FASTSP: linear time calculation of alignment accuracy

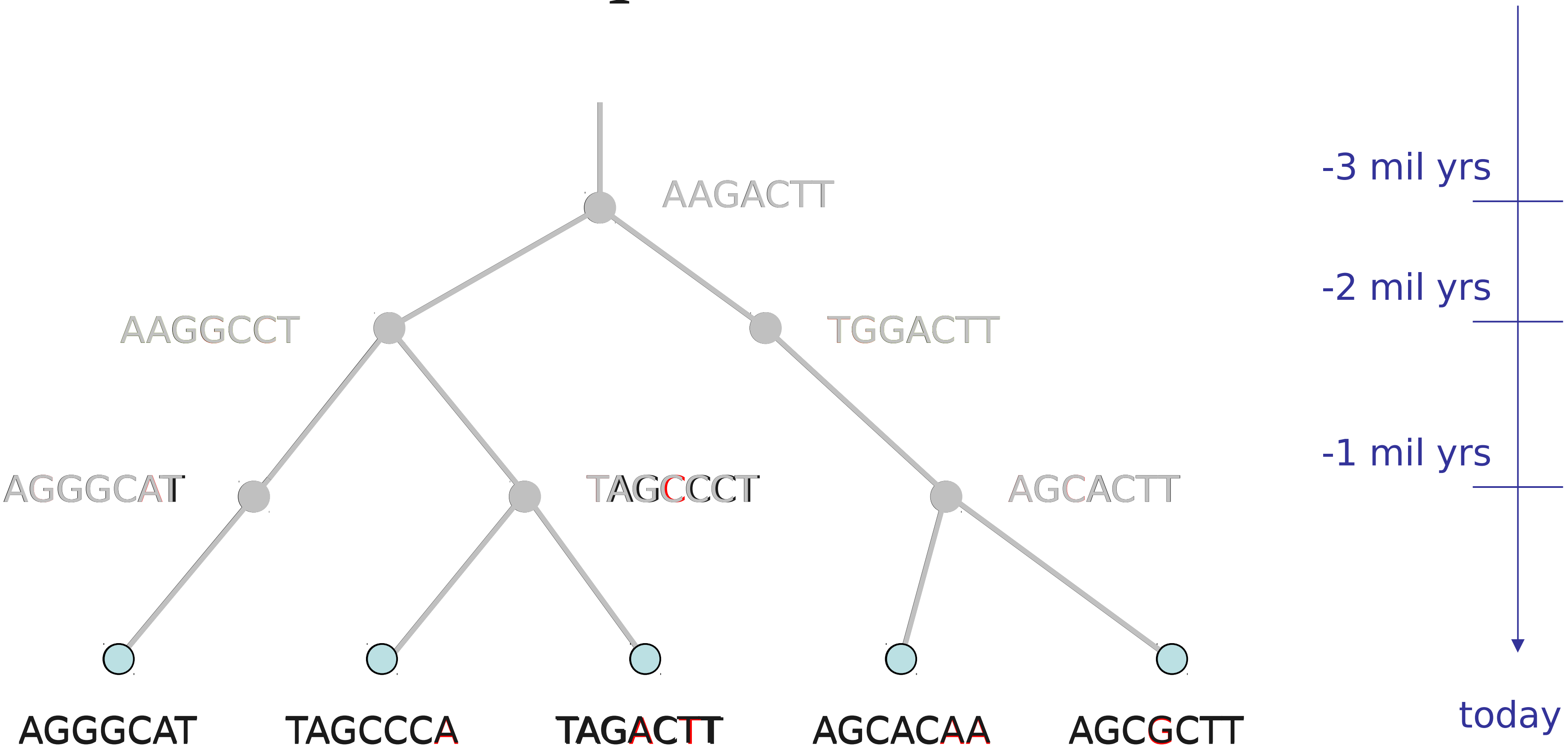
Siavash Mir arabbaygi

Research Preparation Exam

FastSP

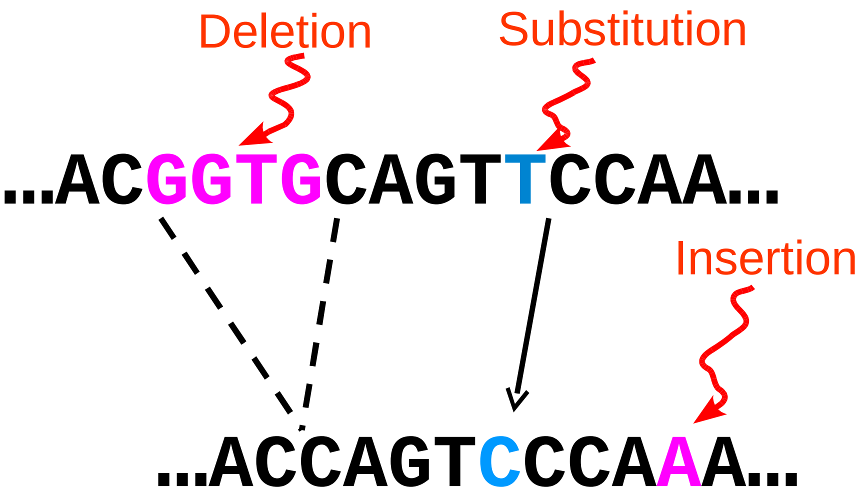
- Objective: Comparing very large Multiple Sequence Alignments efficiently (in linear time)
- Publication: Mirarab, S. and Warnow, T. (2011). Bioinformatics, 27(23), 3250–3258.
- Software: http://www.cs.utexas.edu/~phylo/software/fastsp/

DNA Sequence Evolution



```
...AGGGCAT...
...TAGCCCA...
...TAGACTT...
...AGCACAA...
...AGCGCTT...
```

Insertions and Deletions (indels)



Deletion

Substitution

...ACGGTGCAGTTTCCAA...

Insertion

...ACCAGTCCCAAA...

Sequence Alignment

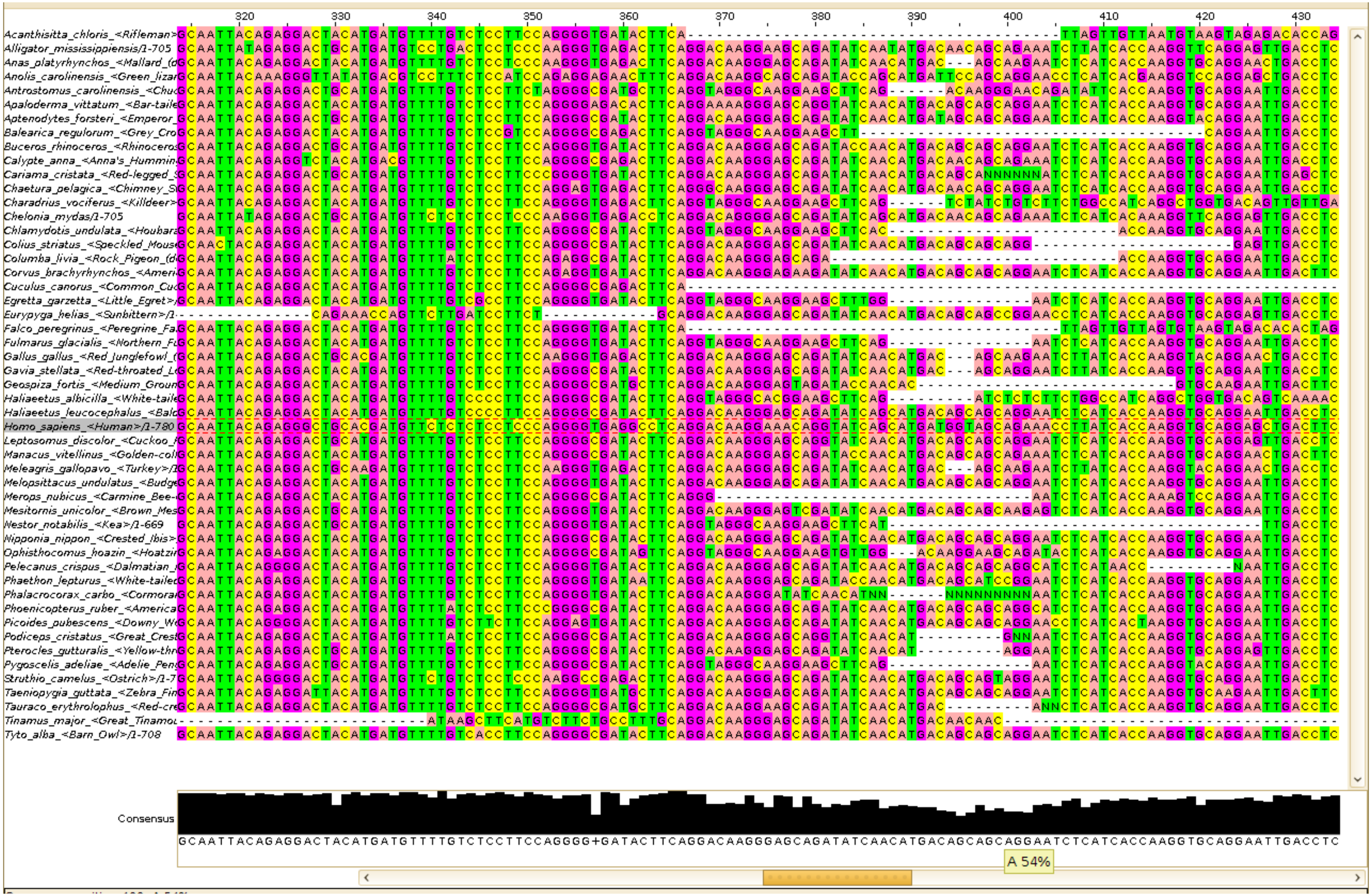
Evolutionary Truth:

```
...ACGGTGCAGTTTCCA-A...
...AC----CAGTCCCAAA...
```

Estimated Alignment:

```
...ACGGTGCAGTTTCC-AA-...
...AC----CAGT-CCCAAA...
```

Multiple Sequence Alignment (MSA)


