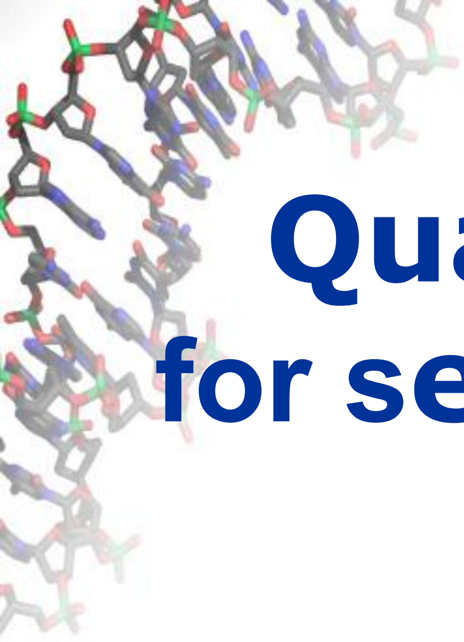


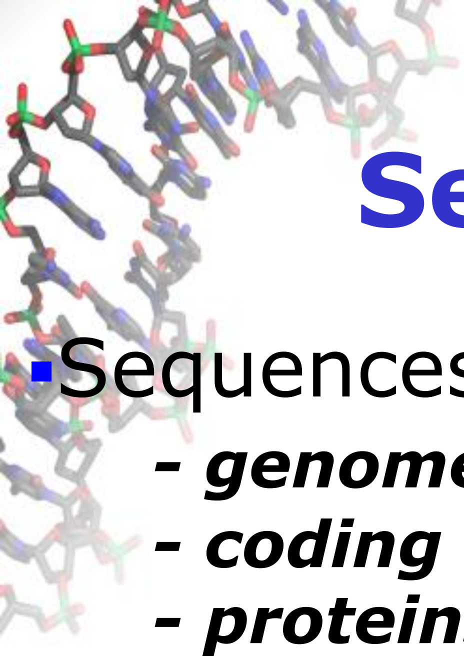
Quality of algorithms for sequence comparison

Mikhail Roytberg

PReMI 2011
Moscow - June 30, 2011

Plan

- ***Introduction***
    - ***- Types of sequence comparison***
    - ***- Interpretation of the results***
- *Local similarity search*
    - - Seed-based methods
    - - Sensitivity and selectivity
    - - Seed models
- *Alignment of homologous sequences*
    - - What is “true” alignment
    - - Accuracy and Confidence
    - - The origin of difference
- *Conclusion*
    - - Be prepared!

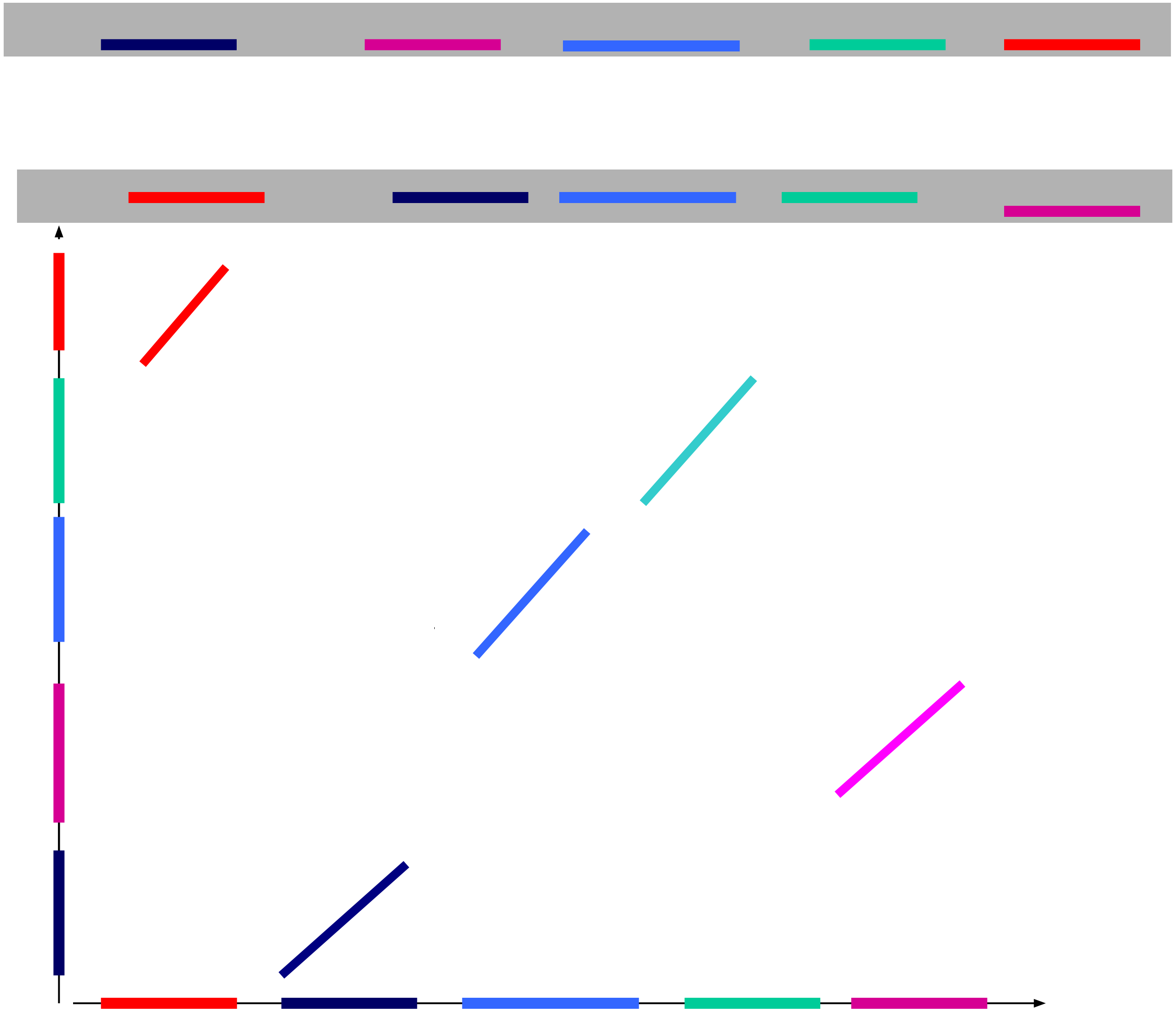


Sequence comparison

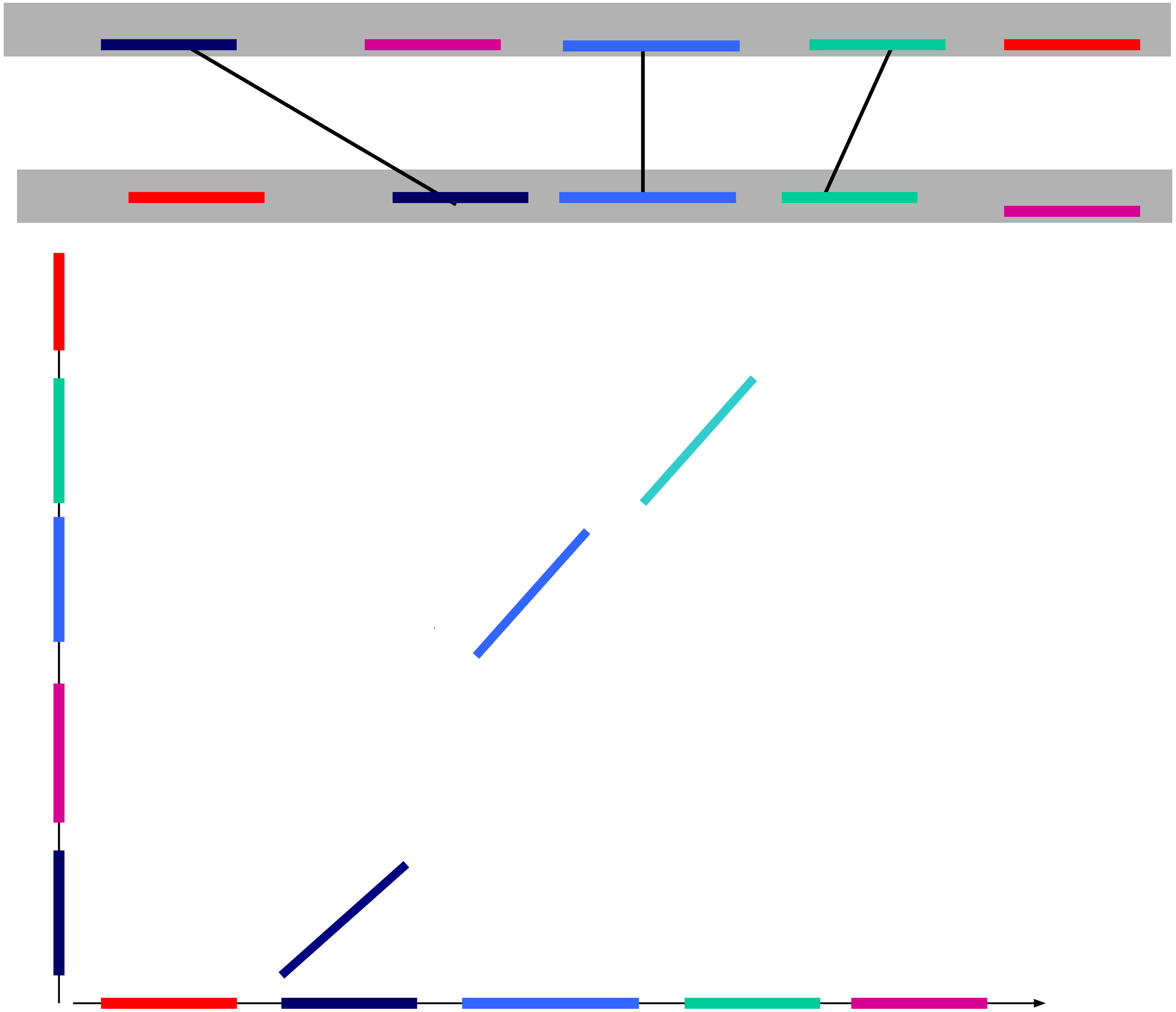
- Sequences: Length:
  - ***- genomes 10^8***
  - ***- coding regions (genes) 10^3***
  - ***- proteins 10^3***

- Similarities:
  - ***- one global*** *[proteins, genome fragments]*
  - ***- chain of non-conflicting local*** *[proteins, genomes]*
  - ***- all local*** *[protein DB, genomes]*

