Genome 540 discussion

February 18th, 2025

Joe Min

Genome Sciences
UNIVERSITY OF WASHINGTON

Agenda

Homework 6 overview

Poisson distributions

Homework 6

Homework 6 overview

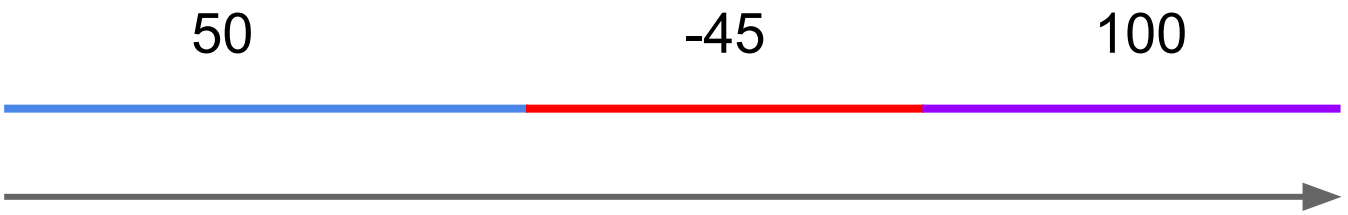
1. Implement the D-segment algorithm to identify regions of elevated copy number variation (CNVs)
2. Run your program on chromosome 16 of the CHM13 long-read-based genome assembly

D-segment review

Goals:

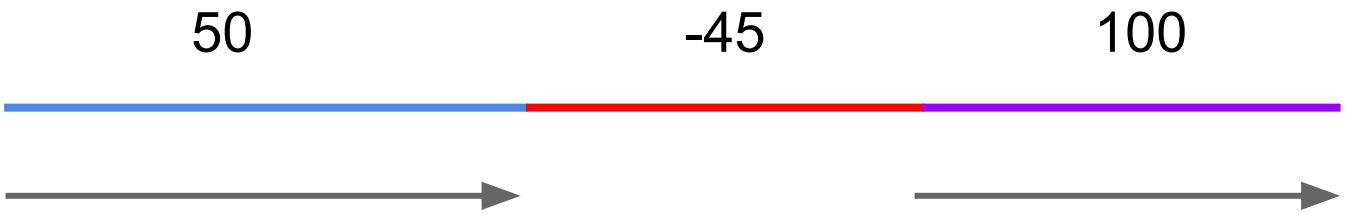
- Identify all high-scoring regions of the genome according to some scoring scheme
- In O(n) time, identify boundaries of these regions by identifying regions with large “drops” in score

