Genome 540 Discussion

February 13th, 2024

Clifford Rostomily

Genome Sciences
UNIVERSITY OF WASHINGTON

Assignment 5 Questions?

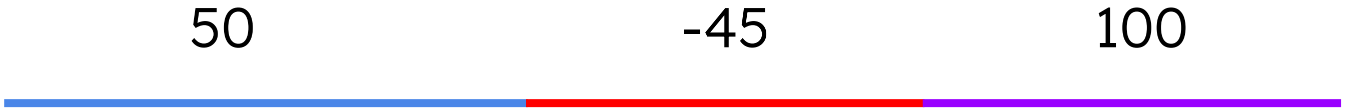
- Part 1
  - Build a weighted edit graph for 3 amino acid sequences of the insulin protein (human, frog, water buffalo) using the BLOSUM62 scoring matrix and save it as a text file
- Part 2:
  - Use your program from HW4 to find the max weight path through the edit graph

Assignment 6

Overview

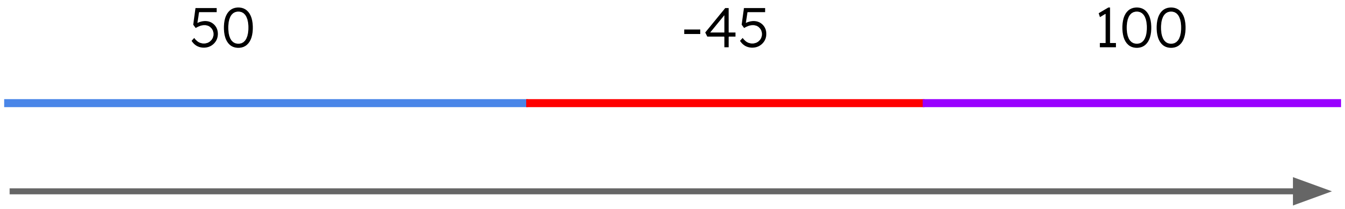
- Write a program to identify regions of elevated copy-number using the D-segment algorithm
- Run the program on chromosome 16 from individual CHM13

