Montagem de Fragmentos

Zanoni Dias

Instituto de Computação – Unicamp

28 de maio de 2009

Montagem de Fragmentos

- Tecnologia atual de sequenciamento não permite obter fragmentos de DNA maiores que 1000 pares de bases.
- Na prática, muitas vezes precisamos obter a sequência de organismos de milhões de pares de bases.
- Montagem de fragmentos é a terefa de, dado um conjunto de fragmentos, reconstruir a sequência que originou os fragmentos (sequência alvo), com base nas sobreposições dos fragmentos.
- Montagem de fragmentos pode ser revolvido com estratégias convencionais de alinhamento múltiplo de sequências?
  - Não! Apesar de parecidos, os problemas tem diferenças importantes e usam técnicas distintas para obter soluções.

Principais Dificuldades

- Erros de sequenciamento.
- Orientação desconhecida dos fragmentos.
- Falta de cobertura da sequência original.
- Tamanho desconhecido da sequência original.
- Regiões repetidas na sequência original.
- Sequências quiméricas.
- Contaminação pelo vetor de sequenciamento.

Modelos Computacionais para Montagem de Fragmentos

- Modelos mais comuns:
  - *Shortest Common Superstring (SCS).*
  - *Reconstruction.*
  - *Multicontig.*
- Todos estes modelos supõem que os fragmentos não possuem contaminações ou quimeras.

Shortest Common Superstring

- Dada uma coleção $\mathcal{F}$ de fragmentos, obter a menor sequência possível S, tal que para todo $f \in \mathcal{F}$, S é uma supersequência de f.
- Modelo essencialmente teórico, sem suporte a maioria dos problemas práticos.
- Pode não produzir a sequência original, devido a dificuldade de lidar com longos trechos repetidos.
- $SCS \in NP$-Completo.

Reconstruction

- Dada uma coleção $\mathcal{F}$ de fragmentos e uma tolerância de erro ϵ ($0 \leq \epsilon \leq 1$), obter a menor sequência possível S, tal que para todo $f \in \mathcal{F}$, temos:

$$min(dist_s(f, S), dist_s(\overline{f}, S)) \leq \epsilon |f|$$

onde $\overline{f}$ é o complemento reverso de f e $dist_s$ é definida como:

$$dist_s(a, b) = min_{s \in S(b)} dist(a, s)$$

onde $S(b)$ é o conjunto das subsequências de b.

- *Reconstruction* $\in NP$-Completo.
- *SCS* é um caso particular de *Reconstruction*.
  - *SCS* é equivalente a *Reconstruction* com $\epsilon = 0$.

Multicontig

- Dada uma coleção $\mathcal{F}$ de fragmentos, um inteiro $t \geq 0$ e uma tolerância de erro ϵ ($0 \leq \epsilon \leq 1$), obter uma partição de $\mathcal{F}$ em um número mínimo de subcoleções, $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$, tal que cada $\mathcal{C}_i$ (com $1 \leq i \leq k$) forma um *contig* com sobreposição mínima t entre os fragmento e taxa de erro ϵ de cada fragmento em relação ao consenso do *contig*.
- *Multicontig* $\in NP$-Completo.

Calculando o Progresso da Montagem

- Seja:
  - n: número de fragmentos.
  - f: tamanho médio dos fragmentos.
  - T: tamanho da sequência alvo a ser montada.
  - t: sobreposição mínima entre dois fragmentos para montagem.
- A cobertura média (c) da sequência alvo pode ser calculada como:
$$c = \frac{nf}{T}$$
- O número esperado de subsequências contíguas montadas com sobreposição mínima t é dado por:
$$p = ne^{\frac{-n(f-t)}{T}}$$
- O número esperado de subsequências contíguas montadas por pelo menos 2 fragmentos, com sobreposição mínima t é dado por:
$$p' = ne^{\frac{-n(f-t)}{T}} - ne^{\frac{-2n(f-t)}{T}}$$

Calculando a Cobertura da Sequência Alvo

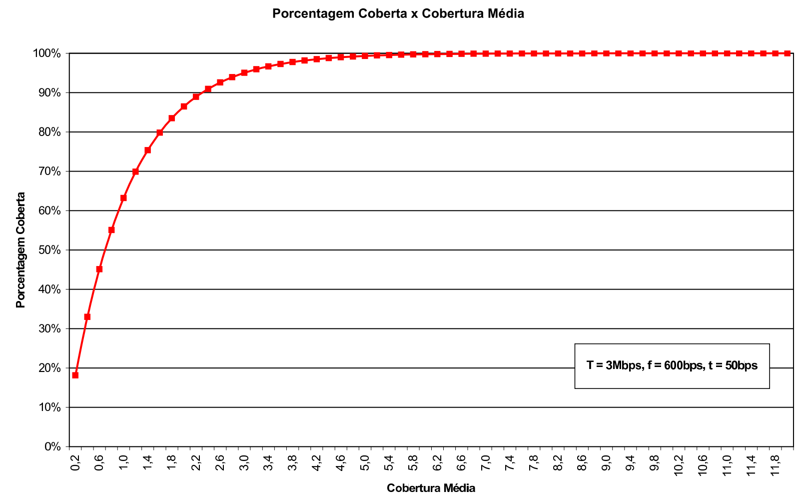
- A fração da sequência alvo coberta por exatamente k fragmentos é dado por:

