CSE 566 Spring 2023

Whole-Genome Comparison

Instructor: Mingfu Shao

Sequence Comparison

Sequence Comparison

- Orthologous genes may have different sequences.

  Seq1: **A C G T A G A C C G C T A C A G**

  Seq2: **T C C T A G A T C C G T T C G**

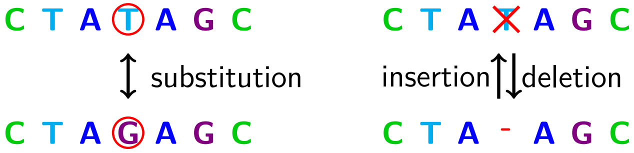
Sequence Comparison

- Orthologous genes may have different sequences.

Seq1: **A C G T A G A C C G C T A C A G**

Seq2: **T C C T A G A T C C G T T C G**

- Point mutations
  - Nucleotide substitutions
  - Nucleotide indels



Pairwise Sequence Comparison

- **Edit distance:** the **minimum** number of substitutions and indels that can transform one sequence into the other.

Pairwise Sequence Comparison

- **Edit distance:** the **minimum** number of substitutions and indels that can transform one sequence into the other.

Seq1: **A C G T A G A C C G C T A C A G**

Seq2: **T C C T A G A T C C G T T C G**

Pairwise Sequence Comparison

- **Edit distance:** the **minimum** number of substitutions and indels that can transform one sequence into the other.

Seq1: A C G T A G A C C G C T A C A G

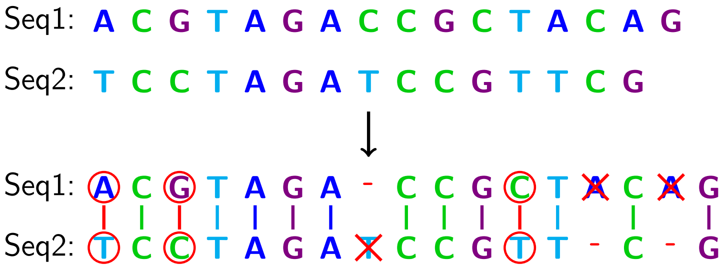
Seq2: T C C T A G A T C C G T T C G

$$\downarrow$$

$$
\begin{array}{lccccccccccccccccc}
\text{Seq1:} & \textcircled{A} & C & \textcircled{G} & T & A & G & A & - & C & C & G & \textcircled{C} & T & \cancel{A} & C & \cancel{A} & G \\
 & | & | & | & | & | & | & | & & | & | & | & | & | & & | & & | \\
\text{Seq2:} & \textcircled{T} & C & \textcircled{C} & T & A & G & A & \cancel{T} & C & C & G & \textcircled{T} & T & - & C & - & G
\end{array}
$$

Pairwise Sequence Comparison

- **Edit distance:** the **minimum** number of substitutions and indels that can transform one sequence into the other.



Seq1: A C G T A G A C C G C T A C A G
Seq2: T C C T A G A T C C G T T C G
Seq1: A C G T A G A - C C G C T A C A G
Seq2: T C C T A G A T C C G T T - C - G

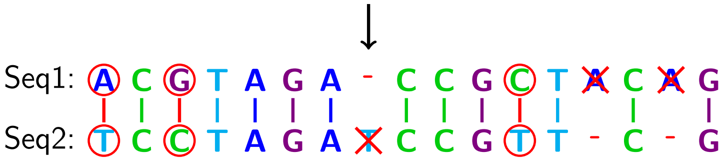
- Determines the **distance/similarity** and the **one-to-one correspondence** between the nucleotides.

Pairwise Sequence Comparison

- **Edit distance:** the **minimum** number of substitutions and indels that can transform one sequence into the other.

Seq1: A C G T A G A C C G C T A C A G

Seq2: T C C T A G A T C C G T T C G



Seq1: A C G T A G A - C C G C T A C A G
Seq2: T C C T A G A T C C G T T - C - G

- Determines the **distance/similarity** and the **one-to-one correspondence** between the nucleotides.

- **Efficient algorithm:** dynamic programming

Multiple Sequence Comparison

Multiple Sequence Comparison

Seq1: **A C G C A G A C C G T A C A**

Seq2: **A C G T A G A C C G C T A C A G**

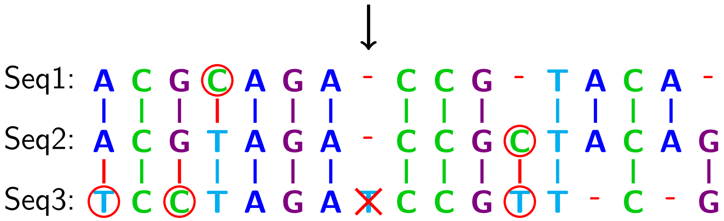
Seq3: **T C C T A G A T C C G T T C G**

Multiple Sequence Comparison

Seq1: A C G C A G A C C G T A C A

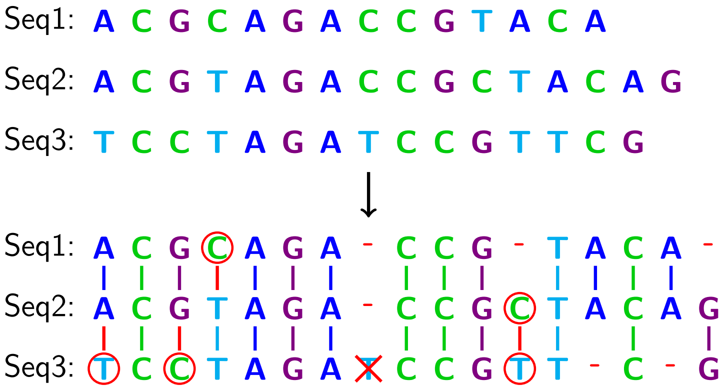
Seq2: A C G T A G A C C G C T A C A G

Seq3: T C C T A G A T C C G T T C G



Seq1: A C G C A G A - C C G - T A C A -
Seq2: A C G T A G A - C C G C T A C A G
Seq3: T C C T A G A T C C G T T - C - G

Multiple Sequence Comparison


Seq1: A C G C A G A C C G T A C A
Seq2: A C G T A G A C C G C T A C A G
Seq3: T C C T A G A T C C G T T C G
Seq1: A C G C A G A - C C G - T A C A -
Seq2: A C G T A G A - C C G C T A C A G
Seq3: T C C T A G A T C C G T T - C - G

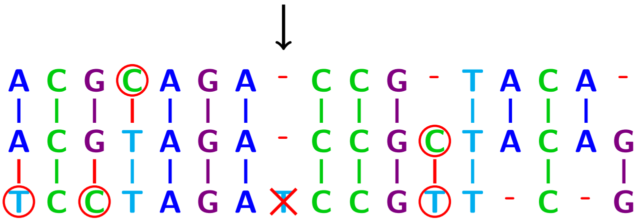
- **NP-hard** for almost all formulations.