D-segment review



Whole region has total score of 105 but…

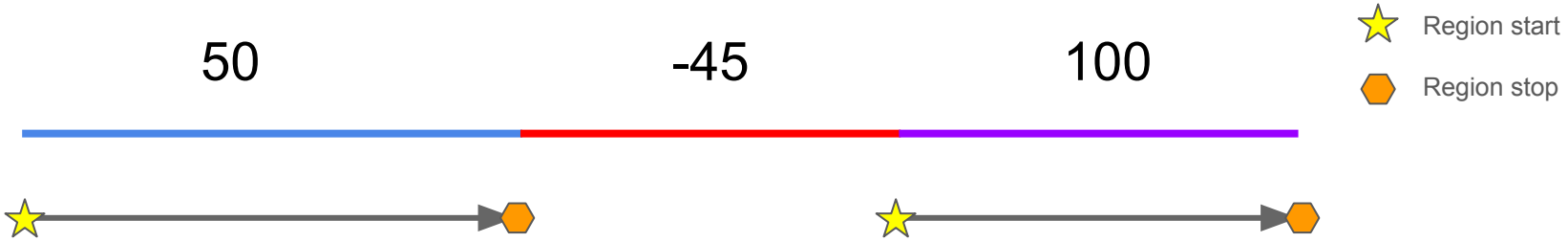
D-segment review



Whole region has total score of 105 but…

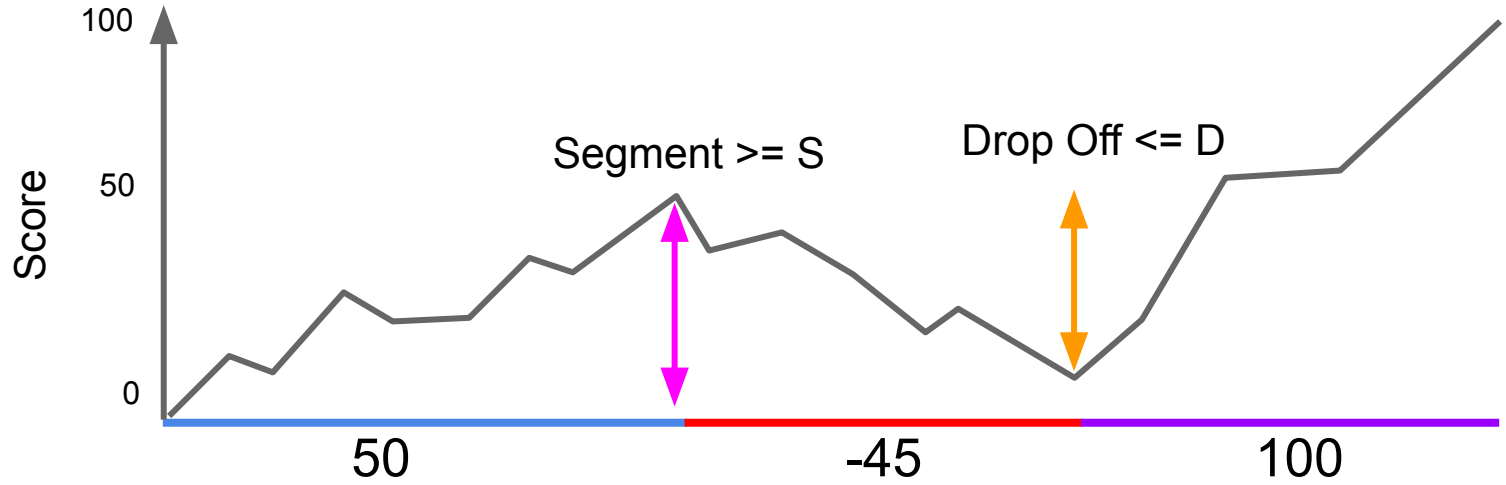
Discontinuous high-scoring regions might have different biological significance

D-segment review



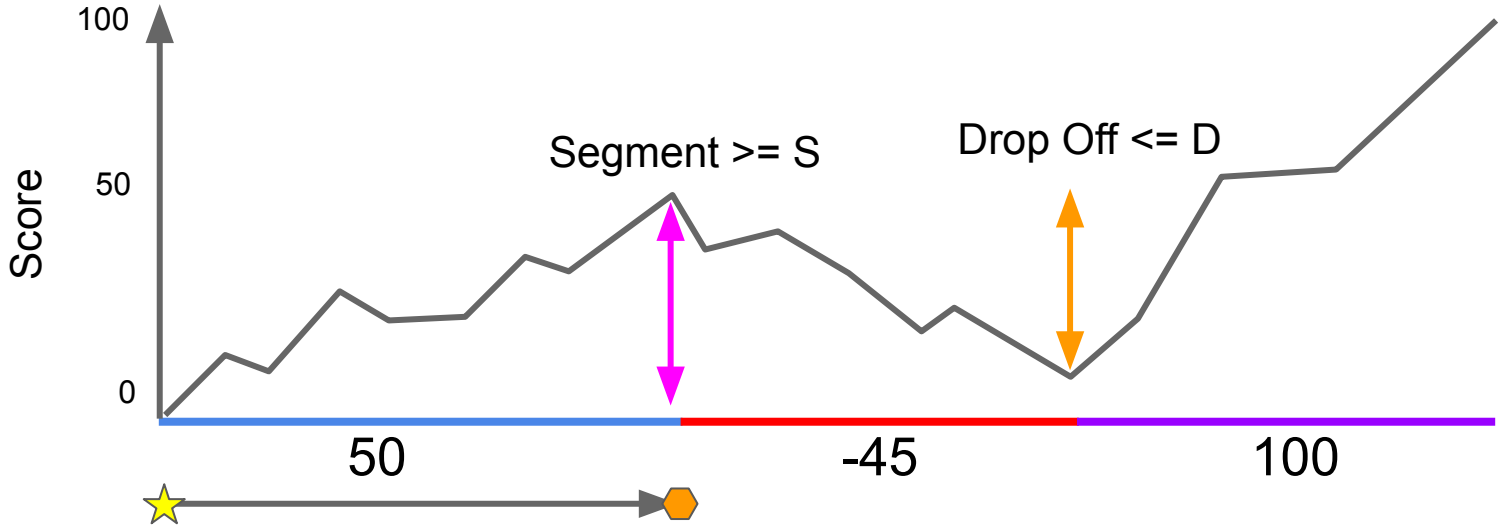
To find *all* high-scoring segments (i.e., all regions with score >= S), we could keep track of region best starts and stops to maximize region scores but this requires going backwards, which could be O(n²) in the worst case

