Laporan Komputasi Genomik: Pemanfaatan Hidden Markov Model untuk Segmentasi Daerah Koding dan Non Koding serta Kaya AT- GC

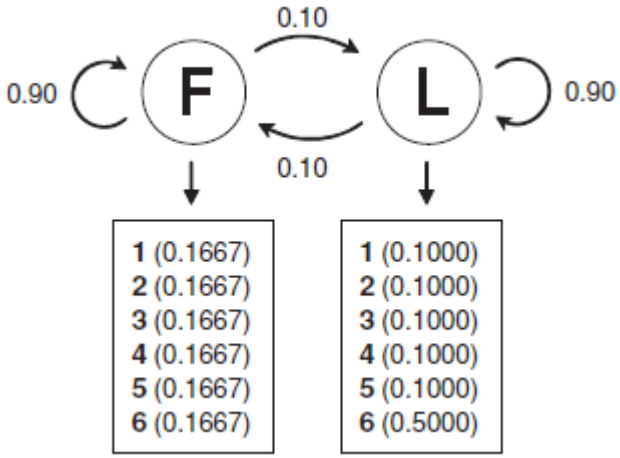
Dosen Pengajar :
Dr. Achmad Arifin, S.T., M.Eng.

Disusun oleh :
I Gede Eka Sulistyawan 07311640000002

Departemen Teknik Biomedik
Fakultas Teknologi Elektro
InstitutTeknologi Sepuluh Nopember Surabaya
2019

BAB I. Pendahuluan

1.1 Pemodelan *Hidden Markov*



Gambar 1. *Markov Chain*. *(Introduction to computational genomics: a case studies approach. Nello C., M. W. Hahn. 2006)*

Pemodelan Hidden Markov (HMM) merupaka jenis pemodelan yang memanfaatkan statistika untuk membagi suatu kondisi menjadi kondisi yang lebih kecil. Pada Bioinfomatika, pemodelan HMM banyak digunakan untuk segmentasi GC-Content, *alignment* dan pencarian DNA. Ide dasar dari pemanfaatan HMM adalah menyusun untai DNA yang paling mirip dengan suatu profil *markov chain*. *Markov Chain* adalah model probabilistik yang menyatakan suatu urutan data sebagai pergantian *state*. Contoh wujud dari model *markov chain* dapat dilihat pada gambar 2.

Gambar 2 menunjukkan *markov chain* dengan dua buah *state* F dan L yang menyatakan kemungkinan pelemparan dua buah dadu. Dadu pertama memiliki karakteristik seperti pada *state F* dan dadu kedua memiliki karakteristik *state* L. Karakteristik ini yang kemudian disebut sebagai *emission probability*. Didefinisikan pula kecenderungan pengguna mengganti pelemparan dadu dari dadu pertama ke dadu kedua dan sebaliknya atau menggunakan dadu tetap merupakan bagian dari *transition probability*. Pada gambar 2, *Emission Probability* masing-masing *state* diletakkan pada kotak dibawahnya, contoh pada *state F* angka 1 pada dadu muncul dengan kemungkinan 0.1667 sedangkan pada *state* L muncul dengan

kemungkinan 0.1 dan begitu seterusnya. Kondisi lain, *Transition Probability* pada gambar 2 melekat pada arah panah pergantian *state*. Tiga bagian penting dari sebuah *markov chain* sebagai model dasar telah disebutkan diatas, yakni *Emission Probability, Transition Probability*, dan *State*. Satu bagian penting lainnya adalah *observe* atau kumpulan data yang terobservasi. *Observe* biasanya merupakan bagian yang "tampak" seperti misalnya data hasil pelemparan dadu. Melalui data hasil pelemparan dadu (*observe*) dan model *markov chain*, dapat ditentukan dadu mana yang saat itu sedang digunakan. Misal suatu data *observe* hasil lemparan dadu

$$\boldsymbol{s} = 45536531633635551333626651321416366516 66$$

dari data *observe* lemparan dadu **s**, kemudian ditentukan kemungkinan *state* "tersembunyi" yang menyatakan dadu yang digunakan. Dengan memanfaatkan model *markov chain* seperti pada gambar 2, didapatkan komposisi *state* **h**.

$$\boldsymbol{h} = 11111111111111111111222211111112222222 22$$

Komputasi dapat dilakukan untuk estimasi *hidden state* berdasarkan suatu model *markov chain*. Algoritma yang dapat digunakan disebut dengan algoritma Viterbi. Masing-masing *Emission* maupun *Transition* keduanya tersusun atas matriks dengan dimensi sesuai dengan jumlah *state*. Matriks *Transition* (T) tersusun atas *N x N* probabilitas dengan N adalah banyak *state*. Matriks *Emission* (E) tersusun atas *N x M* probabilitas dengan M adalah banyak jenis data *observe*, empat (A, T, G, atau C) pada kasus DNA.

$$T(k,l) = P(h_i = l | h_{i-1} = k)$$
$$E(k,b) = P(s_i = b | h_i = k)$$

Berdasarkan dua persamaan diatas maka, matriks T berisikan probabilitas *state l* saat ini jika diketahui *state* sebelumnya adalah *k*. Matriks E berisikan probabilitas munculnya *observe b* jika diketahui sekarang berada pada *state k*. Kemungkinan keluarnya *hidden h*, dengan kondisi *observe s* ditentukan berdasarkan persamaan dibawah, dengan H adalah banyak *hidden state*. Pemecahan masalah adalah mencari sekuens data yang paling sesuai dengan model *markov chain*.

$$P(s_i, h_i) = \arg\max_{n \in H} \big(P(s_{i-1}, h_n) * E(s_i, h_n) * T(h_i, h_n) \big)$$

Persamaan diatas diselesaikan dengan pemrograman dinamik yang disebut dengan algoritma Viterbi.

1.6.1 *GC Content*

Pada ilmu biomolekuler, DNA dapat dianalisa berdasarkan ikatan nukelotida bersebelahan yang dibentuknya. Densitas GC pada DNA dapat memengaruhi termostabilitas, pelengkungan, transisi B-Z dan kurva DNA (Alexander, 2003). Analisa demikian dipengaruhi oleh ikatan terbentuk antara Guanin dan Sitosin terbukti secara termodinamik memiliki titik denaturasi kalor yang lebih tinggi, artinya diantara ikatan nukleotida bersebelahan lainnya GC/CG memiliki stabilitas termodinamik yang lebih tinggi (Santalucia, 1998).

Pertimbangan konten GC memunculkan segmentasi GC pada DNA sebagai salah satu permasalahan ilmu bioinformatik. Segmentasi yang umum dilakukan adalah membagi satu untai DNA ke dalam dua kondisi *GC-Rich* atau *AT-Rich*. Dua pertimbangan ini didasarkan pada karakteristik termodinamik ikatan DNA.

