Short read alignment

Andrey Prjibelski
Center for Algorithmic Biotechnology, SPbU

Alignment

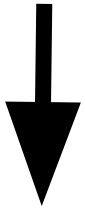
AACGCTAACGGTAA

AACCGCGAACTAA

Alignment

AACGCTAACGGTAA

AACCGCGAACTAA



AAC - GCTAACGGTAA

AACCGCGAAC - - TAA

Short read alignment

Find the read in the genome

Short read alignment

- Challenges?

Short read alignment

- Challenges
  - Small length
  - Gigabytes of data
  - Different sequencing errors
  - SNPs
  - Genomic repeats
- Tools
  - Bowtie2, BWA MEM, minimap2 (Genomic)
  - HiSat2, STAR (RNA-Seq)
  - and many more

Bowtie

Mapping Illumina reads

Burrows-Wheeler transform

a c a a c g

Burrows-Wheeler transform

a c a a c g $

Burrows-Wheeler transform

a c a a c g $

$ a c a a c g

Burrows-Wheeler transform

a c a a c g $

$ a c a a c g

g $ a c a a c

Burrows-Wheeler transform

a c a a c g $

$ a c a a c g

g $ a c a a c

c g $ a c a a

a c g $ a c a

a a c g $ a c

c a a c g $ a

Burrows-Wheeler transform

$ a c a a c g

a a c g $ a c

a c a a c g $

a c g $ a c a

c a a c g $ a

c g $ a c a a

g $ a c a a c

Burrows-Wheeler transform

$ a c a a c g

a a c g $ a c

a c a a c g $

a c g $ a c a

c a a c g $ a

c g $ a c a a

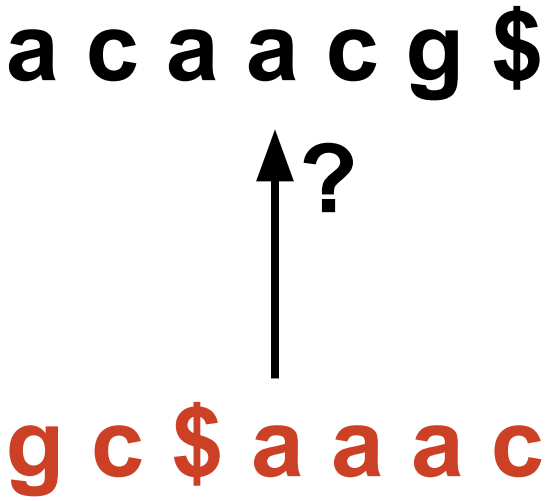
g $ a c a a c

Burrows-Wheeler transform

a c a a c g $

g c $ a a a c

Burrows-Wheeler transform



Burrows-Wheeler transform

$ a c a a c g

a a c g $ a c

a c a a c g $

a c g $ a c a

c a a c g $ a

c g $ a c a a

g $ a c a a c

Burrows-Wheeler transform

$ a c a a c g

a a c g $ a c

a c a a c g $

a c g $ a c a

c a a c g $ a

c g $ a c a a

g $ a c a a c

Burrows-Wheeler transform

g $ a c a a c

c a a c g $ a

$ a c a a c g

a a c g $ a c

a c a a c g $

a c g $ a c a

c g $ a c a a

Burrows-Wheeler transform

$ a c a a c g

a a c g $ a c

a c a a c g $

a c g $ a c a

c a a c g $ a

c g $ a c a a

g $ a c a a c

Burrows-Wheeler transform

g	$ a c a a c g
c	a a c g $ a c
$	a c a a c g $
a	a c g $ a c a
a	c a a c g $ a
a	c g $ a c a a
c	g $ a c a a c

Burrows-Wheeler transform

$	$ a c a a c g
a	a a c g $ a c
a	a c a a c g $
a	a c g $ a c a
c	c a a c g $ a
c	c g $ a c a a
g	g $ a c a a c

Burrows-Wheeler transform

$	$ a c a a c g
a	a a c g $ a c
a	a c a a c g $
a	a c g $ a c a
c	c a a c g $ a
c	c g $ a c a a
g	g $ a c a a c

Burrows-Wheeler transform

g $	$ a c a a c g
c a	a a c g $ a c
$ a	a c a a c g $
a a	a c g $ a c a
a c	c a a c g $ a
a c	c g $ a c a a
c g	g $ a c a a c

Burrows-Wheeler transform

$ a	$ a c a a c g
a a	a a c g $ a c
a c	a c a a c g $
a c	a c g $ a c a
c a	c a a c g $ a
c g	c g $ a c a a
g $	g $ a c a a c

Burrows-Wheeler transform

$ a	$ a c a a c g
a a	a a c g $ a c
a c	a c a a c g $
a c	a c g $ a c a
c a	c a a c g $ a
c g	c g $ a c a a
g $	g $ a c a a c

Burrows-Wheeler transform

g $ a	$ a c a a c g
c a a	a a c g $ a c
$ a c	a c a a c g $
a a c	a c g $ a c a
a c a	c a a c g $ a
a c g	c g $ a c a a
c g $	g $ a c a a c

Burrows-Wheeler transform

$ a c	$ a c a a c g
a a c	a a c g $ a c
a c a	a c a a c g $
a c g	a c g $ a c a
c a a	c a a c g $ a
c g $	c g $ a c a a
g $ a	g $ a c a a c

Burrows-Wheeler transform

$	a	c	$	a	c	a	a	c	g
a	a	c	a	a	c	g	$	a	c
a	c	a	a	c	a	a	c	g	$
a	c	g	a	c	g	$	a	c	a
c	a	a	c	a	a	c	g	$	a
c	g	$	c	g	$	a	c	a	a
g	$	a	g	$	a	c	a	a	c

Burrows-Wheeler transform

g $ a c	$ a c a a c g
c a a c	a a c g $ a c
$ a c a	a c a a c g $
a a c g	a c g $ a c a
a c a a	c a a c g $ a
a c g $	c g $ a c a a
c g $ a	g $ a c a a c

Burrows-Wheeler transform

$ a c a	$ a c a	a c g
a a c g	a a c g	$ a c
a c a a	a c a a	c g $
a c g $	a c g $	a c a
c a a c	c a a c	g $ a
c g $ a	c g $ a	c a a
g $ a c	g $ a c	a a c

Burrows-Wheeler transform

g $ a c a	$ a c a a c g
c a a c g	a a c g $ a c
$ a c a a	a c a a c g $
a a c g $	a c g $ a c a
a c a a c	c a a c g $ a
a c g $ a	c g $ a c a a
c g $ a c	g $ a c a a c

Burrows-Wheeler transform

$ a c a a	$ a c a a c	g
a a c g $	a a c g $ a	c
a c a a c	a c a a c g	$
a c g $ a	a c g $ a c	a
c a a c g	c a a c g $	a
c g $ a c	c g $ a c a	a
g $ a c a	g $ a c a a	c

Burrows-Wheeler transform

g $ a c a a	$ a c a a c g
c a a c g $	a a c g $ a c
$ a c a a c	a c a a c g $
a a c g $ a	a c g $ a c a
a c a a c g	c a a c g $ a
a c g $ a c	c g $ a c a a
c g $ a c a	g $ a c a a c

Burrows-Wheeler transform

$ a c a a c	$ a c a a c g
a a c g $ a	a a c g $ a c
a c a a c g	a c a a c g $
a c g $ a c	a c g $ a c a
c a a c g $	c a a c g $ a
c g $ a c a	c g $ a c a a
g $ a c a a	g $ a c a a c

Burrows-Wheeler transform

g $ a c a a c	$ a c a a c g
c a a c g $ a	a a c g $ a c
$ a c a a c g	a c a a c g $
a a c g $ a c	a c g $ a c a
a c a a c g $	c a a c g $ a
a c g $ a c a	c g $ a c a a
c g $ a c a a	g $ a c a a c

Burrows-Wheeler transform

$ a c a a c g	$ a c a a c g
a a c g $ a c	a a c g $ a c
a c a a c g $	a c a a c g $
a c g $ a c a	a c g $ a c a
c a a c g $ a	c a a c g $ a
c g $ a c a a	c g $ a c a a
g $ a c a a c	g $ a c a a c

First-last property

$_1$ a c a a c g
a$_1$ a c g $ a c
a$_2$ c a a c g $
a$_3$ c g $ a c a
c$_1$ a a c g $ a
c$_2$ g $ a c a a
g$_1$ $ a c a a c

First-last property

$$\$_1\ a\ c\ a\ a\ c\ g_1$$
$$a_1\ a\ c\ g\ \$\ a\ c$$
$$a_2\ c\ a\ a\ c\ g\ \$_1$$
$$a_3\ c\ g\ \$\ a\ c\ a$$
$$c_1\ a\ a\ c\ g\ \$\ a$$
$$c_2\ g\ \$\ a\ c\ a\ a$$
$$g_1\ \$\ a\ c\ a\ a\ c$$

First-last property

$_1$ a c a a c g$_1$

a$_1$ a c g $ a c

a$_2$ c a a c g $$_1$

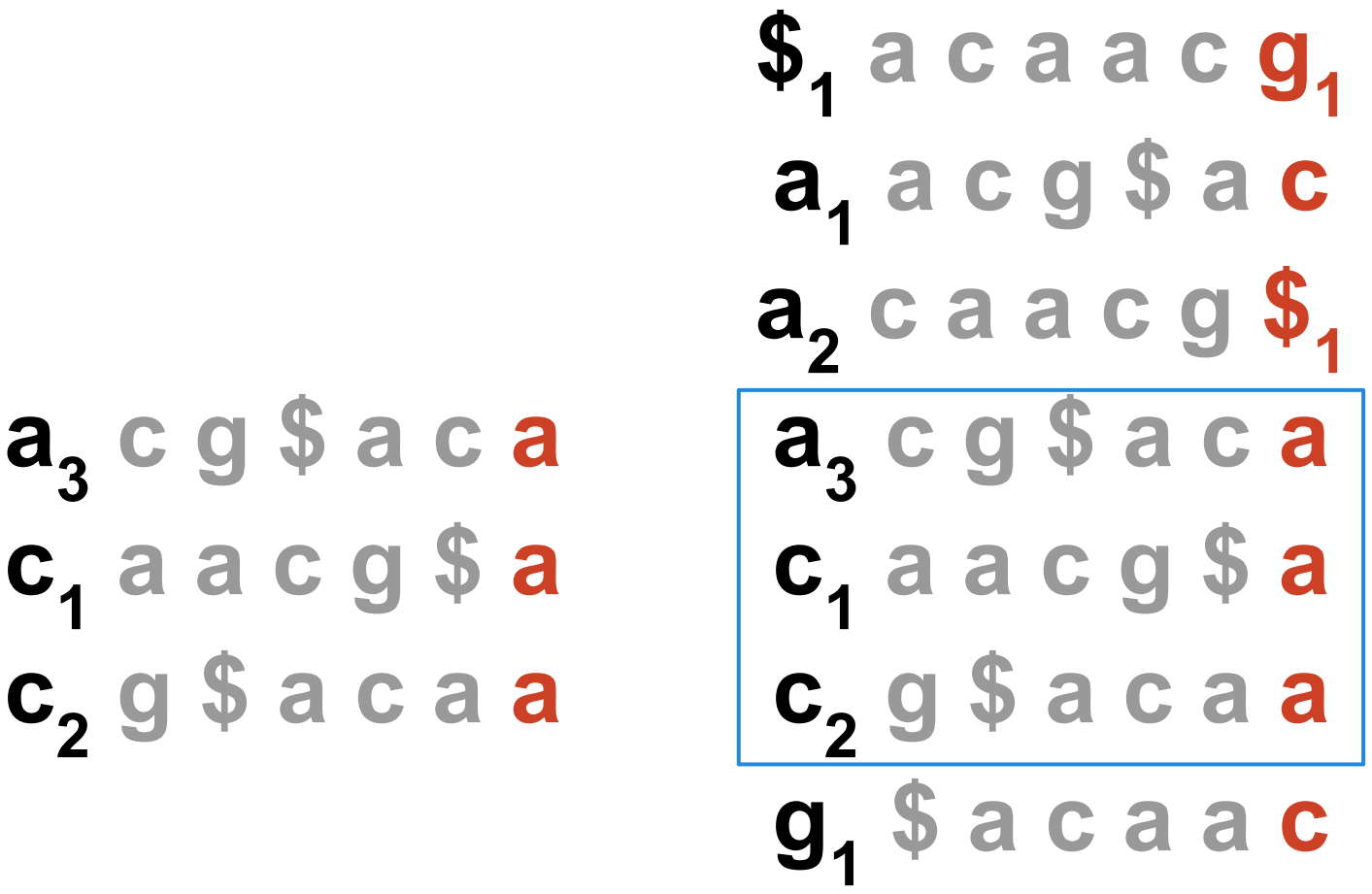
a$_3$ c g $ a c a

c$_1$ a a c g $ a

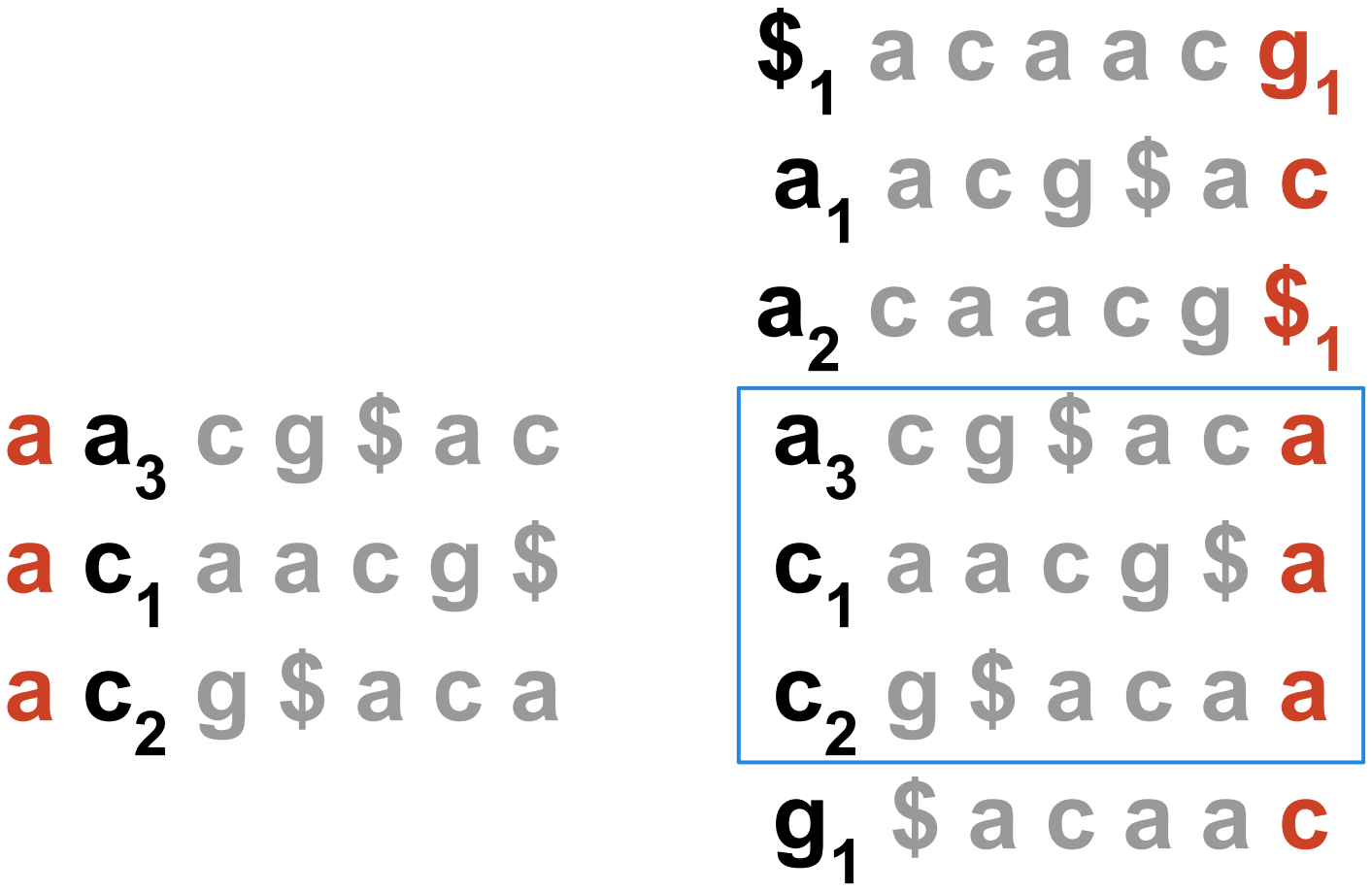
c$_2$ g $ a c a a

g$_1$ $ a c a a c

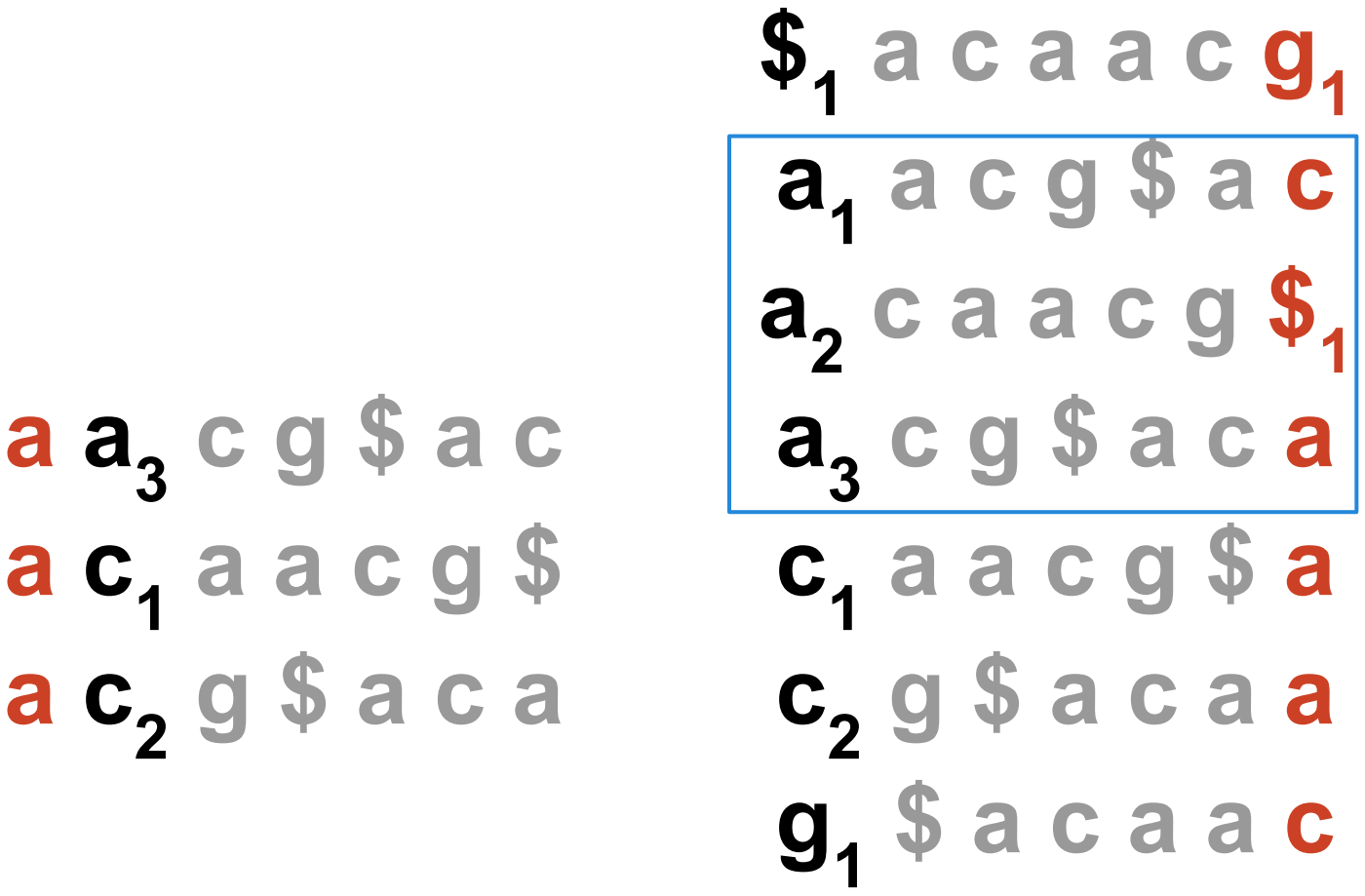
First-last property



First-last property



First-last property



First-last property

$a_1\ a_3\ c\ g\ \$\ a\ c$

$a_2\ c_1\ a\ a\ c\ g\ \$$

$a_3\ c_2\ g\ \$\ a\ c\ a$

$\$_1\ a\ c\ a\ a\ c\ g_1$

$a_1\ a\ c\ g\ \$\ a\ c$

$a_2\ c\ a\ a\ c\ g\ \$_1$

$a_3\ c\ g\ \$\ a\ c\ a$

$c_1\ a\ a\ c\ g\ \$\ a$

$c_2\ g\ \$\ a\ c\ a\ a$

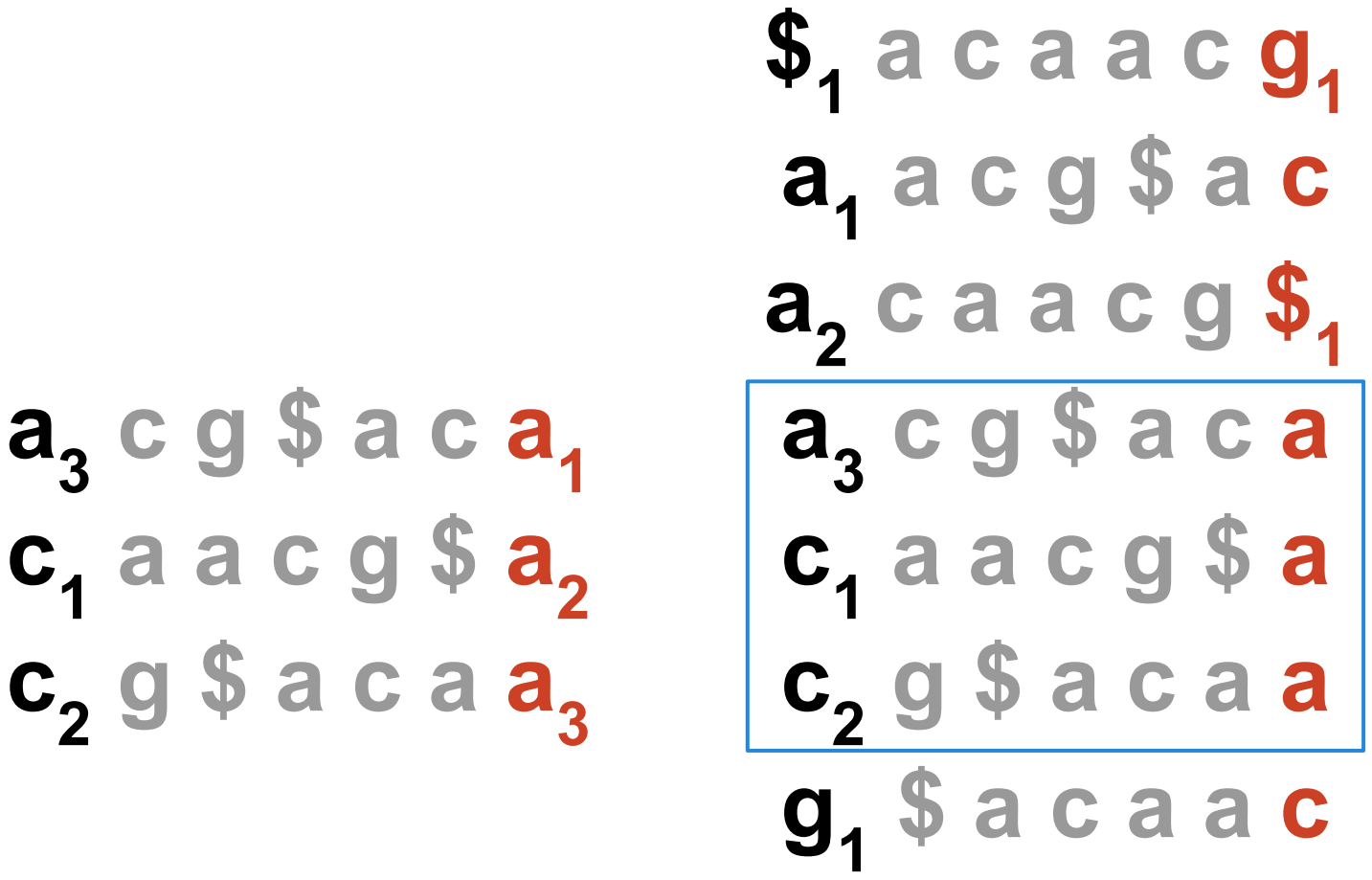
$g_1\ \$\ a\ c\ a\ a\ c$

First-last property

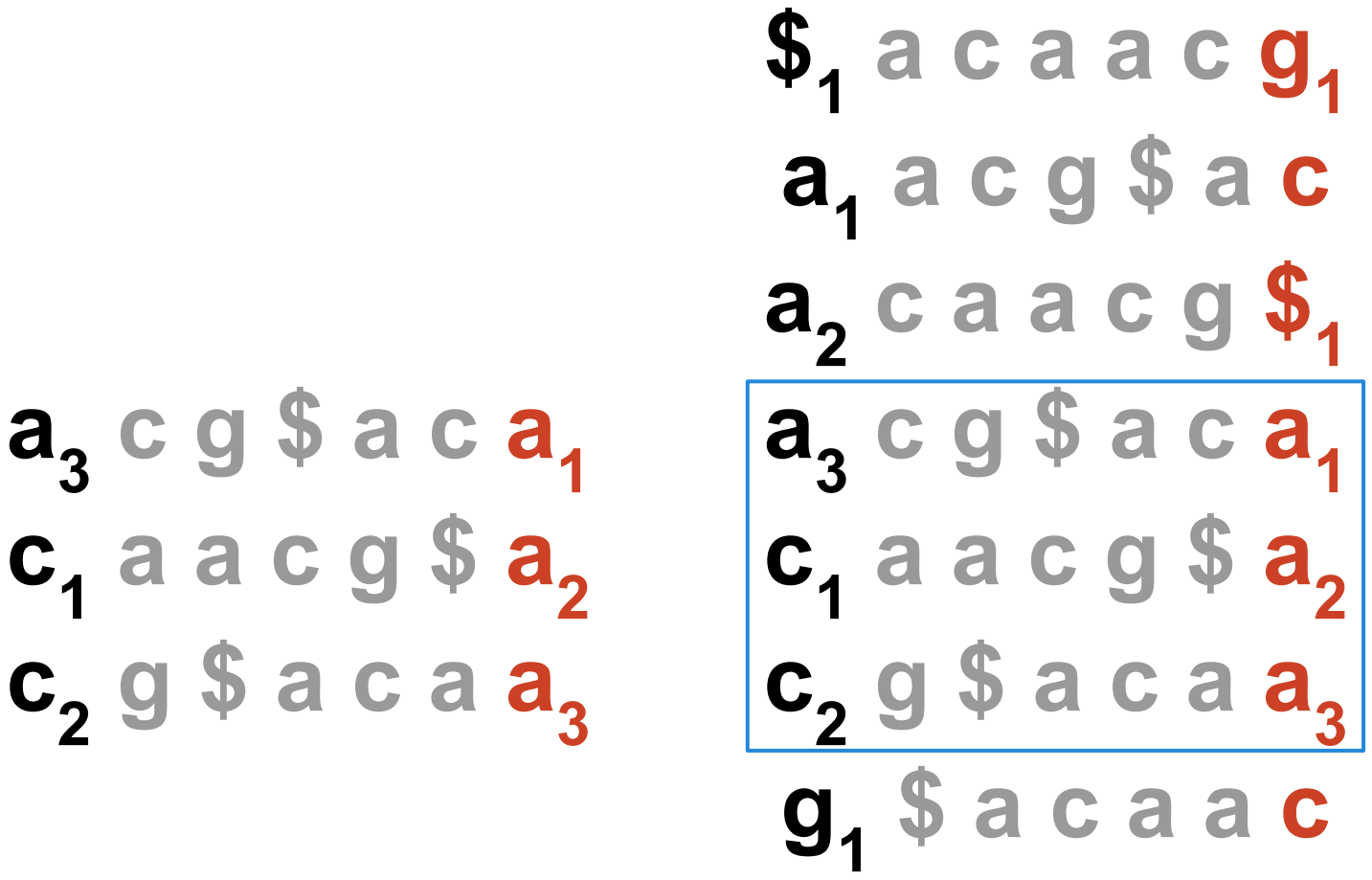
$$\$_1\ a\ c\ a\ a\ c\ g_1$$
$$a_1\ a\ c\ g\ \$\ a\ c$$
$$a_2\ c\ a\ a\ c\ g\ \$_1$$
$$a_3\ c\ g\ \$\ a\ c\ a_1$$
$$c_1\ a\ a\ c\ g\ \$\ a_2$$
$$c_2\ g\ \$\ a\ c\ a\ a_3$$

$$a_3\ c\ g\ \$\ a\ c\ a$$
$$c_1\ a\ a\ c\ g\ \$\ a$$
$$c_2\ g\ \$\ a\ c\ a\ a$$
$$g_1\ \$\ a\ c\ a\ a\ c$$

First-last property



First-last property



First-last property

$_1$ a c a a c g$_1$

a$_1$ a c g $ a c$_1$

a$_2$ c a a c g $_1$

a$_3$ c g $ a c a$_1$

c$_1$ a a c g $ a$_2$

c$_2$ g $ a c a a$_3$

g$_1$ $ a c a a c$_2$

Suffix array

6.	$
2.	a a c g $
0.	a c a a c g $
3.	a c g $
1.	c a a c g $
4.	c g $
5.	g $

Suffix array and BWT

6. $	$ a c a a c g
2. a a c g $	a a c g $ a c
0. a c a a c g $	a c a a c g $
3. a c g $	a c g $ a c a
1. c a a c g $	c a a c g $ a
4. c g $	c g $ a c a a
5. g $	g $ a c a a c

Suffix array and BWT

$B[i] = \$$ if $S[i] = 0$

$B[i] = X[S[i] - 1]$ otherwise

Suffix array interval

6.	$
2.	a a c g $
0.	a c a a c g $
3.	a c g $
1.	c a a c g $
4.	c g $
5.	g $

R(“”) = (0, 6)

Suffix array interval

6. $

2. a a c g $

0. a c a a c g $

3. a c g $

1. c a a c g $

4. c g $

5. g $

R(“a”) = (1, 3)

Suffix array interval

6. $

2. a a c g $

0. a c a a c g $

3. a c g $

1. c a a c g $

4. c g $

5. g $

R("a c") = (2, 3)

Suffix array interval

$$R_L(W) = \min \{k: W \text{ is prefix of } X_{S[k]} \}$$

$$R_H(W) = \max \{k: W \text{ is prefix of } X_{S[k]} \}$$

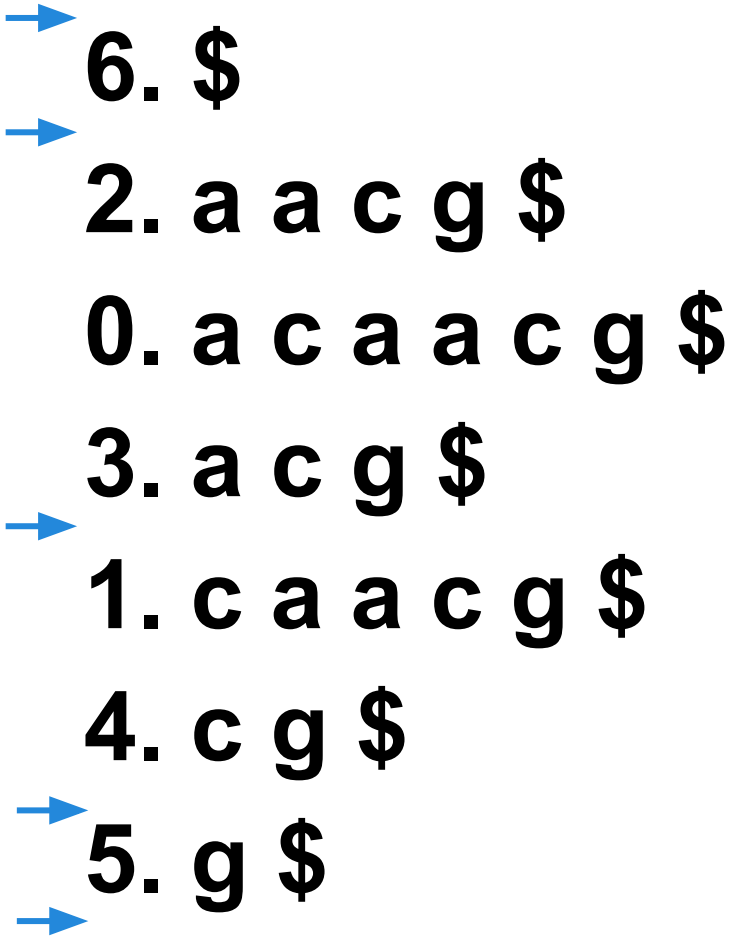
Suffix array interval

$$C(x) = |\{ 0 \le j \le n-2 : X[j] < x \}|$$

a c a a c g $

C(a) = 0, C(c) = 3, C(g) = 5, ...