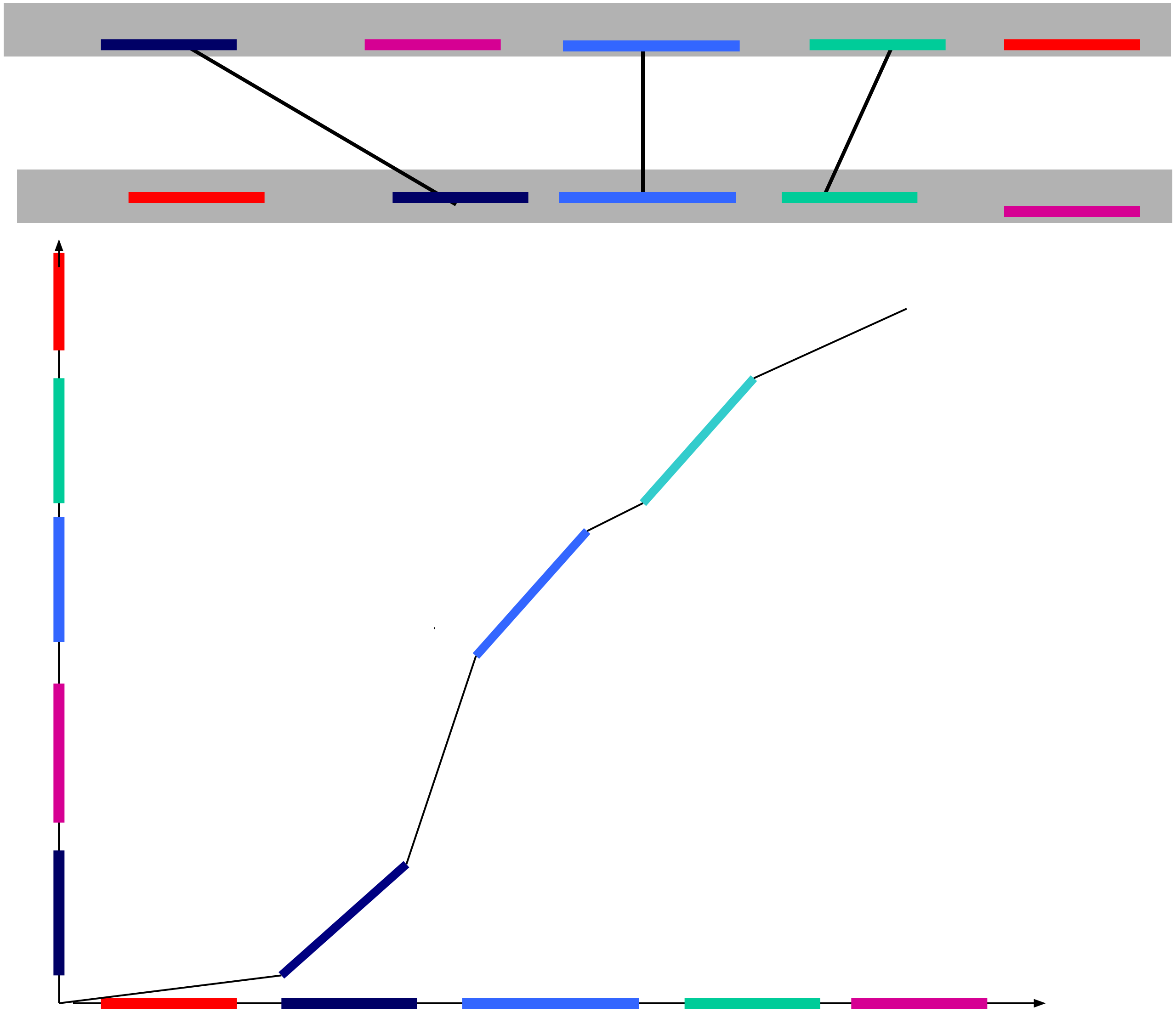
Local similarities

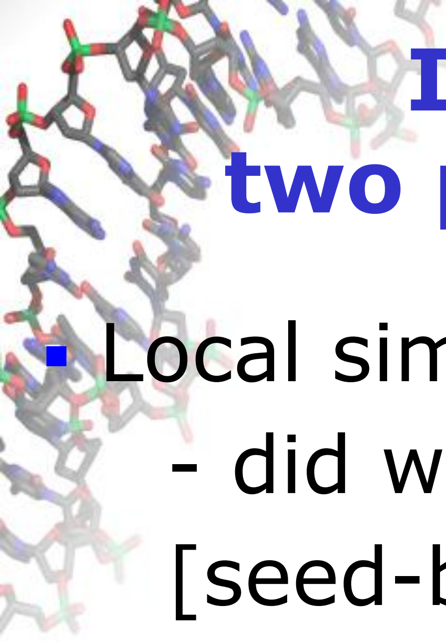


Chain of non-conflicting local similarities



Global alignment based on Local similarities

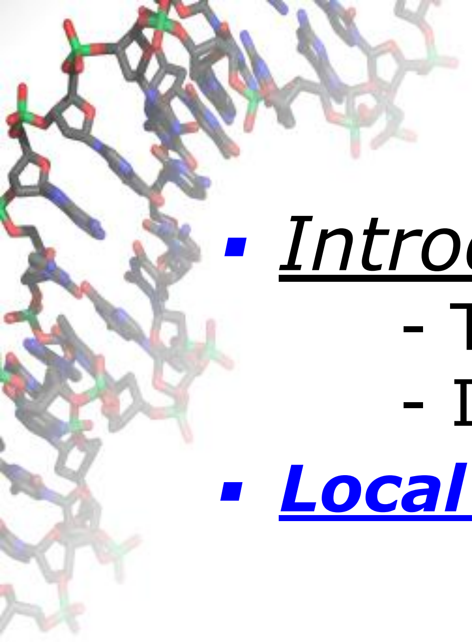




Interpretation of results: two problems and two questions

- Local similarities search:
  - did we find all similarities?
  [seed-based algorithms]

- [Global] Alignment of similar fragments:
  - is the alignment “evolutionary true”?
  [seed-based or DP algorithms]

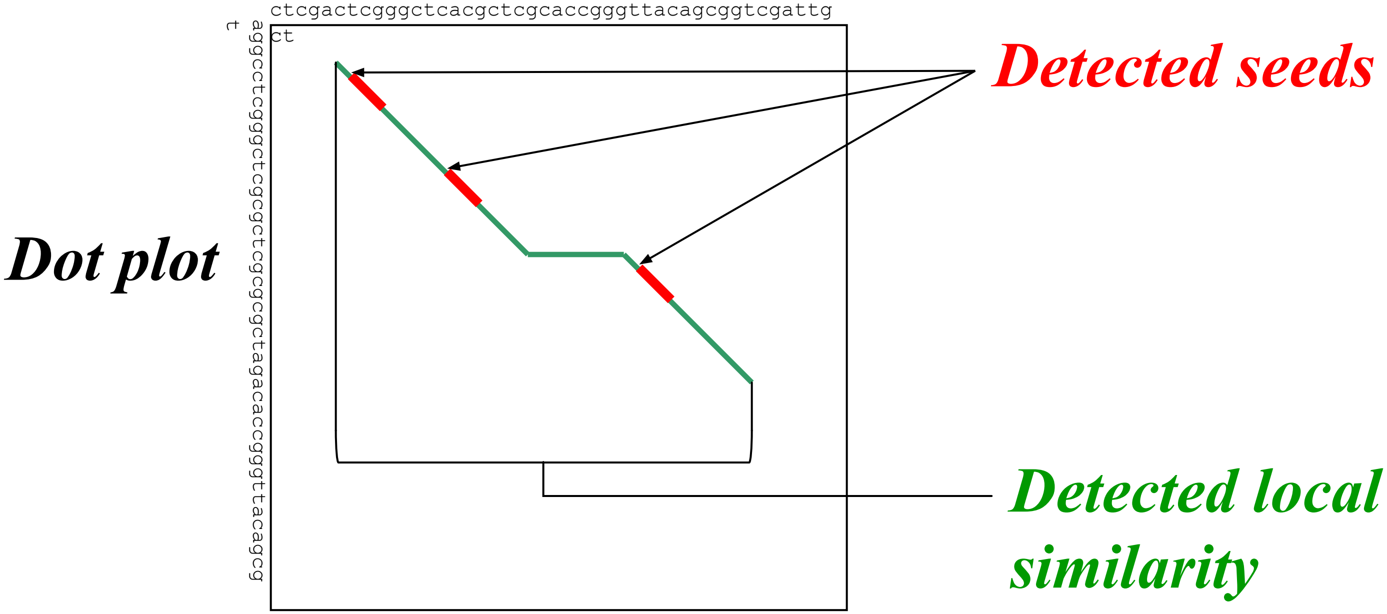


Plan

- *Introduction*
  - Types of sequence comparison
  - Interpretation of the results
- ***Local similarity search***
  - ***Seed-based methods***
  - ***Sensitivity and selectivity***
  - ***Seed models***
- *Alignment of homologous sequences*
  - What is “true” alignment
  - Accuracy and Confidence
  - The origin of difference
- *Conclusion*
  - Be prepared!

Seed-based filtering

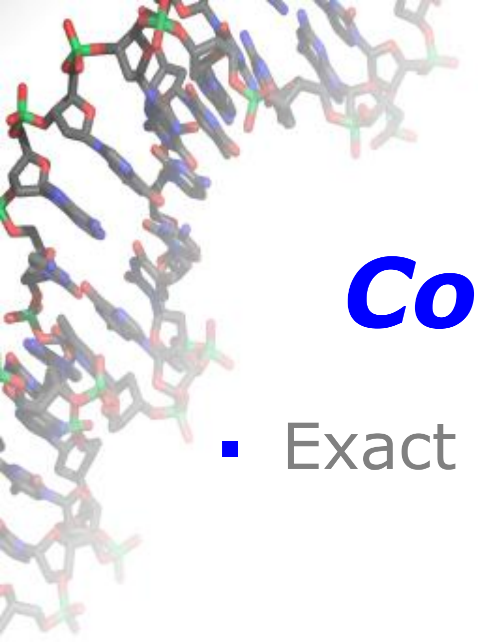
- Start with small conserved and easily detected similar fragments (*seed similarities*).
- One or several seeds, considered to be a witness a potential local similarity, a trigger to build the alignment of the similar fragments



Detected seeds

Dot plot

Detected local similarity



Example: *Contiguous* seed [BLAST]

- Exact similarity :ATCAGT

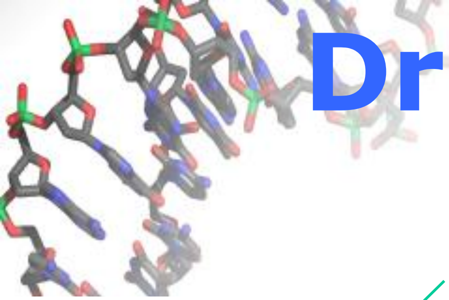
```
ATCAGT
||||||
ATCAGT
```

Seed Pattern : ######

Weight **: 6** [number of #]

- **Example : 16 matches of 20**

```
      ######
ATCAGTGCAATGCTCATGAA
|||.|.||||||:||.|||
ATCGGCGCAATGCGCAAGAA
```



