

# Unpaired Kidney Exchange:

Overcoming the double coincidence of wants without a medium of exchange

**Conférence AFSE et DG Trésor**

Mohammad Akbarpour, *Stanford GSB*

**Julien Combe, CREST – Ecole polytechnique**

Yinghua He, *Rice U*

Victor Hiller, LEMMA - *University Paris II*

Robert Shimer, *University of Chicago*

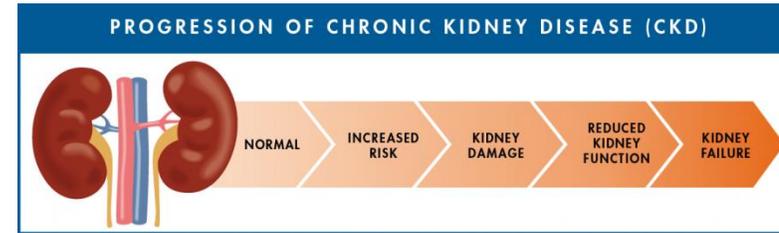
Olivier Tercieux, *CNRS & PSE*



Décembre 2019

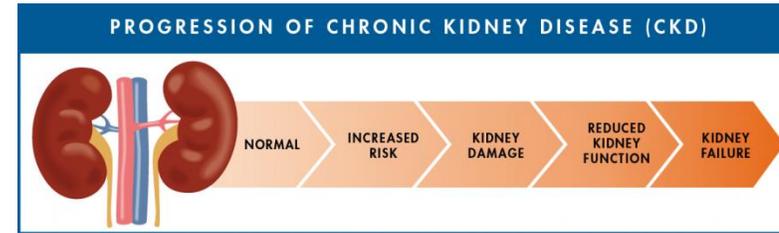


# Don croisé



- Maladie rénale chronique (MRC):
  - 11e cause de décès au monde (2016 Global Burden of Disease Study)
  - 1.2 millions morts dans le monde par an
  - A doublé depuis 1990

# Don croisé

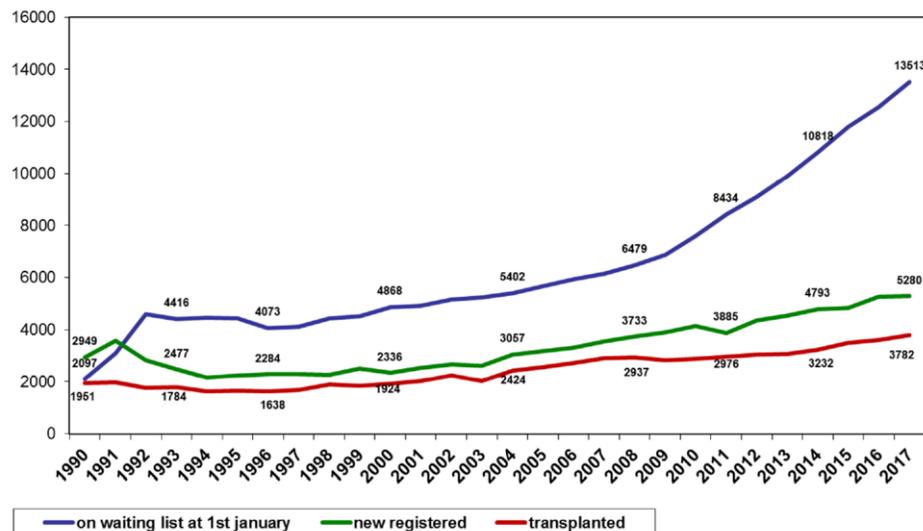


- Maladie rénale chronique (MRC):
  - 11e cause de décès au monde (2016 Global Burden of Disease Study)
  - 1.2 millions morts
  - A doublé depuis 1990
- Greffe v.s. dialyse:
  - Meilleure espérance de vie (+ 7 ans)
  - Plus bas coût (environ 730 000\$ par patient)



# Don croisé

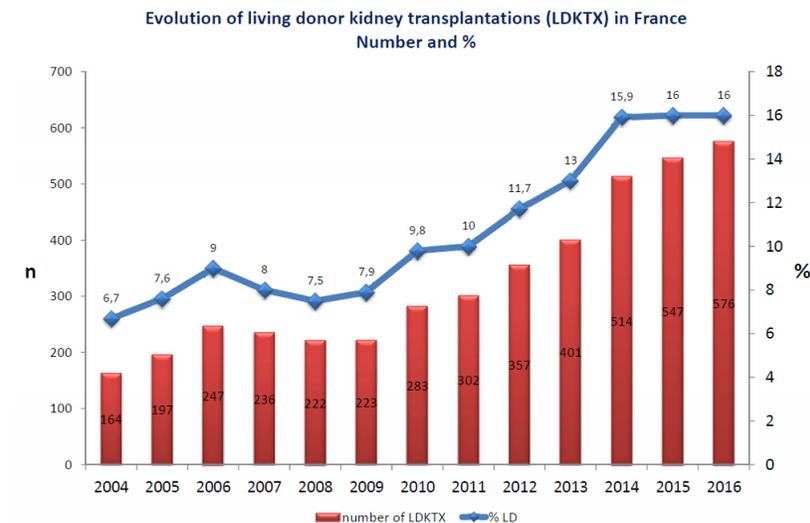
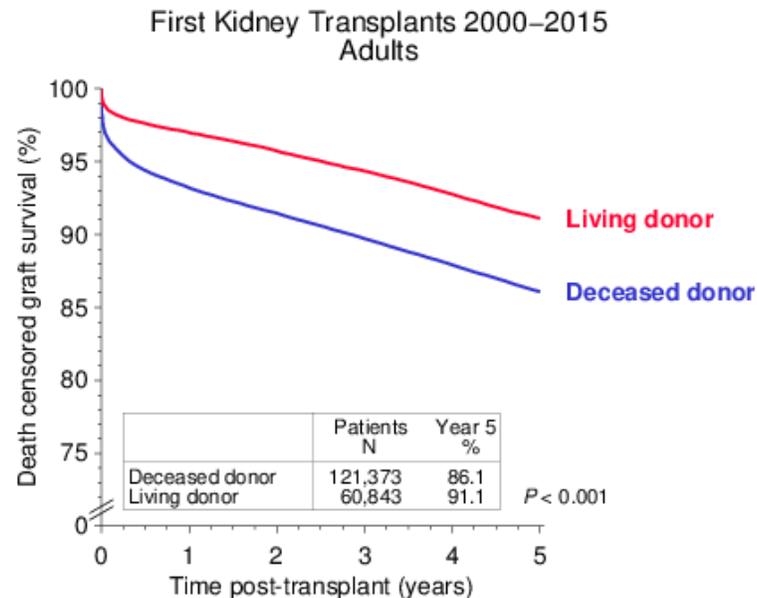
- Greffe depuis des **donneurs décédés**:
  - Offre relativement fixe depuis 10 ans
  - Demande a augmenté: hausse du nombre de patients sur liste d'attente
    - + 65% aux Etats-Unis, + 100% en France (2007-2017)



Source: Agence de la Biomédecine (France)

# Don croisé

- Greffe depuis des **donneurs vivants**
  - En moyenne: meilleure qualité
  - A pris de l'ampleur: 40% des greffes de reins dans le monde en 2015...
  - ... mais beaucoup de patients incompatibles avec leurs donneurs:  $\approx 40\%$  in Europe



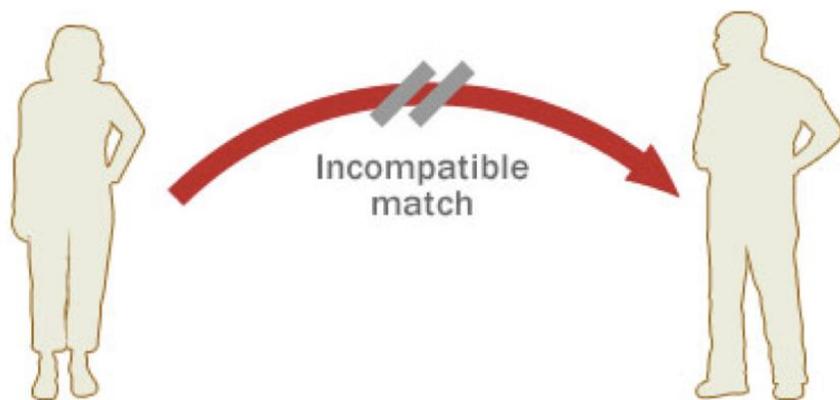
# Don croisé

- Greffe depuis des **donneurs vivants**
  - En moyenne: meilleure qualité
  - A pris de l'ampleur: 40% des greffes de reins dans le monde en 2015...
  - ... mais beaucoup de patients incompatibles avec leurs donneurs:  $\approx 40\%$  in Europe

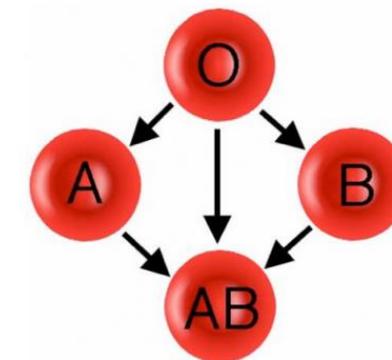


Développement des programmes d'échange  
de reins (don croisé)

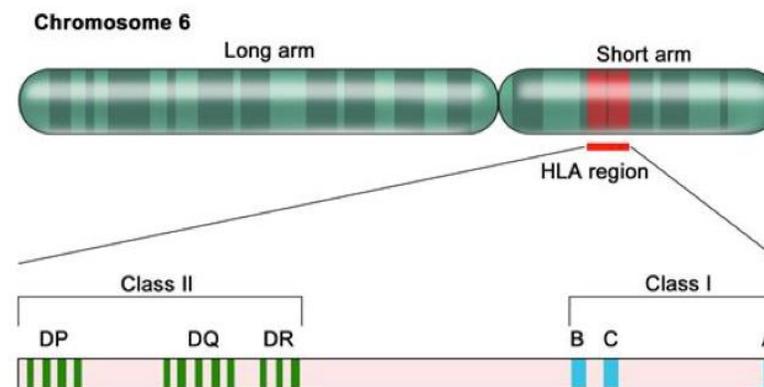
# Don croisé



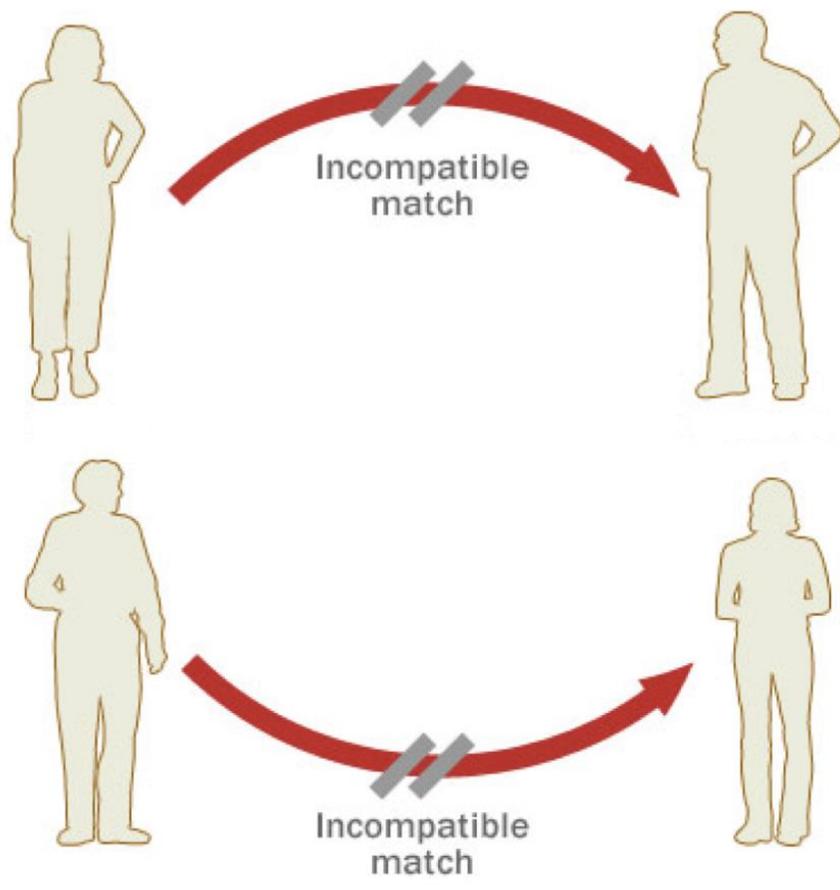
Groupe sanguin:



