

2 The Information Retrieval data model

2.1 Introduction: Data models

2.2 Text in BioSciences

2.3 Information Retrieval Models

2.3.1. Boolean Model

2.3.2 Vector space Model

2.3.3 Efficient implementation of VSM

2.4 Evaluation of Retrieval effectiveness

2.5 Molecular Biology application

LOCUS AL33DMOR4 314 bp ds-DNA PHG 24-MAY-1991
 DEFINITION Bacteriophage alpha3 deletion mutant DNA for origin region (-ori)
 of replication

ACCESSION X15716

KEYWORDS origin of replication.

SOURCE Bacteriophage alpha3 DNA.

ORGANISM Bacteriophage alpha3
 Viridae; ds-DNA nonenveloped viruses; Siphoviridae.

REFERENCE 1 (bases 1 to 314) ← structured fields

AUTHORS Kodaira,K.I.

JOURNAL Unpublished (1989)

STANDARD full automatic

REFERENCE 2 (sites)

AUTHORS Nakano,K., Kodaira,K.I. and Taketo,A. ← multiple values

TITLE Properties of the bacteriophage alpha3 mutants with deletion and/or
 insertion in the complementary strand origin

JOURNAL Biochim. Biophys. Acta 1048, 43-49 (1990)

STANDARD full staff_review

COMMENT *source: clone=delo L105; See <J02444>, <M25640>, <X13332>, <X15713>
 to <X15715>, and <X15717> to <X15721> for other -ori sequences.

Data kindly reviewed (06-JAN-1990) by Kodaira K.-I. ← Natural Language (NL) text

From EMBL entry BA3DMOR4; dated 16-JAN-1991. ← reference to foreign data sets, "micro syntax"

BASE COUNT 68 a 67 c 82 g 97 t

ORIGIN

1 tgaagttgag cattaacca tgaaatgic ttgttgctc ggtgggttt gtctctgctg

61 cgtttgtct acgttgata ttaactcctg tactctgtt aataatgitt ttgttggtg

121 gatttgctt agocataat ggaaccctgg tctgtgatt gglaccatcg ctacgacctg

181 tgatttcat gaataccaag tctctcagcc gcttaataa aagcgctcgc cactccggt

241 taqtatcctg cccagtttag ggcagaccgg tacggagata cccgataaac taqgaacgtg ← long fields

2.1 Introduction: Datamodels

- What is appropriate for bio data?
 - No clear answer
 - Ideological positions not helpful
 - adaption of new techniques makes sense – but takes a long time - i.g. ASN.1 ⇔ XML, relational ⇔ Oo (?)
- Pragmatic requirements
 - Flexibility ... new types of objects, change of identifiers, ...
 - ease of use ...everything resulting from progress of science
 - few restrictions

Anything more flexible than natural language?

HS / BioDBS05-2-IR 3

Data models: the spectrum

- Databases
 - Rigid data models: relational, object-oriented
 - Database conformant to schema
 - Semantics of query q: subset of database
 - No texts, images, ... (originally)
- Semi structured DB / XML
 - Schema more flexible – if any
 - Many schema items
 - Text plays a big role
 - Semantics of queries: substructure of DB
- Information Retrieval
 - Data model: objects are sequences of terms
 - No modeling restrictions (natural language!?)
 - Semantics of query q: DB entries ordered by similarity to q (Ranking)
- Natural Language
 - VERY difficult to process automatically
 - ⇨ not really an option

More structure

Less structure

HS / BioDBS05-2-IR 4

2.2 Text in Bioinformatics

- Automatic processing of text
- Use statistical methods and heuristics for simple tasks!

Example: finding abbreviations

"Abbreviation mining" (Chang, Schütze, Altmann, Stanford)

Method: text alignment using dynamic programming

MEDLINE: "According to a system proposed by the European group for the immunological classification of leukemia (EGL)....."

European group for the immunological classification of leukemia
 E.....G.....I.....L.....