320
330
340
350
360
370
380
390
400
410
420
430
Acanthisitta_chloris_<Rifleman>
Alligator_mississippiensis/1-705
Anas_platyrhynchos_<Mallard_(d
Anolis_carolinensis_<Green_lizar
Antrostomus_carolinensis_<Chuc
Apaloderma_vittatum_<Bar-taile
Aptenodytes_forsteri_<Emperor_
Balearica_regulorum_<Grey_Cro
Buceros_rhinoceros_<Rhinoceros
Calypte_anna_<Anna's_Hummin
Cariama_cristata_<Red-legged_S
Chaetura_pelagica_<Chimney_S
Charadrius_vociferus_<Killdeer>
Chelonia_mydas/1-705
Chlamydotis_undulata_<Houbara
Colius_striatus_<Speckled_Mous
Columba_livia_<Rock_Pigeon_(d
Corvus_brachyrhynchos_<Ameri
Cuculus_canorus_<Common_Cuc
Egretta_garzetta_<Little_Egret>
Eurypyga_helias_<Sunbittern>/1
Falco_peregrinus_<Peregrine_Fa
Fulmarus_glacialis_<Northern_Fu
Gallus_gallus_<Red_junglefowl_(
Gavia_stellata_<Red-throated_L
Geospiza_fortis_<Medium_Groun
Haliaeetus_albicilla_<White-taile
Haliaeetus_leucocephalus_<Bald
Homo_sapiens_<Human>/1-780
Leptosomus_discolor_<Cuckoo_R
Manacus_vitellinus_<Golden-coll
Meleagris_gallopavo_<Turkey>/1
Melopsittacus_undulatus_<Budge
Merops_nubicus_<Carmine_Bee-
Mesitornis_unicolor_<Brown_Mes
Nestor_notabilis_<Kea>/1-669
Nipponia_nippon_<Crested_Ibis>
Opisthocomus_hoazin_<Hoatzin
Pelecanus_crispus_<Dalmatian_P
Phaethon_lepturus_<White-tailed
Phalacrocorax_carbo_<Cormora
Phoenicopterus_ruber_<America
Picoides_pubescens_<Downy_W
Podiceps_cristatus_<Great_Crest
Pterocles_gutturalis_<Yellow-thr
Pygoscelis_adeliae_<Adelie_Pen
Struthio_camelus_<Ostrich>/1-7
Taeniopygia_guttata_<Zebra_Fin
Tauraco_erythrolophus_<Red-cre
Tinamus_major_<Great_Tinamou
Tyto_alba_<Barn_Owl>/1-708
Consensus
GCAATTACAGAGGACTACATGATGTTTTGTCTCCTTCCAGGGG+GATACTTCAGGACAAGGGAGCAGATATCAACATGACAGCAGCAGGAATCTCATCACCAAGGTGCAGGAATTGACCTC
A 54%

MSA Estimation Methods

Basis: score alignments based on a similarity matrix and gap penalties

Most formulations of the problem are NP-complete.

Polynomial for two sequences (dynamic programming)

There are plenty of methods to estimate alignments:

- Progressive methods: use a guide tree to align sequences two at a time, from most similar to more distantly related.
- Iterative methods: similar to progressive, but allow updating pair-wise alignments if scores are improved
- Hidden Markov models: model “current”' alignment as a Markov model, and use Viterbi algorithm to successively add new sequences to the current alignment

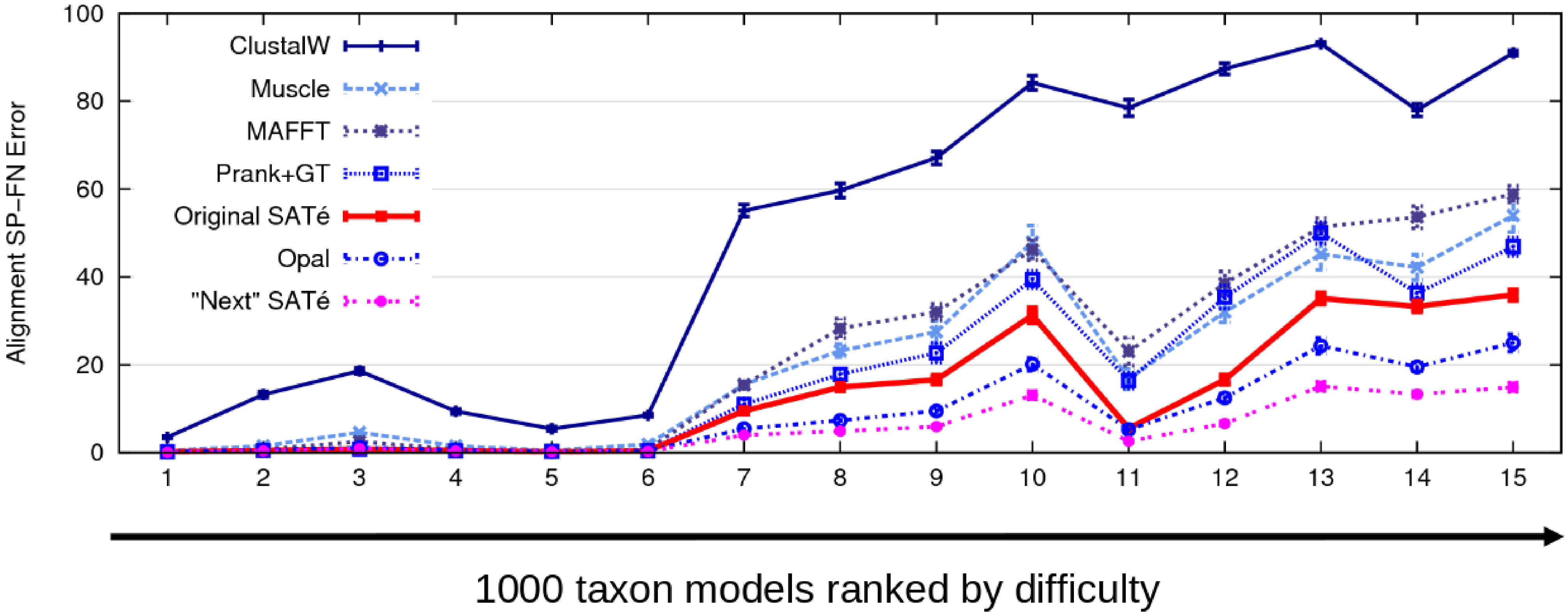
Alignment Comparison

- Many ways to estimate alignments

- Alignments need to be compared

Alignment Comparison: performance Study

- Assessing accuracy in performance studies
- Example:



From: Liu,K. et al. (2009) Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. Science, 324, 1561–1564.

Alignment Comparison: Phylogenetic Uncertainty

- Different MSA methods produce alignments that differ enough to introduce phylogenetic uncertainty (Wong et al., 2008)

- Alignment error increases with the size of the dataset (Liu et al., 2009, 2010)

- Using several alignments, and comparisons of these alignments

Alignments Comparison Metrics

- The Developer score = SP-score (sum-of-pairs):

  Percentage of Homologies in Reference Alignment that are found in the estimated alignment (shared homologies).

Homologies

- The Developer score = SP-score (sum-of-pairs):

  Percentage of Homologies in Reference Alignment that are found in the estimated alignment (shared homologies).

Homology: Any pair of characters in the same column of a MSA

```
  012345678           0123456789
0 AGTGCTTC-         0 AGTGCTTC--
1 A---CTCCA         1 A---CT-CCA
2 AC-CGTCCA         2 ACC-GT-CCA
```

Homologies

- The Developer score = SP-score (sum-of-pairs):

  Percentage of Homologies in Reference Alignment that are found in the estimated alignment (shared homologies).

Homology: Any pair of characters in the same column of a MSA

```
  012345678             0123456789

0 AGTGCTTC-           0 AGTGCTTC--

1 A---CTCCA           1 A---CT-CCA

2 AC-CGTCCA           2 ACC-GT-CCA
```

Homologies

- The Developer score = SP-score (sum-of-pairs):

  Percentage of Homologies in Reference Alignment that are found in the estimated alignment (shared homologies).

Homology: Any pair of characters in the same column of a MSA

```
  012345678        0123456789

0 AGTGCTTC-      0 AGTGCTTC--

1 A---CTCCA      1 A---CT-CCA

2 AC-CGTCCA      2 ACC-GT-CCA
```

Homologies

- The Developer score = SP-score (sum-of-pairs):

  Percentage of Homologies in Reference Alignment that are found in the estimated alignment (shared homologies).

Homology: Any pair of characters in the same column of a MSA

```
  012345678            0123456789
0 AGTGCTTC-          0 AGTGCTTC--
1 A---CTCCA          1 A---CT-CCA
2 AC-CGTCCA          2 ACC-GT-CCA
```

Homologies (count)

- The Developer score = SP-score (sum-of-pairs):

  Percentage of Homologies in Reference Alignment that are found in the estimated alignment (shared homologies).

Number of Homologies: two chose number of characters per column

```
  012345678
0 AGTGCTTC-
1 A---CTCCA
2 AC-CGTCCA
  310133331
```

total=18

```
  0123456789
0 AGTGCTTC--
1 A---CT-CCA
2 ACC-GT-CCA
  3110330311
```

total=16

Representing Characters

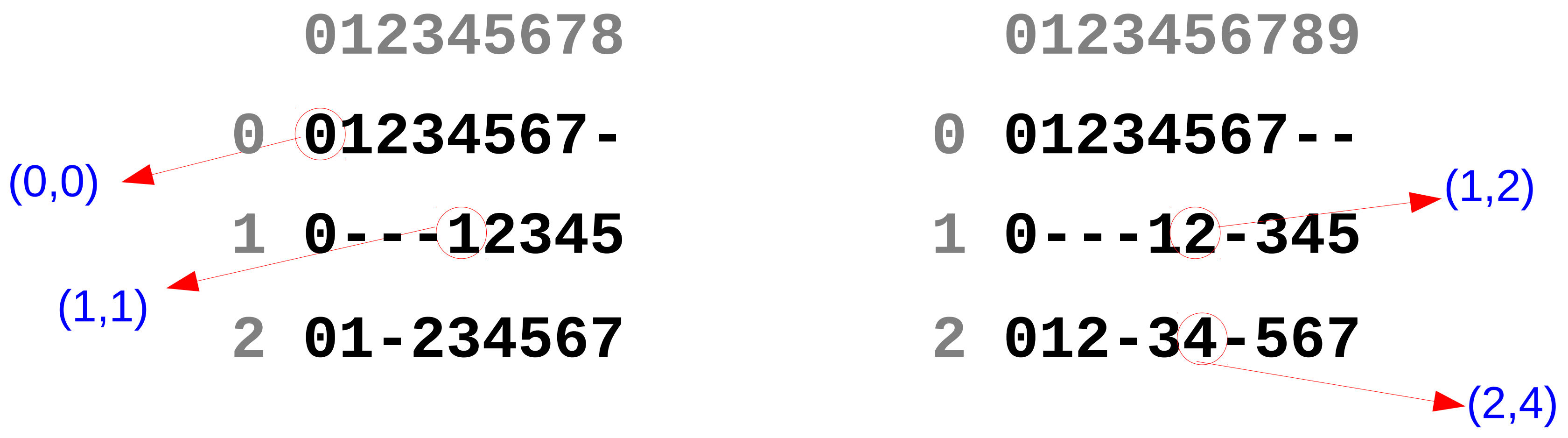
- The Developer score = SP-score (sum-of-pairs):

  Percentage of Homologies in Reference Alignment that are found in the estimated alignment (shared homologies).