D-segment motivation



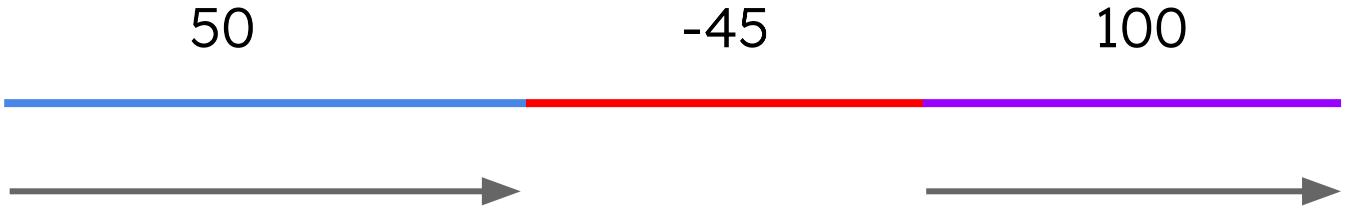
50 -45 100

D-segment motivation



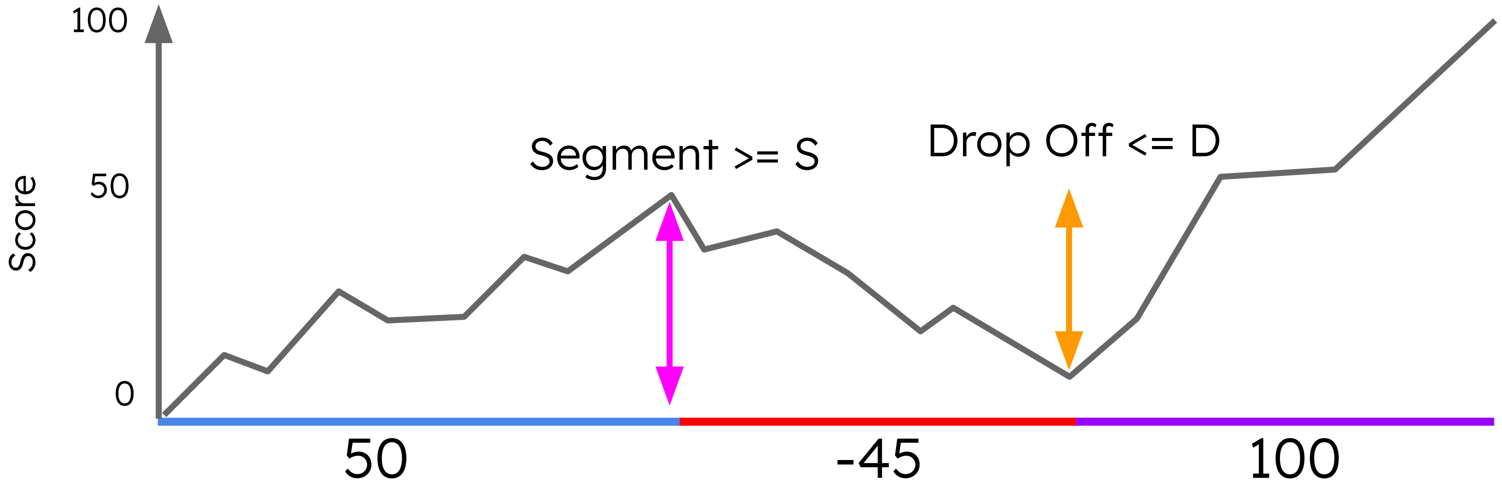
Whole region has a score of 105.

D-segment motivation



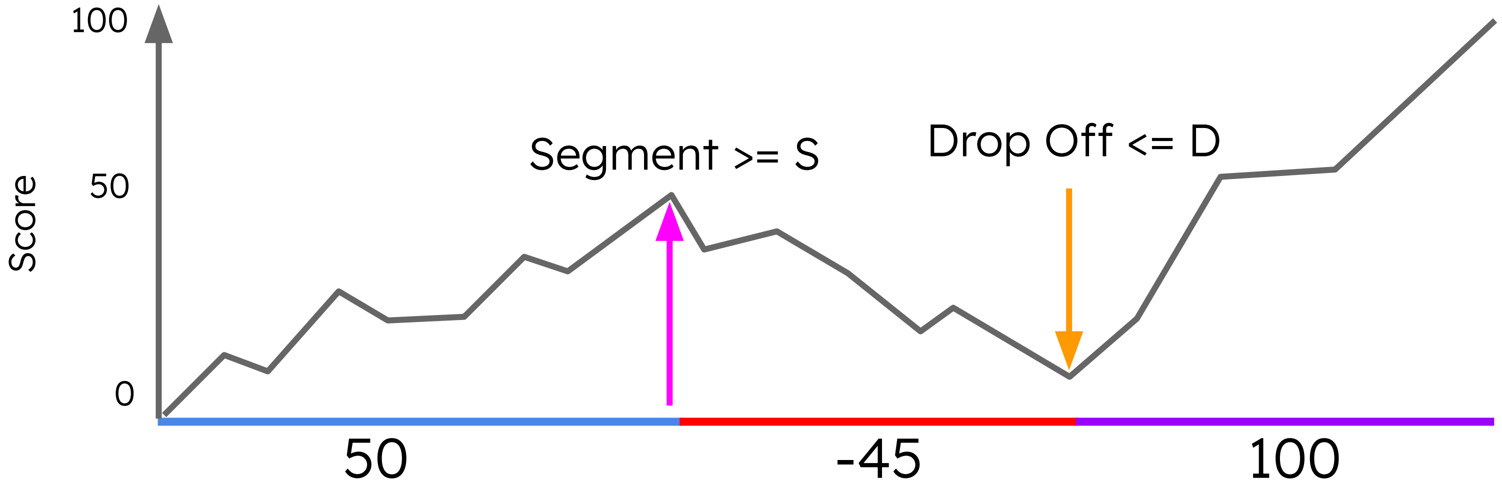
However, these two sub-segments may represent biologically distinct events...

D-segment algorithm



What values of S and D would separate these segments?