Multiple Sequence Comparison

Seq1: **A C G C A G A C C G T A C A**

Seq2: **A C G T A G A C C G C T A C A G**

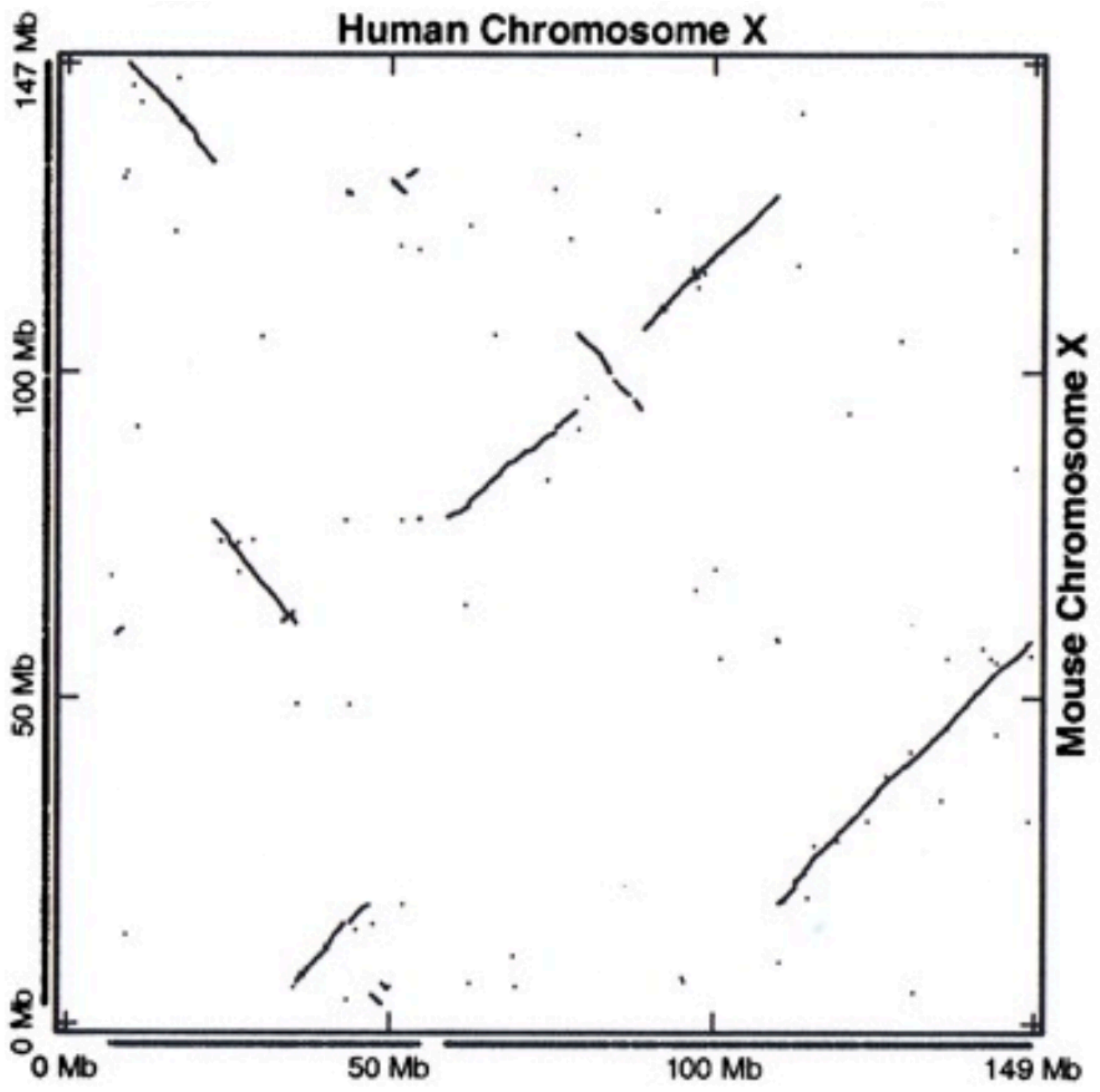
Seq3: **T C C T A G A T C C G T T C G**

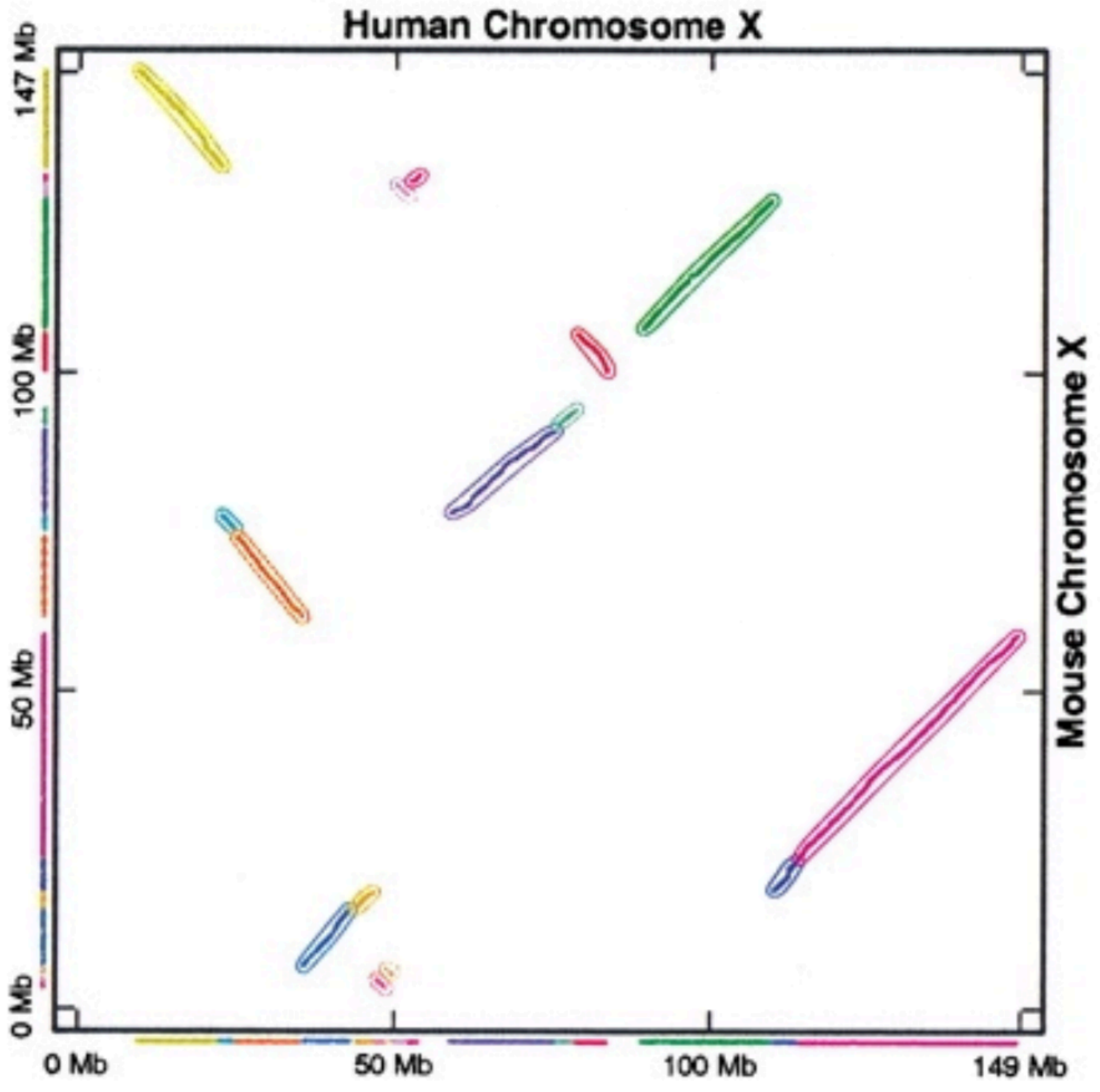


Seq1: A C G C A G A - C C G - T A C A -
Seq2: A C G T A G A - C C G C T A C A G
Seq3: T C C T A G A T C C G T T - C - G

- **NP-hard** for almost all formulations.
- **Heuristics:** progressive methods, iterative algorithms.

Comparing Whole-Genome



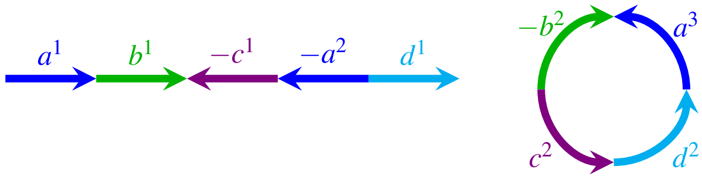


Model of Whole-genomes

- **Genome:** a set of chromosomes.
- **Chromosome:** a linear/circular list of genes (synteny blocks).

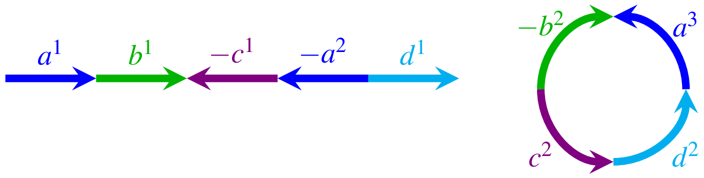
Model of Whole-genomes

- **Genome:** a set of chromosomes.
- **Chromosome:** a linear/circular list of genes (synteny blocks).



Model of Whole-genomes

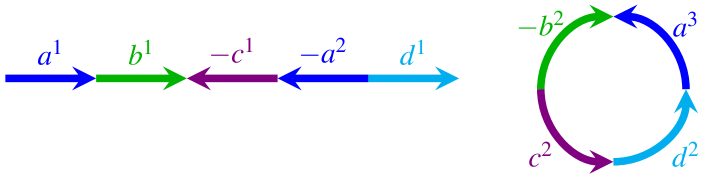
- **Genome:** a set of chromosomes.
- **Chromosome:** a linear/circular list of genes (synteny blocks).



- Each gene has a **sign**, indicating its transcriptional direction.

Model of Whole-genomes

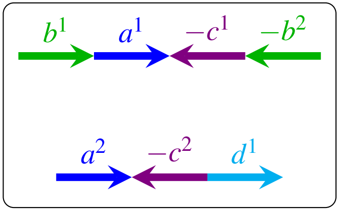
- **Genome:** a set of chromosomes.
- **Chromosome:** a linear/circular list of genes (synteny blocks).



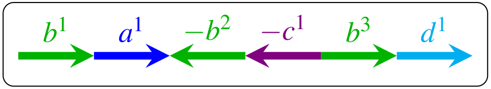
- Each gene has a **sign**, indicating its transcriptional direction.
- All genes are grouped into **gene families**.

Whole-genome Comparison

Whole-genome Comparison

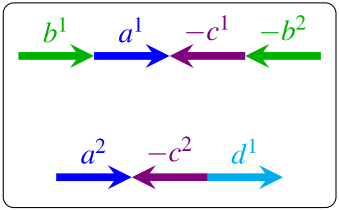


(G_1)

