Sequence Database Search Using Jumping Alignments

Constantin Bannert[1], Marc Rehmsmeier[2], Rainer Spang[1], and Jens Stoye[1]

Please visit http://jali.molgen.mpg.de for further information.

[1] Computational Molecular Biology, Max-Planck-Institut für Molekulare Genetik
Ihnestraße 73, D-14195 Berlin, Germany ({bannert|spang|stoye}@molgen.mpg.de)

[2] Theoretical Bioinformatics, Deutsches Krebsforschungszentrum
Im Neuenheimer Feld 280, D-69120 Heidelberg, Germany (M.Rehmsmeier@dkfz.de)

Overview

We present an algorithm for amino acid sequence classification and the detection of remote homologues. The rationale is to exploit *vertical and horizontal* information of a multiple alignment in a well balanced manner. Established methods like profiles and hidden Markov models (HMMs) focus on vertical information only, summarizing the individual columns of the multiple alignment. We also take into account row dependencies.
Our setting is the following: For a given, uncharacterized protein "candidate sequence", we want to find from a database of protein families that family where this sequence is likely to belong to. For each protein family from the database, a multiple alignment is constructed. The candidate sequence is then tested against this alignment by means of a new Jumping Alignment algorithm (Jali). It computes a local alignment of the candidate sequence and the protein family alignment. The method is published in [5].

```
SequenceNo  8888888888888884444444444422222222222222222222222222222
d1jae 2 3.  NEYLVADGRPWNERVQPVSYIINTRSGD-------ESAFTDRTRRCRDAGVRIYVDAVIN
d1edt       TALRAANPDKIISLYN-------------------IGPAASR-LSYGGVDVSDKFDYAWN
d2ebn       YETKQAMPNKLVTVYV-------------------YSRTSSFPTAVDGVNAGSYVDYAIN
d1bgga      FLQPIYQGSYPQFLVD----WFAEQGATVPIQD---GDMDIGEPIDMIGINYYS-HSVN
d1p0ga      ILDATYLGHYSDKTMEGVNHILAENGGELDLRDEDFQALDAAKDLNDFLGINYYMSDWMQ
d2myf       FMGPLTNGTYPQIMID-------TVGARLPTFSP---ETNLVKGSYDFLGLNYYFTQYAQ
d1taxa      NLDSATYPKTQA-IVN-----------------RVKQWRAAGVPIDGIGSQTHLSAQQG
d1xyza      NIEDLG-PKSNA-VFN-----------------MIKSMKERGVPIDGVGFQCHFINGMS
d1b9za2     NGYLSMYGKDYLEKYQG----------------ILENHTKLIGELAHNAAFDTTFQVPIG
d1byb       GTYVTEKGKFFLTWYSN----------------KLLNHGDQILDEANKAFLGCKVKLAIK
d1nar       IAPSENNSSHYQKLYN-----------------AKKDYIMWVDYQFSNQQKPVSTD
d1cnv       LLSAAPGCLSPDEYLDN----------------AIQTRHFDYIFVRFYNDKSC
d1edqa2     LMKELRAMLDQLSVETG----------------RKYELTSAISAGKDKIDKVAYNVAQNS
d1exia1     GLMNIHMPAYKNAMDKG----------------VSTVMISYSSWNGVKMHANQDLV
```

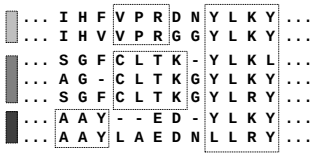
Excerpt from a jumping alignment of alpha-amylase from the yellow mealworm (*Tenebrio molitor*, blue) with 13 other sequences from the same SCOP superfamily (3.1.8; glycosidases).

Motivation

The characterization of new sequence data produced by the genome projects requires fast and easy methods of functional annotation. Most proteins with similar function descend from common ancestors. The common function is reflected by conserved regions at the sequence level.
Given an uncharacterized *candidate* protein sequence, the question is whether this sequence fits into one of the known families. The following figure shows a possible local alignment of a candidate sequence (below the dashed line) to a multiple alignment.

```
... I H F V P R D N Y L K Y ...
... I H V V P R G G Y L K Y ...
... S G F C L T K - Y L K L ...
... A G - C L T K G Y L K Y ...
... S G F C L T K G Y L R Y ...
... A A Y - - E D - Y L K Y ...
... A A Y L A E D N L L R Y ...
-----------------------------
... S H Y C P E K N L I R A ...
```

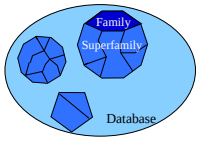
A purely *vertical* view suggests a good fit of the candidate sequence to the family, however, none of the individual sequences is very similar to the candidate sequence. The next figure shows that there is more information contained in the alignment.



