Using Secondary Structure to Perform Multiple Alignment

Giuliano Armano [1], Luciano Milanesi [2], Alessandro Orro [1]

[1] DIEE - University of Cagliari, Cagliari, Italy
{armano,orro}@diee.unica.it

[2] ITB-CNR Milano, Italy
luciano.milanesi@itb.cnr.it

Outline of the talk

- Introduction
  - – The Multiple Alignment Problem
  - – Related works
- The proposed solution
  - – Architecture (Abstraction)
  - – Algorithm
- Experimental results
- Concluding remarks

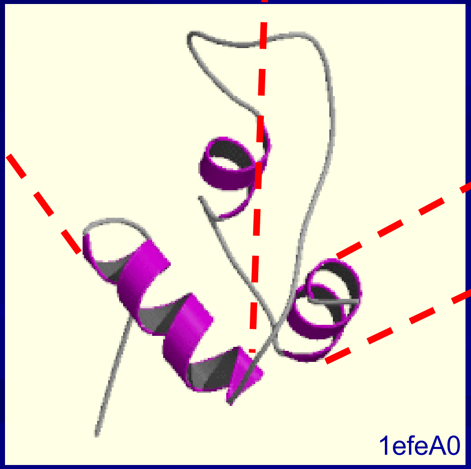
Multiple Alignment

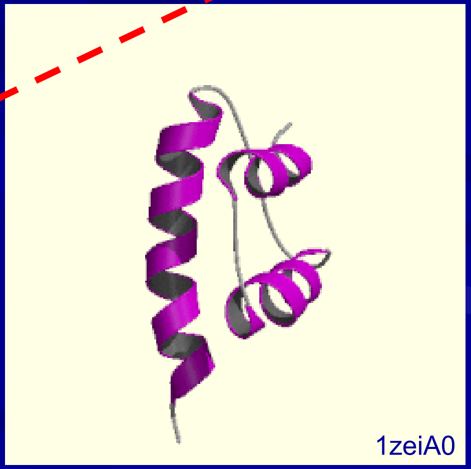
- Sequence comparison is one of the most important bioinformatics tasks
- Applications:
  - structural similarity ↔ similar functionality
  - infer sequence homology
  - family membership checking

Multiple Alignment

- Example: two sequences with similar structure and function

```
cHHHHHHHHHHHHcccccccccccccccccccccHHHHHHHccc ...
|||||||||||||||||||||            |||||||||||
HHHHHHHHHHHHHcccccccc-------cccccHHHHHHHHccc ...
```


1efeA0


1zeiA0

Multiple Alignment

- Multiple Alignment
  - – a sistematic approach to multiple sequence comparison
  - – find the configuration that best represents the relations amongst sequences
  - – rappresents relations in terms of
    - insertion
    - deletion
    - match/substitution

Multiple Alignment

- Example

```
. . .  A  R  L  D  K  P  K     . . .  target
. . .  A  R  -  D  K  P  K     . . .  deletion
. . .  A  R  D  D  K  P  K     . . .  mutation
. . .  A  R  L  V  D  K  P  K  . . .  insertion
```

Multiple Alignment

- Example: 5 sequences arranged in a multiple alignment showing the conserved residues (core blocks)

```
GLVYQVVEAGKGEA.PKDSDTVVVNYKGTLID.GKEFDNSYT.......
...IKRIPVEDCLIKAMPGDKVKVHYTGSLLESGTVFDSSYS.......
GLQFRVINQGEGAI.PARTDRVRVHYTGKLID.GTVFDSSVA.......
....SVLKKGDKTNFPKKGDVVHCWYTGTLQD.GTVFDTNIQTSSKKKK
.LQYRVVKEGTGRV.LSGKPTALLHYTGSFID.GKVFDSSEK.......
```

