Spis rzeczy

6 Mapy fizyczne genomów **97**
6.3 Mapy hybrydyzacyjne 97
6.3.1 Własność kolejnych jedynek (C1P) 98
6.3.2 Algorytm heurystyczny pozwalający odkrywać klony chimeryczne 106
6.3.3 Heurystyczne porządkowanie próbników 107

6 Mapy fizyczne genomów

6.3 Mapy hybrydyzacyjne

Przypomnijmy, że mapa fizyczna genomu (lub jego fragmentu) przedstawia rozmieszczenia pewnych markerów wzdłóż tego fragmentu. Typowe zastosowanie map fizycznych polega na tym, że jeśli mamy zsekwencjonowany pewien fragment S, to mając mapę fizyczną chromozomu, z którego S pochodzi, możemy ustalić położenia markerów na S i w ten sposób dopasować miejsce S na chromozomie. Mapy restrykcyjne można traktować jako jeden z rodzajów map fizycznych. Innym rodzajem takich map są *mapy hybrydyzacyjne*.

Mapy hybrydyzacyjne buduje się w następujący sposób. Mamy pewną bibliotekę klonów (fragmenty DNA), każdy klon zwykle jest długości kilkadziesiąt tysięcy par zasad. Ponadto mamy pewien zbiór próbników (ang. *probes*), które są krótkimi odcinkami DNA. Zadanie polega na stwierdzeniu, które próbniki hybrydyzują się do których klonów. Zbiór wszystkich próbników hybrydyzujących się do danego klonu nazywa się *odciskiem palca* (ang. *figerprint*) tego klonu. Idealna sytuacja ma miejsce wtedy, gdy próbniki hybrydyzują się w sposób jednoznaczny do całego długiego odcinka DNA, tzn. gdy każdy próbnik hybrydyzuje się w dokładnie jednym miejscu. Praca z takim zbiorem próbników nosi nazwę *techniki hybrydyzacji STS* (ang. *Sequence Tagged Site*).

Problem znajdowania odcisków palca poszczególnych klonów jest zawsze obarczony błędami. Są trzy rodzaje takich błędów:

- Próbnik może się nie hybrydyzować do miejsca, gdzie powinien (*false negative*).

- Próbnik może się hybrydyzować do miejsca, gdzie nie powinien (*false positive*).
- W czasie procesu klonowania dwa klony mogą się połączyć w jeden kawałek, tak jakby reprezentowały jeden klon (tzw. *klony chimeryczne*). Im dłuższe klony, tym większa obecność klonów chimerycznych (czasami dochodzi aż do 60%).

6.3.1 Własność kolejnych jedynek (C1P)

Przyjmujemy następujące założenia:

- Próbniki są jednoznaczne, tzn. każdy hybrydyzuje się w dokładnie jednym miejscu genomu.
- Nie ma błędów.
- Dla każdego klonu i każdego próbnika wiemy czy ten próbnik hybrydyzuje się do tego klonu.

Tak wię mamy n klonów $K_1, \ldots, K_n$ i m próbników $P_1, \ldots, P_m$. Ponadto mamy zero-jedynkową macierz M rozmiaru $(n \times m)$, taką że $M_{i,j} = 1 \iff P_j$ hybrydyzuje się do K_i. Zadanie polega na znalezieniu permutacji kolumn macierzy M, aby w każdym wierszu wszystkie jedynki stały obok siebie, bądź też na stwierdzeniu, że takiej permutacji nie ma. Mówimy, że macierz, dla której taka permutacja istnieje ma *własność kolejnych jedynek* (C1P). Zanlezienie permutacji, o której mowa powyżej pozwala ustalić kolejność, w jakiej próbniki hybrydyzują się do każdego z klonów.

Przykład 6.3.1 Poniżej znajduje się przykład takiej macierzy.