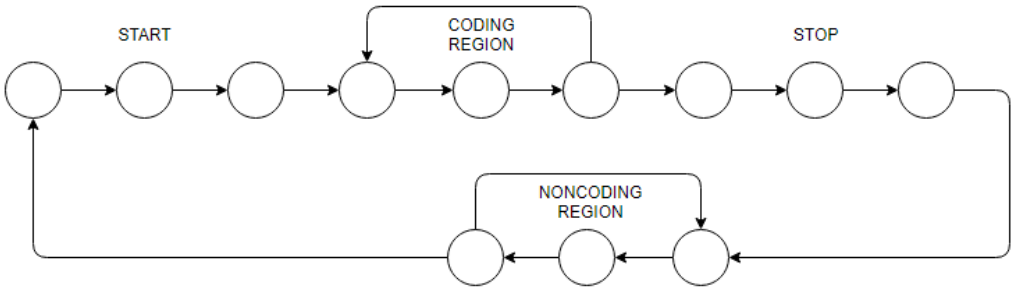
BAB II. Hasil dan Pembahasan

2.1 Model Hidden Markov

Pemodelan Hidden Markov digunakan untuk menganalisa kemungkinan ORF dan segmentasi AT-GC Rich berdasarkan pertimbangan. Langkah pertama dalam analisa HMM sebagai segmentasi adalah dengan menyusun *markov chain model*. Langkah selanjutnya adalah menentukan isi matriks seluruh kemungkinan *path Hidden*. Terakhir adalah memanfaatkan Algoritma Viterbi untuk mencari *most likely hidden sequences*.

2.2 HMM untuk Segmentasi *Coding-NonCoding Region*

Model Hidden Markov dapat digunakan untuk melakukan segmentasi *coding-noncoding region*. Mekanisme segmentasi mirip seperti pencarian ORF secara analitik, dimana daerah ORF diawali promotor ATG, kemudian dilanjutkan dengan *coding region* dan diakhiri oleh terminator TAA/TAG/TGA. Setelah terjadi



Gambar 2. *Markov chain Coding Non-Coding region*.

terminasi dilanjutkan dengan daerah *non-coding*. Secara lengkap, *markov chain* yang dihasilkan seperti gambar 2.

2.2.1 Promotor ATG

Promotor ATG pada pemodelan terletak pada *state* 1 sampai 3. Promotor ATG terdiri dari tiga basa dengan komposisi tetap ATG, maka pada pemodelan *hidden markov* digunakan emisi dan transisi (baris sebagai *state* sebelumnya) sebagai berikut

	State 1	State 2	State 3
A	1	0	0
T	0	1	0
G	0	0	1
C	0	0	0

	State1	State2	State 3
State 1	0	1	0
State 2	0	0	1
State 3	0	0	0

Setelah menemukan promotor, maka model senantiasa akan menuju *coding region*. Akibatnya, transisi dari *state* 3 ke *state* 4 yang merupakan transisi dari promotor ke *coding region* bernilai 1.

2.2.2 *Coding Region*

Coding region sesuai dengan yang telah dipelajari pada ilmu biomolekuler, selalu terdiri atas tiga basa atau disebut sebagai kodon. *Coding region* selalu memiliki tiga basa sehingga diperlukan tiga buah *state* untuk merepresentasikan *coding region*, diletakkan pada *State* 4 hingga 6. Masing-masing basa pada *coding region* memiliki kesempatan yang sama untuk muncul pada posisi yang sama, sehingga keseluruhan emisi untuk basa A T G C pada *coding region* memiliki probabilitas 0.25. Transisi pada *coding region* bernilai 1 untuk transisi dari *state* 4

ke 5 dan 5 ke 6 karena pasti terjadi. Setelah *coding region* terdapat kemungkinan *state* akan menuju ke terminator atau tetap pada *coding region*, kemungkinan ini direpresentasikan oleh transisi *state* 6 kembali ke 4 atau *state* 6 ke 7.

	State 4	State 5	State 6
A	0.25	0.25	0.25
T	0.25	0.25	0.25
G	0.25	0.25	0.25
C	0.25	0.25	0.25

	State4	State5	State 6
State 4	0	1	0
State 5	0	0	1
State 6	1-T1	0	0

Jika dimisalkan T1 adalah probabilitas state 6 menuju state 7, maka probabilitas state 6 kembali ke state 4 adalah 1 – T1.

2.2.3 Terminator TAA/TAG/TGA

Teminator selalu terdiri dari tiga basa, sehingga diberikan tiga buah pada state 7 hingga 9. Masing-masing state terdiri dari emisi yang berbeda tergantung probabilitas kejadian basa pada terminator. Pada state 7 pasti terjadi T, pada state 8 dan 9 kemungkinan terjadi A adalah 2 dari 3 dan terjadi G adalah 1 dari 3. Transisi pada terminator tidak jauh berbeda dari transisi promotor karena memiliki sifat yang sama.

	State 7	State 8	State 9
A	0	0.67	0.67
T	1	0	0
G	0	0.33	0.33
C	0	0	0

	State7	State8	State 9
State 7	0	1	0
State 8	0	0	1
State 9	0	0	0

Transisi dari state 9 ke 10 atau yang merepresentasikan terminator ke *non coding region* pasti terjadi, sehingga nilai transisi 9 ke 10 adalah 1.

2.2.4 Non-Coding Region

Komposisi model *non-coding region* sama seperti *coding region*, tetapi terletak pada state 10 hingga 12. Masing-masing basa memiliki kesempatan yang sama untuk muncul. Transisi yang terjadi, terutama ketika *non-coding region* cenderung terjadi, sangat memengaruhi probabilitas dari state 12 menemukan state 1 atau dari *noncoding region* menemukan promotor.

	State 10	State 11	State 12
A	0.25	0.25	0.25
T	0.25	0.25	0.25
G	0.25	0.25	0.25
C	0.25	0.25	0.25

	State10	State11	State12
State10	0	1	0
State11	0	0	1
State12	1-T2	0	0

Diperkenalkan transisi T2 yang merepresentasikan transisi dari state 12 ke state 1, sehiingga kecenderungan basa berada pada *non-coding* region adalah 1-T2.

2.2.5 Komputasi

Langkah pertama untuk menerapkan model HMM pada segmentasi *coding non-coding region* adalah dengan melakukan *forwarding*. Algoritma ini merupakan algoritma yang menghitung probabilitas kejadian *P* berdasarkan kejadian *P* sebelumnya. Algoritma *forward* menghitung nilai tertinggi yang mungkin dicapai berdasarkan satu kejadian unik sebelumnya. Penerapan perhitungan bersifat rekursif, sehingga untuk mengurangi *error* diperlukan transformasi logaritmik log10 untuk merepresentasikan kemungkinan.

2.2.6 Inisialisasi

Tahap awal komputasi memerlukan inisialisasi berupa kemungkinan kejadian pertama. Pada tahap ini didefinisikan basa pertama hanya mungkin dimulai dari promotor atau *non-coding region*, sehingga probabilitas yang terjadi untuk promotor dan *non-coding region* masing-masing adalah 0.5. Simulasi segmentasi diterapkan pada DNA *E. Phage Lambda*. Nilai T1 dan T2 diberikan 0.6. Perhitungan algoritma *forward* dan *viterbi* masing-masing dijelaskan pada halaman berikutnya.

