Topics of today:

1. Short introduction to / review of ILP
2. DCJ distance of balanced genomes
3. ILP for computing the DCJ distance of balanced genomes

Linear Programming (LP)

Linear Optimization Problem

Given a set of **decision variables**
(each variable has a **domain**, stating its valid values)

Given a set of **linear constraints** in the given variables
where...
feasible solution: tuple of valid values that satisfies all constraints
infeasible solution: tuple of valid values that violates at least one constraint
solution space: set of all feasible solutions
(if no solution is feasible, the solution space is empty
and the optimization problem itself is infeasible)

The problem is finding a best feasible solution by **minimizing or maximizing**
the linear **objective function** in the given variables
(the objetive function defines the quality of the feasible solutions)

LP - simple example: optimal meal problem

Let the following table give the nutritional values of a portion of each of two types of food

	Vitamin			
	A	C	D	Calories
Food P	225	100	200	600
Food B	600	100	75	300
Min. intake	1800	550	600	

A solution to this optimization problem is a meal composed of portions of Food P and Food B

The **variables** are $\begin{cases} x_p\text{: \# of portions of Food P} \\ x_b\text{: \# of portions of Food B} \end{cases}$

The number of portions cannot be negative, therefore the respective **domains** are

$x_p \geq 0$ and $x_b \geq 0$

For achieving the minimum intake of each vitamin, any meal has to respect the **constraints** on the amount of vitamins:

$$\begin{aligned} 225x_p + 600x_b &\geq 1800 \text{ (Vitamin A)} \\ 100x_p + 100x_b &\geq \ \ 550 \text{ (Vitamin C)} \\ 200x_p + 75x_b &\geq \ \ 600 \text{ (Vitamin D)} \end{aligned}$$

The objective is the minimization of calories in the meal: minimize $600x_p + 300x_b$.

(As not eating violates the vitamin constraints, the empty meal is an **infeasible solution**.)

LP - simple example: optimal meal problem

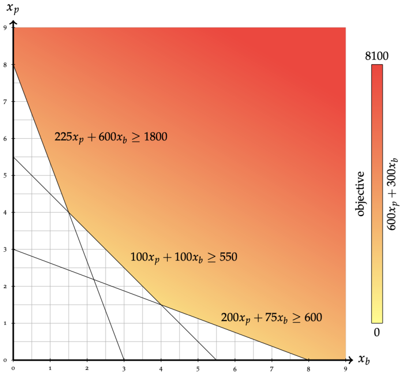
Graphical illustration:

The lines represent the constraints

The colored area corresponds to the solution space (set of feasible solutions), and the gradient indicates the value of the objective function in the solution space

The optimal meal consists of:

$$\begin{cases} x_p = 1.5 \text{ (\# of portions of Food P)} \\ x_b = 4 \quad \text{(\# of portions of Food B)} \\ \text{for a total of 2100 calories} \end{cases}$$



Example and picture taken from "Lecture Notes on Integer Linear Programming", by Roel van den Broek

The computation time of an LP is polynomial in the number of variables and constraints.

Integer Linear Programming (ILP)

Variation of LP, where each variable is restricted to integer values

Powerful tool in combinatorial optimization:
many problems feature discrete decisions that can be modeled in an ILP

In contrast to linear programs, which are all solvable to optimality in polynomial time (in the number of variables and constraints), often there is no known polynomial bound on the computation time of integer linear programs

Each ILP has a corresponding LP relaxation, by allowing the variables to assume non-integers values:

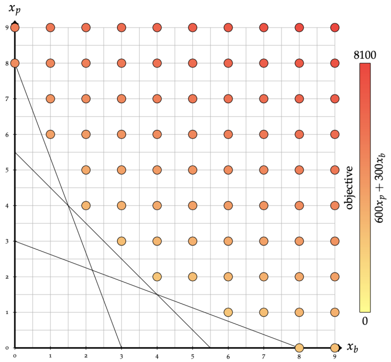
- Each feasible solution in the ILP is also feasible in the LP relaxation, but not vice versa.
- The optimal solution of the LP relaxation is a lower/upper bound for the optimal solution of the ILP.