Character Representation: a pair (a,b) where

a indicates the row in the alignment matrix

b indicates the position of the character in *unaligned* sequence



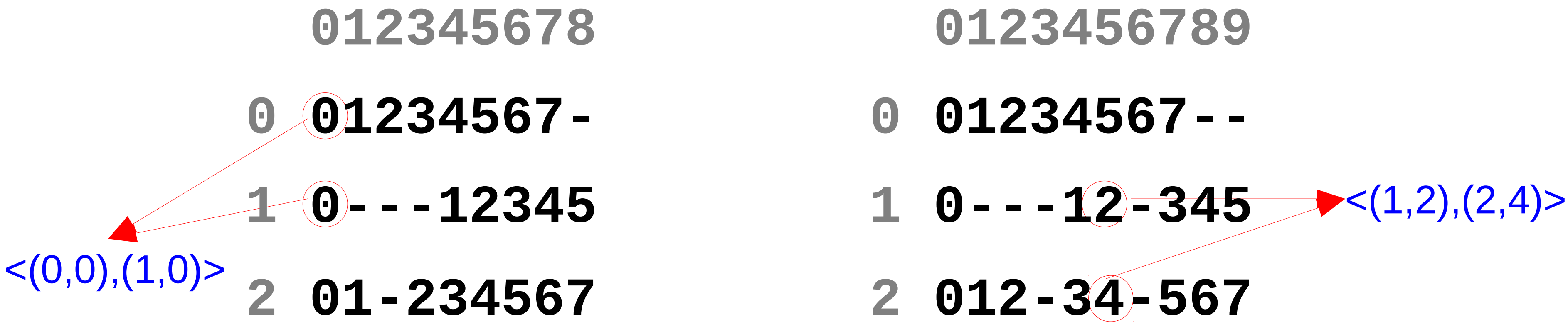
```
  012345678        0123456789
0 01234567-      0 01234567--
1 0---12345      1 0---12-345
2 01-234567      2 012-34-567
```

Representing Homologies (homology)

- The Developer score = SP-score (sum-of-pairs):

Percentage of Homologies in Reference Alignment that are found in the estimated alignment (shared homologies).

Homology Representation: a pair <(a,b),(c,d)> where (a,b) each represent a character in the alignment, and (a,b) and (c,d) are in the same column of the alignment.



```
  012345678

0 01234567-
1 0---12345
2 01-234567
```

<(0,0),(1,0)>

```
  0123456789

0 01234567--
1 0---12-345
2 012-34-567
```

<(1,2),(2,4)>

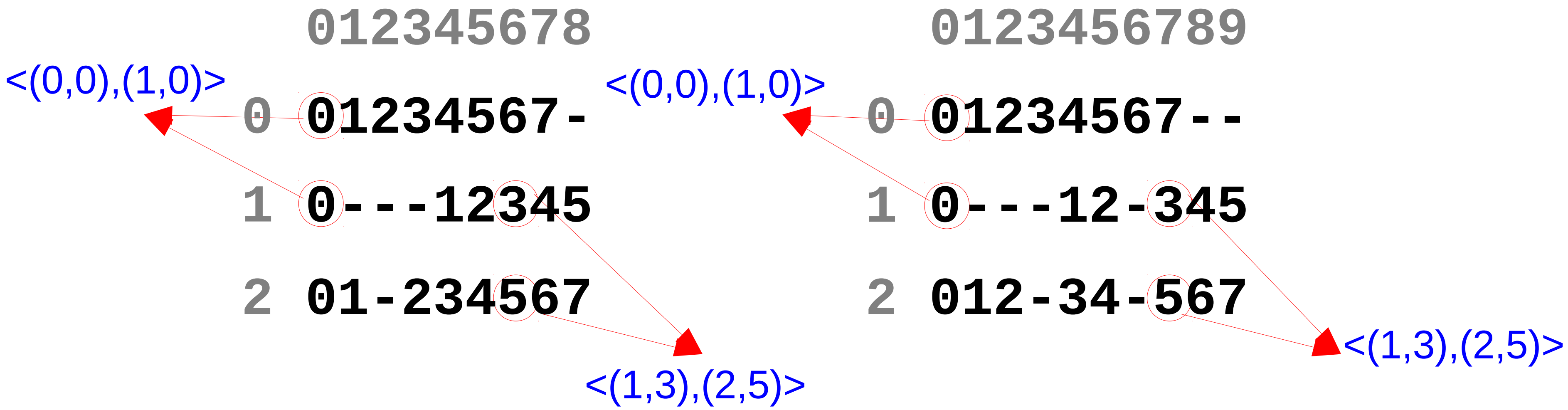
Note: the order doesn't matter: <(a,b),(c,d)> = <(c,d),(a,b)>

Shared Homology

- The Developer score = SP-score (sum-of-pairs):

  Percentage of Homologies in Reference Alignment that are found in the estimated alignment (shared homologies).

Shared Homologies: two homologies are shared between the two alignments if they have the exact same representation.



```
  012345678
0 01234567-
1 0---12345
2 01-234567
```

<(0,0),(1,0)>

<(1,3),(2,5)>

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

<(0,0),(1,0)>

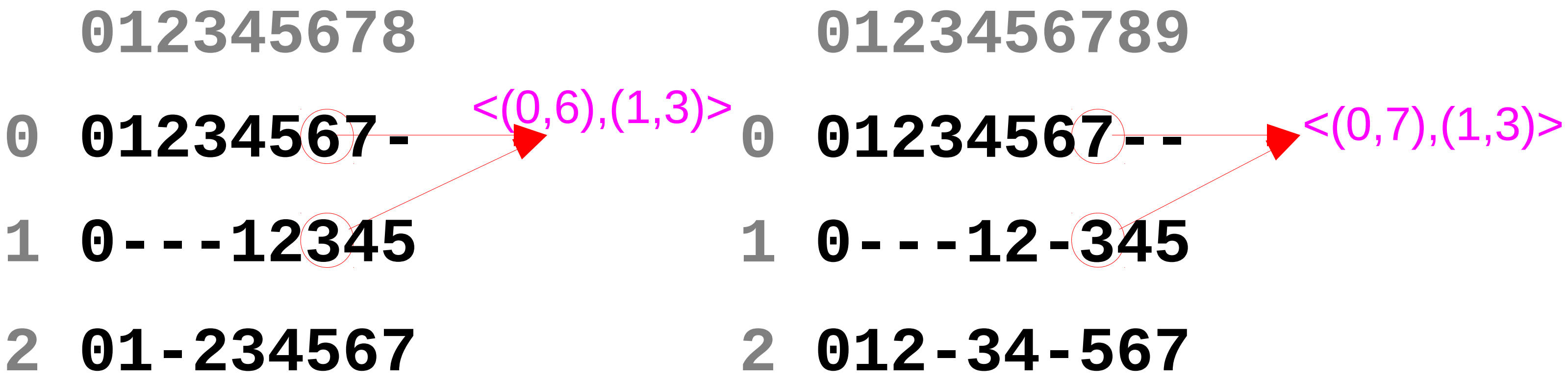
<(1,3),(2,5)>

Shared Homology

- The Developer score = SP-score (sum-of-pairs):

  Percentage of Homologies in Reference Alignment that are found in the estimated alignment (shared homologies).

Shared Homologies: two homologies are shared between the two alignments if they have the exact same representation.



```
  012345678
0 01234567-
1 0---12345
2 01-234567
```

<(0,6),(1,3)>

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

<(0,7),(1,3)>

SP-Score

- The Developer score = SP-score (sum-of-pairs):

  Percentage of Homologies in Reference Alignment that are found in the estimated alignment (shared homologies).

SP-Score: find all homologies in both alignments, find those that are shared, and divide by the number of homologies in the reference alignment.

```
Reference:   012345678      Estimated:   0123456789

           0 01234567-                 0 01234567--

           1 0---12345                 1 0---12-345

           2 01-234567                 2 012-34-567

ALL:         310133331   =18
                                       SP-Score=13/18=72%
SHARED:      310033111   =13
```

Modeler Score

- The Modeler score:

  Percentage of Homologies in the estimated Alignment that are found in the reference alignment (shared homologies).

  SP-Score: find all homologies in both alignments, find those that are shared, and divide by the number of homologies in the reference alignment.

```
Reference:  012345678     Estimated:  0123456789

         0  01234567-              0  01234567--

         1  0---12345              1  0---12-345

         2  01-234567              2  012-34-567

            310133331     ALL:        3110330311   =16

                          SHARED:     3100330111   =13
```

Modeler Score= 13/16=81%

Total Column Score

- Total Column (TC) score:

  Percentage of *aligned* columns in the reference alignment that are found in the estimated alignment.

```
Reference:   012345678     Estimated:   0123456789

           0 01234567-                0 01234567--

           1 0---12345                1 0---12-345

           2 01-234567                2 012-34-567

ALIGNED:     YYNYYYYYY     =8

SHARED:      YYNNYYNNY     =6
```

TC Score= 6/8=75%

Definitions

k = number of characters in the longest sequence

k1 = number of sites in the reference alignment

k2 = number of sites in the estimated alignment

n = number of sequences

```
Reference:  012345678    Estimated:  0123456789

          0 01234567-              0 01234567--

          1 0---12345              1 0---12-345

          2 01-234567              2 012-34-567

            k=7 k1=9

            n=3 k2=10
```

Brute Force Calculation

- Homologies in each alignment can be represented as a presence/absence matrix with n.k rows and columns

- $O(n^2k^2)$ time and memory.