2.2.7 *Forwarding*

S\O	Init	*max*	C	*max*	C
1	*log (0.5)*		-18.2218		-18.8239
2	*log (10E-8)*		-10.3010		-20.9031
3	-8		-18.0000		-20.3010
4	-8		-8.6021		-9.6021
5	-8		-8.6021		-9.2041
6	-8		-8.6021		-9.2041
7	-8		-18.2218		-18.8239
8	-8		-18.0000		-20.9031
9	-8		-18.0000		-20.9031
10	*log10(0.5)*		-8.6021		-9.6021
11	-8		-0.9031		-9.2041
12	-8		-8.6021		-1.5051

Perhitungan diatas pada *observable* pertama

$$P_1(C|S_i) = \max_{n \in state} T(S_i|S_{i-1}) E(C|S_i) P_0(O_0|S_i)$$

Contoh pada *obersvable* pertama untuk *state 1*, probabilitas tertinggi akan terjadi pada state 1 jika datang dari state 12, maka

$$P_1(C|S_1) = T(S_1|S_{12}) + E(C|S_1) + (-8)$$
$$P_1(C|S_1) = \log(0.6) + \log(0) - 8$$
$$P_1(C|S_1) = -0.2218 - 10 - 8$$
$$P_1(C|S_1) = -18.2218$$

Pada kondisi tersebut, log (0) merupakan nilai yang sangat kecil (10E-10). Agoritma *forward* terjadi sebanyak total basepair. Selain menyimpan probabilitas kejadian *obervable* pada sekuens untuk satu *state* tertentu, komputasi dilakukan paralel dengan sekaligus menyimpan *state* sebelumnya yang menyebabkan *observable* maksimum (*arg max).* Data ini disimpan pada *array* PATH.

PATH akan digunakan untuk algoritma Viterbi, yakni penentuan *most likely hidden sequence* untuk kasus *coding* dan *noncoding region*.

2.2.8 Viterbi

Algoritma Viterbi dijalankan dari basa terakhir. Pada basa terakhir, dideteksi probabilitas *hidden state* paling tinggi diantara lainnya. Kemudian berdasarkan PATH pada *hidden state* basa terakhir tersebut akan menuju *hidden state* pada basa sebelumnya. Contoh dibawah adalah pelaksanaan algoritma Viterbi pada tiga basa terakhir.

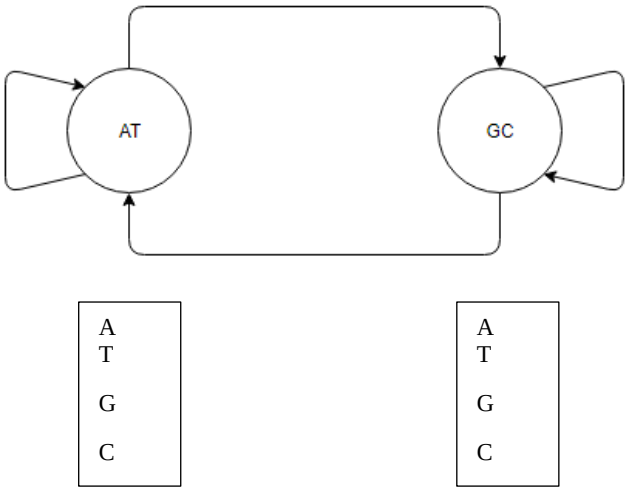
S\O	PATH			*P basa terakhir*
1	12	12	12	-35142.3090
2	1	7	8	-35147.2270
3	6	2	8	-35147.2270
4	6	6	3	-35130.7070
5	4	4	4	-35135.3290
6	5	5	5	-35128.7280
7	6	6	6	-35144.3290
8	6	7	8	-35147.2270
9	8	7	8	-35137.2270
10	12	12	12	-35133.0870
11	10	10	10	-35133.3090
12	11	11	11	-35131.1080

2.2.9 Hasil

Berdasarkan uji diatas, dilakukan pembandingan dengan pencarian manual promotor ATG, *coding region*, terminator, dan *non-coding region*. Hasil yang didapatkan adalah HMM dengan T1 dan T2 sebesar 0.6 memiliki tingkat kesamaan 50.18% dengan pencarian manual. Pemilihan T1 dan T2 yang baik masih menjadi topik pengembangan karena memerlukan pengtahuan lebih lanjut terkait kecenderungan *coding region* untuk mengakhiri ORF serta kecenderungan *non-coding region* untuk menemui promotor.

Nilai T1 berhubungan dengan panjang *coding region*, jika nilai T1 besar maka *coding region* akan semakin pendek karena model akan cenderung mengakhiri *coding region*. Nilai T2 berhubungan dengan jumlah ORF

2.3 HMM untuk Segmentasi AT-GC Rich



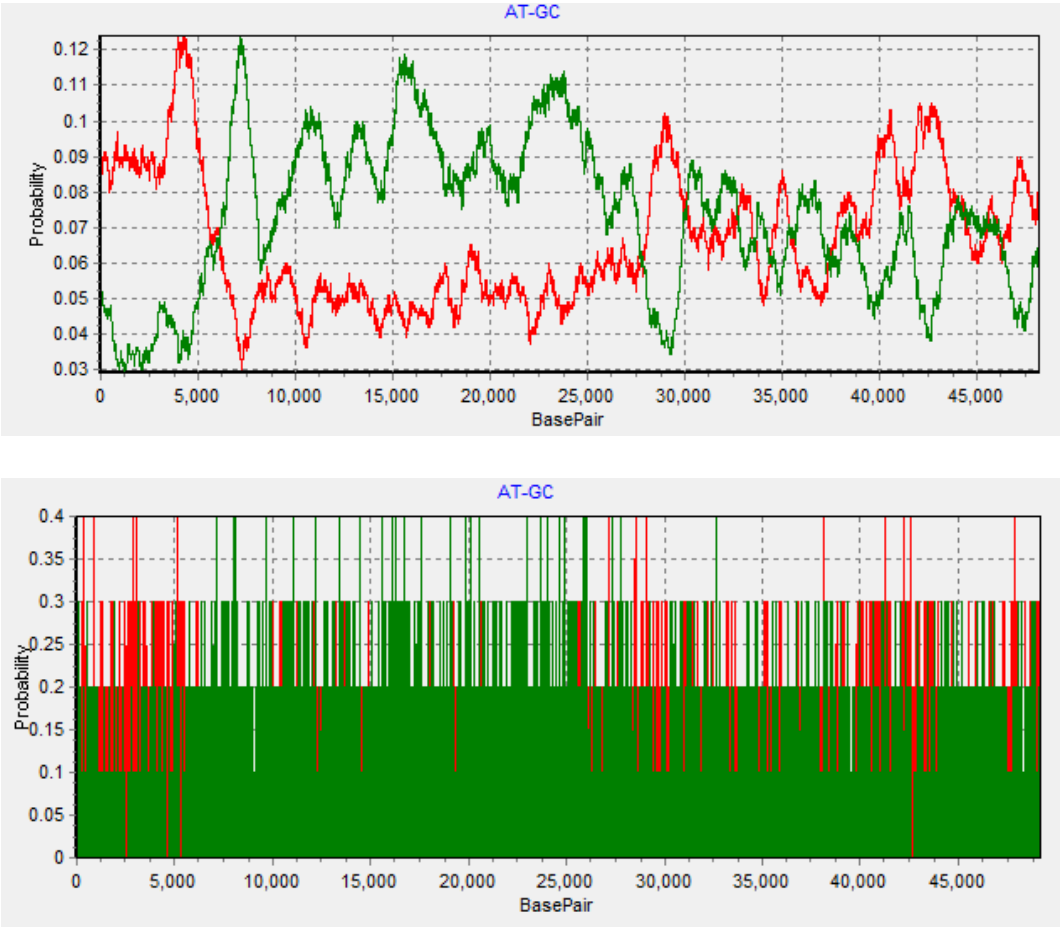
Gambar 3. *Markov chain* Kaya AT- GC.