ILP - simple example: optimal meal problem

Consider the LP for solving the optimal meal problem, and suppose that we want to avoid meals that consist of partial portions of the two types of food.

For this goal we only need to set the domains of the variables x_p and x_b to be non-negative integers:

$$\begin{array}{lll}
\min & 600x_p + 300x_b & \text{(objective function)} \\
\text{s.t.} & & \\
 & 225x_p + 600x_b \geq 1800 & \text{(Vitamin A)} \\
 & 100x_p + 100x_b \geq 550 & \text{(Vitamin C)} \\
 & 200x_p + 75x_b \geq 600 & \text{(Vitamin D)} \\
 & x_p \geq 0,\ x_b \geq 0 & \\
 & x_p, x_b \text{ are integers} & \text{(domains)}
\end{array}$$



The ILP formulation has three optimal solutions that result in 2400 calories $\begin{cases} x_p = 2, x_b = 4 \\ x_p = 1, x_b = 6 \\ x_p = 0, x_b = 8 \end{cases}$

ILP - NP-hard problem with simple formulation

MAXIMUM INDEPENDENT SET PROBLEM of a graph $G = (V, E)$

independent set: $\begin{cases} \text{subset of vertices } S \subseteq V \\ \text{no two vertices in } S \text{ are adjacent in graph } G \end{cases}$

ILP formulation:

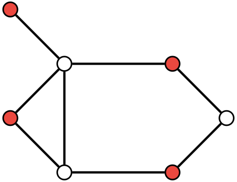
Let $V = \{v_1, v_2, ..., v_n\}$

Associate a **binary variable** to each vertex v_i: $x_i = \begin{cases} 1 & \text{if } v_i \text{ is in the independent set,} \\ 0 & \text{otherwise} \end{cases}$

Constraint: at most one vertex of each edge $(v_i v_j) \in E$ can be included in the independent set: $x_i + x_j \leq 1$

Objective: maximize the sum of all n binary x_k variables.

$$\boxed{\begin{array}{llll} \max \ \sum_{k=1}^{n} x_k & & & \text{(objective function)} \\ \text{s.t.} & & & \\ x_i + x_j & \leq 1 & \forall (v_i v_j) \in E & \text{(non-adjacent)} \\ x_k & \in \{0, 1\} & \forall v_k \in V & \text{(domain)} \end{array}}$$



(MAXIMUM INDEPENDENT SET PROBLEM is NP-hard, but modeling it as an ILP is straightforward)

Quiz 1

1 Which of the following statements are true?

[✓ A] A linear program can always be solved in polynomial time in the number of variables and constraints.

~~B~~ An integer linear program cannot be solved in polynomial time in the number of variables and constraints.

~~C~~ NP-hard problems can be solved by linear programming.

[✓ D] Each feasible solution in the ILP is also feasible in its LP relaxation, but not vice versa.

~~E~~ The optimal solution of an ILP is a lower/upper bound for the optimal solution of its LP relaxation.

DCJ distance of balanced genomes

Balanced genomes $\mathbb{A}$ and $\mathbb{B}$ $\begin{cases} \mathcal{F}_\star = \mathcal{F}(\mathbb{A}) = \mathcal{F}(\mathbb{B}) \\ \mathcal{G}_\star = \mathcal{G}(\mathbb{A}) = \mathcal{G}(\mathbb{B}) \\ \text{for each family } f \in \mathcal{F}_\star,\ \Phi(f, \mathbb{A}) = \Phi(f, \mathbb{B}) \end{cases}$

Transforming $\mathbb{A}$ and $\mathbb{B}$ into **matched** canonical genomes $\mathbb{A}^\ddagger$ and $\mathbb{B}^\ddagger$:

for each family $f \in \mathcal{F}_\star$, determine which occurrence of f in $\mathbb{A}$ matches each occurrence of f in $\mathbb{B}$