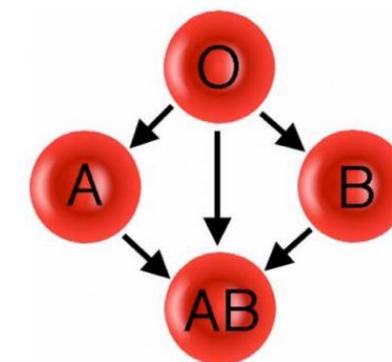
Tissu:



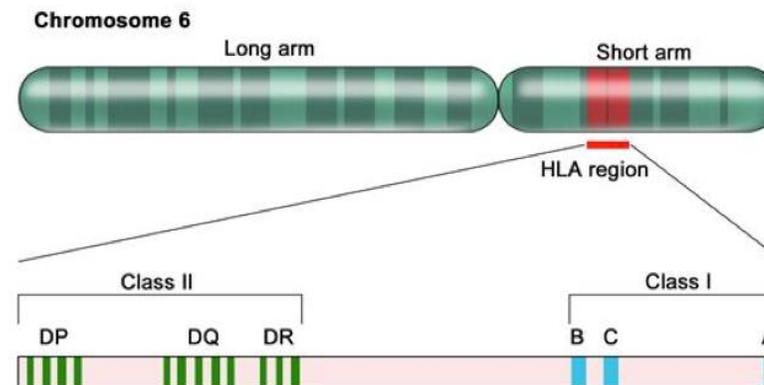
# Don croisé



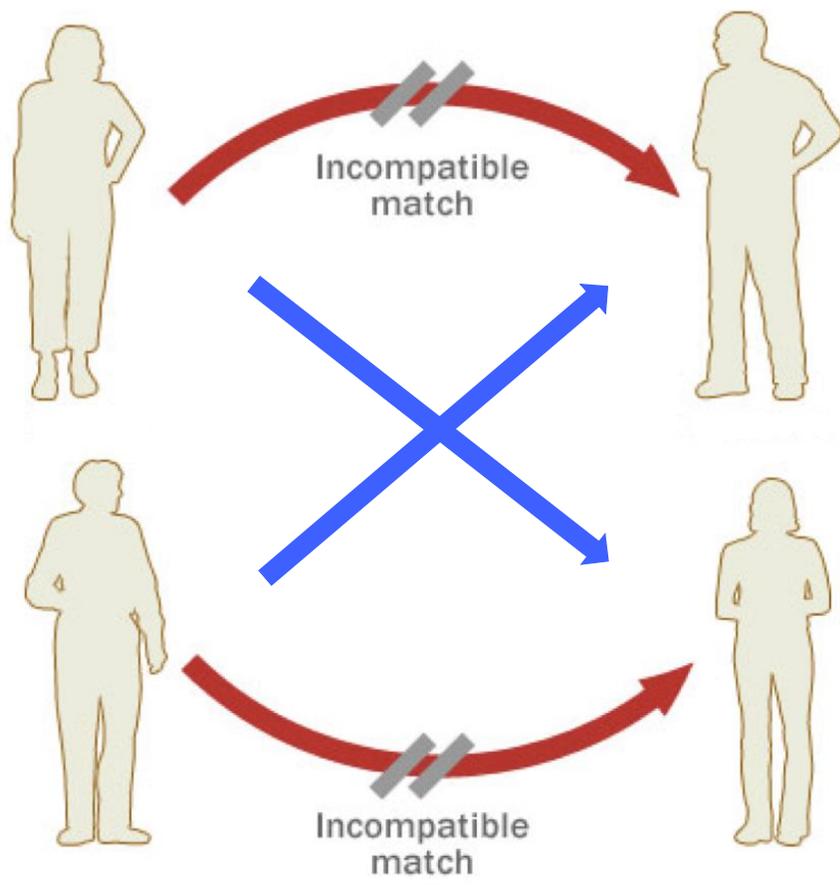
Groupe sanguin:



Tissu:



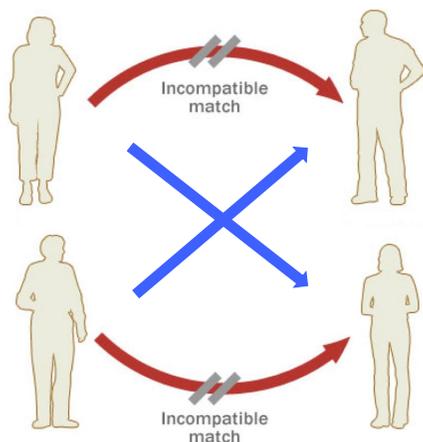
# Don croisé



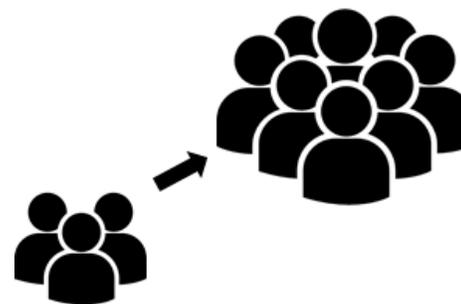
**« Double coïncidence des besoins »**

# Pourquoi ne pas utiliser de la monnaie ?

2 effets d'introduire un « marché du rein » :



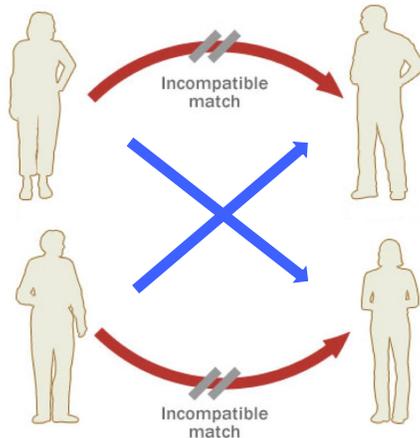
Résoudre la double  
coïncidence des besoins



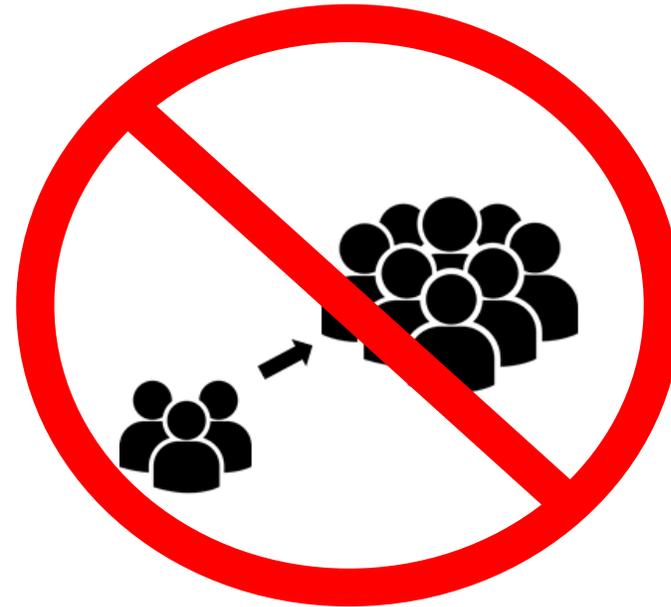
Augmenter l'offre de  
donneurs

# Pourquoi ne pas utiliser de la monnaie ?

2 effets d'introduire un « marché du rein » :



Résoudre la double  
coïncidence des besoins



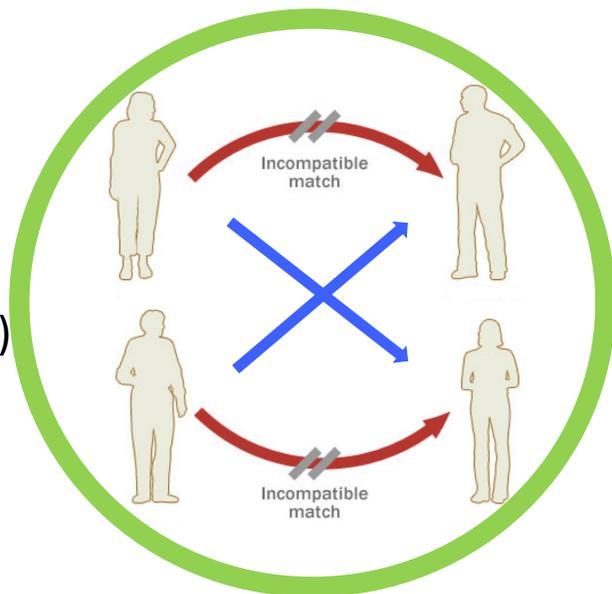
Augmenter l'offre de  
donneurs

- Interdit dans la plupart des pays sauf Iran
- « Repugnant transaction » (Roth, 2006)

# Pourquoi ne pas utiliser de la monnaie ?

2 effets d'introduire un « marché du rein » :

- Peut être résolu sans monnaie
- Mémoire (Kocherlakota, 1998)
- Algorithme « non-croisé » (*Unpaired*)



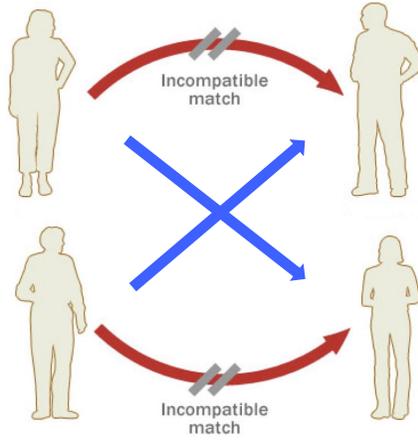
Résoudre la double  
coïncidence des besoins