D-segment algorithm

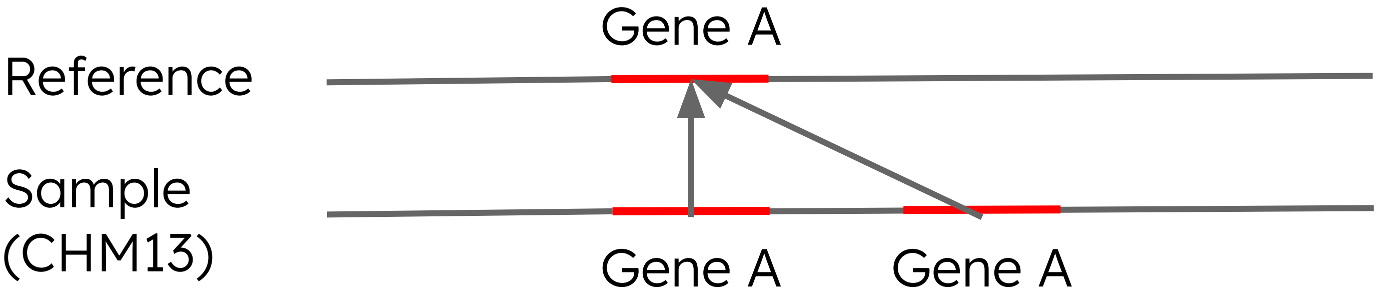


What values of S and D would separate these segments?

S <= 50 and D >= -45

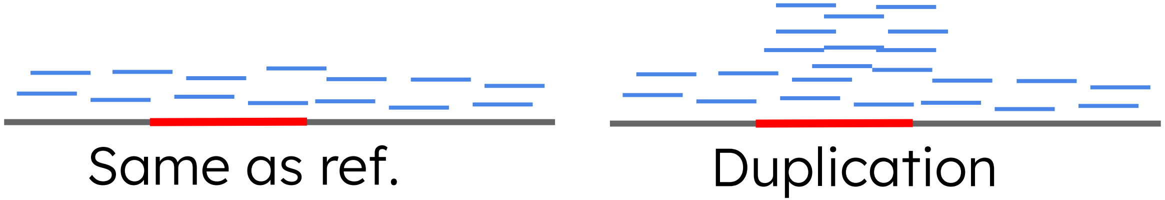
*** D would probably have to be less than -10 as well

Copy Number Variation

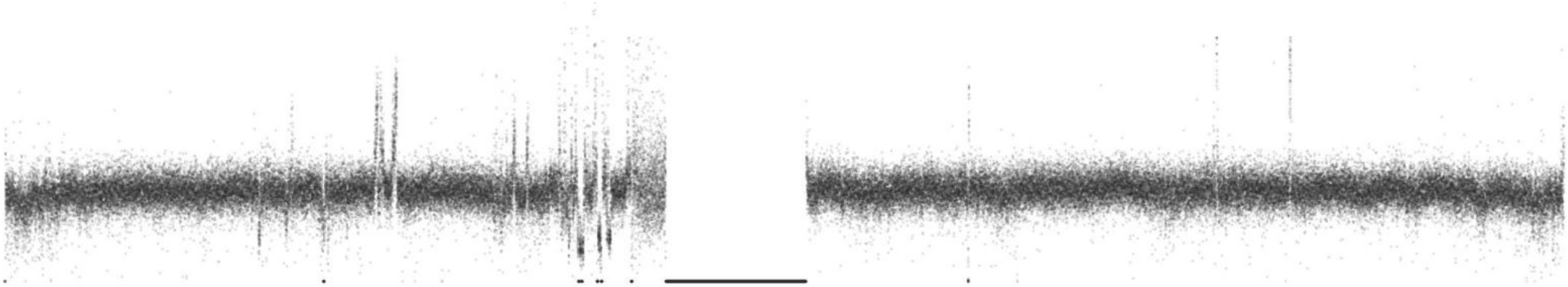


Both copies mapped to same reference allele

NGS read coverage will be higher for that gene when mapped to reference



Data - Read Start Counts



Position
(chr16)