HS / BioDBS05-2-IR 5

Evaluation

- Abbreviation server: <http://abbreviation.stanford.edu/>
local1 local2

Evaluation

37 GB Medline, 452 entries of an lexikon of abbreviations

375 out of 452 found correctly ("Recall")

402 of 452 classified correctly ("Precision")

Details see: Chang et al: [Creating an online dictionary of Abbreviations from Medline.](#)
 Journal of the American Medical Informatics Association Volume 9 Number 6 Nov / Dec 2002

HS / BioDBS05-2-IR 6

Information Retrieval

- ... works on textual data
- Why useful in BioSciences
 - "Most data (knowledge?) buried in text" - journal paper, proceedings, databases...
 - ⇒ Standard retrieval task as in other disciplines
 - ... did you know that the number of scientific papers doubles every ~12 years
 - Information Retrieval techniques may be (are?) useful for similarity search in nucleotide sequences
 - comparison of a query string to EACH sequence in the DB takes time
 - text indexing techniques may help

HS / BioDBS05-2-IR 7

2.3 Information Retrieval models

• Document ("data") model

D = "set of documents",

K = { k_1, \dots, k_n } set of **index terms**

K ~ set of all words occurring in the database
Typically very large

For every $d_j \in D$, $k_i \in K$ there is a **weight** $w_{ij} \geq 0$, $w_{ij} \in \text{Real}$,
if k_i does not occur in $d_j \Rightarrow w_{ij} = 0$

$d_j = (w_{1j}, \dots, w_{nj})$ is the **document representation** of d_j

identify d_j' and d_j in most cases, i.e. $D = \{d_j \mid d_j = (w_{1j}, \dots, w_{nj})\}$

i.e. a document is a high dimensional vector of real numbers, most of them are 0, each component represents a term $\in K$.

HS / BioDBS05-2-IR 8

2.3.1 Boolean retrieval

• Model

- $w_{ij} = 1$ if term k_i occurs in document d_j , else 0
- Query language: boolean expression of $k_i \in K$
- Evaluation of a query q:
 - let $d_j \in D$ a document vector of 0 and 1,
 - if $q = k_i$ then d matches q iff $d_{ij} = 1$
 - if $q = "q_1 \text{ AND } q_2"$ q matches d_{ij} if q_1 matches d_{ij} and q_2 matches d_{ij}
 - if $q = "q_1 \text{ OR } q_2"$ q matches d_{ij} if q_1 matches d_{ij} or q_2 matches d_{ij}
 - if $q = "NOT q_1"$ q matches d_{ij} if q_1 does not match d_{ij}

• Implementation

- Conceptually simple
- Efficient query evaluation
- many library systems / online retrieval systems work use it

HS / BioDBS05-2-IR 9

Boolean retrieval

• Issues

- Very restrictive evaluation: binary decision
Wanted: mapping $s: Q \times D \rightarrow [0, 1]$
Q is the set of all queries
- Every term has the same influence on the result
Wanted: **weight** should reflect "**importance**" of term
- Example:
term "protein" occurs in many documents
many times, term "propylthiouracil" less frequent...
In a search for "propylthiouracil AND protein" both have the same significance.
- For $q = "k_1 \text{ OR } \dots k_n"$ a document matches if at least one term matches.
No difference if one or all terms match.

HS / BioDBS05-2-IR 10

Boolean retrieval

• Coordinate level match

- Let q be in disjunctive normal form:
 $q' = \text{DISJ}_i (t_{i1} \text{ AND } t_{i2} \text{ AND } \dots \text{ AND } t_{ik}), t_{ij} = 0 \text{ or } 1$
- Example: $q = (\text{TEL or gene 6}) \text{ and oncogene}$
 $q' = (111) \text{ OR } (101) \text{ OR } (011)$
- Extend each disjunctive term by 0's for all terms in K not occurring in q
- $q' = (0000100011000) \text{ OR } \dots = q\text{Sig}_1 \text{ OR } \dots \text{ OR } q\text{Sig}_k$

term i = TEL
term j = GENE
term k = oncogene

$$s(q, d) = \max_i (q\text{Sig}_i * d)$$

(* : scalar product)

Means: the more query terms found in document, the better

HS / BioDBS05-2-IR 11

Boolean Retrieval

• Discussion

- (+) Ranking
- (+) number of matching query terms in document d define rank of d
- (-) Rank dependent on number of query terms
- (-) Documents with many terms tend to be ranked higher
- (-) Terms which occur frequently in documents are treated in the same way as infrequent terms

• Requirement

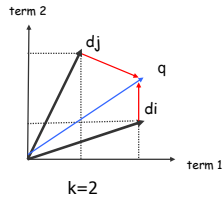
- More general term weights
- Normalization of ranking $s(q, dj)$

HS / BioDBS05-2-IR 12

2.3.2 Vector space model

• Model

- Documents: points in a $|K| = n$ – dimensional vector space.
- Weights normalized e.g. $0 \leq w \leq 1$
- Terms are independent of each other ("orthogonal")



- Queries
.... are (formally) documents: $q = (q_1, q_2, \dots, q_n)$

- Needed: measure of similarity between document and query, e.g. vector difference.