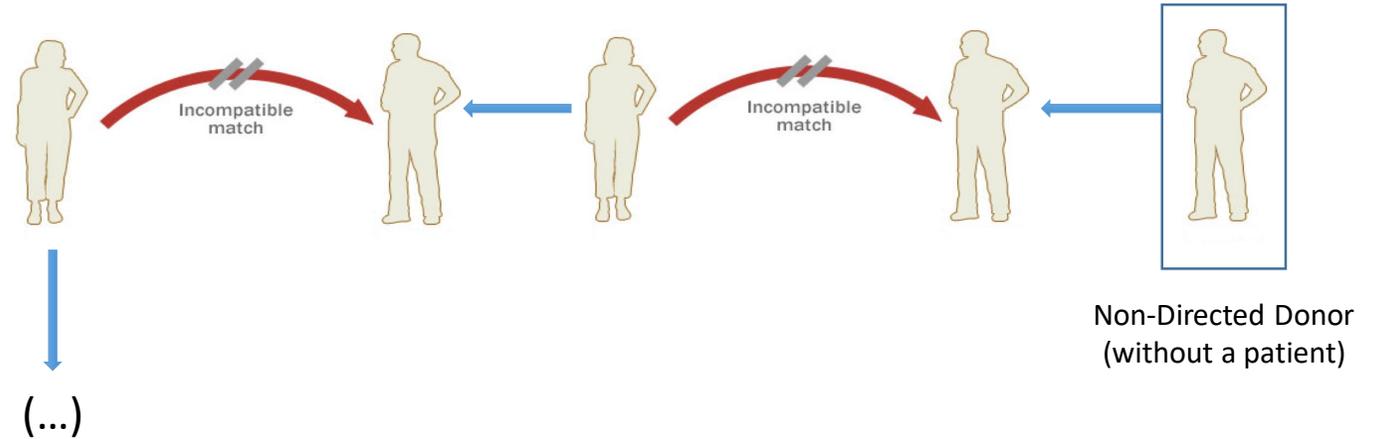
Augmenter l'offre de  
donneurs

- Interdit dans la plupart des pays sauf Iran
- « Repugnant transaction » (Roth, 2006)

# Technologies d'échanges de reins



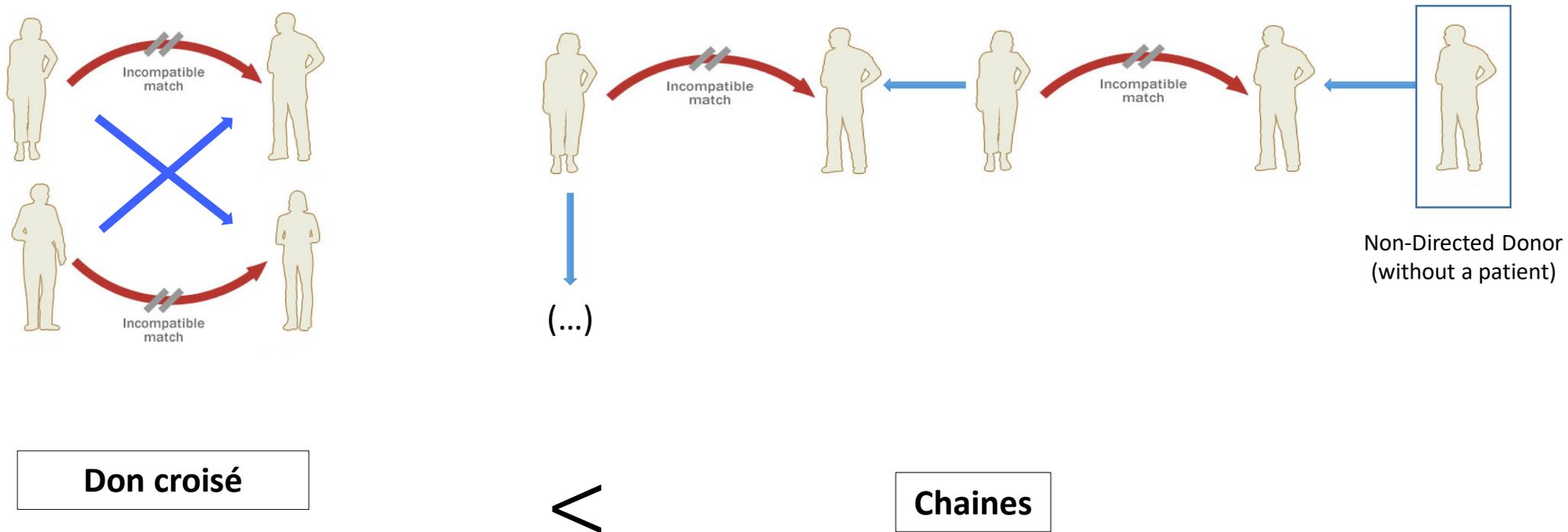
Don croisé



Chaines

Non-Directed Donor  
(without a patient)

# Technologies d'échanges de reins

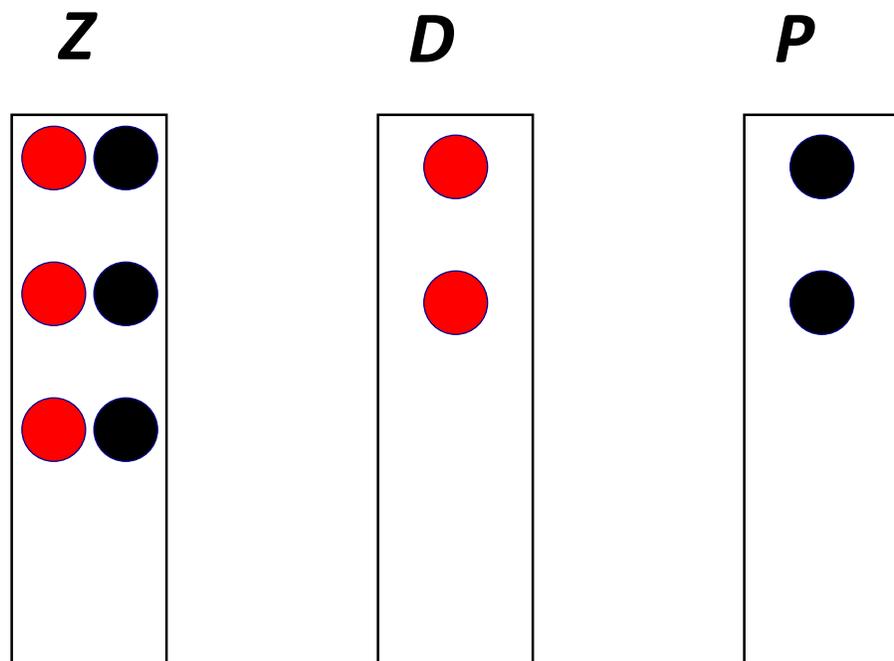


Ashlagi et al. (2012, 2019)

# Proposition: Don non-croisé

- **Algorithme de don non-croisé:** résoudre la double coincidence des besoins
  - Un donneur peut donner avant que son patient reçoive: Pbm de temps d'attente
  - Un patient peut recevoir avant que son donneur donne: Pbm de refus⇒ Réalise les greffes dès qu'elles sont possibles
- **Théorie** (non présenté aujourd'hui)
  - Un modèle dynamique simple:  $W(\text{Optimal}) \approx W(\text{Non croisé}) < W(\text{Chaine}) < W(\text{Croisé})$
- **Empirique**
  - Contre-factuelles sur données françaises
  - Implémentation pratique: modifier l'algorithme en utilisant des donneurs décédés⇒ Résout les 2 problèmes du temps d'attente et du refus

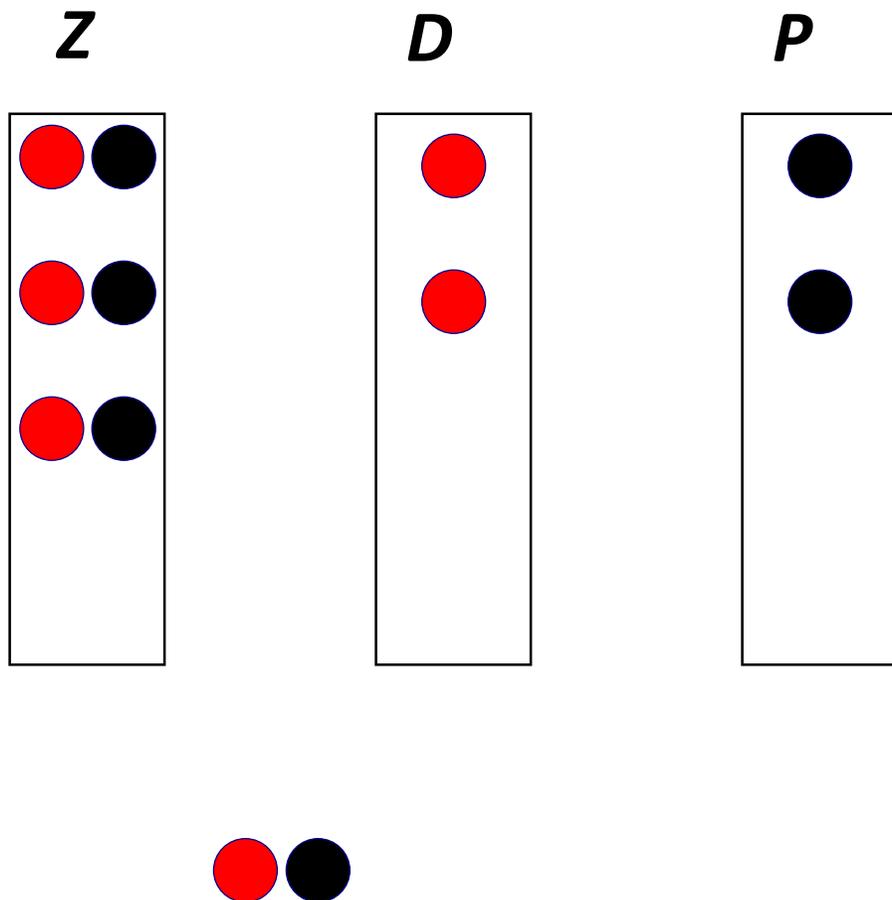
# Algorithme de don non-croisé



*A chaque date t:*

- *Ensemble de donneurs  $D(t)$*
- *Ensemble de patients  $P(t)$*
- *Ensemble de paires  $Z(t)$*