Drawbacks of filtering

`##☹### [16 of 20!]`

`ATCAGTGCGATGCTCATGAA`

`|||.|||||:|||:||.|||`

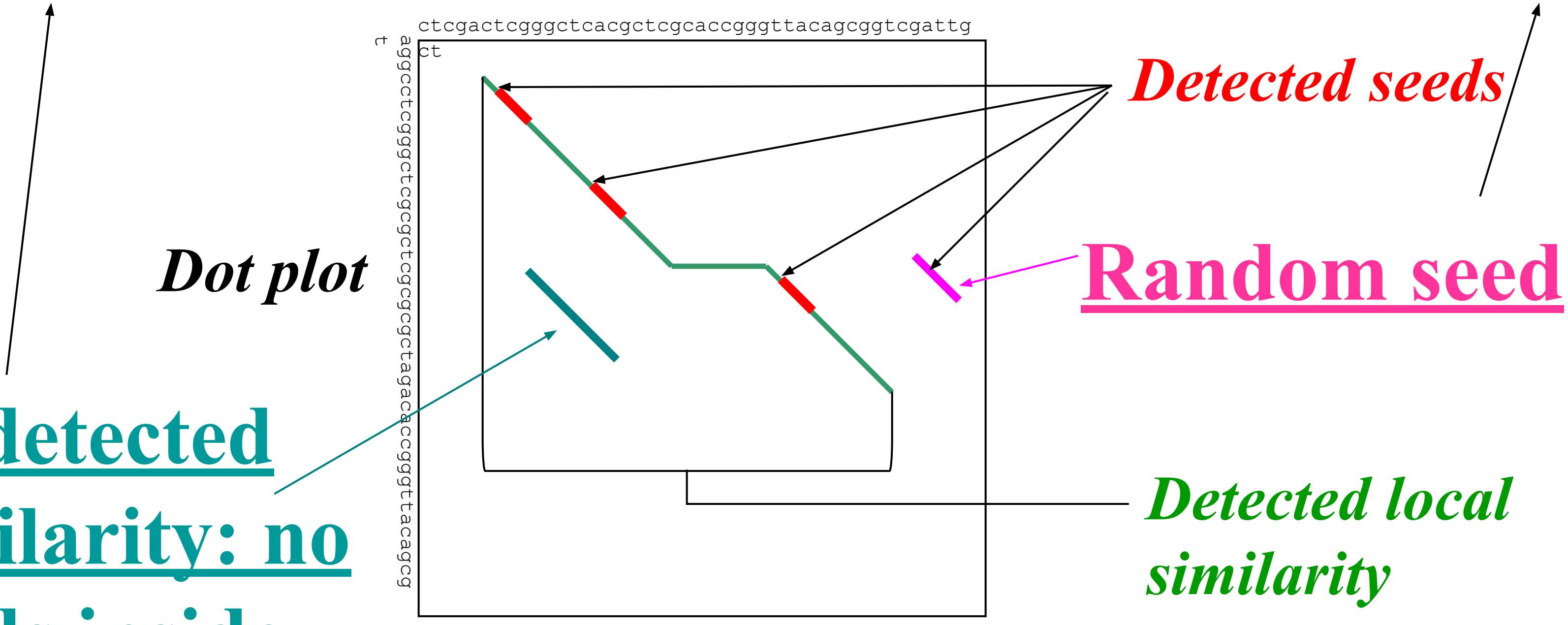
`ATCGGTGCGGTGCGCAAGAA`

`#######`

`TCAGTGCAATGCTCATGAA`

`:|::::||||||:::..::`

`CCGACACAATGCGTGACCC`



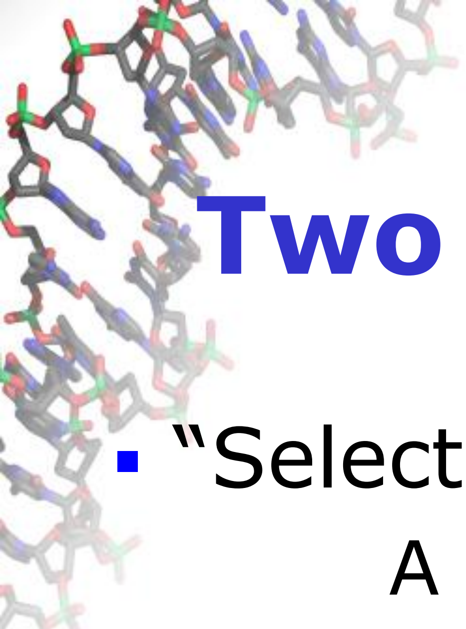
Dot plot

Detected seeds

Random seed

Undetected similarity: no seeds inside

Detected local similarity



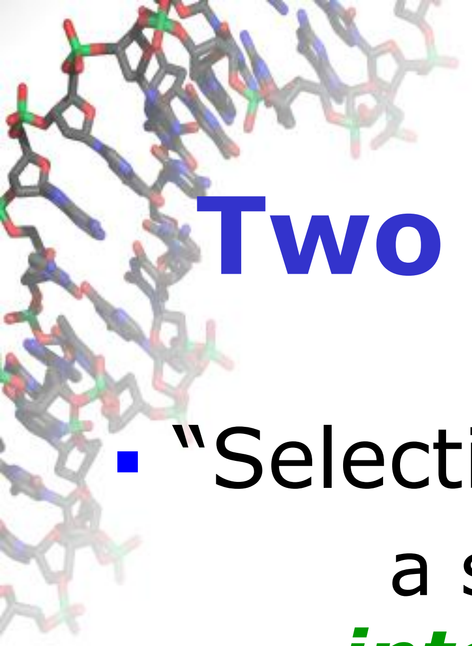
Two problems

- “Selectivity problem”

  A seed may NOT be a part of *an interesting similarity.*

- “Sensitivity problem”

  An interesting similarity may not contain a seed.



Two problems: refinement

- "Selectivity problem"

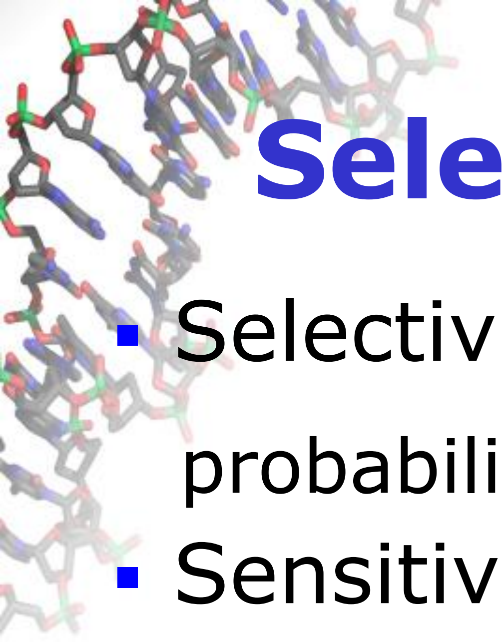
  a seed may NOT be a part of ***an interesting similarity.***

- "Sensitivity problem"

  an interesting similarity *may not contain a seed.*

To be specified:

What is ***an interesting similarity****?*



Selectivity and Sensitivity

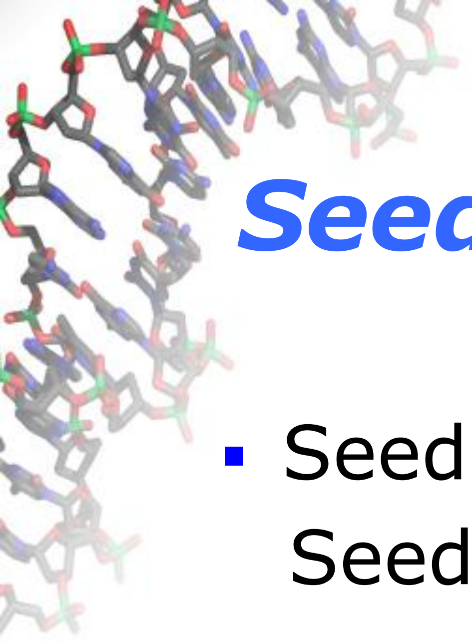
- Selectivity of the seed pattern:

  probability of random occurrences ~ $4^{-weight}$
- Sensitivity of the seed pattern:

  probability *for the seed to detect an* ***interesting similarity****.*

To be specified:

- **What set of similarities do we want to detect?**
- **What is the probability of each interesting similarity?**



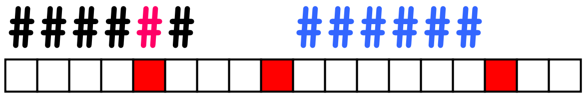
Seed detects the similarity…

- Seed pattern ###### ☐ *seed*

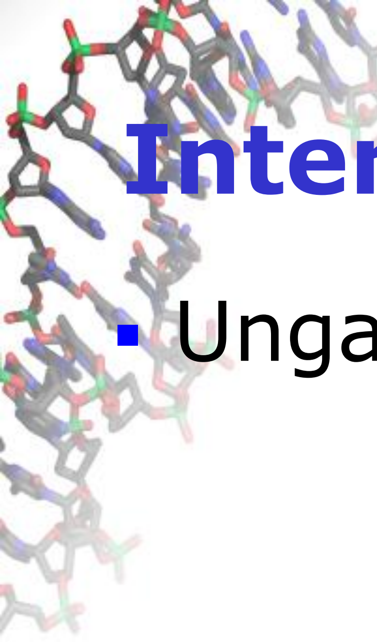
  Seed similarity (=seed alignment)

ATGCAA

ATGCAA

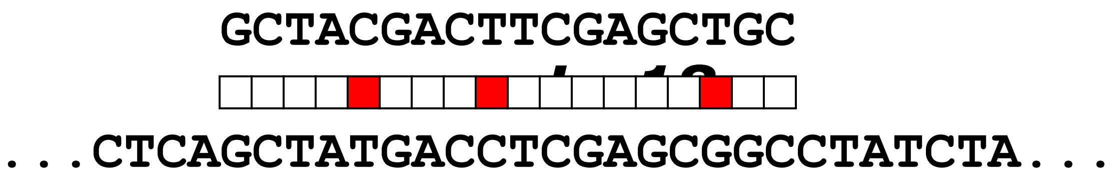


Seed fits the alignment



Interesting [target] alignments

- Ungapped alignments of a given length



GCTACGACTTCGAGCTGC

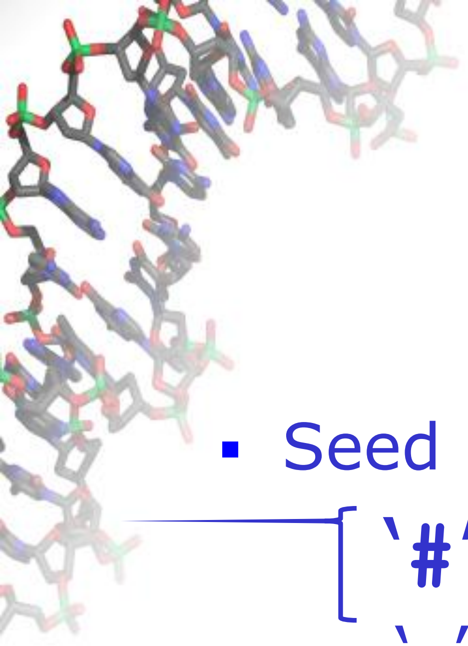
...CTCAGCTATGACCTCGAGCGGCCTATCTA...

- Probability model: ***Bernoully*** model;

  Random alignments: Prob(match) =0.25

  Target alignments: Prob(match) >> 0.25

Generalizations: Markov models, HMM
{not in this talk}



Spaced Seeds

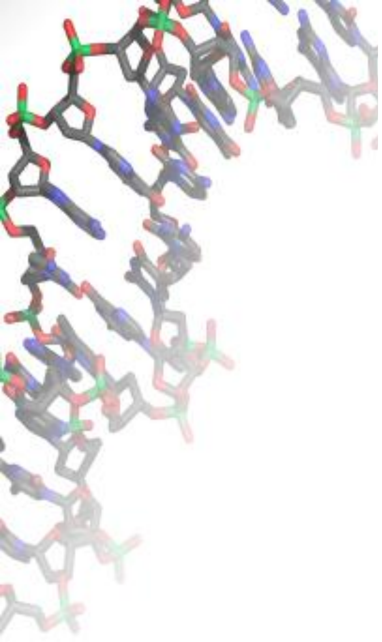
Ma, Tromp, Li 2002 (PatternHunter)

- Seed Pattern: **###--#-##**
  - ‘**#**’ : obligatory match position
  - ‘–’ : ***joker*** position (“don’t care” position)

Weight **: 6**[number of #]