Suffix array interval



Suffix array interval

$$O(x, i) = |\{ 0 \le j \le i : B[j] = x \}|$$

g c $ a a a c

$$O(a, 0) = 0,\ O(a, 1) = 0,\ O(a, 2) = 0,$$
$$O(a, 3) = 1,\ O(a, 4) = 2,\ \ldots$$

Suffix array interval

		a	c	g	t
$\$_1$	a c a a c g_1	0	0	1	0
a_1	a c g $ a c_1	0	1	1	0
a_2	c a a c g $\$_1$	0	1	1	0
a_3	c g $ a c a_1	1	1	1	0
c_1	a a c g $ a_2	2	1	1	0
c_2	g $ a c a a_3	3	1	1	0
g_1	$ a c a a c_2	3	2	1	0

Suffix array interval

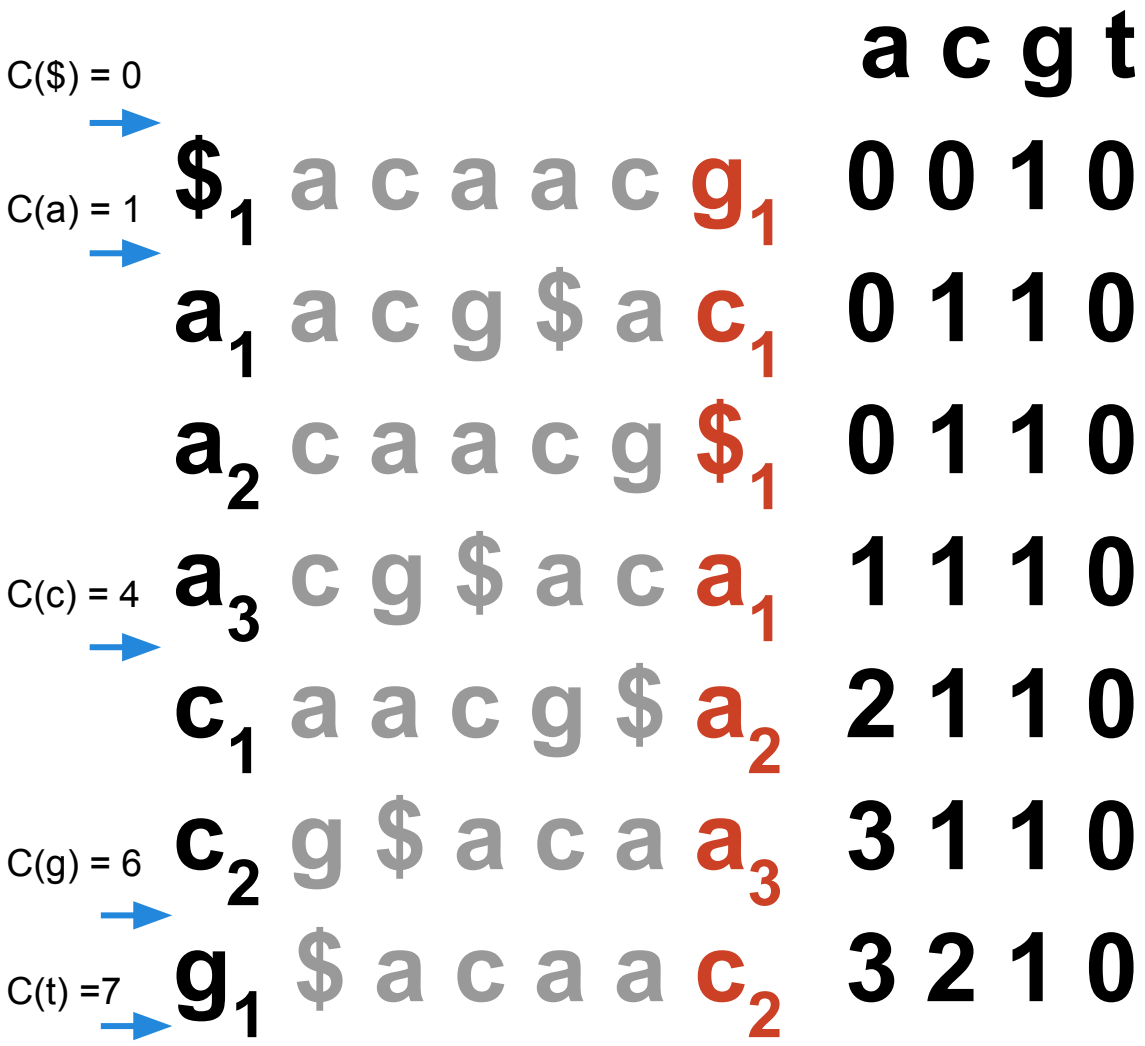
$$R_L(xW) = C(x) + O(x, R_L(W) - 1)$$

$$R_H(xW) = C(x) + O(x, R_H(W)) - 1$$

$$R_L(\text{“”}) = 0$$

$$R_H(\text{“”}) = \text{len}(X) - 1$$

Pattern search



Pattern search

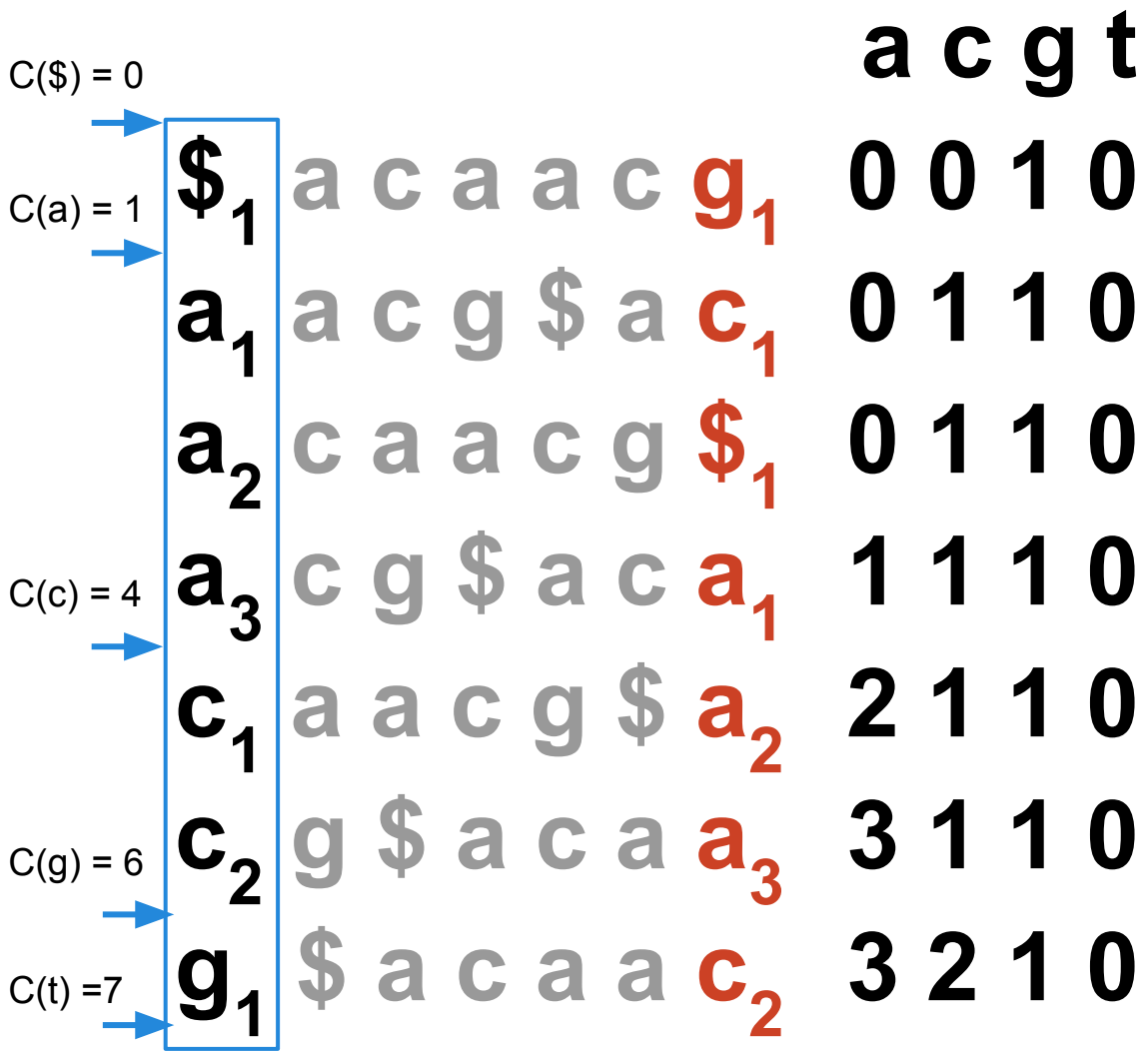
a a c

Pattern search

a a c

$R_L(\text{“”}) = 0$

$R_H(\text{“”}) = 6$

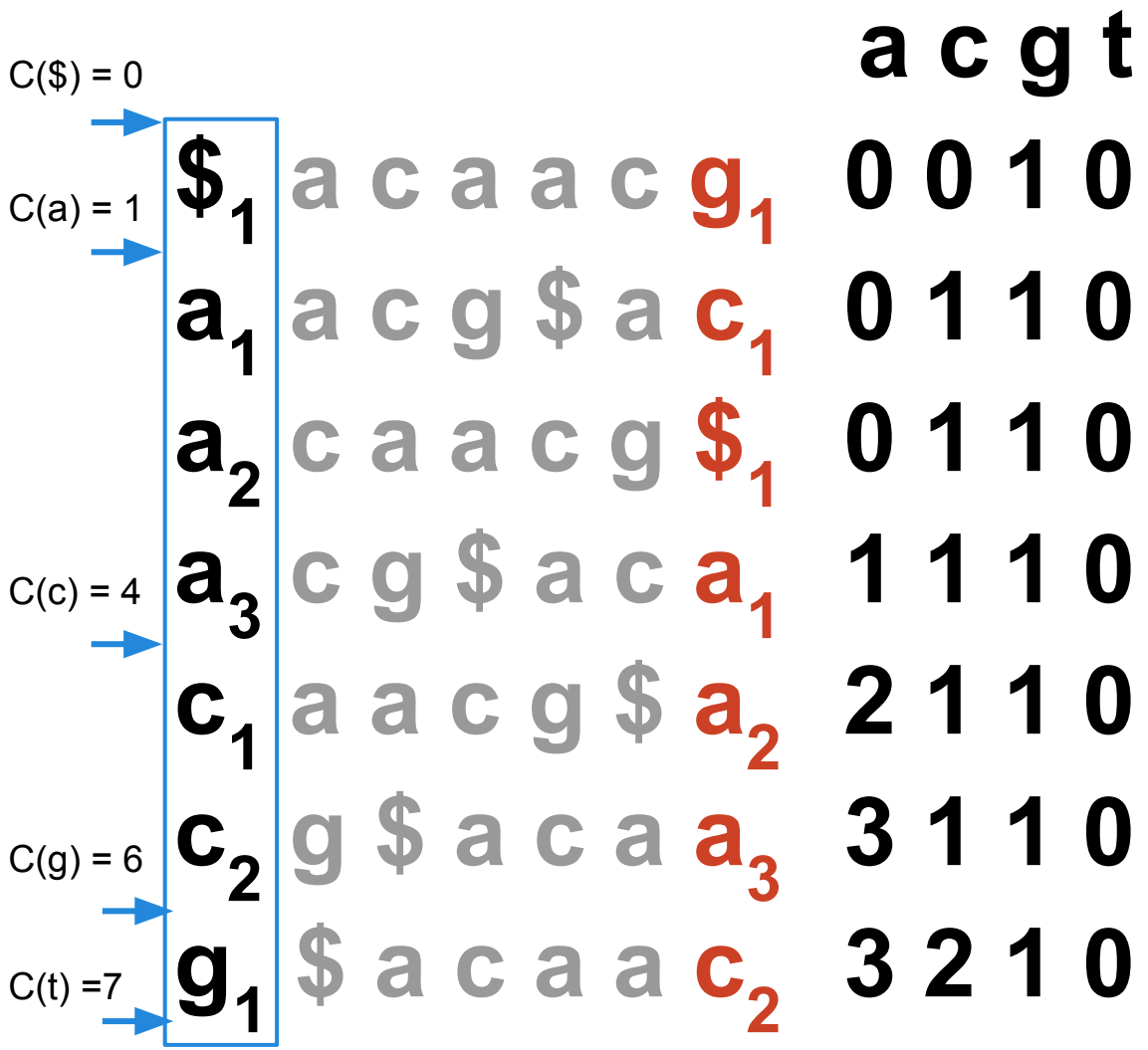


Pattern search

a a c

$R_L(xW) = C(x) + O(x, R_L(W) - 1)$

$R_H(xW) = C(x) + O(x, R_H(W)) - 1$

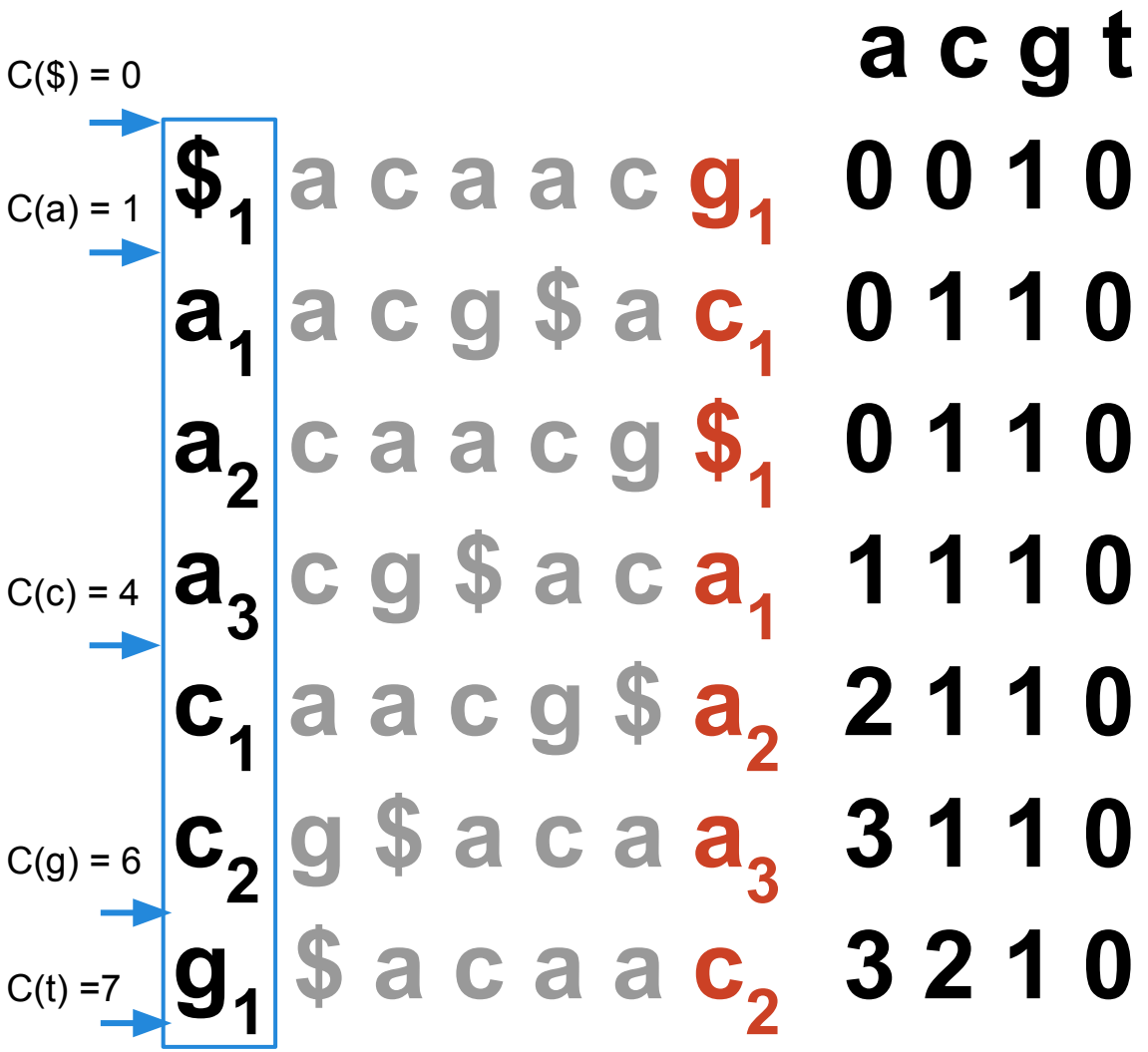


Pattern search

a a c

$R_L(c) = C(c) + O(c, R_L(\text{“”}) - 1)$

$R_H(c) = C(c) + O(c, R_H(\text{“”})) - 1$

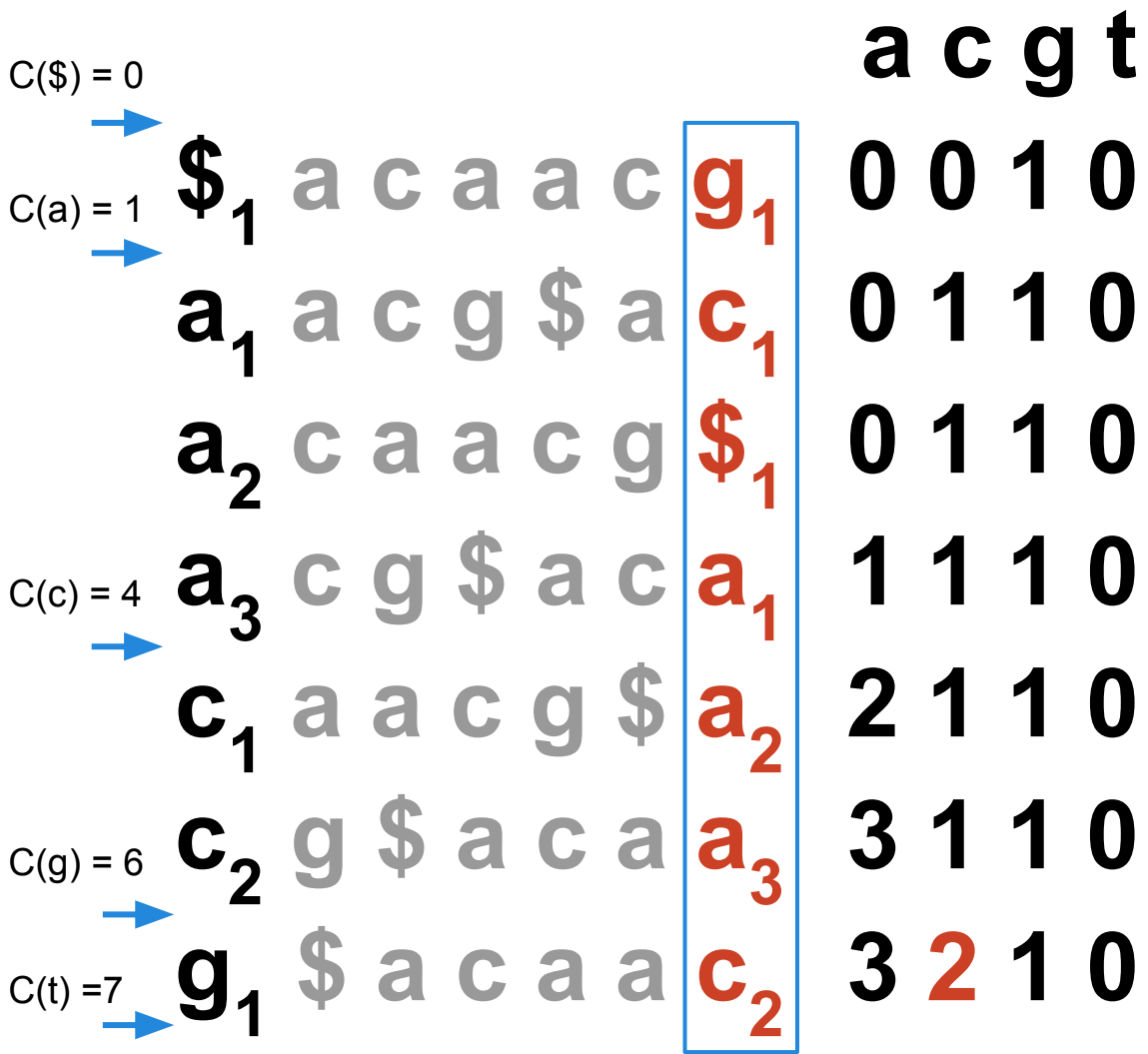


Pattern search

a a c

$R_L(c) = C(c) + \mathbf{O(c, -1)}$

$R_H(c) = C(c) + \mathbf{O(c, 6)} - 1$

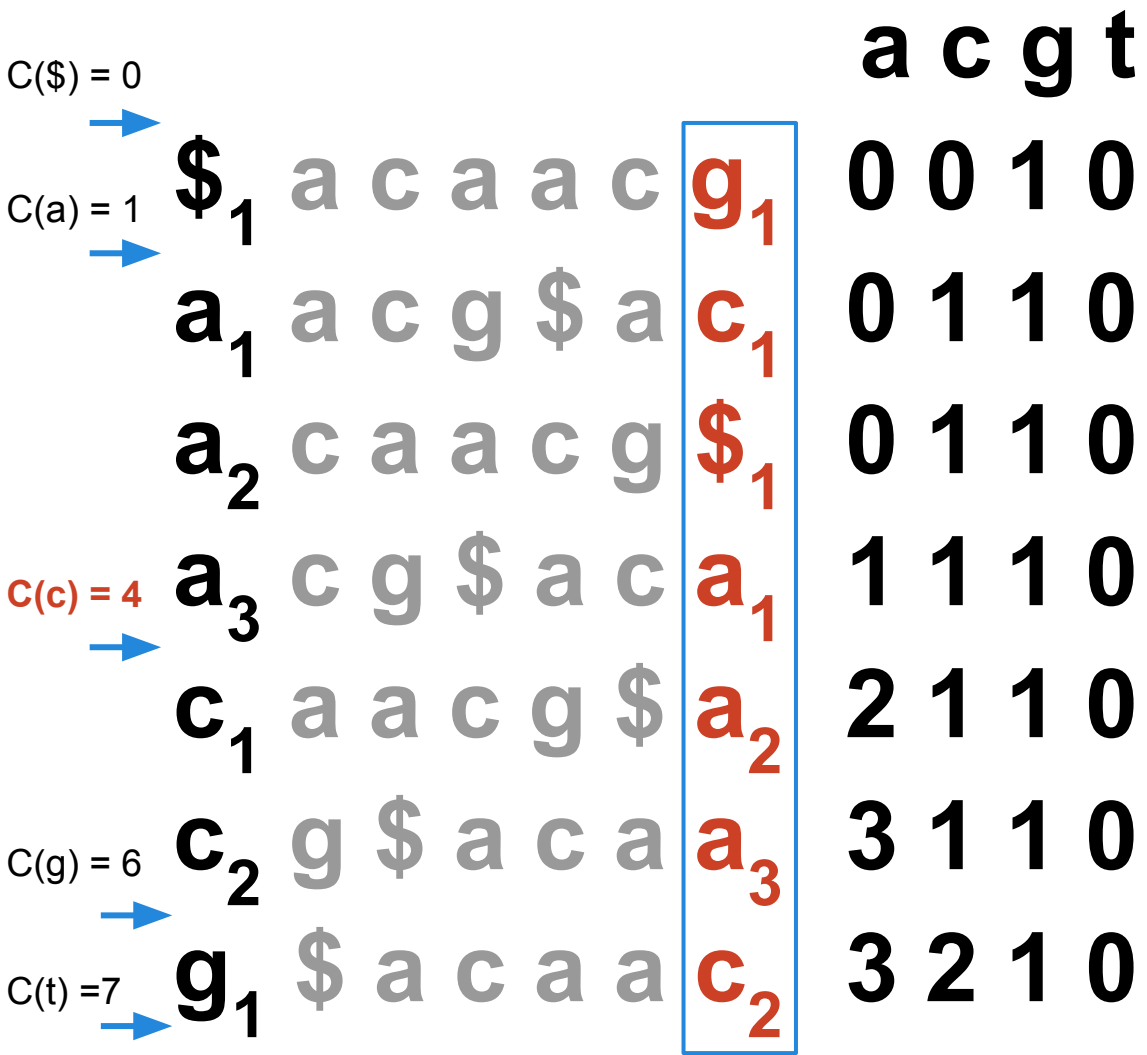


Pattern search