Segmentasi *AT-GC rich* ditentukan dengan memanfaatkan model *markov* dengan dua *state* dan empat probabilitas emisi (A, T, G, C). Model *markov* ditampilkan seperti pada gambar 5. Terdapat empat transisi, dan empat probabilitas emisi. Pemberian nilai pada probabilitas emisi dan transmisi didapatkan dari *moving window* pada analisa statistik.

Estimasi transisi dan emisi dari *markov chain* untuk model diatas ditentukan berdasarkan pengamatan pada pergantian Kaya AT-GC analisa statistik dengan pergeseran *window*. Pemilihan *window* yang tepat untuk pengamatan dapat diamati berdasarkan panjang ORF yang menjadi ketertarikan dan lokasinya pada DNA. HMM digunakan untuk segmentasi kaya AT-GC yang terjadi pada ORF kemudian melakukan perbandingan dengan data yang didapat berdasarkan moving window.

2.3.1 *Windowing*

Pemodelan *markov chain* pada gambar 3 diparameterisasi dengan *windowing* pada keselurhan rentang DNA. Lebar *window* menjadi variabel yang harus divariasikan untuk mengamati pengaruh HMM terhadap perbaikan pelabelan. Variasi lebar *window* yang dilakukan sebanyak dua kali, pertama dengan lebar *window* 1000 dan kedua dengan lebar 10. Pemilihan panjang 10 karena ORF yang diujikan yakni ORF1 memiliki panjang ORF 15. Masing-masing *window* kemudian dihitung dengan memanfaatkan algoritma *forward* dan Viterbi.



Gambar 4. Densitas AT-GC dengan *window* 1000 (atas) dan 10 (bawah).

Berdasarkan pemodelan ini kemudian dilakukan *profiling* model atau pemberian parameter sesuai dengan observasi statistik *window*. Masing-masing *window* mengkarakterisasi profil sebagai berikut

Tabel 1. Profil HMM *window* 1000

	Kaya AT	Kaya GC
A	0.3011	0.2082
T	0.2763	0.2399
G	0.2105	0.2507
C	0.2121	0.3012

i-1\i	AT	GC
AT	0.9986	0.0014
GC	0.0009	0.9991

Tabel 2. Profil HMM dengan *window* 10

	Kaya AT	Kaya GC
A	0.3009	0.1982
T	0.2979	0.2146
G	0.1911	0.2735
C	0.2101	0.3137

i-1\i	AT	GC
AT	0.9256	0.0744
GC	0.0662	0.9338

pengujian segmentasi AT GC dilakukan pada ORF1 dengan panjang 15 bp.

$$O = ATGAGAAGCAGATAA$$

variabel yang akan digunakan dalam pengujian adalah

$P_{observable}^{state}$ yang menyatakan probabilitas *observable.*

$Path_{observable}^{state}$menyatakan *state* sebelumnya yang menyebabkan State sekarang maksimum.

$T(S1, S2)$ menyatakan probabilitas transisi S1 ke S2.

$E(O, S)$ menyatakan probabilitas emisi *O* ketika state S.

Inisialisasi dengan kemungkinan masing-masing state adalah 0.5 atau dalam fungsi logaritmik 10 adalah -0.3010. Seluruh perhitungan akan memanfaatkan penjumlahan logaritmik dibandingkan dengan perkalian probabilitas. Pada pelabela, label 1 menandakan kaya AT dan 2 untuk kaya GC.

2.3.2 Pengujian

Pengujian satu memanfaatkan profil yang dihasilkan oleh *windowing* dengn lebar 1000. Model pengujian dilakukan seperti langkah pada gambar 5.

- *Observable 1*

$$P_{O1=A}^{Kaya\ AT} = max\begin{pmatrix} P_{O0}^{AT} + log10\big(T(AT, AT)\big) + log10\big(E(A, AT)\big) \\ P_{O0}^{GC} + log10\big(T(GC, AT)\big) + log10\big(E(A, AT)\big) \end{pmatrix}$$

$$= \max\begin{pmatrix} -0.3010 + log10(0.9986) + log10(0.3011) \\ -0.3010 + log10(0.0009) + log10(0.3011) \end{pmatrix}$$

$$= \max\begin{pmatrix} -0.8829 \\ -3.8680 \end{pmatrix} = -0.8829$$

$$Path_{O1}^{AT} = \arg\max\begin{pmatrix}-0.8829\\-3.8680\end{pmatrix} = 1$$

$$P_{O1=A}^{Kaya\ GC} = max\begin{pmatrix}P_{O0}^{AT} + log10\big(T(AT,GC)\big) + log10\big(E(A,GC)\big)\\P_{O0}^{GC} + log10\big(T(GC,GC)\big) + log10\big(E(A,GC)\big)\end{pmatrix}$$

$$= \max\begin{pmatrix}-0.3010 + log10(0.0014) + log10(0.2082)\\-0.3010 + log10(0.9991) + log10(0.2082)\end{pmatrix}$$

$$= \max\begin{pmatrix}-3.8364\\-0.9829\end{pmatrix} = -0.9829$$

$$Path_{O1}^{GC} = \arg\max\begin{pmatrix}-0.8829\\-3.8680\end{pmatrix} = 2$$

- *Observable 2*

$$P_{O2=T}^{Kaya\ AT} = max\begin{pmatrix}P_{O1}^{AT} + log10\big(T(AT,AT)\big) + log10\big(E(T,AT)\big)\\P_{O1}^{GC} + log10\big(T(GC,AT)\big) + log10\big(E(T,AT)\big)\end{pmatrix}$$

$$= \max\begin{pmatrix}-0.8829 + log10(0.9986) + log10(0.2763)\\-0.9829 + log10(0.0009) + log10(0.2763)\end{pmatrix}$$

$$= \max\begin{pmatrix}-1.3821\\-4.5873\end{pmatrix} = -1.3821$$

$$Path_{O2}^{AT} = \arg\max\begin{pmatrix}-1.3821\\-4.5873\end{pmatrix} = 1$$

$$P_{O2=T}^{Kaya\ GC} = max\begin{pmatrix}P_{O1}^{AT} + log10\big(T(AT,GC)\big) + log10\big(E(T,GC)\big)\\P_{O1}^{GC} + log10\big(T(GC,GC)\big) + log10\big(E(T,GC)\big)\end{pmatrix}$$