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
K_1	1	1	0	1	1	0	1	0	1
K_2	0	1	1	1	1	1	1	1	1
K_3	0	1	0	1	1	0	1	0	1
K_4	0	0	1	0	0	0	0	1	0
K_5	0	0	1	0	0	1	0	0	0
K_6	0	0	0	1	0	0	1	0	0
K_7	0	1	0	0	0	0	1	0	0
K_8	0	0	0	1	1	0	0	0	1

□

W ogólności zakładamy, że macierz nie zawiera wiersza składającego się z samych zer oraz, że nie zawiera dwóch identyczntch wierszy. Dla klonu K_i niech

$$S_i = \{j \mid M_{i,j} = 1\}.$$

Zbiór S_i będziemy nazywać *odciskiem palca* (bądź też *nośnikiem*) klonu K_i. Niech $\equiv$ będzie najmniejszą relacją równoważności w zbiorze klonów zawierającą relację $\sim$ zdefiniowaną poniżej

$$K_i \sim K_j \Longleftrightarrow S_i \cap S_j \neq \emptyset,\ S_i - S_j \neq \emptyset,\ S_j - S_i \neq \emptyset.$$

Przykład 6.3.2 Dla macierzy z Przykładu 6.3.1 mamy następujące klasy abstrakcji relacji $\equiv$

$$\alpha = \{K_1, K_2\}, \quad \beta = \{K_3\}, \quad \gamma = \{K_4, K_5\}, \quad \delta = \{K_6, K_7, K_8\}.$$

□

Spójne składowe (częściowe permutacje)

Dla każdej klasy abstrakcji $\{K_1, \ldots, K_l\}$ relacji $\equiv$ zbudujemy pewną rodzinę permutacji, które ustalają własność C1P dla fragmentu macierzy zawierającego w/w klony oraz wszystkie próbniki (bądź też uzyskamy informację, że takiej permutacji nie ma).

Permutacje będą reprezentowane przez *uporządkowane częściowe podziały* (UCP) zbioru $\{1, \ldots, m\}$. Każdy taki UCP to *ciąg* niepustych, parami rozłącznych zbiorów $Y_1, \ldots, Y_k$ spełniających warunek

$$\bigcup_{i=1}^{k} Y_i = \bigcup_{j=1}^{l} S_j.$$

Powiemy, że taki UCP jest zbudowany dla klonów $K_1, \ldots, K_l$.

Permutacje z takiego UCP są otrzymywane następująco: niech $X \subseteq \{1, \ldots, m\} - \bigcup Y_i$ i niech $X' = \{1, \ldots, m\} - (X \cup \bigcup Y_i)$. Porządkujemy liniowo każdy ze zbiorów $X, Y_1, \ldots, Y_k, X'$ i to daje permutację. Każda permutacja otrzymana w ten sposób (również dla różnych X) jest jedną z permutacji reprezentowanych przez UCP.

Będziemy budować UCP iteracyjnie. Pierwszy UCP to S_1. Przypuśćmy, że już mamy UCP $Y_1, \ldots, Y_k$ dla klonów $K^{(1)}, \ldots, K^{(j)}$, (gdzie $j < l$) spełniających warunek, że pomiędzy każdymi dwoma klonami z tego zbioru można

przejść przy pomocy $\sim$, używając jedynie klonów z tego zbioru. Niech $K^{(j+1)}$ będzie klonem takim, że dla pewnego $i \leq j$ zachodzi $K^{(j+1)} \sim K^{(i)}$. Niech $X = S^{(j+1)}$ będzie odciskiem palca tego klonu. Rozważmy zbiór

$$I = \{i \mid Y_i \cap X \neq \emptyset\}.$$

Musi być tak, że dla pewnych $i_1 \leq i_2$, zbiór I ma postać $I = \{i \mid i_1 \leq i \leq i_2\}$. Jeśli I nie ma takiej postaci, to dla rozważanego zbioru klonów nie ma własności C1P. Musi być też tak, że dla $i_1 < i < i_2$, zachodzi $Y_i \subseteq X$ (jeśli tak nie jest to nie ma własności C1P).