a a c

$R_L(c) = \mathbf{C(c)} + 0$

$R_H(c) = \mathbf{C(c)} + 2 - 1$

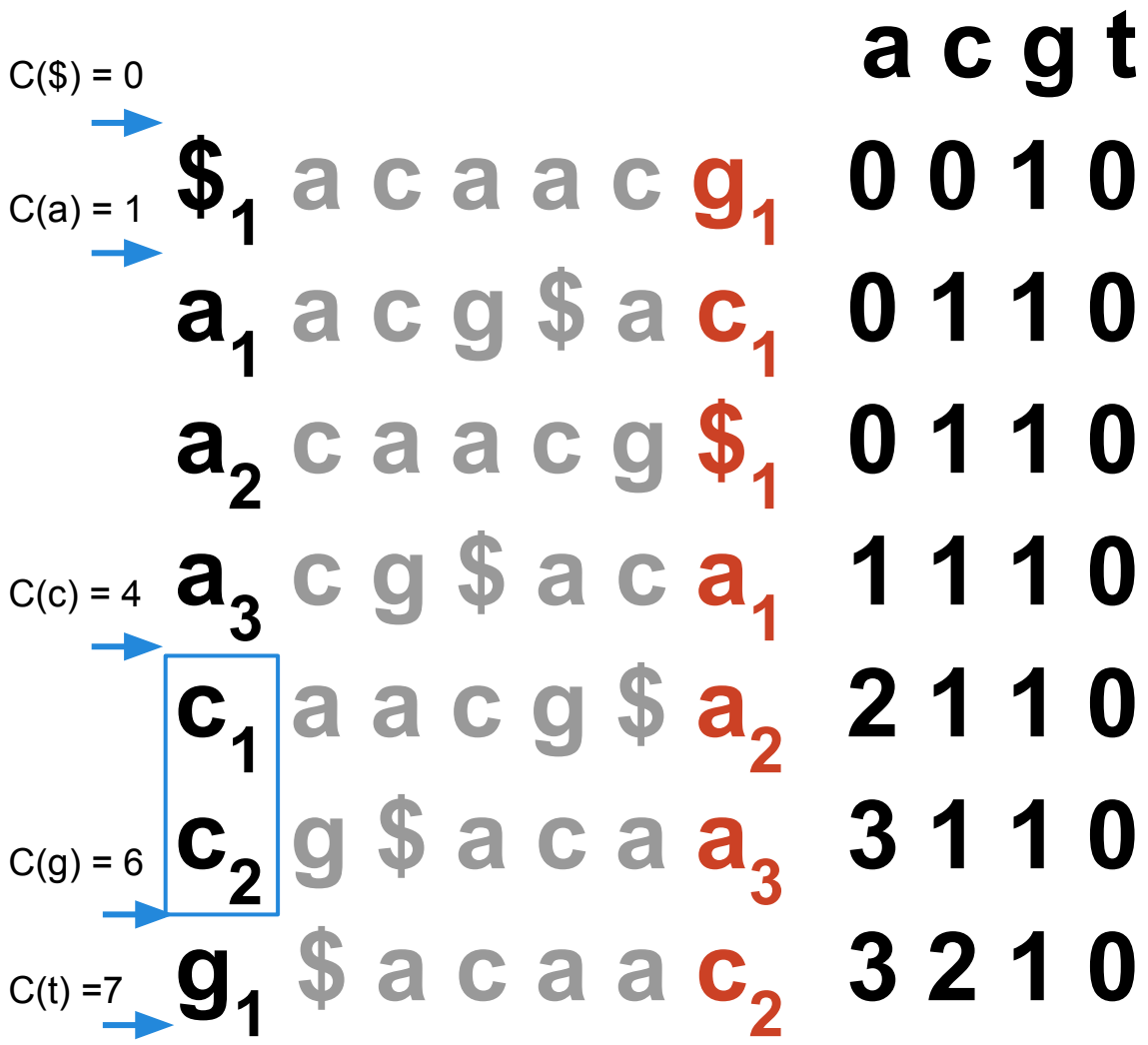


Pattern search

a a c

$R_L(c) = 4 + 0 = 4$

$R_H(c) = 4 + 2 - 1 = 5$

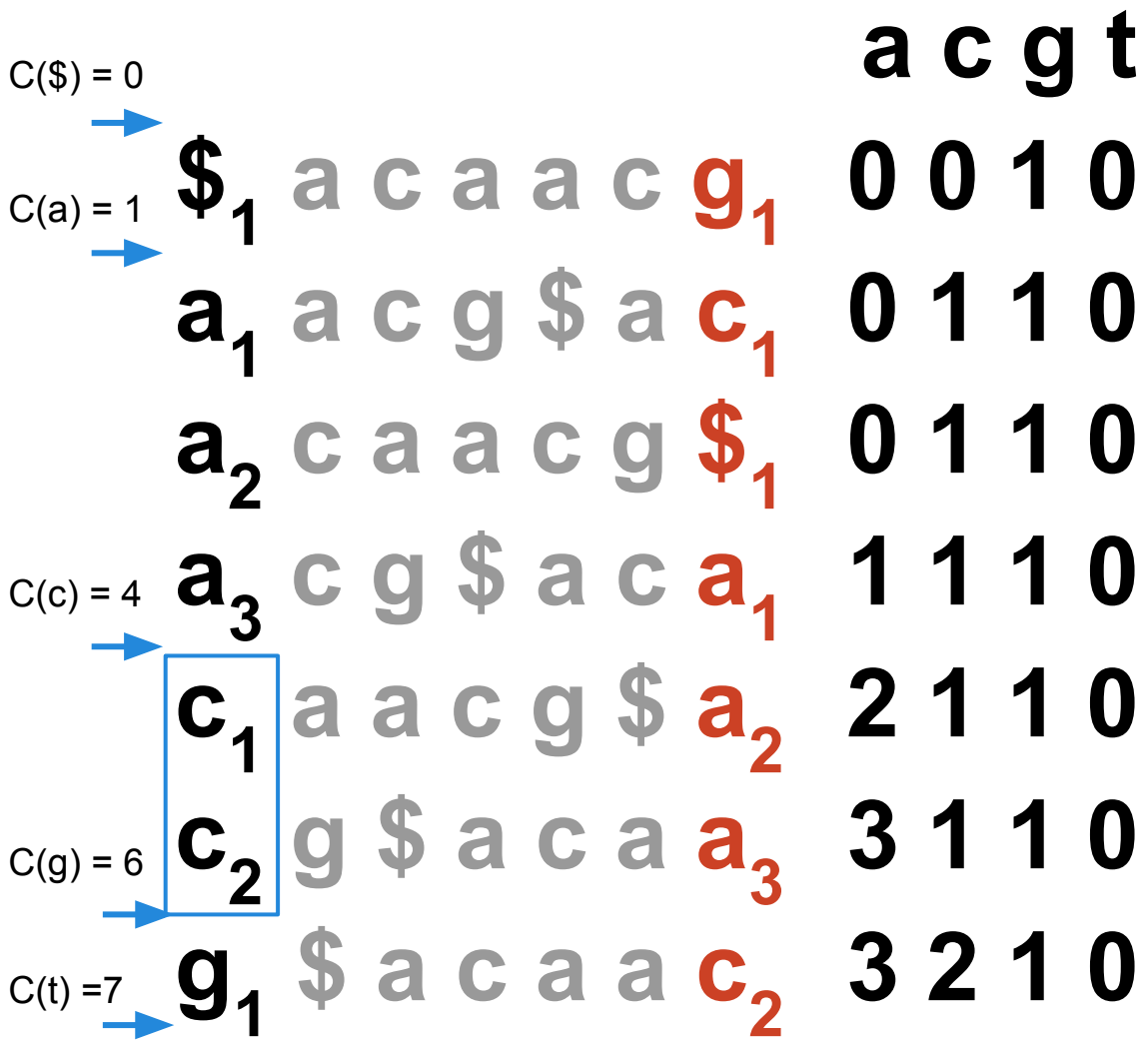


Pattern search

a a c

$R_L(ac) = C(a) + O(a, R_L(c) - 1)$

$R_H(ac) = C(a) + O(a, R_H(c)) - 1$

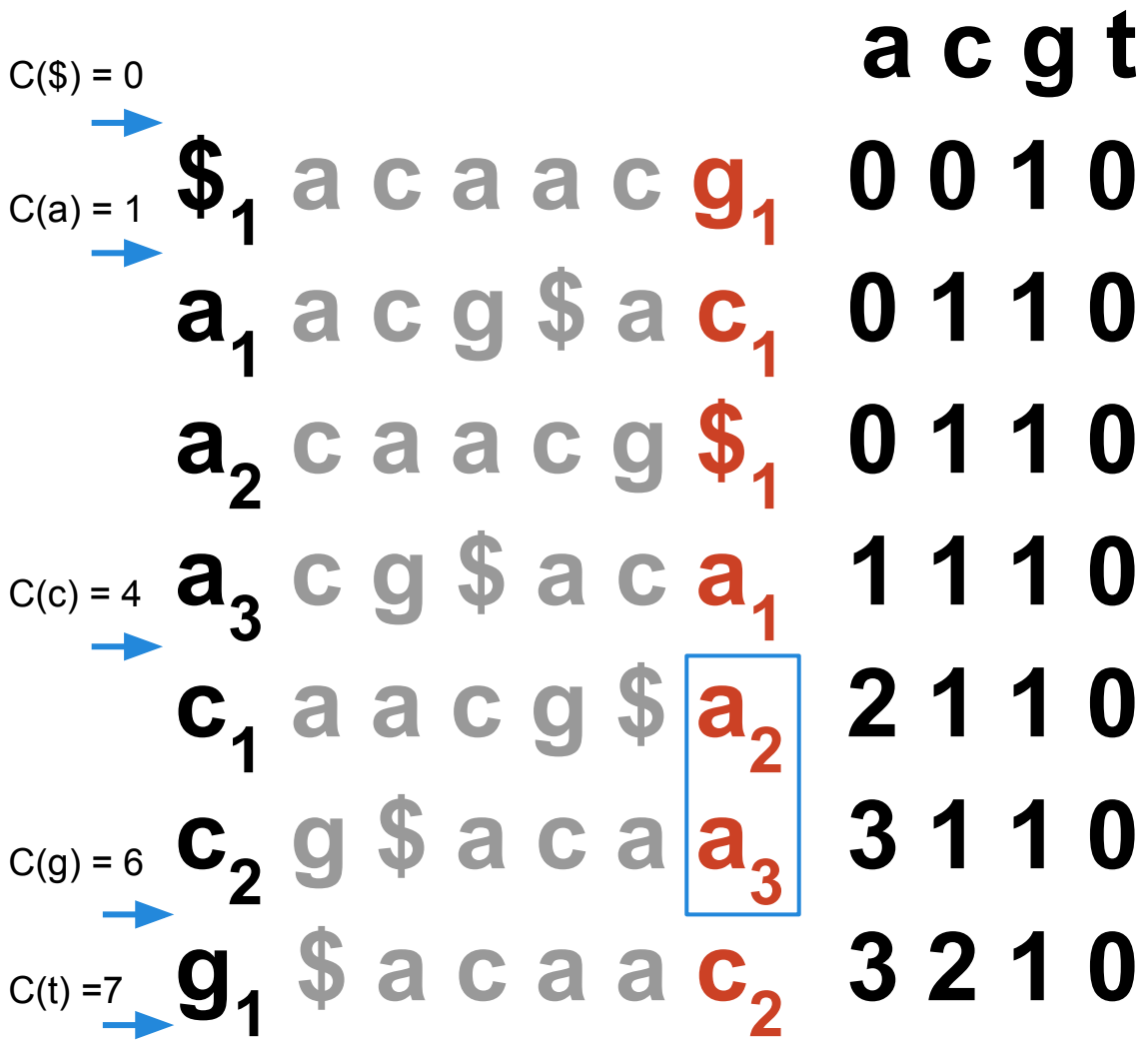


Pattern search

a a c

$R_L(ac) = C(a) + \mathbf{O(a, 3)}$

$R_H(ac) = C(a) + \mathbf{O(a, 5)} - 1$

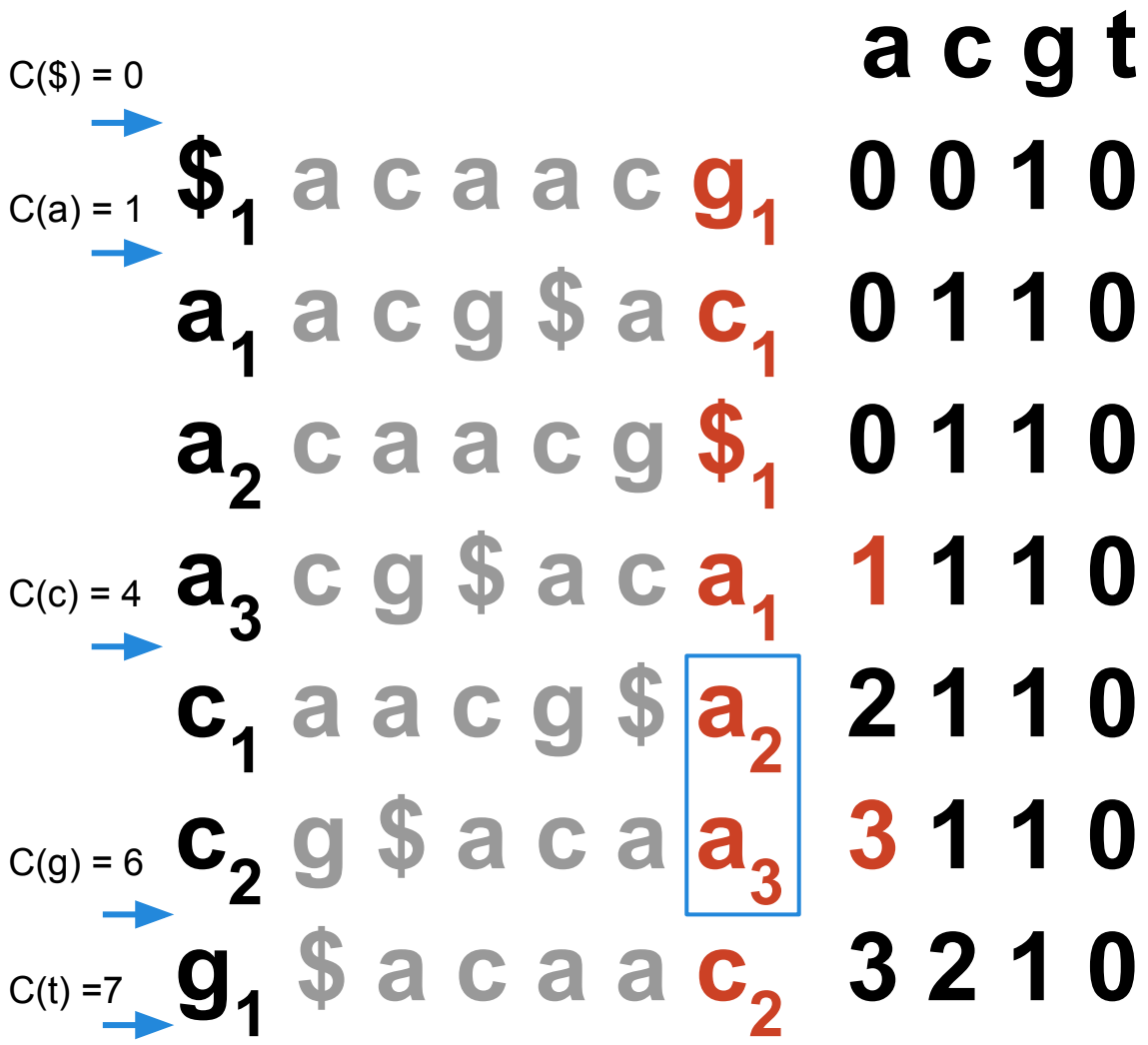


Pattern search

a a c

$R_L(ac) = C(a) + \mathbf{O(a, 3)}$

$R_H(ac) = C(a) + \mathbf{O(a, 5)} - 1$

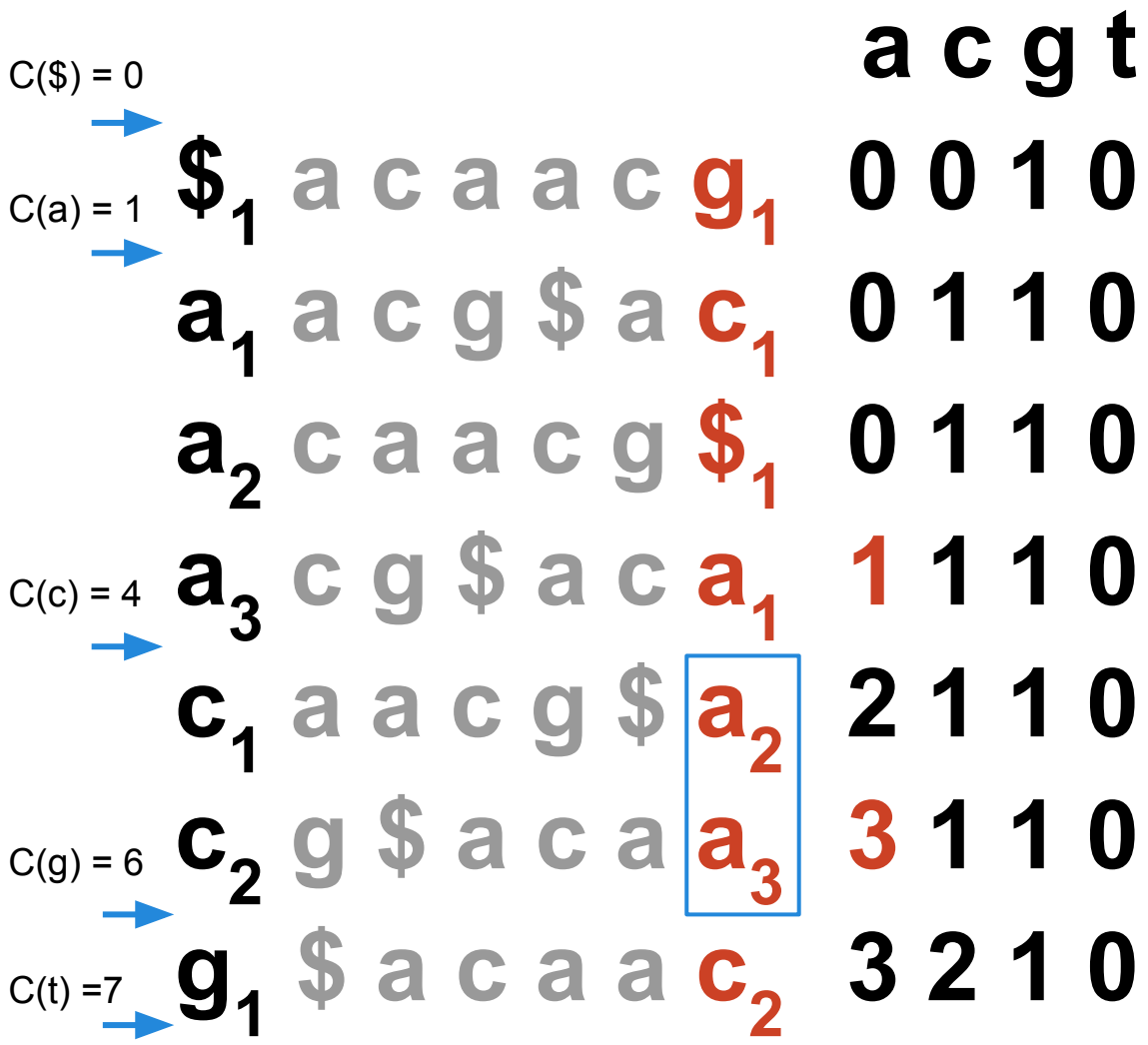


Pattern search

a a c

$R_L(ac) = C(a) + 1$

$R_H(ac) = C(a) + 3 - 1$

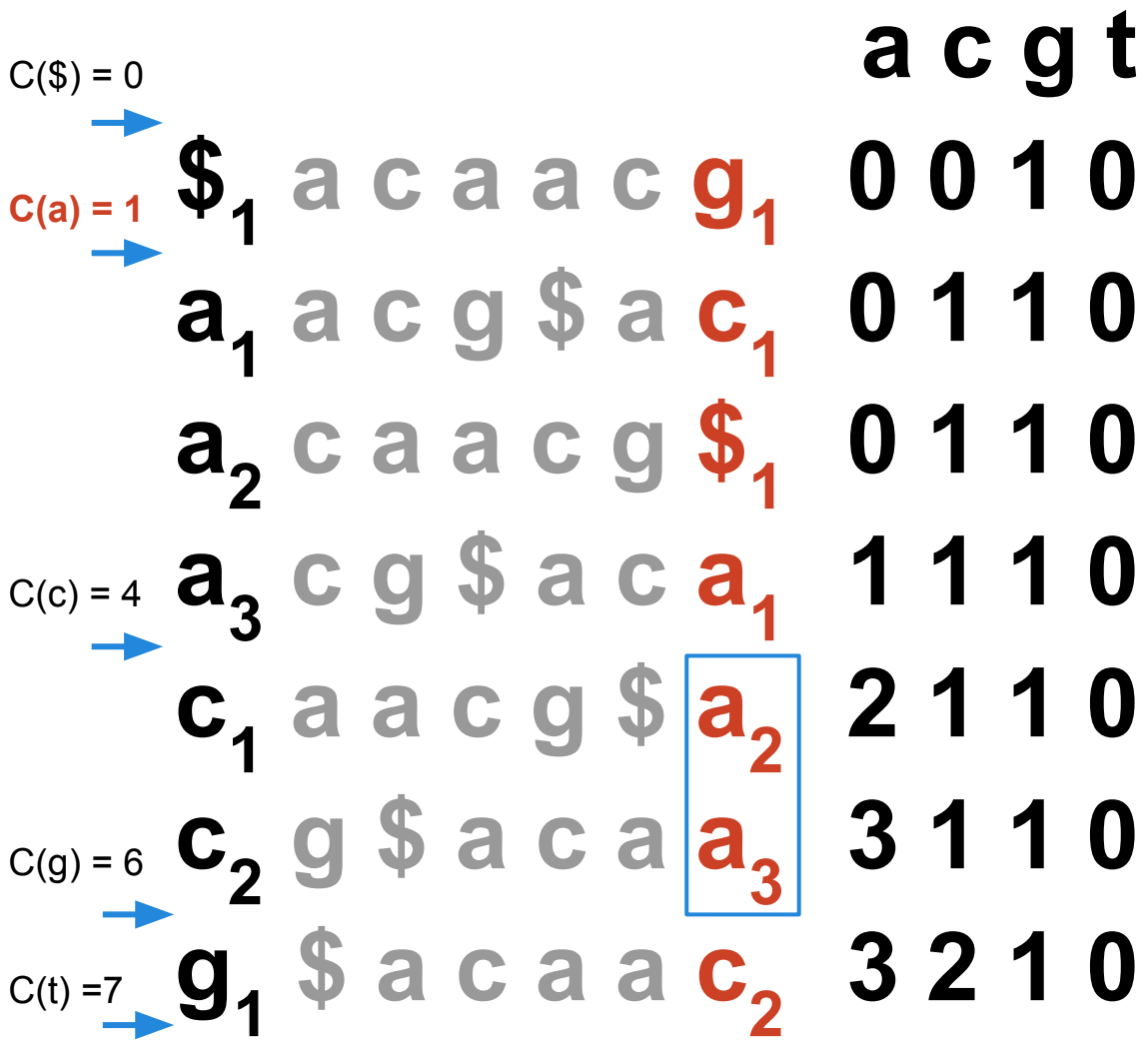


Pattern search

a a c

$R_L(ac) = \mathbf{C(a)} + 1$

$R_H(ac) = \mathbf{C(a)} + 3 - 1$

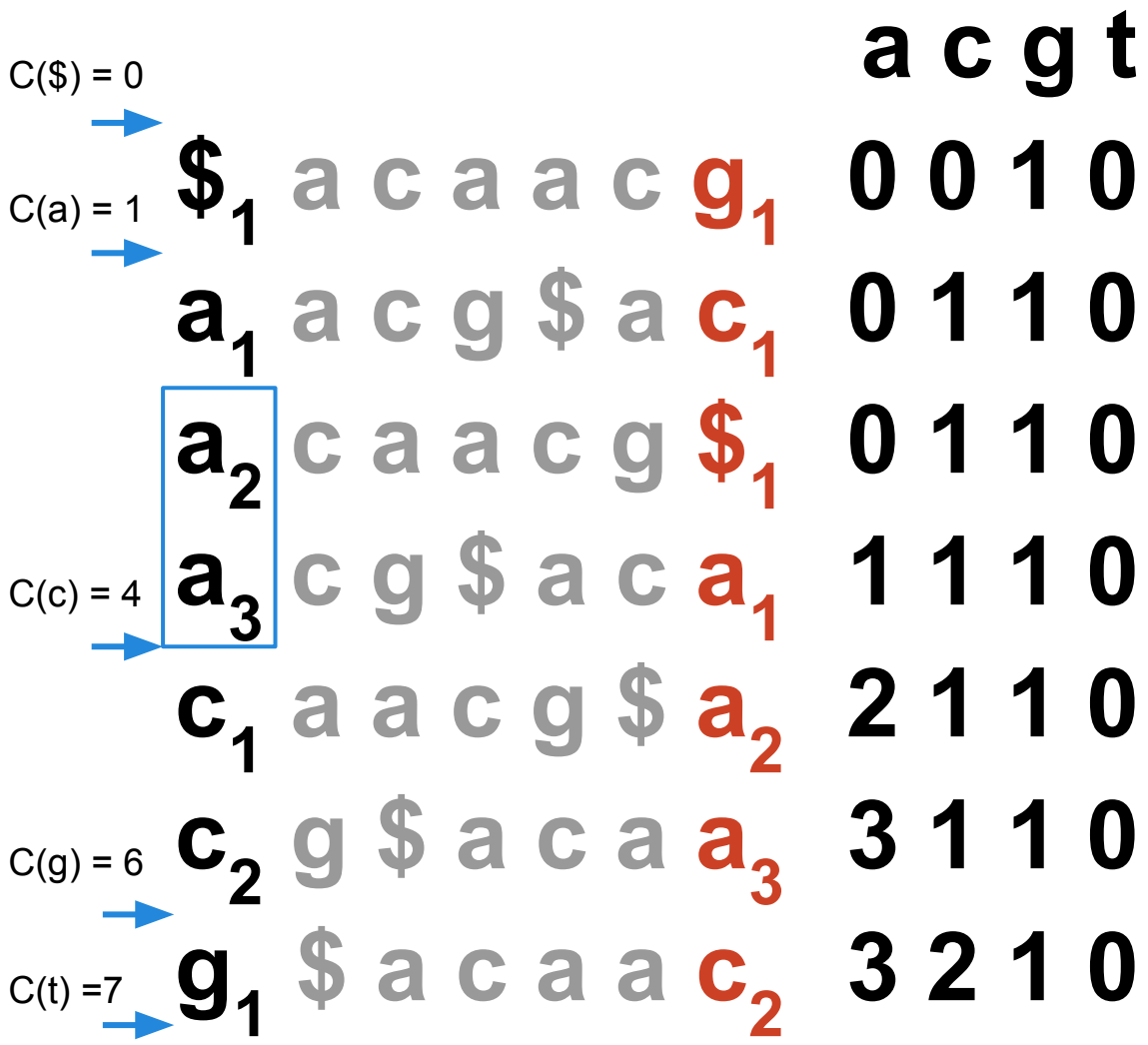


Pattern search

a a c

$R_L(ac) = 1 + 1 = 2$

$R_H(ac) = 1 + 3 - 1 = 3$

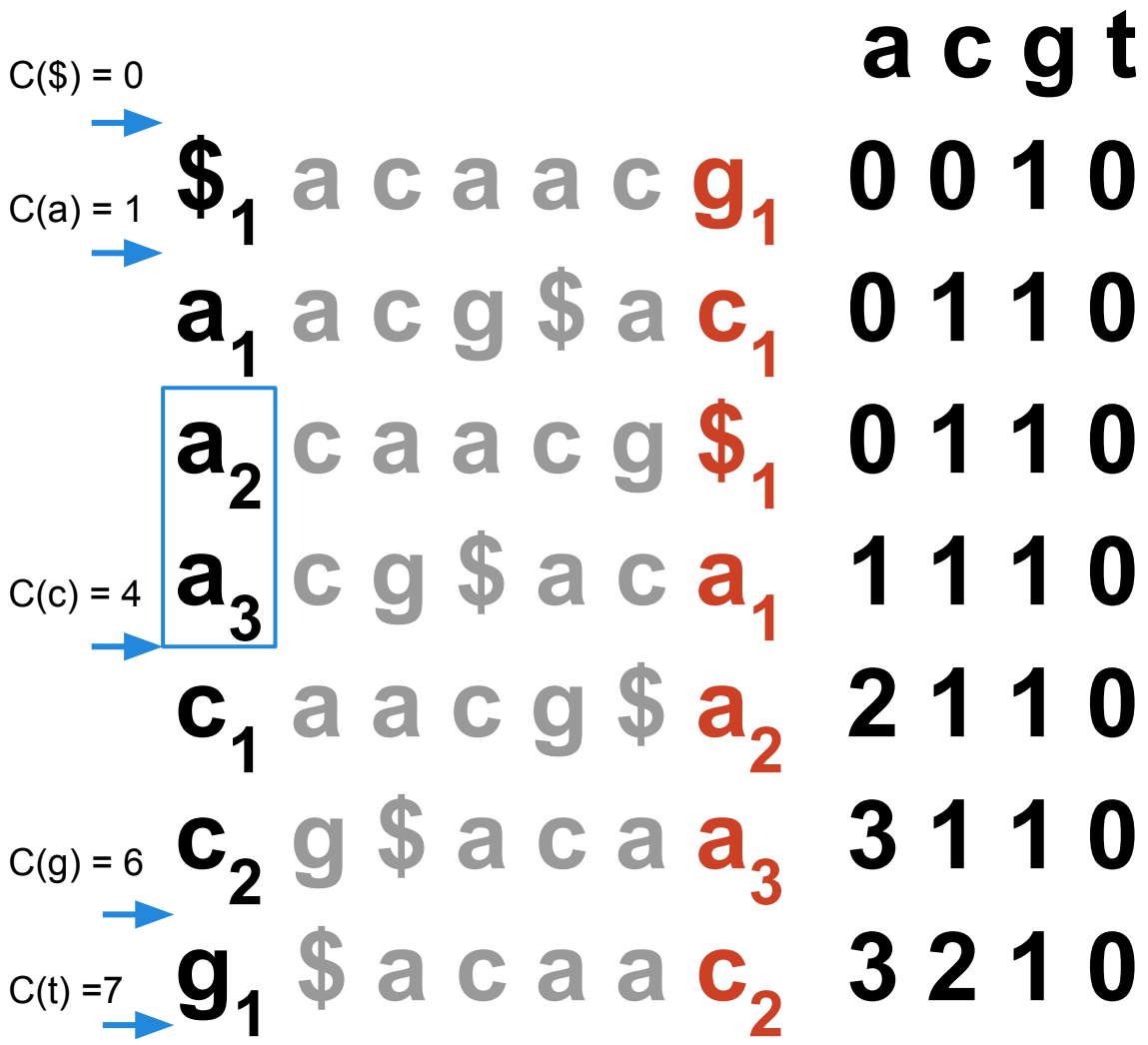


Pattern search

a a c

$R_L(aac) = C(a) + O(a, R_L(ac) - 1)$

$R_H(aac) = C(a) + O(a, R_H(ac)) - 1$



Pattern search

a a c

$R_L(aac) = C(a) + \mathbf{O(a, 2 - 1)}$

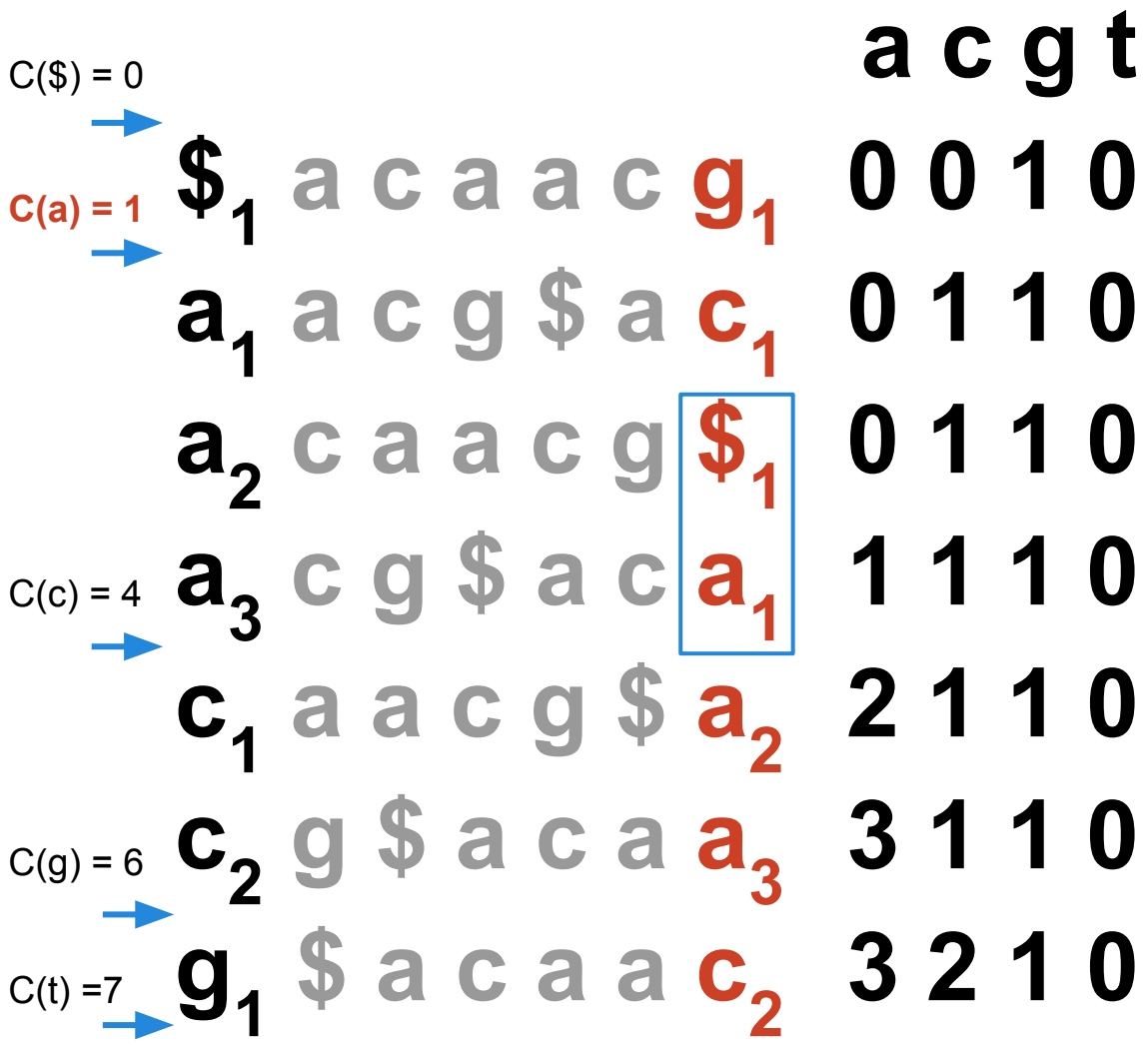
$R_H(aac) = C(a) + \mathbf{O(a, 3)} - 1$