- **Example :**

```
       ###--#-##
ATCAGTGCAATGCTCAAGA
|||||.||.||||:|||||
ATCAGCGCGATGCGCAAGA
```

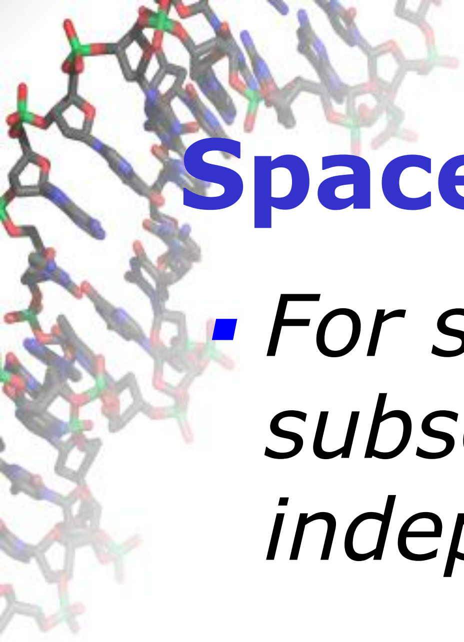


```
######

ATCAGTGCAATGCTCAAGA
|||||.||.||||:|||||
ATCAGCGCGATGCGCAAGA
######
 ######
  ######
   ######
    ######
     ######
      ######
       ######
        ######
         ######
          ######
           ######
            ######
             ######
```

```
###--#-##

ATCAGTGCAATGCTCAAGA
|||||.||.||||:|||||
ATCAGCGCGATGCGCAAGA
###--#-##
 ###--#-##
  ###--#-##
   ###--#-##
    ###--#-##
     ###--#-##
      ###--#-##
       ###--#-##
        ###--#-##
         ###--#-##
          ###--#-##
```



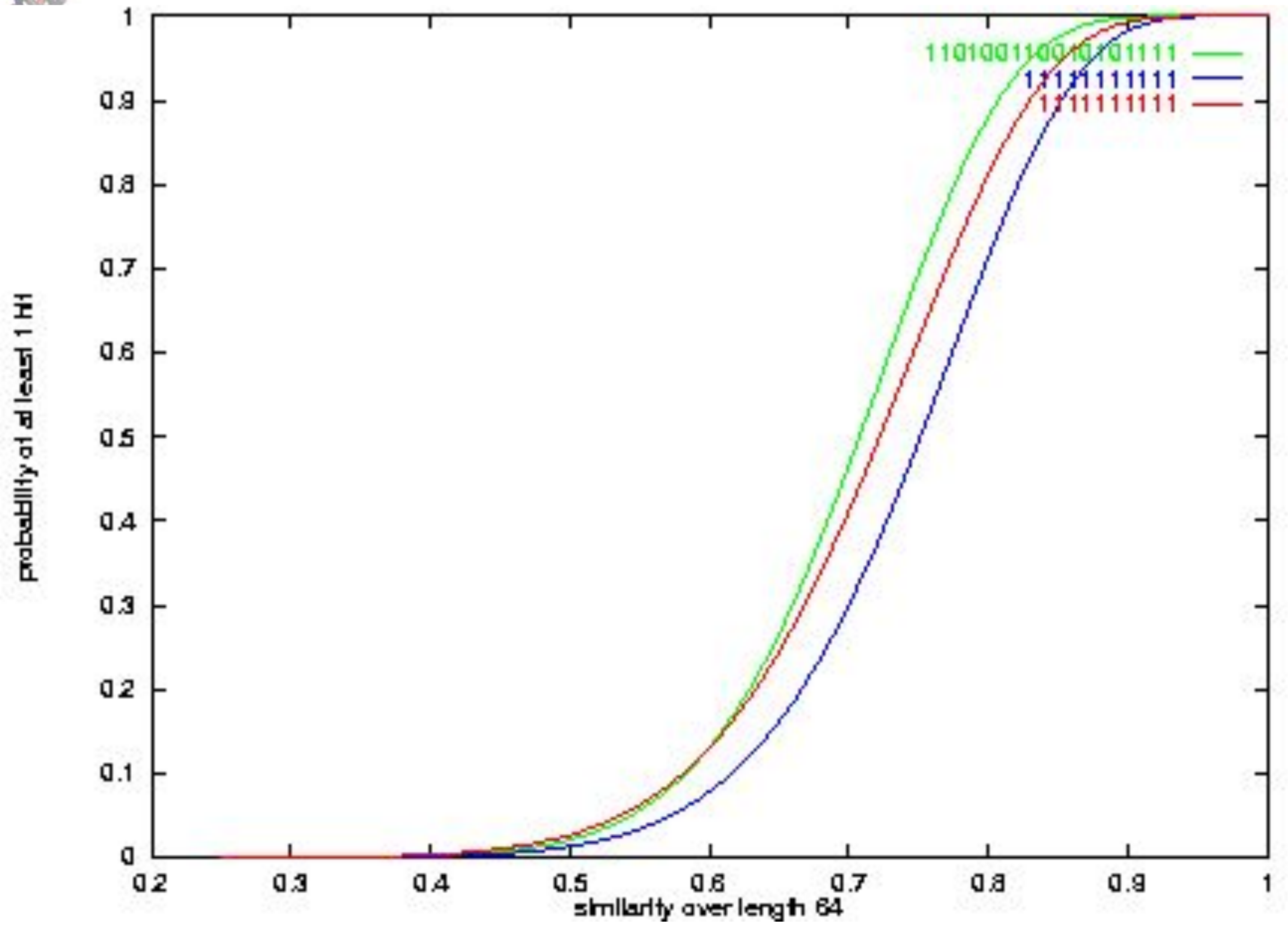
Spaced Seeds: the background

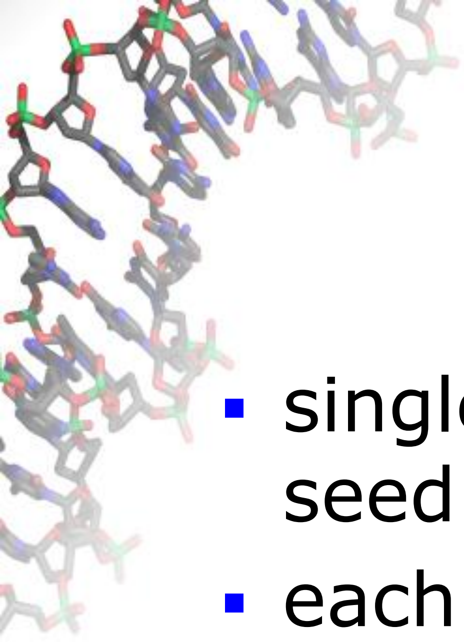
- *For spaced seeds, hits at subsequent positions are more independent events*

- *For contiguous vs. spaced seeds of the same weight:*
    - *the expected number of hits is (basically) the same*
    - *the probabilities of having* ***at least one hit*** *are very different*

Sensitivity: PH weight 11 seed vs BLAST 11 & 10

[after Ma, Tromp and Li]





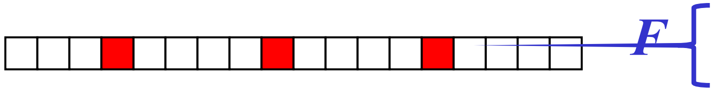
Multi-seeds: Families of seeds

- single filter based on several distinct seed patterns
- each seed pattern detects a part of interesting similarities but together they detect [almost] all of them

- Li, Ma, Kisman, Tromp 2004 (PatternHunter II)
- Sun, Buhler, RECOMB 2004
- Kucherov, Noe, Roytberg, 2005

Example: (18,3)-problem

(18,3)-problem: detect all similarities of length 18 with 3 mismatches



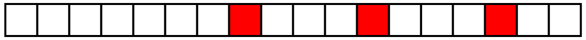
##-#-####

###---#--##-#

– every (18,3)-instance contains an occurrence of a seed of *F*

- all seeds of the family have the same weight 7

Example: (18.3)-problem (cont)



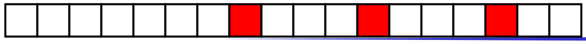
###---#--##-#

###---#--##-#

##-#-####

###---#--##-#

w=7



###-##---#-###

##-##-#####

###-####--##

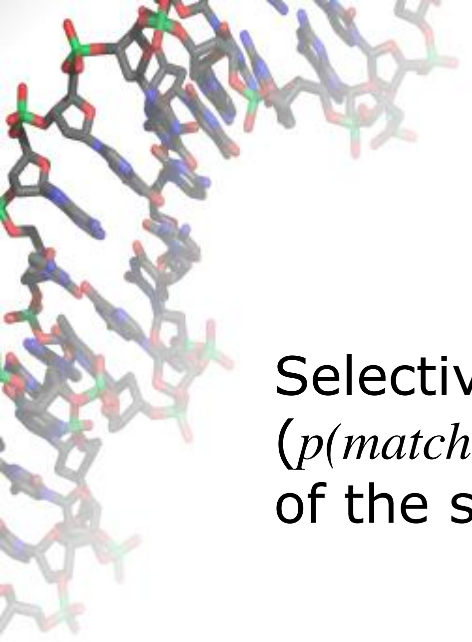
###-##---#-###

##----####-###

###---#-#-##-##

###-#-#-#-----###

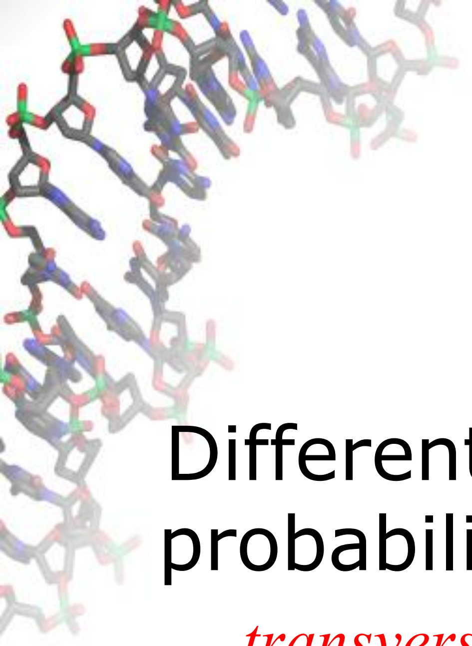
w=9



Comparative selectivity

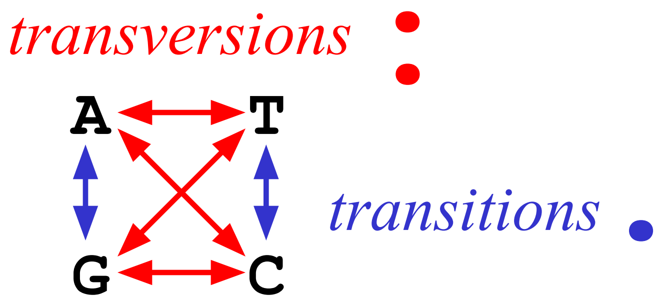
Selectivity of families on Bernoulli similarities (*p(match)* = *1/4*) estimated as the probability for one of the seeds to occur at a given position

####	***w=4***	*~39.* 10^{-4}
###-##	***w=5***	*~9.8* 10^{-4}
##-#-#### ###---#--##-#	***w=7***	*~1.2* 10^{-4}
##-##-##### ###-####--## ###-##---#-### ##----####-### ###---#-#-##-## ###-#-#-#-----###	***w=9***	*~0.23* 10^{-4}



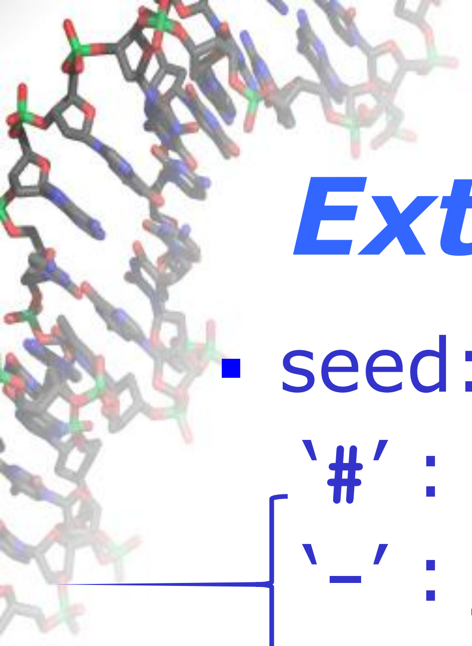
Subset seeds

Different mutational events have different probabilities



```
ATCAGTGCAATGCTCAAGA
|||||.||.||||:|||||
ATCAGCGCGATGCGCAAGA
```

Transitions are usually over-represented.