$$r_k = \frac{e^{-c}c^k}{k!}$$

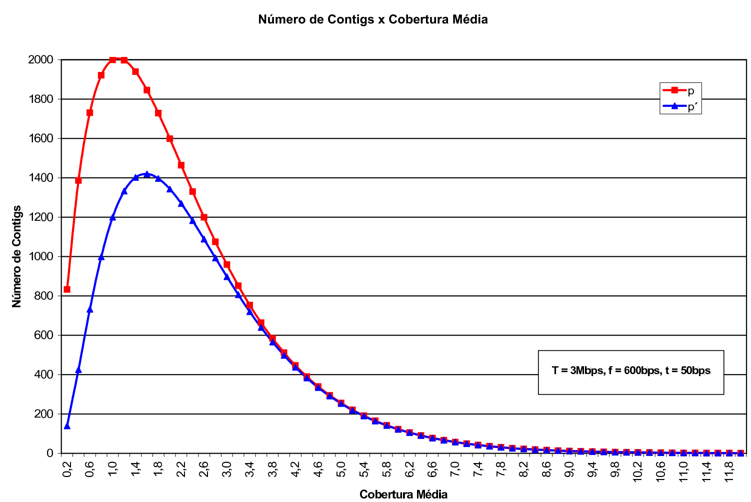
- A fração da sequência alvo coberta por pelo menos um fragmento é dado por:

$$r = 1 - \left(1 - \frac{f}{T}\right)^n$$

Calculando a Cobertura da Sequência Alvo



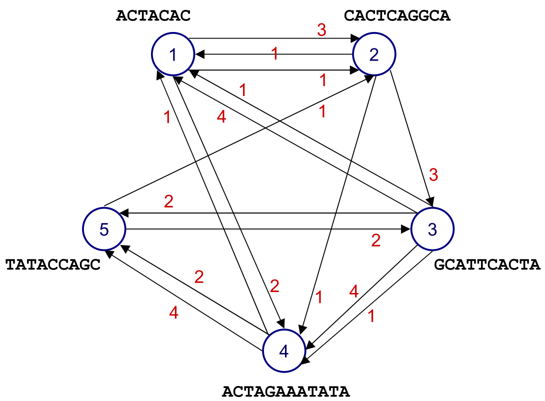
Calculando a Cobertura da Sequência Alvo



Representando Sobreposição de Fragmentos

- O Multigrafo de Sobreposição $\mathcal{OM}(\mathcal{F})$ (ou *Overlap Multigraph*) de uma coleção de fragmentos de sequências $\mathcal{F}$ é um multigrafo orientado e ponderado. O conjunto de vétices V representa cada um dos fragmentos $f \in \mathcal{F}$. Uma aresta entre os vértices a e b ($a \neq b$), com peso $t \geq 0$, existe se o sufixo do fragmento representado por a, com t caracteres, é um prefixo do fragmento representado por b.
- Note que $\mathcal{OM}(\mathcal{F})$ não admite autolaços.
- Podem existir múltiplas arestas entre dois vértices.
- Existe pelo menos uma aresta entre todo par de vértices (com $t = 0$).

Multigrafo de Sobreposição - $\mathcal{OM}(\mathcal{F})$

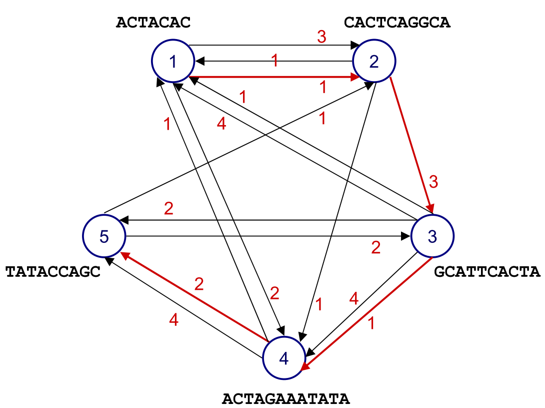


`Sobreposição mínima: t = 1`

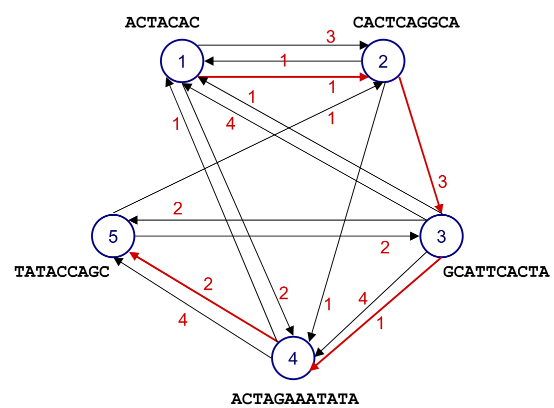
Caminhos e Supersequências

- Caminhos no Multigrafo de Sobreposição $\mathcal{OM}(\mathcal{F})$ representam supersequências envolvendo os fragmentos representados pelos vértices do caminho.
- Seja:
  - P: um caminho em $\mathcal{OM}(\mathcal{F})$.
  - $w(P)$: a soma dos pesos de todas as arestas de P.
  - $\mathcal{F}(P)$: o conjunto de fragmentos representados pelos véties de P.
  - $||\mathcal{F}(P)||$: a soma dos tamanhos de todos os fragmentos de $\mathcal{F}(P)$.
  - $S(P)$: a sequência consenso originada por P.
- A seguinte relação é verdadeira:
  - $||\mathcal{F}(P)|| = w(P) + |S(P)|$
- Obter uma *SCS* para a coleção $\mathcal{F}$, é equivalente a encontrar um caminho de peso máximo que passe por todos os vértices de $\mathcal{OM}(\mathcal{F})$.
- Logo, uma solução para *SCS* pode ser obtida através de um Caminho Hamiltoniano Máximo no multigrafo $\mathcal{OM}(\mathcal{F})$.