$\Rightarrow$ Matched occurrences receive the same **index** in $\mathbb{A}^\ddagger$ and in $\mathbb{B}^\ddagger$

The number of common genes between any pair of matched genomes $\mathbb{A}^\ddagger$ and $\mathbb{B}^\ddagger$ is $n_* = |\mathcal{G}_\star|$

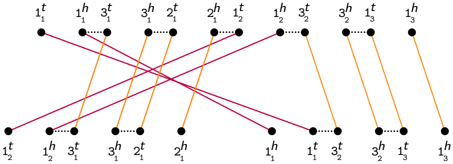
$\mathfrak{M}$: set of all possible pairs of matched canonical genomes obtained from balanced genomes $\mathbb{A}$ and $\mathbb{B}$

DCJ distance of $\mathbb{A}$ and $\mathbb{B}$:

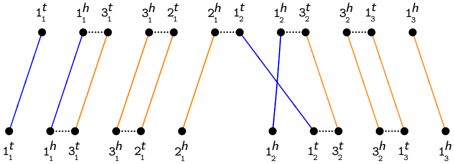
$$\mathrm{d}_{\mathrm{DCJ}}(\mathbb{A}, \mathbb{B}) = \min_{(\mathbb{A}^\ddagger, \mathbb{B}^\ddagger) \in \mathfrak{M}} \{\mathrm{d}_{\mathrm{DCJ}}(\mathbb{A}^\ddagger, \mathbb{B}^\ddagger)\}$$

Multi-relational graph of balanced genomes

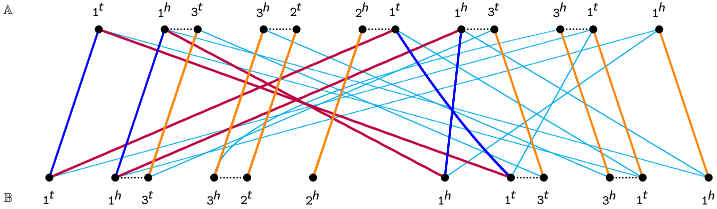
Example: $\mathbb{A} = [1\,3\,2\,1\,3\,1]$ and $\mathbb{B} = [1\,3\,2]\ \ [\bar{1}\,3\,1]$


1_1^t 1_1^h 3_1^t 3_1^h 2_1^t 2_1^h 1_2^t 1_2^h 3_2^t 3_2^h 1_3^t 1_3^h
1_2^t 1_2^h 3_1^t 3_1^h 2_1^t 2_1^h 1_1^h 1_1^t 3_2^t 3_2^h 1_3^t 1_3^h

$d_{\mathrm{DCJ}} = 6 - 2 - \frac{2}{2} = 3$


1_1^t 1_1^h 3_1^t 3_1^h 2_1^t 2_1^h 1_2^t 1_2^h 3_2^t 3_2^h 1_3^t 1_3^h
1_1^t 1_1^h 3_1^t 3_1^h 2_1^t 2_1^h 1_2^h 1_2^t 3_2^t 3_2^h 1_3^t 1_3^h

$d_{\mathrm{DCJ}} = 6 - 3 - \frac{2}{2} = 2$


$\mathbb{A}$
1^t 1^h 3^t 3^h 2^t 2^h 1^t 1^h 3^t 3^h 1^t 1^h
$\mathbb{B}$
1^t 1^h 3^t 3^h 2^t 2^h 1^h 1^t 3^t 3^h 1^t 1^h

Multi-relational graph of balanced genomes

Given two balanced genomes $\mathbb{A}$ and $\mathbb{B}$, their **multi-relational graph** $MRG(\mathbb{A}, \mathbb{B}) = (V, E)$ is described as follows:

1. $V = V(\xi(\mathbb{A})) \cup V(\xi(\mathbb{B}))$: there is a vertex for each extremity of each gene in $\mathbb{A}$
   and a vertex for each extremity of each gene in $\mathbb{B}$

   Each vertex v has a label $\ell(v)$, that corresponds to the gene extremity it represents.