FastSP: Objectives

Show that all three scores can be calculated in linear time (with respect to k.n)

Implement an efficient algorithm to calculate alignment scores

FastSP: Idea

- Characters in each column (x) of the reference alignment are dispersed in one or more columns in the estimated alignment.

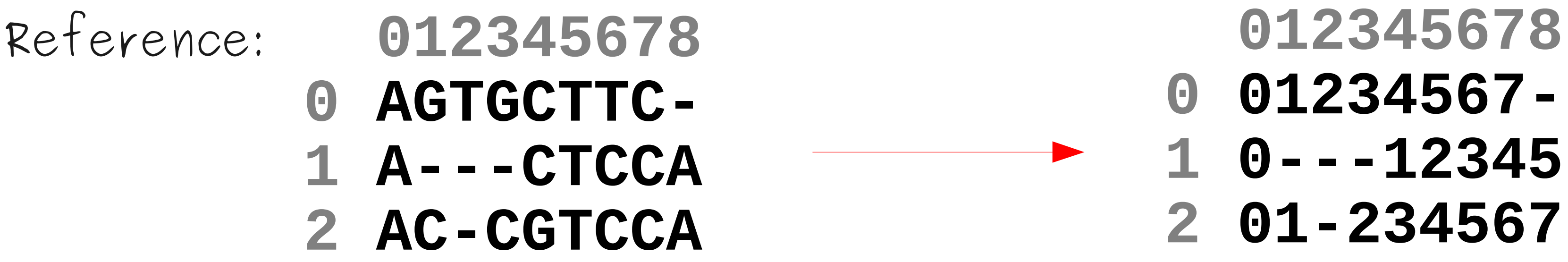
- Divide characters in column x into equivalence classes, such that all characters in the same equivalence class are in the same column in the estimated alignment

- Number of shared homologies contributed by column x is
  - sum (for all equivalence classes S of x) |S| choose 2

```
Reference: 012345678     Estimated: 0123456789
         0 01234567-              0 01234567--
         1 0---12345              1 0---12-345
         2 01-234567              2 012-34-567
```

$$\binom{1}{2}+\binom{2}{2}=1$$

FastSP: Algorithm



Reference:

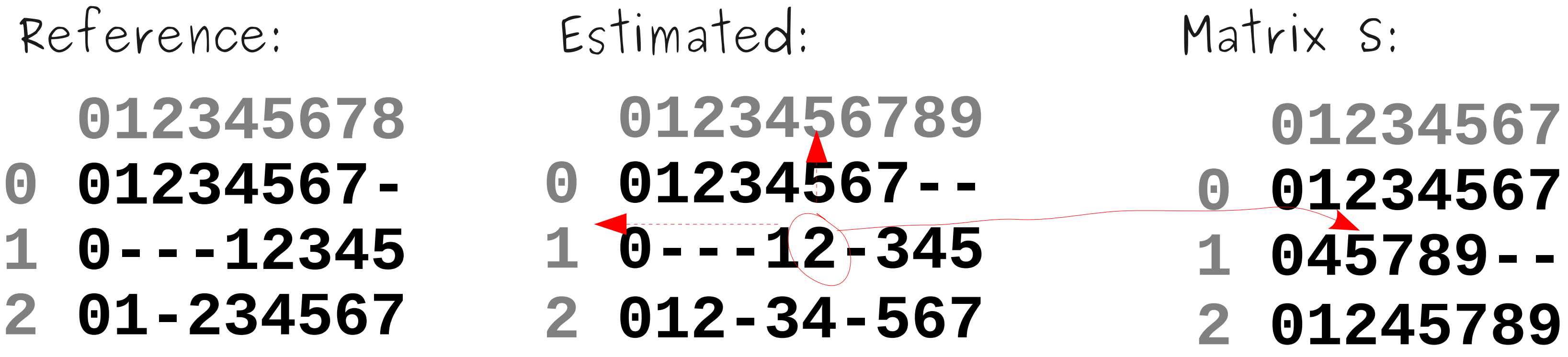
```
  012345678
0 AGTGCTTC-
1 A---CTCCA
2 AC-CGTCCA
```

```
  012345678
0 01234567-
1 0---12345
2 01-234567
```

1- Read reference alignment and save it with this character representation

- (also find k and n).

FastSP: Algorithm



Reference:

```
  012345678
0 01234567-
1 0---12345
2 01-234567
```

Estimated:

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

Matrix S:

```
  01234567
0 01234567
1 045789--
2 01245789
```

2- Read estimated alignment and create a n by k matrix S such that

- S[i,j]=x iff Estimated_Alignment[i,x]=j.

FastSP

Reference:

```
  012345678
0 01234567-
1 0---12345
2 01-234567
```

Estimated:

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

Matrix S:

```
  01234567
0 01234567
1 045789--
2 01245789
```

3- For each column of reference alignment (x)

- Mu= An array of length k2 initialized by 0 (or a dictionary)
- For character M in row r
  - Increment Mu[S[r][M]]
- Shared [x] =$\sum \binom{Mu_j}{2}$

FastSP

Reference:

```
  012345678
0 01234567-
1 0---12345
2 01-234567
```

Estimated:

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

Matrix S:

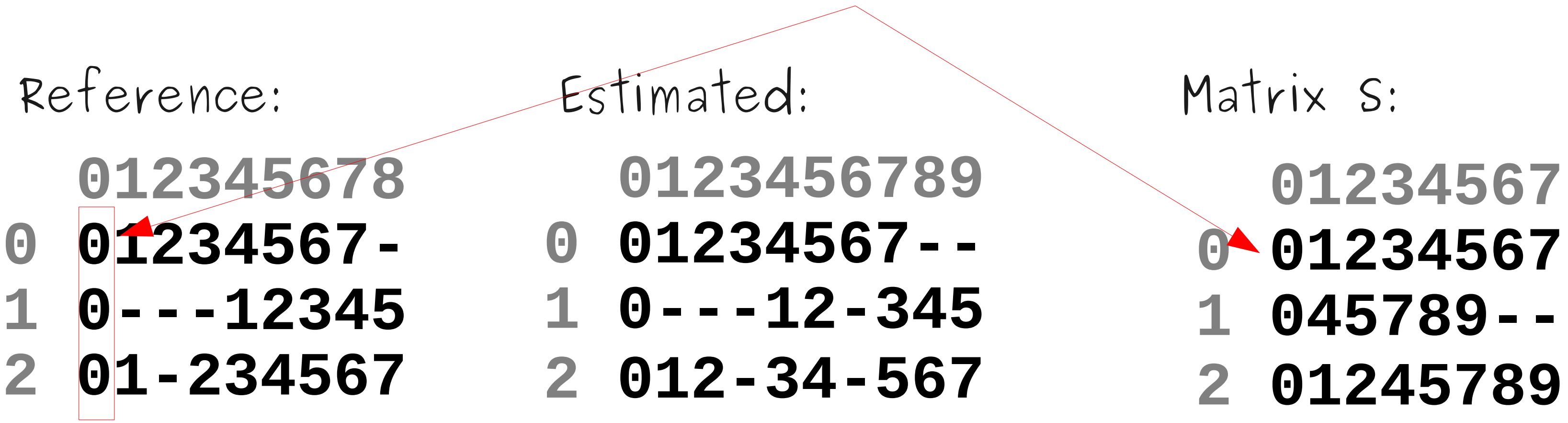
```
  01234567
0 01234567
1 045789--
2 01245789
```

Mu=[0 0 0 0 0 0 0 0 0 0]

3- For each column of reference alignment

- Mu= An array of length k2 initialized by 0 (or a dictionary)
- For character M in row r
  - Increment Mu[S[r][M]]
- Shared [x] =$\sum \binom{Mu_j}{2}$

FastSP



Reference:

```
  012345678
0 01234567-
1 0---12345
2 01-234567
```

Estimated:

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

Matrix S:

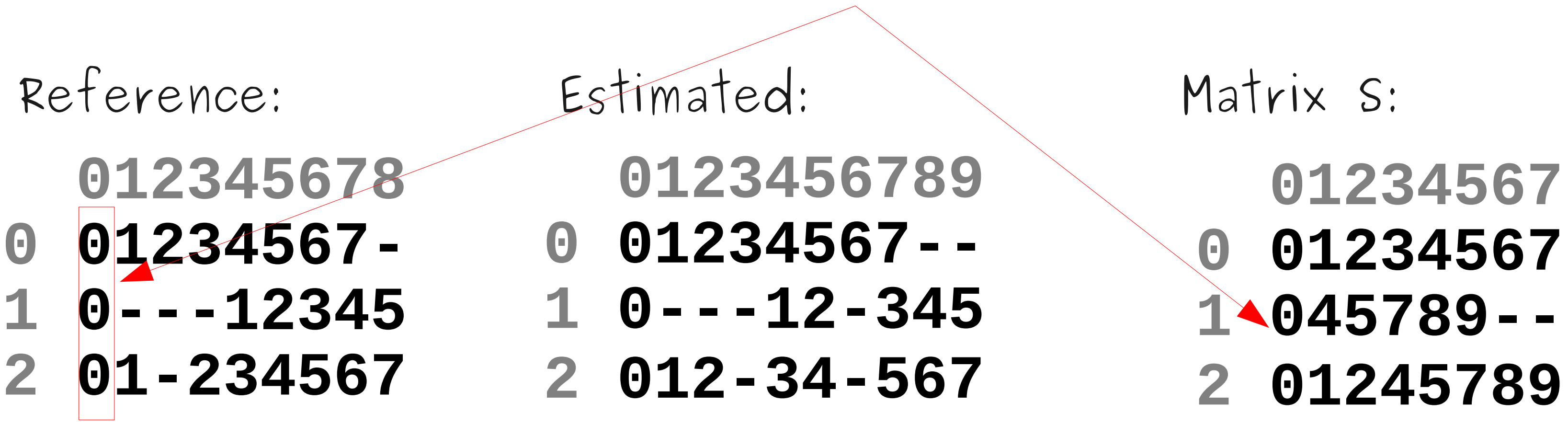
```
  01234567
0 01234567
1 045789--
2 01245789
```

Mu=[1 0 0 0 0 0 0 0 0 0]