Caminhos Hamiltonianos no Multigrafo de Sobreposição

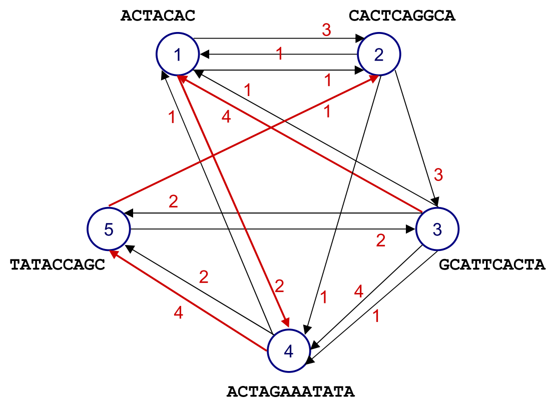


Caminhos Hamiltonianos no Multigrafo de Sobreposição

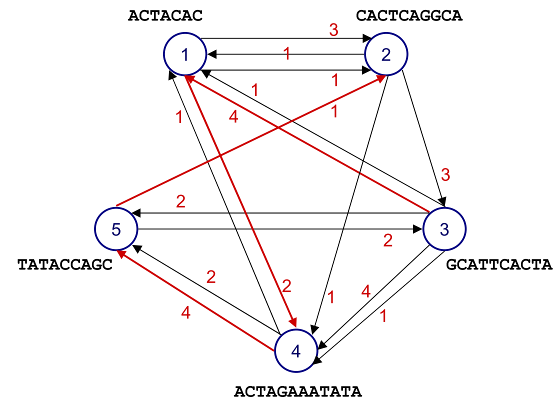


ACTACACACTCAGGCATTCACTACTAGAAATATATACCAGC

Caminhos Hamiltonianos no Multigrafo de Sobreposição

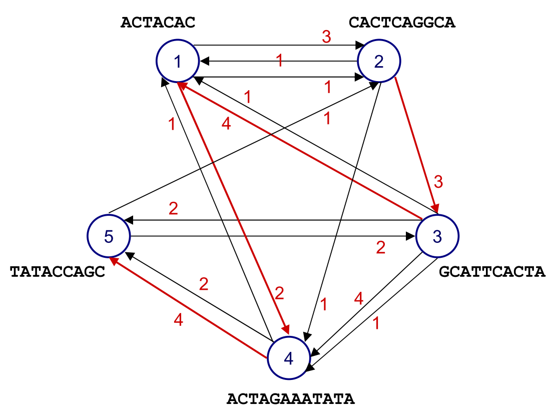


Caminhos Hamiltonianos no Multigrafo de Sobreposição

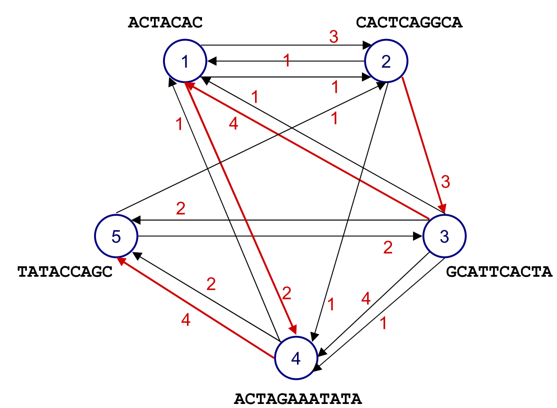


GCATTCACTACACTAGAAATATACCAGCACTCAGGCA

Caminhos Hamiltonianos no Multigrafo de Sobreposição

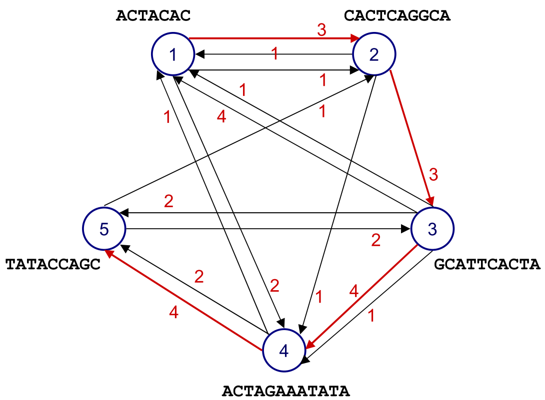


Caminhos Hamiltonianos no Multigrafo de Sobreposição

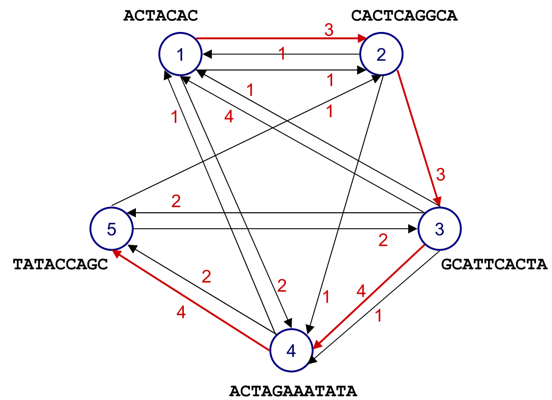


CACTCAGGCATTCACTACACTAGAAATATACCAGC

Caminhos Hamiltonianos no Multigrafo de Sobreposição



Caminhos Hamiltonianos no Multigrafo de Sobreposição

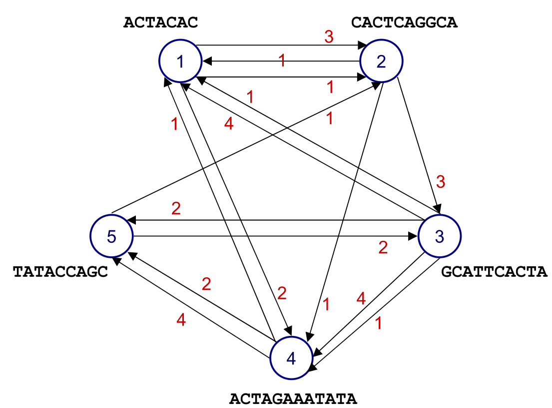


ACTACACTCAGGCATTCACTAGAAATATACCAGC

Algoritmo Guloso para SCS

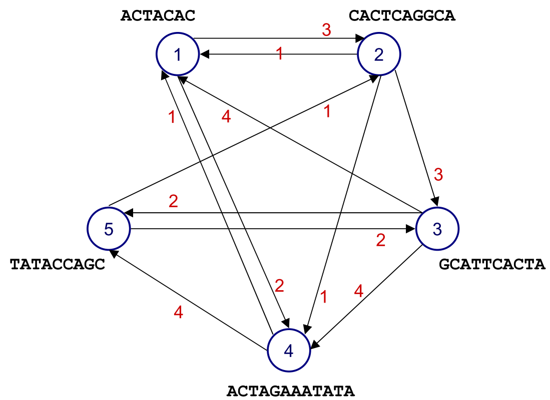