Multiple Alignment

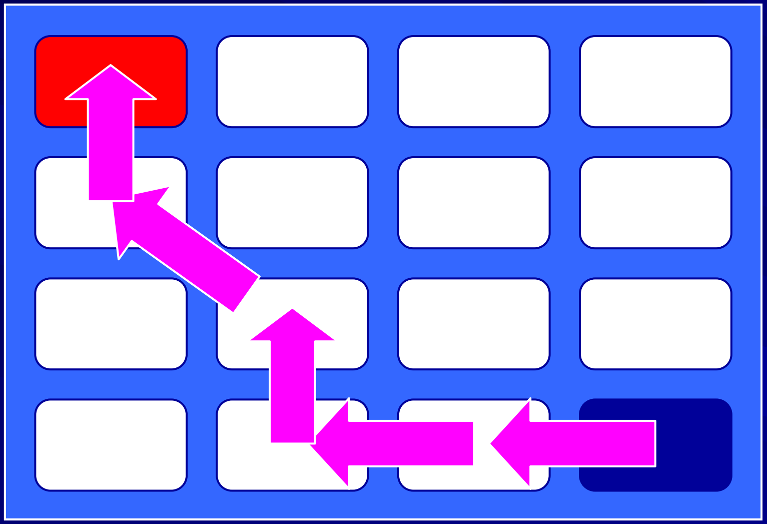
- Two crucial issues:
  - score model
    what function should be maximized to obtain the optimal alignment?
  - optimization algorithm
    what optimization technique should be used?

Proposed Solution

- We propose an abstraction-based strategy that exploits secondary structure information to perform multiple alignment
- Implementing abstraction allows to mimic the human ability of simplifying a problem by disregarding, at different levels of granularity, some details deemed irrelevant

Abstraction

- Example of search algorithm with abstraction

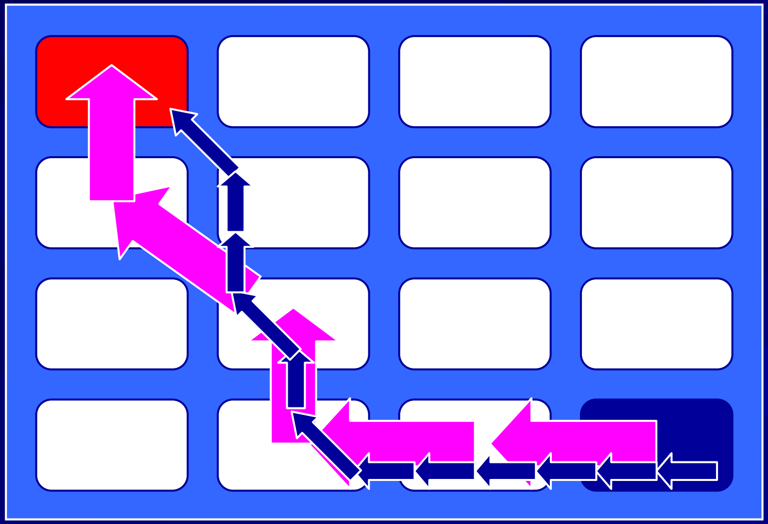


Search space at abstract level

1. start from a ground representation setting
2. build an abstract representation
3. find a solution (path) at the abstract level

Abstraction

- Example of search algorithm with abstraction



Search space at abstract level

1. start from a ground representation setting
2. build an abstract representation
3. find a solution (path) at the abstract level
4. return to the ground level
5. refine the solution at the ground level
6. iterate