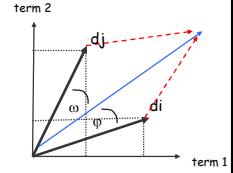
HS / BioDBS05-2-IR 13

Vector space: similarity function

• Heuristic similarity functions

- Scalar product?
 $w_1j \cdot q_1 + w_2j \cdot q_2 + \dots + w_nj \cdot q_n$
not bounded, may become arbitrarily large
- Cosine measure
 $\text{Cos}(d_j, q) = \cos \varphi$
 $= \frac{d_j \cdot q}{|d_j| \cdot |q|}$
 $= \frac{\sum w_{ij} \cdot q_i}{\sqrt{(\sum w_{ij}^2)} \cdot \sqrt{(\sum q_i^2)}}$

Measures angle between query vector and document and normalize.



HS / BioDBS05-2-IR 14

Weights

• How to assign weights to documents / queries

- Manual weight? Impossible! (more or less)
- wanted: $d_j = (w_{1j}, \dots, w_{nj})$

• Document frequency

- Remember: infrequent terms are typically more significant than frequent ones
"protein" compared to "interleukin 3"
- Hypothesis: importance of a term depends on number of documents it occurs in
- Justification: Zipf's law
Frequency of an event is inversely proportional to its significance
(Human Behaviour and the Principle of Least effort (G. Zipf 1949))

HS / BioDBS05-2-IR 15

Weights

... Zipf's law

example (see <http://information-retrieval.de/irrb>)

- Consistent to information theory (Shannon) -->

=>

Weight w of term t inverse proportional to document frequency

Document frequency DF of term t :

the number of documents, term t occurs in

Term frequency TF of term t in d :

number of occurrences of t within one document d

HS / BioDBS05-2-IR 16

Information Theory

• Huffman – Code

Given an alphabet $A = \{a_1, \dots, a_n\}$ and probabilities of occurrence $p_i = p(a_i)$ in a text for each a_i .

Find a binary code for A which minimizes

$$H(A) = \sum p_i \cdot \text{length}(cw_i), \quad cw_i = \text{binary codeword of } a_i$$

$$H(A) \text{ is minimized for } \text{length}(cw_i) = \lceil \log_2 1/p_i \rceil$$

well known how to construct it... => intro to algorithms

$H(A) = - \sum p_i \cdot \log_2 p_i$: important characterization of A
what does it mean?

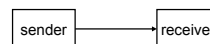
HS / BioDBS05-2-IR 17

Entropy: interpretations

• Entropy

$$H(A) = - \sum p_i \cdot \log_2 p_i$$

- minimal number of bits to encode A



- amount of uncertainty of receiver before seeing an event (a character transmitted)
- amount of surprise when seeing the event
- the amount of information gained after receiving the event.

HS / BioDBS05-2-IR 18

Information Theory and alphabets

• Example

$$L = \{A, C, T, G\}, p(A) = p(C) = p(T) = p(G) = 1/4,$$

Boring: seeing a "T" in a sequence is as interesting as seeing a "G" or seeing an "A".

$$H(L) = -1/4 * \sum \log 1 - \log 4 = 2$$

But:

$$L' = \{A, C, T, G\}, p(A) = 0.7, p(C) = 0.2, p(T) = p(G) = 0.05$$

Seeing a "T" or a "G" is exciting as opposed to "A"

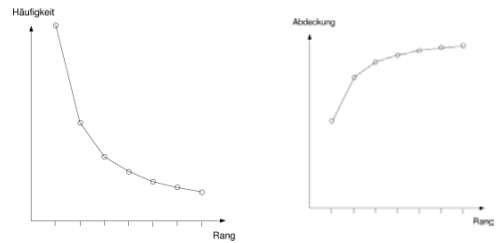
$$H(L') = -(-0.7 * 0.514 - 0.2 * 2.31 - 2 * 0.05 * 4.32) \\ = 0.36 + 0.464 + 0.432 = 1.256$$

Low entropy more interesting

What is the lowest value?