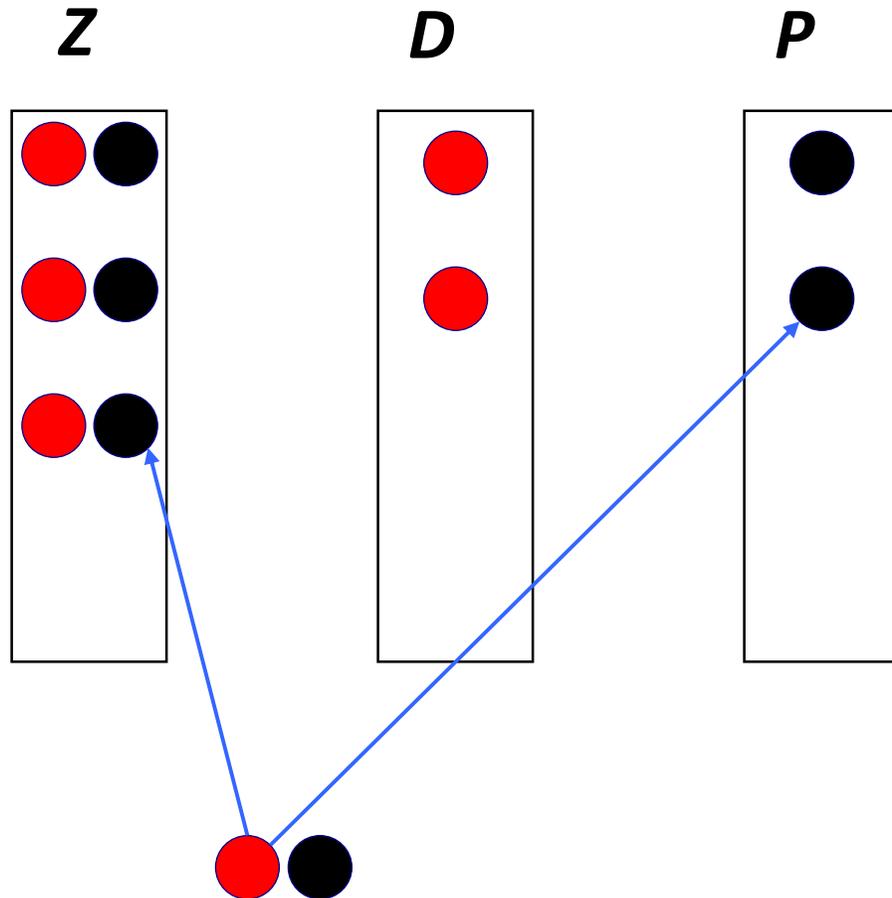
# Algorithme de don non-croisé



Quand une paire patient-donneur arrive...

- Check si le patient peut recevoir d'un donneur de **D** ou **Z**
- Check si le donneur peut donner à un patient de **P** ou **Z**
- Si plusieurs match  $\Rightarrow$  règle de priorité
  1. Donneur dans **D** préféré à un donneur dans **Z**
  2. Patient dans **P** préféré à un patient dans **Z**
  3. Paire dans **Z** préférée à un patient et donneur de paires différentes dans **Z**
- Si indifférence: règle arbitraire

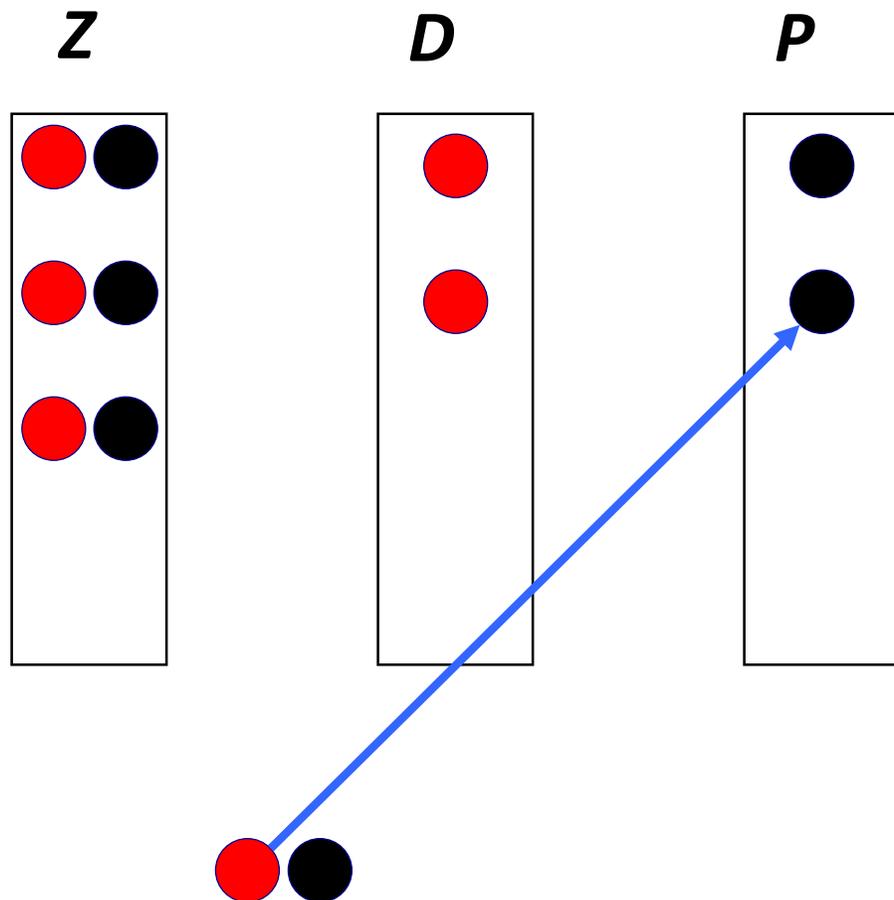
# Algorithme de don non-croisé



Quand une paire patient-donneur arrive...

- Check si le patient peut recevoir d'un donneur de **D** ou **Z**
- Check si le donneur peut donner à un patient de **P** ou **Z**
- Si plusieurs match  $\Rightarrow$  règle de priorité
  1. Donneur dans **D** préféré à un donneur dans **Z**
  2. Patient dans **P** préféré à un patient dans **Z**
  3. Paire dans **Z** préférée à un patient et donneur de paires différentes dans **Z**
- Si indifférence: règle arbitraire

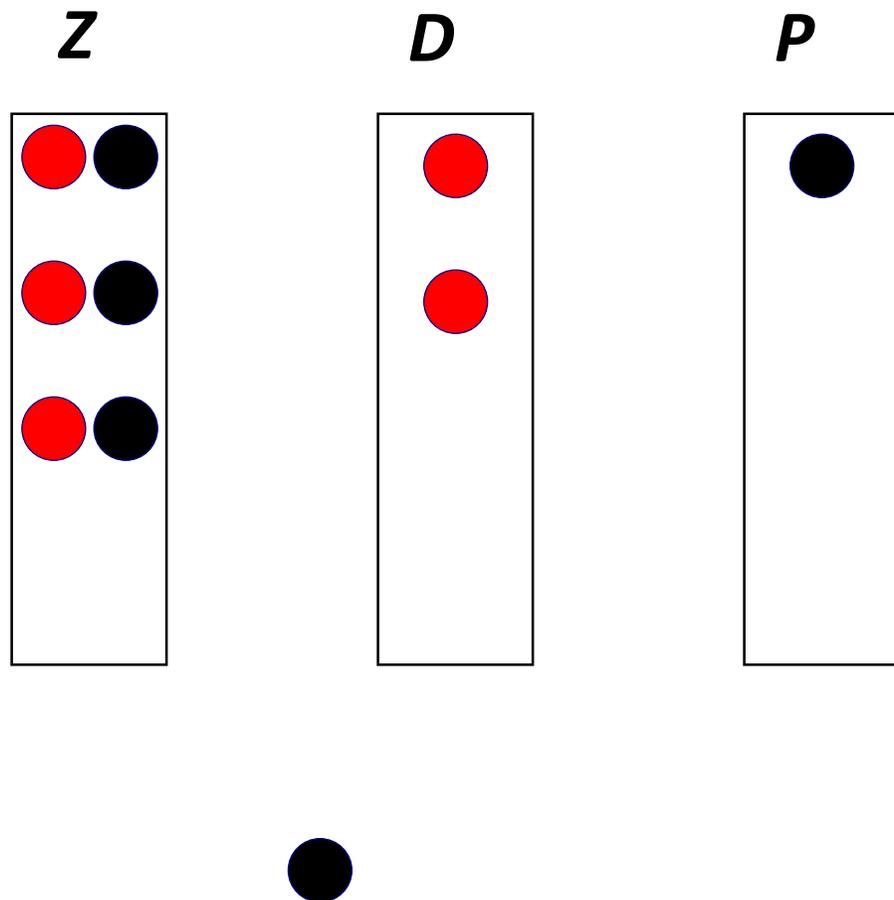
# Algorithme de don non-croisé



Quand une paire patient-donneur arrive...

- Check si le patient peut recevoir d'un donneur de **D** ou **Z**
- Check si le donneur peut donner à un patient de **P** ou **Z**
- Si plusieurs match  $\Rightarrow$  règle de priorité
  1. Donneur dans **D** préféré à un donneur dans **Z**
  2. Patient dans **P** préféré à un patient dans **Z**
  3. Paire dans **Z** préférée à un patient et donneur de paires différentes dans **Z**
- Si indifférence: règle arbitraire

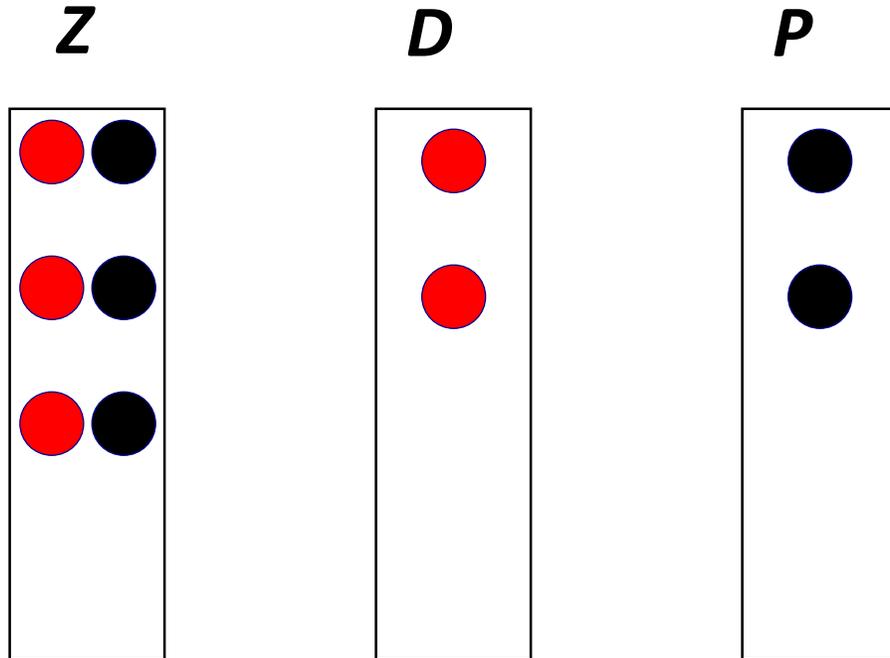
# Algorithme de don non-croisé



Quand une paire patient-donneur arrive...