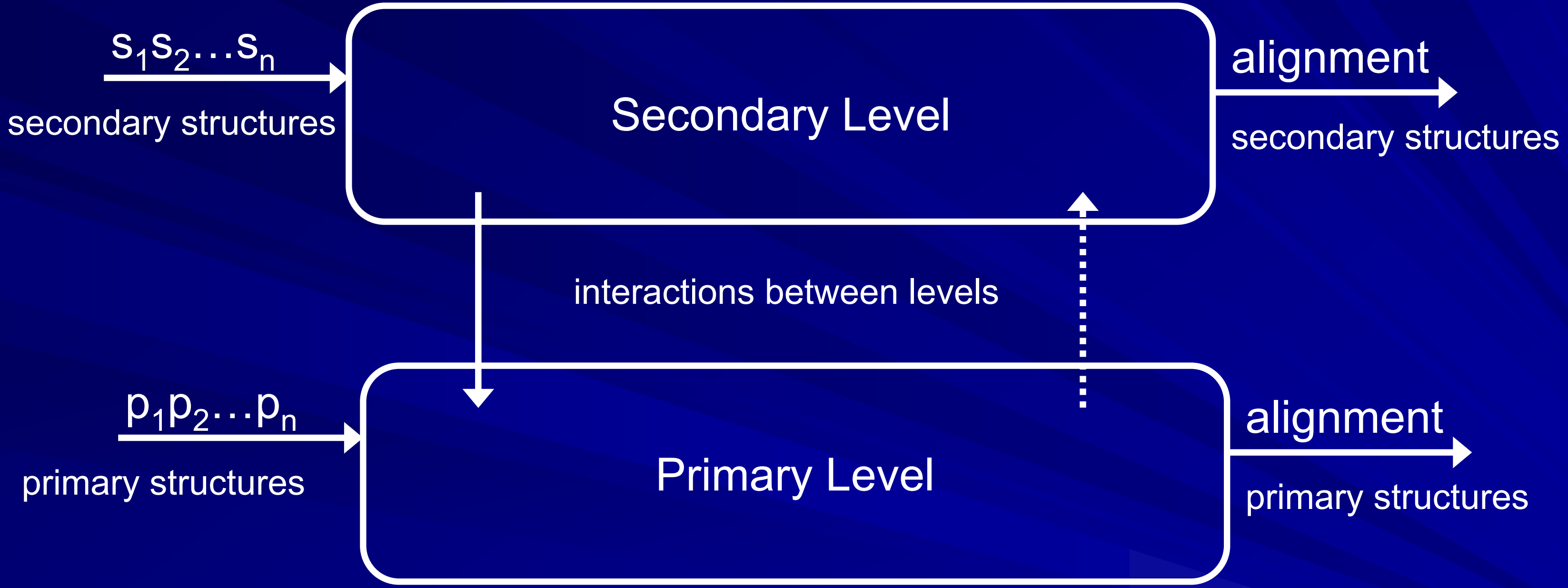
Proposed Solution

- Using abstraction in the multiple alignment problem
  - the abstract level is the domain of secondary structure elements
  - the abstract search is performed by aligning secondary structure elements (alpha-helix, beta-sheet, coil)
  - the ground search is performed by locally optimizating the alignment

Proposed Solution

- Why using the secondary structure?
  - it is more conserved with respect to the corresponding aminoacid sequence
  - it is a simple description
  - a secondary structure alignment is a good starting point for computing the final alignment