D-segment review



We can instead keep track of a region's maximum cumulative score, as well as a cumulative drop from that to identify the region's ending boundary

D-segment review



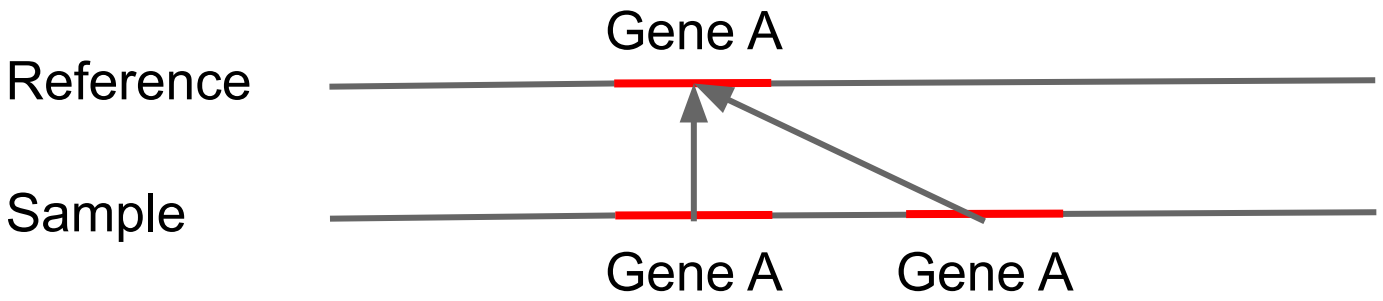
We can instead keep track of a region's maximum cumulative score, as well as a cumulative drop from that to identify the region's ending boundary

D-segment review

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

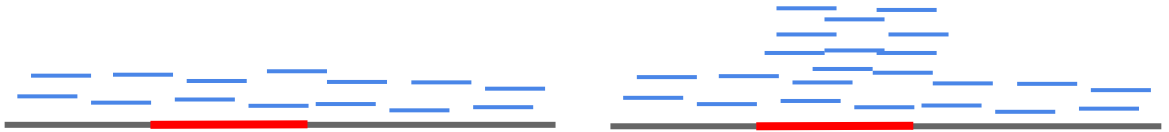
```
cumul = max = 0; start = 1;
for (i = 1; i ≤ N; i++) {
    cumul += s[i];
    if (cumul ≥ max)
        {max = cumul; end = i;}
    if (cumul ≤ 0 or cumul ≤ max + D or i == N) {
        if (max ≥ S)
            {print start, end, max; }
        max = cumul = 0; start = end = i + 1; /* NO BACKTRACKING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

Applying D-segment reveals CNVs



Both copies mapped to same reference allele

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



Same as ref.

Duplication

Quick example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	1	2	3	0	2	0	0
Score	-0.75	0.50	1.0	1.25	-0.75	1.0	-0.75	-0.75

Sample scoring scheme

0: -0.75

1: 0.50

2: 1.0

>=3: 1.25

D: -3

S: 3

max = 0

start = 0

end = 0

cumul = 0

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! */
    }
}
```