- Check si le patient peut recevoir d'un donneur de **D** ou **Z**
- Check si le donneur peut donner à un patient de **P** ou **Z**
- Si plusieurs match  $\Rightarrow$  règle de priorité
  1. Donneur dans **D** préféré à un donneur dans **Z**
  2. Patient dans **P** préféré à un patient dans **Z**
  3. Paire dans **Z** préférée à un patient et donneur de paires différentes dans **Z**
- Si indifférence: règle arbitraire

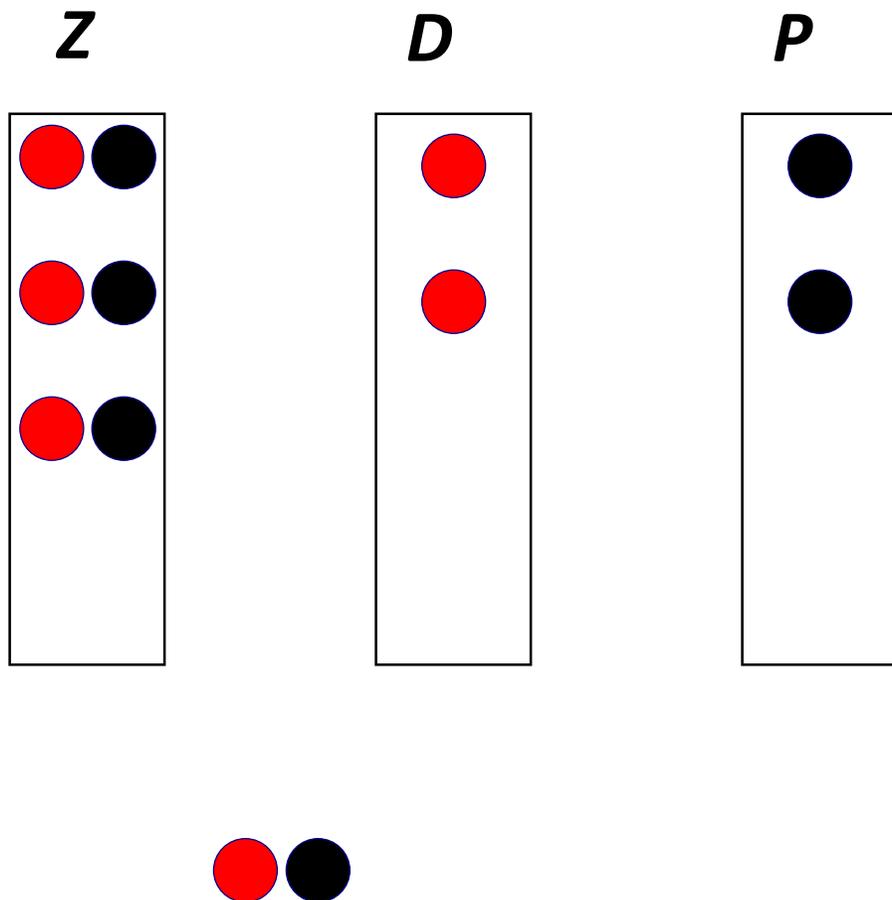
# Algorithme de don non-croisé



Quand une paire patient-donneur arrive...

- Check si le patient peut recevoir d'un donneur de **D** ou **Z**
- Check si le donneur peut donner à un patient de **P** ou **Z**
- Si plusieurs match  $\Rightarrow$  règle de priorité
  1. Donneur dans **D** préféré à un donneur dans **Z**
  2. Patient dans **P** préféré à un patient dans **Z**
  3. Paire dans **Z** préférée à un patient et donneur de paires différentes dans **Z**
- Si indifférence: règle arbitraire

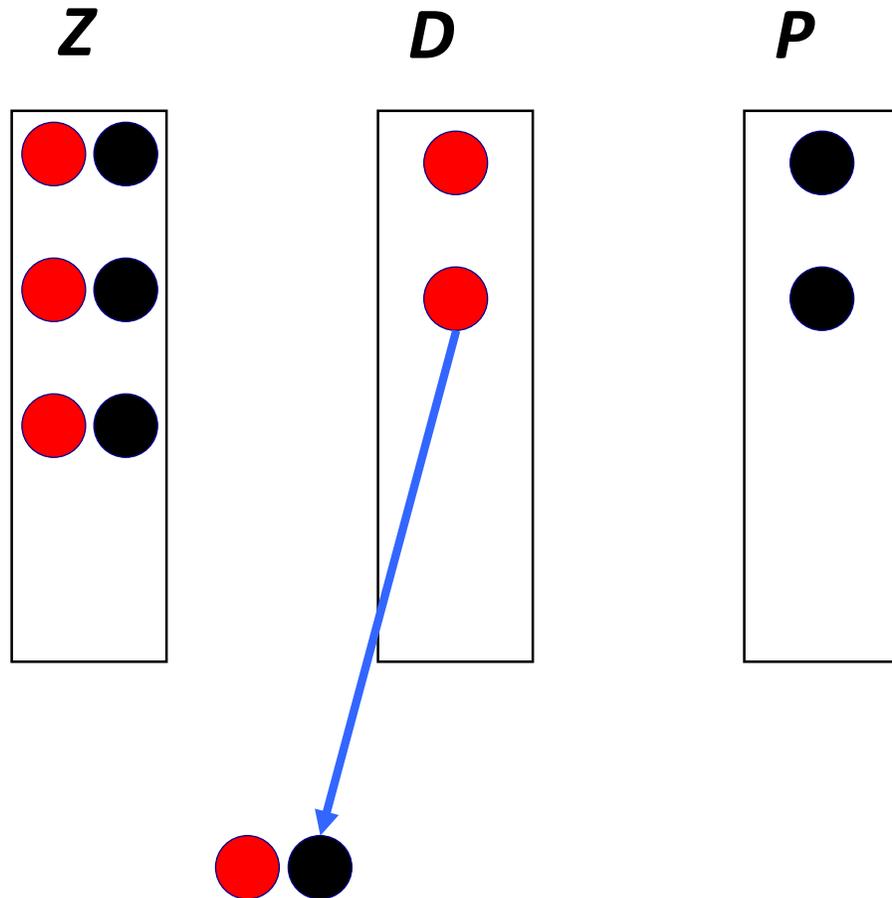
# Algorithme de don non-croisé



Quand une paire patient-donneur arrive...

- Check si le patient peut recevoir d'un donneur de **D** ou **Z**
- Check si le donneur peut donner à un patient de **P** ou **Z**
- Si plusieurs match  $\Rightarrow$  règle de priorité
  1. Donneur dans **D** préféré à un donneur dans **Z**
  2. Patient dans **P** préféré à un patient dans **Z**
  3. Paire dans **Z** préférée à un patient et donneur de paires différentes dans **Z**
- Si indifférence: règle arbitraire

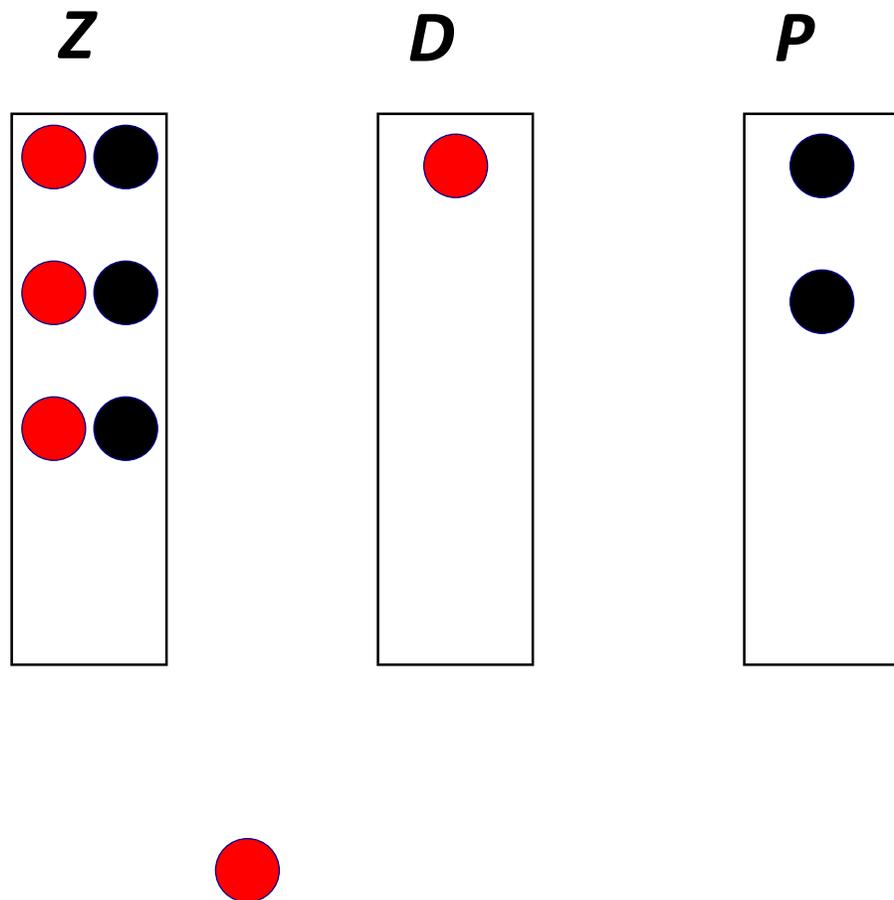
# Algorithme de don non-croisé



Quand une paire patient-donneur arrive...

- Check si le patient peut recevoir d'un donneur de **D** ou **Z**
- Check si le donneur peut donner à un patient de **P** ou **Z**
- Si plusieurs match  $\Rightarrow$  règle de priorité
  1. Donneur dans **D** préféré à un donneur dans **Z**
  2. Patient dans **P** préféré à un patient dans **Z**
  3. Paire dans **Z** préférée à un patient et donneur de paires différentes dans **Z**
- Si indifférence: règle arbitraire

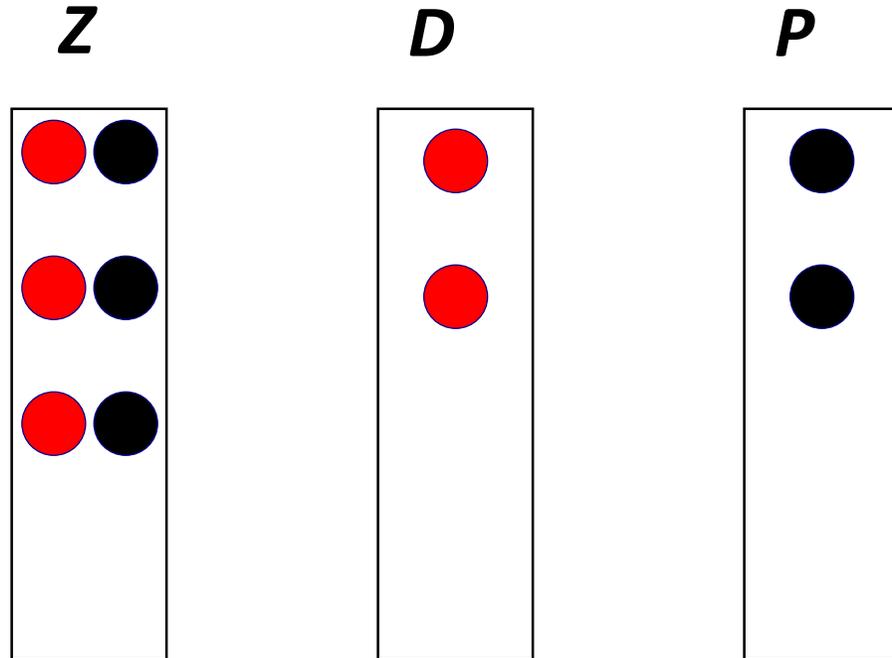
# Algorithme de don non-croisé



Quand une paire patient-donneur arrive...