Pattern search

a a c

$R_L(aac) = \mathbf{C(a)} + 0$

$R_H(aac) = \mathbf{C(a)} + 1 - 1$

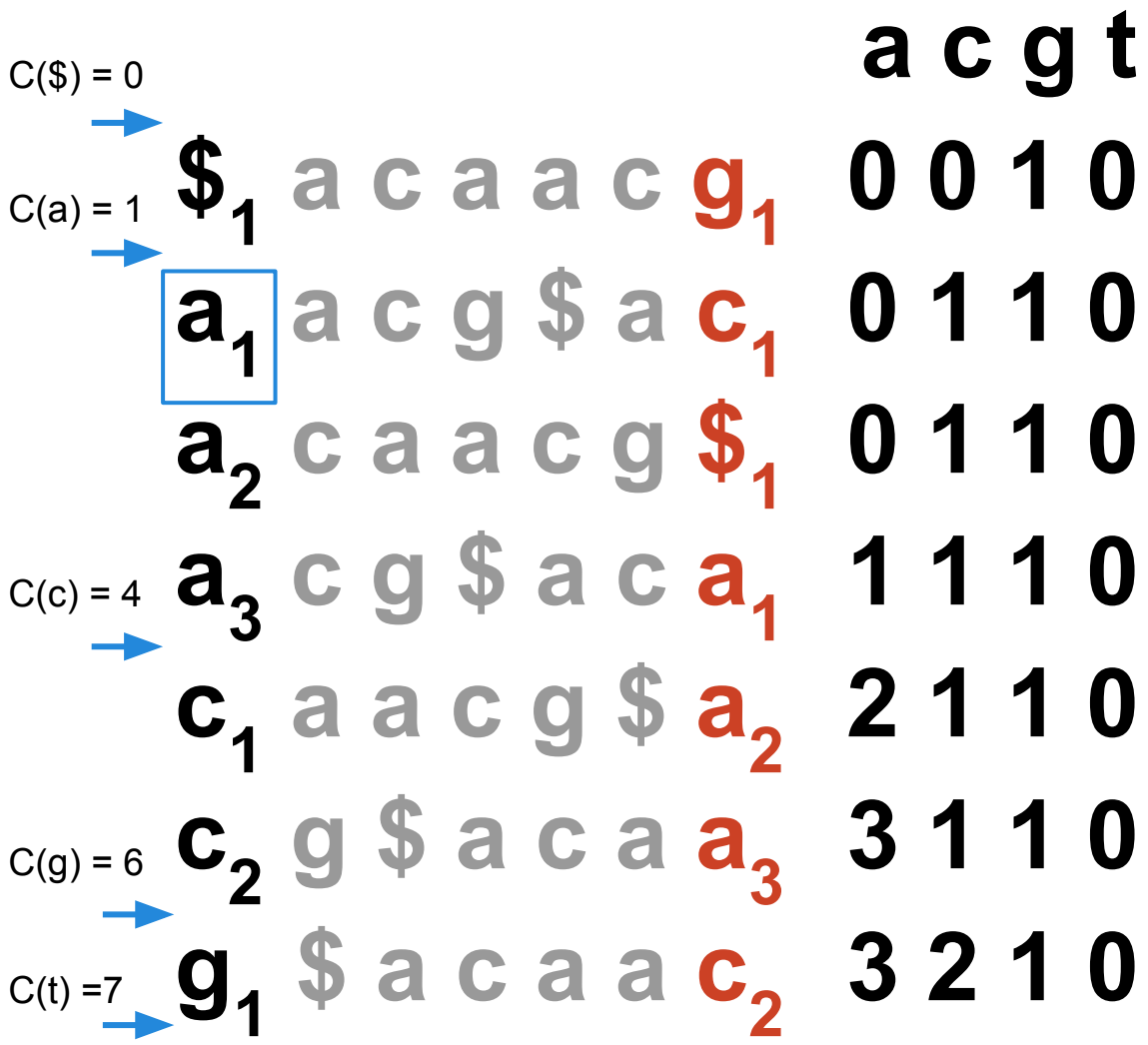


Pattern search

a a c

$R_L(aac) = 1 + 0 = 1$

$R_H(aac) = 1 + 1 - 1 = 1$



Pattern search

a a c

$R_L(aac) = 1$

$R_H(aac) = 1$

		a	c	g	t
C($) = 0					
	$_1$ a c a a c g$_1$	0	0	1	0
C(a) = 1	a$_1$ a c g $ a c$_1$	0	1	1	0
	a$_2$ c a a c g $_1$	0	1	1	0
C(c) = 4	a$_3$ c g $ a c a$_1$	1	1	1	0
	c$_1$ a a c g $ a$_2$	2	1	1	0
C(g) = 6	c$_2$ g $ a c a a$_3$	3	1	1	0
C(t) =7	g$_1$ $ a c a a c$_2$	3	2	1	0

Pattern search

For how long does it work?

Pattern search

For how long does it work?

O(m)

Pattern search with errors

$_1$ a c a a c g$_1$

a$_1$ a c g $ a c$_1$

a$_2$ c a a c g $$_1$

a$_3$ c g $ a c a$_1$

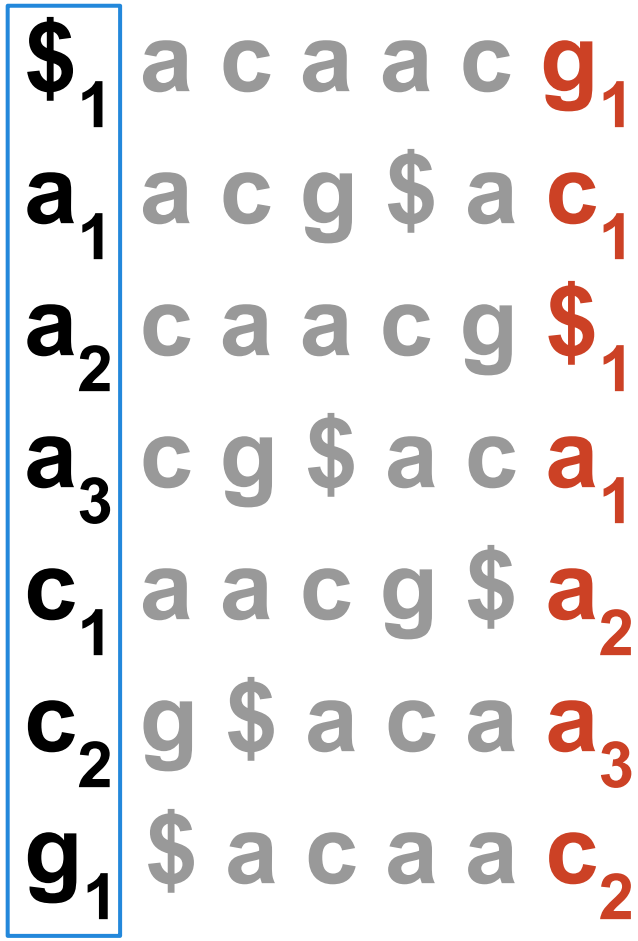
c$_1$ a a c g $ a$_2$

c$_2$ g $ a c a a$_3$

g$_1$ $ a c a a c$_2$

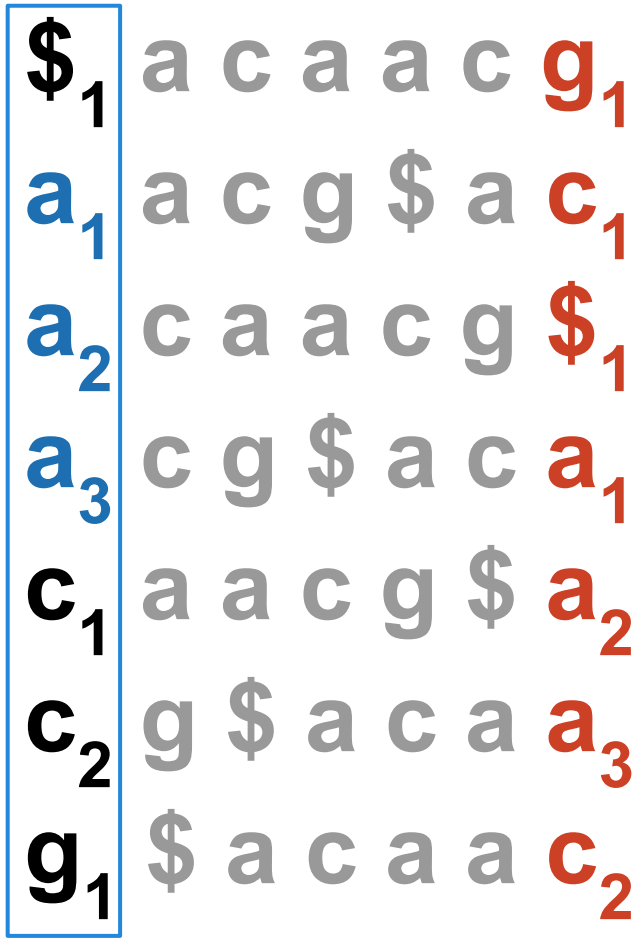
g c a

Pattern search with errors



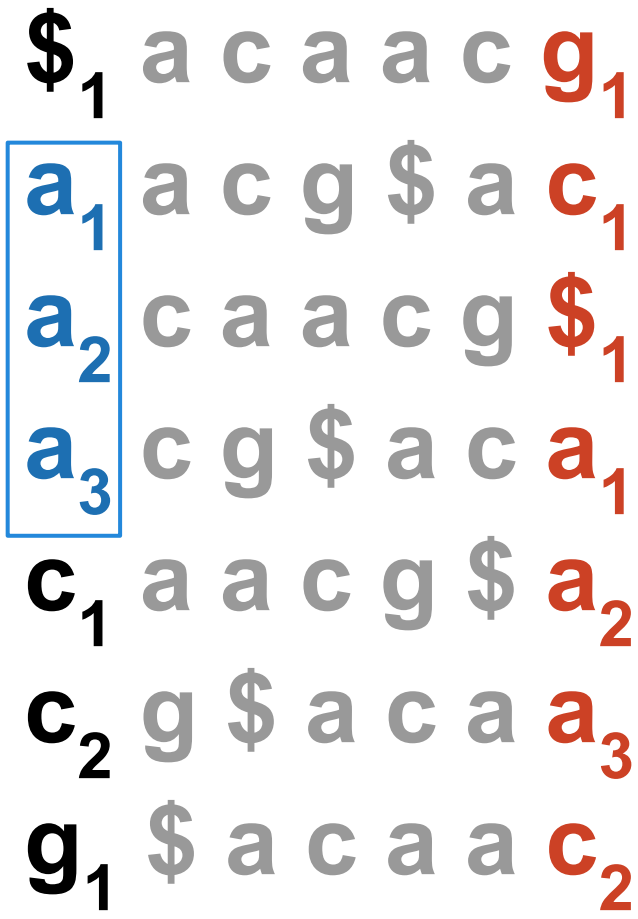
g c a

Pattern search with errors



g c a

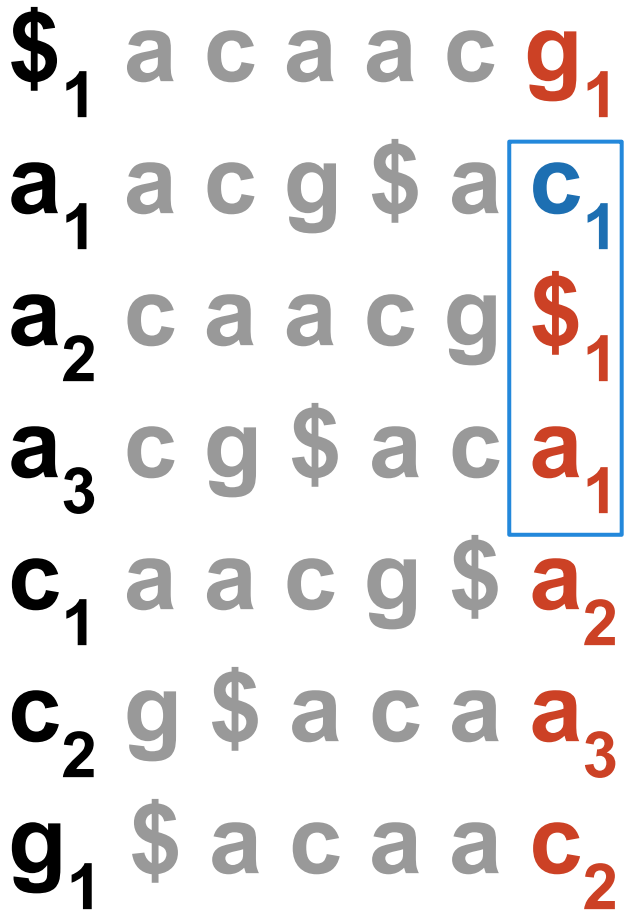
Pattern search with errors



g c a

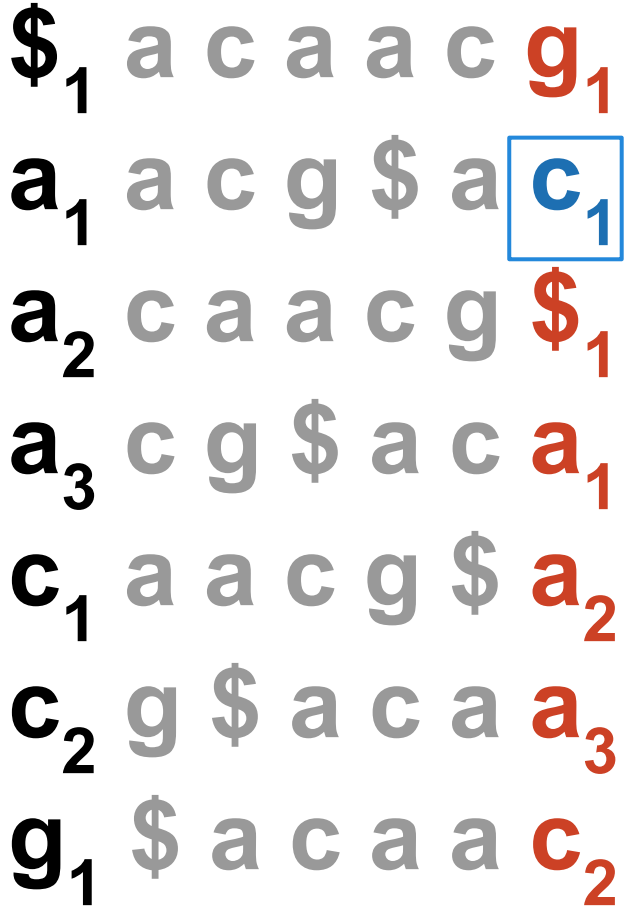
Pattern search with errors

Pattern search with errors



g c a

Pattern search with errors



g c a

Pattern search with errors

$_1$ a c a a c g$_1$

a$_1$ a c g $ a c$_1$

a$_2$ c a a c g $_1$

a$_3$ c g $ a c a$_1$

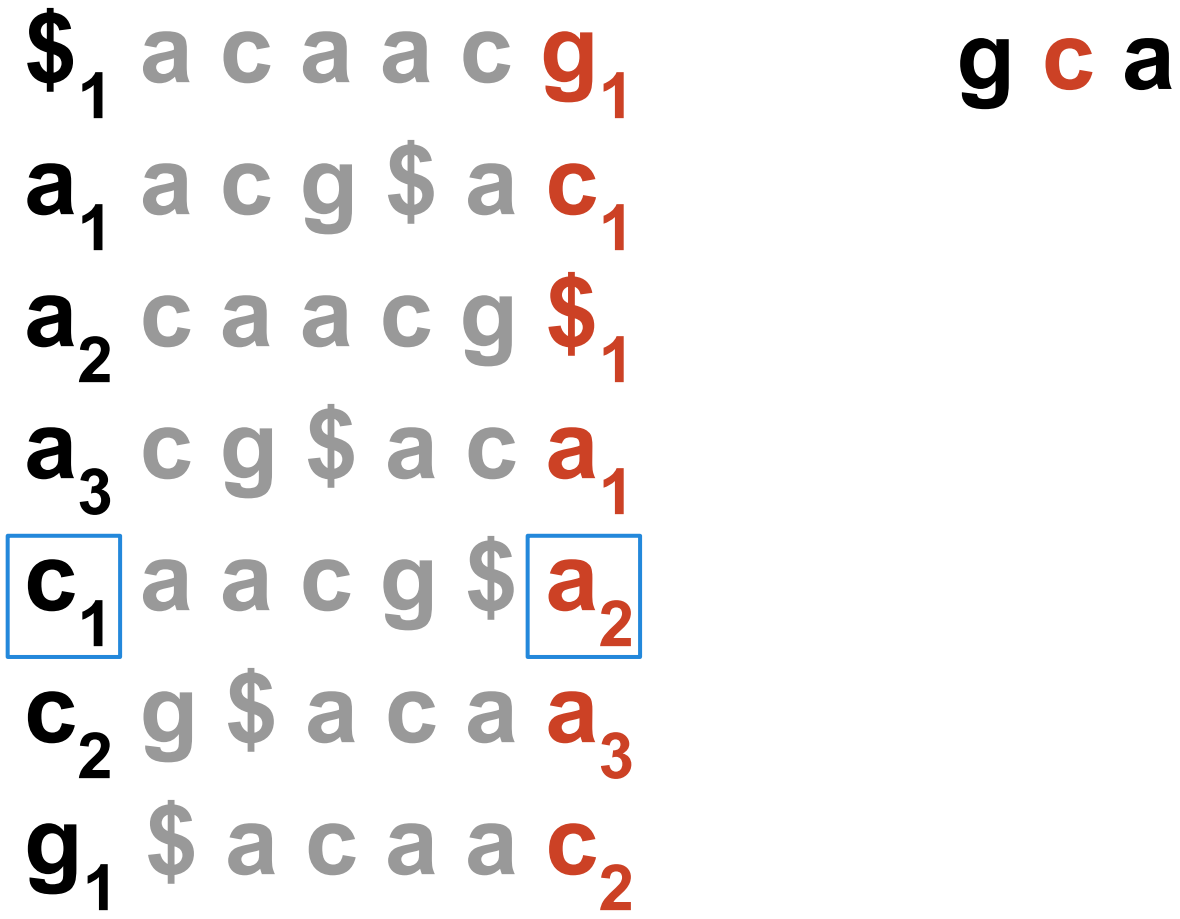
c$_1$ a a c g $ a$_2$

c$_2$ g $ a c a a$_3$

g$_1$ $ a c a a c$_2$

g c a

Pattern search with errors



Pattern search with errors

$_1$ a c a a c g$_1$

a$_1$ a c g $ a c$_1$

a$_2$ c a a c g $$_1$

a$_3$ c g $ a c a$_1$

c$_1$ a a c g $ a$_2$

c$_2$ g $ a c a a$_3$

g$_1$ $ a c a a c$_2$

g c a

Pattern search with errors

$_1$ a c a a c g$_1$

a$_1$ a c g $ a c$_1$

a$_2$ c a a c g $$_1$

a$_3$ c g $ a c a$_1$

c$_1$ a a c g $ a$_2$

c$_2$ g $ a c a a$_3$

g$_1$ $ a c a a c$_2$

g c a

Pattern search with errors

$_1$ a c a a c g$_1$
a$_1$ a c g $ a c$_1$
a$_2$ c a a c g $$_1$
a$_3$ c g $ a c a$_1$
c$_1$ a a c g $ a$_2$
c$_2$ g $ a c a a$_3$
g$_1$ $ a c a a c$_2$

g c a

Pattern search with errors

$_1$ a c a a c g$_1$

a$_1$ a c g $ a c$_1$

a$_2$ c a a c g $$_1$

a$_3$ c g $ a c a$_1$

c$_1$ a a c g $ a$_2$

c$_2$ g $ a c a a$_3$

g$_1$ $ a c a a c$_2$

g c a

Pattern search with errors

$_1$ a c a a c g$_1$

a$_1$ a c g $ a c$_1$

a$_2$ c a a c g $_1$

a$_3$ c g $ a c a$_1$

c$_1$ a a c g $ a$_2$

c$_2$ g $ a c a a$_3$

g$_1$ $ a c a a c$_2$

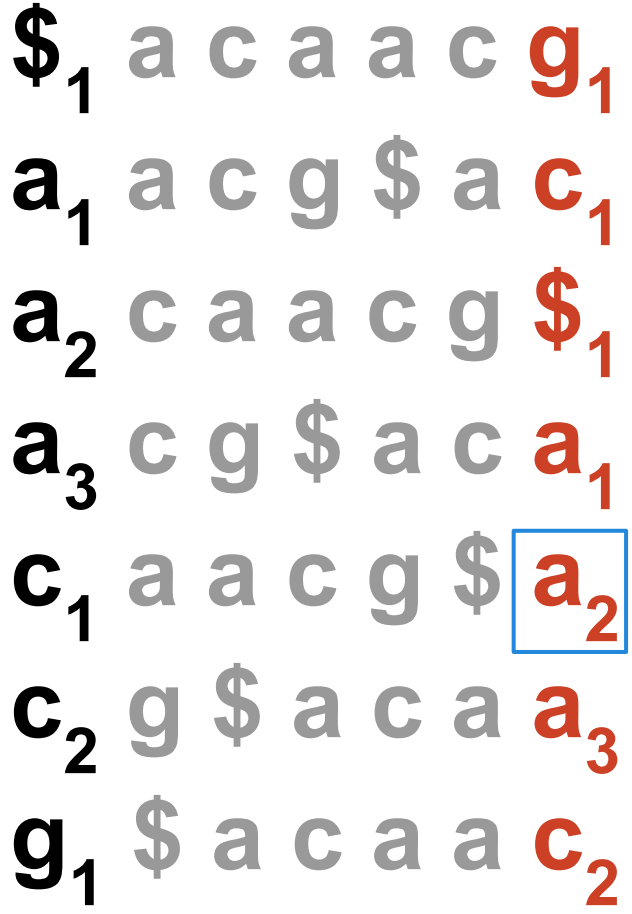
c c a

Pattern search with errors

$_1$ a c a a c g$_1$
a$_1$ a c g $ a c$_1$
a$_2$ c a a c g $$_1$
a$_3$ c g $ a c a$_1$
c$_1$ a a c g $ a$_2$
c$_2$ g $ a c a a$_3$
g$_1$ $ a c a a c$_2$

c c a

Pattern search with errors



a c a

Pattern search with errors

$_1$ a c a a c g$_1$

a$_1$ a c g $ a c$_1$

a$_2$ c a a c g $$_1$

a$_3$ c g $ a c a$_1$

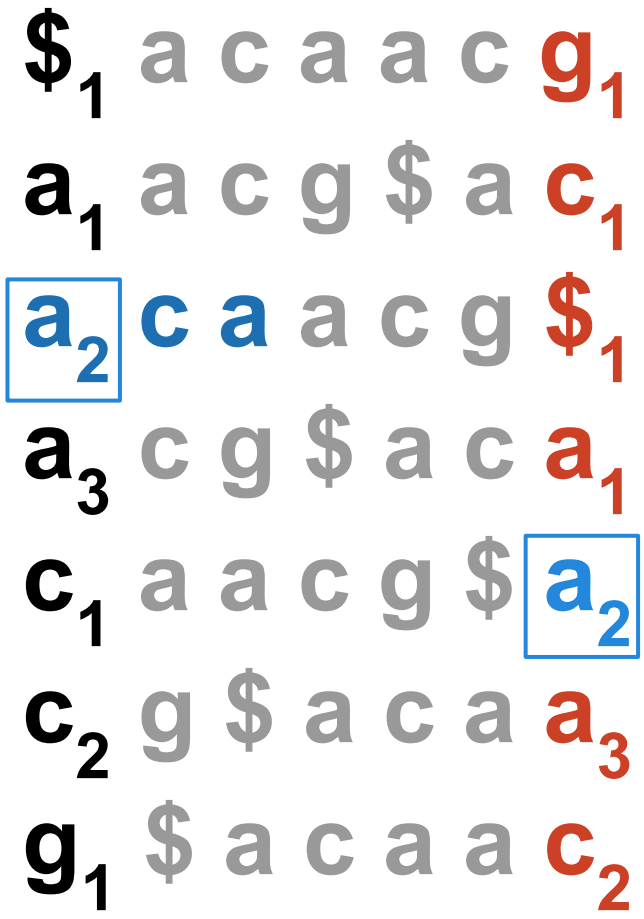
c$_1$ a a c g $ a$_2$

c$_2$ g $ a c a a$_3$

g$_1$ $ a c a a c$_2$

a c a

Pattern search with errors



a c a

Pattern search with errors

$_1$ a c a a c g$_1$
a$_1$ a c g $ a c$_1$
a$_2$ c a a c g $$_1$
a$_3$ c g $ a c a$_1$
c$_1$ a a c g $ a$_2$
c$_2$ g $ a c a a$_3$
g$_1$ $ a c a a c$_2$

a c a

Bowtie

- Backtracking

Bowtie

- Backtracking
  - Quality-aware
  - Limit total number of backtracks
  - Limit quality distance

Bowtie

- Backtracking
  - Quality-aware
  - Limit total number of backtracks
  - Limit quality distance
- Seeds
  - Selected at high-quality end
  - Used to prevent excessive backtracking

Burrows-Wheeler mirror transform

a c a a c g $ **a a c**

Burrows-Wheeler mirror transform

$ g c a a c a a a c

Burrows-Wheeler mirror transform

$ g c a a c a c a a

Burrows-Wheeler mirror transform

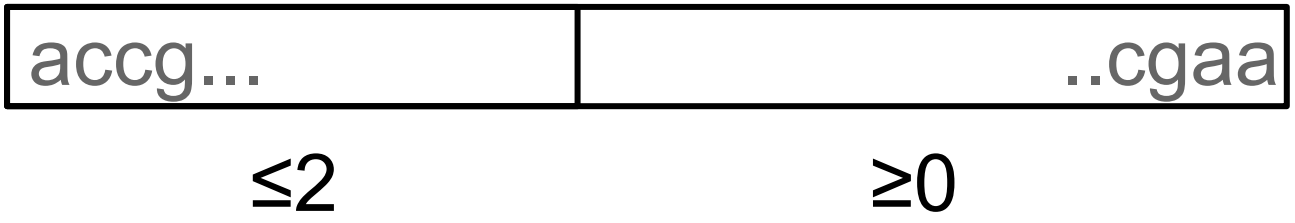
Burrows-Wheeler mirror transform

Bowtie

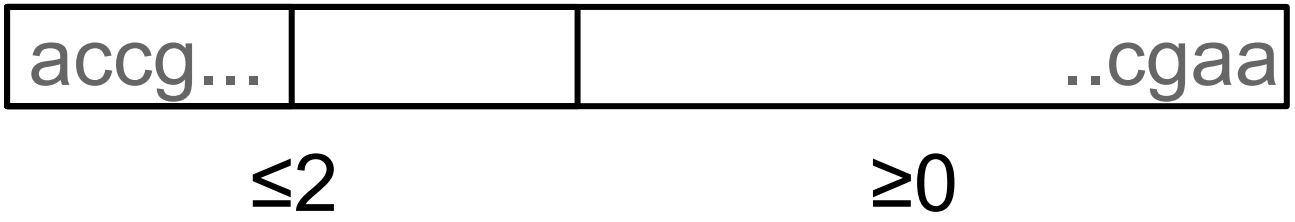
- Backtracking
  - Quality-aware
  - Limit total number of backtracks
  - Limit quality distance
- Seeds
  - Selected at high-quality end
  - Used to prevent excessive backtracking