The 7 sequences subdivide into 3 subfamilies (shaded bars) with certain conservation patterns (dotted boxes). These patterns are not seen by column-based approaches. To address these shortcomings we developed Jali.

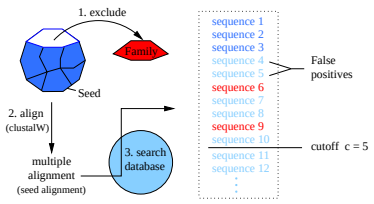
Evaluation

We evaluate our method using the SCOP database [2]. It contains classified protein domains. *Superfamilies* are defined as sets of homologues sequences. Each superfamily consists of one or more less divergent *families*.



Setting

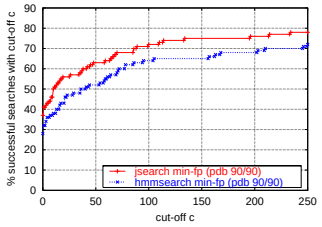
Our evaluation procedure was first described by [3]. From a SCOP superfamily, one family is split off, called the *excluded subfamily*. The remaining sequences form the *seed*, which is used to construct a multiple alignment. The database is then searched for members of the excluded subfamily.

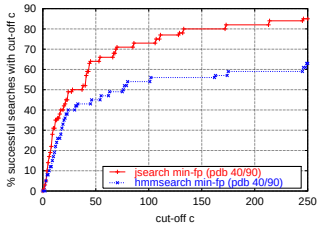


A large number of false positives is a common observation, and it is realistic to define a limit. We speak of an FP-count with cutoff c. If the FP-count is lower than the cutoff, the search is considered *successful*.

Results

Previous evaluations using an older version of SCOP indicated that the performance of Jali is at least able to reach that of a popular implementation of HMMs, HMMER [1]. Our new results on SCOP 1.53 underpin our earlier findings.



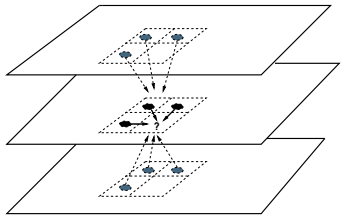


The evaluation was performed with *Phase4*, an automatic evaluation environment for database search methods [4].

Algorithm and Implementation

Algorithm

The algorithm is based on the dynamic programming paradigm. It can be viewed as an extension of the Smith-Waterman algorithm for aligning pairs of sequences. For every candidate sequence we compute the optimal local *jumping alignment score*, where hopping between the sequences of the seed alignment is penalized.
Instead of one edit matrix D, our method employs K edit matrices $D_1, \ldots, D_K$, one for each alignment row. For $1 \leq k \leq K$, $D_k(i,j)$ holds the maximal score of all alignments between the candidate sequence and the seed which end with positions s_i and $A_{k,j}$. The computation of cell $D_k(i,j)$ requires to consider (and maximize over) $3K$ predecessor cells.



A direct implementation leads to an $O(nmK^2)$ time algorithm. The time complexity can be reduced to $O(nmK)$, whereas space usage remains $O(nmK)$ (see [5] for details).

Implementation

We have implemented the above algorithm including affine gap costs in two programs. The first one (`JALI`) calculates a 'jumping alignment' with the corresponding score. The second program (`JSEARCH`) uses the Jali algorithm in a database search context. Both programs are written in standard C and have been compiled on several UNIX platforms. The programs are available for academic users free of charge from `http://jali.molgen.mpg.de`.

Future projects

We expect to improve the discriminatory power of Jali by using secondary structure information in the jumping strategy. We are also planning to set up a Jali WWW server.

Acknowledgements

Martin Vingron had some of the initial ideas for this work. We owe him the distinction between the horizontal and the vertical aspect of a multiple alignment.

References

[1] S. Eddy. Profile hidden Markov models. *Bioinformatics*, 14:755–763, 1998.

[2] T. Hubbard, B. Ailey, S. Brenner, A. Murzin, and C. Chothia. SCOP: A Structural Classification of Proteins database. *Nucl. Acids. Res.*, 27:254–256, 1999.

[3] T. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. *J. Comp. Biol.*, 7:95–114, 2000.

[4] M. Rehmsmeier. Phase4 — automatic evaluation of database search methods. In preparation.

[5] R. Spang, M. Rehmsmeier, and J. Stoye. Sequence database search using jumping alignments. In *Proc. of the Eighth International Conference on Intelligent Systems for Molecular Biology, ISMB 00*, pages 367–375, Menlo Park, CA, 2000. AAAI Press.