Simulations de Données Génomiques à grande échelle
Quelques Outils

Bertrand Servin

Réunion Rules and Tools, 23 Sept. 2010

Simuler pour quoi ?

- Production de données parfaites:
  - Tester de nouvelles méthodes dans des situations entièrement connues
  - Etudier les comportements de systèmes génétiques complexes aux propriétés analytiquement indérivables ...
- Test d'hypothèse: Dériver empiriquement les distributions des statistiques sous une hypothèse nulle (en général)
- Dans les deux cas: importance de simuler des données (i) réalistes et (ii) à grande échelle.

En Avant et à Reculons

Deux classes de simulations

Backward: Coalescence

- **Avantages**
  - Rapidité (cf. Simon): seules les généalogies nécessaires et seuls les polymorphismes actuels sont simulés.
  - Obtention d'échantillons à l'équilibre mutation / dérive (par définition)
  - Possibilité de simuler des scenarios relativement compliqués
- **Inconvénients**
  - Modèle d'évolution de Wright-Fisher: limitations des scénarios possibles
  - Simulation de la sélection: restreinte

En Avant et à Reculons

Deux classes de simulations

Forward

- **Avantages**
  - Possibilité de simuler des scénarios arbitrairement complexes (pedigree, sélection, phénotypes ...)
- **Inconvénients**
  - Population de départ (?)
  - Peu efficaces pour créer des populations en HWE (multilocus !)

En Avant et à Reculons

Deux classes de simulations

Le meilleur des deux mondes

- **Avantages**
  - Rapidité (cf. Simon): seules les généalogies nécessaires et seuls les polymorphismes actuels sont simulés.
  - Obtention d'échantillons à l'équilibre mutation / dérive (par définition)
  - Possibilité de simuler des scénarios arbitrairement complexes (pedigree, sélection, phénotypes ...)
- **Inconvénients**
  - Aucun :)

`ms`: l'historique

- Ecrit par Richard R. Hudson. Distribué sous forme de code source (C).
- Aussi: `cosi` (Shaffner). Similaire, paramétrage plus simple, adaptable à des cartes génétiques arbitrairement complexes.
- Exemple (`ms`):

Ne	1000 diploïdes
c	10^{-8} (1 cM/Mb)
μ	10^{-8}/base (mais peut être fixé aussi)
n	1000 haplotypes simulés
commande	`ms 1000 1 -t 40 -r 40 1000000` (pour L=1Mb)

L (Mb)	1	5	10	50
t (sec)	0.018	0.12	0.4	∞(?)

A partir d'une certaine taille de segment, la création de lignées par recombinaison ralentit de manière extrême la convergence.

Simulations Tout Génome

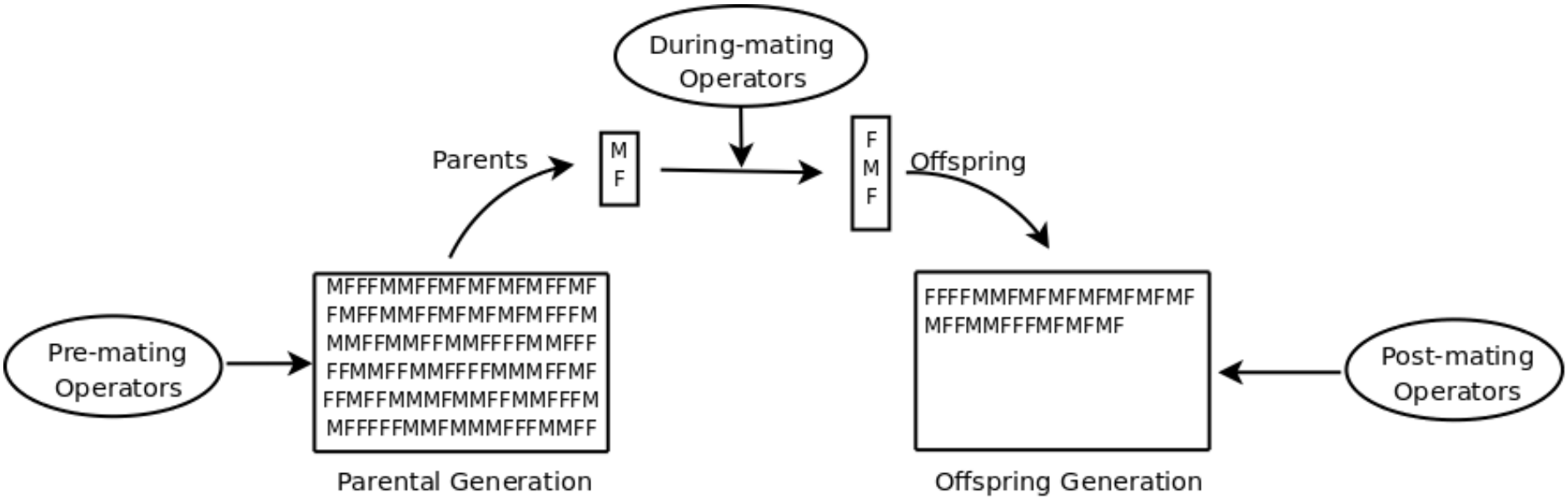
- Résoudre les problèmes de convergence en ignorant les recombinaisons trop proches:
  - considérer un chromosome comme un chapelet de zones non-recombinantes (`simcoal`, `msms`,`ms`)
  - **ou** Simuler des points chauds de recombinaison (`cosi`,`mshot`)
  - $+$ autoriser les évènements multiples de coalescence et restreindre le nombre max de lignées ($<$ taille de pop) (`genome`).
- Exemple précédent pour un chromosome de 100 Mb prend $\sim 10 - 30$ secondes. (testé avec `ms`, `msms`, `genome` et `cosi` (hotspots)).