Convert Counts to Scores

- Background:
  - m = mean(counts starts)
  - count = counts at a position
  - B ~ Poisson(m)
  - L(B|count) = P(count| B)
- Heterozygous duplication:
  - D ~ Poisson(1.5*m)
  - L(D|count) = P(count| D)
- Score
  - Score = log2(LR(L(D|count)/L(B|count)))

Pseudocode

O(*N*) algorithm to find all maximal D-segs:

```
cumul = max = 0; start = 1;
for (i = 1; i ≤ N; i++) {
    cumul += s[i];
    if (cumul ≥ max)
        {max = cumul; end = i;}
    if (cumul ≤ 0 or cumul ≤ max + D or i == N) {
        if (max ≥ S)
           {print start, end, max; }
        max = cumul = 0; start = end = i + 1; /* NO BACKTRACKING
           NEEDED! */
    }
}
```

Example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	0	1	2	0	2	0	0
Score	-0.5	-0.5	0.52	1.1	-0.5	1.1	-0.5	-0.5

D = -3

S = 3

max = 0

start = 1

end = 1

cumul = 0

$O(N)$ algorithm to find all maximal D-segs:

```
cumul = max = 0; start = 1;
for (i = 1; $i \leq N$; i++) {
    cumul += s[i];
    if (cumul $\geq$ max)
        {max = cumul; end = i;}
    if (cumul $\leq$ 0 or cumul $\leq$ max + D or i == N) {
        if (max $\geq$ S)
          {print start, end, max; }
        max = cumul = 0; start = end = i + 1; /* NO BACKTRACKING
          NEEDED! */
    }
}
```

Example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	1	2	2	0	2	0	0
Score	-0.5	0.52	1.1	1.1	-0.5	1.1	-0.5	-0.5

D = -3

S = 3

max = 0

start = 1

end = 1

cumul = 0

O(*N*) algorithm to find all maximal D-segs:

```
cumul = max = 0; start = 1;
for (i = 1; $i \le N$; i++) {
    cumul += s[i];
    if ($\text{cumul} \ge \text{max}$)
        {max = cumul; end = i;}
    if ($\text{cumul} \le 0$ or $\text{cumul} \le \text{max} + D$ or i == N) {
        if ($\text{max} \ge S$)
            {print start, end, max; }
        max = cumul = 0; start = end = i + 1; /* NO BACKTRACKING
            NEEDED! */
    }
}
```

Example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	1	2	2	0	2	0	0
Score	-0.5	0.52	1.1	1.1	-0.5	1.1	-0.5	-0.5

D = -3
S = 3
max = 0.52
start = 2
end = 2
cumul = 0.52

$O(N)$ algorithm to find all maximal D-segs:

```
cumul = max = 0; start = 1;
for (i = 1; i ≤ N; i++) {
    cumul += s[i];
    if (cumul ≥ max)
        {max = cumul; end = i;}
    if (cumul ≤ 0 or cumul ≤ max + D or i == N) {
        if (max ≥ S)
          {print start, end, max; }
        max = cumul = 0; start = end = i + 1; /* NO BACKTRACKING
          NEEDED! */
    }
}
```

Example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	1	2	2	0	2	0	0
Score	-0.5	0.52	1.1	1.1	-0.5	1.1	-0.5	-0.5

D = -3
S = 3
max = 1.62
start = 2
end = 3
cumul = 1.62

$O(N)$ algorithm to find all maximal D-segs:

```
cumul = max = 0; start = 1;
for (i = 1; i ≤ N; i++) {
    cumul += s[i];
    if (cumul ≥ max)
        {max = cumul; end = i;}
    if (cumul ≤ 0 or cumul ≤ max + D or i == N) {
        if (max ≥ S)
            {print start, end, max; }
        max = cumul = 0; start = end = i + 1; /* NO BACKTRACKING
          NEEDED! */
    }
}
```

Example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	1	2	2	0	2	0	0
Score	-0.5	0.52	1.1	1.1	-0.5	1.1	-0.5	-0.5