- Neste caso podemos trabalhar com o Grafo de Sobreposição $\mathcal{OG}(\mathcal{F})$ (*Overlap Graph*), que pode ser obtido a partir de $\mathcal{OM}(\mathcal{F})$ mantendo-se apenas a aresta mais pesada entre cada par de vértices.
- Algoritmos gulosos fazem escolhas locais ótimas.
- Para tentar maximizar o peso do caminho a ser montado, o algoritmo, a cada passo, escolhe a aresta válida mais pesada de $\mathcal{OG}(\mathcal{F})$.
- Uma aresta é dita válida se a inclusão dela na solução corrente respeita as seguintes condições:
  - Duas arestas não podem sair de um mesmo vértice.
  - Duas arestas não podem chegar em um mesmo vértice.
  - Nenhum ciclo pode ser formado.
- O algoritmo termina quando o caminho P contiver todos os vértices de $\mathcal{OG}(\mathcal{F})$.

Multigrafo de Sobreposição - $\mathcal{OM}(\mathcal{F})$



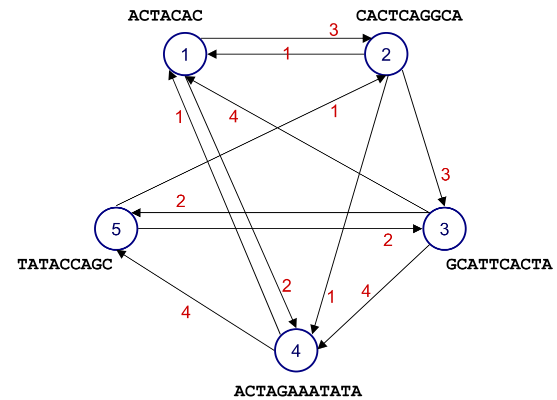
Sobreposição mínima: t = 1

Grafo de Sobreposição - $\mathcal{OG}(\mathcal{F})$



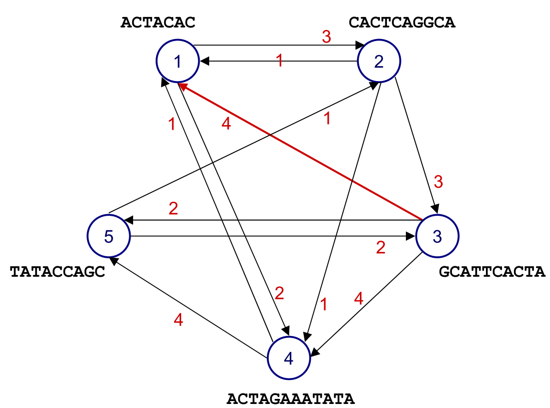
`Sobreposição mínima: t = 1`

Algoritmo Guloso para SCS



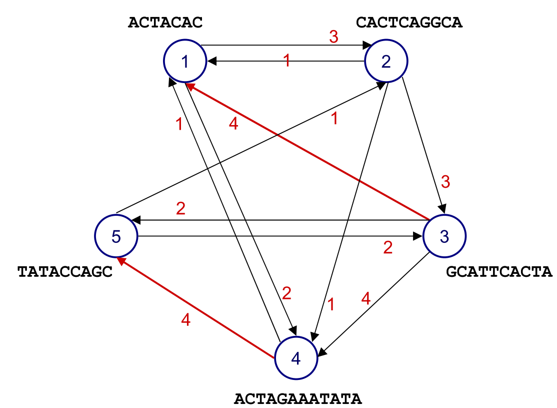
ACTACAC | CACTCAGGCA | GCATTCACTA | ACTAGAAATATA | TATACCAGC