Niech $Z = X - \bigcup_{i \in I} Y_i$. Jeśli $Z \neq \emptyset$, to musi zajść jedna z następujących dwóch możliwości:
(a.) $i_1 = 1$ oraz $Y_1 \subseteq X$.
(b.) $k = 1$ lub ($i_2 = k$ oraz $Y_k \subseteq X$).

W przypadku (a.) definiujemy nowy UCP następująco

$$Z, Y_1, \ldots, Y_{i_2-1}, X \cap Y_{i_2}, Y_{i_2} - X, Y_{i_2+1}, \ldots, Y_k.$$

Natomiast w przypadku (b.) nowe UCP wygląda następująco

$$Y_1, \ldots, Y_{i_1-1}, Y_{i_1} - X, X \cap Y_{i_1}, Y_{i_1+1}, \ldots, Y_k, Z.$$

Jeśli natomiast $Z = \emptyset$, to nowy UCP wygląda następująco

$$Y_1, \ldots, Y_{i_1-1}, Y_{i_1} - X, X \cap Y_{i_1}, Y_{i_1+1}, \ldots, Y_{i_2-1}, X \cap Y_{i_2}, Y_{i_2} - X, Y_{i_2+1}, \ldots, Y_k.$$

Poprawność powyższego algorytmu wynika z następujących uwag:

- Z założenia spójności grafu $\langle \{K^{(1)}, \ldots, K^{(j)}, \sim \rangle$ wynika, że bloków $Y_1, \ldots, Y_k$ nie można rozdzielić przez wstawianie nowych zbiorów.
- Bloki podziału zbioru X występujące w nowym UCP muszą stać obok siebie.

Czas jaki zajmuje wykonanie poszczególnych kroków algorytmu wygląda następująco.

- Zbudowanie grafu $\langle \{K^{(1)}, \ldots, K^{(j)}, \sim \rangle$ zajmuje czas $O(n^2 m)$ (dla każdej pary K_i, K_j, sprawdzenie czy zachodzi $K_i \sim K_j$ zajmuje $O(m)$ kroków).
- Spójne składowe grafu przechodzimy metodą przeszukiwania w głąb. Dostawienie nowego klonu zabiera $O(m)$ kroków. Łącznie mamy więc $O(nm)$ kroków.

Zatem łączny czas wykonania całej fazy budowania UCP dla poszczególnych spójnych składowych wynosi $O(n^2m)$.

Przykład 6.3.3 Kontynuujemy Przykład 6.3.1.
(a.) Dla składowej $\{K_1, K_2\}$ mamy UCP otrzymany w następującym ciągu kroków:
dla K_1 : $\{1,2,4,5,7,9\}$ $[= S_1]$.
dla K_1, K_2 : $\{1\},\{2,4,5,7,9\},\{3,6,8\}$.

(b.) Dla składowej $\{K_3\}$, UCP jest równe S_3:
$\{2,4,5,7,9\}$.

(c.) Dla składowej $\{K_4, K_5\}$ mamy:
dla K_4 : $\{3,8\}$.
dla K_4, K_5 : $\{8\},\{3\},\{6\}$.

(d.) Wreszcie dla składowej $\{K_6, K_7, K_8\}$ mamy:
dla K_6 : $\{4,7\}$.
dla K_6, K_7 : $\{4\},\{7\},\{2\}$.
dla K_6, K_7, K_8 : $\{5,9\},\{4\},\{7\},\{2\}$.
Przykładową permutacją reprezentowaną przez powyższe UCP jest

	1	3	6	8	5	9	4	7	2
6	0	0	0	0	0	0	1	1	0
7	0	0	0	0	0	0	0	1	1
8	0	0	0	0	1	1	1	0	0

Kolumny 5 i 9 można zamienić miejscami oraz dowolne z kolumn 1,3,6,8 można przenieść na prawy skraj tablicy. □

Scalanie składowych

Niech $\alpha, \beta, \gamma, \ldots$ będą klasami abstrakcji relacji $\equiv$ (czyli spójnymi składowymi), dla których mamy już zdefiniowane UCP. Zdefiniujemy relację $\leq$ na składowych. Niech $Y_1, \ldots, Y_k$ będzie UCP dla α oraz niech $X_1, \ldots, X_p$ będzie UCP dla β.

$$\alpha \leq \beta \Longleftrightarrow \exists i \leq k \forall j \leq p \; X_j \subseteq Y_i.$$

Przykład 6.3.4 Dla klas abstrakcji z Przykładu 6.3.2 mamy

$$\alpha \leq \beta \leq \delta, \quad \alpha \leq \gamma.$$

□

Lemat 6.3.1 $\leq$ *jest relacją częściowego prządku w zbiorze klas abstrakcji relacji* $\equiv$.