$$= \max\begin{pmatrix}-0.8829 + log10(0.0014) + log10(0.2399)\\-0.9829 + log10(0.9991) + log10(0.2399)\end{pmatrix}$$

$$= \max\begin{pmatrix}-4.3567\\-1.6033\end{pmatrix} = -1.6033$$

$$Path_{O2}^{GC} = \arg\max\begin{pmatrix}-4.3567\\-1.6033\end{pmatrix} = 2$$

- *Observable 3*

$$P_{O3=G}^{Kaya\ AT} = max\begin{pmatrix}P_{O2}^{AT} + log10\big(T(AT,AT)\big) + log10\big(E(G,AT)\big)\\P_{O2}^{GC} + log10\big(T(GC,AT)\big) + log10\big(E(G,AT)\big)\end{pmatrix}$$

$$= \max\begin{pmatrix}-1.3821 + log10(0.9986) + log10(0.2105)\\-1.6033 + log10(0.0009) + log10(0.2105)\end{pmatrix}$$

$$= \max\begin{pmatrix}-2.0595\\-5.3258\end{pmatrix} = -2.0595$$

$$Path_{O3}^{AT} = \arg\max\begin{pmatrix}-2.0595\\-5.3258\end{pmatrix} = 1$$

$$P_{O3=G}^{Kaya\ GC} = max\begin{pmatrix} P_{O2}^{AT} + log10\big(T(AT,GC)\big) + log10\big(E(G,GC)\big) \\ P_{O2}^{GC} + log10\big(T(GC,GC)\big) + log10\big(E(G,GC)\big) \end{pmatrix}$$

$$= \max\begin{pmatrix} -1.3821 + log10(0.0014) + log10(0.2507) \\ -1.6033 + log10(0.9991) + log10(0.2507) \end{pmatrix}$$

$$= \max\begin{pmatrix} -4.8368 \\ -2.2045 \end{pmatrix} = -2.2045$$

$$Path_{O3}^{GC} = \arg\max\begin{pmatrix} -4.3567 \\ -1.6033 \end{pmatrix} = 2$$

- *Observable 4*

$$P_{O4=A}^{Kaya\ AT} = max\begin{pmatrix} P_{O3}^{AT} + log10\big(T(AT,AT)\big) + log10\big(E(A,AT)\big) \\ P_{O3}^{GC} + log10\big(T(GC,AT)\big) + log10\big(E(A,AT)\big) \end{pmatrix}$$

$$= \max\begin{pmatrix} -2.0595 + log10(0.9986) + log10(0.3011) \\ -2.2045 + log10(0.0009) + log10(0.3011) \end{pmatrix}$$

$$= \max\begin{pmatrix} -2.5813 \\ -5.7715 \end{pmatrix} = -2.5813$$

$$Path_{O4}^{AT} = \arg\max\begin{pmatrix} -2.5813 \\ -5.7715 \end{pmatrix} = 1$$

$$P_{O4=A}^{Kaya\ GC} = max\begin{pmatrix} P_{O3}^{AT} + log10\big(T(AT,GC)\big) + log10\big(E(A,GC)\big) \\ P_{O3}^{GC} + log10\big(T(GC,GC)\big) + log10\big(E(A,GC)\big) \end{pmatrix}$$

$$= \max\begin{pmatrix} -2.0595 + log10(0.0014) + log10(0.2082) \\ -2.2045 + log10(0.9991) + log10(0.2082) \end{pmatrix}$$

$$= \max\begin{pmatrix} -5.5949 \\ -2.8864 \end{pmatrix} = -2.8864$$

$$Path_{O4}^{GC} = \arg\max\begin{pmatrix} -5.5949 \\ -2.8864 \end{pmatrix} = 2$$

- *Observable 5*

$$P_{O5=G}^{Kaya\ AT} = max\begin{pmatrix} P_{O4}^{AT} + log10\big(T(AT,AT)\big) + log10\big(E(G,AT)\big) \\ P_{O4}^{GC} + log10\big(T(GC,AT)\big) + log10\big(E(G,AT)\big) \end{pmatrix}$$

$$= \max\begin{pmatrix} -2.5813 + log10(0.9986) + log10(0.2105) \\ -2.8864 + log10(0.0009) + log10(0.2105) \end{pmatrix}$$

$$= \max\begin{pmatrix} -3.2588 \\ -6.6089 \end{pmatrix} = -3.2588$$

$$Path_{O5}^{AT} = \arg\max\begin{pmatrix} -3.2588 \\ -6.6089 \end{pmatrix} = 1$$

$$P_{O5=G}^{Kaya\ GC} = max\begin{pmatrix} P_{O4}^{AT} + log10\big(T(AT,GC)\big) + log10\big(E(G,GC)\big) \\ P_{O4}^{GC} + log10\big(T(GC,GC)\big) + log10\big(E(G,GC)\big) \end{pmatrix}$$

$$= \max\begin{pmatrix} -2.5813 + log10(0.0014) + log10(0.2507) \\ -2.8864 + log10(0.9991) + log10(0.2507) \end{pmatrix}$$

$$= \max\begin{pmatrix} -6.0361 \\ -3.4876 \end{pmatrix} = -3.4876$$

$$Path_{O5}^{GC} = \arg\max\begin{pmatrix} -6.0361 \\ -3.4876 \end{pmatrix} = 2$$

- *Observable 6*

$$P_{O6=A}^{Kaya\ AT} = max\begin{pmatrix} P_{O5}^{AT} + log10\big(T(AT,AT)\big) + log10\big(E(A,AT)\big) \\ P_{O5}^{GC} + log10\big(T(GC,AT)\big) + log10\big(E(A,AT)\big) \end{pmatrix}$$

$$= \max\begin{pmatrix} -3.2588 + log10(0.9986) + log10(0.3011) \\ -3.4876 + log10(0.0009) + log10(0.3011) \end{pmatrix}$$

$$= \max\begin{pmatrix} -3.7806 \\ -7.0546 \end{pmatrix} = -3.7806$$

$$Path_{O6}^{AT} = \arg\max\begin{pmatrix} -3.7806 \\ -7.0546 \end{pmatrix} = 1$$

$$P_{O6=A}^{Kaya\ GC} = max\begin{pmatrix} P_{O5}^{AT} + log10\big(T(AT,GC)\big) + log10\big(E(A,GC)\big) \\ P_{O5}^{GC} + log10\big(T(GC,GC)\big) + log10\big(E(A,GC)\big) \end{pmatrix}$$

$$= \max\begin{pmatrix} -3.2588 + log10(0.0014) + log10(0.2082) \\ -3.4876 + log10(0.9991) + log10(0.2082) \end{pmatrix}$$

$$= \max\begin{pmatrix} -6.7942 \\ -4.1694 \end{pmatrix} = -4.1694$$

$$Path_{O6}^{GC} = \arg\max\begin{pmatrix} -6.7942 \\ -4.1694 \end{pmatrix} = 2$$