3- For each column of reference alignment

- Mu= An array of length k2 initialized by 0 (or a dictionary)
- For character M in row r
  - Increment Mu[S[r][M]]
- Shared [x] =$\sum \binom{Mu_j}{2}$

FastSP



Reference:

```
  012345678
0 01234567-
1 0---12345
2 01-234567
```

Estimated:

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

Matrix S:

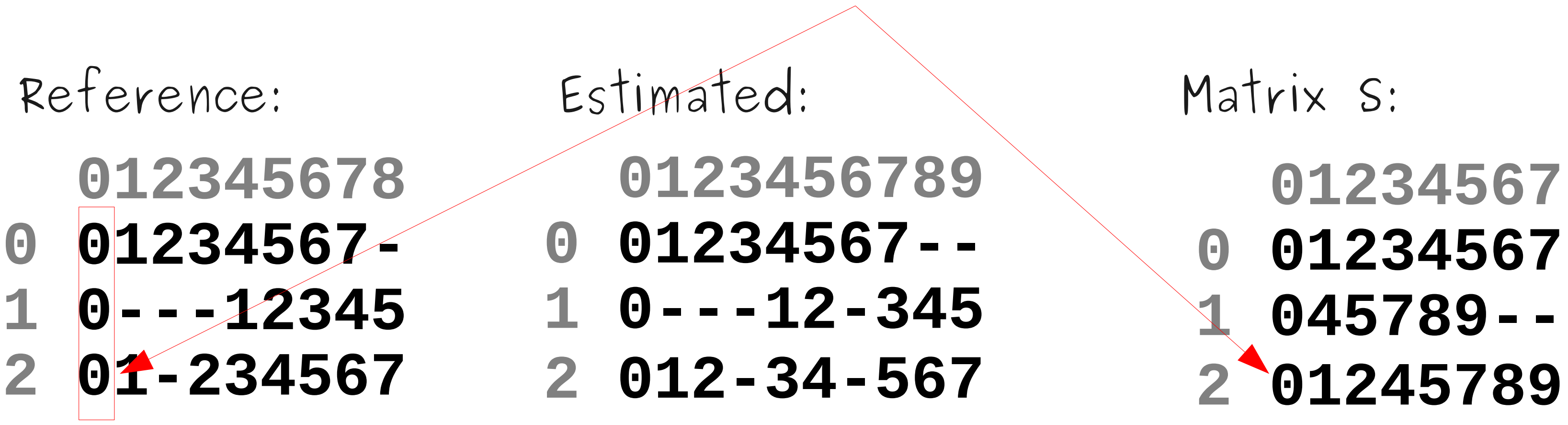
```
  01234567
0 01234567
1 045789--
2 01245789
```

Mu=[2 0 0 0 0 0 0 0 0 0]

3- For each column of reference alignment

- Mu= An array of length k2 initialized by 0 (or a dictionary)
- For character M in row r
  - Increment Mu[S[r][M]]
- Shared [x] =$\sum \binom{Mu_j}{2}$

FastSP



Reference:

```
  012345678
0 01234567-
1 0---12345
2 01-234567
```

Estimated:

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

Matrix S:

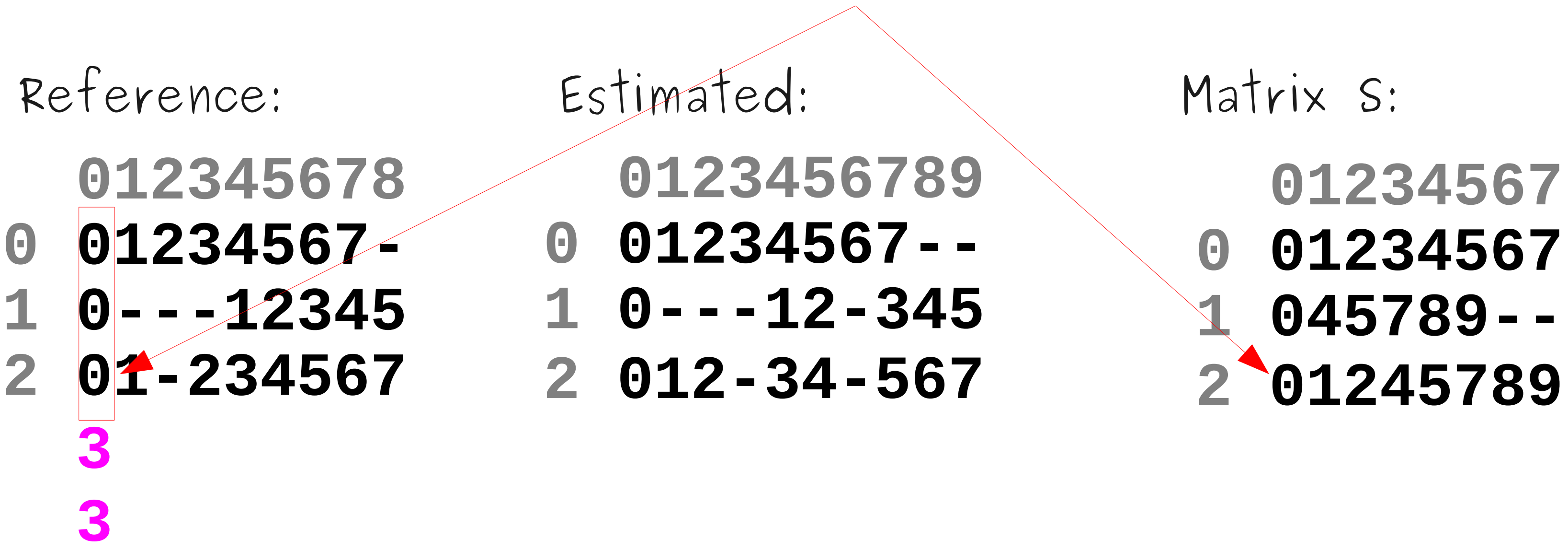
```
  01234567
0 01234567
1 045789--
2 01245789
```

Mu=[3 0 0 0 0 0 0 0 0 0]

3- For each column of reference alignment

- Mu= An array of length k2 initialized by 0 (or a dictionary)
- For character M in row r
  - Increment Mu[S[r][M]]
- Shared [x] $=\sum \binom{Mu_j}{2}$

FastSP



Reference:

```
  012345678
0 01234567-
1 0---12345
2 01-234567
  3
  3
```

Estimated:

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

Matrix S:

```
  01234567
0 01234567
1 045789--
2 01245789
```

Shared=$\binom{3}{2}+\binom{0}{2}+\binom{0}{2}+\binom{0}{2}+\binom{0}{2}+\binom{0}{2}+\ldots$ = 3   Mu=[3 0 0 0 0 0 0 0 0 0]

3- For each column of reference alignment

- Mu= An array of length k2 initialized by 0 (or a dictionary)
- For character M in row r
  - Increment Mu[S[r][M]]
- Shared [x] =$\sum \binom{Mu_j}{2}$

FastSP

Reference:

```
  012345678
0 01234567-
1 0---12345
2 01-234567
  31
  31
```

Estimated:

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

Matrix S:

```
  01234567
0 01234567
1 045789--
2 01245789
```

$\text{Shared}=\binom{0}{2}+\binom{2}{2}+\binom{0}{2}+\binom{0}{2}+\binom{0}{2}+\binom{0}{2}+\binom{0}{2}=1$

Mu=[0 2 0 0 0 0 0 0 0 0]

3- For each column of reference alignment

- Mu= An array of length k2 initialized by 0 (or a dictionary)
- For character M in row r
  - Increment Mu[S[r][M]]
- Shared [x]= $\sum \binom{Mu_j}{2}$

FastSP

Reference:

```
  012345678
0 01234567-
1 0---12345
2 01-234567
  310
  310
```

Estimated:

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

Matrix S:

```
  01234567
0 01234567
1 045789--
2 01245789
```

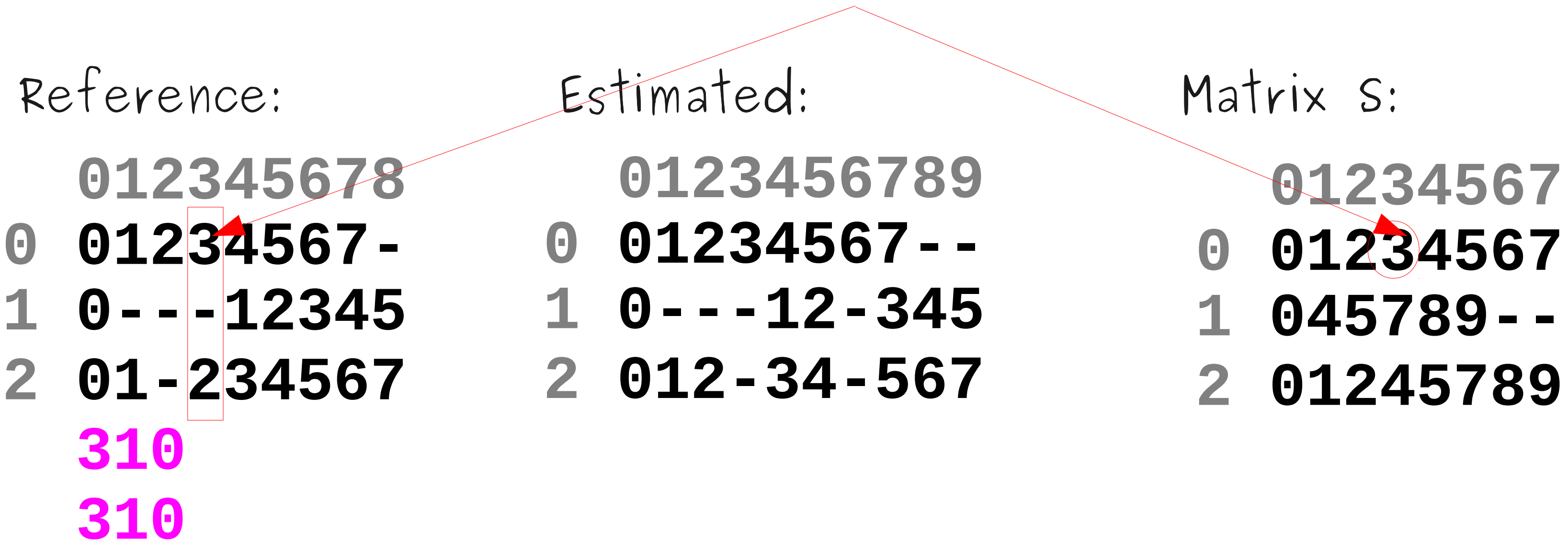
Shared= $\binom{0}{2}+\binom{0}{2}+\binom{1}{2}+\binom{0}{2}+\binom{0}{2}+\binom{0}{2}+\binom{0}{2}=0$