2. $E = E_\alpha(\mathbb{A}) \cup E_\alpha(\mathbb{B}) \cup E_\xi$, where:
   - **Adjacency edges:** $\begin{cases} E_\alpha(\mathbb{A}) = \{uv : u, v \in V(\xi(\mathbb{A})) \text{ and } \ell(u)\ell(v) \in \alpha(\mathbb{A})\} \\ E_\alpha(\mathbb{B}) = \{uv : u, v \in V(\xi(\mathbb{B})) \text{ and } \ell(u)\ell(v) \in \alpha(\mathbb{B})\} \end{cases}$
   - **Extremity edges:** $E_\xi = \{uv : u \in V(\xi(\mathbb{A})) \text{ and } v \in V(\xi(\mathbb{B})) \text{ and } \ell(u) = \ell(v)\}$

Vertices can have degree greater than two in the multi-relational graph:

For each family $f \in \mathcal{F}_\star$, let $m_f = \phi(f, \mathbb{A}, \mathbb{B})$.

The number of extremity edges inciding in each vertex representing an extremity of an occurrence of f is m_f.

Sibling-sets of $MRG(\mathbb{A}, \mathbb{B})$

$f_{(\mathbb{A})}$ and $f_{(\mathbb{B})}$: genes (occurrences of the same family $f \in \mathcal{F}_\star$) in genomes $\mathbb{A}$ and $\mathbb{B}$.

Siblings: pair of extremity edges that connect $f^h_{(\mathbb{A})}$ to $f^h_{(\mathbb{B})}$ and $f^t_{(\mathbb{A})}$ to $f^t_{(\mathbb{B})}$

Sibling-set $S \subseteq E_\xi$ $\begin{cases} \text{is composed of pairs of siblings} \\ \text{does not contain any pair of incident edges} \end{cases}$ (is matching)

There is a bijection between pairs of (partially) matched genomes and sibling-sets of $MRG(\mathbb{A}, \mathbb{B})$:

- Denote by $\mathbb{A}^{\ddagger S}$ and $\mathbb{B}^{\ddagger S}$ the matched genomes corresponding to the sibling-set S
- If S is **maximal**, $\mathbb{A}^{\ddagger S}$ and $\mathbb{B}^{\ddagger S}$ are **canonical** genomes

Consistent decompositions of $MRG(\mathbb{A}, \mathbb{B})$

Consistent decomposition $D[S]$

– is **induced** by a maximal sibling-set S

– is the union of S with all adjacency edges

– covers all vertices of $MRG(\mathbb{A}, \mathbb{B})$

– is composed of cycles and paths:

weight of $D[S]$: $w(D[S]) = |\mathcal{C}^D| + \frac{|\mathcal{P}^D_{\mathbb{A}\mathbb{B}}|}{2}$

$\mathrm{d}_{\mathrm{DCJ}}(D[S]) = n_* - w(D[S])$

The DCJ distance of balanced $\mathbb{A}$ and $\mathbb{B}$ can then be computed by the following equation:

$$\mathrm{d}_{\mathrm{DCJ}}(\mathbb{A}, \mathbb{B}) = \min_{S \in \mathfrak{S}_{\mathrm{MAX}}} \{\mathrm{d}_{\mathrm{DCJ}}(D[S])\} = n_* - \max_{S \in \mathfrak{S}_{\mathrm{MAX}}} \{w(D[S])\},$$

where $\begin{cases} \mathfrak{S}_{\mathrm{MAX}} \text{ is the set of all maximal sibling-sets of } MRG(\mathbb{A}, \mathbb{B}) \\ n_* \text{ is constant for any consistent decomposition} \end{cases}$

If $\mathrm{d}_{\mathrm{DCJ}}(D[S]) = \mathrm{d}_{\mathrm{DCJ}}(\mathbb{A}, \mathbb{B})$, the consistent decomposition $D[S]$ is said to be **optimal**.