- *Observable 7*

$$P_{O7=A}^{Kaya\ AT} = max\begin{pmatrix} P_{O6}^{AT} + log10\big(T(AT,AT)\big) + log10\big(E(A,AT)\big) \\ P_{O6}^{GC} + log10\big(T(GC,AT)\big) + log10\big(E(A,AT)\big) \end{pmatrix}$$

$$= \max\begin{pmatrix} -3.7806 + log10(0.9986) + log10(0.3011) \\ -4.1694 + log10(0.0009) + log10(0.3011) \end{pmatrix}$$

$$= \max\begin{pmatrix} -4.3024 \\ -7.7364 \end{pmatrix} = -4.3024$$

$$Path_{O7}^{AT} = \arg\max\begin{pmatrix}-4.3024\\-7.7364\end{pmatrix} = 1$$

$$P_{O7=A}^{Kaya\ GC} = max\begin{pmatrix}P_{O6}^{AT} + log10\big(T(AT,GC)\big) + log10\big(E(A,GC)\big)\\P_{O6}^{GC} + log10\big(T(GC,GC)\big) + log10\big(E(A,GC)\big)\end{pmatrix}$$

$$= \max\begin{pmatrix}-3.7806 + log10(0.0014) + log10(0.2082)\\-4.1694 + log10(0.9991) + log10(0.2082)\end{pmatrix}$$

$$= \max\begin{pmatrix}-7.3159\\-4.8512\end{pmatrix} = -4.8512$$

$$Path_{O7}^{GC} = \arg\max\begin{pmatrix}-7.3159\\-4.8512\end{pmatrix} = 2$$

- *Observable 8*

$$P_{O8=G}^{Kaya\ AT} = max\begin{pmatrix}P_{O7}^{AT} + log10\big(T(AT,AT)\big) + log10\big(E(G,AT)\big)\\P_{O7}^{GC} + log10\big(T(GC,AT)\big) + log10\big(E(G,AT)\big)\end{pmatrix}$$

$$= \max\begin{pmatrix}-4.3024 + log10(0.9986) + log10(0.2105)\\-4.8512 + log10(0.0009) + log10(0.2105)\end{pmatrix}$$

$$= \max\begin{pmatrix}-4.9798\\-8.5737\end{pmatrix} = -4.9798$$

$$Path_{O8}^{AT} = \arg\max\begin{pmatrix}-4.9798\\-8.5737\end{pmatrix} = 1$$

$$P_{O8=G}^{Kaya\ GC} = max\begin{pmatrix}P_{O7}^{AT} + log10\big(T(AT,GC)\big) + log10\big(E(G,GC)\big)\\P_{O7}^{GC} + log10\big(T(GC,GC)\big) + log10\big(E(G,GC)\big)\end{pmatrix}$$

$$= \max\begin{pmatrix}-4.3024 + log10(0.0014) + log10(0.2507)\\-4.8512 + log10(0.9991) + log10(0.2507)\end{pmatrix}$$

$$= \max\begin{pmatrix}-7.5717\\-5.4524\end{pmatrix} = -5.4524$$

$$Path_{O8}^{GC} = \arg\max\begin{pmatrix}-7.5717\\-5.4524\end{pmatrix} = 2$$

- *Observable 9*

$$P_{O9=C}^{Kaya\ AT} = max\begin{pmatrix}P_{O8}^{AT} + log10\big(T(AT,AT)\big) + log10\big(E(C,AT)\big)\\P_{O8}^{GC} + log10\big(T(GC,AT)\big) + log10\big(E(C,AT)\big)\end{pmatrix}$$

$$= \max\begin{pmatrix}-4.9798 + log10(0.9986) + log10(0.2121)\\-5.4524 + log10(0.0009) + log10(0.2121)\end{pmatrix}$$

$$= \max\begin{pmatrix}-5.6539\\-9.1716\end{pmatrix} = -5.6539$$

$$Path_{O9}^{AT} = \arg\max\begin{pmatrix}-5.6539\\-9.1716\end{pmatrix} = 1$$

$$P_{O9=C}^{Kaya\ GC} = max\begin{pmatrix}P_{O8}^{AT} + log10\big(T(AT,GC)\big) + log10\big(E(C,GC)\big)\\P_{O8}^{GC} + log10\big(T(GC,GC)\big) + log10\big(E(C,GC)\big)\end{pmatrix}$$

$$= \max\begin{pmatrix}-4.9798 + log10(0.0014) + log10(0.3012)\\-5.4524 + log10(0.9991) + log10(0.3012)\end{pmatrix}$$

$$= \max\begin{pmatrix}-8.3548\\-5.9740\end{pmatrix} = -5.9740$$

$$Path_{O9}^{GC} = \arg\max\begin{pmatrix}-8.3548\\-5.9740\end{pmatrix} = 2$$

- *Observable 10*

$$P_{O10=A}^{Kaya\ AT} = max\begin{pmatrix}P_{O9}^{AT} + log10\big(T(AT,AT)\big) + log10\big(E(A,AT)\big)\\P_{O9}^{GC} + log10\big(T(GC,AT)\big) + log10\big(E(A,AT)\big)\end{pmatrix}$$

$$= \max\begin{pmatrix}-5.6539 + log10(0.9986) + log10(0.3011)\\-5.9740 + log10(0.0009) + log10(0.3011)\end{pmatrix}$$

$$= \max\begin{pmatrix}-6.1757\\-9.5410\end{pmatrix} = -6.1757$$

$$Path_{O10}^{AT} = \arg\max\begin{pmatrix}-6.1757\\-9.5410\end{pmatrix} = 1$$

$$P_{O10=A}^{Kaya\ GC} = max\begin{pmatrix}P_{O9}^{AT} + log10\big(T(AT,GC)\big) + log10\big(E(A,GC)\big)\\P_{O9}^{GC} + log10\big(T(GC,GC)\big) + log10\big(E(A,GC)\big)\end{pmatrix}$$

$$= \max\begin{pmatrix}-5.6539 + log10(0.0014) + log10(0.2082)\\-5.9740 + log10(0.9991) + log10(0.2082)\end{pmatrix}$$

$$= \max\begin{pmatrix}-9.1893\\-6.6558\end{pmatrix} = -6.6558$$

$$Path_{O10}^{GC} = \arg\max\begin{pmatrix}-9.1893\\-6.6558\end{pmatrix} = 2$$

- *Observable 11*

$$P_{O11=G}^{Kaya\ AT} = max\begin{pmatrix}P_{O10}^{AT} + log10\big(T(AT,AT)\big) + log10\big(E(G,AT)\big)\\P_{O10}^{GC} + log10\big(T(GC,AT)\big) + log10\big(E(G,AT)\big)\end{pmatrix}$$

$$= \max\begin{pmatrix} -6.1757 + log10(0.9986) + log10(0.2105) \\ -6.6558 + log10(0.0009) + log10(0.2105) \end{pmatrix}$$

$$= \max\begin{pmatrix} -6.8531 \\ -10.3783 \end{pmatrix} = -6.8531$$

$$Path_{O11}^{AT} = \arg\max\begin{pmatrix} -6.8351 \\ -10.3783 \end{pmatrix} = 1$$

$$P_{O11=G}^{Kaya\ GC} = max\begin{pmatrix} P_{O10}^{AT} + log10\big(T(AT, GC)\big) + log10\big(E(G, GC)\big) \\ P_{O10}^{GC} + log10\big(T(GC, GC)\big) + log10\big(E(G, GC)\big) \end{pmatrix}$$

$$= \max\begin{pmatrix} -6.1757 + log10(0.0014) + log10(0.2507) \\ -6.6558 + log10(0.9991) + log10(0.2507) \end{pmatrix}$$

$$= \max\begin{pmatrix} -9.6304 \\ -7.2570 \end{pmatrix} = -7.2570$$

$$Path_{O11}^{GC} = \arg\max\begin{pmatrix} -9.6304 \\ -7.2570 \end{pmatrix} = 2$$