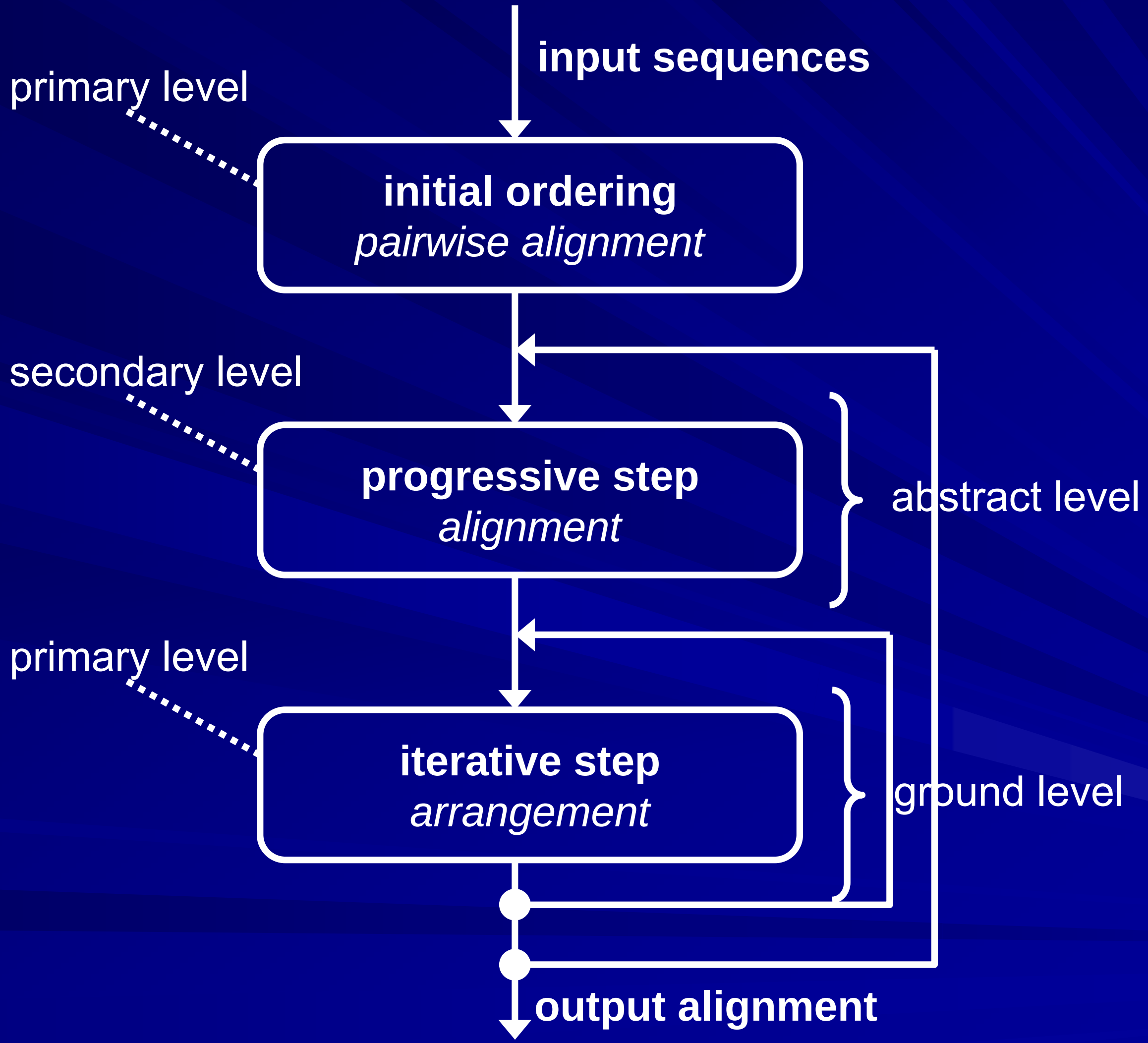
System Architecture



Algorithm

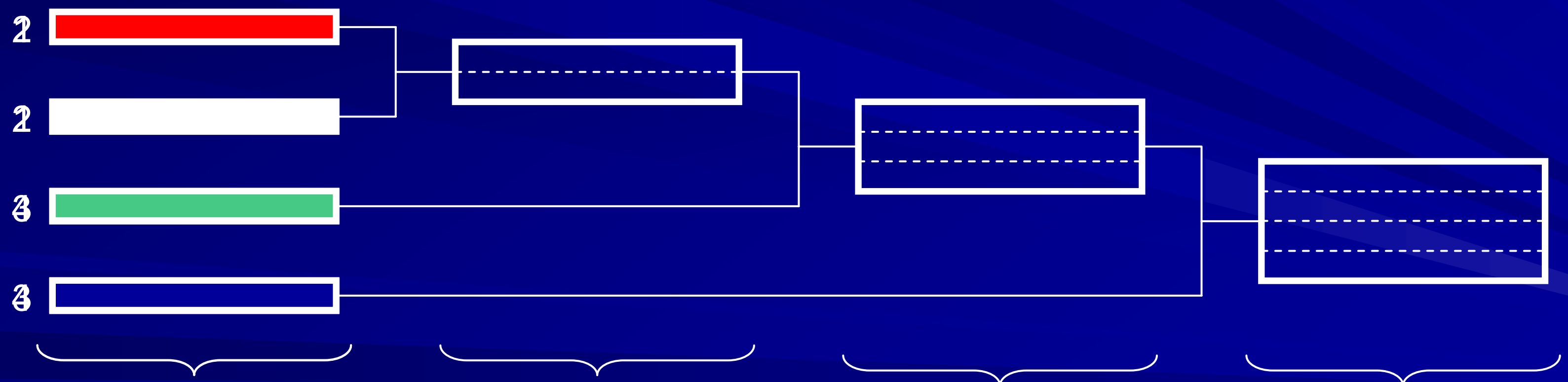
- Ground level
  - deals with primary structure according to an iterative technique
- Abstract level
  - deals with secondary structure according to a progressive technique

Algorithm


input sequences
primary level
initial ordering
pairwise alignment
secondary level
progressive step
alignment
abstract level
primary level
iterative step
arrangement
ground level
output alignment

Algorithm

- initial ordering
  - – using pairwise dynamic programming, establish the ranking to be followed while progressively embodying sequences in the alignment



input sequences / ordered sequences

partial alignment

partial alignment

final alignment

Abstract Level

- each sequence is added to the alignment using dynamic programming
- score model for secondary structures
  - substitution matrix
  - gap opening / gap extension penalties
  - cost for breaking secondary structure elements

Abstract Level

- Substitution Matrix (secondary structure)