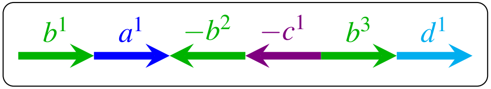


(G_2)

Whole-genome Comparison



(G_1)



(G_2)

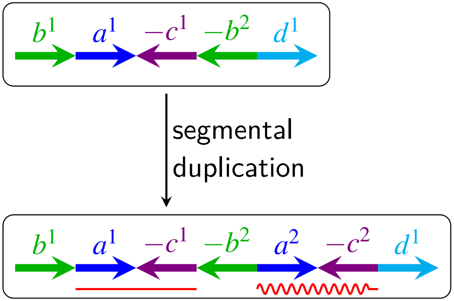
- Different copy numbers for some gene families.
- Different number of chromosomes.
- Different gene orders and orientations.

Large-scale Events

- **Content-modifying** events: change copy numbers
  - Segmental duplication
  - Tandem duplication
  - Lateral gene transfer
  - Gene loss

Large-scale Events

- **Content-modifying** events: change copy numbers
  - Segmental duplication
  - Tandem duplication
  - Lateral gene transfer
  - Gene loss

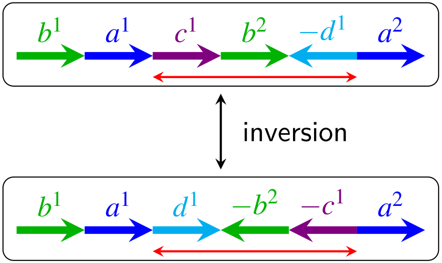

b^1 a^1 $-c^1$ $-b^2$ d^1
segmental duplication
b^1 a^1 $-c^1$ $-b^2$ a^2 $-c^2$ d^1

Large-scale Events

- **Rearrangements**: change number of chromosomes, gene orders and orientations.
  - Inversion
  - Translocation
  - Chromosomal fission
  - Chromosomal fusion

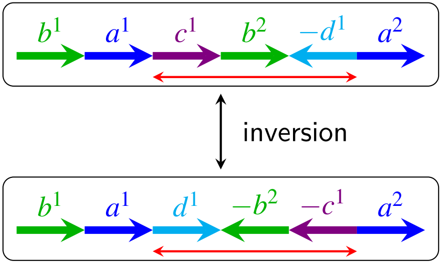
Large-scale Events

- **Rearrangements**: change number of chromosomes, gene orders and orientations.
  - Inversion
  - Translocation
  - Chromosomal fission
  - Chromosomal fusion



Large-scale Events

- **Rearrangements**: change number of chromosomes, gene orders and orientations.
  - Inversion
  - Translocation
  - Chromosomal fission
  - Chromosomal fusion

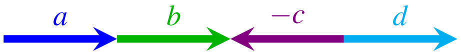

b^1 a^1 c^1 b^2 $-d^1$ a^2
inversion
b^1 a^1 d^1 $-b^2$ $-c^1$ a^2

- Almost all rearrangements can be represented by the universal **DCJ** (double-cut-and-join) operation.