Extended seed alphabet

- seed: `##@#-#@-###`

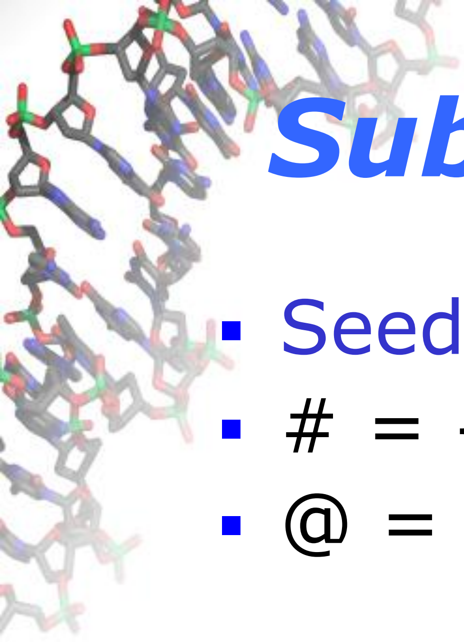
  '#' : obligatory match position

  '-' : joker position ("don't care" position)

  '@' : ***transition-constrained*** position

  position that corresponds to either a match or a transition.

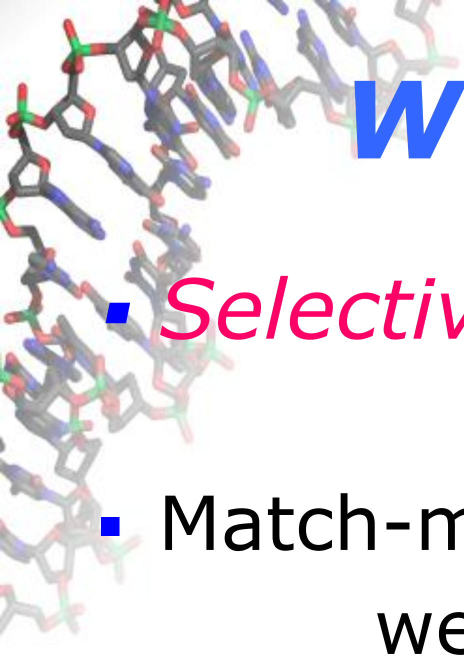
```
      ##@#-#@-###
ATCAGTGCAATGCTCAAGA
|||||.||.||||:|||||
ATCAGCGCGATGCGCAAGA
```



Subset letters and seeds

- Seed ***letter*** is a subset of aligned pairs.
- \# = {(A,A), (C,C), (G,G), (T,T)}
- @ = { (A,A), (C,C), (G,G), (T,T),
  (A,G), (G,A), (T,C), (C,T)}
- \- = {all pairs}

```
      ##@#-#@-###
ATCAGTGCAATGCTCAAGA
|||||.||.||||:|||||
ATCAGCGCGATGCGCAAGA
```



Weight of subset seed

- *Selectivity: probability of random occurrences of a seed*
- Match-mismatch case:

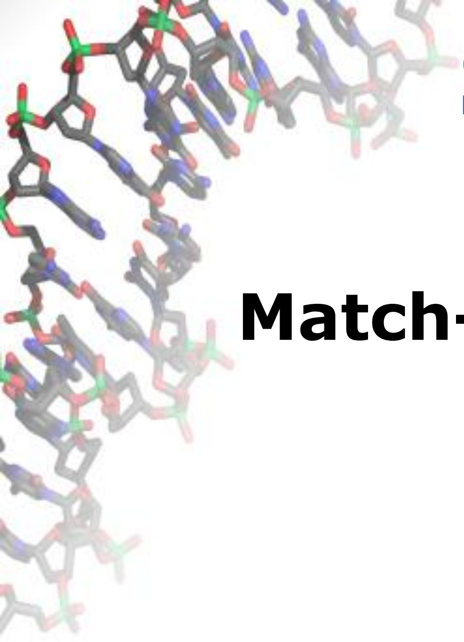
  weight – number of #;

  $S = 4^{-weight}$
- **General case:**

  $S = 4^{-weight}$ (by definition)
- seed: `##@#-#@-###`

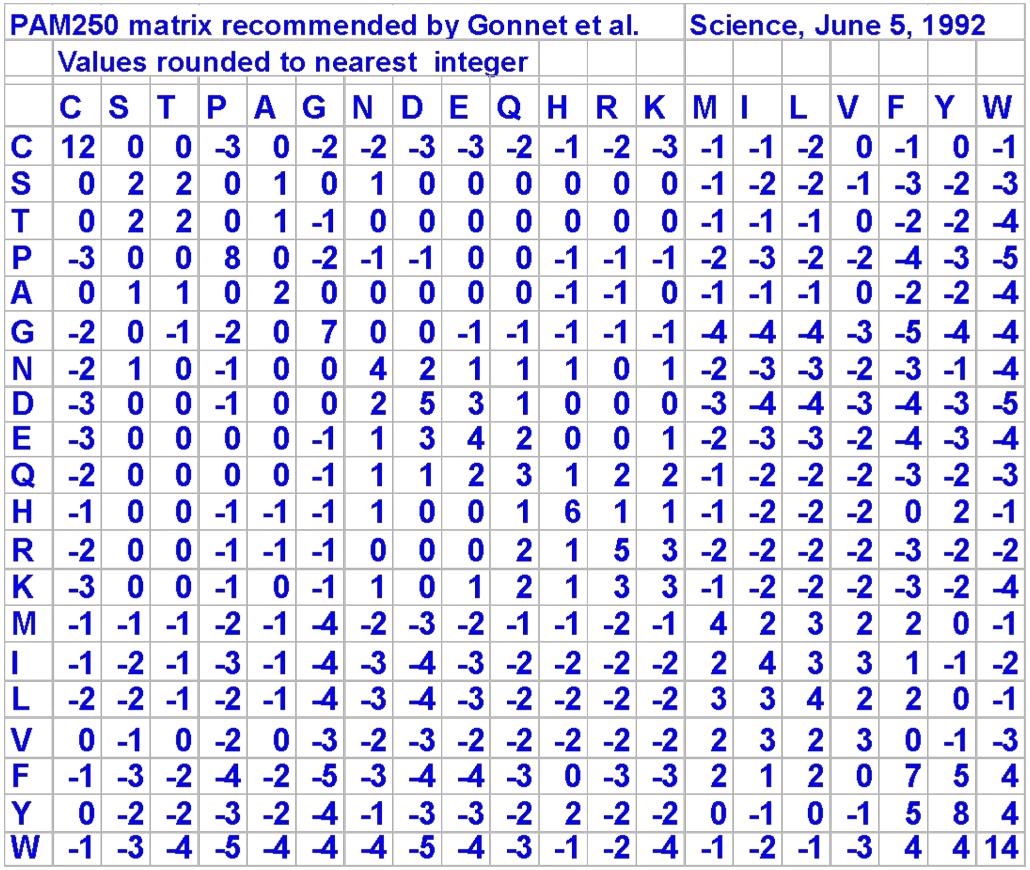
  ***Weight* : 8** [number of # + half number of @]

  @ carries *1 bit* of information whereas **#** carries *2 bits.*

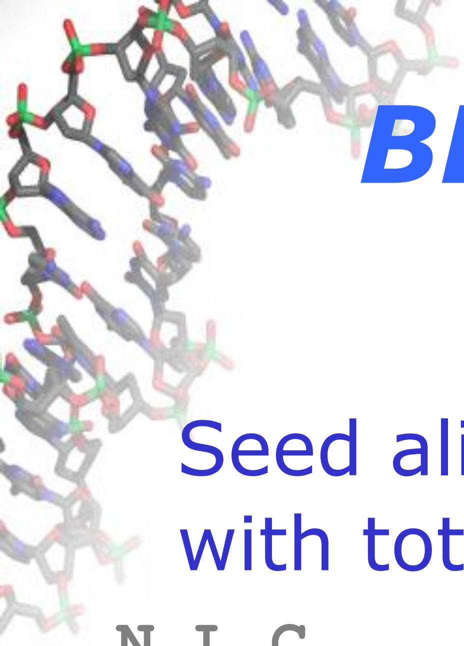


Seeds for proteins

Match-mismatch model is inadequate



PAM250 matrix recommended by Gonnet et al.													Science, June 5, 1992							
	Values rounded to nearest integer																			
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	12	0	0	-3	0	-2	-2	-3	-3	-2	-1	-2	-3	-1	-1	-2	0	-1	0	-1
S	0	2	2	0	1	0	1	0	0	0	0	0	0	-1	-2	-2	-1	-3	-2	-3
T	0	2	2	0	1	-1	0	0	0	0	0	0	0	-1	-1	-1	0	-2	-2	-4
P	-3	0	0	8	0	-2	-1	-1	0	0	-1	-1	-1	-2	-3	-2	-2	-4	-3	-5
A	0	1	1	0	2	0	0	0	0	0	-1	-1	0	-1	-1	-1	0	-2	-2	-4
G	-2	0	-1	-2	0	7	0	0	-1	-1	-1	-1	-1	-4	-4	-4	-3	-5	-4	-4
N	-2	1	0	-1	0	0	4	2	1	1	1	0	1	-2	-3	-3	-2	-3	-1	-4
D	-3	0	0	-1	0	0	2	5	3	1	0	0	0	-3	-4	-4	-3	-4	-3	-5
E	-3	0	0	0	0	-1	1	3	4	2	0	0	1	-2	-3	-3	-2	-4	-3	-4
Q	-2	0	0	0	0	-1	1	1	2	3	1	2	2	-1	-2	-2	-2	-3	-2	-3
H	-1	0	0	-1	-1	-1	1	0	0	1	6	1	1	-1	-2	-2	-2	0	2	-1
R	-2	0	0	-1	-1	-1	0	0	0	2	1	5	3	-2	-2	-2	-2	-3	-2	-2
K	-3	0	0	-1	0	-1	1	0	1	2	1	3	3	-1	-2	-2	-2	-3	-2	-4
M	-1	-1	-1	-2	-1	-4	-2	-3	-2	-1	-1	-2	-1	4	2	3	2	2	0	-1
I	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-2	-2	-2	2	4	3	3	1	-1	-2
L	-2	-2	-1	-2	-1	-4	-3	-4	-3	-2	-2	-2	-2	3	3	4	2	2	0	-1
V	0	-1	0	-2	0	-3	-2	-3	-2	-2	-2	-2	-2	2	3	2	3	0	-1	-3
F	-1	-3	-2	-4	-2	-5	-3	-4	-4	-3	0	-3	-3	2	1	2	0	7	5	4
Y	0	-2	-2	-3	-2	-4	-1	-3	-3	-2	2	-2	-2	0	-1	0	-1	5	8	4
W	-1	-3	-4	-5	-4	-4	-4	-5	-4	-3	-1	-2	-4	-1	-2	-1	-3	4	4	14

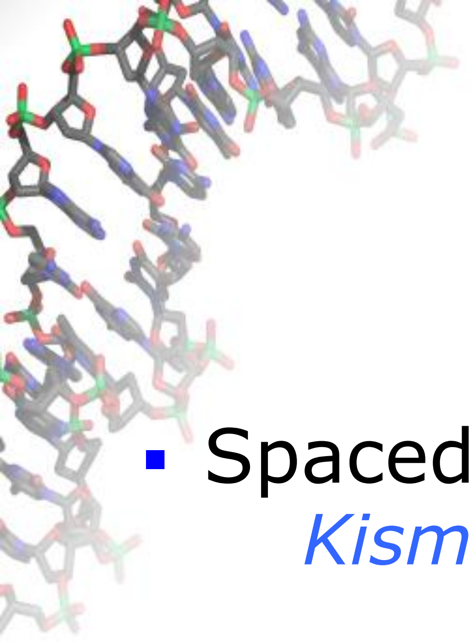


BLASTP: vector seeds

Seed alignment: any 3-letter alignment with total score exceeding a given cut-off