Simulations Tout Génome

- Résoudre les problèmes de convergence en ignorant les recombinaisons trop proches:
  - considérer un chromosome comme un chapelet de zones non-recombinantes (`simcoal`, `msms`,`ms`)
  - **ou** Simuler des points chauds de recombinaison (`cosi`,`mshot`)
  - $+$ autoriser les évènements multiples de coalescence et restreindre le nombre max de lignées ($<$ taille de pop) (`genome`).
- Exemple précédent pour un chromosome de 100 Mb prend $\sim 10 - 30$ secondes. (testé avec `ms`, `msms`, `genome` et `cosi` (hotspots)).
- Autre option `fastPHASE -U`: estime le modèle sur des données réelles et simule des haplotypes conditionnellement aux paramètres. Ne pas utiliser de méthodes basées sur ce modèle derrière (fastPHASE, beagle ...), c'est de la triche.

simuPOP: introduction

- Simulateur de populations forward
- Ensemble de *briques* qui permettent de simuler des processus arbitrairement complexes.
- Programmation python, documentation très fournie.



Exemple: Split et Sélection

- Simulation de deux populations à partir d'une pop ancestrale (haplotypes générés avec `ms`)
- Dans une des pops, sélection pour un locus au milieu de la région
- Stocker les fréquences alléliques des locus pour chaque population

Exemple: Split et Sélection

```
def simulPop(haps,rho,s,nGener,n1=1000,n2=1000):
    nLoc=len(haps[0])
    pop=simp.Population(size=[n1,n2], loci=nLoc,
                        infoFields=['fitness'])
    simp.initGenotype(pop, haplotypes=haps)
    pop.evolve(
        initOps = simp.InitSex(),# f(male)=0.5
        preOps = simp.MapSelector(
            loci=nLoc/2,
            fitness={(0,0):1,(0,1):1+s,(1,1):(1+s)**2},
            subPops=0),
        matingScheme = simp.RandomMating(
            ops = [simp.Recombinator(rates=rho)]),
        finalOps = [simp.Stat(alleleFreq=range(nLoc),
                              vars=['alleleFreq_sp'])],
        gen = nGener)
    return pop
```

Exemple: Evolution de fréquence conditionellement à un pedigree donné

- Lignées CMJR Porc: Pedigree connu
- Objectif: simuler l'évolution des fréquences alléliques dans ce pedigree.
- Fichier pedigree: `id pere mere sexe affecte`

Exemple: Evolution de fréquence conditionellement à un pedigree donné

```
def simulate(pedfic,p0):
    ped=sim.loadPedigree(pedfic)
    N=ped.ancestralGens() ## pedcheck omis
    # Identification des  fondateurs
    IDs = [x.ind_id for x in ped.allIndividuals(ancGens=N)]
    sex = [x.sex() for x in ped.allIndividuals(ancGens=N)]
    # Simulation d'un seul locus
    pop = sim.Population(size=len(IDs), loci=1,
                         infoFields='ind_id')
    sim.initInfo(pop, IDs, infoFields='ind_id')
    sim.initSex(pop, sex=sex)
    pop.evolve(
        initOps=sim.InitGenotype(freq=[p0,1-p0]),
        matingScheme=sim.PedigreeMating(ped,
                        ops=sim.MendelianGenoTransmitter()),
        postOps= [ sim.Stat(alleleFreq=[0])], gen=N )
    return pop.vars()['alleleFreq'][0][0]
```

simuPop: Conclusions

- simuPOP: génétique des Populations
- Permet de simuler des scénarios complexes et d'être greffé à un simulateur de coalescents.
- Très bonne documentation, mailing list active.
- `Python` :)
- Cas de la sélection sur index: faisable mais dur.

QMSim

- Simulateurs de populations animales (QTL)
- Population fondatrice créée par simulations forward
- Calcul des EBV est intégré
- Croisements aléatoires ou optimisés / consanguinité
- Paramétrage par fichier
- Code source non fourni. Greffage à un simulateur de coalescents impossible.

Conclusion

- coalescent (ms, cosi ...) + simuPop : couteau suisse. Phase d'apprentissage nécessaire.
- Simulations "Génétique Animale" classiques : QMSim (LDSO ?) mais pop ancestrale créée par forward.