HS / BioDBS05-2-IR 19

Zipf's law



Zipf's law applied to text documents: Frequency and cover of text

from Ferber: Information Retrieval
<http://information-retrieval.de/ir/ir.part.1.html>
 y-coord??

HS / BioDBS05-2-IR 20

Weights

• Term frequency

Base hypothesis of Information Retrieval:

1. Frequency of term t TF characterizes contents of document j

$$TF(t, j) := f_{tj}$$

2. Document frequency characterizes term: $DF(t) := f_t$

inverse document frequency $IDF = 1/f_t$ used in similarity functions

HS / BioDBS05-2-IR 21

TF / IDF

• Normalization

– TF should not be linear (...why?)

normalization heuristics \Rightarrow normalized term frequency r_{tj}

$$\text{e.g. } r_{tj} = 1 + \log f_{tj}$$

$$\text{or } r_{tj} = k + (1-k) f_{tj} / \max_i f_{ij}, \text{ some constant } 0 < k \leq 1$$

– IDF should be independent of number of documents

• normalization heuristics \Rightarrow weight w_t :

$$\text{e.g. } w_t = \log(1 + N/f_t), N = \text{number of documents}$$

$$\text{or } w_t = \log(1 + f_{\max} / f_t)$$

... many other heuristics

HS / BioDBS05-2-IR 22

Weights

• Cumulative weight of term t in document j

$$w_{tj} = g(TF, 1/DF) = g(TF, IDF), \text{ some function } g$$

Weight of term t in document j ("TF / IDF heuristics")

$$\text{Typical: } w_{tj} = r_{tj} * w_t$$

$$\text{i.e. } dj = (w_{t1}, \dots, w_{tn})$$

• Weight of a query term

$$w_{tq} = q_t * w_t$$

q_t = weight relative to query.

Typical: $q_t = 1$ ("All terms equally important")

HS / BioDBS05-2-IR 23

Ranking

Calculating similarity of query q and document d_j using cosine measure

Document d_j sometimes abbreviated as "

$\text{Cos}(d_j, q)$

$$= dj \cdot q / |dj| * |q|$$

$$= \sum_{t \in dj \cap q} w_{tj} * w_t / \sqrt{(\sum w_{tj}^2)} * \sqrt{(\sum w_{tq}^2)}$$

$$t \in dj \cap q$$

$$= 1 / (W_j * W_q) * \sum (1 + \log f_{tj}) * (\log(1 + N/f_t))^2$$

$$\text{where } W_j = \sqrt{(\sum w_{tj}^2)}, W_q = \sqrt{(\sum w_{tq}^2)} = \sqrt{(\sum w_t^2)}$$

Note: document frequency has double influence - counts in d_j as well as q . Reasonable?

HS / BioDBS05-2-IR 24

Ranking

Most often used for ranking of document / query similarity:

$$\text{Cos}(d_j, q) = 1 / (W_j * W_q) * \sum_{t \in d_j \cap q} (1 + \log f_{tj}) * \log(1 + N/f_t)$$

Rank for each document in document set D
 ==> Ranking of result set

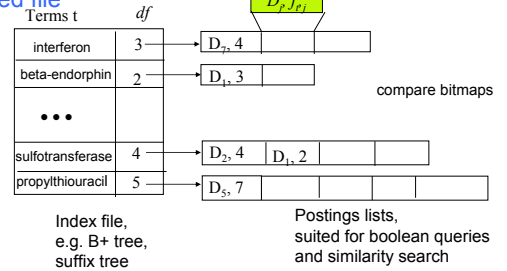
Issues:

- Efficient implementation
- Evaluation of "retrieval effectiveness"
- Many more similarity measures.
- Specific measures for Web documents (e.g. Google: "page rank")
- Domain specific measures

HS / BioDBS05-2-IR 25

2.3.3 Implementation of vector space model

Inverted file



Most values can be calculated before processing a query and put into the posting list, e.g. $1 + \log f_{tj}$

HS / BioDBS05-2-IR 26

2.4 Evaluation : recall / precision

• Issues

- Subjectiveness of judgement
 How relevant is a document with respect to a query?
- Elaborate, costly empirical tests required
 many queries, many individual judgements for each query, mean of judgements?