Seeds

Seeds

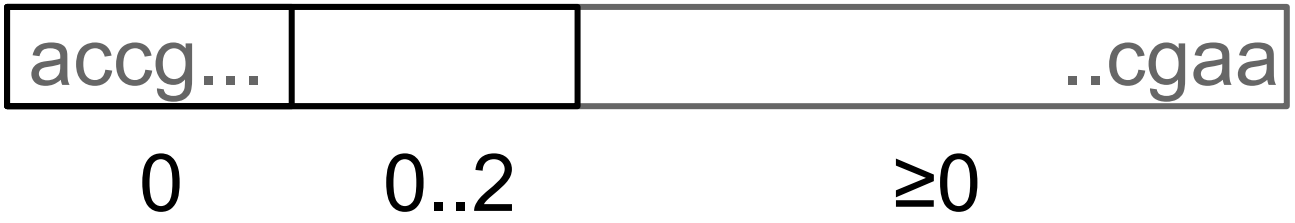


Seeds

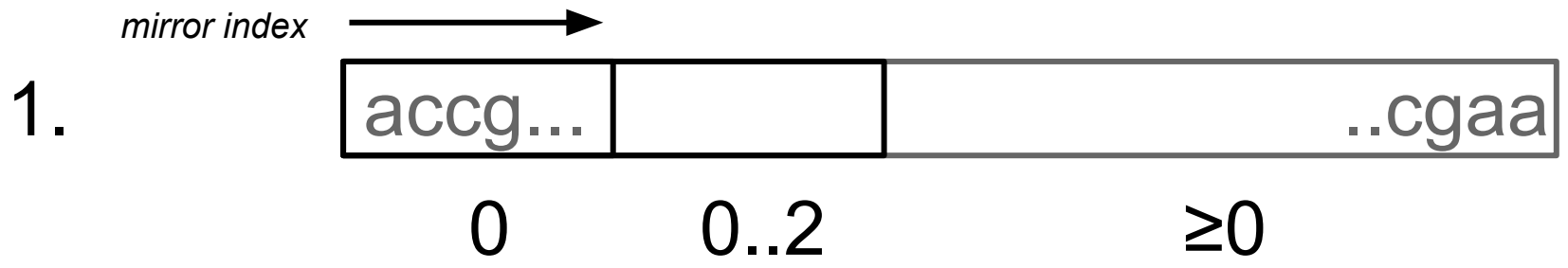


Seeds

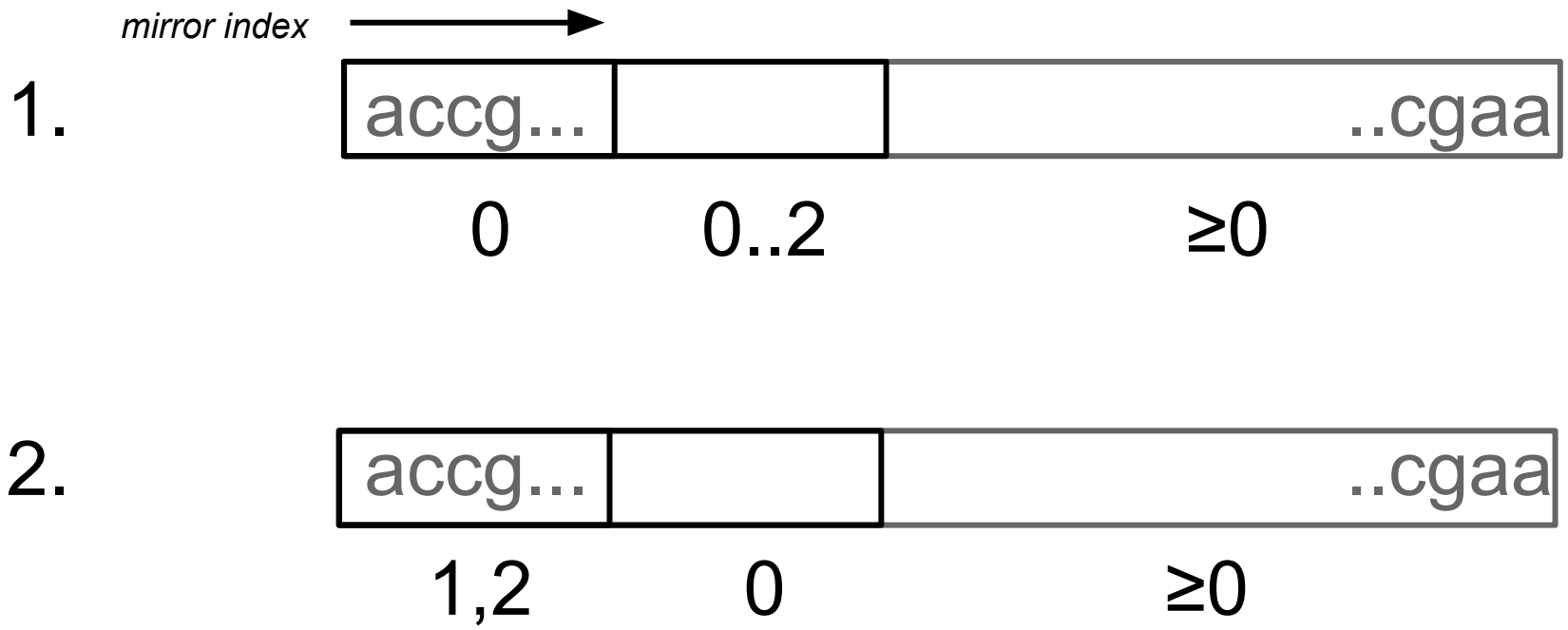
1.



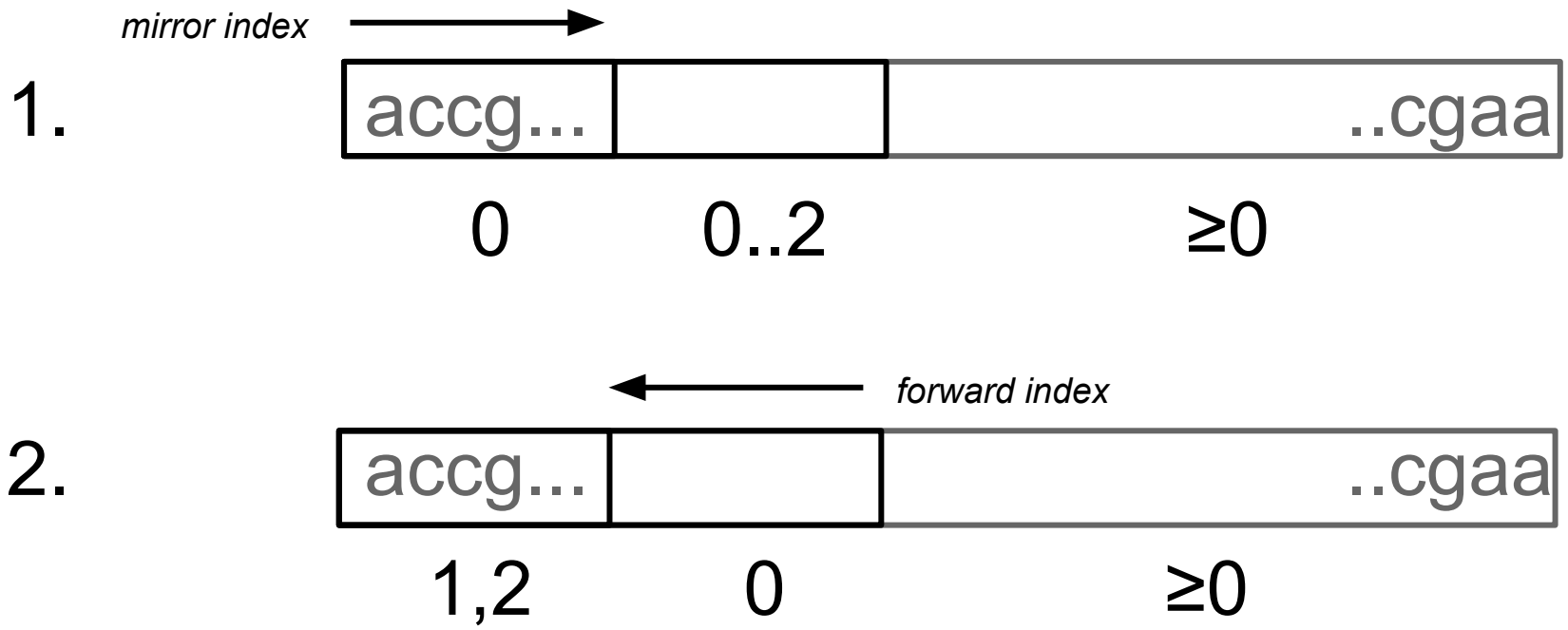
Seeds



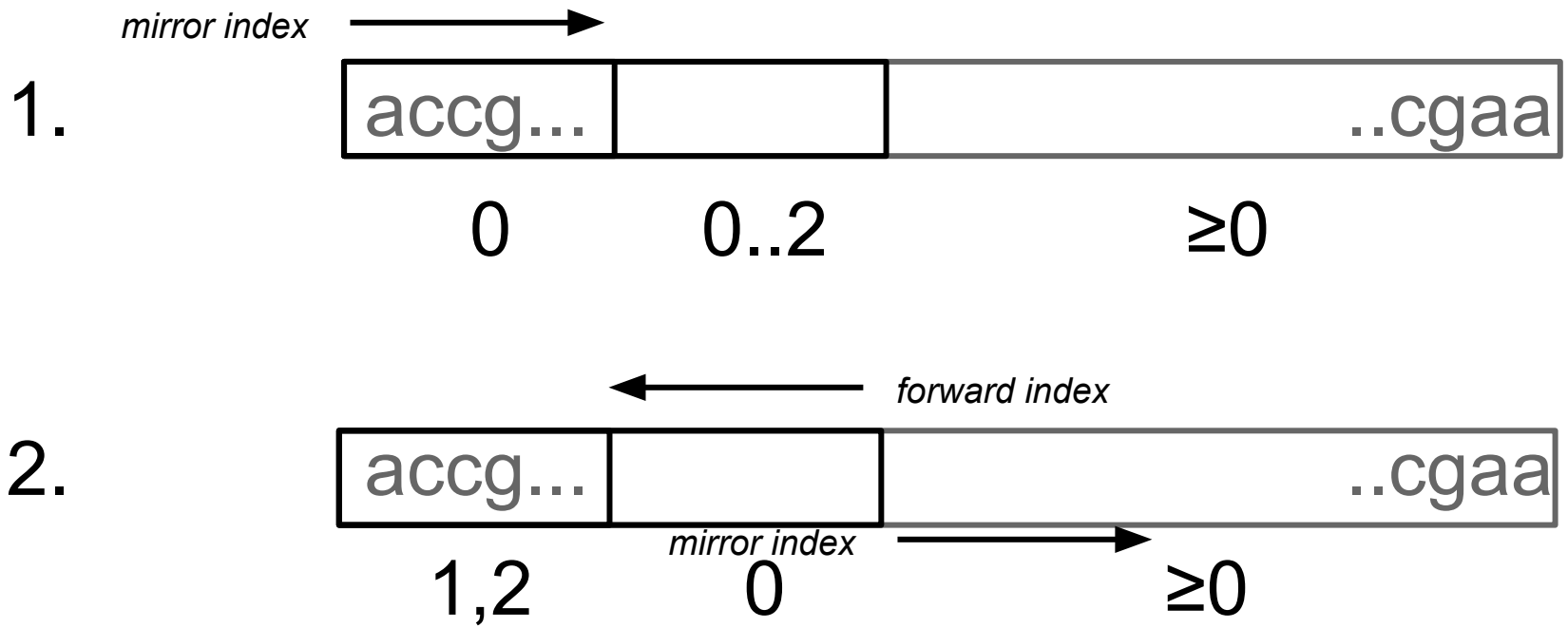
Seeds



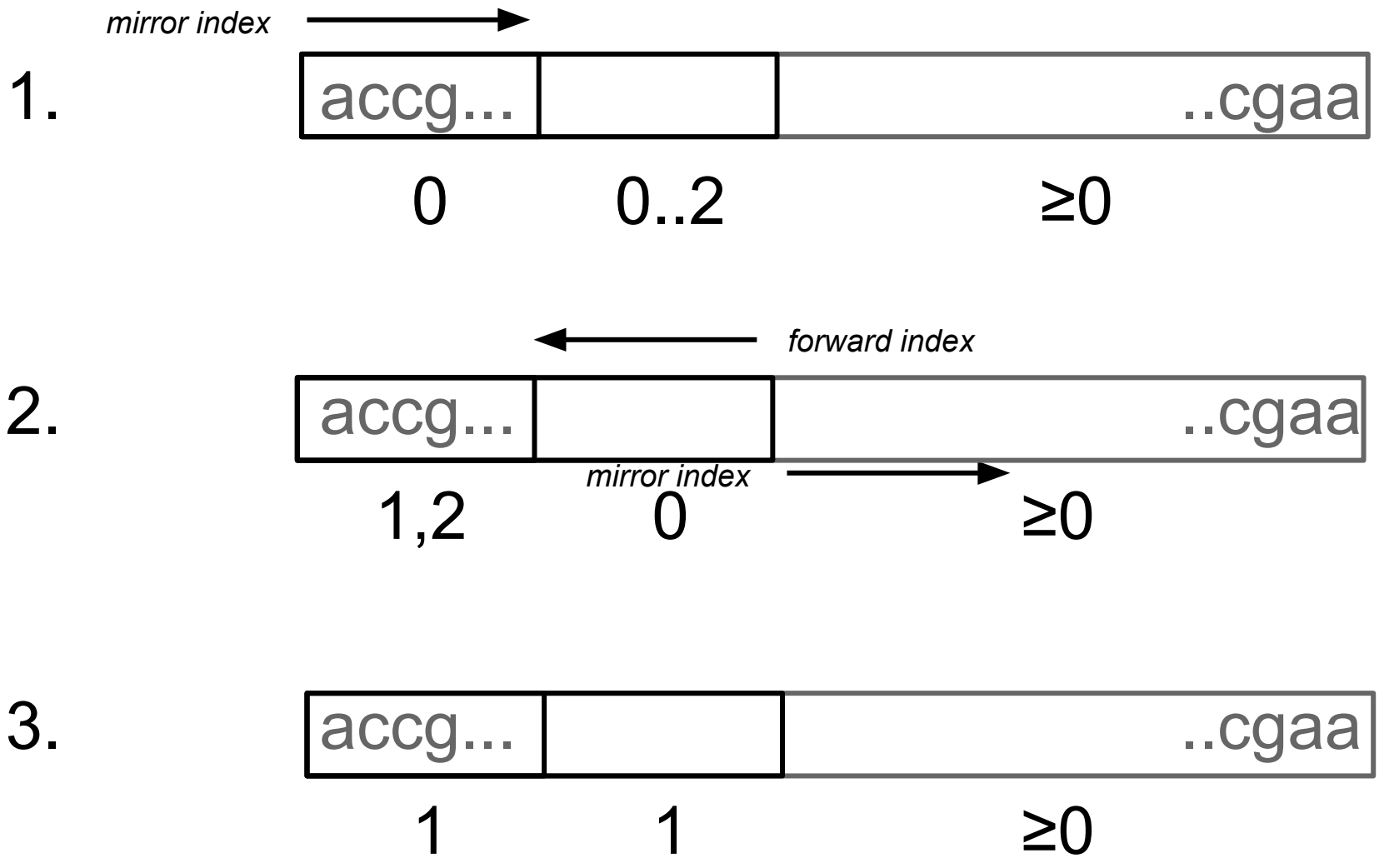
Seeds



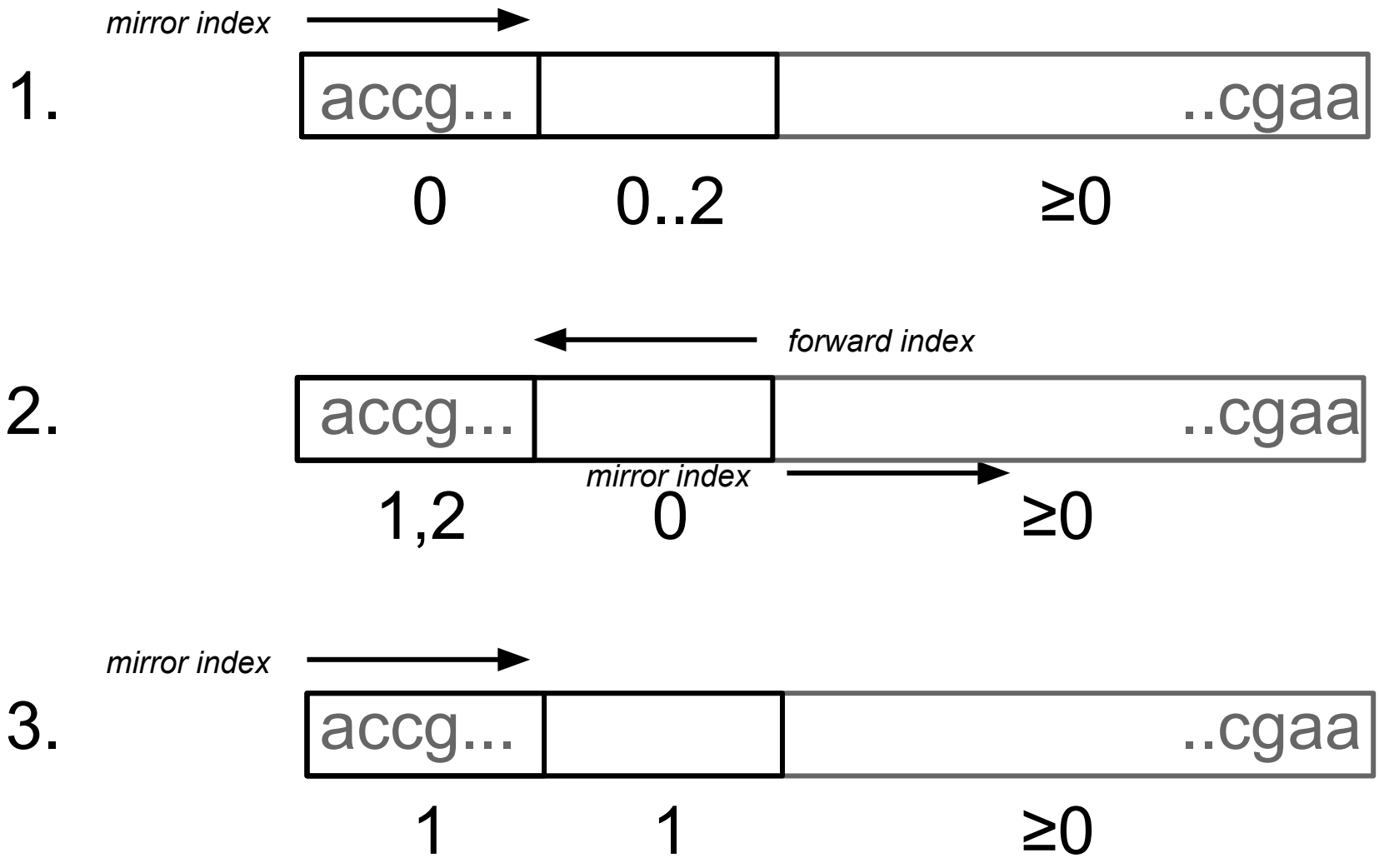
Seeds


mirror index
1.
accg...
..cgaa
0
0..2
≥0
forward index
2.
accg...
..cgaa
mirror index
1,2
0
≥0

Seeds


mirror index
1.
accg...
..cgaa
0
0..2
≥0
forward index
2.
accg...
..cgaa
mirror index
1,2
0
≥0
3.
accg...
..cgaa
1
1
≥0

Seeds


mirror index
1.
accg...
..cgaa
0
0..2
≥0
forward index
2.
accg...
..cgaa
mirror index
1,2
0
≥0
mirror index
3.
accg...
..cgaa
1
1
≥0