Dowód: Jeśli $\alpha \leq \beta$ oraz $\beta \leq \alpha$ to UCP dla obydwu klas są jednoelementowe i są identyczne. Zatem α i β składają się z jednego klonu i nośniki tych klonów są identyczne. Sprzeczność. ■

Niech

$$S(\alpha) = \bigcup \{Y \mid Y \text{ występuje w UCP dla } \alpha\}.$$

Lemat 6.3.2 *Jeśli* $\beta \not\leq \gamma$ *oraz* $\gamma \not\leq \beta$, *to* $S(\beta) \cap S(\gamma) = \emptyset$.

Dowód: Przyjmijmy założenia lematu i niech $S(\beta) \cap S(\gamma) \neq \emptyset$. Zatem istnieją $K_i \in \beta$ oraz $K_j \in \gamma$ takie, że $S_i \cap S_j \neq \emptyset$. Ponieważ $K_i \not\sim K_j$, to $S_i \subseteq S_j$ lub $S_j \subseteq S_i$. Możemy przyjąć, że pierwsze zawieranie ma miejsce. Niech $K_p \in \beta$ będzie dowolny taki, że $S_p \cap S_i \neq \emptyset$. Wówczas $S_p \cap S_j \neq \emptyset$ i znowu pokazujemy, że $S_p \subseteq S_j$ lub $S_j \subseteq S_p$. Teraz już drugie zawieranie jest niemożliwe (bo wówczas $S_i \subseteq S_p$). Zatem $S_p \subseteq S_j$. W ten sposób pokazaliśmy, że

$$\forall K_p \in \beta \; S_p \subseteq S_j.$$

Weźmy teraz dowolne $K \in \beta$ oraz dowolne $K' \in \gamma$ takie, że ich nośniki S oraz S' mają niepuste przecięcie. Podobnie jak poprzednio pokazujemy, że dla dowolnego $K'' \in \beta$ o nośniku S'' mamy $S'' \subseteq S'$. Ponieważ bloki UCP dla γ powstają przez przecięcie pewnej liczby nośników klonów należących do γ, to z poprzedniego zdania wynika, że dla pewnego Y będącego blokiem UCP dla γ mamy

$$\forall K_i \in \beta \; S_i \subseteq Y.$$

To dowodzi, że $\gamma \leq \beta$. Otrzymana sprzeczność kończy dowód lematu. ■

Konstrukcja ostatecznej permutacji powstaje przez rozszerzanie kolejnych UCP. Konstrukcję tę przeprowadzamy przez indukcję ze względu na porządek $\leq$ na składowych.

Krok bazowy. Wybieramy element minimalny w sensie $\leq$ i bierzemy jego UCP.

Krok indukcyjny. Przypuśćmy, że mamy skonstruowane dotąd UCP $Y_1, \ldots, Y_k$, dla pewnego zbioru składowych $\mathcal{K}$ o następujących własnościach.

(W1) Jeśli $\alpha \in \mathcal{K}$ oraz $\beta \leq \alpha$, to $\beta \in \mathcal{K}$.

(W2) $\bigcup_{i=1}^{k} Y_i = \bigcup_{\beta \in \mathcal{K}} S(\beta)$.

Niech γ będzie minimalną składową nie należącą do $\mathcal{K}$ i niech $X_1, \ldots, X_p$ będzie UCP dla γ. Rozważmy dwa przypadki:

(A) Dla każdego $i \leq k$ zachodzi $S(\gamma) \cap Y_i = \emptyset$.

