MO640/MC668

Guilherme P. Telles

IC-Unicamp

Avisado está

- Estes slides são incompletos.
- Estes slides contêm erros.

Parte I

Mapeamento de reads

Mapeamento de reads

- Dados um genoma de referência e um conjunto de reads, queremos determinar a posição em que cada read ocorre no genoma.
- Vamos supor que o genoma é uma cadeia G de tamanho n e que temos m reads com tamanhos $\{\ell_1, \ldots, \ell_m\}$.

Solução direta

- Sem admitir erros: comparar cada subcadeia do genoma contra cada read. É $O(n \sum_{i=1}^{m} \ell_i)$.

Solução direta

- Sem admitir erros: comparar cada subcadeia do genoma contra cada read. É $O(n \sum_{i=1}^{m} \ell_i)$.
- Admitindo erros: comparar cada subcadeia do genoma contra cada read usando PD. É $O(n \sum_{i=1}^{m} \ell_i^2)$.

Solução direta

- Sem admitir erros: comparar cada subcadeia do genoma contra cada read. É $O(n \sum_{i=1}^{m} \ell_i)$.
- Admitindo erros: comparar cada subcadeia do genoma contra cada read usando PD. É $O(n \sum_{i=1}^{m} \ell_i^2)$.
- Considerando que as tecnologias correntes de seqüenciamento produzem muitos reads, não são viáveis.

- Vamos supor que os reads tenham tamanho fixo, digamos 40 bp.
- Vamos supor que tanto os reads como os genomas podem ter a letra N, que indica uma base desconhecida.

Estratégias

- Há duas grandes classes de estratégias para mapping:
  - baseadas em hashing.
  - baseadas em vetor de sufixos e Transformada de Burrows-Wheeler.

Hashing

- Em linhas gerais, as estratégias baseadas em hashing buscam subcadeias de cada read nos genomas (sementes) e estendem o casamento para todo o comprimento do read.

Hashing

- Uma possibilidade é construir uma tabela de hashing com todas as subcadeias de tamanho 40 que aparecem no genoma.
- Para cada subcadeia, a tabela registra a posição onde elas ocorrem no genoma.
- Para verificar se um read aparece no genoma basta consultar a tabela de hashing, o que leva tempo esperado constante.
- Há potencialmente n entradas nessa tabela.
- Essa solução é simples, mas produz uma estrutura muito grande e não permite aceitar erros com facilidade.

Hashing

- Outra possibilidade é construir uma tabela de hashing para subcadeias de tamanho k bem menor que os reads.
- Por exemplo, para $k = 10$ há $4^{10} = 1.048.576$ entradas na tabela no máximo.

Hashing

- Para verificar se um read aparece no genoma, cada subcadeia de tamanho k do read é buscada na tabela.
- Chamamos essa subcadeia de semente.
- Se essa região for encontrada então o read é alinhado contra a região do genoma usando PD.
- Se há muitas regiões repetidas essa estratégia é lenta.

Hashing

- Outra possibilidade é quebrar o read em subcadeias não sobrepostas de tamanho k.
- Para determinar se um read aparece no genoma sem erros, verifica-se se as subcadeias batem consecutivamente no genoma.
- Para determinar se um read aparece no genoma admitindo erros, verifica-se se uma parte subcadeias batem no genoma em posições corretas. Por exemplo, se duas das quatro subcadeias batem nas posições adequadas, então executa-se PD entre o read e a região do genoma.

Hashing

- Outra possibilidade é considerar todas as subcadeias de tamanho k de um read.
- Para determinar se um read aparece no genoma admitindo erros, verifica-se se um número mínimo de subcadeias aparece no genoma em posições corretas.

Erros

- Se o valor de k é muito pequeno então o número de ocorrências de sementes será muito grande e o processamento será inviável.
- Um valor de k menor que 10 não costuma ser usado na prática.
- Então os métodos anteriores não são muito flexíveis quanto aos erros (ou são muito lentos).

Erros

- Alguns métodos usam uma máscara, por exemplo $111xx11xxx$, para indicar que o símbolo na posição marcada com x não será comparado contra um símbolo do genoma.
- Isso permite aceitar mais erros ao verificar se uma subcadeia do read estáno genoma, mas aumenta o tempo de computação.

Erros

- Alguns métodos admitem mais erros ao determinar as sementes, mas usam um filtro para limitar o tempo de computação.
- A partir de uma semente, eles contam o número de cada letra nas regiões de extensão do read e do genoma. Se os números estiverem dentro de um certo limiar de diferença então executam a PD.

Parte II

Árvore de sufixos

Árvore de sufixos

- Uma árvore de sufixos para uma cadeia $s\$$ de tamanho m é uma árvore enraizada com folhas numeradas e arestas rotuladas tal que
  - cada nó interno tem pelo menos dois filhos, exceto pela raiz,
  - cada aresta é rotulada com uma subcadeia de s,
  - toda folha é rotulada com um inteiro distinto entre 1 e m,
  - o rótulo de cada aresta que sai de um nó começa com símbolos diferentes,
  - a concatenação dos rótulos em um caminho da raiz até a folha i é igual a $s[i..m]\$$.

Terminador

- Seja s uma cadeia para a qual queremos construir uma árvore de sufixos.
- Para garantir que uma árvore de sufixos sempre exista, adicionamos o símbolo terminador $\$$ a s.
- $\$$ não faz parte do alfabeto que originou s e é lexicograficamente menor que todos os outros símbolos.
- Assim nenhum sufixo de s é prefixo de outro sufixo.
- Por exemplo, não haveria árvore de sufixos para $xabxa$, já que xa não seria folha.