Capped multi-relational graph $CMRG(\mathbb{A}, \mathbb{B})$

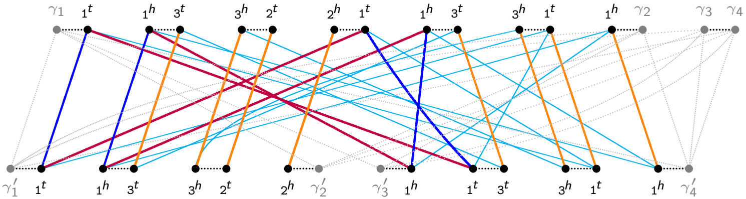
Example: $\mathbb{A} = [1\,3\,2\,1\,3\,1]$ and $\mathbb{B} = [1\,3\,2]\ \ [\bar{1}\,3\,1]$, $p_* = \max\{\kappa(\mathbb{A}), \kappa(\mathbb{B})\} = 2$, $a_* = |\kappa(\mathbb{A}) - \kappa(\mathbb{B})| = 1$

Add $2p_*$ cap extremities
to each genome

Add the new adjacencies
(including the a_* adj. between caps)
to $E_\alpha(\mathbb{A})$ and to $E_\alpha(\mathbb{B})$

$E_{\xi'}$: set of edges connecting
cap extremities

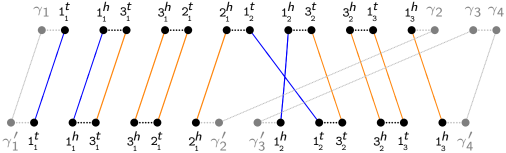
$E = E_\alpha(\mathbb{A}) \cup E_\alpha(\mathbb{B}) \cup E_\xi \cup E_{\xi'}$


γ_1 1^t 1^h 3^t 3^h 2^t 2^h 1^t 1^h 3^t 3^h 1^t 1^h γ_2 γ_3 γ_4
γ'_1 1^t 1^h 3^t 3^h 2^t 2^h γ'_2 γ'_3 1^h 1^t 3^t 3^h 1^t 1^h γ'_4

$CMRG(\mathbb{A}, \mathbb{B})$ includes all possible cappings of each maximal sibling-set

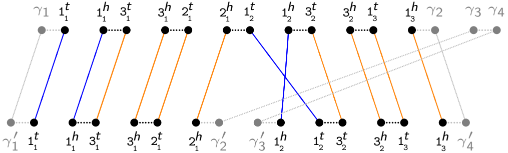
Two distinct cappings of the maximal sibling-set composed of blue + orange edges:

Non-optimal capping


γ_1 1^t_1 1^h_1 3^t_1 3^h_1 2^t_1 2^h_1 1^t_2 1^h_2 3^t_2 3^h_2 1^t_3 1^h_3 γ_2 γ_3 γ_4
γ'_1 1^t_1 1^h_1 3^t_1 3^h_1 2^t_1 2^h_1 γ'_2 γ'_3 1^h_2 1^t_2 3^t_2 3^h_2 1^t_3 1^h_3 γ'_4

$\mathrm{d}_{\mathrm{DCJ}} = n + p_* - c = 6 + 2 - 5 = 3$

Optimal capping


γ_1 1^t_1 1^h_1 3^t_1 3^h_1 2^t_1 2^h_1 1^t_2 1^h_2 3^t_2 3^h_2 1^t_3 1^h_3 γ_2 γ_3 γ_4
γ'_1 1^t_1 1^h_1 3^t_1 3^h_1 2^t_1 2^h_1 γ'_2 γ'_3 1^h_2 1^t_2 3^t_2 3^h_2 1^t_3 1^h_3 γ'_4