BWA

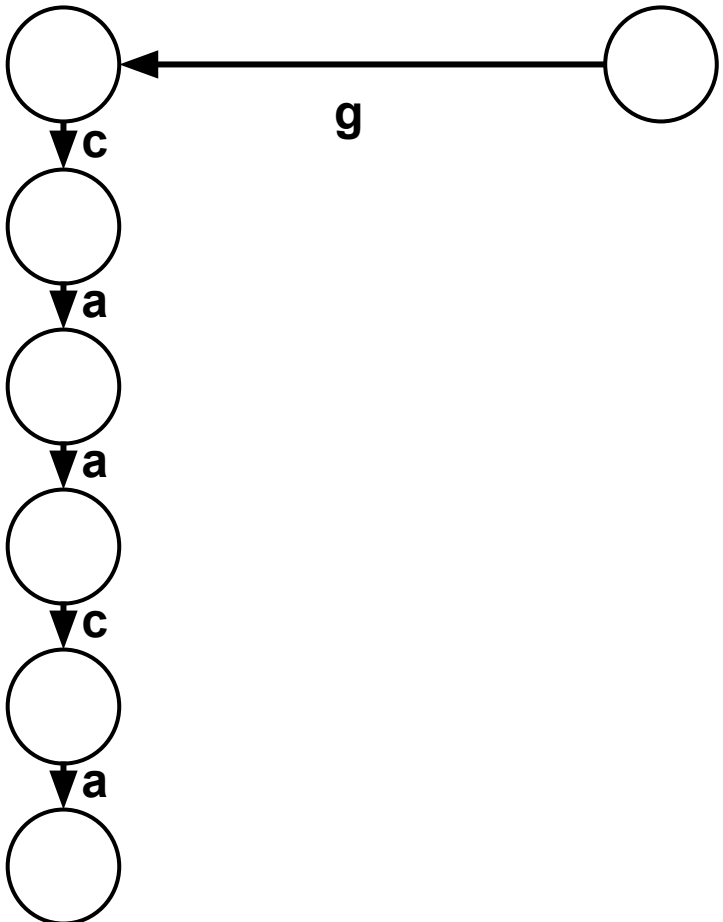
Mapping IonTorrent/454 reads

Prefix trie

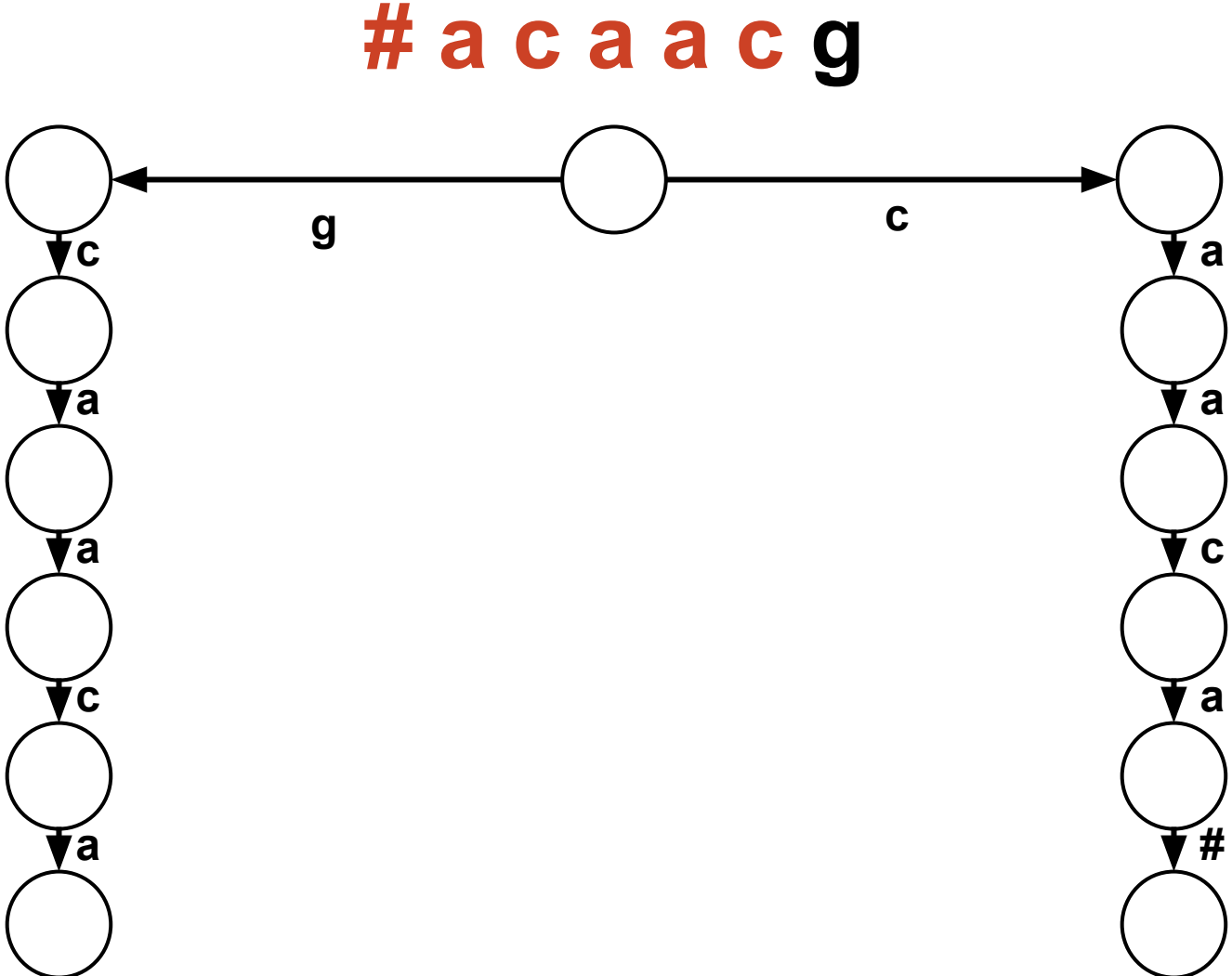
 # a c a a c g

Prefix trie

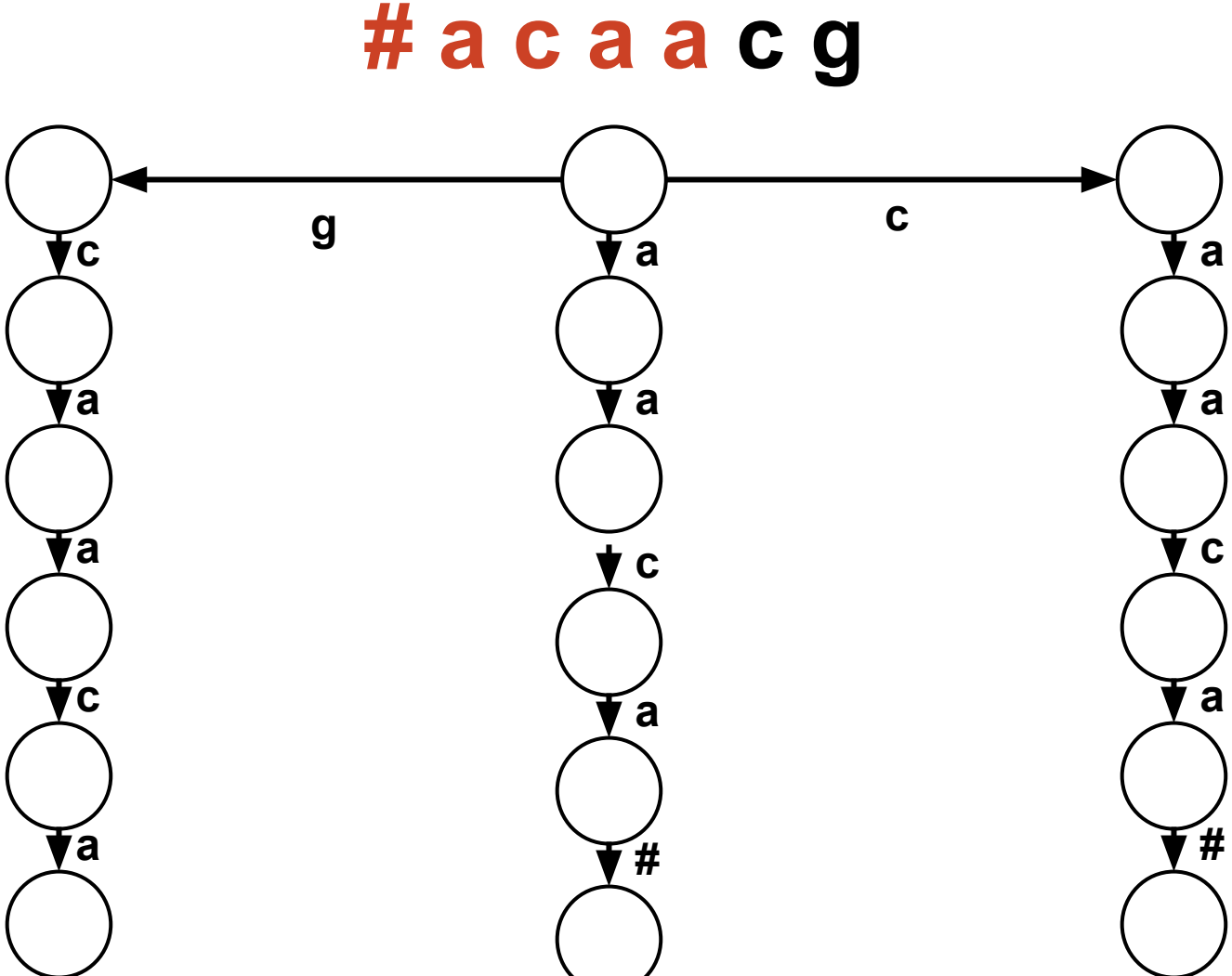
 # a c a a c g



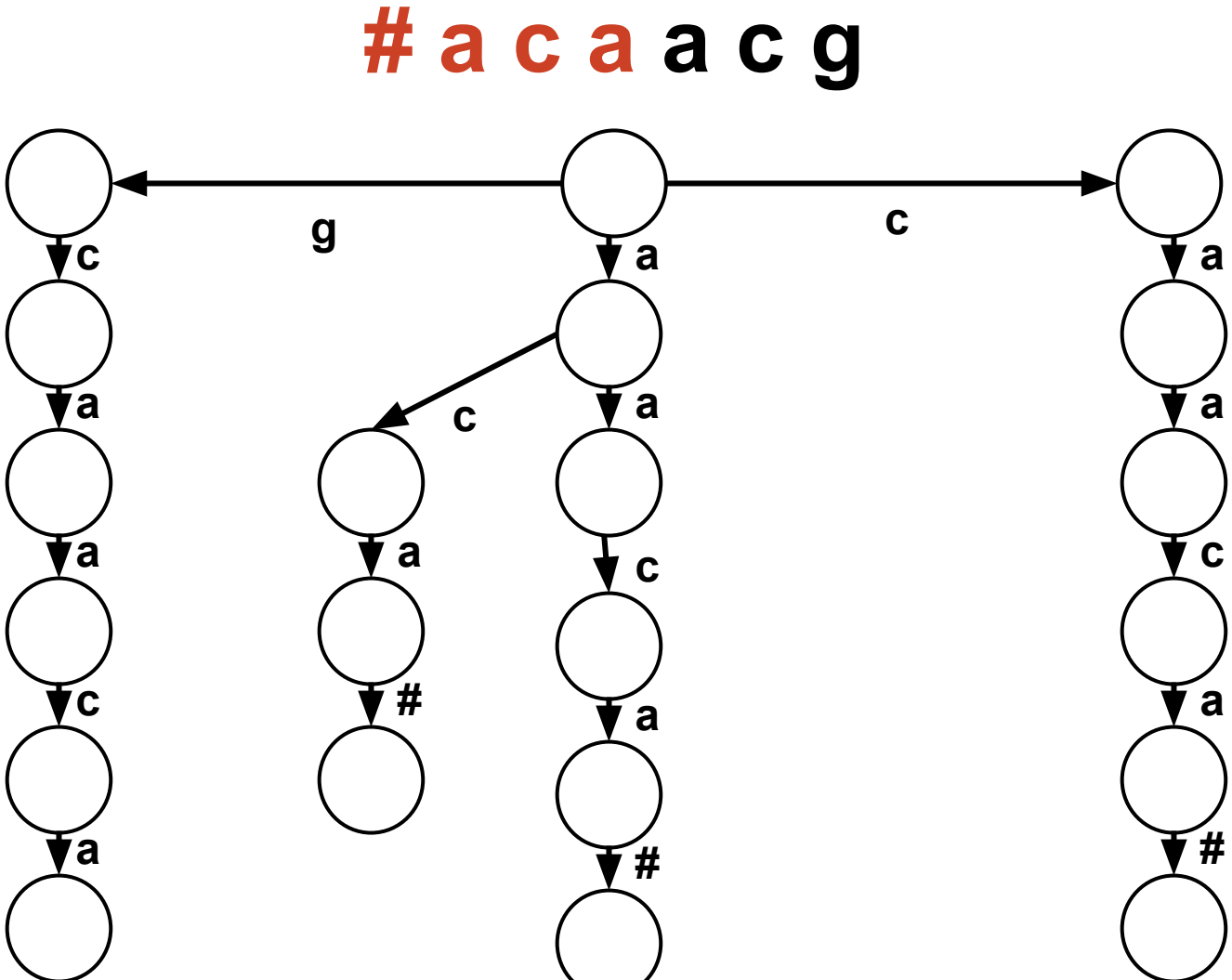
Prefix trie



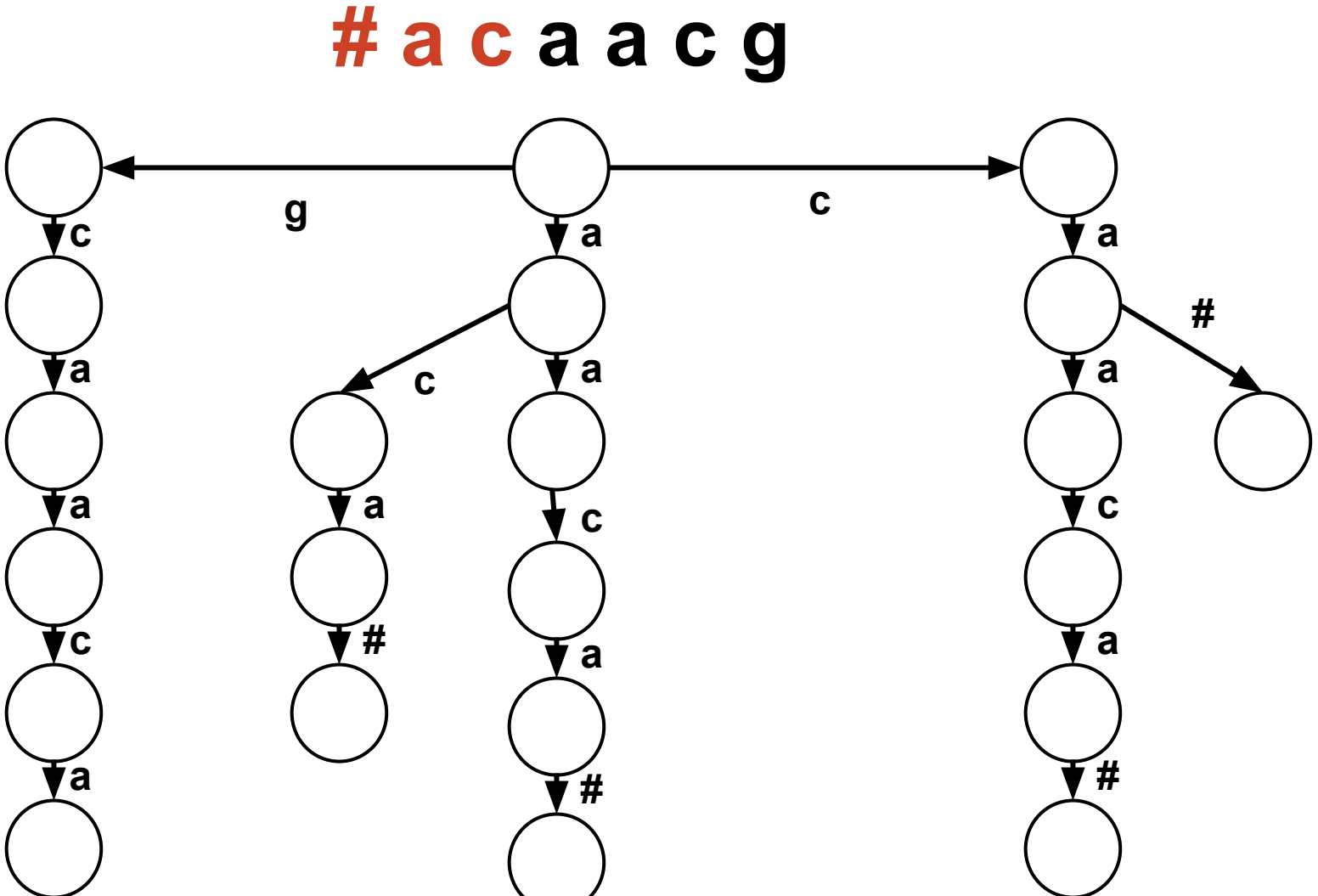
Prefix trie



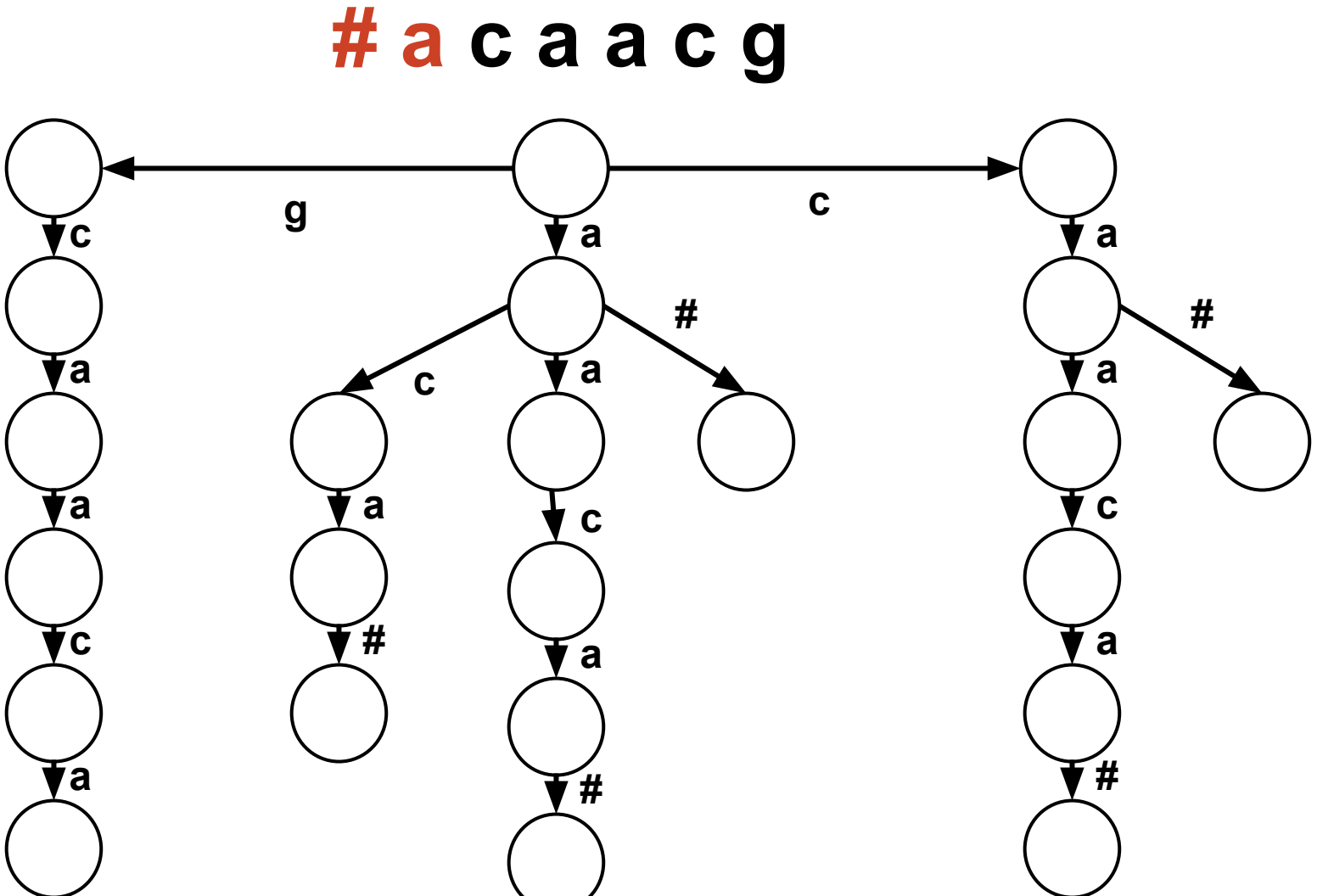
Prefix trie



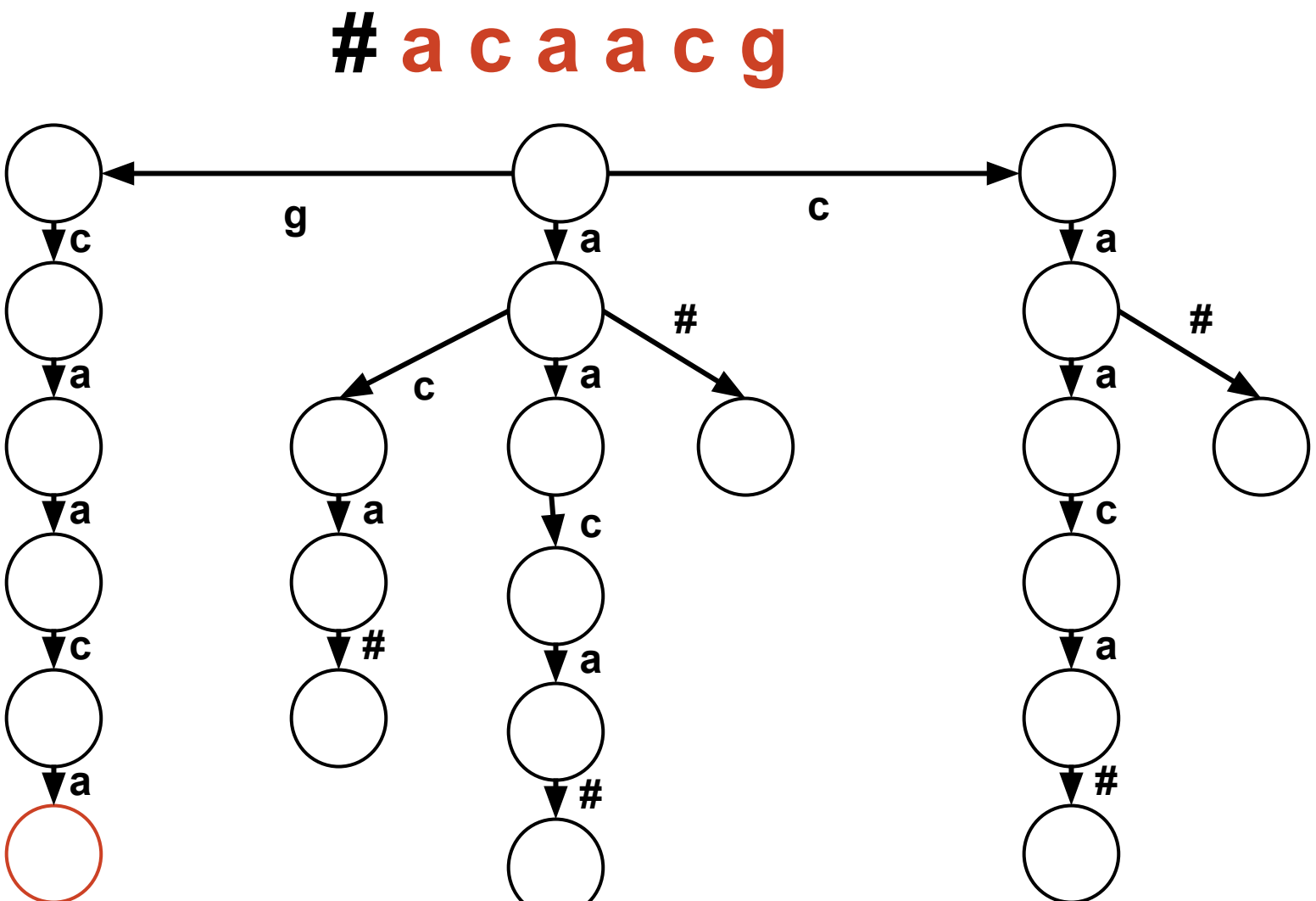
Prefix trie



Prefix trie



Prefix trie



Suffix array

$

a a c g $

a c a a c g $

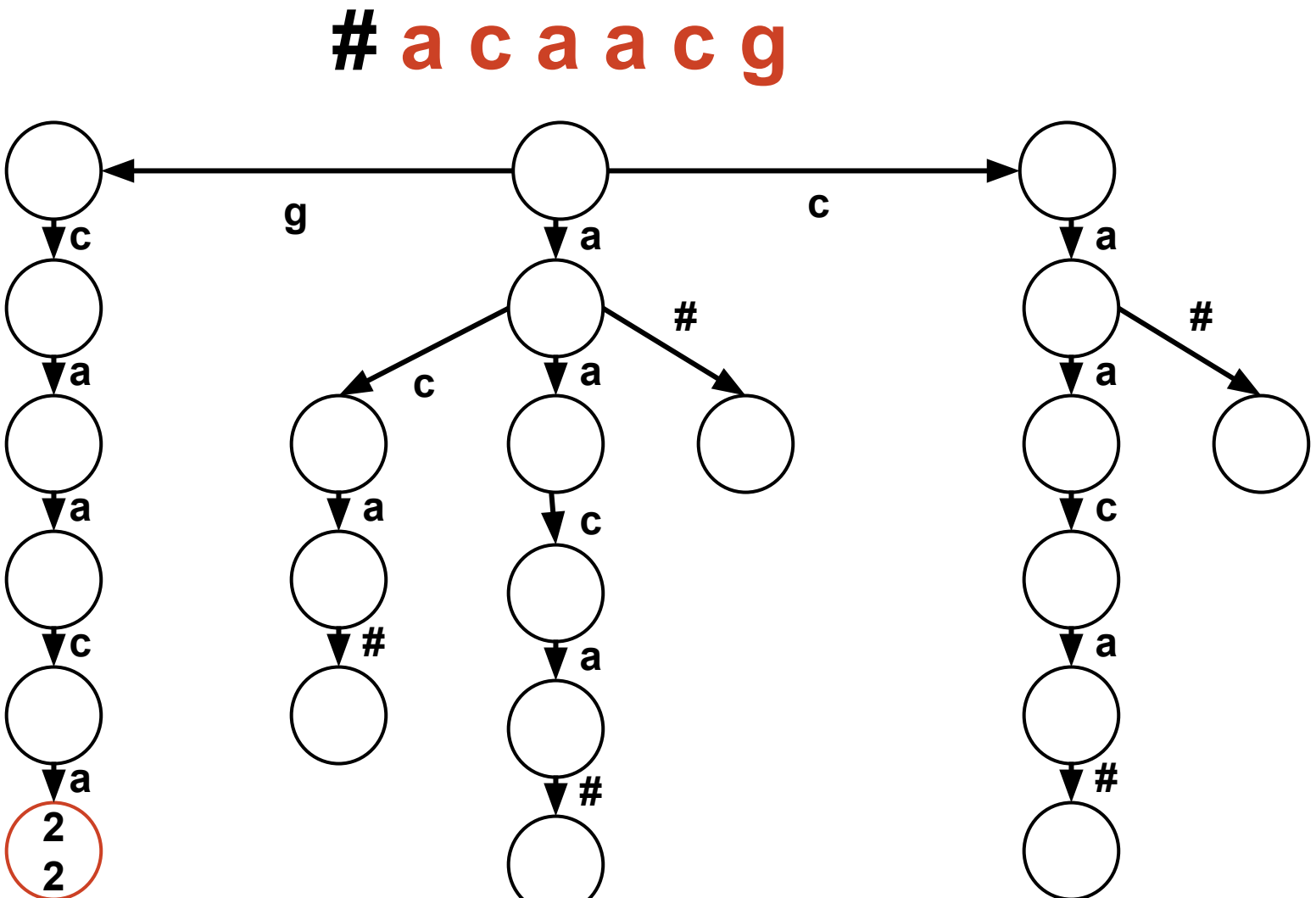
a c g $

c a a c g $

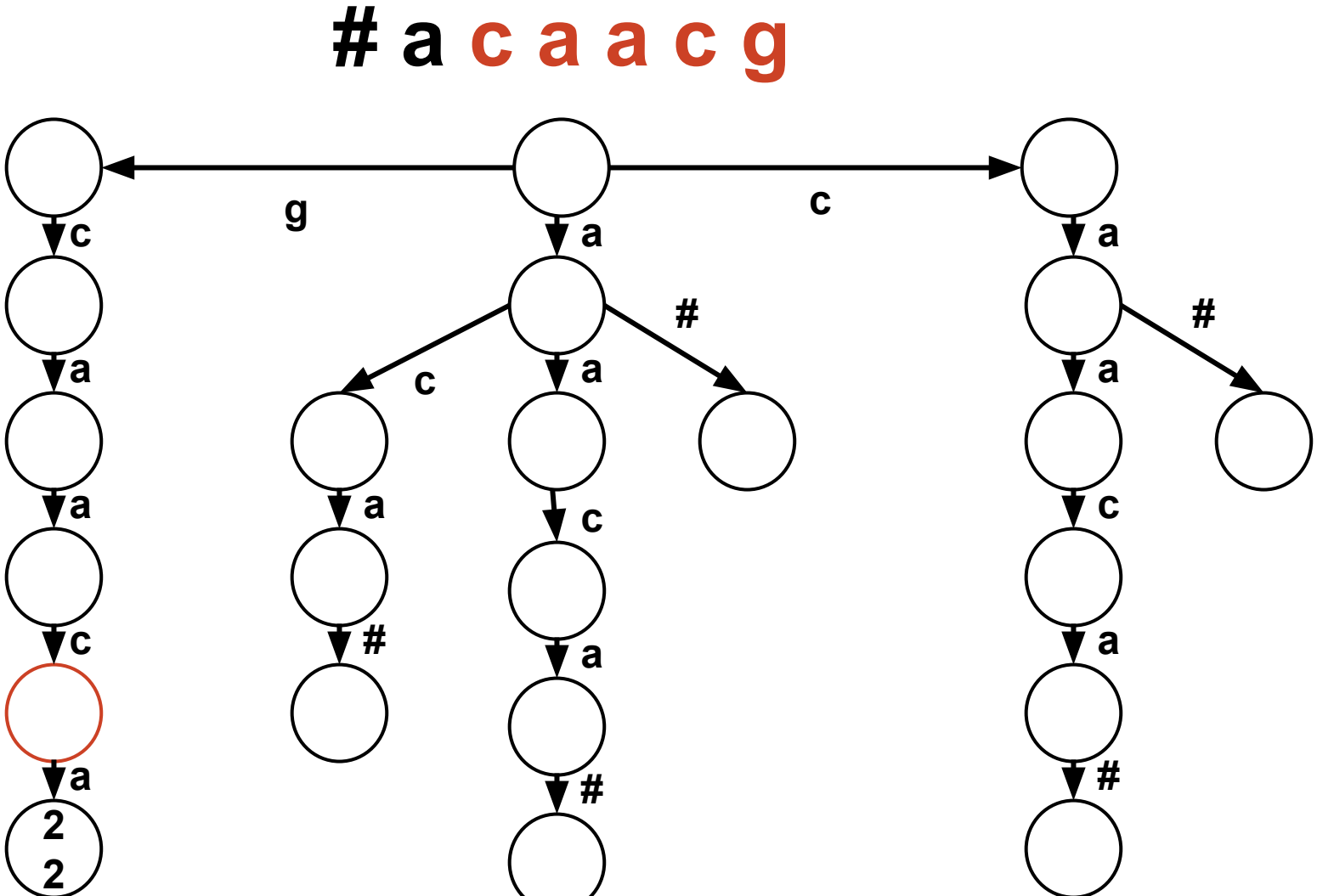
c g $

g $

Prefix trie



Prefix trie



Suffix array

$

a a c g $

a c a a c g $

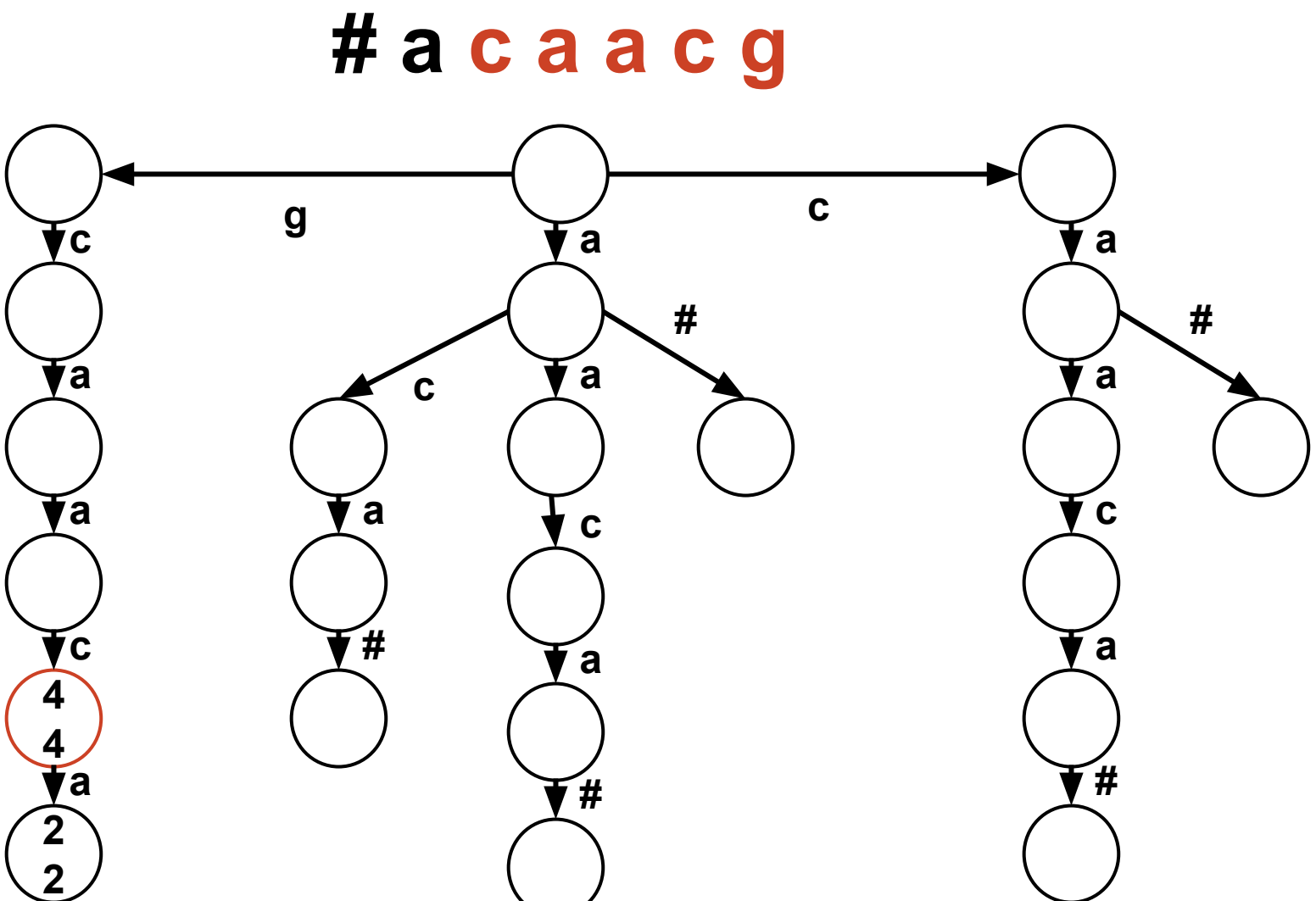
a c g $

c a a c g $

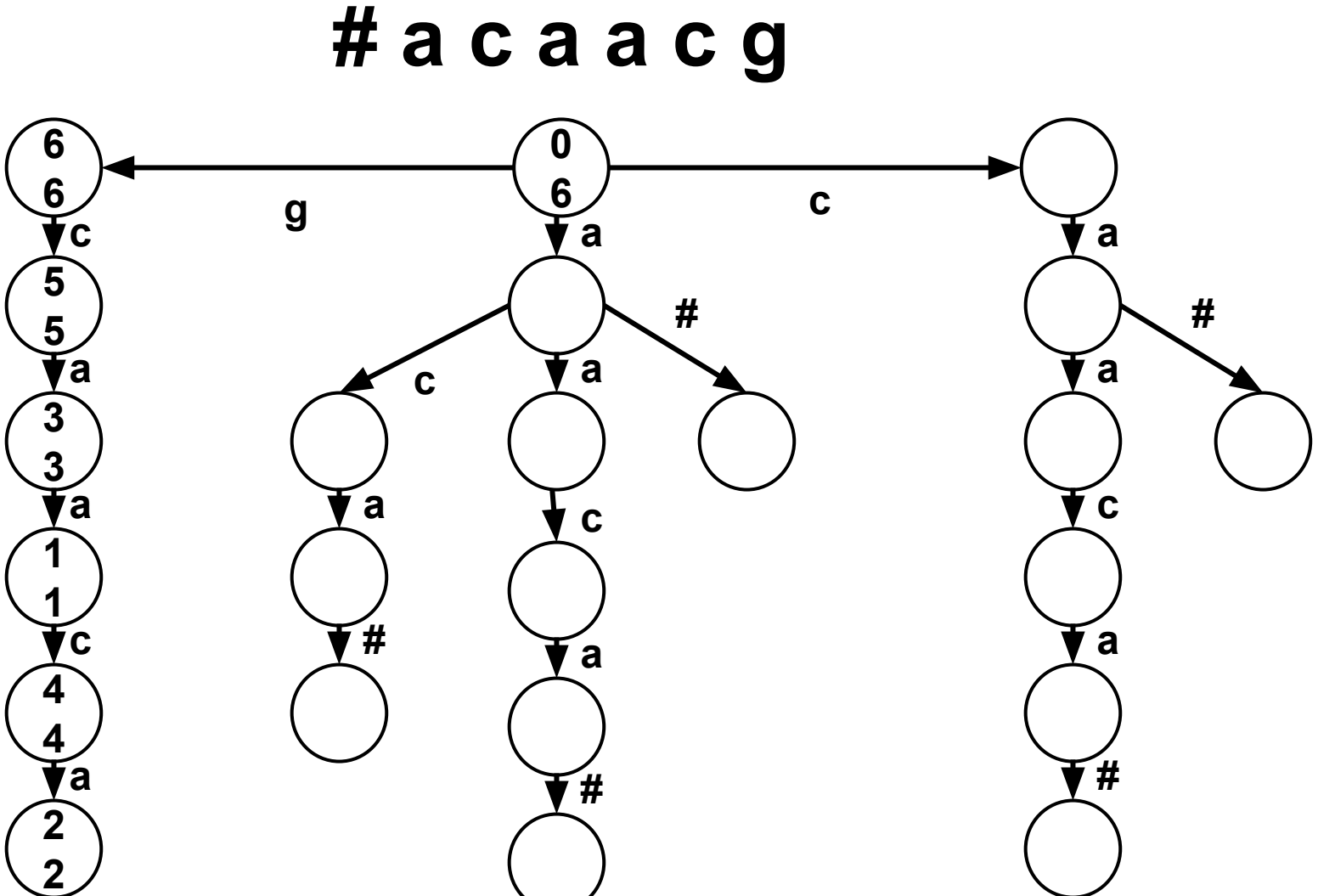
c g $

g $

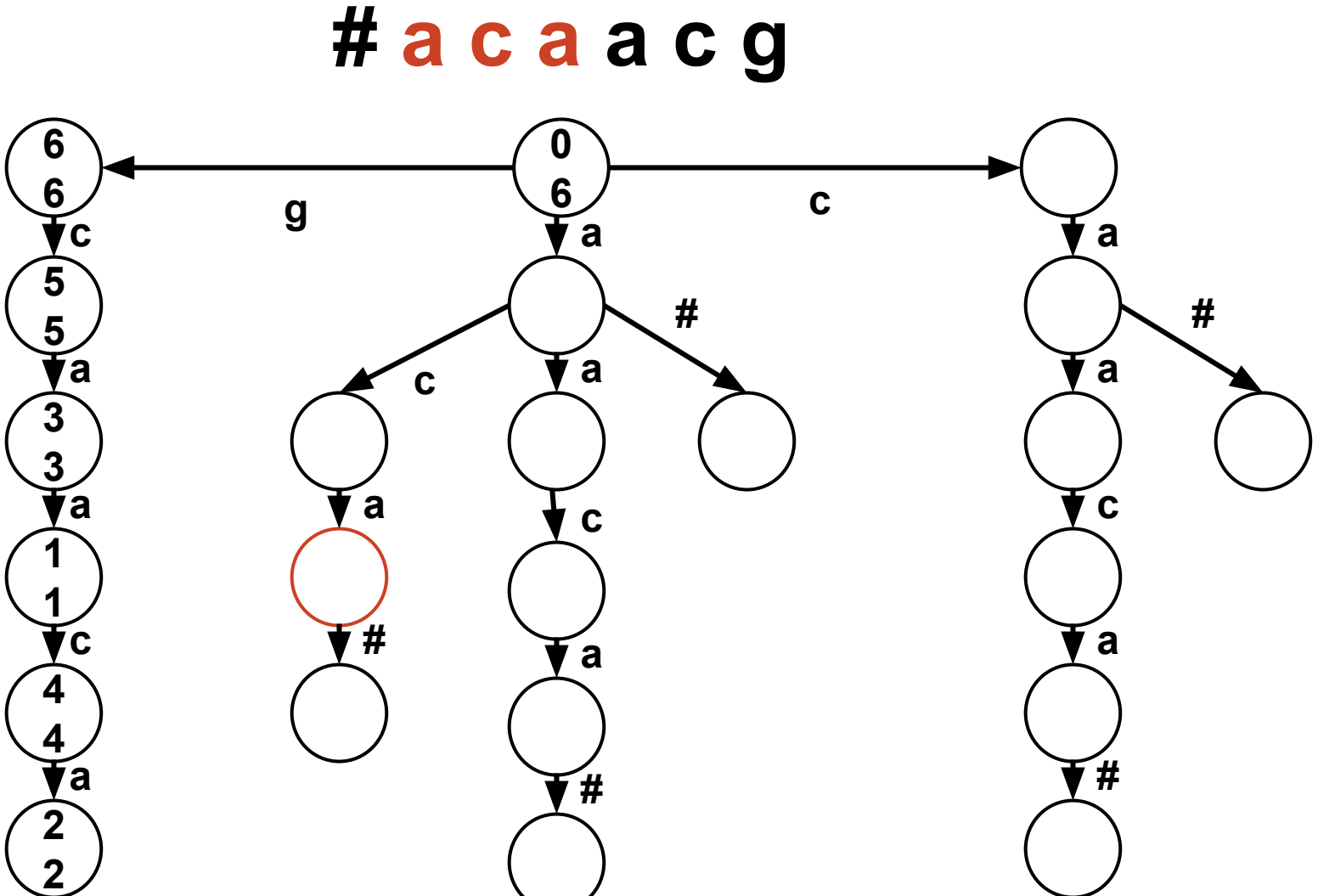
Prefix trie



Prefix trie



Prefix trie



Suffix array

$

a a c g $

a c a a c g $

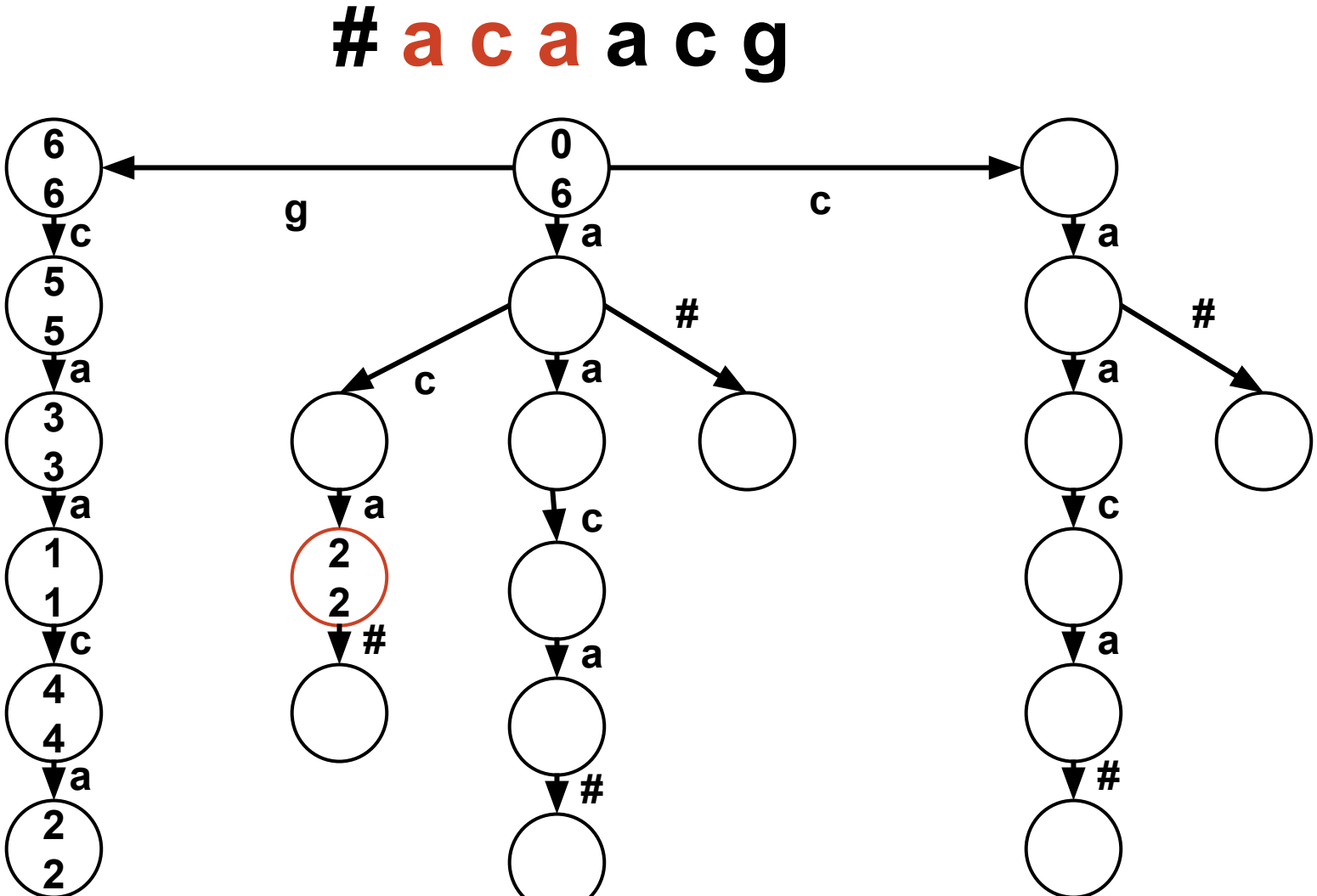
a c g $

c a a c g $

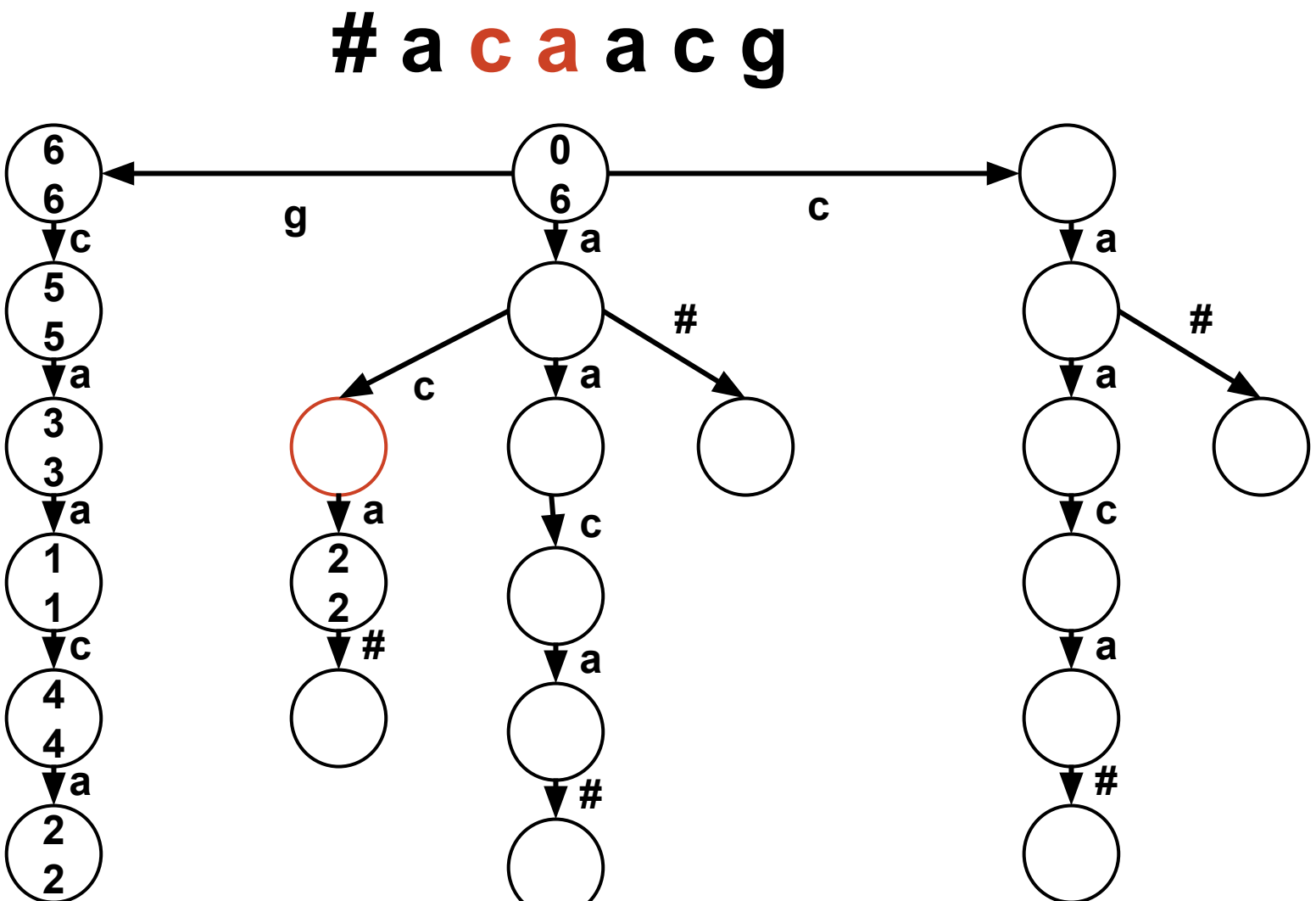
c g $

g $

Prefix trie



Prefix trie



Suffix array

$

a a c g $

a c a a c g $

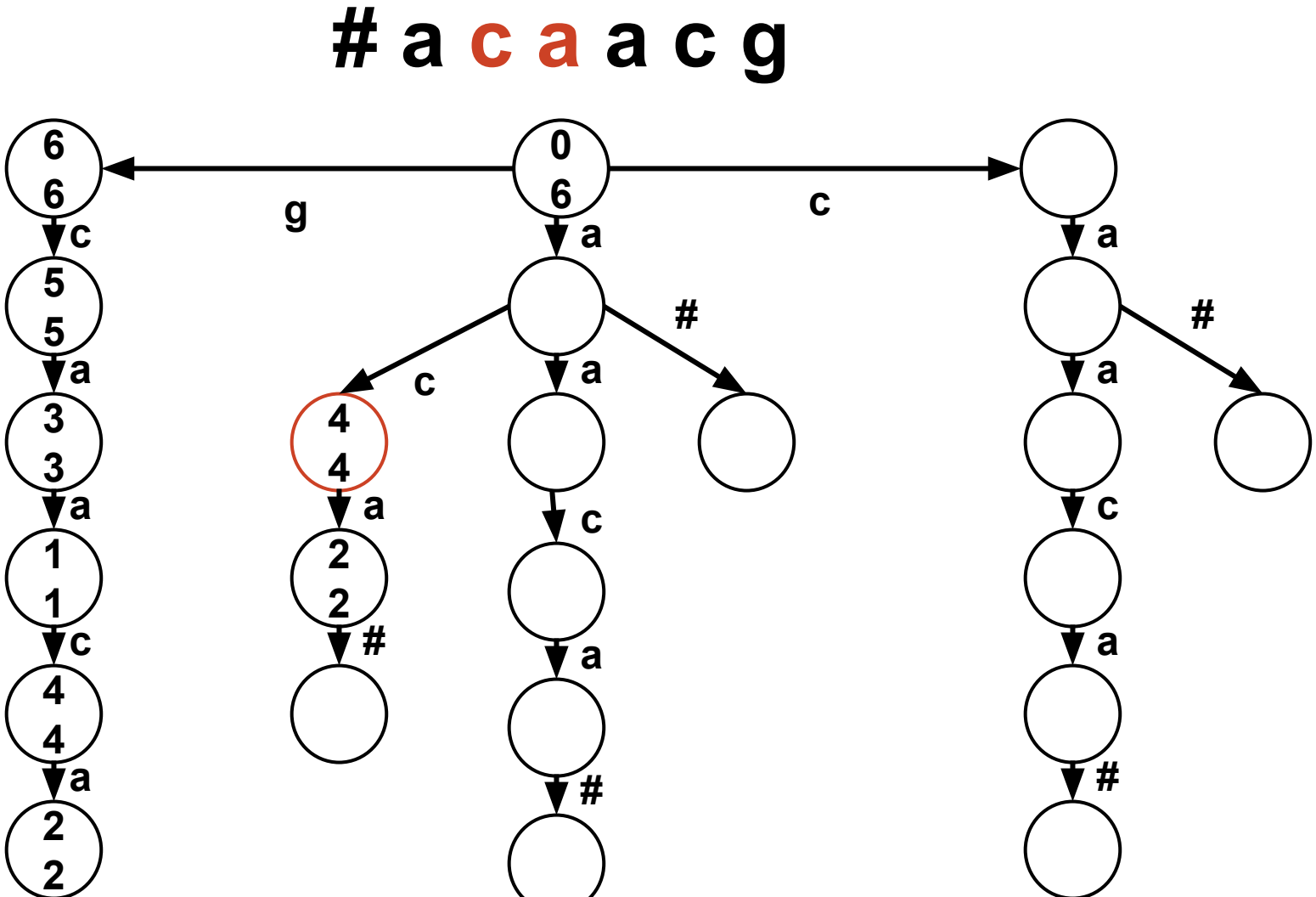
a c g $

c a a c g $

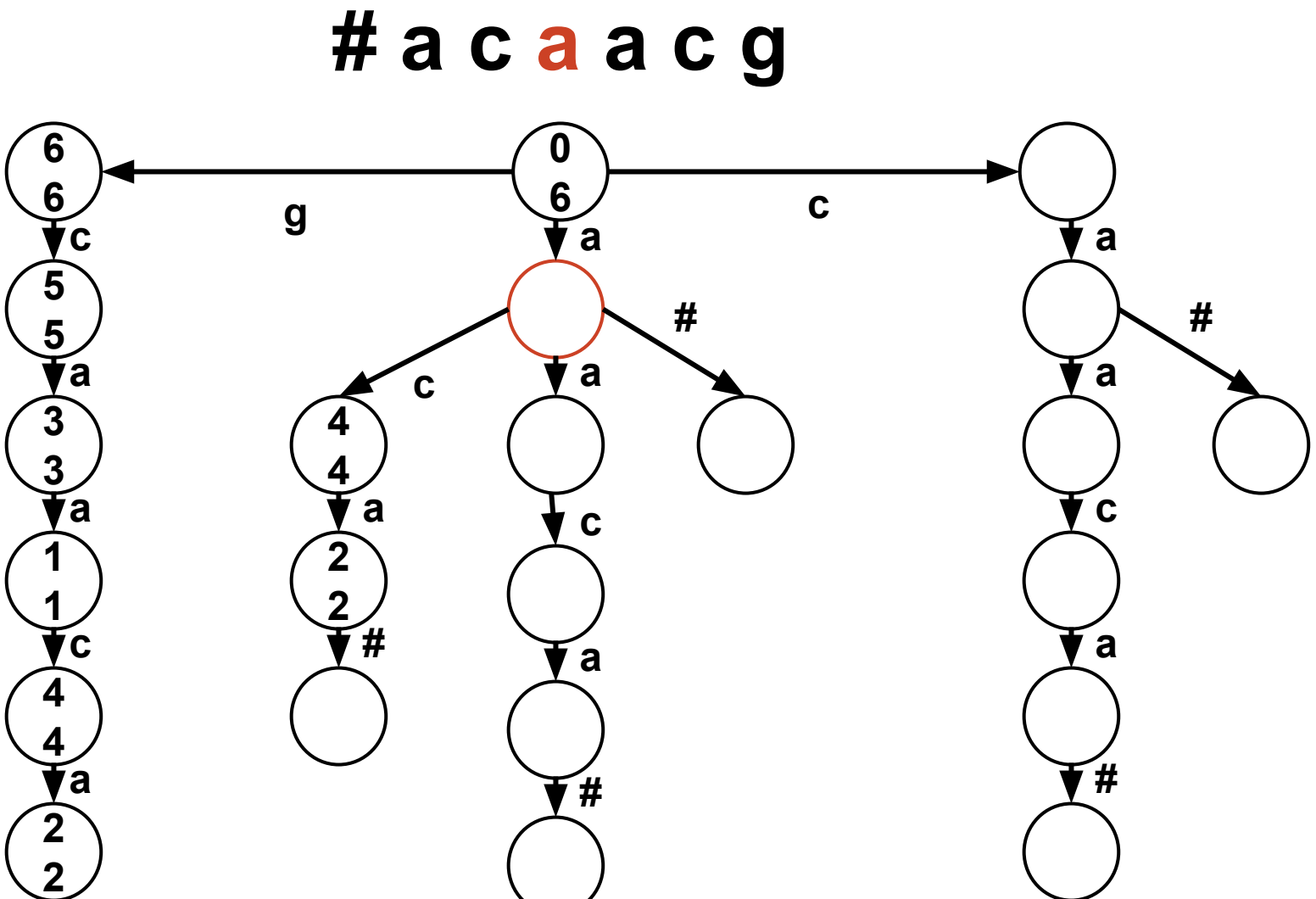
c g $

g $

Prefix trie



Prefix trie



Suffix array

$

a a c g $

a c a a c g $

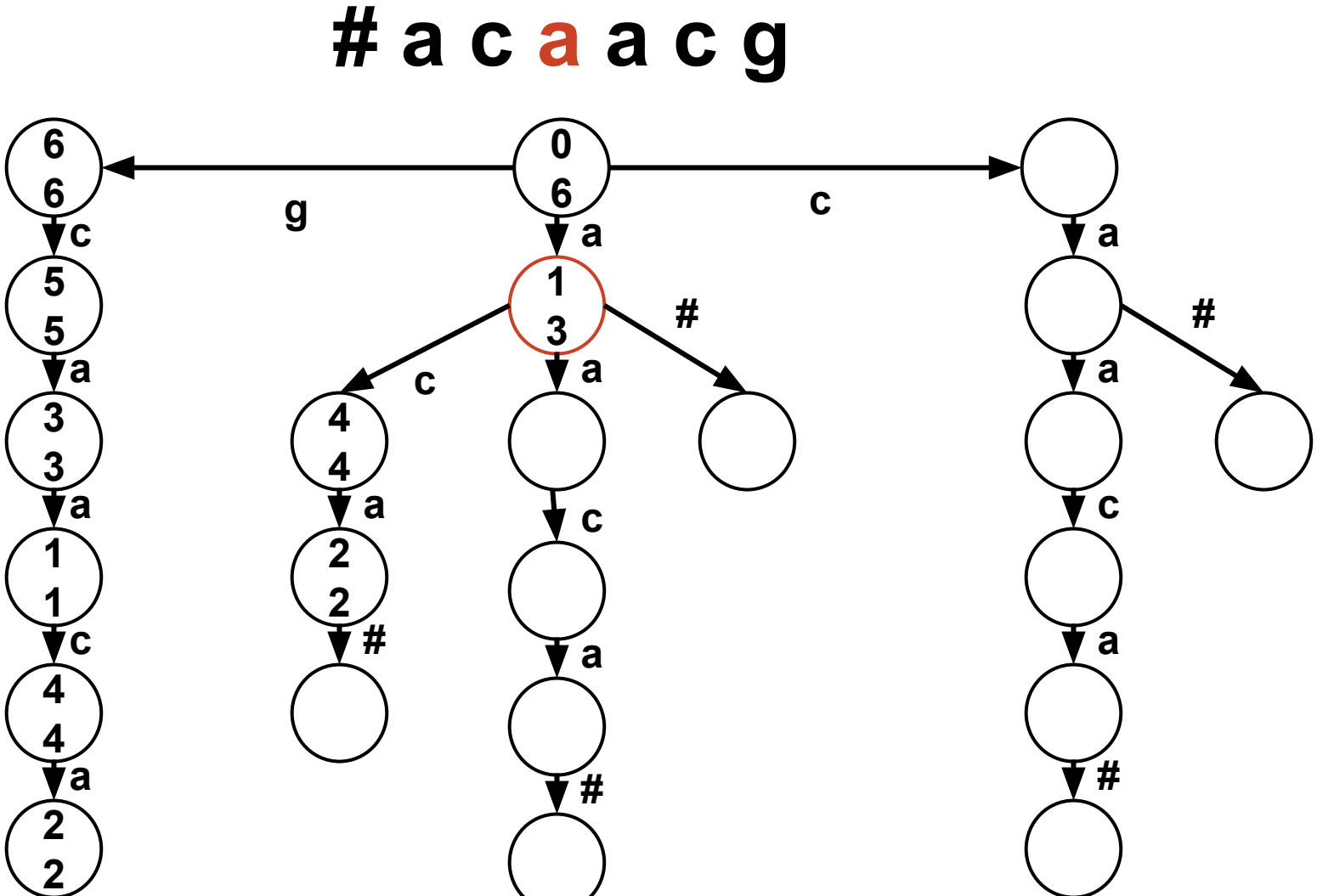
a c g $

c a a c g $

c g $

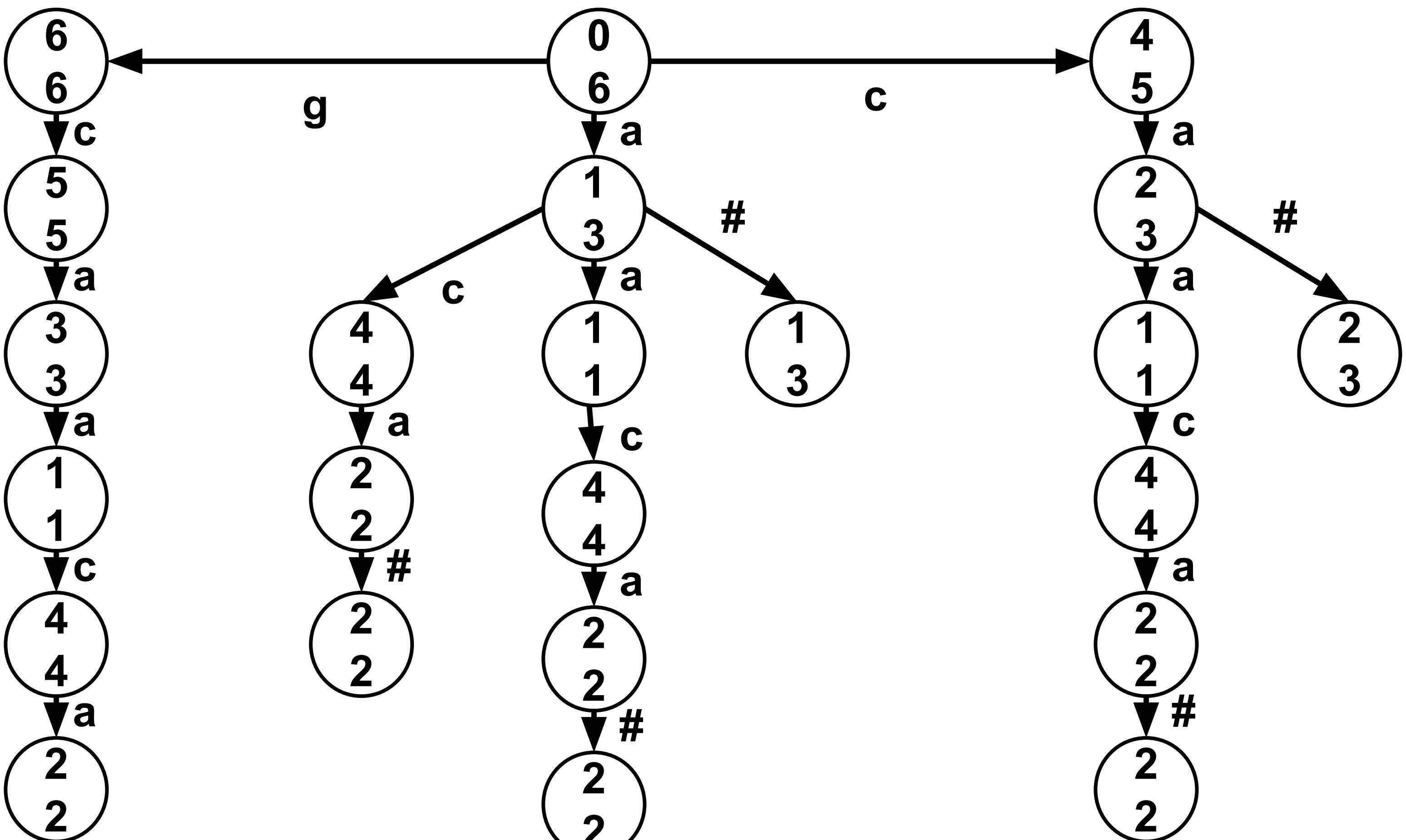
g $

Prefix trie

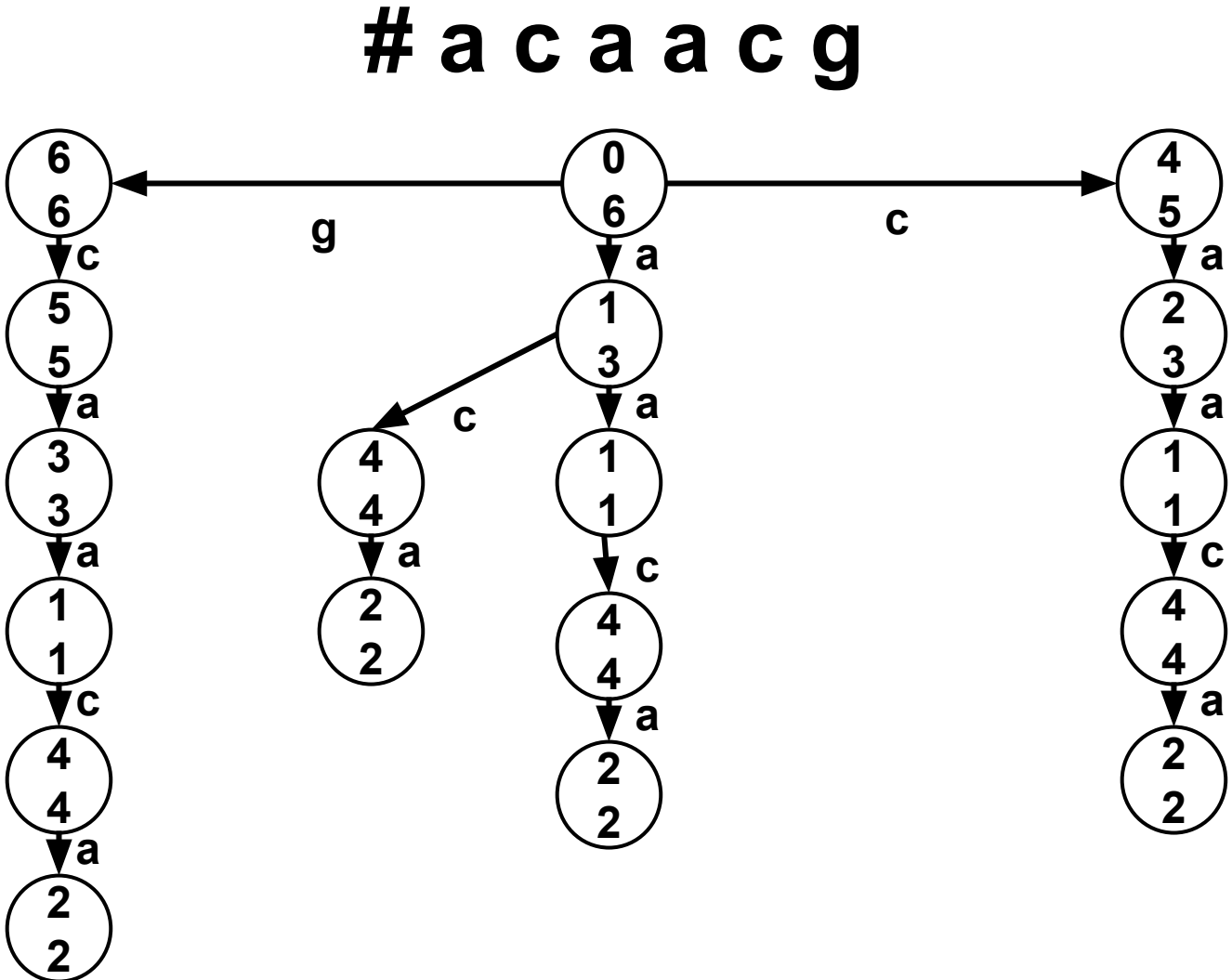


Prefix trie

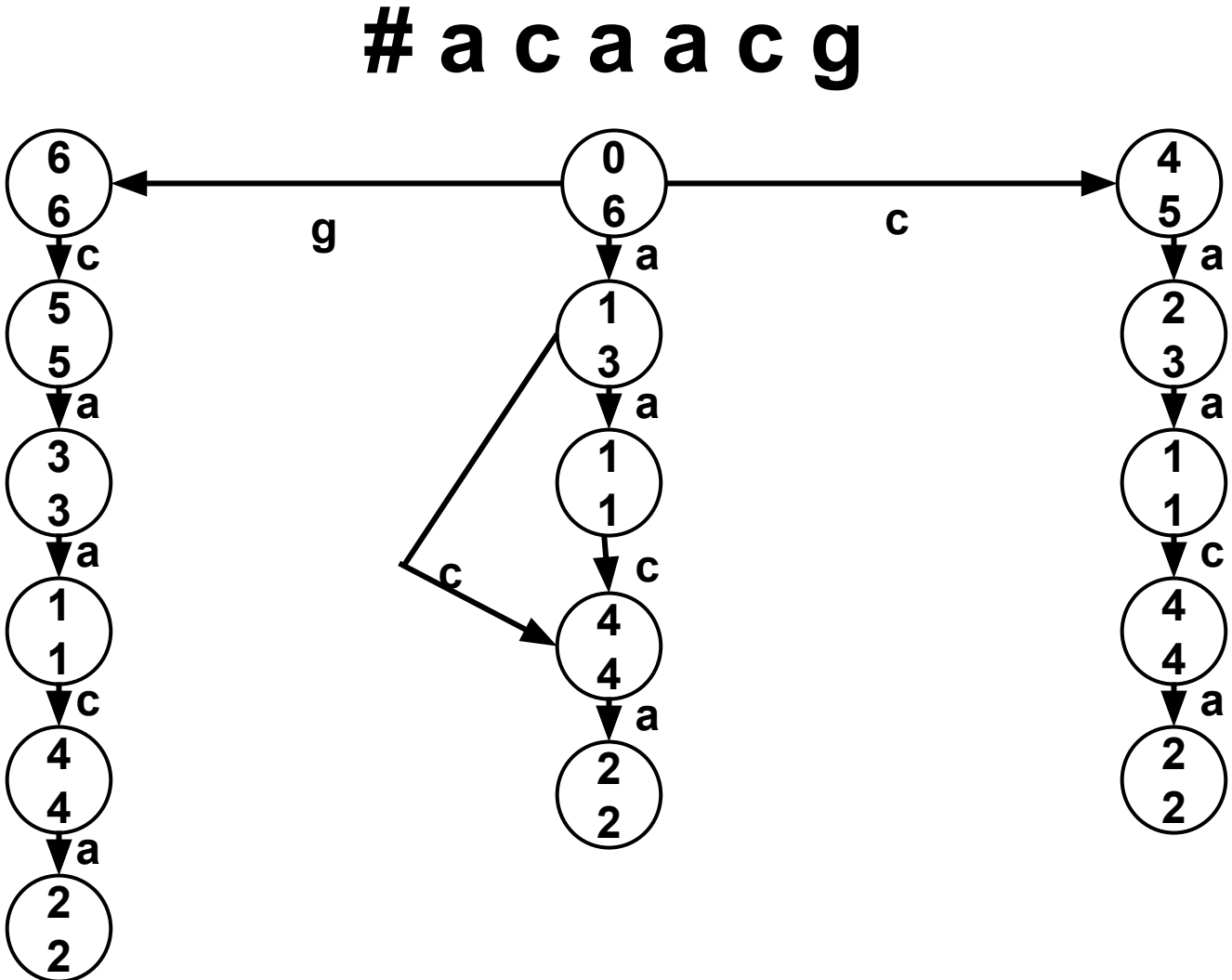
 # a c a a c g



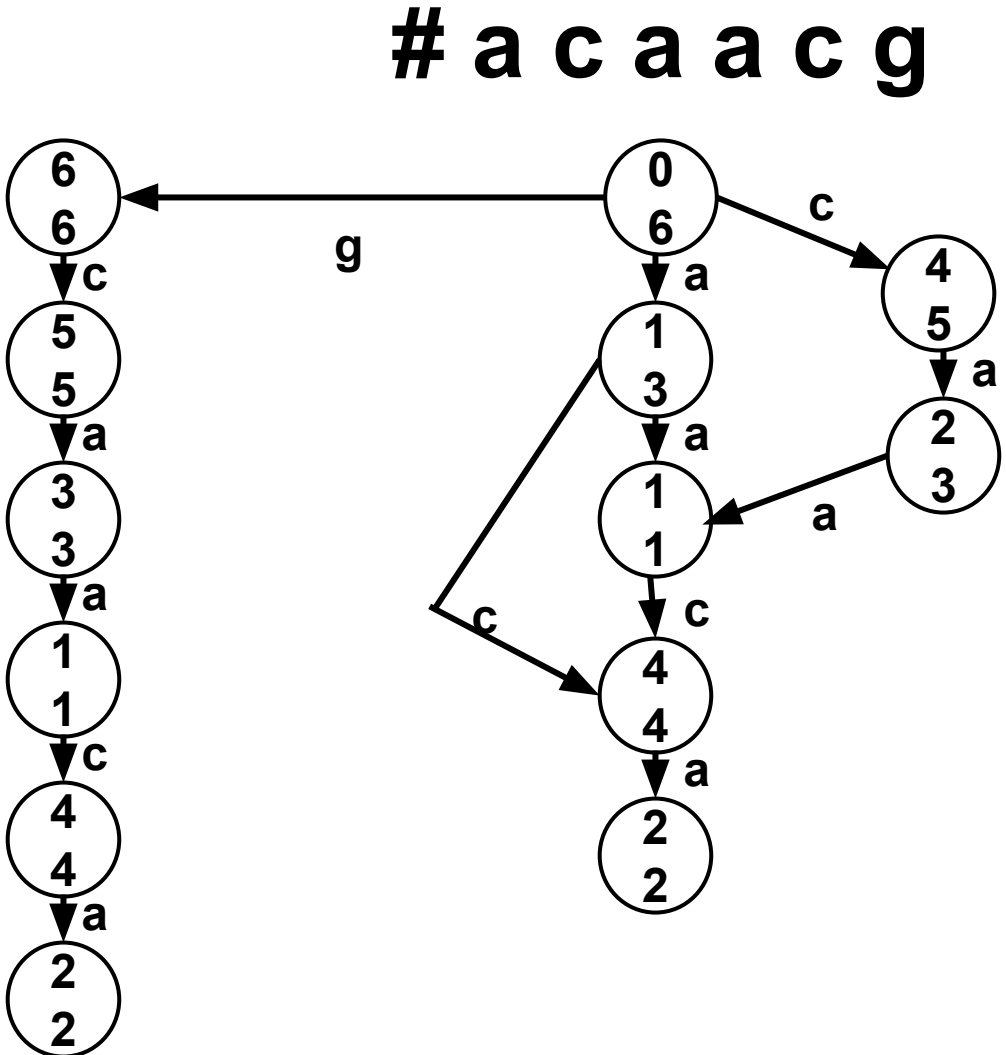
Prefix DAWG


a c a a c g
6 6
0 6
4 5
g
c
c
5 5
1 3
2 3
a
a
a
3 3
4 4
1 1
1 1
a
c
a
a
1 1
2 2
4 4
4 4
c
c
4 4
2 2
2 2
a
a
a
2 2
2 2
2 2

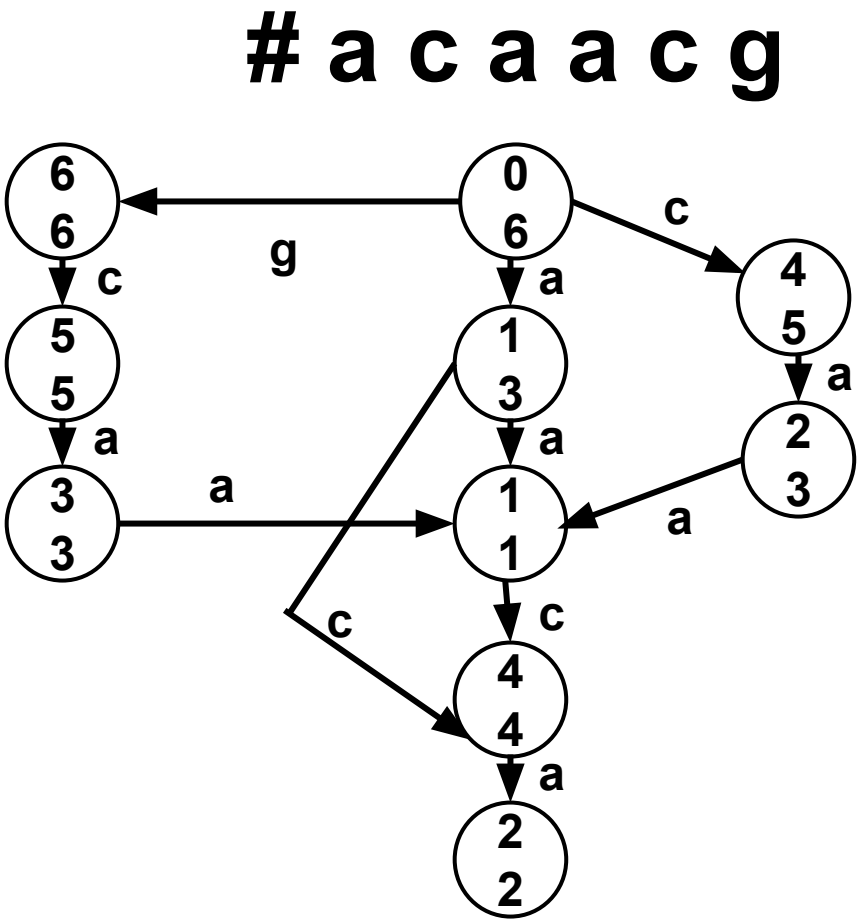
Prefix DAWG



Prefix DAWG



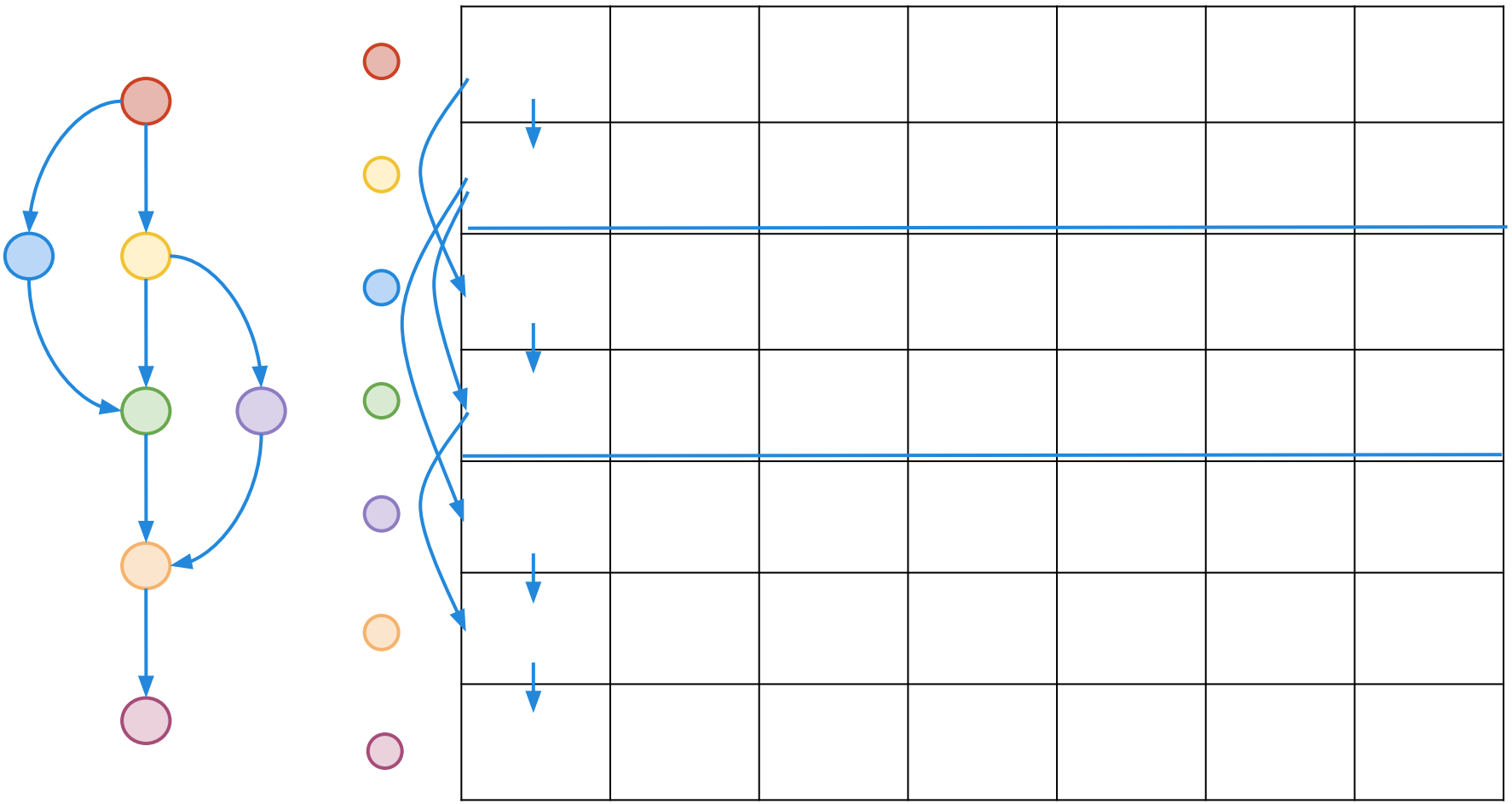
Prefix DAWG



BWT-SW

- Construct prefix trie T(X) from the reference
- Construct prefix DAWG Γ(W) from the query
- Calculate score between Γ(W) and T(X) using dynamic programming

BWT-SW



Alignment

- Substitution scoring matrix
  - $S(x, x) = 1$
  - $S(x, y) = -1, x \neq y$

- Affine gap model
  - Gap open score S_{op} ($S_{op} = 1$)
  - Gap extension score S_{ex} ($S_{ex} = 1$)

Dynamic programming

$u \in \Gamma(W)$, u' is a parent of u

$v \in T(X)$, v' is a parent of v

- $I_{uv \mid u'} = \max \{ I_{u' v} - S_{ex}, G_{u' v} - S_{op} \}$
- $D_{uv \mid u'} = \max \{ D_{uv'} - S_{ex}, G_{uv'} - S_{op} \}$
- $G_{uv \mid u'} = \max \{ G_{u'v'} + S(uu', vv'), I_{uv}, D_{uv} \}$

Dynamic programming

- Select maximal score $G_{uv \mid u^*}$ (among all parents of u)
