Estudo e Implementação de Ordenação por Reversão Com Sinal

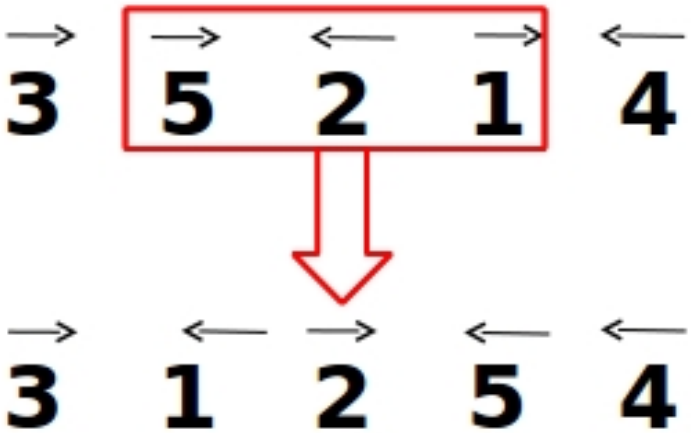
Fernando Chirigati, Rafael Dahis, Rafael Lopes, Victor Bursztyn

Biologia Computacional
Engenharia de Sistemas e Computação
Mestrado em Engenharia de Sistemas e Computação
Universidade Federal do Rio de Janeiro
Professoras Celina e Marília

30 de Maio de 2011

Agenda

- Comparar genomas = descobrir a distância entre genes
- Reversão = um tipo de operação de reorganização das bases de um gene
- Ordenação por reversão = reordenar uma cadeia utilizando somente reversões



Agenda

- Desenvolvimento em C
- Alocação dinâmica de memória
- Limitação de Escopo: não consideramos tratamento de componentes ruins

- Desenvolvimento em C
- Alocação dinâmica de memória
- Limitação de Escopo: não consideramos tratamento de componentes ruins
- Mas devemos sempre fazer o teste!

- Desenvolvimento em C
- Alocação dinâmica de memória
- Limitação de Escopo: não consideramos tratamento de componentes ruins
- Mas devemos sempre fazer o teste!
- Algoritmo termina se encontrar alguma componente ruim

- Desenvolvido por Hannenhalli e Pevzner em 1995.
- Baseado no Diagrama de Realidade e Desejo(DRD).
- Complexidade alta.

Conceitos

- Arestas divergentes $\rightarrow$ ao dar orientação em um ciclo, duas arestas de desejo que estão em sentidos opostos no DRD.

Conceitos

- Arestas divergentes $\rightarrow$ ao dar orientação em um ciclo, duas arestas de desejo que estão em sentidos opostos no DRD.
- Arestas convergentes $\rightarrow$ caso as arestas de desejo estejam no mesmo sentido.

Conceitos

- Arestas divergentes $\rightarrow$ ao dar orientação em um ciclo, duas arestas de desejo que estão em sentidos opostos no DRD.
- Arestas convergentes $\rightarrow$ caso as arestas de desejo estejam no mesmo sentido.
- Ciclo bom $\rightarrow$ um ciclo que possui ao menos duas arestas divergentes.

Conceitos

- Arestas divergentes $\rightarrow$ ao dar orientação em um ciclo, duas arestas de desejo que estão em sentidos opostos no DRD.
- Arestas convergentes $\rightarrow$ caso as arestas de desejo estejam no mesmo sentido.
- Ciclo bom $\rightarrow$ um ciclo que possui ao menos duas arestas divergentes.
- Ciclo ruim $\rightarrow$ um ciclo que todas as arestas são convergentes.

Conceitos

- Arestas divergentes $\rightarrow$ ao dar orientação em um ciclo, duas arestas de desejo que estão em sentidos opostos no DRD.
- Arestas convergentes $\rightarrow$ caso as arestas de desejo estejam no mesmo sentido.
- Ciclo bom $\rightarrow$ um ciclo que possui ao menos duas arestas divergentes.
- Ciclo ruim $\rightarrow$ um ciclo que todas as arestas são convergentes.
- Componente de um diagrama $\rightarrow$ todos os ciclos que se cruzam formam uma componenete.

Conceitos

- Arestas divergentes $\rightarrow$ ao dar orientação em um ciclo, duas arestas de desejo que estão em sentidos opostos no DRD.
- Arestas convergentes $\rightarrow$ caso as arestas de desejo estejam no mesmo sentido.
- Ciclo bom $\rightarrow$ um ciclo que possui ao menos duas arestas divergentes.
- Ciclo ruim $\rightarrow$ um ciclo que todas as arestas são convergentes.
- Componente de um diagrama $\rightarrow$ todos os ciclos que se cruzam formam uma componenete.
- Componente boa $\rightarrow$ possui ao menos um ciclo bom.

Conceitos

- Arestas divergentes $\rightarrow$ ao dar orientação em um ciclo, duas arestas de desejo que estão em sentidos opostos no DRD.
- Arestas convergentes $\rightarrow$ caso as arestas de desejo estejam no mesmo sentido.
- Ciclo bom $\rightarrow$ um ciclo que possui ao menos duas arestas divergentes.
- Ciclo ruim $\rightarrow$ um ciclo que todas as arestas são convergentes.
- Componente de um diagrama $\rightarrow$ todos os ciclos que se cruzam formam uma componenete.
- Componente boa $\rightarrow$ possui ao menos um ciclo bom.
- Componente ruim $\rightarrow$ todos os ciclos são ruins.

