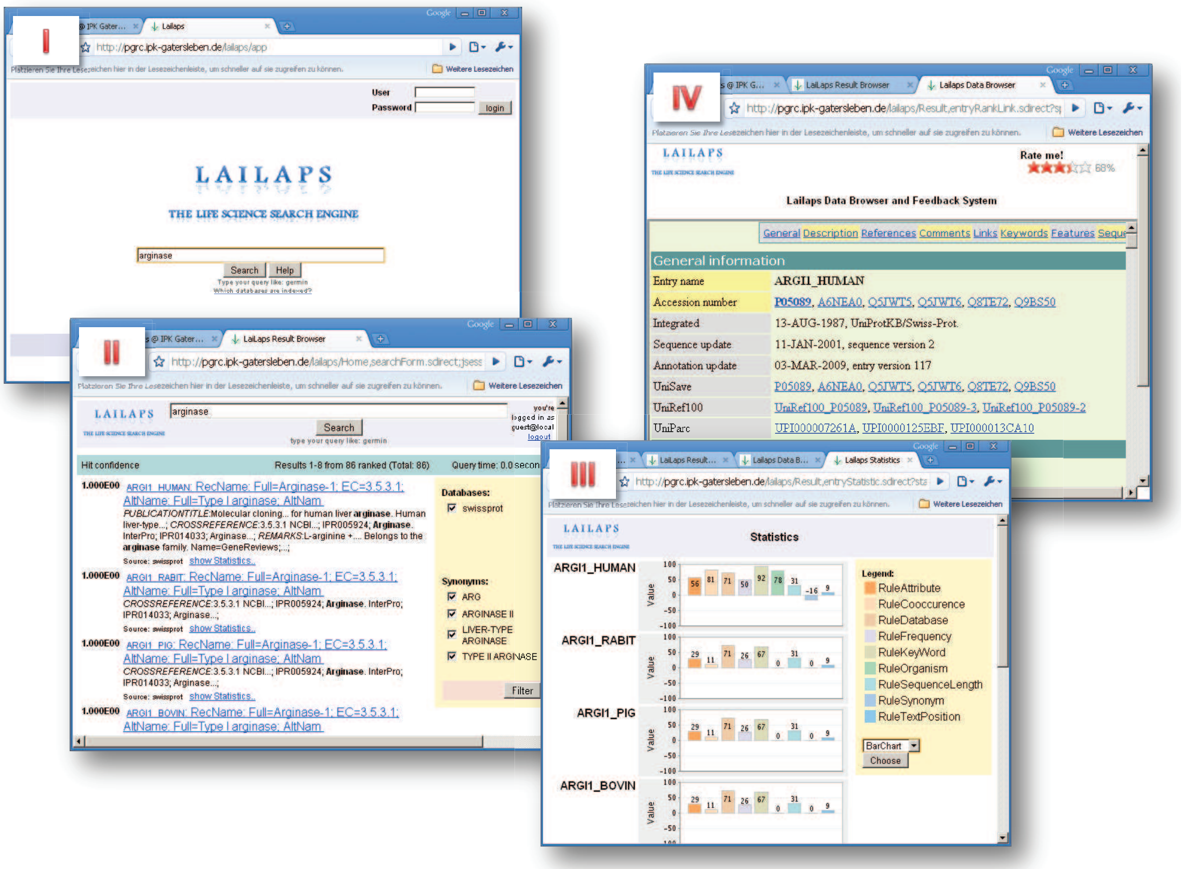
Searching scientific databases effectively necessitates the use of contemporary software to locate desired and meaningful information according to the users scientific or project priorities. However, the combination of relevance ranking and life science data retrieval is still missing in life science information systems. LAILAPS fills this gap and provides a search engine for integrated or single instance life science databases in combination with an efficient ranking system.