• Evaluation model

- Ideal observer: knows relevant documents for each query
- Check for each query q
 - how many relevant documents found
 - how many irrelevant documents found
- Calculate mean over many queries

HS / BioDBS05-2-IR 27

Evaluation

Recall:

fraction of *relevant* documents *found* out of all *relevant* documents
 $R = r / (r + v)$

Precision:

fraction of *relevant* documents *found* out of all documents *found*
 $P = r / (r + n)$

F-Measure:

$$F = 2P * R / (P + R)$$

Noise:?

Relevant objects found should occur as soon as possible in Output set.

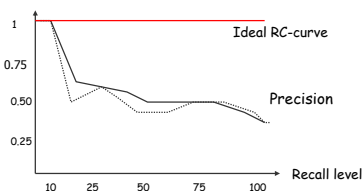
How to evaluate ranking order?

HS / BioDBS05-2-IR 28

Evaluation

Recall level	Precision 25	Precision 50
1 ✓	10	100
2	10	50
3	10	33
4 ✓	20	50
5 ✓	30	60
6	30	50
7 ✓	40	57
8	40	50
9	40	44
10	40	40
11	40	36
12 ✓	50	42
13 ✓	60	46
14 ✓	70	50
15	70	47
16 ✓	80	50
17	80	47
18	80	44
19 ✓	90	47
20	90	45
21	90	43
22 ✓	100	45
23	100	43
24	100	42
25	100	40

Recall-Precision Graph



Recall level n:
 n % of all relevant Documents have been found

RC curve:
 Precision at recall level n

for a particular query and document set!

HS / BioDBS05-2-IR 29

2.5 Molecular biology applications

• TREC competition

- on **retrieval of publications** in MEDLINE concerning the function of a gene
 ⇨ GeneRIF ("reference into function")

Query example:

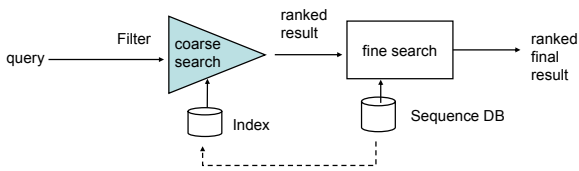
For gene X, find all MEDLINE references that focus on the basic biology of the gene or its protein products from the designated organism. Basic biology includes isolation, structure, genetics and function of genes/proteins in normal and disease states.

details see [paper](#)

HS / BioDBS05-2-IR 30

Molecular biology applications

Does indexing help to reduce search time for similar sequences ?



see: Aung et al: Rapid 3D protein structure Database searching using information retrieval techniques. *Bioinformatics* 20(7), 2004 pp1945-1952
 see: William, Zobel: Indexing and Retrieval for Genomic Databases, 2002

HS / BioDBS05-2-IR 31

Indexing sequences

• Intervals

Insulin - Goose
 AANQHLCGSHLVEALYLVCGERGFFYSPKT/GIVEQCENPCSLYQLENYCN*
 >P1;INAQ
 Insulin - American alligator
 AANQRLCGSHLVDALYLVCGERGFFYSPKG/GIVEQCCHNTCSLYQLENYCN*
 [] ... [] ...

n length of each interval (n -grams) ,
 k length of sequence \Rightarrow k-n+1 intervals

Example: acctgtc , n=3 : acc, cct, ctg, tgt, and gtc.

Construct inverted index for interval occurrence in the sequences of DB

correspondence: Document \Leftrightarrow sequence, word (term) \Leftrightarrow interval
 HS / BioDBS05-2-IR 32

Inverted genomic index

Posting list:

<interval> [seqNo, (noOfMatches: positions)]*

e.g. GAGA -> 52,(3: 147,233,256), 83,(2: 17, 256)
 means: GAGA occurs 2 times in sequence 52 at positions 147, 233, 256 and 2 times in 83...