	G	H	T	B	E	S	C
G	7.9	-0.8	2.0	-1.5	-8.5	0.2	-1.2
H	-0.8	2.3	-1.7	-8.2	-18.8	-5.0	-6.2
T	2.0	-1.7	5.4	-2.2	-5.0	-1.9	-1.1
B	-1.5	-8.2	-2.2	10.0	0.8	1.1	1.5
E	-8.5	-18.8	-5.0	0.8	3.1	-2.4	-0.9
S	0.2	-5.0	-1.9	1.1	-2.4	5.5	1.8
C	-1.2	-6.2	-1.1	1.5	-0.9	1.8	4.0

Ground Level

- the alignment is refined by local operators that rearrange gap positions
- if the alignment performed at the secondary level is close to the “true” solution, local operators can easily reach it
- standard score model for primary structures
  - substitution matrix (BLOSUM80)
  - gap opening / gap extension penalty

Ground Level

- Locals operators
  - limited range
  - so far, only gap moving is allowed (no gaps are added/removed to the alignment)
  - sub-optimal results

Ground Level

- example: rearranging the primary structure

```
 ..QQRLIFA...................GKQLEDGR.TLSDYNIQKESTLHLVLRLRGG. GIPPD.
 ..CAVFRLLHEH...............KGKKARLDWNTDAASLIG..EELQVDFL..... GLQPEC
CGACSVILD....................GKVVR...ACVTKMKRVADGAQITTIEGVGQ. GVKVGC
CSSCAGKVESGEVDQSDQSFLDDAQM...GKGFVL..TCVAYPT...SDVTILTHQEAALY .LPYSC
```

```
.GKQLEDGR.
KGKKARLDW.
.GKVVR....
.GKGFVL...
```

Best configuration

Experimental Results

- Dataset BAliBASE
- Quality measures
- Programs used for assessing experimental results: prrp, clustal, saga, dialign, pima, multialign, pileup8, multal, hmmt, tcoffee
- Using the RASCAL optimizer

Experimental Results

- BAliBASE
  is aimed at testing different features of multiple alignment programs:
  - ref1: equidistant sequences, without large extensions or insertions
  - ref2: strictly-related sequences, together with some added "orphans"
  - ref3: sequences taken from a limited number of different families (up to four)
  - ref4: sequences with long N/C terminal gaps
  - ref5: sequences with long internal gaps

Experimental Results

- Alignment quality is computed comparing the alignment with the correct one (using the bali_score program)
  - SP (sum-of-pairs)
    percent of residue pairs correctly aligned
  - CS (column score)
    percent of columns correctly aligned

Experimental Results

Preliminary results

	ref1 (sp)	ref2 (sp)	ref3 (cs)	ref4 (cs)	ref5 (cs)
PRRP	87,63	54,06	53,24	32,25	70,01
ClustalW	86,42	58,33	44,65	36,11	70,48
SAGA	84,14	58,63	50,55	28,88	64,18
DIALIGN	78,76	38,44	31,45	85,25	83,64
SB_PIMA	82,15	37,91	26,69	79,38	50,84
ML_PIMA	80,99	37,08	37,15	70,54	57,23
MULTALN	83,38	51,74	30,29	29,22	62,71
PILEUP8	83,21	42,87	32,31	71,30	63,89
MULTAL	76,27				
HMMT	48,68	40,10			
TCOFFEE	86,23	85,02	47,66	69,39	89,58
A3	**87,22**	**84,46**	**35,32**	**74,08**	**61,05**

Experimental Results

- When A3 performs better?
  - divergent proteins
  - weak signal of homology
  - adequate amount of secondary structure information
- Can A3 output be further improved?
  - we used the RASCAL post-processing tool

Experimental results

- Results with the RASCAL optimizer, and selecting only sequences according to the rule:

ID < 50% and SEC>30%

	CS score
TCOFFEE	63.03
ClustalW	61.05
A3	67.29

Experimental results

- Results with the RASCAL optimizer, and selecting only sequences according to the rule:

ID < 55% and SEC>10%

	CS score
TCOFFEE	60.75
ClustalW	58.12
A3	62.92

Conclusions

- Conclusions
  - algorithms based on abstraction can be successfully used to perform protein alignment
  - encoding a sequence using secondary structure information appears to be a natural choice for implementing abstraction techniques in this particular research field