Algoritmo Guloso para SCS



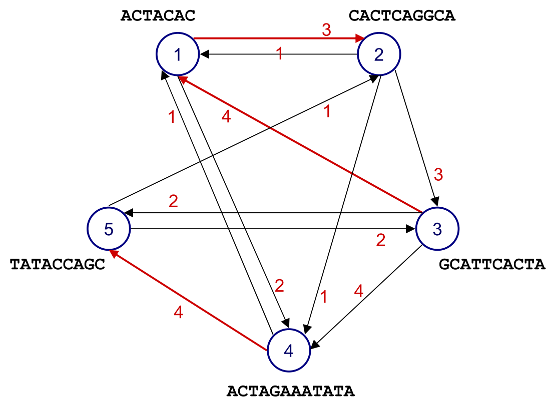
GCATTCACTACAC | CACTCAGGCA | ACTAGAAATATA | TATACCAGC

Algoritmo Guloso para SCS



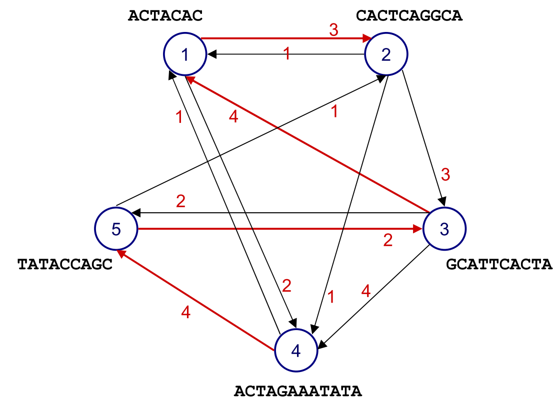
ACTAGAAATATACCAGC | GCATTCACTACAC | CACTCAGGCA

Algoritmo Guloso para SCS



ACTAGAAATATACCAGC | GCATTCACTACACTCAGGCA

Algoritmo Guloso para SCS



ACTAGAAATATACCAGCATTCACTACACTCAGGCA

Algoritmo Guloso para SCS

- Complexidade:
  1. Construir o grafo $\mathcal{OG}(\mathcal{F})$: $O(||\mathcal{F}|| + n^2)$, usando árvores de prefixos.
  2. Ordenar as arestas de $\mathcal{OG}(\mathcal{F})$ em função do peso: $O(n^2 \log n)$.
  3. Para toda aresta, testar se ela é válida: $O(n^2\alpha(n))$, usando conjuntos disjuntos.
  4. Para toda aresta válida encontrada, expandir um caminho: $O(n\alpha(n))$, usando conjuntos disjuntos.
  5. Dado o Caminho Hamiltoniano em $\mathcal{OG}(\mathcal{F})$, construir a SCS: $O(||\mathcal{F}||)$.
  - Total: $O(||\mathcal{F}|| + n^2 \log n)$.
- Algoritmo proposto independentemente por Jorma Tarhio e Esko Ukkonen (1988) e Jonathan Turner (1989).

Conjectura

O algoritmo guloso para SCS é um algoritmo de aproximação com fator 2.

Algoritmos de Aproximação para SCS

- Avrim Blum, Tao Jiang, Ming Li, John Tromp e Mihalis Yannakakis (1994).
  - Provaram que o algoritmo guloso é um algoritmo de aproximação com fator no máximo 4.
  - Apresentaram um algoritmo de aproximação com fator 3.
- Shang-Hua Teng e Frances Yao (1993).
  - Apresentaram um algoritmo de aproximação com fator $2 + 8/9$.
- Artur Czumaj, Leszek Gasieniec, Marek Piotrow e Wojciech Rytter (1994).
  - Apresentaram um algoritmo de aproximação com fator $2 + 5/6$.
- Chris Armen e Clifford Stein (1995).
  - Apresentaram um algoritmo de aproximação com fator $2 + 3/4$.
- Chris Armen e Clifford Stein (1996).
  - Apresentaram um algoritmo de aproximação com fator $2 + 2/3$.
- Elizabeth Sweedyk (1999).
  - Apresentou um algoritmo de aproximação com fator $2 + 1/2$.

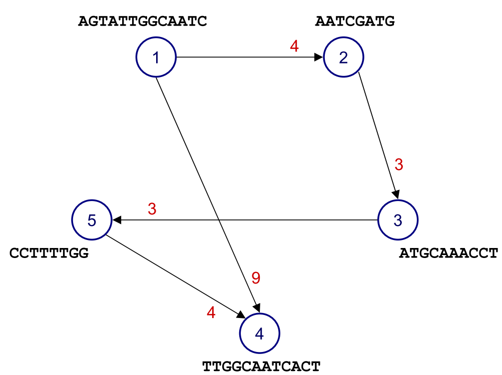
Algoritmo Guloso para SCS

- Como adaptar o algoritmo para lidar como erros de sequenciamento?
- E como lidar com o problema da orientação desconhecida?