N L C	G D I	P C P
S S C	G Q V	P K P
1+1+12 = 14	7+1+3 = 11	8-3+8 = 13

An amino-acid triple T has a lot of *neighbors*, i.e. other triples forming a seed alignment with T



Improvements…

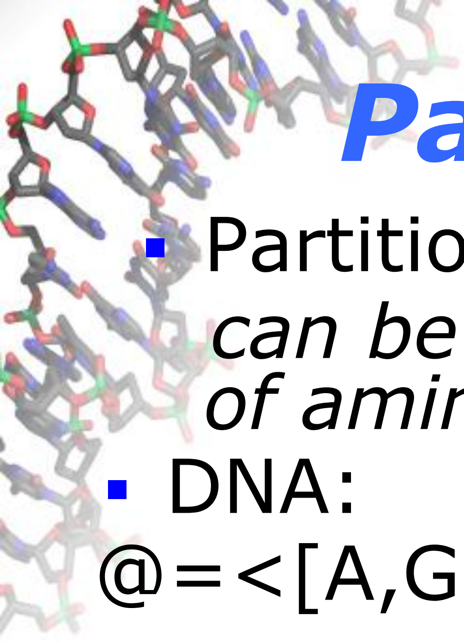
- Spaced vector seeds

  Kisman, Ma, Li, Wang 2005; Brown, 2005

- Subset seeds

  Kucherov, Noe, Roytberg, et al, 2007

- Multiple seeds *[both cases]*



Partition subset seeds

- Partition subset seeds: *each subset letter can be described by a partition of the set of aminoacid letters*
- DNA:

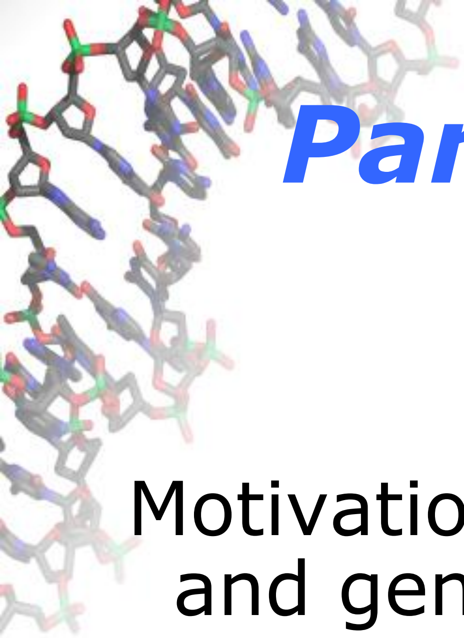
@=<[A,G];[T,C]>={(A,A), (A,G), (G,A), (G,G), (T,T), (T,C), (C,T), (C,C)}

- Proteins:

1)[C] [G] [P] [IVLM] [AST] [HWFY] [NDRKQE]

2)[C] [G] [P] [IV] [LM] [A] [ST] [H] [WFY] [N] [D] [RK] [QE]

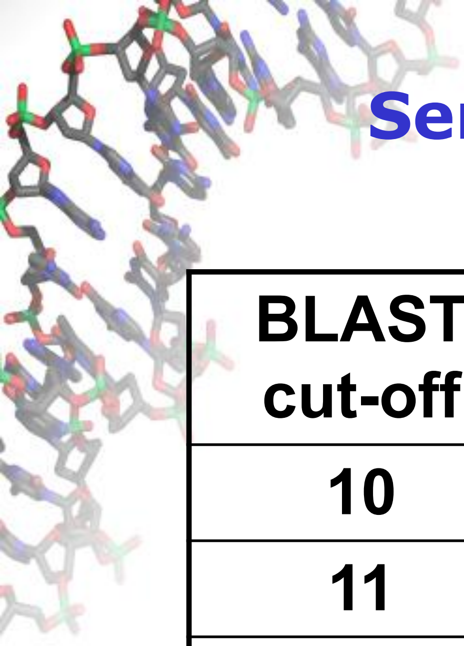
3)[C] [G] [P] [IV] [LM] [A] [S] [T] [H] [W] [FY] [N] [D] [RK] [QE]



Partition subset seeds *(cont)*

Motivation: In case of vector (BLAST-like) and general subset seeds each amino-acid triple T has a lot of *neighbors*, i.e. other triples forming a seed alignment with T

Partition seeds significantly decrease the number of neighbors of an amino-acid tuple



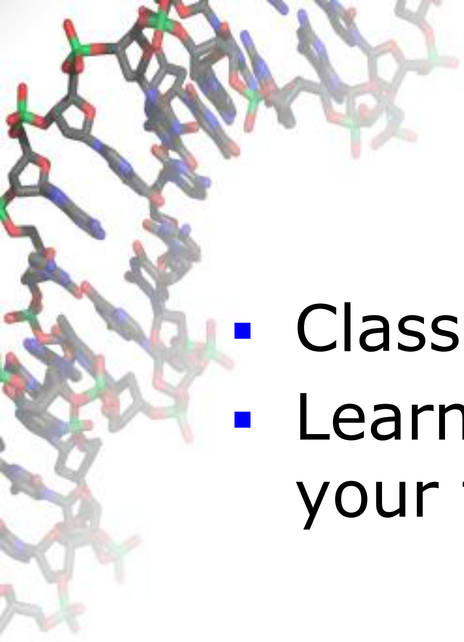
Sensitivity of different seed models

Sensitivity (%)

BLAST cut-off	BLAST (1 seed)	Partition seed (M)	Subset seed (M)	Vector seed (M)
10	*97.6*	*97.7*	*98.3*	*98.4*
11	*94.8*	*95.6*	*96.2*	*96.2*
12	*89.5*	*91.5*	*93.1*	*93.1*

Lost similarities (%) = 100-Sensitivity

BLAST cut-off	BLAST (1 seed)	Partition seed (M)	Subset seed (M)	Vector seed (M)
10	*2.4*	*2.3*	*1.7*	*1.6*
11	*5.2*	*4.4*	*3.8*	*3.8*
12	*10.5*	*8.5*	*6.9*	*6.9*

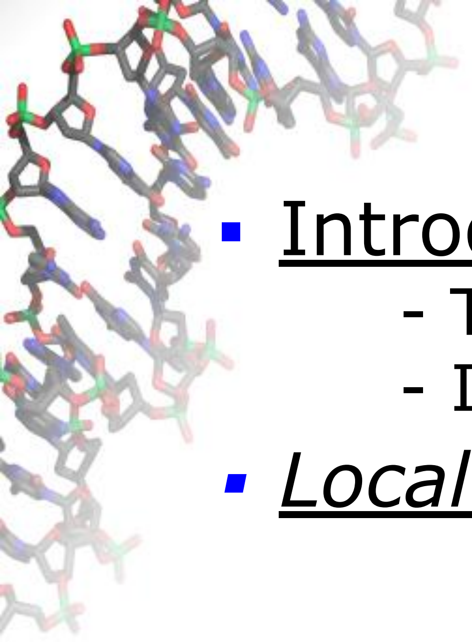


Seed summary

- Classic seeds are not optimal
- Learn about sensitivity of seeds in use wrt your target set of similarities

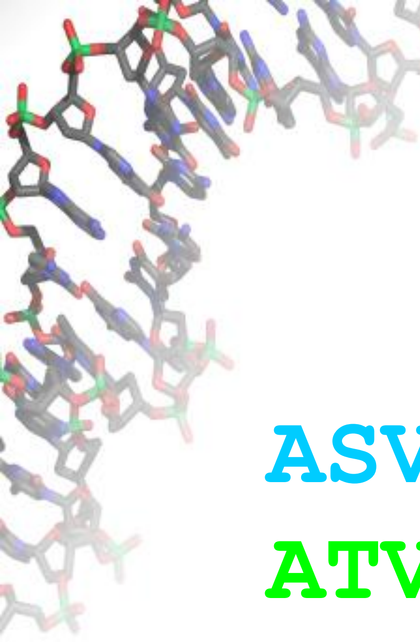
What was NOT discussed:

- How to find good seeds?
- How to calculate seed sensitivity?
- Criteria of hit extension
- More complicated types of similarities (e.g. containing inversions)….



Plan

- Introduction
  - - Types of sequence comparison
  - - Interpretation of the results
- *Local similarity search*
  - - Seed-based methods
  - - Sensitivity and selectivity
  - - Seed models
- **Alignment of homologous sequences**
  - **- What is “true” alignment**
  - **- Accuracy and Confidence**
  - **- The origin of difference**
- *Conclusion*
  - - Be prepared!



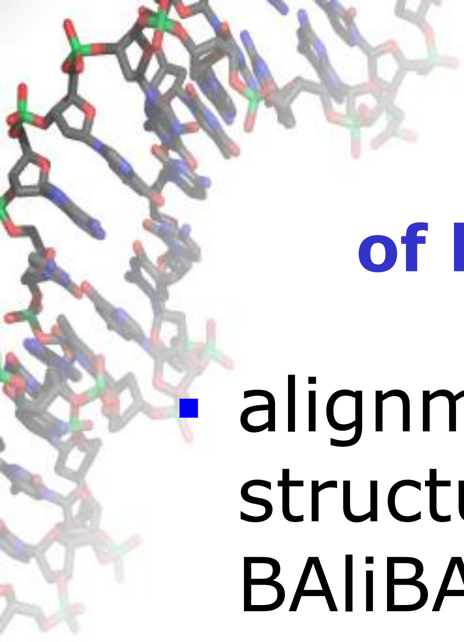
Biologically correct alignment

ASVVLDFTGT

```
ASVVLDFTGT   AS-VVLDFTGT
ATVVI—-TGS   GSMVLLEFSGT
```

```
AS-VVLDFTGT AS-VVLDFTGT
AT-VVI—-TGS GSMVLLEFSGT
```

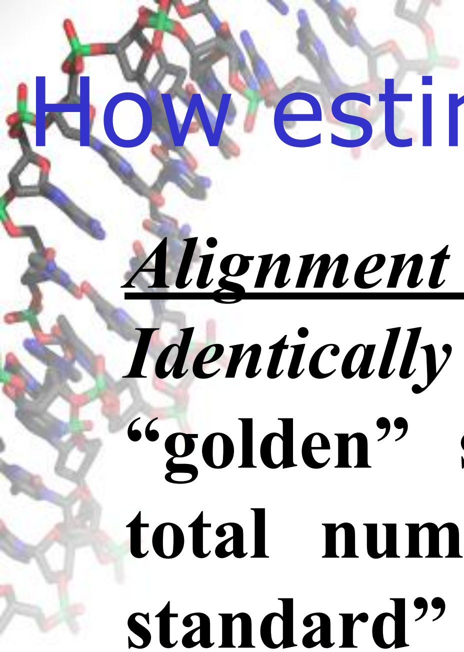
```
AS-VVLDFTGT
AT-VVI—-TGS
GSMVLLEFSGT
```



Approximations

of biologically correct alignments

- alignments of 3D-structures (databases of structural alignments like FSSP, 3D_Ali, BAliBASE);
- manually curated multiple alignments (databases of multiple alignments like SMART or Pfam);
- artificial sequences created according the proper model of evolution



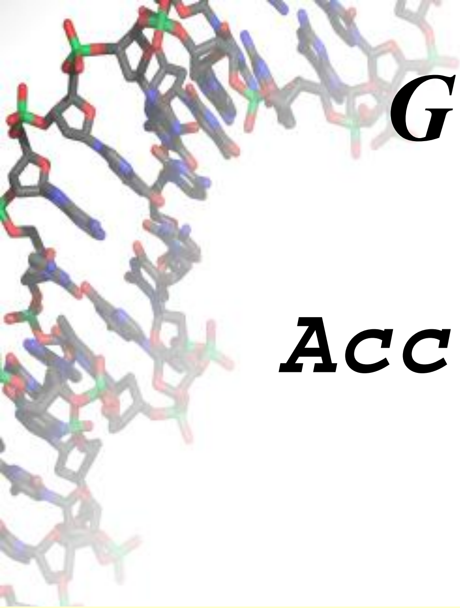
How estimate a quality of alignment?