- Check si le patient peut recevoir d'un donneur de **D** ou **Z**
- Check si le donneur peut donner à un patient de **P** ou **Z**
- Si plusieurs match  $\Rightarrow$  règle de priorité
  1. Donneur dans **D** préféré à un donneur dans **Z**
  2. Patient dans **P** préféré à un patient dans **Z**
  3. Paire dans **Z** préférée à un patient et donneur de paires différentes dans **Z**
- Si indifférence: règle arbitraire

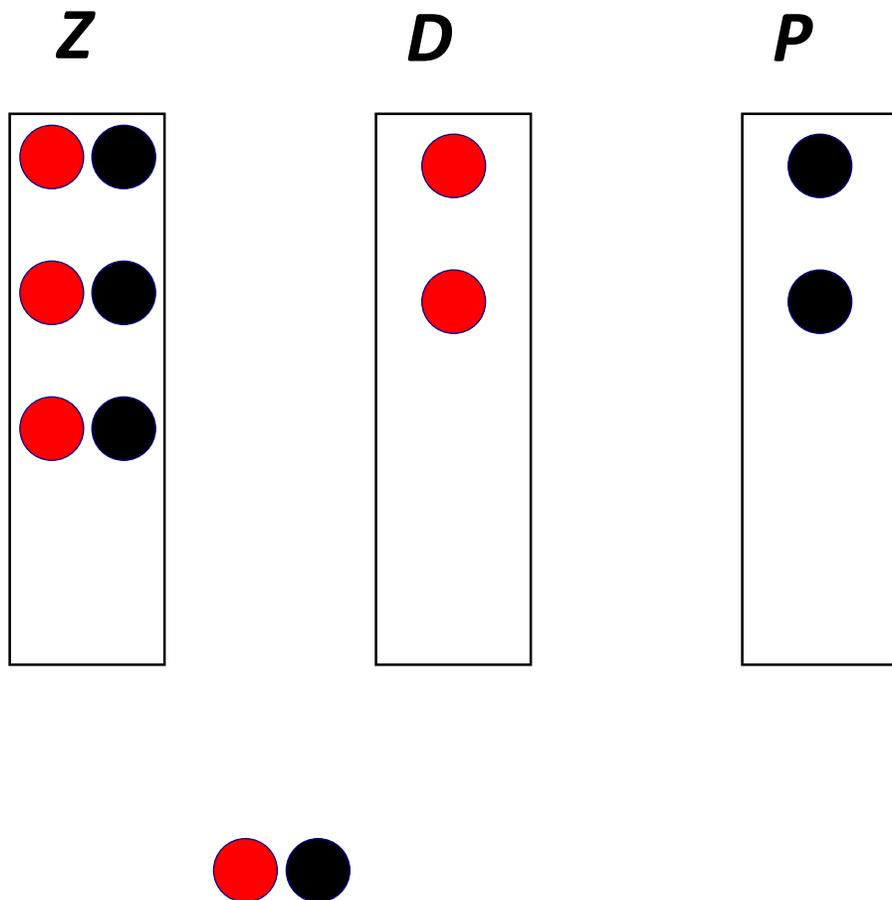
# Algorithme de don non-croisé



Quand une paire patient-donneur arrive...

- Check si le patient peut recevoir d'un donneur de **D** ou **Z**
- Check si le donneur peut donner à un patient de **P** ou **Z**
- Si plusieurs match  $\Rightarrow$  règle de priorité
  1. Donneur dans **D** préféré à un donneur dans **Z**
  2. Patient dans **P** préféré à un patient dans **Z**
  3. Paire dans **Z** préférée à un patient et donneur de paires différentes dans **Z**
- Si indifférence: règle arbitraire

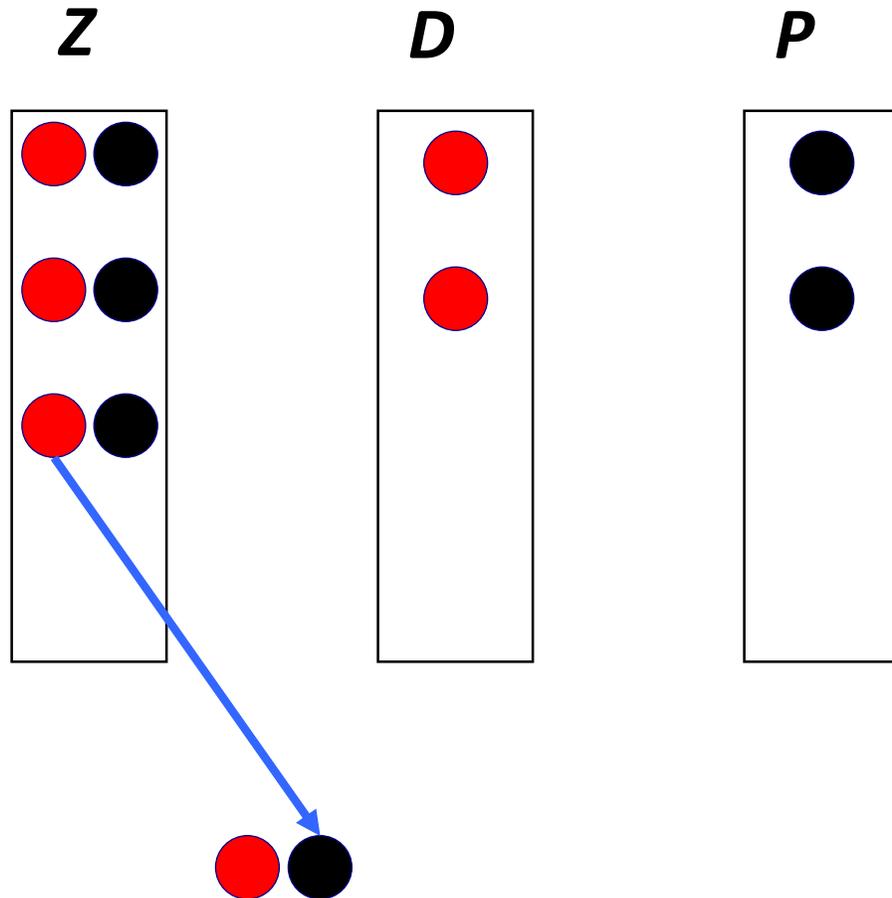
# Algorithme de don non-croisé



Quand une paire patient-donneur arrive...

- Check si le patient peut recevoir d'un donneur de **D** ou **Z**
- Check si le donneur peut donner à un patient de **P** ou **Z**
- Si plusieurs match  $\Rightarrow$  règle de priorité
  1. Donneur dans **D** préféré à un donneur dans **Z**
  2. Patient dans **P** préféré à un patient dans **Z**
  3. Paire dans **Z** préférée à un patient et donneur de paires différentes dans **Z**
- Si indifférence: règle arbitraire

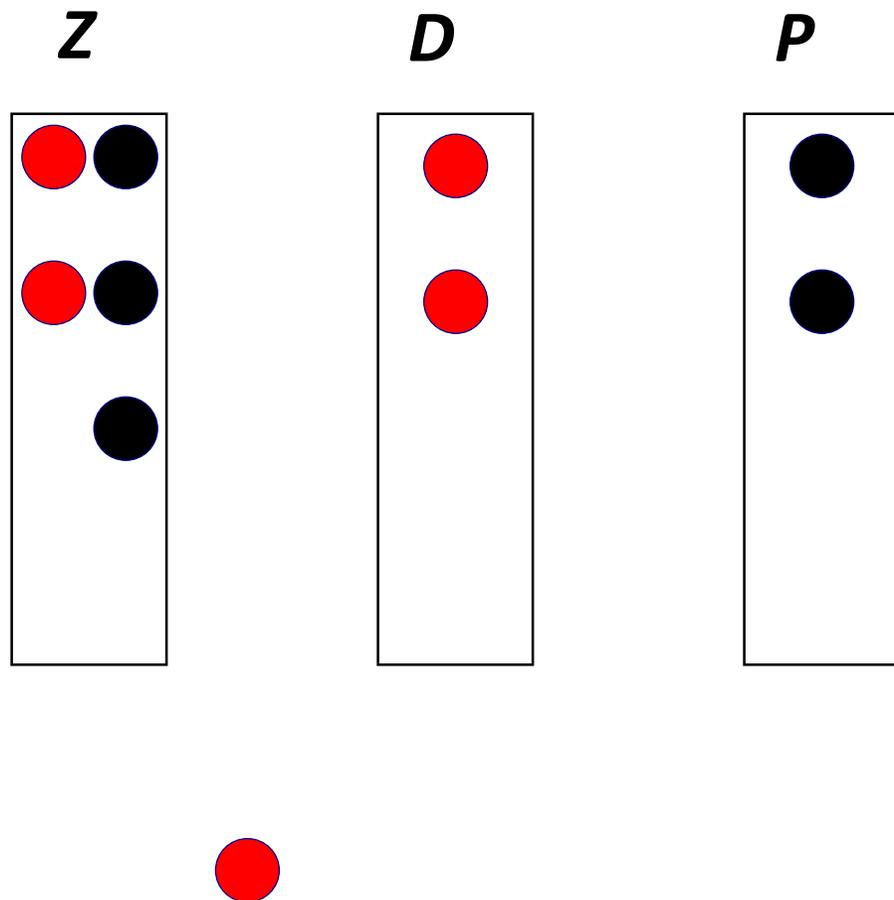
# Algorithme de don non-croisé



Quand une paire patient-donneur arrive...

- Check si le patient peut recevoir d'un donneur de **D** ou **Z**
- Check si le donneur peut donner à un patient de **P** ou **Z**
- Si plusieurs match  $\Rightarrow$  règle de priorité
  1. Donneur dans **D** préféré à un donneur dans **Z**
  2. Patient dans **P** préféré à un patient dans **Z**
  3. Paire dans **Z** préférée à un patient et donneur de paires différentes dans **Z**
- Si indifférence: règle arbitraire

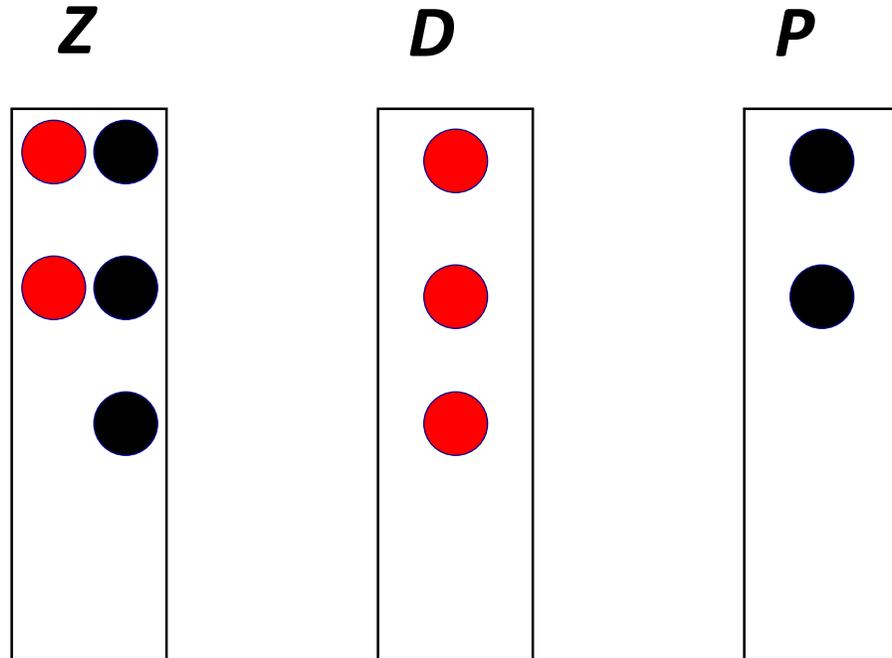
# Algorithme de don non-croisé



Quand une paire patient-donneur arrive...