Montagem de Fragmentos em Grafos de Sobreposições Acíclicos

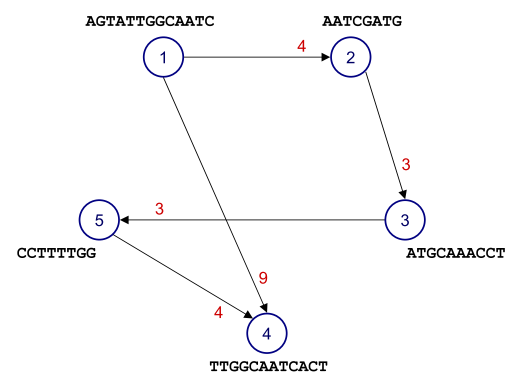
- Considere o grafo o $\mathcal{OG}(\mathcal{F}, t)$, que pode ser construído a partir de $\mathcal{OG}(\mathcal{F})$ removendo as arestas de peso menor do que t.
- Se $\mathcal{OG}(\mathcal{F}, t)$ possui um ciclo orientado, então existe uma repetição de tamanho maior ou igual a t na sequência original.
- Seja $\mathcal{F}$ uma coleção de fragmentos tal que nenhum fragmento esteja completamente contido em outro desta mesma coleção. Se a sequência original (S) for totalmente coberta por um único *contig*, com sobreposição mínima t entre os fragmentos, e sem nenhuma repetição de tamanho maior ou igual a t, então existe um único Caminho Hamiltoniano (P) em $\mathcal{OG}(\mathcal{F}, t)$ e $S = S(P)$.
- Neste caso, o Caminho Hamiltoniano em $\mathcal{OG}(\mathcal{F}, t)$, pode ser obtido através de uma ordenação topológica, em $O(n^2)$.

Montagem de Fragmentos em Grafos de Sobreposições Acíclicos
Algoritmo Guloso



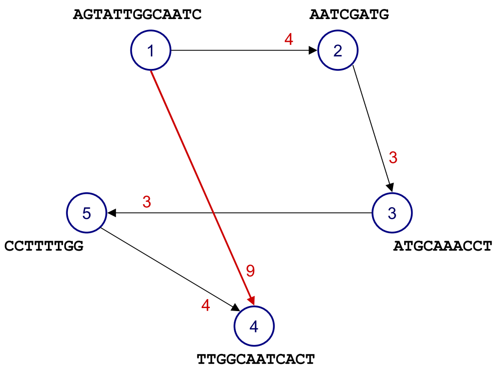
`Sobreposição mínima: t = 3`

Montagem de Fragmentos em Grafos de Sobreposições Acíclicos
Algoritmo Guloso



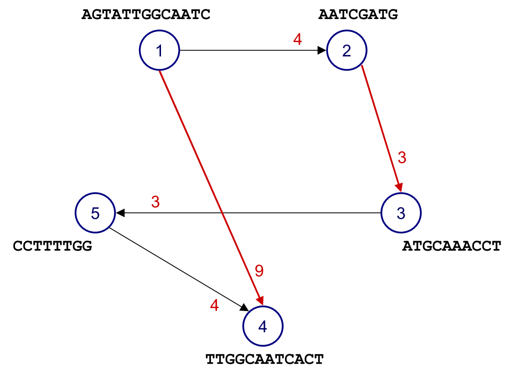
`AGTATTGGCAATC|AATCGATG|ATGCAAACCT|TTGGCAATCACT|CCTTTTGG`

Montagem de Fragmentos em Grafos de Sobreposições Acíclicos
Algoritmo Guloso



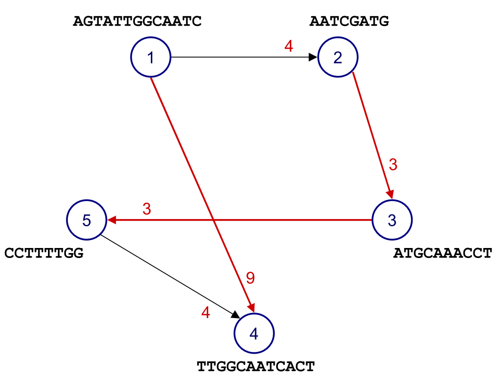
AGTATTGGCAATCACT | AATCGATG | ATGCAAACCT | CCTTTTGG

Montagem de Fragmentos em Grafos de Sobreposições Acíclicos
Algoritmo Guloso



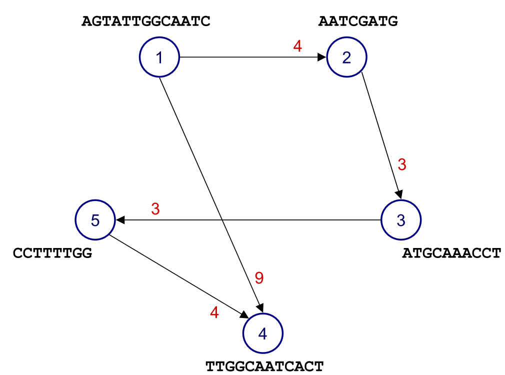
AGTATTGGCAATCACT | AATCGATGCAAACCT | CCTTTTGG