Quick example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	1	2	3	0	2	0	0
Score	-0.75	0.50	1.0	1.25	-0.75	1.0	-0.75	-0.75

Sample scoring scheme

0: -0.75

1: 0.50

2: 1.0

>=3: 1.25

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 ≤ 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! */
    }
}
```

Quick example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	1	2	3	0	2	0	0
Score	-0.75	0.50	1.0	1.25	-0.75	1.0	-0.75	-0.75

Sample scoring scheme

0: -0.75

1: 0.50

2: 1.0

>=3: 1.25

D: -3

S: 3

max = **0.5**

start = **2**

end = **2**

cumul = **0.5**

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! */
    }
}
```

Quick example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	1	2	3	0	2	0	0
Score	-0.75	0.50	1.0	1.25	-0.75	1.0	-0.75	-0.75

Sample scoring scheme

0: -0.75

1: 0.50

2: 1.0

>=3: 1.25

D: -3

S: 3

max = **1.5**

start = 2

end = **3**

cumul = **1.5**

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! */
    }
}
```

Quick example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	1	2	3	0	2	0	0
Score	-0.75	0.50	1.0	1.25	-0.75	1.0	-0.75	-0.75

Sample scoring scheme

0: -0.75

1: 0.50

2: 1.0

>=3: 1.25

D: -3

S: 3

max = **2.75**

start = 2

end = **4**

cumul = **2.75**

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! */
    }
}
```

Quick example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	1	2	3	0	2	0	0
Score	-0.75	0.50	1.0	1.25	-0.75	1.0	-0.75	-0.75

Sample scoring scheme

0: -0.75

1: 0.50

2: 1.0

>=3: 1.25

D: -3

S: 3

max = 2.75

start = 2

end = **5**

cumul = **2.0**

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! */
    }
}
```

Quick example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	1	2	3	0	2	0	0
Score	-0.75	0.50	1.0	1.25	-0.75	1.0	-0.75	-0.75

Sample scoring scheme

0: -0.75

1: 0.50

2: 1.0

>=3: 1.25

D: -3

S: 3

max = **3.0**

start = 2

end = **6**

cumul = **3.0**

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! */
    }
}
```

Quick example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	1	2	3	0	2	0	0
Score	-0.75	0.50	1.0	1.25	-0.75	1.0	-0.75	-0.75

Sample scoring scheme

0: -0.75

1: 0.50

2: 1.0

>=3: 1.25

D: -3

S: 3

max = 3.0

start = 2

end = 6

cumul = **2.25**

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! */
    }
}
```

Quick example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	1	2	3	0	2	0	0
Score	-0.75	0.50	1.0	1.25	-0.75	1.0	-0.75	-0.75

Sample scoring scheme

0: -0.75

1: 0.50

2: 1.0

>=3: 1.25

D: -3

S: 3

max = 3.0

start = 2

end = 6

cumul = **1.5**

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! */
    }
}
```

Quick example

Position	1	2	3	4	5	6	7	8
Read Start Counts	0	1	2	3	0	2	0	0
Score	-0.75	0.50	1.0	1.25	-0.75	1.0	-0.75	-0.75