- Check si le patient peut recevoir d'un donneur de **D** ou **Z**
- Check si le donneur peut donner à un patient de **P** ou **Z**
- Si plusieurs match  $\Rightarrow$  règle de priorité
  1. Donneur dans **D** préféré à un donneur dans **Z**
  2. Patient dans **P** préféré à un patient dans **Z**
  3. Paire dans **Z** préférée à un patient et donneur de paires différentes dans **Z**
- Si indifférence: règle arbitraire

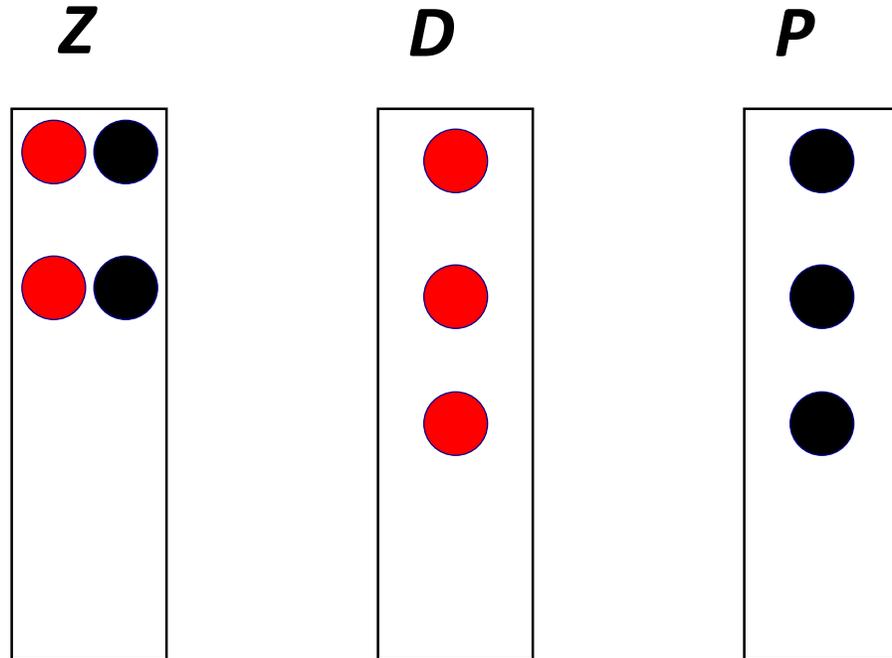
# Algorithme de don non-croisé



Quand une paire patient-donneur arrive...

- Check si le patient peut recevoir d'un donneur de **D** ou **Z**
- Check si le donneur peut donner à un patient de **P** ou **Z**
- Si plusieurs match  $\Rightarrow$  règle de priorité
  1. Donneur dans **D** préféré à un donneur dans **Z**
  2. Patient dans **P** préféré à un patient dans **Z**
  3. Paire dans **Z** préférée à un patient et donneur de paires différentes dans **Z**
- Si indifférence: règle arbitraire

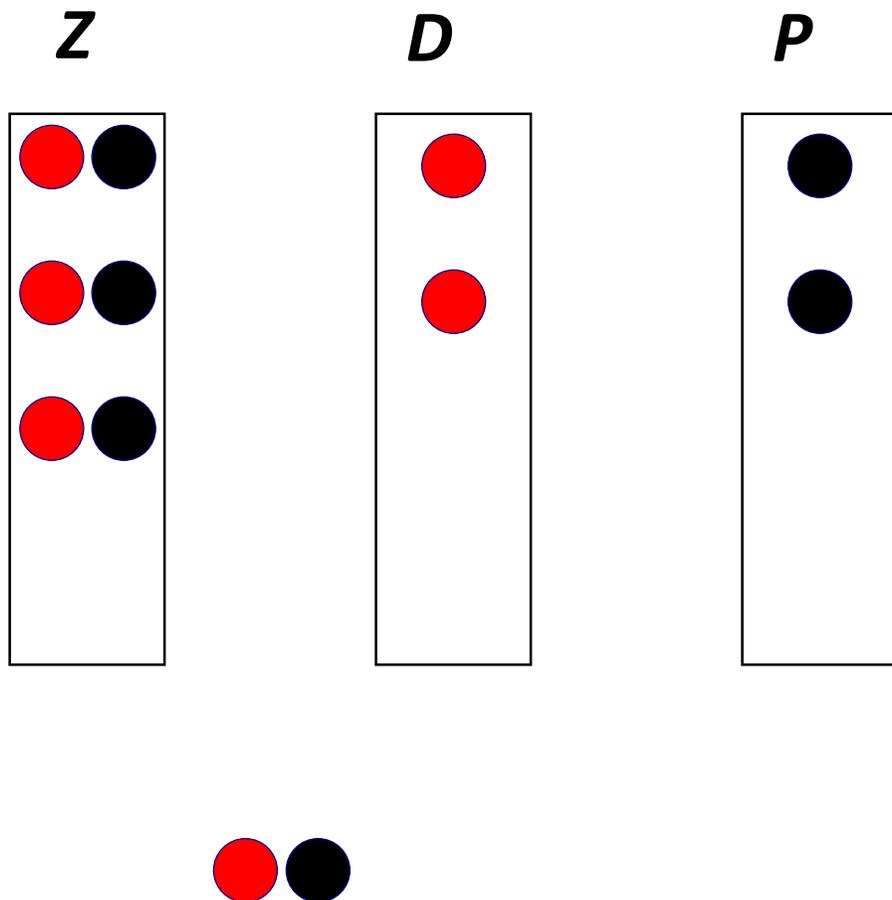
# Algorithme de don non-croisé



Quand une paire patient-donneur arrive...

- Check si le patient peut recevoir d'un donneur de **D** ou **Z**
- Check si le donneur peut donner à un patient de **P** ou **Z**
- Si plusieurs match  $\Rightarrow$  règle de priorité
  1. Donneur dans **D** préféré à un donneur dans **Z**
  2. Patient dans **P** préféré à un patient dans **Z**
  3. Paire dans **Z** préférée à un patient et donneur de paires différentes dans **Z**
- Si indifférence: règle arbitraire

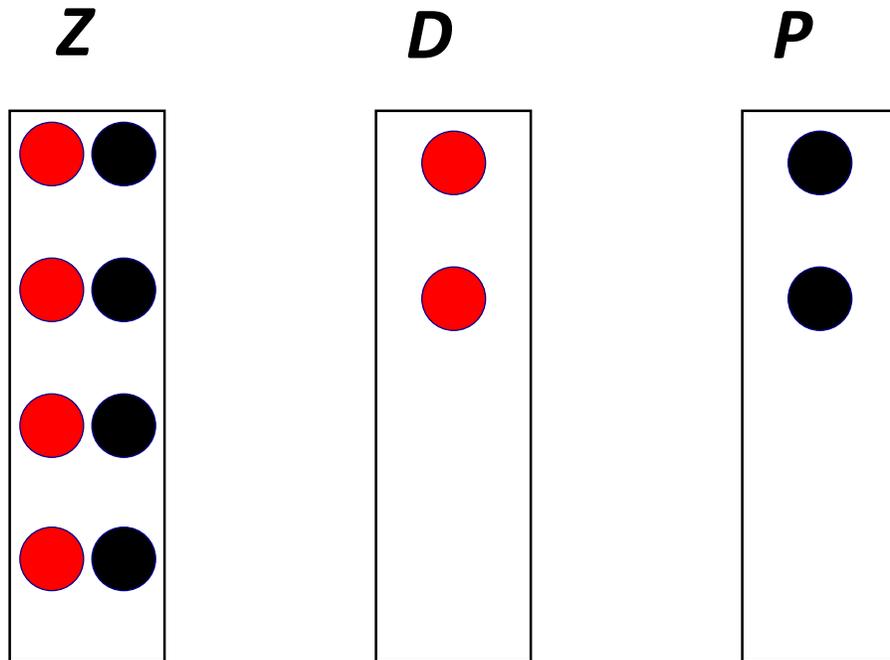
# Algorithme de don non-croisé



Quand une paire patient-donneur arrive...

- Check si le patient peut recevoir d'un donneur de **D** ou **Z**
- Check si le donneur peut donner à un patient de **P** ou **Z**
- Si plusieurs match  $\Rightarrow$  règle de priorité
  1. Donneur dans **D** préféré à un donneur dans **Z**
  2. Patient dans **P** préféré à un patient dans **Z**
  3. Paire dans **Z** préférée à un patient et donneur de paires différentes dans **Z**
- Si indifférence: règle arbitraire

# Algorithme de don non-croisé



Quand une paire patient-donneur arrive...

- Check si le patient peut recevoir d'un donneur de **D** ou **Z**
- Check si le donneur peut donner à un patient de **P** ou **Z**
- Si plusieurs match  $\Rightarrow$  règle de priorité
  1. Donneur dans **D** préféré à un donneur dans **Z**
  2. Patient dans **P** préféré à un patient dans **Z**
  3. Paire dans **Z** préférée à un patient et donneur de paires différentes dans **Z**
- Si indifférence: règle arbitraire

# Empirique

# Objectifs des simulations:

1. Tester la prediction des résultats théoriques:
  - ✓ Différence entre don non-croisé et croisé ?
  - ✓ Différence entre don non-croisé et “optimal” ?
2. Résoudre deux problèmes pratiques:
  - ✓ Temps d’attente des patients seuls: ***P***
  - ✓ Temps d’attente des donneurs seuls: ***D***

# Objectifs des simulations:

## 1. Tester la prediction des résultats théoriques:

- ✓ Différence entre don non-croisé et croisé ?
- ✓ Différence entre don non-croisé et “optimal” ?

## 2. Résoudre deux problèmes pratiques:

- ✓ Temps d’attente des patients seuls: ***P***
- ✓ Temps d’attente des donneurs seuls: ***D***

**Analyse contre-factuelle** sur  
les données du don croisé  
Français

# Objectifs des simulations:

## 1. Tester la prediction des résultats théoriques:

- ✓ Différence entre don non-croisé et croisé ?
- ✓ Différence entre don non-croisé et “optimal” ?