Montagem de Fragmentos em Grafos de Sobreposições Acíclicos
Algoritmo Guloso



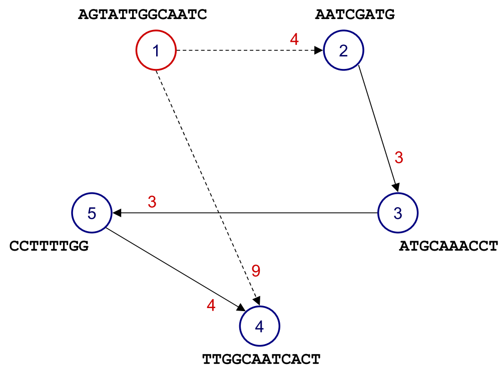
AGTATTGGCAATCACT | AATCGATGCAAACCTTTTGG

Montagem de Fragmentos em Grafos de Sobreposições Acíclicos
Ordenação Topológica



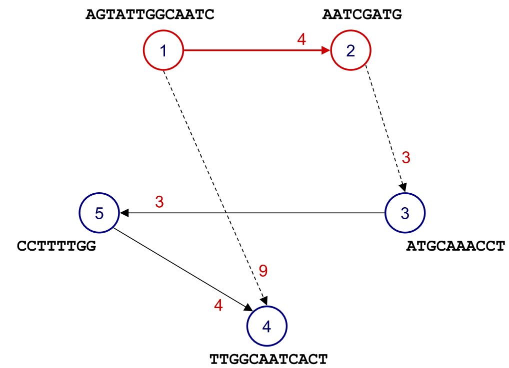
Sobreposição mínima: t = 3

Montagem de Fragmentos em Grafos de Sobreposições Acíclicos
Ordenação Topológica



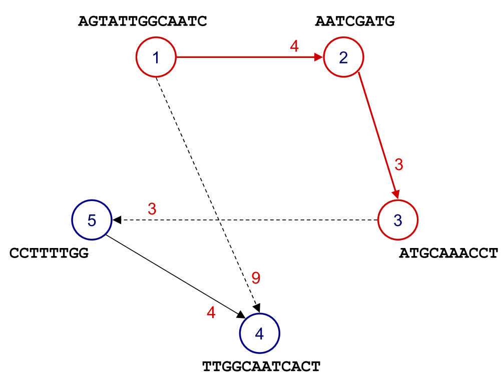
AGTATTGGCAATC

Montagem de Fragmentos em Grafos de Sobreposições Acíclicos
Ordenação Topológica



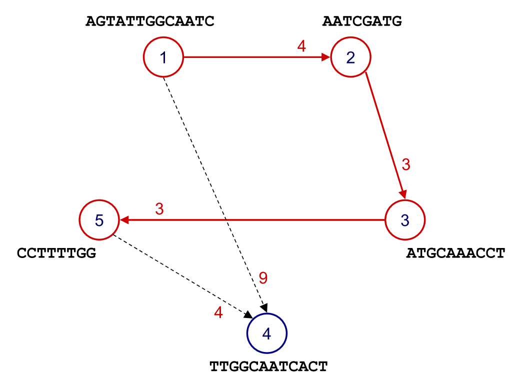
AGTATTGGCAATCGATG

Montagem de Fragmentos em Grafos de Sobreposições Acíclicos
Ordenação Topológica



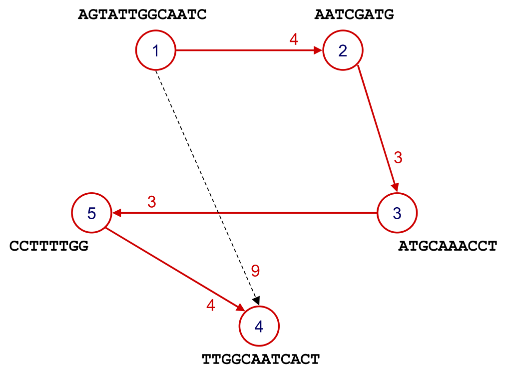
AGTATTGGCAATCGATGCAAACCT

Montagem de Fragmentos em Grafos de Sobreposições Acíclicos
Ordenação Topológica



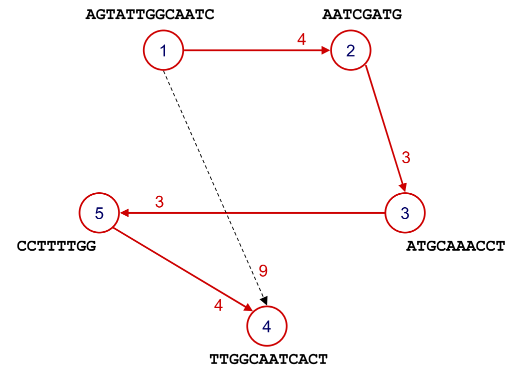
AGTATTGGCAATCGATGCAAACCTTTTGG

Montagem de Fragmentos em Grafos de Sobreposições Acíclicos
Ordenação Topológica



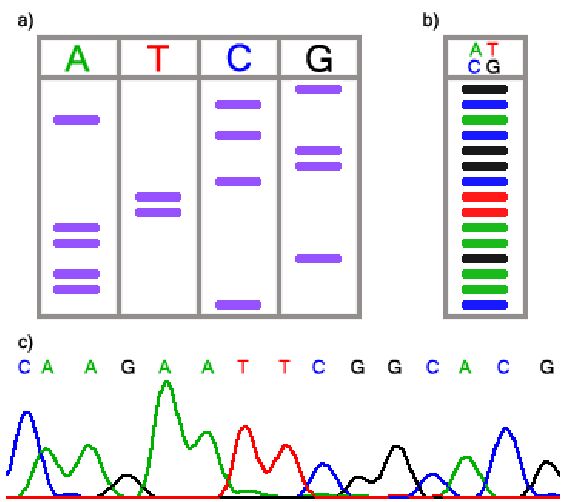
AGTATTGGCAATCGATGCAAACCTTTTGGCAATCACT

Montagem de Fragmentos em Grafos de Sobreposições Acíclicos
Ordenação Topológica

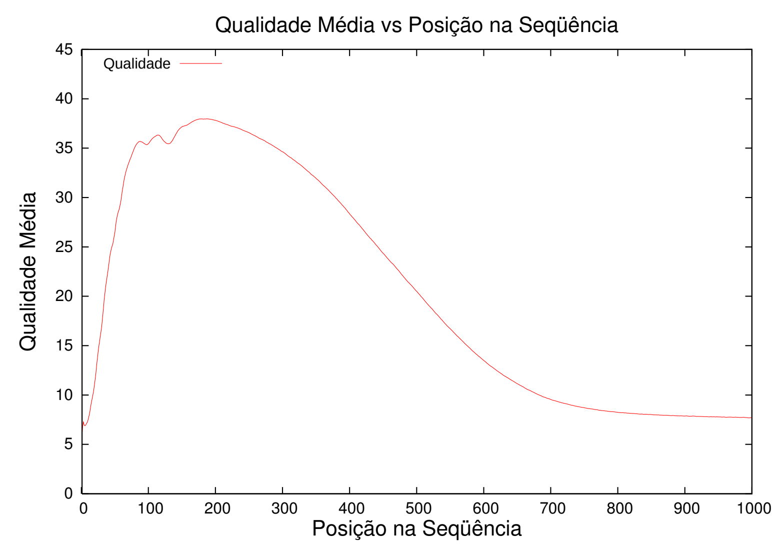


AGTATTGGCAATCGATGCAAACCTTTTGGCAATCACT

Base-Calling



Base-Calling: SUCEST



Phred

- Ferramenta de *base-calling* produzida por Phil Green, Brent Ewing, LaDeana Hillier e Michael Wendl (1998).
- O método é composto por 4 fases:
  - Predição das localizações dos picos.
  - Identificação dos picos observados.
  - Comparação entre os picos previstos e observados.
  - Verificação dos picos observados que não são compatíveis com os picos previstos.
- Phred associa um valor de qualidade para cada base da sequência lida:

$$Q = -10 \times \log_{10} P_e$$

  onde P_e é a probabilidade da base estar errada.
- Exemplo:
  - $Q = 10 \Longrightarrow P_e = 10\%$
  - $Q = 20 \Longrightarrow P_e = 1\%$
  - $Q = 30 \Longrightarrow P_e = 0.1\%$
- Phred pode ser usado para remover pontas de baixas qualidades.

Phrap

- Ferramenta de montagem de sequências produzida por Phil Green (1998).
- Principais características:
  - Usa a sequência inteira, não apenas os trechos de alta qualidade.