Mu=[0 0 1 0 0 0 0 0 0 0]

3- For each column of reference alignment

- Mu= An array of length k2 initialized by 0 (or a dictionary)
- For character M in row r
  - Increment Mu[S[r][M]]
- Shared [x]= $\sum\binom{Mu_j}{2}$

FastSP



Reference:

```
  012345678
0 01234567-
1 0---12345
2 01-234567
  310
  310
```

Estimated:

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

Matrix S:

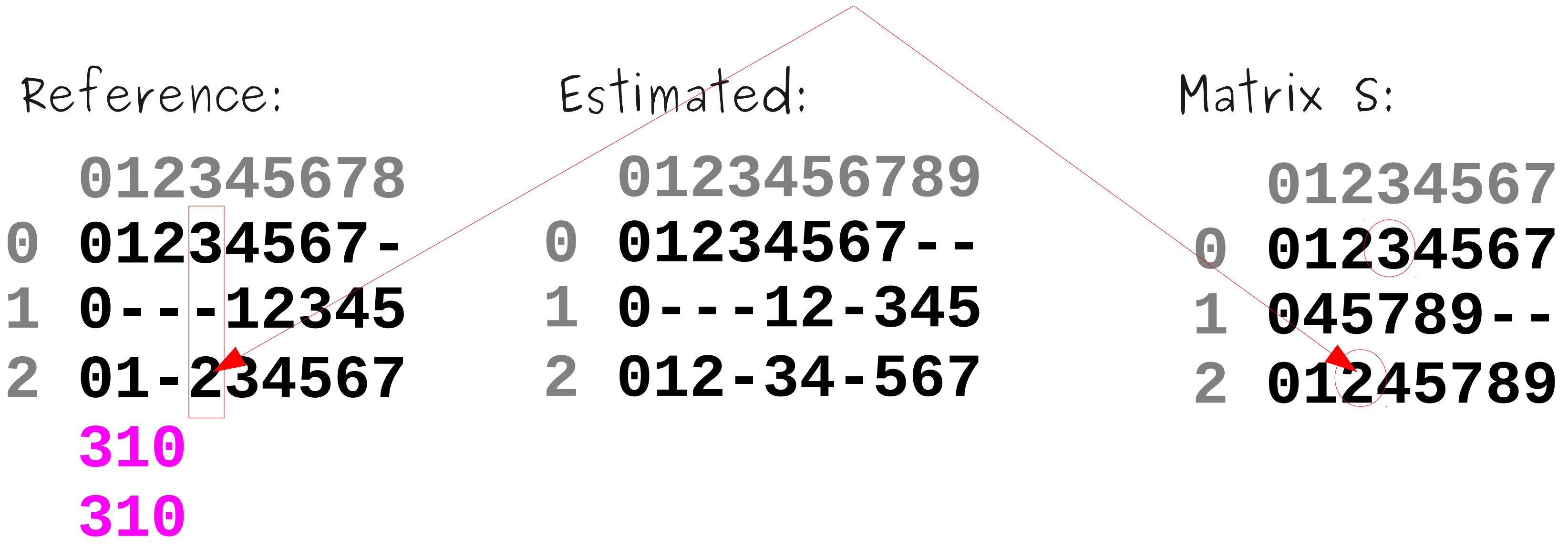
```
  01234567
0 01234567
1 045789--
2 01245789
```

Mu=[0 0 0 1 0 0 0 0 0 0]

3- For each column of reference alignment

- Mu= An array of length k2 initialized by 0 (or a dictionary)
- For character M in row r
  - Increment Mu[S[r][M]]
- Shared [x]= $\sum \binom{Mu_j}{2}$

FastSP



Reference:

```
  012345678
0 01234567-
1 0---12345
2 01-234567
  310
  310
```

Estimated:

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

Matrix S:

```
  01234567
0 01234567
1 045789--
2 01245789
```

Mu=[0 0 1 1 0 0 0 0 0 0]

3- For each column of reference alignment

- Mu= An array of length k2 initialized by 0 (or a dictionary)
- For character M in row r
  - Increment Mu[S[r][M]]
- Shared [x]= $\sum \binom{Mu_j}{2}$

FastSP

Reference:

```
  012345678
0 01234567-
1 0---12345
2 01-234567
  3100
  3101
```

Estimated:

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

Matrix S:

```
  01234567
0 01234567
1 045789--
2 01245789
```

Shared=$\binom{0}{2}+\binom{0}{2}+\binom{1}{2}+\binom{1}{2}+\binom{0}{2}+\binom{0}{2}+\dots$ = 0

Mu=[0 0 1 1 0 0 0 0 0 0]

3- For each column of reference alignment

- Mu= An array of length k2 initialized by 0 (or a dictionary)
- For character M in row r
  - Increment Mu[S[r][M]]
- Shared [x]= $\sum \binom{Mu_j}{2}$

FastSP

Reference:

```
  012345678
0 01234567-
1 0---12345
2 01-234567
  31003
  31013
```

Estimated:

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

Matrix S:

```
  01234567
0 01234567
1 045789--
2 01245789
```

Mu=[0 0 0 0 3 0 0 0 0 0]

3- For each column of reference alignment

- Mu= An array of length k2 initialized by 0 (or a dictionary)
- For character M in row r
  - Increment Mu[S[r][M]]
- Shared [x]= $\sum \binom{Mu_j}{2}$

FastSP

Reference:

```
  012345678
0 01234567-
1 0---12345
2 01-234567
  310033
  310133
```

Estimated:

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

Matrix S:

```
  01234567
0 01234567
1 045789--
2 01245789
```

Mu=[0 0 0 0 0 3 0 0 0 0]

3- For each column of reference alignment

- Mu= An array of length k2 initialized by 0 (or a dictionary)
- For character M in row r
  - Increment Mu[S[r][M]]
- Shared [x]= $\sum \binom{Mu_j}{2}$

FastSP

Reference:

```
  012345678
0 01234567-
1 0---12345
2 01-234567
  3100331
  3101333
```

Estimated:

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

Matrix S:

```
  01234567
0 01234567
1 045789--
2 01245789
```

Shared= $\ldots+\binom{0}{2}+\binom{0}{2}+\binom{0}{2}+\binom{1}{2}+\binom{2}{2}+\ldots$ = 1

Mu=[0 0 0 0 0 0 1 2 0 0]

3- For each column of reference alignment

- Mu= An array of length k2 initialized by 0 (or a dictionary)
- For character M in row r
  - Increment Mu[S[r][M]]
- Shared [x]= $\sum \binom{Mu_j}{2}$

FastSP

Reference:

```
  012345678
0 01234567-
1 0---12345
2 01-234567
  31003311
  31013333
```

Estimated:

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

Matrix S:

```
  01234567
0 01234567
1 045789--
2 01245789
```

Shared= $\ldots+\binom{0}{2}+\binom{0}{2}+\binom{0}{2}+\binom{1}{2}+\binom{2}{2}+\ldots$ = 1

Mu=[0 0 0 0 0 0 0 1 2 0]

3- For each column of reference alignment

- Mu= An array of length k2 initialized by 0 (or a dictionary)
- For character M in row r
  - Increment Mu[S[r][M]]
- Shared [x]= $\sum\binom{Mu_j}{2}$

FastSP

Reference:

```
  012345678
0 01234567-
1 0---12345
2 01-234567
  310033111
  310133331
```

Estimated:

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

Matrix S:

```
  01234567
0 01234567
1 045789--
2 01245789
```

Shared=$\ldots+\binom{0}{2}+\binom{0}{2}+\binom{0}{2}+\binom{0}{2}+\binom{2}{2}$ = 1

Mu=[0 0 0 0 0 0 0 0 0 2]

3- For each column of reference alignment

- Mu= An array of length k2 initialized by 0 (or a dictionary)
- For character M in row r
  - Increment Mu[S[r][M]]
- Shared [x]= $\sum \binom{Mu_j}{2}$

FastSP

Reference:

```
  012345678
0 01234567-
1 0---12345
2 01-234567
  310033111=13
  310133331=18
```

Estimated:

```
  0123456789
0 01234567--
1 0---12-345
2 012-34-567
```

Matrix S:

```
  01234567
0 01234567
1 045789--
2 01245789
```

SP-Score=13/18

4- Report (sum of shared)/(sum of reference) as SP-Score

Running Time Analysis

1- read reference alignment and save it with our character representation

O(n.k1)

2- read estimated alignment and create a n by k matrix S such that

- S[i,j]=x iff Estimated_Alignment[i,x]=j.

O(n.k2)

3- For each column of reference alignment (k1)

- Mu= An array of length k2 initialized by 0 (or a dictionary)
- For character M in row r (n)
  - Increment Mu[S[r][M]]
- Shared [x]= $\sum \binom{Mu_j}{2}$

O(n.k1)

4- Report (sum of shared)/(sum of reference)

O(k1)

Overall=O(max(k1,k2).n)

Memory Analysis

1- read reference alignment and save it with our character representation

O(n.k1)

2- read estimated alignment and create a n by k matrix S such that

- S[i,j]=x iff Estimated_Alignment[i,x]=j.

O(n.k)

3- For each column of reference alignment

- Mu= An array of length k2 initialized by 0 (or a dictionary) O(k2) or O(n)
- For character M in row r
    - Increment Mu[S[r][M]]
- Shared [x]= $\sum \binom{Mu_j}{2}$

4- Report (sum of shared)/(sum of reference)

Overall=O((k1+k).n)

Memory Analysis

- Trick: choose smallest of k1 and k2 as reference alignment and the other as estimated alignment. Number of shared homologies will be the same either way.

Overall=$O((\min(k1,k2)+k).n)$

Modeler and TC scores

- Both Modeler and TC scores can be calculated with FastSP algorithm without any sacrifice to running time and memory
- Modeler: we just need to calculate total number of homologies in the estimated alignment
- TC: As we go through column of reference alignment, if we get only one equivalence class, we have a correct column.