DCJ (double-cut-and-join) Operation

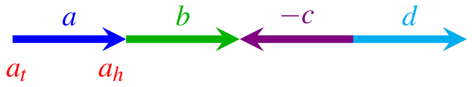
DCJ (double-cut-and-join) Operation

- Some definitions



DCJ (double-cut-and-join) Operation

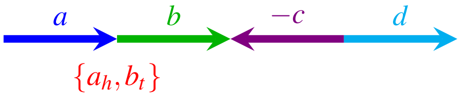
- Some definitions



  - **Extremities:** two ends (**head** and **tail**) of a gene

DCJ (double-cut-and-join) Operation

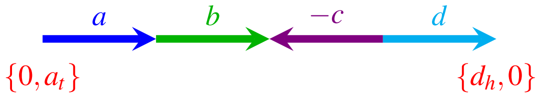
- Some definitions



  - **Extremities:** two ends (**head** and **tail**) of a gene
  - **Adjacency:** two consecutive extremities

DCJ (double-cut-and-join) Operation

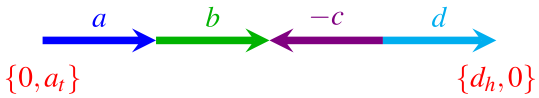
- Some definitions



  - **Extremities:** two ends (**head** and **tail**) of a gene
  - **Adjacency:** two consecutive extremities
  - **Null extremity:** special extremity 0 added to each end of the linear chromosomes

DCJ (double-cut-and-join) Operation

- Some definitions

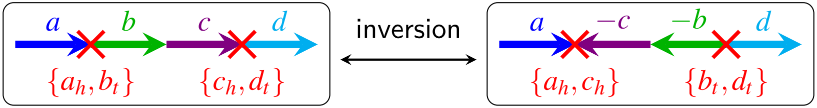


  - **Extremities:** two ends (**head** and **tail**) of a gene
  - **Adjacency:** two consecutive extremities
  - **Null extremity:** special extremity 0 added to each end of the linear chromosomes

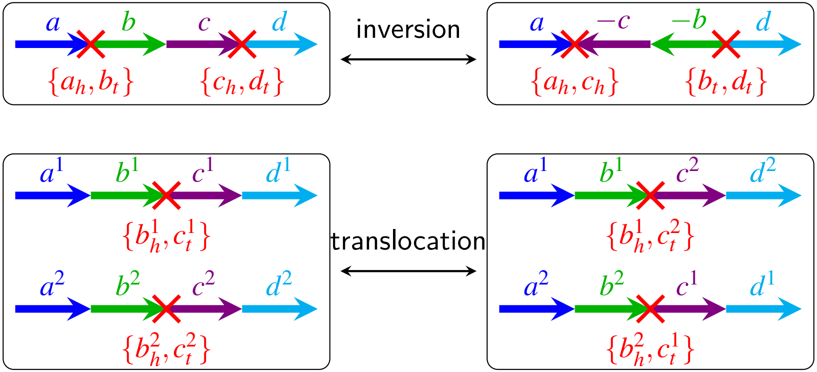
- DCJ operation: $\{p,q\}+\{r,s\} \Longrightarrow \{p,r\}+\{q,s\}$

Various Cases of DCJ Operation

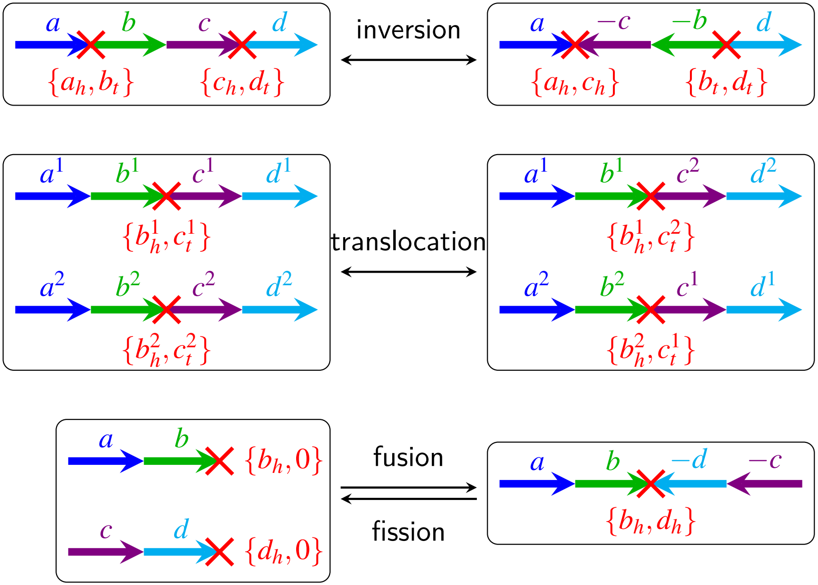
Various Cases of DCJ Operation



Various Cases of DCJ Operation



Various Cases of DCJ Operation

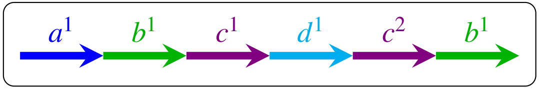


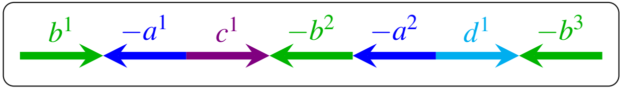
Pairwise Whole-genome Comparison

- **Edit distance:** given two genomes G_1 and G_2, to compute the **minimum** number events (specified by an evolutionary model) that can transform G_1 into G_2.

Pairwise Whole-genome Comparison

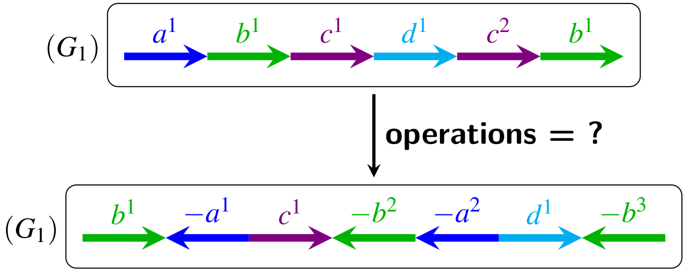
- **Edit distance:** given two genomes G_1 and G_2, to compute the **minimum** number events (specified by an evolutionary model) that can transform G_1 into G_2.

(G_1)


(G_1)


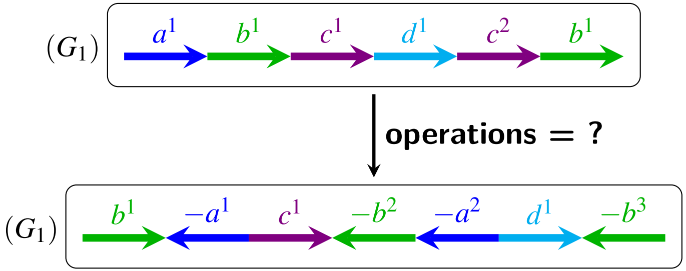
Pairwise Whole-genome Comparison

- **Edit distance:** given two genomes G_1 and G_2, to compute the **minimum** number events (specified by an evolutionary model) that can transform G_1 into G_2.