(B) Dla pewnego $i \leq k$ mamy $S(\gamma) \cap Y_i \neq \emptyset$.

W przypadku (A) nowym UCP jest

$$X_1, \ldots, X_p, Y_1, \ldots, Y_k.$$

Przed rozważeniem przypadku (B) weźmy dwa UCP $X_1, \ldots, X_p$ oraz $Y_1, \ldots, Y_k$ takie, że $\bigcup_{i=1}^{p} X_i \subseteq \bigcup_{i=1}^{k} Y_i$. Będziemy używać następującej notacji. Dla $j_i \leq j_2$, przez $[j_1, j_2]$ będziemy oznaczać przedział domknięty

$$[j_1, j_2] = \{j \mid j_1 \leq j \leq j_2\}.$$

Ponadto przyjmijmy następującą relację pomiędzy przedziałami

$$[i_1, i_2] \leq [j_1, j_2] \iff \forall i \in [i_1, i_2] \forall j \in [j_1, j_2]\ i \leq j.$$

Zastanówmy się nad tym jakie warunki muszą być spełnione, aby można było stworzyć UCP będące jednocześnie rozdrobnieniem obydwu danych UCP. Niech $1 \leq i \leq k$ i rozważmy zbiór

$$I_i = \{j \in [1, p] \mid X_j \cap Y_i \neq \emptyset\}.$$

Natępujące warunki muszą być oczywiście spełnione.

(U1) Każde niepuste I_i musi być przedziałem domkniętym, tzn. istnieją $j_1 \leq j_2$ takie, że $I_i = [j_1, j_2]$.

(U2) Jeśli $i_1 < i_2$, to

- **(a.)** $I_{i_1} \leq I_{i_2}$.
- **(b.)** $|I_{i_1} \cap I_{i_2}| \leq 1$.

(c.) Jeśli $I_{i_1} \neq \emptyset \neq I_{i_2}$, to dla każdego i spełniającego $i_1 < i < i_2$ zachodzi $I_i \neq \emptyset$.

(U3) $\bigcup_{i=1}^{k} I_i = [1, p]$.

Zauważmy, że z (U2) wynika iż dla dowolnych $i_1 < i_2$, jeśli $I_{i_1} \cap I_{i_2} \neq \emptyset$, to istnieją takie trzy elementy $j_1 \leq j \leq j_2$, że $I_{i_1} = [j_1, j]$, $I_{i_2} = [j, j_2]$ oraz dla wszystkich i spełniających $i_1 < i < i_2$ zachodzi $I_i = [j, j]$. Również wynika z (U2), że jeśli dla pewnego i mamy $I_i = \emptyset$, to albo dla wszystkich $i' \leq i$ zachodzi $I_{i'} = \emptyset$, lub dla wszystkich $i' \geq i$ zachodzi $I_{i'} = \emptyset$.

Warunki (U1)-(U3) są oczywiście konieczne dla istnienia UCP będącego rozdrobnieniem obydwu wyjściowych UCP. Warunki te są również wystarczające. Istotnie, rozważmy wszystkie niepuste przecięcia $Y_i \cap X_j$ dla $i \in [1, k]$ oraz $j \in [1, p]$. Niech

$$D = \{(i, j) \mid Y_i \cap X_j \neq \emptyset\} \cup \{(i, 0) \mid \forall j \; Y_i \cap X_j = \emptyset\}.$$

Zbiór D porządkujemy liniowo:

$$(i, j) \preceq (i', j') \iff i < i' \vee (i = i' \wedge j \leq j').$$

Ponieważ każda para $(i, j) \in D$ jednoznacznie wyznacza pewien niepusty zbiór $Z_{(i,j)}$ (równy $Y_i \cap X_j$, gdy $j \neq 0$; Y_i, gdy $j = 0$), to

$$(\{Z_{(i,j)} \mid (i, j) \in D\}, \preceq)$$

tworzy pewien UCP. Jest to szukane UCP. Nazwiemy go *minimalnym rozdrobnieniem* dla wyjściowych UCP.