  - Usa a qualidade das sequências para obter uma montagem de alta qualidade.
  - Constrói os consensos dos contigs como um mosaico das partes de mais alta qualidade das sequências.
  - Atribui valores de qualidade para as sequências consenso.
  - Faz comparação entre as sequências usando uma variação do algoritmo de Smith-Waterman, onde as comparações são iniciadas apenas se existir um trecho idêntico de tamanho mínimo (por padrão 30), em ambas as sequências. A extensão do alinhamento é realizada usando apenas uma faixa restrita da matriz de Programação Dinâmica (por padrão, faixa de tamanho 14).

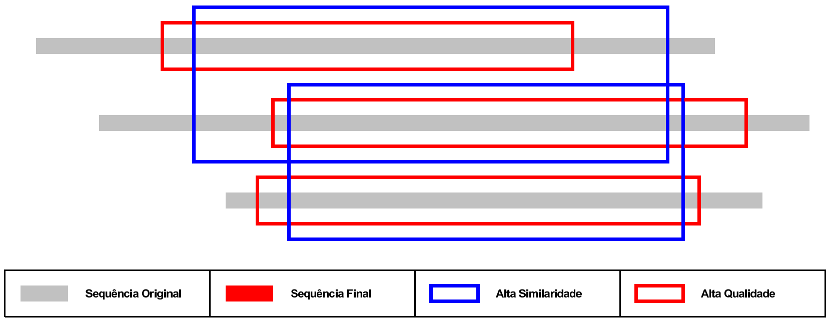
Consed

- Ferramenta de visualização e edição de montagens de sequências, com suporte a "fechamento" de montagem, desenvolvida por David Gordon, Chris Abajian e Phil Green (1998).
- Desenvolvido originalmente para dar suporte apenas ao Phrap.
- Hoje suporta uma vasta gama de montadores (que produzem arquivos no formato `ace`, lidos pelo Consed), inclusive os montadores desenvolvidos para as novas tecnologias 454 e Solexa (de sequências curtas e muitas curtas).

CAP3

- Ferramenta de montagem de sequências produzida por Xiaoqiu Huang e Anup Madan (1999).
- Passos principais:
  - Remoção das extremidades de baixa qualidade.
  - Identificação das sobreposição entre as sequências.
  - Remoção das falsas sobreposições.
  - Construção dos *contigs*.
  - Alinhamento múltiplo e geração da sequência consenso, considerando as somas das qualidades das bases de cada coluna.
- Sobreposições identificadas em duas fases:
  1. Alinhamento local ponderado restrito a uma faixa de tamanho k:
     - ★ $Match' = Match \times min(q_1, q_2)$
     - ★ $Mismatch' = Mismatch \times min(q_1, q_2)$
     - ★ $Gap' = Gap \times min(q_1, q_2)$
  2. Alinhamento Global, restrito a uma faixa de tamanho $2k$, centralizado na posição inicial do alinhamento local ótimo calculado previamente.
- Geralmente produz *contigs* mais curtos, porém de maior qualidade, quando comparados com os *contigs* gerados pelo Phrap.

CAP3 - Remoção de Extremidades de Baixa Qualidades



CAP3 - Remoção de Extremidades de Baixa Qualidades