$\mathrm{d}_{\mathrm{DCJ}} = n + p_* - c = 6 + 2 - 6 = 2$

Capped consistent decompositions of $CMRG(\mathbb{A}, \mathbb{B})$

Capping-set $P \subseteq E_{\xi'}$: does not contain any pair of incident edges (is a matching)

Capped consistent decomposition $Q[S, P]$

- is **induced** by a maximal sibling-set S and a maximal caping-set P
- is the union of S with P and with all adjacency edges
- covers all vertices of $CMRG(\mathbb{A}, \mathbb{B})$
- is composed of cycles only:

  weight of $Q[S, P]$: $w(Q[S, P]) = |\mathcal{C}^Q|$

  $\mathrm{d}_{\mathrm{DCJ}}(Q[S, P]) = n_* + p_* - w(Q[S, P])$

For each maximal sibling set S: $\quad \mathrm{d}_{\mathrm{DCJ}}(D[S]) = \min_{P \in \mathfrak{P}_{\mathrm{MAX}}} \{\mathrm{d}_{\mathrm{DCJ}}(Q[S, P])\}$

The DCJ distance of balanced genomes $\mathbb{A}$ and $\mathbb{B}$ can be computed by the following equation:

$$\mathrm{d}_{\mathrm{DCJ}}(\mathbb{A}, \mathbb{B}) = \min_{S \in \mathfrak{S}_{\mathrm{MAX}}, P \in \mathfrak{P}_{\mathrm{MAX}}} \{\mathrm{d}_{\mathrm{DCJ}}(Q[S, P])\} = n_* + p_* - \max_{S \in \mathfrak{S}_{\mathrm{MAX}}, P \in \mathfrak{P}_{\mathrm{MAX}}} \{w(Q[S, P])\},$$

where

- $\mathfrak{S}_{\mathrm{MAX}}$ is the set of all maximal sibling-sets of $CMRG(\mathbb{A}, \mathbb{B})$
- $\mathfrak{P}_{\mathrm{MAX}}$ is the set of all maximal capping-sets of $CMRG(\mathbb{A}, \mathbb{B})$
- n_* and p_* are constant for any capped consistent decomposition

If $\mathrm{d}_{\mathrm{DCJ}}(Q[S, P]) = \mathrm{d}_{\mathrm{DCJ}}(\mathbb{A}, \mathbb{B})$, the capped consistent decomposition $Q[S, P]$ is said to be **optimal**.

ILP to find an optimal capped consistent decomposition of $CMRG(\mathbb{A}, \mathbb{B})$

Selecting a capped consistent decomposition:

Each edge $e \in E$ has a binary variable $x_e = \begin{cases} 1 & \text{if } e \text{ is selected to be in the decomposition,} \\ 0 & \text{otherwise} \end{cases}$

All adjacency edges must be in the decomposition:

$$x_a \;=\; 1 \qquad \forall\; a \in E_\alpha(\mathbb{A}) \cup E_\alpha(\mathbb{B})$$

Consistency $\begin{cases} \text{either both edges of a pair of siblings are selected} \\ \text{or both edges of a pair of siblings are not selected} \end{cases}$

$$x_e \;=\; x_d \qquad \forall\; e, d \in E_\xi,\; e, d \text{ are siblings}$$

The decomposition must be a collection of cycles - all vertices must have degree 2:

$$\sum_{uv \in E} x_{uv} \;=\; 2 \qquad \forall\; u \in V$$

(Guarantees maximality of the selected sibling-set)

ILP to find an optimal capped consistent decomposition of $CMRG(\mathbb{A}, \mathbb{B})$

Counting cycles:

For each vertex v_i, variable $\ell_i \geq 0$ corresponds to the label of v_i and is upper bounded by i:

$$\ell_i \leq i \qquad \text{for } 1 \leq i \leq |V|$$

All vertices in the same cycle in the decomposition must have the same label and this can be achieved by assigning the same label to the two vertices connected by a selected edge:

$$\left.\begin{array}{rcl} \ell_i & \leq & \ell_j + i(1 - x_{v_i v_j}) \\ \ell_j & \leq & \ell_i + j(1 - x_{v_i v_j}) \end{array}\right\} \ \forall\ v_i v_j \in E \quad \Big]$$

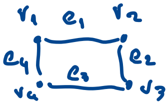
For each vertex v_i, binary variable $c_i \in \{0, 1\}$ tests whether ℓ_i is equal to its upper bound i:

$$\ell_i \geq i c_i \qquad \forall\ 1 \leq i \leq |V| \Big\}$$

$1 \geq \ell_1 \geq 1.c_1 \Rightarrow c_1$ can be 1

$1 \geq \ell_2 \geq 2.c_2 \Rightarrow c_2 = \emptyset$

Since all vertices in the same cycle have the same label and all upper bounds are distinct, there is exactly one vertex in each cycle whose label can be equal to its upper bound.



$\ell_1 \leq 1$ $\quad \ell_3 \leq 3$

$\ell_2 \leq 2$ $\quad \ell_4 \leq 4$

$\ell_1 \leq \ell_2$; $\ell_2 \leq \ell_1 \Rightarrow \ell_1 = \ell_2 \leq 1$

$\ell_1, \ell_2, \ell_3, \ell_4 \leq 1$

ILP to find an optimal capped consistent decomposition of $CMRG(\mathbb{A}, \mathbb{B})$

DCJ distance of balanced genomes

$$\begin{aligned}
\mathrm{d}_{\mathrm{DCJ}}(\mathbb{A}, \mathbb{B}) &= \min_{S \in \mathfrak{S}_{\mathrm{MAX}}, P \in \mathfrak{P}_{\mathrm{MAX}}} \{\mathrm{d}_{\mathrm{DCJ}}(Q[S, P])\} \\
&= n_* + p_* - \max_{S \in \mathfrak{S}_{\mathrm{MAX}}, P \in \mathfrak{P}_{\mathrm{MAX}}} \{w(Q[S, P])\} \\
&= n_* + p_* - \max_{S \in \mathfrak{S}_{\mathrm{MAX}}, P \in \mathfrak{P}_{\mathrm{MAX}}} \left\{ \left| \mathcal{C}^{Q[S,P]} \right| \right\}
\end{aligned}$$

Objective function:

$$\max \sum_{1 \le i \le |V|} c_i$$

Quiz 2

1 Which of the following statements are true?

~~A~~ The multi-relational graph is a collection of paths and cycles.

[✓ B] A consistent decomposition of the multi-relational graph is a collection of paths and cycles.

[✓ C] There is a bijection between consistent decompositions of $MRG(\mathbb{A}, \mathbb{B})$ and pairs of matched canonical genomes.

~~D~~ There is a bijection between capped consistent decompositions of $CMRG(\mathbb{A}, \mathbb{B})$ and pairs of matched canonical genomes.

s, p

2 Given that $\Phi(f, \mathbb{A}, \mathbb{B})$ is the number of occurrences of family f in $\mathbb{A}$ and in $\mathbb{B}$, the number of pairs of matched canonical genomes derived from balanced genomes $\mathbb{A}$ and $\mathbb{B}$ is...

[✓ A] $\prod_{f \in \mathcal{F}_\star} \Phi(f, \mathbb{A}, \mathbb{B})!$

B $2 \sum_{f \in \mathcal{F}_\star} \Phi(f, \mathbb{A}, \mathbb{B})!$

3 The number of distinct caping sets is

A $2p_*$

[✓ B] $(2p_*)!$

C $(2p_*)^2$

References

An Exact Algorithm to Compute the Double-Cut-and-Join Distance for Genomes with Duplicate Genes

(Mingfu Shao, Yu Lin, and Bernard M. E. Moret)

JCB, vol. 22, no. 5, pp 425–435 (2015)