Pairwise Whole-genome Comparison

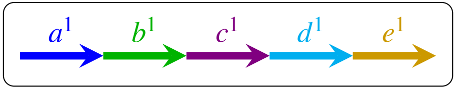
- **Edit distance:** given two genomes G_1 and G_2, to compute the **minimum** number events (specified by an evolutionary model) that can transform G_1 into G_2.



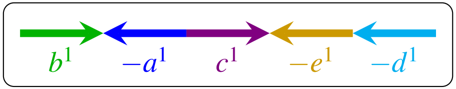
- Diversity of the edit distance problems
  - Assumptions on the given genomes.
  - Which types of events are in the evolutionary model.

Edit Distance (without Duplicate Genes)

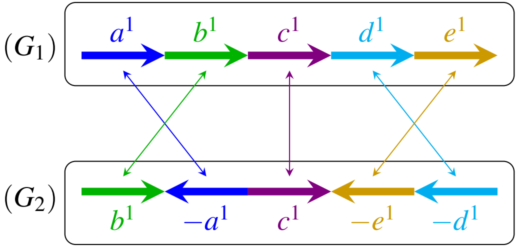
(G_1)



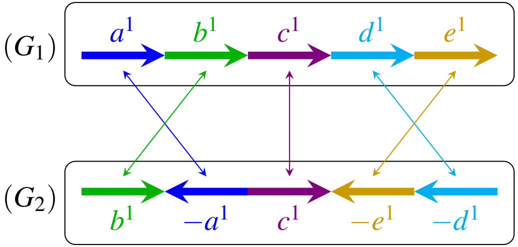
(G_2)



Edit Distance (without Duplicate Genes)

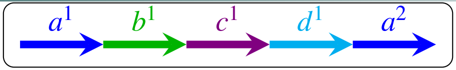


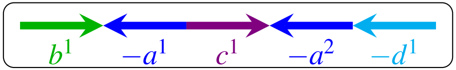
Edit Distance (without Duplicate Genes)



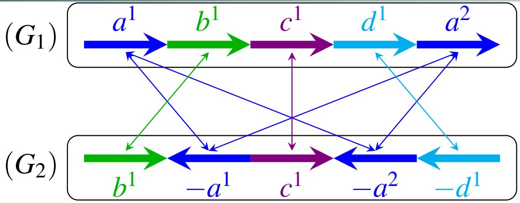
- Edit distances that can be computed in **linear time**:
  - **Inversion distance** (*Hannenhalli and Pevzner, 1995; Bader et al., 2001*)
  - **DCJ distance** (*Bergeron et al., 2006*)
  - **DCJ + Insertion + Deletion** (*Braga et al., 2010; Compeau et al., 2013*)

Edit Distance (with Duplicate Genes)

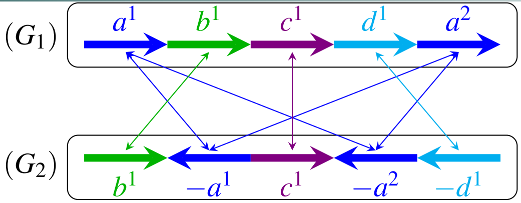
(G_1)


(G_2)


Edit Distance (with Duplicate Genes)

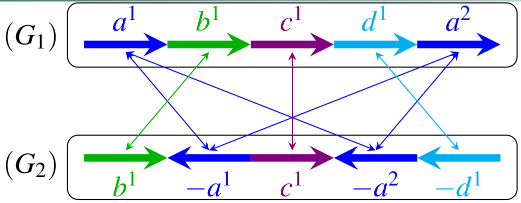


Edit Distance (with Duplicate Genes)



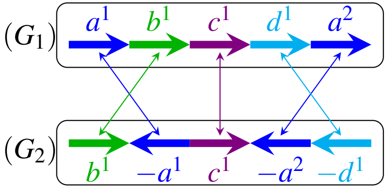
- **Matching:** To build a matching between homologous genes, such that a certain distance is minimized.

Edit Distance (with Duplicate Genes)



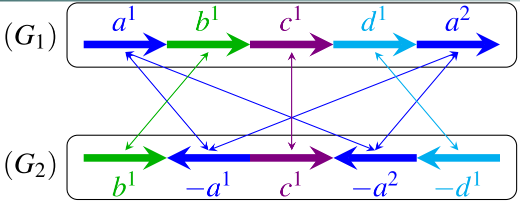
(G_1)
a^1 b^1 c^1 d^1 a^2
(G_2)
b^1 $-a^1$ c^1 $-a^2$ $-d^1$

- **Matching:** To build a matching between homologous genes, such that a certain distance is minimized.

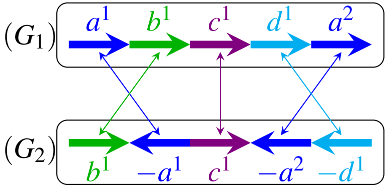


(G_1)
a^1 b^1 c^1 d^1 a^2
(G_2)
b^1 $-a^1$ c^1 $-a^2$ $-d^1$

Edit Distance (with Duplicate Genes)

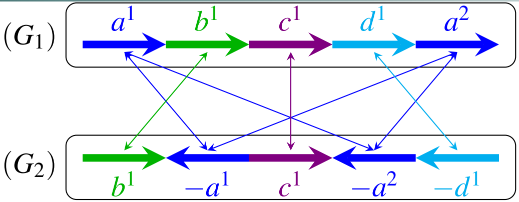


- **Matching:** To build a matching between homologous genes, such that a certain distance is minimized.

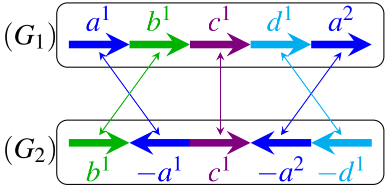


- **Exemplar (*Sankoff, 1999*):** To select one gene in each family, such that a certain distance is minimized.

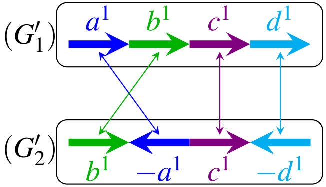
Edit Distance (with Duplicate Genes)



- **Matching:** To build a matching between homologous genes, such that a certain distance is minimized.



- **Exemplar (*Sankoff, 1999*):** To select one gene in each family, such that a certain distance is minimized.



Edit Distance (with Duplicate Genes)

- Most of the formulations are **NP**-hard.