Coarse homology search of sequence s in the DB:

- Cut s into intervals
- locate matching sequences in DB and the matches of intervals of s
- rank the matching sequences
- fine search (e.g. alignment) on sequences

HS / BioDBS05-2-IR 33

Ranking

"Frame-based" ranking (Williams / Zobel),

Frame= set of matching intervals with the same offset

```

A      10      20      30      40      50
ACCTGAGGTTTTTTTGGGAGAGCTTCTCTTAGAGAGGAGGCTAGCTAGCTTCG
      ::::      :::::
      GTGTGTTTTGTGTGGGGTAAGTCTCTCTTT
          10      20      30
B      10      20      30      40      50
ACCTGAGGTTTTTTTGGGAGAGCTTCTCTTAGAGAGGAGGCTAGCTAGCTTCG
      ::::
      GTGTGTTTTGTGTGGGGTAAGTCTCTCTTT
          10      20      30
C      10      20      30      40      50
ACCTGAGGTTTTTTTGGGAGAGCTTCTCTTAGAGAGGAGGCTAGCTAGCTTCG
      ::::
      GTGTGTTTTGTGTGGGGTAAGTCTCTCTTT
          10      20      30
  
```

Frame F2 of A and search sequence: matching offset for interval GTT: 2
 Matches of F2 with offset 2: (9,7), (10,8),(27,25),(28,26),(29,27),(30,28),(31,29), (32,30)

Frame F26 of C and search sequence: matching offset for interval GTT: 2
 Matches of F2 with offset 26: (53,27), (54,28)

HS / BioDBS05-2-IR 34

Ranking

Frame F2 of A and search sequence: matching offset for interval GTTT: 2
 Frame with offset 2: (9,7), (10,8),(27,25),(28,26),(29,27),(30,28),(31,29), (32,30)

Frame F26 of C and search sequence: matching offset for interval GTTT: 2
 Frame with offset 26: (53,27), (54,28)

Simple ranking scheme :

Rank s, t according to number of intervals in each frame for s and t.*

$\text{rank}(s,t) = \max (|F(s) \cap I(t)|)$ I(x) = number of intervals in x, F(): returns sets of intervals at the same offset.

How effective? Time, precision, recall, storage space?

* May be improved, not important in this context, see paper

HS / BioDBS05-2-IR 35

Evaluation: P/R Graph

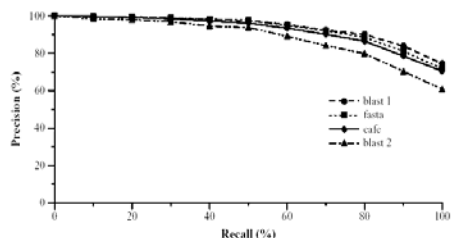


Figure 8: Mean recall-precision for both versions of BLAST, FASTA, and CAFÉ. The PIRSF collection is used, with 1,834 amino-acid queries. We parameterise CAFÉ to use a banded local alignment fine search (similar to that used in FASTA) and the NEIGHBOURHOOD frames ranking metric. All systems use a PAM-250 matrix scoring matrix.

cf. from Williams / Zobel

HS / BioDBS05-2-IR 36

Evaluation: Time

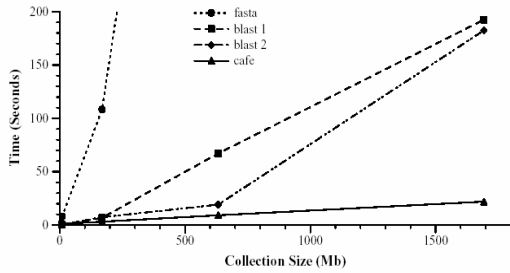


Figure 9: Plot of the time for BLAST, CAFE, and FASTA to search the nucleotide collections GBMAM, VERTE, GENBANK97, and GENBANK108, averaged over 41 queries.

cf. from Williams / Zobel

HS / BioDBS05-2-IR 37

Evaluation: Space

Collection	Inverted lists (Mb)	Other structures (Mb)
GBMAM	25.4	3.3
VERTE	427.7	3.3
GENBANK97	1420.4	3.3
GENBANK108	3082.6	3.3
PIRSF	23.0	0.1

Table 1: CAFE index size for the nucleotide collections, GENBANK108, GENBANK97, VERTE, and GBMAM and the amino-acid collection PIRSF. For nucleotide searching, CAFE has an interval length of $n = 9$ and, for amino-acid searching, an interval length of $n = 3$.

Index size

Genbank: factor of 2.2 compressed
factor > 10 uncompressed

HS / BioDBS05-2-IR 38

Summary

- Information retrieval:
 - essential for text retrieval
 - in molecular biology in particular
 - who published the data when? which experiments?...
 - Indexing seems to be an alternative for a coarse searching step
 - Ranking functions: domain specific heuristics

HS / BioDBS05-2-IR 39