Sample scoring scheme

0: -0.75 **D:** -3

1: 0.50 **S:** 3

2: 1.0

>=3: 1.25

max = 3.0

start = 2

end = 6

cumul = 1.5

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! */
    }
}
```

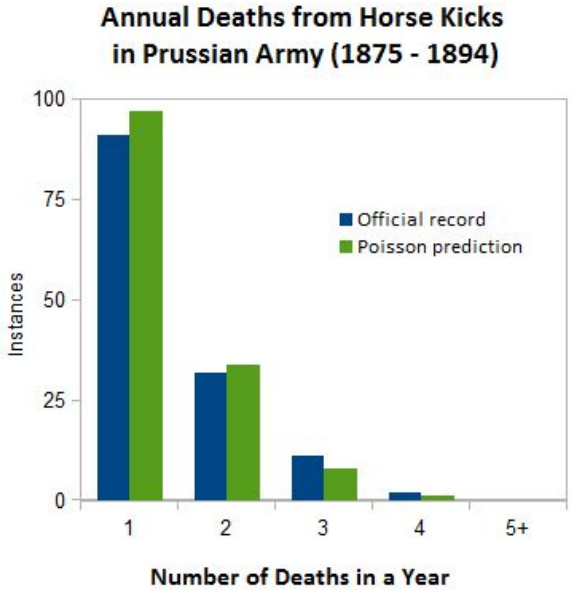
Poisson distributions

What is the Poisson distribution?

Used for discrete, countable data over given intervals

Classic example: yearly deaths by horsekick in the Prussian army (from von Bortkiewicz, 1898)

- Many years had zero deaths
- Some years had one
- Fewer had two, etc.

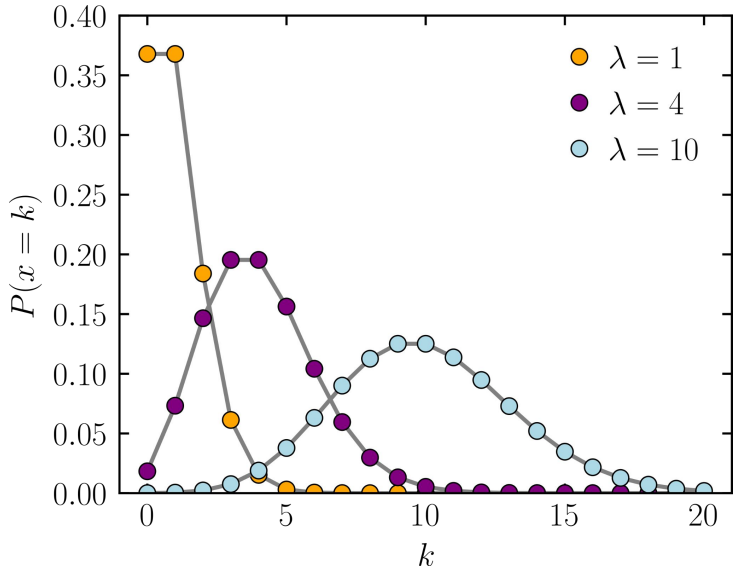


https://mindyourdecisions.com/blog/wp-content/uploads/2013/06/happy-birthday-poisson-chart.png

What is the Poisson distribution?

Because we are dealing with count data

- Support only from natural numbers; thus the distribution often shifts left toward zero
- Shape depends on the parameter (λ), which in turn relies on the likelihood of each event



https://en.wikipedia.org/wiki/Poisson_distribution#/media/File:Poisson_pmf.svg

Defining the Poisson

A random variable X has a Poisson distribution if its probability function is defined by:

$$f(k;\lambda) = \Pr(X{=}k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

That is; the probability that the measured count X has value k is given by $\frac{\lambda^k e^{-\lambda}}{k!}$

- k represents numbers of occurences (0, 1, 2, …)
- λ is both the mean and variance of f(k; λ)

https://en.wikipedia.org/wiki/Poisson_distribution

Working with the Poisson

Has nice properties:

- λ is both the mean and variance of $f(k; \lambda)$
- Serves as an approximation of the binomial distribution (if the number of events is sufficiently high and the probability of each event is sufficiently low)
- Has a known conjugate prior (Gamma), giving a closed-form posterior for use in Bayesian inference (basically; it’s easier to update models with new data)

Office hours

Reminder:

Homework 6 is due Sunday, February 23rd at 11:59pm!