Edit Distance (with Duplicate Genes)

- Most of the formulations are **NP**-hard.
- Under the **matching** strategy:
  - **Inversion distance:** Heuristics (SOAR, *Chen et al., 2005*)
  - **Inversion + Single-gene Duplication:** Heuristics (MSOAR, *Fu et al., 2007*)

Edit Distance (with Duplicate Genes)

- Most of the formulations are **NP**-hard.
- Under the **matching** strategy:
  - **Inversion distance:** Heuristics (SOAR, *Chen et al., 2005*)
  - **Inversion + Single-gene Duplication:** Heuristics (MSOAR, *Fu et al., 2007*)
- Under the **exemplar** strategy:
  - **Inversion distance:** Branch-and-Bound (*Sankoff et al., 1999*)
  - **Breakpoint distance:** Divide-and-Conquer (*Nguyen et al., 2005*)

Multiple Whole-genome Comparison

- **Median distance problem:** Given k genomes $G_1, G_2, \cdots, G_k$, to compute a *median* genome G_0 such that $\sum_{i=1}^{k} d(G_0, G_i)$ is minimized.

Multiple Whole-genome Comparison

- **Median distance problem:** Given k genomes $G_1, G_2, \cdots, G_k$, to compute a *median* genome G_0 such that $\sum_{i=1}^{k} d(G_0, G_i)$ is minimized.
- The median problem has been widely studied and applied on multi-genome comparison and phylogeny construction (*Moret et al., 2002, Eriksen, 2007, Adam et al., 2008*).

Multiple Whole-genome Comparison

- **Median distance problem:** Given k genomes $G_1, G_2, \cdots, G_k$, to compute a *median* genome G_0 such that $\sum_{i=1}^{k} d(G_0, G_i)$ is minimized.
- The median problem has been widely studied and applied on multi-genome comparison and phylogeny construction (*Moret et al., 2002, Eriksen, 2007, Adam et al., 2008*).
- Most of the formulations are **NP**-hard.

Multiple Whole-genome Comparison

- **Median distance problem:** Given k genomes $G_1, G_2, \cdots, G_k$, to compute a *median* genome G_0 such that $\sum_{i=1}^{k} d(G_0, G_i)$ is minimized.
- The median problem has been widely studied and applied on multi-genome comparison and phylogeny construction (*Moret et al., 2002, Eriksen, 2007, Adam et al., 2008*).
- Most of the formulations are **NP**-hard.
- **Inversion median distance:** Branch-and-Bound (*Siepel and Moret, 2001*), Heuristics (*Rajan et al., 2010*)

Multiple Whole-genome Comparison

- **Median distance problem:** Given k genomes $G_1, G_2, \cdots, G_k$, to compute a *median* genome G_0 such that $\sum_{i=1}^{k} d(G_0, G_i)$ is minimized.
- The median problem has been widely studied and applied on multi-genome comparison and phylogeny construction (*Moret et al., 2002, Eriksen, 2007, Adam et al., 2008*).
- Most of the formulations are **NP**-hard.
- **Inversion median distance:** Branch-and-Bound (*Siepel and Moret, 2001*), Heuristics (*Rajan et al., 2010*)
- **DCJ median distance:** Branch-and-Bound (*Zhang et al, 2009*), Decomposition scheme (*Xu and Sankoff, 2008*)

Importance of these Problems

Importance of these Problems

1. Edit distance and median distance reflect a basic and global evolutionary relationship among species. They can be applied to reconstruct phylogenetic trees.

Importance of these Problems

1. Edit distance and median distance reflect a basic and global evolutionary relationship among species. They can be applied to reconstruct phylogenetic trees.
2. After clarifying the evolutionary events in the history, orthologs and paralogs can be naturally inferred.

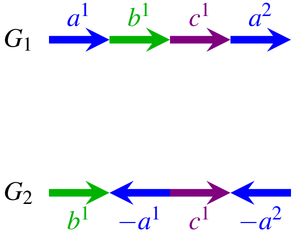
Importance of these Problems

1. Edit distance and median distance reflect a basic and global evolutionary relationship among species. They can be applied to reconstruct phylogenetic trees.
2. After clarifying the evolutionary events in the history, orthologs and paralogs can be naturally inferred.
3. Systematically and automatically annotating genomes can be achieved through genome comparison.

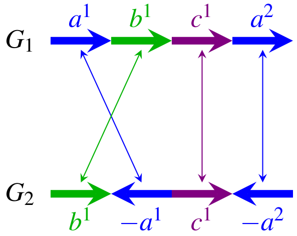
Importance of these Problems

1. Edit distance and median distance reflect a basic and global evolutionary relationship among species. They can be applied to reconstruct phylogenetic trees.
2. After clarifying the evolutionary events in the history, orthologs and paralogs can be naturally inferred.
3. Systematically and automatically annotating genomes can be achieved through genome comparison.
4. These problems are interesting for computer scientists, since they usually have very special structures.

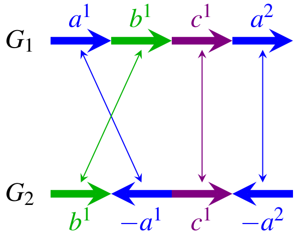
DCJ Distance (without Duplicate Genes)



DCJ Distance (without Duplicate Genes)

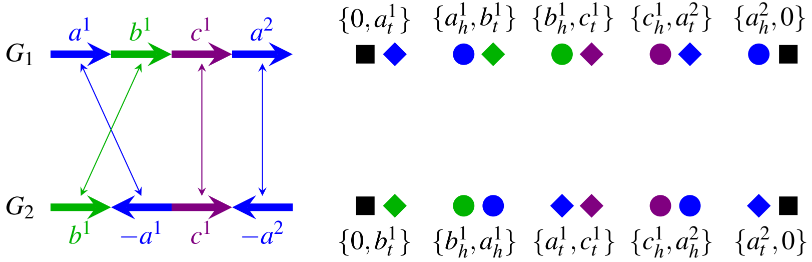


DCJ Distance (without Duplicate Genes)



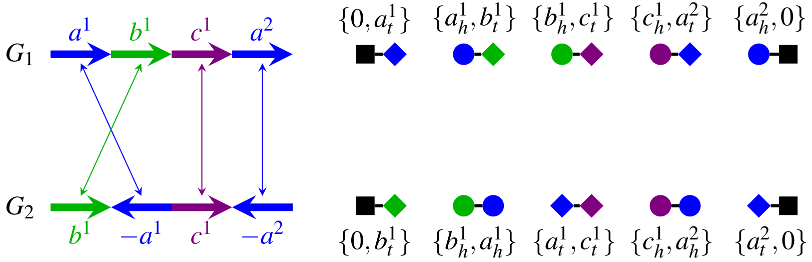
- Adjacency Graph

DCJ Distance (without Duplicate Genes)



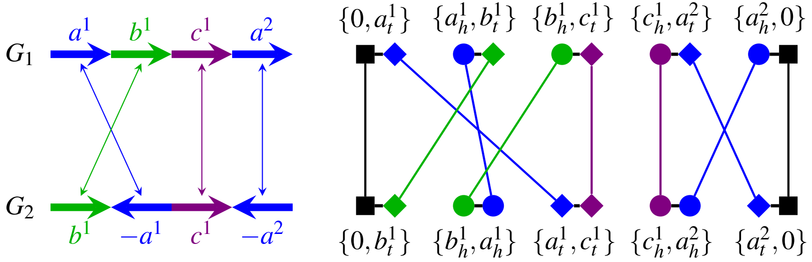
- Adjacency Graph

DCJ Distance (without Duplicate Genes)