We have evaluated the relevance prediction using the standard measures for precision Pr, recall Re and F_1 score:

$$Pr = \frac{TP}{TP + FP}; \quad Re = \frac{TP}{TP + FN}; \quad F_1 = \frac{2 * Pr * Re}{Pr + Re}$$

As mentioned, we have used a curated reference set of plant metabolic queries. In order to decide whether a database entry has been correctly ranked or not, we do not consider its concrete

[5]Asynchronous JavaScript and XML



Hit confidence — Results 1-8 from 86 ranked (Total: 86) — Query time: 0.0 secon

1.000E00 ARGI1_HUMAN: RecName: Full=Arginase-1; EC=3.5.3.1; AltName: Full=Type I arginase; AltNam
PUBLICATIONTITLE:Molecular cloning... for human liver **arginase**. Human liver-type...; *CROSSREFERENCE*:3.5.3.1 NCBI...; IPR005924; **Arginase**. InterPro; IPR014033; Arginase...; *REMARKS*:L-arginine +.... Belongs to the **arginase** family. Name=GeneReviews;...;
Source: swissprot show Statistics..

1.000E00 ARGI1_RABIT: RecName: Full=Arginase-1; EC=3.5.3.1; AltName: Full=Type I arginase; AltNam
CROSSREFERENCE:3.5.3.1 NCBI...; IPR005924; **Arginase**. InterPro; IPR014033; Arginase...;
Source: swissprot show Statistics..

1.000E00 ARGI1_PIG: RecName: Full=Arginase-1; EC=3.5.3.1; AltName: Full=Type I arginase; AltNam
CROSSREFERENCE:3.5.3.1 NCBI...; IPR005924; **Arginase**. InterPro; IPR014033; Arginase...;
Source: swissprot show Statistics..

1.000E00 ARGI1_BOVIN: RecName: Full=Arginase-1; EC=3.5.3.1; AltName: Full=Type I arginase; AltNam

Databases: ☑ swissprot
Synonyms: ☑ ARG ☑ ARGINASE II ☑ LIVER-TYPE ARGINASE ☑ TYPE II ARGINASE
Filter

Lailaps Data Browser and Feedback System

General Description References Comments Links Keywords Features Seque

General information

Entry name	ARGI1_HUMAN
Accession number	P05089, A6NEA0, Q5JWT5, Q5JWT6, Q8TE72, Q9BS50
Integrated	13-AUG-1987, UniProtKB/Swiss-Prot.
Sequence update	11-JAN-2001, sequence version 2
Annotation update	03-MAR-2009, entry version 117
UniSave	P05089, A6NEA0, Q5JWT5, Q5JWT6, Q8TE72, Q9BS50
UniRef100	UniRef100_P05089, UniRef100_P05089-3, UniRef100_P05089-2
UniParc	UPI000007261A, UPI0000125EBF, UPI000013CA10

Figure 4: LAILAPS view of an example query session – The four browser windows represent the common query workflow of LAILAPS. Start screen with an optional login to load a customized ranking profile (I). The query might be specified as a single word, combination of words delimited by a whitespace or quoted phrases. The ranked query result is presented as a list of relevance sorted database accessions (II), with a short hit description, the evidence value, the hyper link to the original data source and a link to the scoring statistics in window (III). The exploration of hits is supported by a detail browser and feedback system (IV). The original data is displayed and the user might rank the relevance of the hit for later training of the user ranking profile.

ranking position. Because of combinatorial explosion, for human curators it is nearly impossible to find a correct relevance order among hundreds of database entries. Rather, the knowledge quality of a certain database entry is crucial. Consequently, the database entries were classified into three confidence classes: ”HIGH”, ”MEDIUM” and ”LOW”. The ”HIGH”-class comprises the top entries with proven and reliable knowledge. The class ”MEDIUM” includes all those, that could be interesting but are uncertain. The class ”LOW” includes all data, that has insufficient knowledge value or bad quality indicators. For each query, those classes form sub sets of a query result R such as $R = R_H \cup R_M \cup R_L$, whereas $R_H \cap R_M \cap R_L = \emptyset$. Each set forms a continuous window in the list of results. E.g., for a query result of 100 database entries, the window for R_H ranges from position 1-20, R_M from position 21-80, and R_L from 81-100.