__Alignment accuracy__: **the number I of positions *Identically* superimposed in algorithmic and “golden” standard” alignment divided by the total number G of positions in the “Golden standard” alignment**

$$Acc = I / G$$

__Alignment confidence:__ **the number I of positions *Identically* superimposed in algorithmic and “golden” standard” alignment divided by the total number A of aligned positions in the Algorithmic alignment**

$$Conf = I/A$$

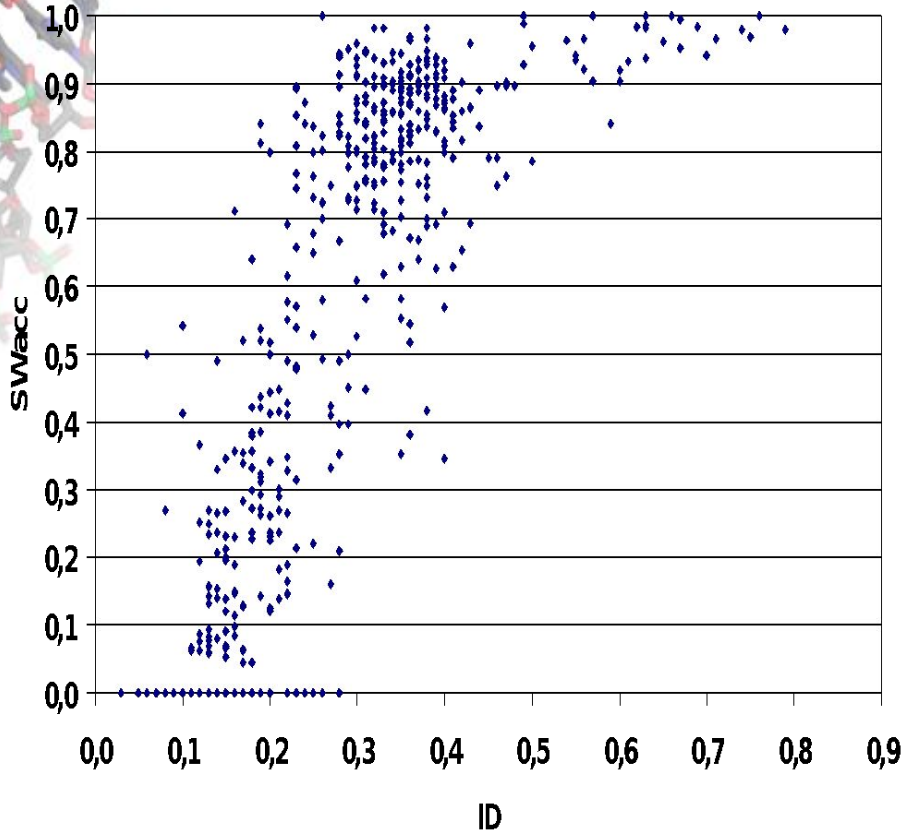


$G = 58 \qquad I = 42 \qquad A = 52$

$Acc = 42 / 58 \qquad Conf = 42/52$

```
GS)
  LKCnqli...ppfwkTCPkgkNLCYkmtmraapmvpvkRGCidvCPKssllikYMCCntDKCN.
  RICfnhqssqpqttkTCSpgeSSCYhkqwsdfrgtiieRGCg..CPTvkpgikLSCCesEVCNn
    *       ****************                           ******     *******************
    1              16                    6               19
SW)
    1              16                    6               19
    *        ****************                             ******   *******************
lk..C...nqliPPFWKTCPKGKNLCYK...mtmraapmvPVKRGCIDVCPKSSLLIKYMCCNTDKCN.
..riCfnhqssqPQTTKTCSPGESSCYHkqwsdfrgt...IIERGC..GCPTVKPGIKLSCCESEVCNN
```

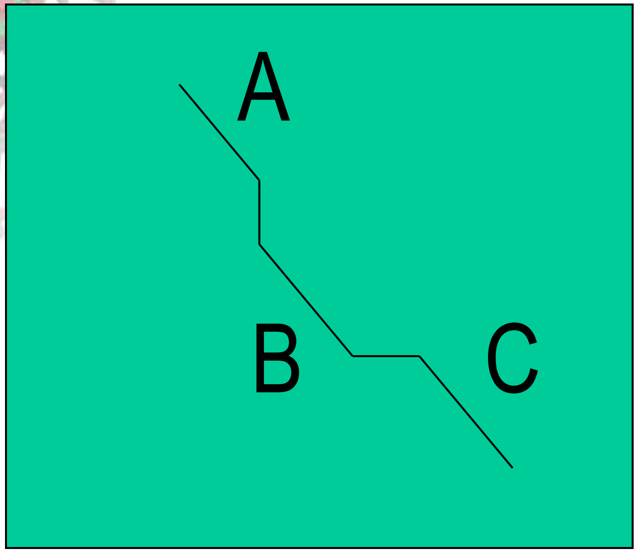
Accuracy of Smith-Waterman algorithm



Smith-Waterman algorithm doesn't allow to get right (SWacc > 0.5) alignment for the sequences with identity less than 0.3

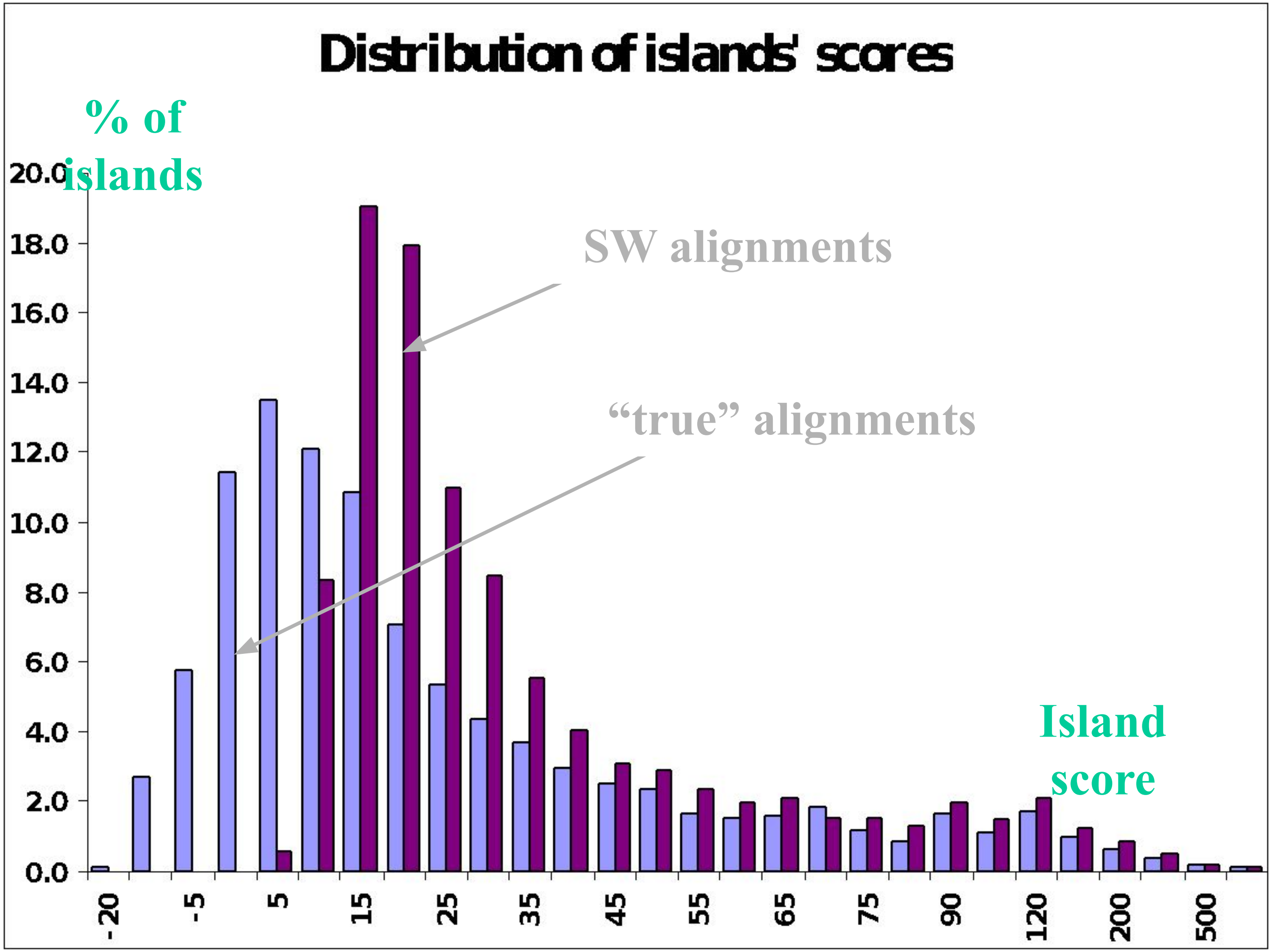
ID	SWacc
< 0,1	0,037
0,1-0,3	0,306
0,3-0,4	0,818
>0,4	0,893

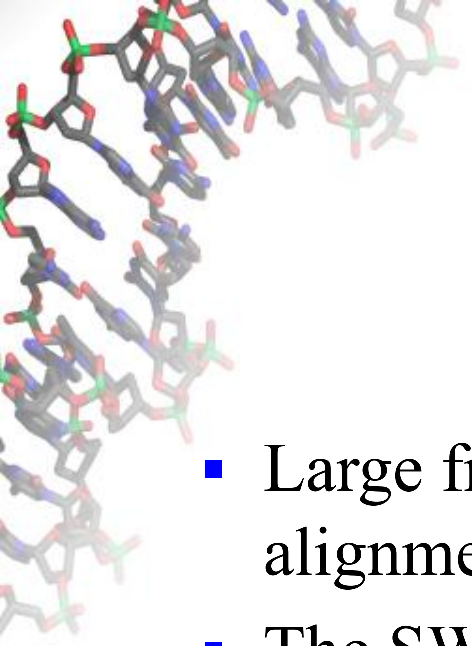
Islands = ungapped segments



A, B, C - islands

What are the scores of “true” islands?

