pairHMMs forward algorithm

Zijie Zhang

June 2022

Contents

1 Introduction

Next-Generation Sequencing (NGS) platforms nowadays are able to generate large amounts of DNA sequencing data at low cost. However, handling with these amounts of data consumes much computation time and requires high computational capabilities. For this reason, are developed tools like *GATK* - Genomics Analysis ToolKit - to help researchers study and investigate such DNA data.

GATK HaplotypeCaller (HC) is a widely used variant caller tool to identify DNA variants by comparing a DNA sequencing data with a reference genome.

Since in GATK HC, the PairHMMs forward algorithm accounts for a large percentage of the total execution time, which is applied to study the overall alignment probability of two sequences, optimize the forward algorithm becomes relevant.

In this report is introduced forward algorithm with a briefly discussion on how to optimize it.

2 Hidden Markov Model

Before talking about the forward algorithm, is indispensable introduce HMM.

A *Hidden Markov Model (*HMM*)* is a probabilistic model of the sequence of events that occur in a system. In the model is required three states, M corresponding to a **match**, and two states corresponding to **insertion** and **deletion**, named I_x and I_y.

State M has *emission probability* distribution p_{x_i,y_j} for emitting an aligned pair $x_i : y_j$ and states I_x and I_y have distributions q_{x_i} and q_{y_j} for emitting symbol against a gap.

Are denoted also *transition probabilities* between the states: the transition from M to an insert state (I_x or I_y) by δ and staying in an insert state by ε.

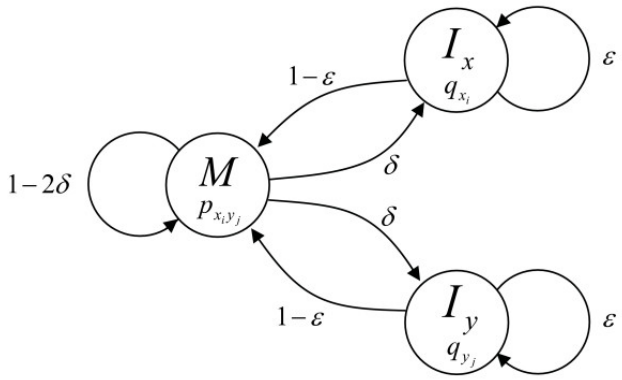


Figure 1: Finite state machine of transitions among states.

Adding an explicit *Begin* and *End* state formalise the initialisation and termination conditions, and introduces the need for another parameter: the probability of a transition into the End state, which is assumed for now to be the same from each of M, X and Y.

The new probabilistic model instead of emitting a single sequence it emits a pairwise alignment and is denoted as a pair HMM.

Having a pair HMM allows to bypass the problem of weak-similarity between sequences by calculating the probability that a given pair of sequences are related according to the HMM by any alignment: the *forward algorithm.*

3 Forward algorithm

The idea of forward algorithm is to find the maximal scoring alignment, but add rather than take the maximum at each step.

The forward algorithm is performed as shown in equations (1)-(3). m and n are the length of the read R and the haplotype H.

$M_{i,j}$ is the overall alignment probability of two subsequences $R_1, \ldots, R_i$ and $H_1, \ldots, R_j$ when R_i is aligned to H_j.

$I_{i,j}$ is the overall alignment probability of $R_1, \ldots, R_i$ and $H_1, \ldots, R_j$ when R_i is aligned to a gap.

$D_{i,j}$ is the overall alignment probability of $R_1, \ldots, R_i$ and $H_1, \ldots, R_j$ when H_j is aligned to a gap.

$\alpha, \delta, \varepsilon$ are transition probabilities. In particular δ and ε are set to be constant (by default are 0.9 and 0.1, respectively). $\lambda_{i,j}$ is the emission probability, its calculation is shown in equation (4).

Initialization:

$$\begin{cases} M_{i,0} = I_{i,0} = D_{i,0} = 0 \quad (0 \leq i \leq m) \\ M_{0,j} = I_{0,j} = 0 \quad (0 \leq j \leq n) \\ D_{0,j} = 1/n \quad (0 \leq j \leq n) \end{cases} \tag{1}$$

Recursion:

$$\begin{cases} M_{i,j} = \lambda_{i,j}(\alpha_i M_{i-1,j-1} + \beta_i I_{i-1,j-1} + \beta_i D_{i-1,j-1}) \\ I_{i,j} = \delta_i M_{i-1,j} + \varepsilon_i I_{i-1,j} \\ D_{i,j} = \delta_i M_{i,j-1} + \varepsilon_i D_{i,j-1} \end{cases} \tag{2}$$

Termination:

$$Result = \sum_{j=1}^{n} (M_{m,j} + I_{m,j}) \tag{3}$$

Emission probability:

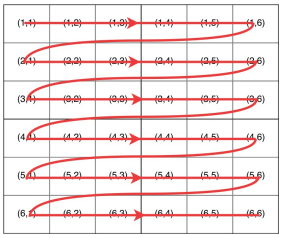
$$\lambda_{i,j} = \begin{cases} \epsilon(q)/3 & (\text{ if } \quad R_i \neq H_j \ ) \\ 1 - \epsilon(q) & (\text{ if } \quad R_i = H_j \ ) \end{cases} \tag{4}$$

where $\epsilon(q) = 10^{-q/10}$ is the error probability given by the phred-scaled quality q.

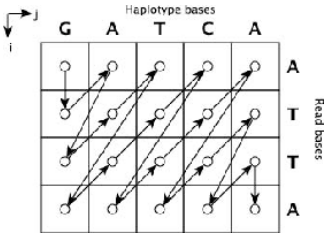
Algorithm 1 PairHMM forward algorithm

```
Initialize M, I, D via equation (1)
for i ← 1 to m do
    for j ← 1 to n do
        Calculate λ_{i,j} via equation (4)
        Calculate M_{i,j}, I_{i,j}, D_{i,j} via equation (2)
    end for
end for
return Σ_{j=1}^{n} (M_{m,j} + I_{m,j})
```

This algorithm employs a 2-layer loop to calculate the elements of three matrices with a computational complexity of $\mathcal{O}(mn)$. As shown before, $M_{i,j}$, $I_{i,j}$ and $D_{i,j}$ are only decided by the *left, top-left,* and *top* elements of three matrices. This implies that the elements on the same *antidiagonal* have not data-dependency, thus are able to be calculated in parallel.



(a) Native implementation



(b) Parallel Computing

4 References

1. R. Durbin. *Biological Sequence analysis.*
2. D. Benjamin. *Pair HMM probabilistic realignment in HaplotypeCaller and Mutect.*
3. Rick Wertenbroek. *Acceleration of the Pair-HMM forward algorithm on FPGA with cloud integration for GATK.*
4. Shanshan Ren. *Efficient Acceleration of the Pair-HMMs Forward Algorithm for GATK HaplotypeCaller on Graphics Processing Units*