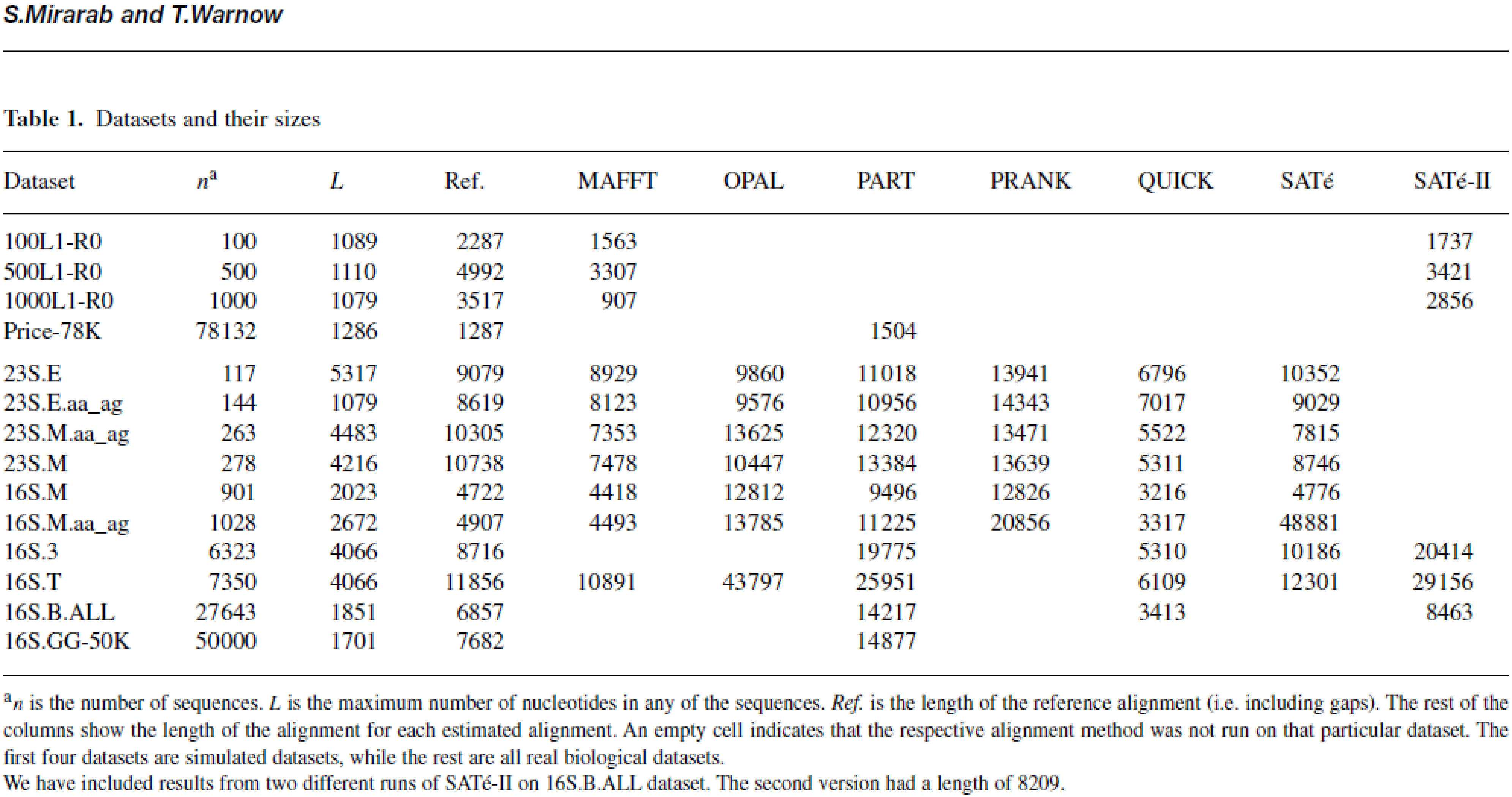
Implementation

- Implemented in Java
  - Computes SP, Modeler, and TC in one run

- 420 LOC

- Available publicly at http://www.cs.utexas.edu/~phylo/software/fastsp/

Performance Study: datasets



S.Mirarab and T.Warnow

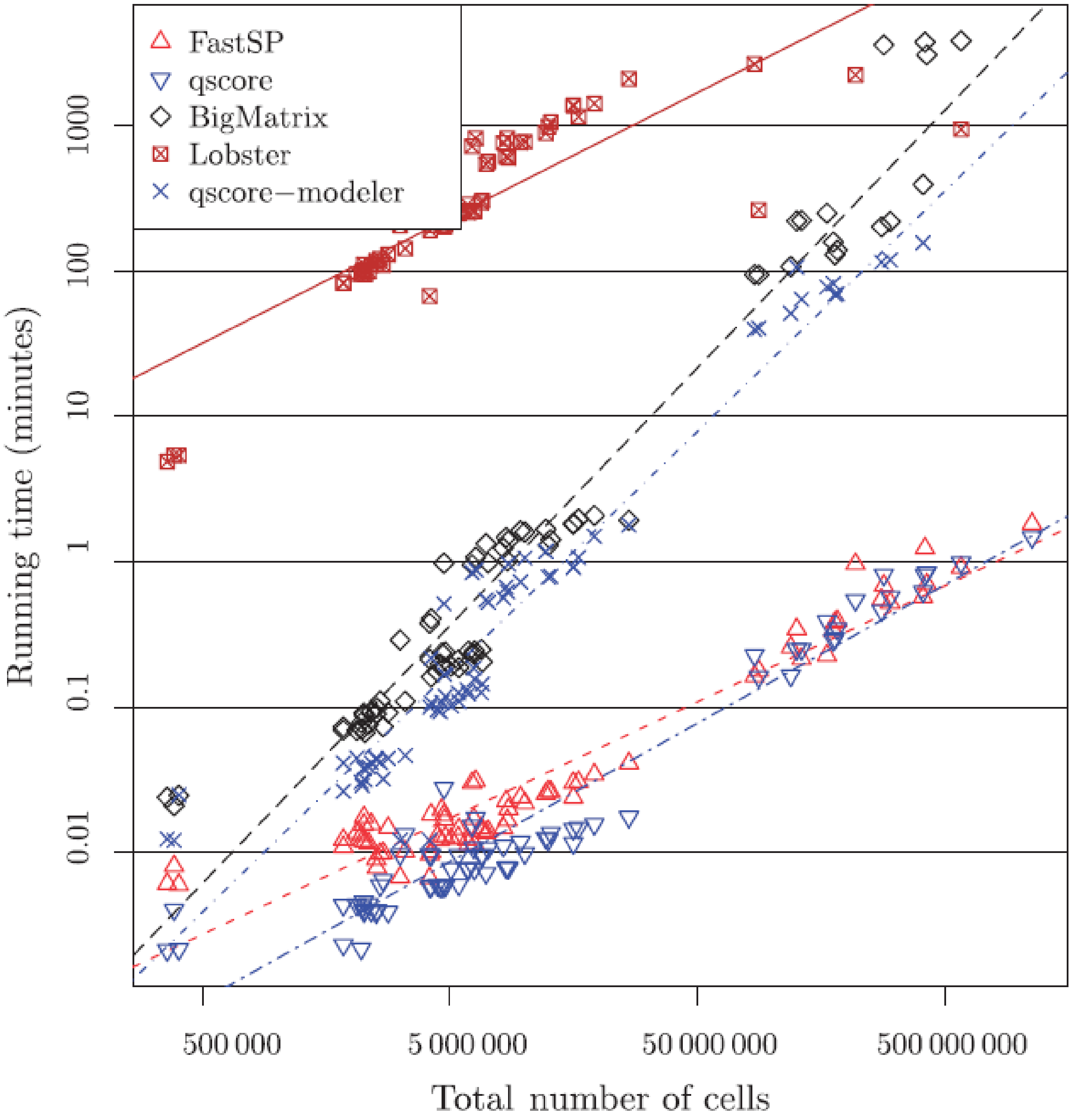
Table 1. Datasets and their sizes

Dataset	*n*[a]	*L*	Ref.	MAFFT	OPAL	PART	PRANK	QUICK	SATé	SATé-II
100L1-R0	100	1089	2287	1563						1737
500L1-R0	500	1110	4992	3307						3421
1000L1-R0	1000	1079	3517	907						2856
Price-78K	78132	1286	1287			1504				
23S.E	117	5317	9079	8929	9860	11018	13941	6796	10352	
23S.E.aa_ag	144	1079	8619	8123	9576	10956	14343	7017	9029	
23S.M.aa_ag	263	4483	10305	7353	13625	12320	13471	5522	7815	
23S.M	278	4216	10738	7478	10447	13384	13639	5311	8746	
16S.M	901	2023	4722	4418	12812	9496	12826	3216	4776	
16S.M.aa_ag	1028	2672	4907	4493	13785	11225	20856	3317	48881	
16S.3	6323	4066	8716			19775		5310	10186	20414
16S.T	7350	4066	11856	10891	43797	25951		6109	12301	29156
16S.B.ALL	27643	1851	6857			14217		3413		8463
16S.GG-50K	50000	1701	7682			14877				

[a] *n* is the number of sequences. *L* is the maximum number of nucleotides in any of the sequences. *Ref.* is the length of the reference alignment (i.e. including gaps). The rest of the columns show the length of the alignment for each estimated alignment. An empty cell indicates that the respective alignment method was not run on that particular dataset. The first four datasets are simulated datasets, while the rest are all real biological datasets.

We have included results from two different runs of SATé-II on 16S.B.ALL dataset. The second version had a length of 8209.