D = -3
S = 3
max = 2.72
start = 2
end = 4
cumul = 2.72

$O(N)$ algorithm to find all maximal D-segs:

```
cumul = max = 0; start = 1;
for (i = 1; i ≤ N; i++) {
    cumul += s[i];
    if (cumul ≥ max)
        {max = cumul; end = i;}
    if (cumul ≤ 0 or cumul ≤ max + D or i == N) {
        if (max ≥ S)
          {print start, end, max; }
        max = cumul = 0; start = end = i + 1; /* NO BACKTRACKING
          NEEDED! */
    }
}
```

Example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	1	2	2	0	2	0	0
Score	-0.5	0.52	1.1	1.1	-0.5	1.1	-0.5	-0.5

D = -3

S = 3

max = 2.22

start = 2

end = 4

cumul = 2.22

$O(N)$ algorithm to find all maximal D-segs:

```
cumul = max = 0; start = 1;
for (i = 1; i ≤ N; i++) {
    cumul += s[i];
    if (cumul ≥ max)
        {max = cumul; end = i;}
    if (cumul ≤ 0 or cumul ≤ max + D or i == N) {
        if (max ≥ S)
          {print start, end, max; }
        max = cumul = 0; start = end = i + 1; /* NO BACKTRACKING
          NEEDED! */
    }
}
```

Example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	1	2	2	0	2	0	0
Score	-0.5	0.52	1.1	1.1	-0.5	1.1	-0.5	-0.5

D = -3

S = 3

max = 3.32

start = 2

end = 6

cumul = 3.32

$O(N)$ algorithm to find all maximal D-segs:

```
cumul = max = 0; start = 1;
for (i = 1; $i \le N$; i++) {
    cumul += s[i];
    if ($cumul \ge max$)
        {max = cumul; end = i;}
    if ($cumul \le 0$ or $cumul \le max + D$ or i == N) {
        if ($max \ge S$)
           {print start, end, max; }
        max = cumul = 0; start = end = i + 1; /* NO BACKTRACKING
           NEEDED! */
    }
}
```

Example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	1	2	2	0	2	0	0
Score	-0.5	0.52	1.1	1.1	-0.5	1.1	-0.5	-0.5

D = -3

S = 3

max = 3.32

start = 1

end = 1

cumul = 2.82

$O(N)$ algorithm to find all maximal D-segs:

```
cumul = max = 0; start = 1;
for (i = 1; i ≤ N; i++) {
    cumul += s[i];
    if (cumul ≥ max)
        {max = cumul; end = i;}
    if (cumul ≤ 0 or cumul ≤ max + D or i == N) {
        if (max ≥ S)
          {print start, end, max; }
        max = cumul = 0; start = end = i + 1; /* NO BACKTRACKING
          NEEDED! */
    }
}
```

Example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	1	2	2	0	2	0	0
Score	-0.5	0.52	1.1	1.1	-0.5	1.1	-0.5	-0.5

D = -3

S = 3

max = 3.32

start = 2

end = 6

cumul = 2.32

O(*N*) algorithm to find all maximal D-segs:

```
cumul = max = 0; start = 1;
for (i = 1; i ≤ N; i++) {
    cumul += s[i];
    if (cumul ≥ max)
        {max = cumul; end = i;}
    if (cumul ≤ 0 or cumul ≤ max + D or i == N) {
        if (max ≥ S)
          {print start, end, max; }
        max = cumul = 0; start = end = i + 1; /* NO BACKTRACKING
          NEEDED! */
    }
}
```

Example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	1	2	2	0	2	0	0
Score	-0.5	0.52	1.1	1.1	-0.5	1.1	-0.5	-0.5

D = -3
S = 3
max = 3.32
start = 2
end = 6
cumul = 2.32

$O(N)$ algorithm to find all maximal D-segs:

```
cumul = max = 0; start = 1;
for (i = 1; i ≤ N; i++) {
    cumul += s[i];
    if (cumul ≥ max)
        {max = cumul; end = i;}
    if (cumul ≤ 0 or cumul ≤ max + D or i == N) {
        if (max ≥ S)
          {print start, end, max; }
        max = cumul = 0; start = end = i + 1; /* NO BACKTRACKING
          NEEDED! */
    }
}
```

Reminders

- HW6 due this Sunday, 11:59pm
- Please have your name in the filename of your homework assignment and match the template