In order to evaluate the LAILAPS rankings, we have to compare a list of relevance ordered database entries with the confidence classes of the reference set. We can’t compare absolute ranking positions, because the elements in the confidence classes have no order. But we can say in which range (window) of rank positions a LAILAPS ranked entry should be, to fall into a certain confidence class. This consideration is used to define the true positives (TP), false positives (FP) and false negatives (FN):

error type	LAILAPS benchmark semantics
true positive (TP)	the rank position is in the same window as in the reference set
false positive (FP)	the rank position is in a different window as in the reference set
false negative (FN)	the database entry was not found by LAILAPS

Table 3: Definition of evaluation error types.

For example, for one use case the curators sorted 20 entries into the class ”HIGH”. 18 of the top 20 LAILAPS ranked result are in the window R_H. In this case, the precision is $\frac{18}{18+2} = 0.9$.

Because all database entries of the reference set were found by LAILAPS, the recall is $Re = 100\%$. This is because the text indexing is the basis for matching query terms. The text index uses a tokenizer, that decomposes text into words. LAILAPS use the same text decomposition rules and the same databases as the reference retrieval systems. Furthermore, no synonym expansion of query terms was used. Thus, the hits of the reference systems could be reproduced by LAILAPS and no false negative hits exits.

The overall benchmarking result of 16 queries is shown in Figure 5. The average recall, preci-

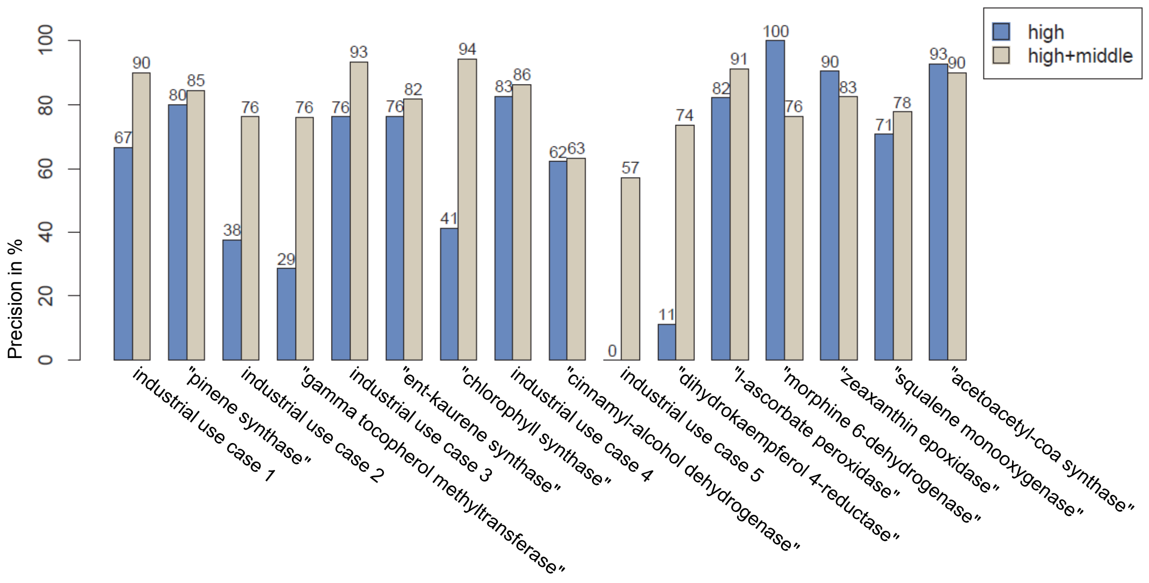


Figure 5: LAILAPS recall for use case queries

sion and F_1 values are shown in Table 4.

confidence class	precision	recall	F_1
”HIGH”	62%	100%	76
”MEDIUM” $\bigcup$ ”HIGH”	81%	100%	90

Table 4: Evaluation of LAILAPS relevance ranking results.