- *Observable 12*

$$P_{O12=A}^{Kaya\ AT} = max\begin{pmatrix} P_{O11}^{AT} + log10\big(T(AT, AT)\big) + log10\big(E(A, AT)\big) \\ P_{O11}^{GC} + log10\big(T(GC, AT)\big) + log10\big(E(A, AT)\big) \end{pmatrix}$$

$$= \max\begin{pmatrix} -6.8531 + log10(0.9986) + log10(0.3011) \\ -7.2570 + log10(0.0009) + log10(0.3011) \end{pmatrix}$$

$$= \max\begin{pmatrix} -7.3749 \\ -10.8240 \end{pmatrix} = -7.3749$$

$$Path_{O12}^{AT} = \arg\max\begin{pmatrix} -7.3749 \\ -10.8240 \end{pmatrix} = 1$$

$$P_{O12=A}^{Kaya\ GC} = max\begin{pmatrix} P_{O11}^{AT} + log10\big(T(AT, GC)\big) + log10\big(E(A, GC)\big) \\ P_{O11}^{GC} + log10\big(T(GC, GC)\big) + log10\big(E(A, GC)\big) \end{pmatrix}$$

$$= \max\begin{pmatrix} -6.8531 + log10(0.0014) + log10(0.2082) \\ -7.2570 + log10(0.9991) + log10(0.2082) \end{pmatrix}$$

$$= \max\begin{pmatrix} -10.3885 \\ -7.9388 \end{pmatrix} = -7.9388$$

$$Path_{O12}^{GC} = \arg\max\begin{pmatrix} -10.3885 \\ -7.9388 \end{pmatrix} = 2$$

- *Observable 13*

$$P_{O13=T}^{Kaya\ AT} = max\begin{pmatrix} P_{O12}^{AT} + log10\big(T(AT,AT)\big) + log10\big(E(T,AT)\big) \\ P_{O12}^{GC} + log10\big(T(GC,AT)\big) + log10\big(E(T,AT)\big) \end{pmatrix}$$

$$= \max\begin{pmatrix} -7.3749 + log10(0.9986) + log10(0.2763) \\ -7.9388 + log10(0.0009) + log10(0.2763) \end{pmatrix}$$

$$= \max\begin{pmatrix} -7.9342 \\ -11.5432 \end{pmatrix} = -7.9342$$

$$Path_{O13}^{AT} = \arg\max\begin{pmatrix} -7.9342 \\ -11.5432 \end{pmatrix} = 1$$

$$P_{O13=T}^{Kaya\ GC} = max\begin{pmatrix} P_{O12}^{AT} + log10\big(T(AT,GC)\big) + log10\big(E(T,GC)\big) \\ P_{O12}^{GC} + log10\big(T(GC,GC)\big) + log10\big(E(T,GC)\big) \end{pmatrix}$$

$$= \max\begin{pmatrix} -7.3749 + log10(0.0014) + log10(0.2399) \\ -7.9388 + log10(0.9991) + log10(0.2399) \end{pmatrix}$$

$$= \max\begin{pmatrix} -10.8487 \\ -8.5593 \end{pmatrix} = -8.5593$$

$$Path_{O13}^{GC} = \arg\max\begin{pmatrix} -10.8487 \\ -8.5593 \end{pmatrix} = 2$$

- *Observable 14*

$$P_{O14=A}^{Kaya\ AT} = max\begin{pmatrix} P_{O13}^{AT} + log10\big(T(AT,AT)\big) + log10\big(E(A,AT)\big) \\ P_{O13}^{GC} + log10\big(T(GC,AT)\big) + log10\big(E(A,AT)\big) \end{pmatrix}$$

$$= \max\begin{pmatrix} -7.9342 + log10(0.9986) + log10(0.3011) \\ -8.5593 + log10(0.0009) + log10(0.3011) \end{pmatrix}$$

$$= \max\begin{pmatrix} -8.4560 \\ -12.1263 \end{pmatrix} = -8.4560$$

$$Path_{O14}^{AT} = \arg\max\begin{pmatrix} -8.4560 \\ -12.1263 \end{pmatrix} = 1$$

$$P_{O14=A}^{Kaya\ GC} = max\begin{pmatrix} P_{O13}^{AT} + log10\big(T(AT,GC)\big) + log10\big(E(A,GC)\big) \\ P_{O13}^{GC} + log10\big(T(GC,GC)\big) + log10\big(E(A,GC)\big) \end{pmatrix}$$

$$= \max\begin{pmatrix} -7.9342 + log10(0.0014) + log10(0.2082) \\ -8.5593 + log10(0.9991) + log10(0.2082) \end{pmatrix}$$

$$= \max\begin{pmatrix} -11.4695 \\ -9.2412 \end{pmatrix} = -9.2412$$

$$Path_{O14}^{GC} = \arg\max\begin{pmatrix} -11.4695 \\ -9.2412 \end{pmatrix} = 2$$

- *Observable 15*

$$P_{O15=A}^{Kaya\ AT} = max\begin{pmatrix} P_{O14}^{AT} + log10\big(T(AT,AT)\big) + log10\big(E(A,AT)\big) \\ P_{O14}^{GC} + log10\big(T(GC,AT)\big) + log10\big(E(A,AT)\big) \end{pmatrix}$$

$$= \max\begin{pmatrix} -8.4560 + log10(0.9986) + log10(0.3011) \\ -9.2412 + log10(0.0009) + log10(0.3011) \end{pmatrix}$$

$$= \max\begin{pmatrix} -8.9778 \\ -12.8082 \end{pmatrix} = -8.9778$$

$$Path_{O15}^{AT} = \arg\max\begin{pmatrix} -8.9778 \\ -12.8082 \end{pmatrix} = 1$$

$$P_{O15=A}^{Kaya\ GC} = max\begin{pmatrix} P_{O14}^{AT} + log10\big(T(AT,GC)\big) + log10\big(E(A,GC)\big) \\ P_{O14}^{GC} + log10\big(T(GC,GC)\big) + log10\big(E(A,GC)\big) \end{pmatrix}$$

$$= \max\begin{pmatrix} -8.4560 + log10(0.0014) + log10(0.2082) \\ -9.2412 + log10(0.9991) + log10(0.2082) \end{pmatrix}$$

$$= \max\begin{pmatrix} -11.4695 \\ -9.9231 \end{pmatrix} = -9.9231$$

$$Path_{O15}^{GC} = \arg\max\begin{pmatrix} -11.9914 \\ -9.2412 \end{pmatrix} = 2$$