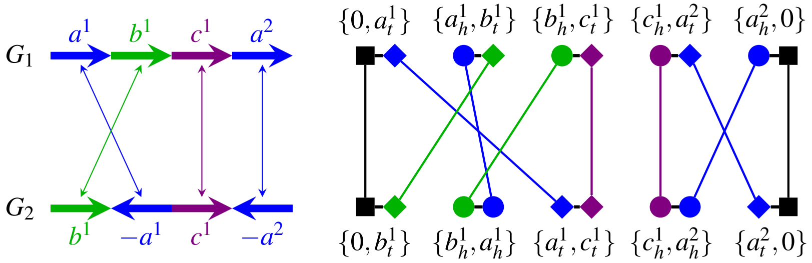
- Adjacency Graph

DCJ Distance (without Duplicate Genes)



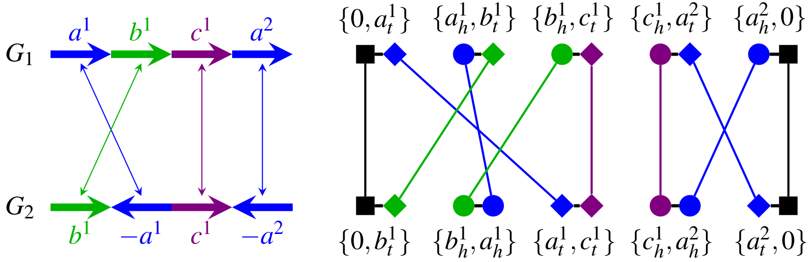
- Adjacency Graph

DCJ Distance (without Duplicate Genes)



- Adjacency Graph
- The adjacency graph consists of vertex-disjoint cycles.

DCJ Distance (without Duplicate Genes)



- Adjacency Graph
- The adjacency graph consists of vertex-disjoint cycles.
- DCJ distance: $d(B) = \#$(adjacencies) - $\#$(cycles).

DCJ Distance (with Duplicate Genes)

- **Problem:** to compute a **matching** between duplicate genes, such that the number of cycles in the corresponding adjacency graph induced by this matching is maximized.

DCJ Distance (with Duplicate Genes)

- **Problem:** to compute a **matching** between duplicate genes, such that the number of cycles in the corresponding adjacency graph induced by this matching is maximized.
- This problem is NP-hard.

DCJ Distance (with Duplicate Genes)

- **Problem:** to compute a **matching** between duplicate genes, such that the number of cycles in the corresponding adjacency graph induced by this matching is maximized.
- This problem is NP-hard.
- Previous work:
  - Heuristics: (*Chen et al., 2005, Suksawatchon et al., 2007*)
  - Approximations: (*Marron et al., 2003, Shao et al., 2012*)

DCJ Distance (with Duplicate Genes)

- **Problem:** to compute a **matching** between duplicate genes, such that the number of cycles in the corresponding adjacency graph induced by this matching is maximized.
- This problem is NP-hard.
- Previous work:
  - Heuristics: (*Chen et al., 2005, Suksawatchon et al., 2007*)
  - Approximations: (*Marron et al., 2003, Shao et al., 2012*)
- A fast and exact algorithm.
  - An ILP formulation that gives the optimal solution.
  - Efficient Algorithms to identify optimal substructures.

Integer Linear Program (ILP)

- Mathematical Formulation

$$\begin{aligned} \max \quad & c^T x && (\textbf{Objective function}) \\ s.t. \quad & Ax \leq b && (\textbf{Constraints}) \\ & x \in \mathbb{Z}^n && (\textbf{Variables}) \end{aligned}$$

Integer Linear Program (ILP)

- Mathematical Formulation

$$\begin{array}{rll} \max & c^T x & (\textbf{Objective function}) \\ s.t. & Ax \leq b & (\textbf{Constraints}) \\ & x \in \mathbb{Z}^n & (\textbf{Variables}) \end{array}$$

- ILP is a general powerful and general approach.

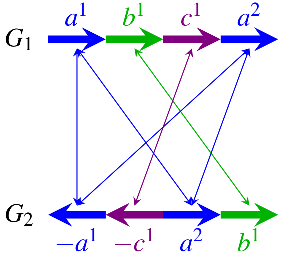
Integer Linear Program (ILP)

- Mathematical Formulation

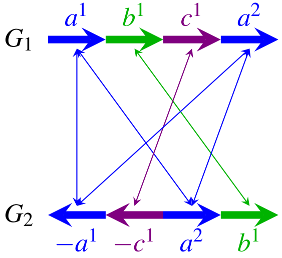
$$\begin{array}{rll} \max & c^T x & (\textbf{Objective function}) \\ s.t. & Ax \le b & (\textbf{Constraints}) \\ & x \in \mathbb{Z}^n & (\textbf{Variables}) \end{array}$$

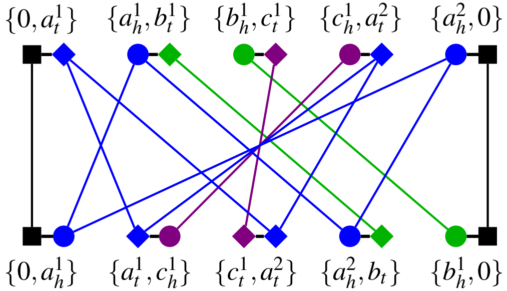
- ILP is a general powerful and general approach.
- Excellent Solvers
  - CPLEX
  - GUROBI