Wracamy teraz do konstrukcji i przyjmijmy, że zachodzi warunek (B). Zauważmy, że wówczas istnieje $\beta \in \mathcal{K}$ takie, że $\beta \leq \gamma$. Istotnie, gdyby dla każdego $\beta \in \mathcal{K}$ zachodziło $\beta \not\leq \gamma$, to ponieważ $\gamma \not\leq \beta$ musi zachodzić (na mocy warunku (W1)), to z Lematu 6.3.2 mamy $S(\beta) \cap S(\gamma) = \emptyset$, dla wszystkich $\beta \in \mathcal{K}$. Zatem z warunku (W2) wynika, że $S(\gamma) \cap Y_i = \emptyset$ dla $i \in [1, k]$. Otrzymana sprzeczność dowodzi, że istnieje $\beta \in \mathcal{K}$ spełniające $\beta \leq \gamma$. Niech X będzie blokiem UCP dla β takim, że $S(\gamma) \subseteq X$. Z warunku (W2) dostajemy

$$\bigcup_{i=1}^{p} X_i = S(\gamma) \subseteq S(\beta) \subseteq \bigcup_{i=1}^{k} Y_i.$$

Zatem możemy przystąpić do konstruowania minimalnego rozdrobnienia dla UCP $X_1, \ldots, X_p$ oraz $Y_1, \ldots, Y_k$. Jeśli warunki (U1)-(U3) są spełnione, to tworzymy minimalne rozdrobnienie w/w podziałów, dołączając γ do zbioru $\mathcal{K}$. W przeciwnym przypadku stwierdzamy, że macierz nie ma własności C1P. To kończy opis algorytmu sprawdzającego własność C1P.

Czas fazy scalania składowych

- Wyznaczenie porządku $\leq$ w zbiorze składowych zajmuje czas $O(mn^2)$ (dla każdej pary składowych α, β stwierdzenie czy zachodzi $\alpha \leq \beta$ zajmuje $O(m)$ kroków).
- Proces scalania składowych zajmuje $O(mn)$ kroków (dla każdego α, wstawienie α do $\mathcal{K}$ i sprawdzenie czy warunki (U1)-(U3) zachodzą zajmuje $O(m)$ kroków).

Mamy w ten sposób udowodnione następujące twierdzenie.

Twierdzenie 6.3.3 *Dla dowolnej $n \times m$ macierzy binarnej można stwierdzić w czasie $O(mn^2)$ czy macierz ta ma własność kolejnych jedynek (C1P).*

Booth i Leuker (1976) podali algorytm sprawdzania C1P dla macierzy $n \times m$ w czasie $O(m+n+r)$, gdzie r jest liczbą jedynek w macierzy. Algorytm ten używa tzw. pq-drzew.

Przykład 6.3.5 Zilustrujemy powyższą metodę. Kontynuujemy Przykład 6.3.4. Zaczynamy od klasy α.
UCP dla α: $\{1\}, \{2,4,5,7,9\}, \{3,6,8\}$.
UCP dla α, β: $\{1\}, \{2,4,5,7,9\}, \{3,6,8\}$.
UCP dla α, β, δ: $\{1\}, \{5,9\}, \{4\}, \{7\}, \{2\}, \{3,6,8\}$.
UCP dla $\alpha, \beta, \delta, \gamma$: $\{1\}, \{5,9\}, \{4\}, \{7\}, \{2\}, \{8\}, \{3\}, \{6\}$.
Powyższy UCP wyznacza dwie permutacje. Jedną z nich jest

	1	5	9	4	7	2	8	3	6
1	1	1	1	1	1	1	0	0	0
2	0	1	1	1	1	1	1	1	1
3	0	1	1	1	1	1	0	0	0
4	0	0	0	0	0	0	1	1	0
5	0	0	0	0	0	0	0	1	1
6	0	0	0	1	1	0	0	0	0
7	0	0	0	0	1	1	0	0	0
8	0	1	1	1	0	0	0	0	0

□

Uwaga: można pokazać, że następujący problem jest NP-zupełny. Dana macierz binarna M i kiczba k. Sprawdzić czy istnieje permutacja kolumn macierzy M, przy której w każdym wierszu macierzy jest co najwyżej k bloków jedynek.