**Analyse contre-factuelle** sur les données du don croisé Français

## 2. Résoudre deux problèmes pratiques:

- ✓ Temps d’attente des patients seuls: ***P***
- ✓ Temps d’attente des donneurs seuls: ***D***

**Résultat:** ces deux problèmes peuvent être résolus **avec la bonne modification de l’algorithme**

# Programme de don croisé français

- Le programme a commencé en Decembre 2013
- **Uniquement des échanges entre 2 paires** (*pairwise exchanges*)
- Fréquence: **match-run** tous les 3 mois
- Centralisé à l'échelle nationale par l'Agence de la Biomédecine (ABM)

# Statistiques descriptives

- De Dec. 2013 à Feb. 2018: 15 match-runs
- 78 paires ont participé au total depuis le début du programme
  - ✓ Taille moyenne du pool: 18 paires par match-run
- Pourcentage de patients greffés: 15% (12 transplantations)
- Méthode alternative: “désensibilisation” 540 greffes incompatibles durant cette période

# Les données

- **Programme don croisé français:**
  - ✓ **Matrice de compatibilité** entre les paires ayant participé
  - ✓ Date et composition de chaque match-run
  
- **Offre de la liste des donneurs décédés (LDD):**
  - ✓ Dates des offres
  - ✓ Matrice de compatibilité avec tous les patients
  - ✓ Information sur la qualité de chaque donneur (KDRI)

# Commentaires: Arrivées

- Pas de dates exactes d'arrivée dans les données
  - ✓ Mais on observe quelles paires participent à quel match-run
- Arrivée séquentielle des paires
- Pour chaque paire, on tire une date d'arrivée aléatoire
  - ✓ Consistante avec l'information de la composition des match-runs
  - ✓ Seule source d'aléas

# Commentaires: Départs

- Départs tous dus à:
  - ✓ Transplantation depuis un donneur décédé
  - ✓ Désensibilisation
- Pas de morts
  - ✓ Patients dans le programme don croisé en relative bonne santé
- Départs (partiellement) dus à la faible performance du programme
  - ✓ Un algorithme avec un pourcentage de greffes plus élevé aurait moins de départs

# Commentaires: Départs

- Nous considérons 2 scénarii:

✓ **Scenario 1:** pas de départs

✓ **Scenario 2:** départs comme dans les données

- Réalité certainement “entre les deux”

# Méthode: analyse contrefactuelle

- **Etude rétrospective:** Que ce serait-il passé si on avait utilisé l'algorithme de don non-croisé à partir de Décembre 2013?
- 1000 iterations des dates d'arrivées
- Echanges possibles déterminés par la matrice de compatibilité
- Benchmark:
  - ✓ Pas de départs
  - ✓ Pas de refus des donneurs

# Méthode: analyse contrefactuelle

- Pour chaque tirage des dates d'arrivées: test de 5 algorithmes
  1. Don croisé entre 2 paires
  2. Don croisé entre 2 ou 3 paires
  3. Don croisé entre 2 paires + chaine
  4. Don non-croisé
  5. "Omniscient": minimise le temps d'attente sur chaque réalisation

# Resultats I: algorithme de don non-croisé



	Don croisé 2 paires	Don croisé 2 ou 3 paires	Chaine (+ don croisé)	Don non-croisé	Omniscient
Nb. de greffes					
% de greffes					
Temps d'attente (jours)					

# Resultats I: algorithme de don non-croisé



	Don croisé 2 paires	Don croisé 2 ou 3 paires	Chaine (+ don croisé)	Don non-croisé	Omniscient
Nb. de greffes	22.74				
% de greffes	29%				
Temps d'attente (jours)	766.03				

# Resultats I: algorithme de don non-croisé



	Don croisé 2 paires	Don croisé 2 ou 3 paires	Chaine (+ don croisé)	Don non-croisé	Omniscient
Nb. de greffes	22.74	23.2	26.7		
% de greffes	29%	30%	34%		
Temps d'attente (jours)	766.03	731.88	725.9		

# Resultats I: algorithme de don non-croisé



	Don croisé 2 paires	Don croisé 2 ou 3 paires	Chaine (+ don croisé)	Don non-croisé	Omniscient
Nb. de greffes	22.74	23.2	26.7	44.5	
% de greffes	29%	30%	34%	58%	
Temps d'attente (jours)	766.03	731.88	725.9	460	

# Resultats I: algorithme de don non-croisé

	Don croisé 2 paires	Don croisé 2 ou 3 paires	Chaine (+ don croisé)	Don non-croisé	Omniscient
Nb. de greffes	22.74	23.2	26.7	44.5	45
% de greffes	29%	30%	34%	57%	58%
Temps d'attente (jours)	766.03	731.88	725.9	460	446

Taux de greffe similaire à l'omniscient mais...

# Resultats I: algorithme de don non-croisé



	Don croisé 2 paires	Don croisé 2 ou 3 paires	Chaine (+ don croisé)	Don non-croisé	Omniscient
Nb. de patient dans P	0	0	0	24.5	27.9
Temps dans P (médiane)	0	0	0	224	498
Temps dans P (90 <sup>th</sup> perc)	0	0	0	1143	1438
Nb. de donneurs dans D	0	3.7	0	22.3	27.5
Temps dans D (médiane)	0	86	0	351	513
Temps dans D (90 <sup>th</sup> perc)	0	1436	0	976	1415

... problemes du temps d'attente dans P et du refus sont réels

# Comment réduire le temps d'attente ?

- 1er problème: réduire le temps d'attente des patients dans P

## **Idée:** Utiliser les offres de donneurs décédés

- ✓ Un patient dans P reçoit une priorité sur la liste d'attente des donneurs décédés
- ✓ On filtre les donneurs décédés pour garder ceux de bonne qualité (slide suivant)
- ✓ Les données contiennent l'ensemble des offres de donneurs décédés
- ✓ Nous supposons qu'un patient accepte la première offre faite

# Comment réduire le temps d'attente ?

- Les donneurs décédés offrent une moins bonne qualité de greffe
- **Filtrage:** Nous considérons uniquement les donneurs décédés de qualité suffisante pour être acceptés par un patient dans P
  - ✓ Mesure de qualité: Kidney Donor Risk Index (KDRI)
    - Indice de risque d'échec post-opératoire de la greffe
  - ✓ Calcule du KDRI également pour chaque donneur vivant associé à chaque patient
  - ✓ **On propose un donneur décédé uniquement si son risque est moins élevé que le donneur associé au patient**

# Comment réduire le temps d'attente ?

- **Problème:**

- ✓ Nous prenons des donneurs de bonne qualité de la liste des donneurs décédés
- ✓ Externalité négative sur les patients attendant sur la liste d'attente

- **Solution:**

- ✓ Pour utiliser un donneur décédé pour un patient dans P, il faut “redonner” immédiatement un donneur vivant dans D

# Resultats II: LDD pour les patients dans P

	Don croisé 2 paires	Don croisé 2 ou 3 paires	Chaine (+ don croisé)	Don non-croisé
Nb. de greffes	22.74	23.2	26.7	67.4 (+22)
Nb. de greffes de donneurs vivants	22.74	22.2	26.7	39.5 (-5)
Temps d'attente (jours)	766.03	731.88	725.9	161 (-285)

Légère baisse des dons de donneurs vivants mais augmentation totale

# Resultats II: LDD pour les patients dans P

	Don croisé 2 paires	Don croisé 2 ou 3 paires	Chaine (+ don croisé)	Don non-croisé
Nb. de patient dans P	0	0	0	29.5
Temps dans P (médiane)	0	0	0	2 (-496)
Temps dans P (90 <sup>th</sup> perc)	0	0	0	138 (-1300)
Nb. De donneurs dans D	0	3.7	0	31.9
Temps dans D (médiane)	0	86	0	7
Temps dans D (90 <sup>th</sup> perc)	0	1436	0	104

... Temps d'attente des patients dans P chute (médiane à 2 jours)

# Resultats II: LDD pour les patients dans P

	Don croisé 2 paires	Don croisé 2 ou 3 paires	Chaine (+ don croisé)	Don non-croisé
Nb. de patient dans P	0	0	0	29.5
Temps dans P (médiane)	0	0	0	2
Temps dans P (90 <sup>th</sup> perc)	0	0	0	138
Nb. De donneurs dans D	0	3.7	0	31.9
Temps dans D (médiane)	0	86	0	7 (-506)
Temps dans D (90 <sup>th</sup> perc)	0	1436	0	104 (-1311)

... Temps d'attente des donneurs dans D chute (médiane at 7 jours)

# Concernant le refus des donneurs dans D

- Les résultats sont les mêmes si on simule une proba de refus de 0.17% par jour
- Possible modification de l'algorithme pour éviter les refus:

**Idée:** Donneur donne dès que son patient reçoit

- ✓ Donne son rein à un patient sur la liste d'attente des donneurs décédés
- ✓ Autorise d'utiliser un rein de donneur décédé dans le futur

⇒ **Résultats similaires mais sans avoir de donneurs seuls dans D**

# Conclusion

- Robustesse:
  - ✓ Départs
  - ✓ Refus des donneurs dans D
  - ✓ Chaines multiples
  - ✓ Pools plus grands

# Conclusion

- **Théorie:**  $W(\text{Optimal}) \approx W(\text{Non-croisé}) < W(\text{Chaine}) < W(\text{Croisé})$
- **Empirique:**
  - ✓ Résultats théoriques confirmés
  - ✓ La version pratique de l'algorithme de don non-croisé permet de résoudre le problème du temps d'attente et du refus des donneurs...
  - ✓ ... tout en greffant plus de 80% des patients à un donneur vivant ou un donneur décédé de bonne qualité.

# Loi de la bioéthique: article 53

- Travail avec ABM pour modifier le programme don croisé
  - ✓ Relacher la simultanéité imposée par le don croisé
- Travaux de simulations:

## **(Note IPP n°41) Perspectives sur le programme de dons croisés de reins en France**

- Inclus dans l'article 53 du projet de loi de la bioéthique
  - ✓ Autorise l'utilisation de donneurs décédés ( $\approx$  patients dans P)
  - ✓ Relacher la simultanéité mais contrainte de 24h
  - ✓ Détails par décret

# Réforme: pistes possibles

- Identifier les patients qui accepteraient que leur donneur donne immédiatement (= entrer dans P)
- Ces patients obtiennent une priorité élevée sur la LDD (à définir probablement juste en dessous “super urgences”)
- Autoriser don croisé 2 à 4 paires + chaines partant du donneur dont le patient accepte de rentrer dans P  
⇒ Maximiser le nombre de greffes

# Réforme: pistes possibles

- Si une chaîne/cycle commence: toutes les opérations en 24h
- Le donneur en fin de chaîne devrait redonner à la LDD (possible ?)

# Réforme: lien avec l'algorithme "non-croisé"

## Points communs:

- Donneurs peuvent donner avant que leur patient reçoive
- Patients peuvent recevoir avant que leur donneur ne donne  
(= P et D non vides)
- Utiliser la LDD pour minimiser l'attente des patients dans P

# Réforme: lien avec l'algorithme “non-croisé”

## **Différences:**

- Possible périodicité (ex: tous les mois)
- Donneur dans D ne peut rester que 24h
- Un patient dans P ne peut recevoir que de la LDD et non d'un donneur vivant dans le futur