ILP Formulation

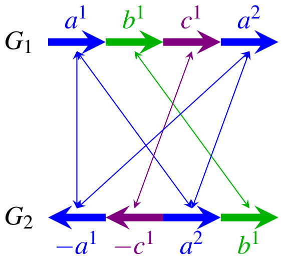


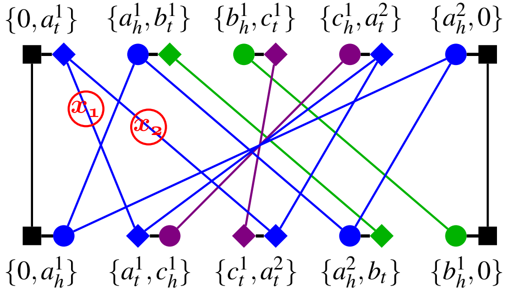
ILP Formulation





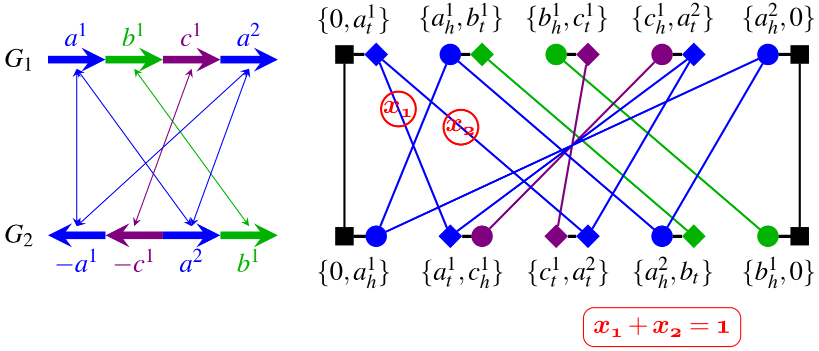
ILP Formulation





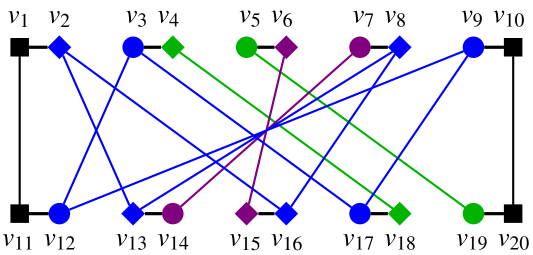
- Variables: $x_e \in \{0, 1\}$, indicating choosing e or not.

ILP Formulation

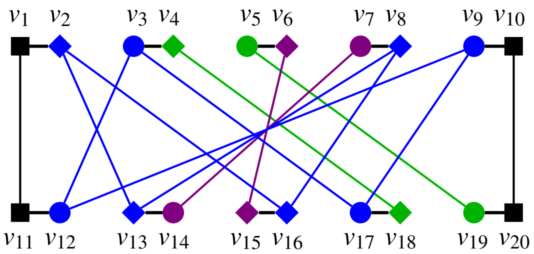


- Variables: $x_e \in \{0, 1\}$, indicating choosing e or not.
- Constraints: Ensure a one-to-one correspondence.

ILP Formulation



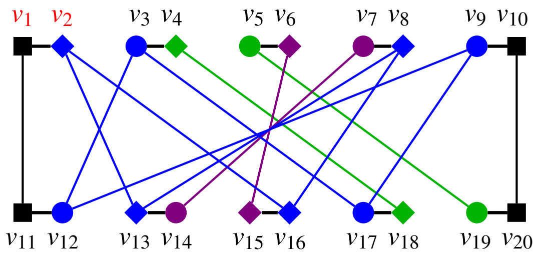
ILP Formulation



- Variables: $y_i \in [1, \boldsymbol{i}]$, representing the label of v_i

ILP Formulation

$y_1 = y_2$

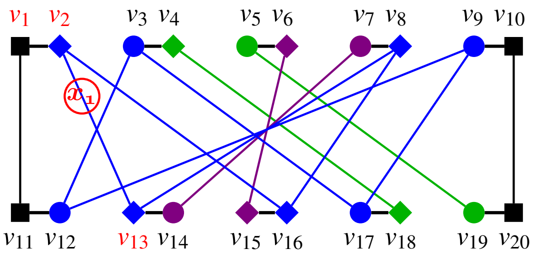


- Variables: $y_i \in [1, \boldsymbol{i}]$, representing the label of v_i
- Constraints:
  - Two vertices inside one adjacency have the same label.

ILP Formulation

$y_1 = y_2$

$y_2 \leq y_{13} + 2 \cdot (1 - x_1)$
$y_{13} \leq y_2 + 13 \cdot (1 - x_1)$



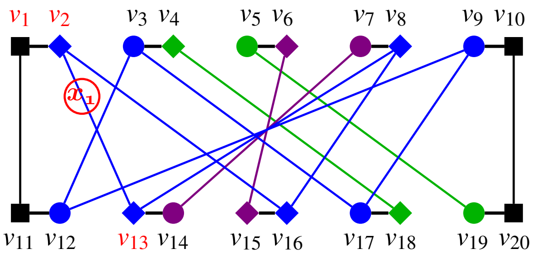
v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8 v_9 v_{10}
x_1
v_{11} v_{12} v_{13} v_{14} v_{15} v_{16} v_{17} v_{18} v_{19} v_{20}

- Variables: $y_i \in [1, \boldsymbol{i}]$, representing the label of v_i
- Constraints:
  - Two vertices inside one adjacency have the same label.
  - Vertices connected by chosen edges have the same label.

ILP Formulation

$y_1 = y_2$

$y_2 \leq y_{13} + 2 \cdot (1 - x_1)$
$y_{13} \leq y_2 + 13 \cdot (1 - x_1)$



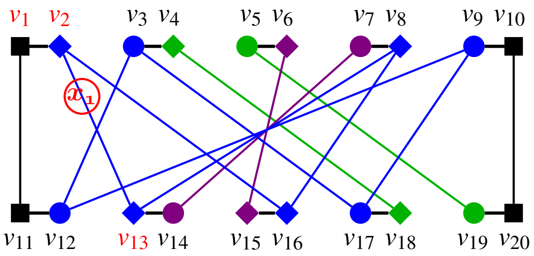
- Variables: $y_i \in [1, \boldsymbol{i}]$, representing the label of v_i
- Constraints:
  - Two vertices inside one adjacency have the same label.
  - Vertices connected by chosen edges have the same label.

$\Rightarrow$ For any feasible solution, all vertices in the same cycle (induced by the solution) must have the same label.

ILP Formulation

$y_1 = y_2$

$y_2 \leq y_{13} + 2 \cdot (1 - x_1)$
$y_{13} \leq y_2 + 13 \cdot (1 - x_1)$



- Variables: $y_i \in [1, \boldsymbol{i}]$, representing the label of v_i
- Constraints:
  - Two vertices inside one adjacency have the same label.
  - Vertices connected by chosen edges have the same label.

$\Rightarrow$ For any feasible solution, all vertices in the same cycle (induced by the solution) must have the same label.

$\Rightarrow$ At most one vertex in each cycle can reach the upper bound.

ILP Formulation

- Variables: $z_i \in \{0,1\}$, indicating whether $y_i = i$

ILP Formulation

- Variables: $z_i \in \{0,1\}$, indicating whether $y_i = i$
- Constraints: ensure that $z_i = 1$ only if $y_i = i$

$$i \cdot z_i \le y_i, \quad 1 \le i \le |V|$$

ILP Formulation

- Variables: $z_i \in \{0, 1\}$, indicating whether $y_i = i$
- Constraints: ensure that $z_i = 1$ only if $y_i = i$

$$i \cdot z_i \leq y_i, \quad 1 \leq i \leq |V|$$

- Objective: maximize the number of cycles

$$\max \sum_{1 \leq i \leq |V|} z_i$$

ILP Formulation

- Variables: $z_i \in \{0,1\}$, indicating whether $y_i = i$
- Constraints: ensure that $z_i = 1$ only if $y_i = i$

$$i \cdot z_i \leq y_i, \quad 1 \leq i \leq |V|$$

- Objective: maximize the number of cycles

$$\max \sum_{1 \leq i \leq |V|} z_i$$

- $O(|E|)$ variables and $O(|E|)$ constraints