Dopuszczanie błedów o jakich mówiliśmy wcześniej pociąga powstawanie więcej niż jednego bloku jedynek w wierszach. Stosuje się wtedy algorytmy aproksymacyjne.

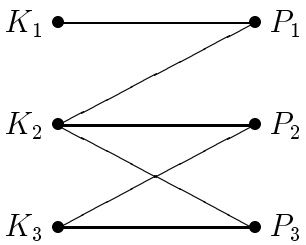
6.3.2 Algorytm heurystyczny pozwalający odkrywać klony chimeryczne

Dla każdego klonu K_i budujemy graf $G_i = \langle V_i, E_i \rangle$, gdzie V_i jest zbiorem wszystkich próbników hybrydyzujących się do K_i oraz relacja E_i jest zdefiniowana następująco

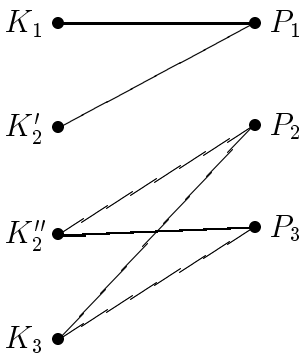
$$(P_j, P_{j'}) \in E_i \Longleftrightarrow \exists i' \neq i \text{ taki, że } P_j, P_{j'} \text{ hybrydyzują sie do } K_{i'}.$$

Gdy graf G_i jest spójny to przypuszczamy, że klon K_i nie jest chimeryczny. Natomiast niespojność G_i jest wskazówka potencjalnej chimeryczności K_i. Wówczas K_i możemy zastąpć 'sztucznymi' klonami, po jednym dla każdej spójnej składowej.

Przykład 6.3.6 Rozważmy następujący graf hybrydyzacyjny.



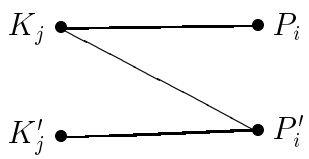
W grafie G_2 wierzchołki P_2 i P_3 są połączone krawędzią, a wierzchołek P_1 jest izolowany. Jeśli zastąpimy klon K_2 dwoma sztucznymi klonami K_2' oraz K_2'', to dostaniemy następujący nowy graf hybrydyzacyjny.



□

6.3.3 Heurystyczne porządkowanie próbników

Algorytm ten jest oparty na pojęciu tzw. splitera. Dany próbnik P_i, niech V_i będzie zbiorem wszystkich wierzchołków grafu hybrudyzacyjnego, które można osiągnąć z wierzchołka P_i drogą



Wówczas do zbioru V_i bierzemy P_i, P'_i, K_j, K'_j. Zbiór V_i przedstawia pewne otoczenia próbnika P_i.

Niech H będzie zbiorem wszystkich klonów i próbników. Jeśli $H - V_i$ jest grafem o dokładnie dwóch składowych, to próbnik P_i nazwiemy *spliterem*, a te składowe nazwiemy składowymi splitera.

Algorytm działa następująco. Wybieramy pewien spliter P_k i niech A_k i B_k będą jego składowymi. Przyjmujemy (arbitralnie), że A_k jest *lewą składową*, a B_k jest *prawą składową*. Ten wybór ustala orientację składowych pozostałych spliterów. Dla splitera P_i i jego składowych X, Y, przyjmujemy, że X jest lewą składową, gdy $|X \cap A_k| > |X \cap B_k|$. Dla każdego próbnika P_i liczymy l_i = liczba lewych składowych zawierających ten próbnik oraz r_i = liczba prawych składowych zawierających ten próbnik. Następnie sortujemy próbniki, stosując zasadę , że próbnik P_i o wiekszej liczbie $l_i - r_i$ jest bardziej na lewo.