In average we achieve a better precision than existing search engines [3]. Training and benchmark data for the non-industrial use cases are available by request to the authors.

The training data were collected in the application scenario of plant research and queries for protein functions. In order to bring LAILAPS to a broad community, we provide a public installation of the LAILAPS search engine. This will help to improve performance and to include more user domain specific ranking profiles. Because of limited database and project resources, the public, non-commercial version is restricted to SWISSPROT data. A comprehensive set of databases is available for registered users on request to the authors.

LAILAPS combines a clean, powerful and easy to use human computer interface with a machine learning based, context sensitive ranking system. It comprises a search engine with a self trained neural network ranking system, which brings a new quality and determinism into the scientific knowledge exploration.

Acknowledgements

Thanks to Thomas Münch and Steffen Flemming for their web hosting service and their valuable technical support. The project was funded by the IPK-Gatersleben and BASF SE Ludwigshafen.

References

[1] A. Divoli, M. Hearst, and Wooldridge M. A. Evidence for Showing Gene/Protein Name Suggestions in Bioscience Literature Search Interfaces. In *Pacific Symposium on Biocomputing*, volume 13, pages 568–579, 2008.

[2] M.l Y. Galperin and G. R. Cochrane. Nucleic Acids Research annual Database Issue and the NAR online Molecular Biology Database Collection in 2009. *Nucleic Acids Research*, 37(suppl_1):D1–4, 2009.

[3] Matthew Richardson, Amit Prakash, and Eric Brill. Beyond pagerank: machine learning for static ranking. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 707–715, New York, NY, USA, 2006. ACM.

[4] J. Day. The quest for information: A guide to searching the internet. *Journal of Contemporary Dental Practice*, 2(4):033–043, 2001.

[5] B. O'Connor, A. Day, S. Cain, O. Arnaiz, L. Sperling, and L. Stein. Gmodweb: a web framework for the generic model organism database. *Genome Biology*, 9(6):R102, 2008.

[6] Nicholas C. Ide, Russell F. Loane, and Dina Demner-Fushman. Essie: A Concept-based Search Engine for Structured Biomedical Text. *Journal of the American Medical Informatics Association*, 14(3):253–263, 2007.

[7] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26, New York, NY, USA, 2006. ACM.

[8] A. Doms and M. Schroeder. GoPubMed: exploring PubMed with the Gene Ontology. *Nucleic Acids Research*, 33(suppl_2):W783–786, 2005.

[9] Daniel Hanisch, Katrin Fundel, Heinz-Theodor Mevissen, Ralf Zimmer, and Juliane Fluck. Prominer: rule-based protein and gene entity recognition. *BMC Bioinformatics*, 6(Suppl 1):S14, 2005.

[10] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 89–96, New York, NY, USA, 2005. ACM.

[11] J. Bargsten. Nutzerprofilgestütztes Schätzen von Rankingparametern für Suchmaschinen-Anfragen an integrierte biologische Datenbanken. Diplomarbeit, Martin-Luther-Universität Halle-Wittenberg, Institut für Informatik, 2009.

[12] P. M. Nadkarni, L. Marenco, R. Chen, E. Skoufos, G. Shepherd, and P. Miller. Organization of Heterogeneous Scientific Data Using the EAV/CR Representation. *Journal of the American Medical Informatics Association*, 6(6):478–493, 1999.

[13] The UniProt Consortium. The Universal Protein Resource (UniProt) 2009. *Nucleic Acids Research*, 37(suppl_1):D169–174, 2009.

[14] I. Schomburg, A. Chang, and D. Schomburg. BRENDA, enzyme data and metabolic information. *Nucleic Acids Research*, 30(1):47–49, 2002.

[15] E. M. Zdobnov, R. Lopez, R. Apweiler, and T. Etzold. The EBI SRS server – new features. *Bioinformatics*, 18(8):1149–1150, 2002.