- Set $(G_{uv}, I_{uv}, D_{uv}) = (G_{uv \mid u^*}, I_{uv \mid u^*}, D_{uv \mid u^*})$ if $G_{uv \mid u^*} > 0$
- Set $(G_{uv}, I_{uv}, D_{uv}) = (-\infty, -\infty, -\infty)$ otherwise

Dynamic programming

- Traverse both prefix trie and prefix DAWG from parents to children
- Stop if vertex has non-positive score
- Allows to traverse only a few vertices near the root
- Works much faster than usual Smith-Waterman: $O(n^{0.628} m)$

Acceleration by standard SW

- Once the vertex has
  - High score
  - Small interval
- Traversing becomes excessive
- Switch to usual Smith-Waterman

Z-best strategy

- Keeping all matching vertices for each node in DAWG requires time and memory
- Keep only Z best scoring vertices
- Possibility to miss some seeds (but we rely on other ones)
- Greatly increases performance

BWA tool

- **bwa index** — index construction
- **bwa aln** — short read aligner
- **bwa sw** — long read aligner
- **bwa mem** — new long read and long sequence local aligner

Alignment applications

Alignment applications

- Quality assessment
  - Error rate
  - Insert size distribution
  - Chimeric read/read-pairs
  - Genome fraction
- SNP calling
- Comparative analysis
  - CNVs
- Transciptomics
  - Gene expression
  - Exon/intron detection

Storing alignments

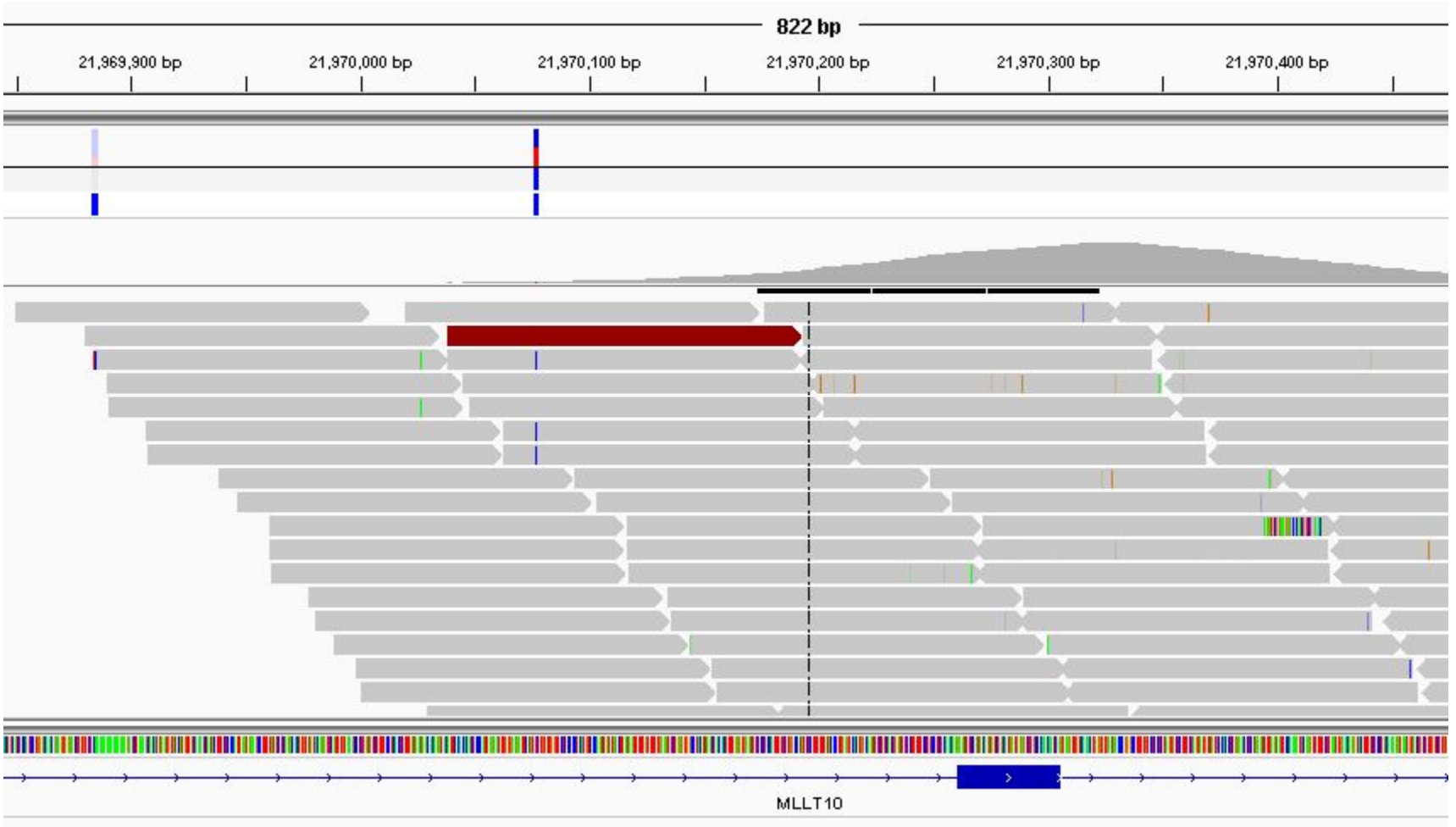
SAM/BAM files

- Read ID (QNAME)
- Reference ID (RNAME)
- Mapping position (POS)
- Mate reference ID (RNEXT)
- Mate position (PNEXT)
- Observed insert length (TLEN)
- Read sequence (SEQ)
- Read quality (QUAL)
- CIGAR string
  - 34M 1I 4M 2D 1X 3M

SAM files

```
@HD     VN:1.0  SO:coordinate
@SQ     SN:chr20        LN:64444167
@PG     ID:TopHat       VN:2.0.14       CL:/srv/dna_tools/tophat/tophat -N 3 --read-edit-dist 5 --read-rea
lign-edit-dist 2 -i 50 -I 5000 --max-coverage-intron 5000 -M -o out /data/user446/mapping_tophat/index/chr
20 /data/user446/mapping_tophat/L6_18_GTGAAA_L007_R1_001.fastq
HWI-ST1145:74:C101DACXX:7:1102:4284:73714        16      chr20   190930  3       100M    *       0       0
       CCGTGTTTAAAGGTGGATGCGGTCACCTTCCCAGCTAGGCTTAGGGATTCTTAGTTGGCCTAGGAAATCCAGCTAGTCCTGTCTCTCAGTCCCCCCTCT
C    BBDCCDDCCDDDDCDDDDDDCDCCCDBC?DDDDDDDDDDDDDDDCCDCDDDDDDDDDDCCCCEDDDC?DDDDDDDDDDDDDDDDDDDDDBDHFFFFDC@@
    AS:i:-15        XM:i:3  XO:i:0  XG:i:0  MD:Z:55C20C13A9 NM:i:3  NH:i:2  CC:Z:=  CP:i:55352714   HI:i:0
HWI-ST1145:74:C101DACXX:7:1114:2759:41961        16      chr20   193953  50      100M    *       0       0
       TGCTGGATCATCTGGTTAGTGGCTTCTGACTCAGAGGACCTTCGTCCCCTGGGGCAGTGGACCTTCCAGTGATTCCCCTGACATAAGGGGCATGGACGA
G    DCDDDDEDDDDDDDCDDDDDDDCCCDDDCDDDDDEEC>DFFFEJJJJJIGJJJJIHGBHHGJIJJJJJJGJJJIJJJJJIHJJJJJJHHHHHFFFFFCCC