Performance Study: All Techniques



Performance Study: Q-Score

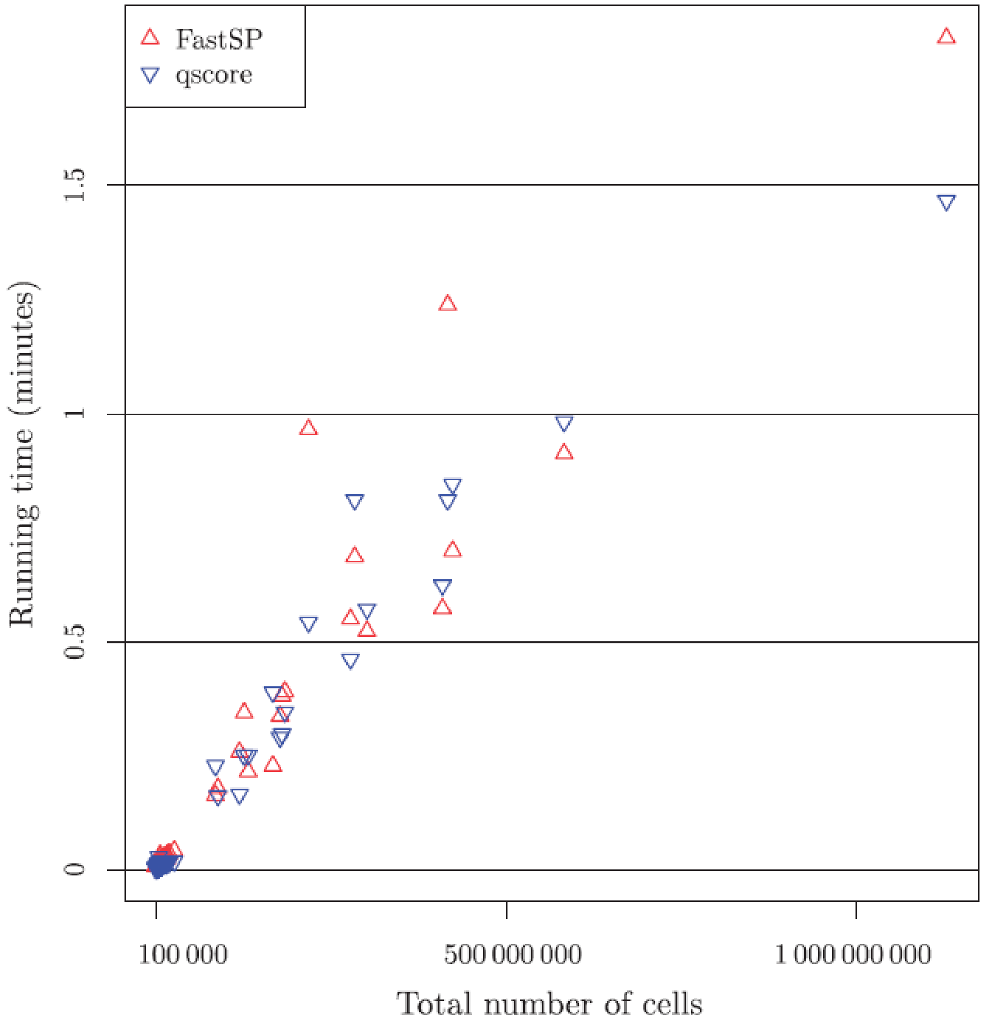


Fig. 4. Comparison of FastSP and qscore-default with respect to running time on machines with 'at least' 8 GB of main memory.

Performance Study: Memory

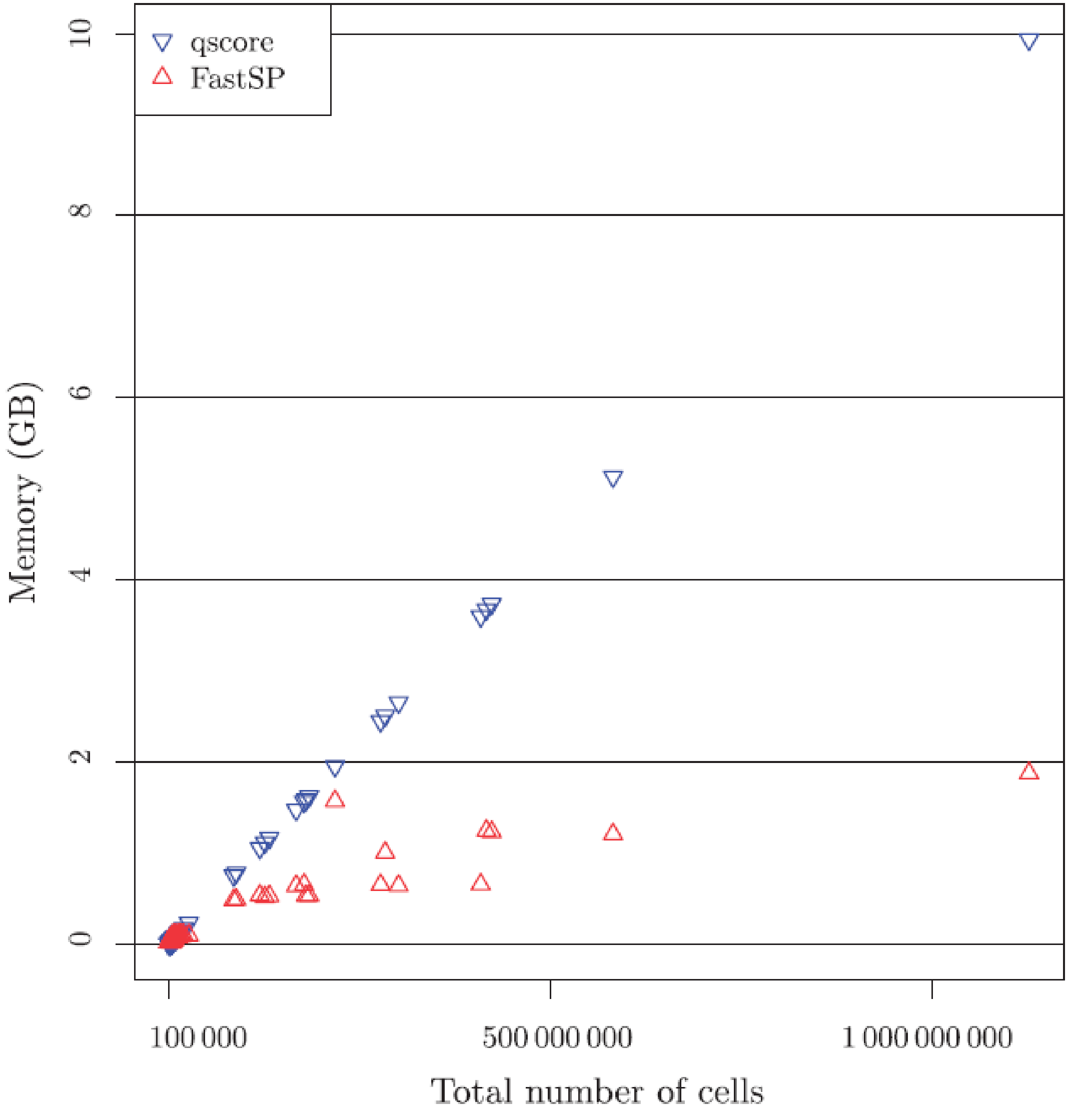


Fig. 5. Comparison of FastSP and qscore-default with respect to peak memory usage on machines with 'at least' 8 GB of main memory.

Performance Study: Limited Memory

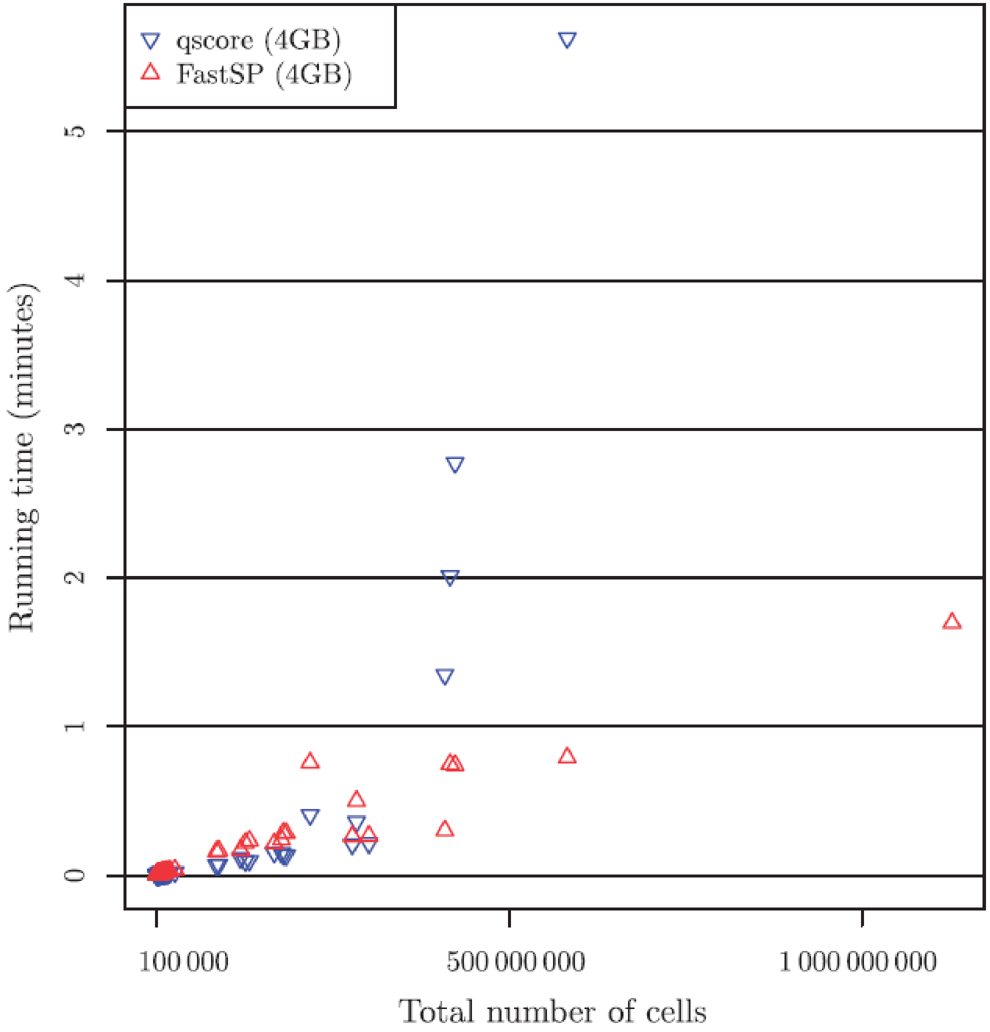


Fig. 1. Log-scaled running time on machines with 4 GB of main memory. QSCORE is run only in default setting, and so computes only SP- and TC-scores. Note that QSCORE fails to analyze the largest dataset.

Summary

- Two alignments can be compared in terms of SP-Score, Modeler Score, and TC in linear time (linear with respect to k.n)

- FastSP provides a memory-efficient tool for comparing alignments