Proses diatas merupakan konfirmasi kebenaran perhitungan HMM bila dibandingkan dengan perhitungan dengan program. Hasil yang didapatkan adalah

$$O = ATGAGAAGCAGATAA$$
$$H = 1111111111111111$$

Konfirmasi dari segmentasi AT- GC *rich* berdasarkan *windowing* adalah

$$O = ATGAGAAGCAGATAA$$
$$H = 1111111111111111$$
$$W = 1111111111111111$$

2.3.3 Analisa

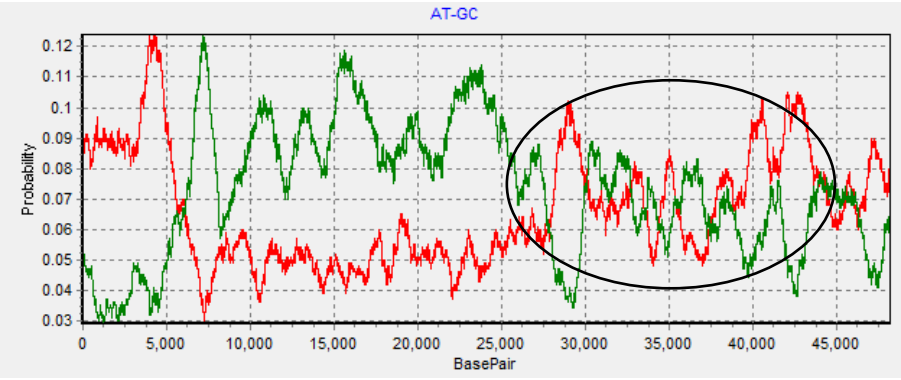
Analisa kaya AT- GC pada ORF dapat diamati berdasarkan *window*. Pada kasus ORF1 dengan panjang 15, ORF cenderung berada pada state kaya AT. Pendekatan dilakukan dengan memperkecil *window* hingga 10. Tampak pada konfirmasi state kaya AT – GC seharusnya terjadi *switching*, akan tetapi dengan pemodelan HMM tetap menunjukkan

$$O = ATGAGAAGCAGATAA$$
$$H = 11111111111111111$$
$$W = 11122222222111111$$

Hal ini terjadi karena berdasarkan HMM, sekalipun pada state terjadi perubahan dari AT ke GC tetapi karena kejadian tersebut berada pada lautan AT sehingga tetap terdeteksi sebagai kaya AT.

HMM untuk deteksi ORF dilakukan dengan memanfaatkn *markov chain* dibawah. Parameter transisi dan emisi dari *state* ditentukan berdasarkan analogi kejadian ORF. Promotor terdiri dari 3 *state* masing-masing untuk A, T dan G dengan emisi 1 sesuai basa yang ditunjuk. Region *coding* terdiri dari tiga state untuk tiap kodon dengan emisi 0.25 tiap-tiap basa. Transisi region *coding* ke *terminator* menjadi pertimbangan kecenderungan panjang ORF pada satu DNA. Pada terminator sendiri memerlukan tiga buah state untuk merepresentasikan kodon stop. Daerah non-coding memiliki komposisi tiga basa dan transisinya ke start menjadi pertimbangan banyak ORF yang akan muncul pada satu DNA.

Data yang digunakan adalah data E. Phage Lambda dengan nilai unik pada transisi *coding-region* ke terminator 0.66 dan transisi *non-coding region* ke promotor dengan nilai 0.5. Didapatkan sebanyak 230 ORF yang akan disimpan untuk pengujian AT-GC rich tiap-tiap ORF. Hal menarik lain yang didapatkan dari hubungan *window* dengan ORF adalah kejadian *switching* kaya AT dan GC pada ORF. Pengaruh *window* sangat besar untuk penampakan *switching* kaya AT-GC



Gambar 5. Kawasan tidak stabil antara kaya AT dan GC pada *phage lambda*

pada ORF. Pada kenyataanya, pemilihan lebar window yang besar maka akan mengurangi *switching* pada tiap-tiap ORF. Hal ini terjadi karena panjang *window* yang sekalipun menampakkan *switching* telah melebihi panjang ORF. Untuk mengujinya, maka dipilih panjang *window* yang kecil kemudian dihitung profil HMM dan diterapkan pada ORF dengan panjang yang besar. ORF akan menyesuaikan dengan *switching* yang terjadi sesuai pergeseran *window*. Profil statisik dari E. Phage Lambda adalah memiliki ORF terpanjang 1694 bp. Dengan pertimbangan ini, maka pemberian *window* mencapai 1000 akan menyebabkan *switching* pada ORF minim terjadi. ORF pada *E. Phage Lambda* banyak berada pada kisaran panjang 70 hingga 120 bp. Estimasi pengujian adalah mencari ORF pada lokasi 30.000 bp hingga maksimal 40.000 bp karena pada E. Lambda Phage tidak stabil pada rentang ini dan mencoba lebar window yang berbeda. ORF yang digunakan adalah ORF122 dengan panjang 714 bp dari posisi basa 38638 hingga 39349. Estimasi pertama adalah dengan memanfaatkan panjang *window* 1000, didapatkan hasil keseluruhan ORF berada pada *state* kaya AT seluruhnya. *Switching* baru terjadi pada ORF ketika lebar window 10. Window sedemikian rupa menyebabkan ORF51 mengalami *switching* pada 449 dari AT ke GC dan pada 563 dari GC ke AT.

Gambar 6. Kejadian *switching* pada Pemodelan HMM pada kasus *window* yang sempit (10) dan ORF yang panjang

BAB III. Kesimpulan

Berdasarkan pengujian yang telah dilakukan, maka dapat disimpulkan beberapa hal berikut

1. Model Hidden Markov (HMM) dapat digunakan sebagai perbaikan segmentasi.
2. HMM dapat diterapkan pada segmentasi *coding-noncoding* region berdasarkan pemahaman terkait *open reading frames*.
3. HMM dapat diterapkan pada segmentasi kaya AT- GC berdasarkan densitas AT-GC pada keseluruhan DNA.
4. *Window* berpengaruh besar terhadap performa segmentasi kaya AT-GC. Pemilihan *window* yang baik dapat didasarkan pada persebaran panjang ORF yang terdeteksi

BAB IV. Referensi

Cristianini, N., & Hahn, M. W. (2006). *Introduction to Computational Genomics: A Case Studies Approach.* New York: Cambridge University Press.

Jones, N. C., & Pevzner, P. A. (2004). *An Introduction to Bioinformatics Algorithm.* Cambridge: MIT Press.

Santalucia, J. (1998). A unified view of polymer, dumbbell, and oligonucleotide DNA. *Proceedings of the National Academy of Sciences of the United States of America*, 1460–1465.

Vinogradov, A. E. (2003). DNA Helix: the importance of being GC-Rich. *Nucleic Acid Research*, 1838-1844.