    AS:i:-16        XM:i:3  XO:i:0  XG:i:0  MD:Z:60G16T18T3 NM:i:3  NH:i:1
HWI-ST1145:74:C101DACXX:7:1204:14760:4030        16      chr20   270877  50      100M    *       0       0
       GGCTTTATTGGTAAAAAAGGAATAGCAGATTTAATCAGAAATTCCCACCTGGCCCAGCAGCACCAACCAGAAAGAAGGGAAGAAGACAGGAAAAAACCA
C    DDDDDDDDDCCDDDDDDDDDDEEEEEEEFFFEFFEGHHHHFGDJJIHJJIJIJJJIIIIGGFJJIHIIIIJJJJJJIGHHFAHGFHJHFGGHFFFDD@BB
    AS:i:-11        XM:i:2  XO:i:0  XG:i:0  MD:Z:0A85G13    NM:i:2  NH:i:1
HWI-ST1145:74:C101DACXX:7:1210:11167:8699        0       chr20   271218  50      50M4700N50M     *       0
       0       GTGGCTCTTCCACAGGAATGTTGAGGATGACATCCATGTCTGGGGTGCACTTGGGTCTCCGAAGCAGAACATCCTCAAATATGACCTCTCG
accepted_hits.sam
```

Alignment visualization with IGV



SNP calling

Process of finding bases in the NGS data that differ from the reference genome

- Typically including an associated statistical confidence score
- Also known as “variant calling”

We need enough coverage to distinguish real variants from sequencing errors

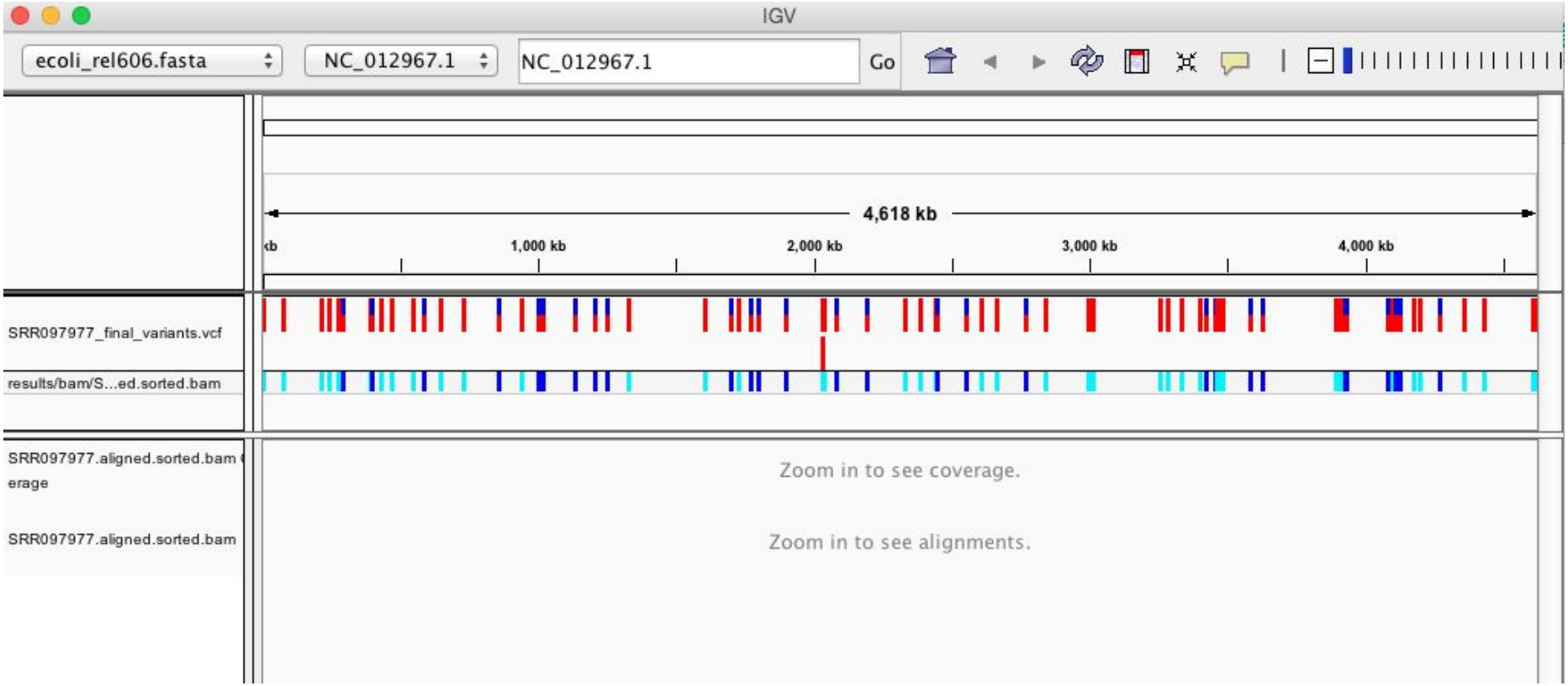
VCF/BCF files

- Chromosome (#CHROM)
- Position (POS)
- Unique identifiers where available (ID)
- Reference base(s) (REF)
- Alternate non-reference alleles (ALT)
- Phred quality score for the variant (QUAL)
- Optional filters (FILTER)
- Additional information (INFO)

VCF files

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO
1	10177	.	A	AC	100	PASS	AC=2130;AF=0.425319;AN=5008;NS=2504
1	10235	.	T	TA	100	PASS	AC=6;AF=0.00119808;AN=5008;NS=2504
1	10352	rs145072688	T	TA	100	PASS	AC=2191;AF=0.4375;AN=5008;NS=2504
1	10505	.	A	T	100	PASS	AC=1;AF=0.000199681;AN=5008;NS=2504
1	10506	.	C	G	100	PASS	AC=1;AF=0.000199681;AN=5008;NS=2504
1	10511	.	G	A	100	PASS	AC=1;AF=0.000199681;AN=5008;NS=2504
1	10539	.	C	A	100	PASS	AC=3;AF=0.000599042;AN=5008;NS=2504
1	10542	.	C	T	100	PASS	AC=1;AF=0.000199681;AN=5008;NS=2504
1	10579	.	C	A	100	PASS	AC=1;AF=0.000199681;AN=5008;NS=2504
1	10616	rs376342519	CCGCCGTTGCAAAGGCGCGCCG	C	100	PASS	AC=4973;AF=0.993011;AN=5008;NS=2504
1	10642	.	G	A	100	PASS	AC=21;AF=0.00419329;AN=5008;NS=2504
1	11008	.	C	G	100	PASS	AC=441;AF=0.0880591;AN=5008;NS=2504
1	11012	.	C	G	100	PASS	AC=441;AF=0.0880591;AN=5008;NS=2504
1	11063	.	T	G	100	PASS	AC=15;AF=0.00299521;AN=5008;NS=2504
1	13011	.	T	G	100	PASS	AC=3;AF=0.000599042;AN=5008;NS=2504
1	13110	.	G	A	100	PASS	AC=134;AF=0.0267572;AN=5008;NS=2504

Alignment visualization with IGV: one sample



Tools

- Alignment and data processing
  - samtools
  - bcftools
- SNP calling and annotation
  - VarScan
  - SnpEff
- Visualization
  - Tablet
  - IGV
- Pipelines
  - GATK

Thank you!

Questions?