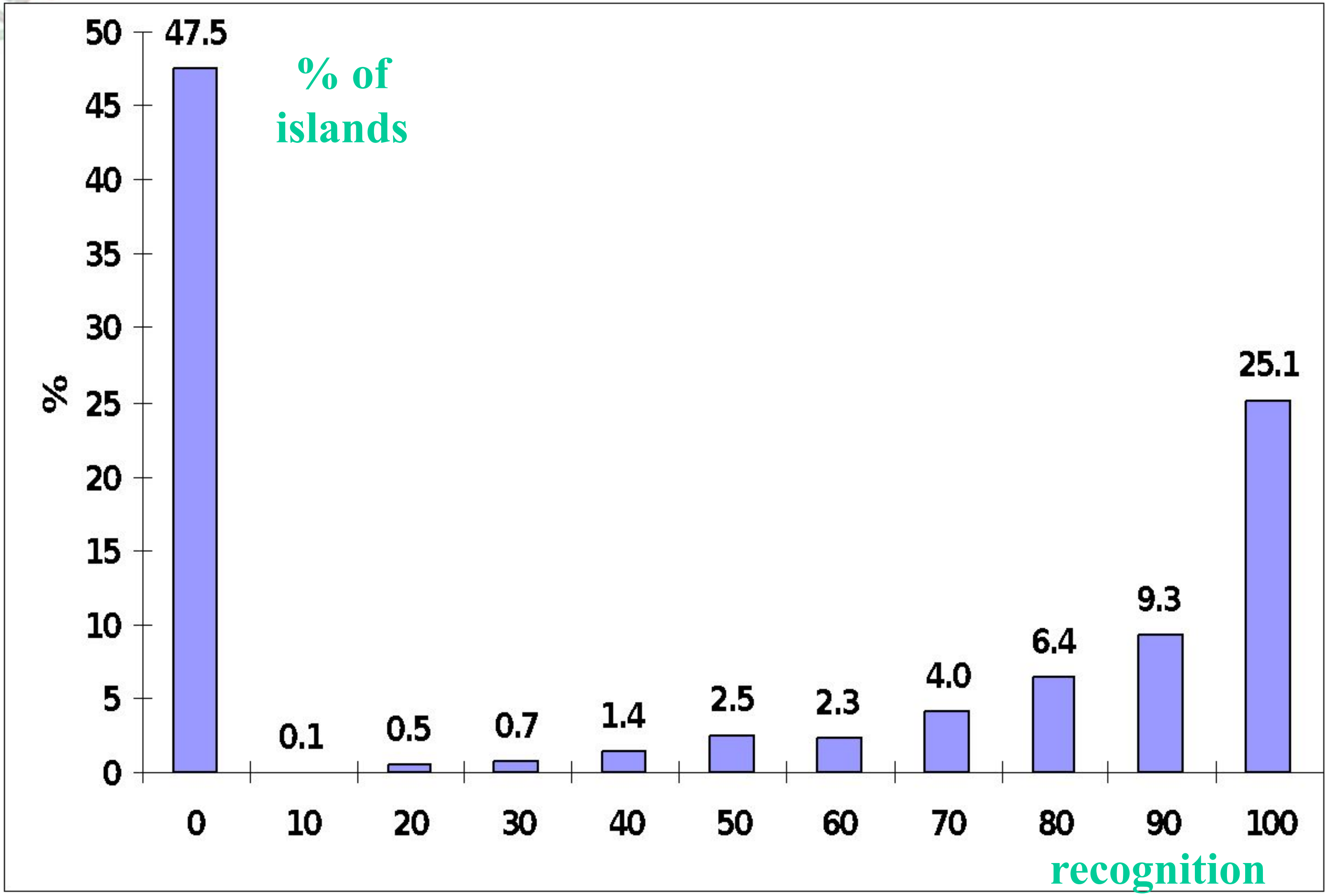


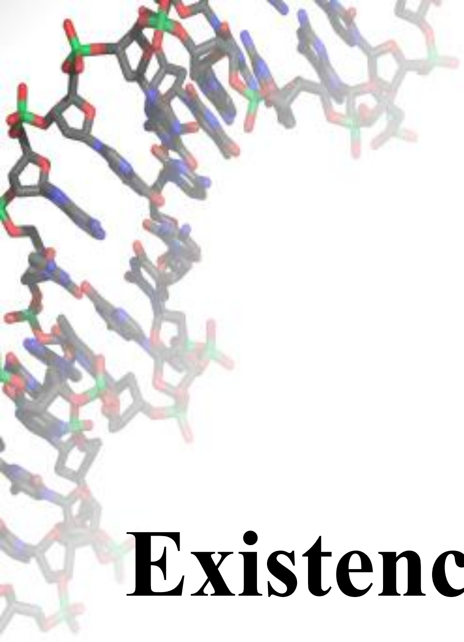


Low scoring islands

- Large fraction of islands (30%) in the “true” alignments have low positive or even negative score
- The SW algorithm is not able to reconstruct these islands
- Low scoring islands may constitute significant portion of the alignment (20%)

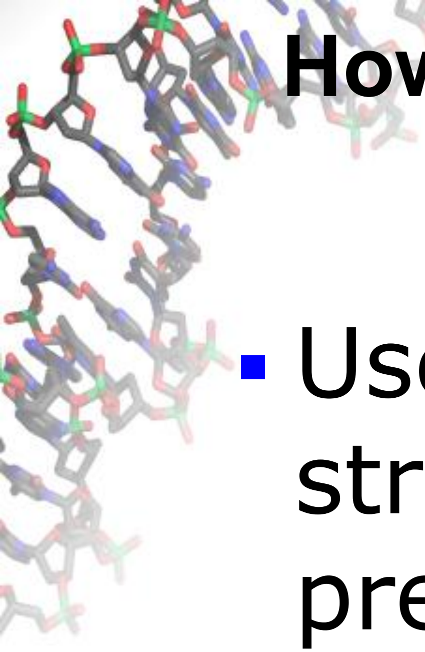
What part of the island is aligned correctly?





Main differences:

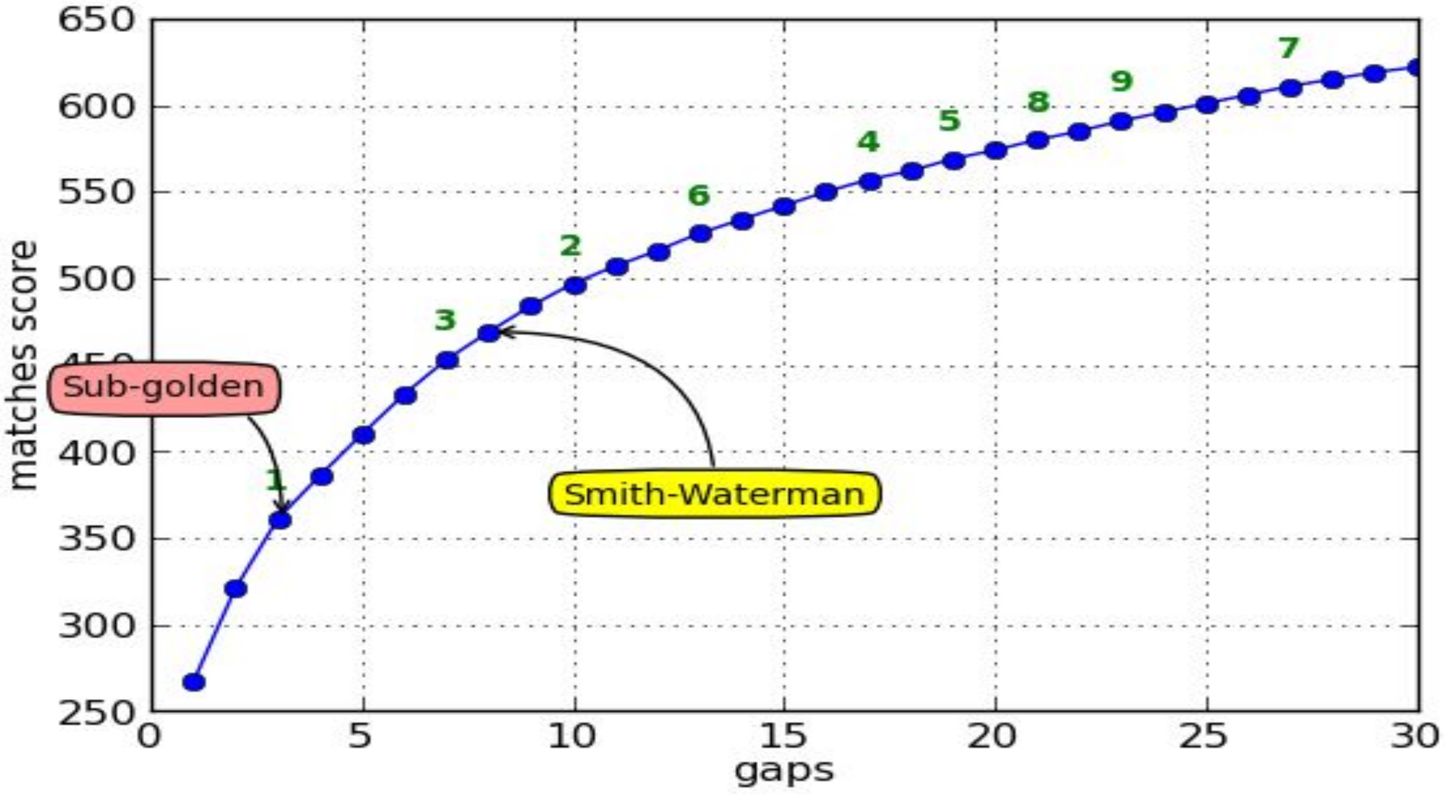
Existence of low-scoring islands
Number of gaps

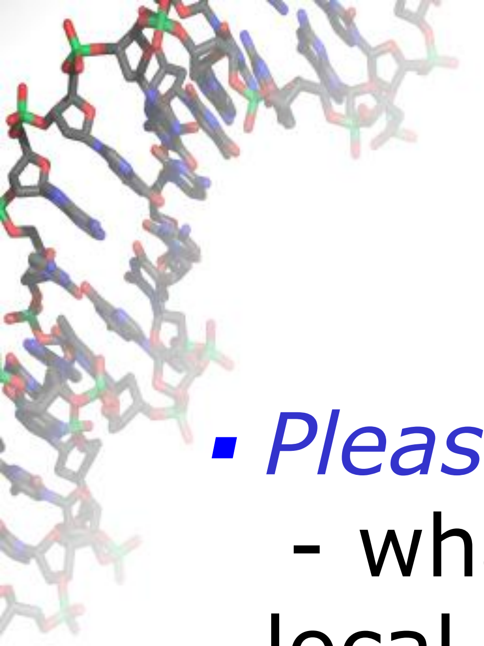


How to improve alignment quality:

- Use info about secondary structure: experimental or predicted (NOT to be discussed here)
- Create a (small) set of alternative candidate alignments

First sequence: **1nbaA (253 symbols)**
Second sequence: **1yacA (165 symbols)**
% identity: **0.146**

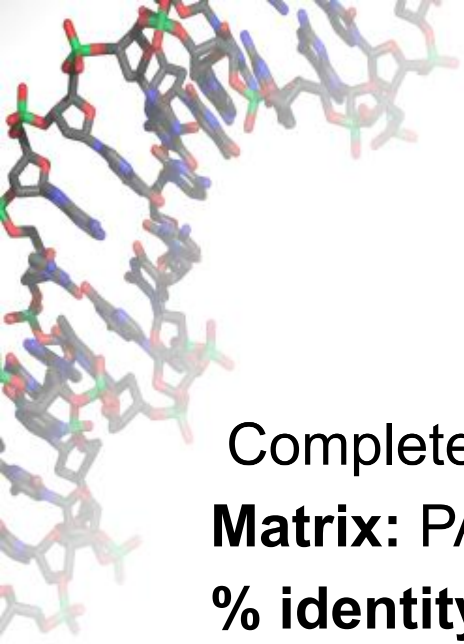




Conclusion

- *Please, learn:*
  - what is the sensitivity of your local search?
  - what is the accuracy of your alignment?

THANKS !



Completed

Matrix: PAM240 **GOP / GEP:** 15.0 / 1.0

% identity: 0.146

First sequence: 1nbaA (253 symbols)

Second sequence: 1yacA (165 symbols)

% identity: 0.146