Algoritmo

entrada: Permutação α
Saída: Uma reversão que ordena para a permutação Identidade.
if *existe uma componente boa no DRD(α)* **then**
    escolha duas arestas de realidade divergentes *e* *f* deste componente, tendo a certeza de que sua reversão não criará componentes ruins.
    return a reversão caracterizada por *e* e *f*.
end
Senão: Caso não tratado. Retorna "Não é possivel reverter a partir deste ponto. Teremos componentes ruins."

Baseado em dois teoremas:

Teorema 1

Em uma reversão ρ ocorrendo em duas arestas de realidade e e f, então:

- Se e e f pertencem a ciclos diferentes, a quantidade de ciclos diminui em um.
- Se e e f são arestas convergentes, então a quantidade de ciclos se mantém.
- Se e e f são arestas divergentes, então a quantidade de ciclos aumenta em um.

Teorema 2

Uma reversão feita por duas arestas divergentes do mesmo ciclo é uma reversão que ordena se e somente se ao reverter não gera nenhuma componente ruim.

Fluxo:

1. Pré-processamento da entrada
2. Criação das arestas de desejo
3. Criação das arestas de realidade
4. Procurar todas as componentes
5. Enquanto número de ciclos != n +1 : aplicar reversões, "making sure" que elas não criaram componentes ruins
6. Criar arquivo de saída, salvar output

Procurar todas as componentes

- Numerar as arestas de realidade
- Caminhar pelas arestas
  - Encontrando ciclos
  - Definindo componentes (conjunto de ciclos)
- Componente será ruim se não houver ao menos duas arestas divergentes em um de seus ciclos

Aplicar reversões, "making sure" que elas não criaram componentes ruins

- Para cada componente, para cada ciclo dentro do componente:
  - se houver arestas divergentes:
    1. Reverter! (inverter essas arestas de realidade)
- Se essa reversão não criou ciclos ruins: prosseguir.
- Caso contrário: testar outra possível reversão.

- Construção do DRD: O(n).
- encontrar ciclos: O(n).
- Determinar se um ciclo é bom ou ruim: O(n). Para todos os ciclos: $O(n^2)$
- Selecinar duas arestas divergentes: $O(n^2)$
- Verificar que sua reversão não cria componentes ruins: $O(n^2)$

- Construção do DRD: O(n).
- encontrar ciclos: O(n).
- Determinar se um ciclo é bom ou ruim: O(n). Para todos os ciclos: $O(n^2)$
- Selecinar duas arestas divergentes: $O(n^2)$
- Verificar que sua reversão não cria componentes ruins: $O(n^2)$
- Logo, os dois itens combinados formam uma complexidade $O(n^4)$.

- Construção do DRD: O(n).
- encontrar ciclos: O(n).
- Determinar se um ciclo é bom ou ruim: O(n). Para todos os ciclos: $O(n^2)$
- Selecinar duas arestas divergentes: $O(n^2)$
- Verificar que sua reversão não cria componentes ruins: $O(n^2)$
- Logo, os dois itens combinados formam uma complexidade $O(n^4)$.
- Devemos fazer essa ordenação até termos n ciclos. Logo, O(n) vezes.

- Construção do DRD: O(n).
- encontrar ciclos: O(n).
- Determinar se um ciclo é bom ou ruim: O(n). Para todos os ciclos: $O(n^2)$
- Selecinar duas arestas divergentes: $O(n^2)$
- Verificar que sua reversão não cria componentes ruins: $O(n^2)$
- Logo, os dois itens combinados formam uma complexidade $O(n^4)$.
- Devemos fazer essa ordenação até termos n ciclos. Logo, O(n) vezes.
- Complexidade final: $O(n^5)$.

Agenda

Introdução

- Proposto por Tannier, Bergeron, Sagot(2005).
- Complexidade de $O(n^2)$.

Introdução

- Proposto por Tannier, Bergeron, Sagot(2005).
- Complexidade de $O(n^2)$.
- Pode ser reduzido a $O(n\sqrt{nlogn})$ se utilizar uma estrutura de dados mais adequada.

Introdução

- Proposto por Tannier, Bergeron, Sagot(2005).
- Complexidade de $O(n^2)$.
- Pode ser reduzido a $O(n\sqrt{nlogn})$ se utilizar uma estrutura de dados mais adequada.
- Utiliza grafo de sobreposição.

Definições

- Arco v_i
- Arcos que se intersectam.
- Grafos de sobreposição.
- Arco e vértices orientados e não-orientados.
- Complemento local do grafo de sobreposição.

- É dado como entrada o conjunto de arcos que definem a permutação.
- Enquanto existir um arco orientado, adicione faça a reversão a partir do arco orientado, atualizando os arcos.
- Salve essas reversões.
- Reaplique as reversões ao contrário, até achar um arco orientado. Salve as reversões nesta feitas em segunda estrutura.
- Recomece, até todos os vértices serem seguros.
- Quando todos os vertices forem seguros, a ordem das reversões dada pelas duas estruturas definem as reversões a serem feitas.

Agenda

Estudo e Implementação de Ordenação por Reversão Com Sinal

Fernando Chirigati, Rafael Dahis, Rafael Lopes, Victor Bursztyn

Biologia Computacional
Engenharia de Sistemas e Computação
Mestrado em Engenharia de Sistemas e Computação
Universidade Federal do Rio de Janeiro
Professoras Celina e Marília

30 de Maio de 2011