

Musique générative

Examples

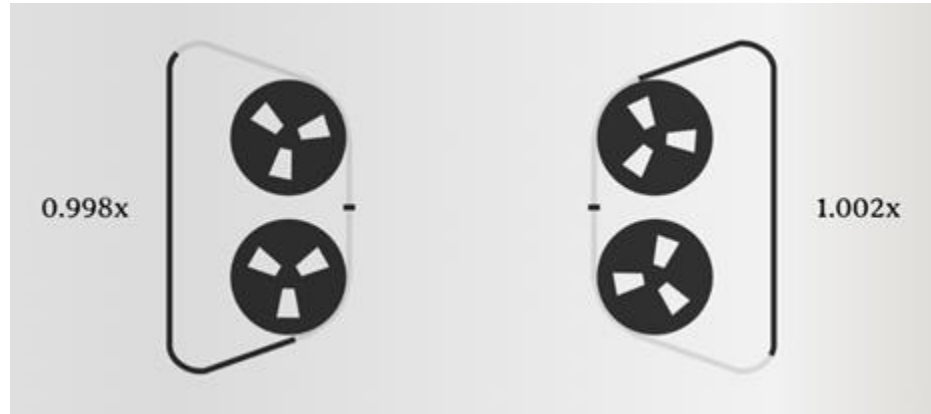


=>



• <https://teropa.in/fo/loop/#/itsgonnarain>

• <https://teropa.in/fo/loop/#/reichphasing>



Software System

- <https://teropa.info/loop/#/stochasticdrummachine>
- <https://teropa.info/loop/#/markovplayer>

Une implémentation en C d'un générateur de musique

.Objectifs :

- Tenter de mieux comprendre le processus créatif de la musique générative
- S'inspirer des architectures de composition musicale pour construire le système
- Apporter une touche personnelle pertinent au projet

Principes de base et fonctionnement

.2 partitions : structures et probabilités

.Restrictions :

- Gamme Do Majeur
- Signature 4/4
- Octaves de 3 à 6 inclus
- Rythme des progressions harmoniques fixé
- Pas d'inversion ou de tensions d'accords

Structures

• Note : vecteur de 3 valeurs

- Fréquence
- Durée
- Vitesse

```
enum key {C3, D3, E3, F3, G3, A3, B3,  
          C4, D4, E4, F4, G4, A4, B4,  
          C5, D5, E5, F5, G5, A5, B5,  
          C6, D6, E6, F6, G6, A6, B6,  
          Silent}; //On est sur la gamme C/a  
typedef enum key key;  
  
enum rythe {full, half, quarter, eighth, sixteenth};  
typedef enum rythe rythe;  
  
enum chord {C, d, e, F, G, a, bdim};  
typedef enum chord chord;  
  
struct note {  
    key pitch;  
    rythe duration;  
    unsigned velocity; //De 0 à 127  
    struct note *next;  
};  
typedef struct note note;  
  
struct channel {  
    note *head;  
};  
typedef struct channel channel;  
  
struct measure {  
    channel* channels[10]; //On permet jusqu'à 10 canaux sonores simultanés par mesure : 5 pour la mélodie, 5 pour les accords  
};  
typedef struct measure measure;  
  
struct staff {  
    measure* measures[4];  
};  
typedef struct staff staff;
```

• Canal : liste chaînée de notes

• Mesure : Ensemble de 10 canaux

• Portée : Ensemble de 4 mesures

Spécificités

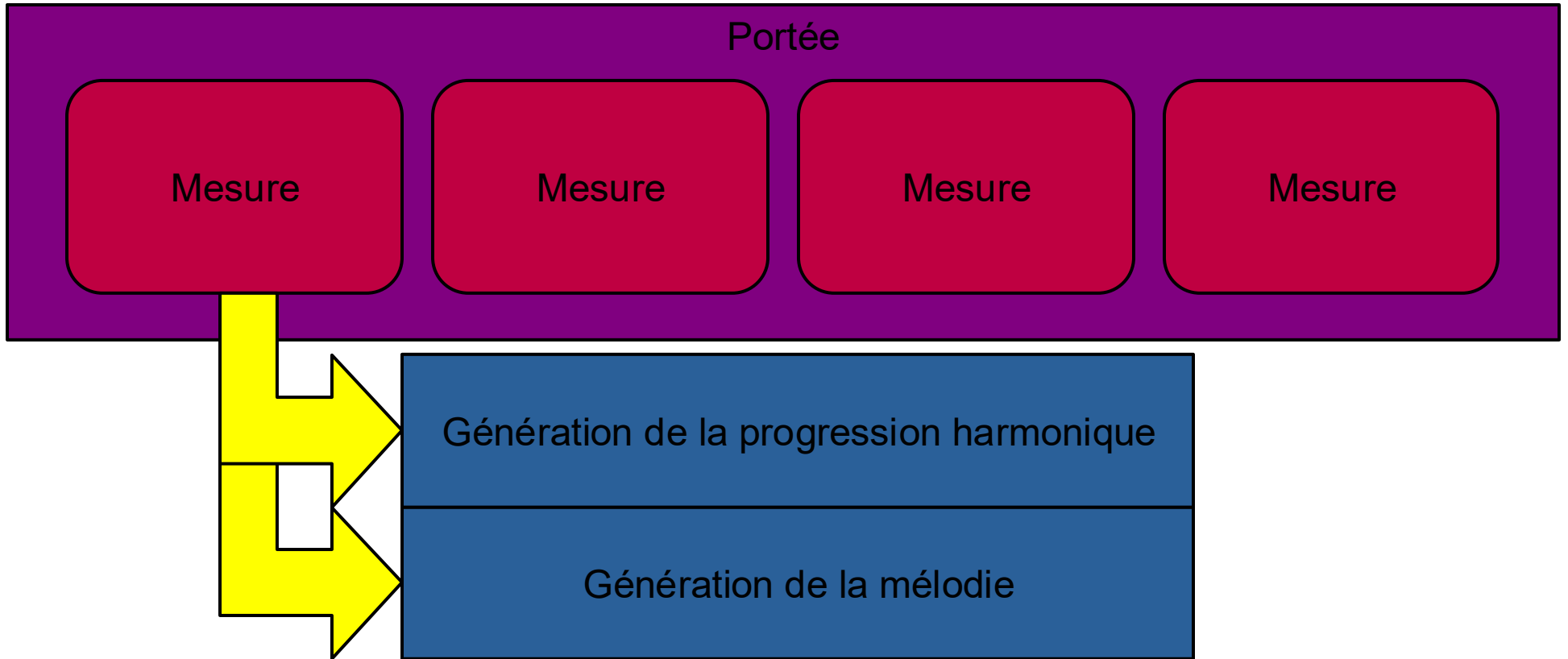
•Notes :

- Peut ne pas avoir de fréquence et être silencieuse
- Durée énumérée comme fraction de la mesure
- Vitesse de 0 à 127

•Canaux : complet si les durées se somment à 1

Mesures : 5 premiers canaux pour la mélodie

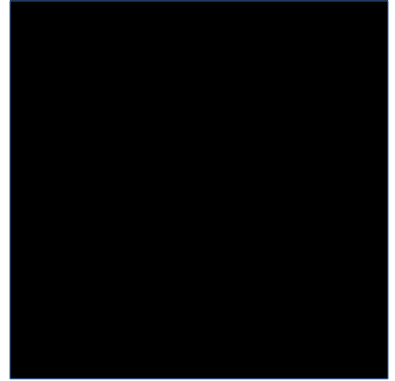
Processus de génération



Processus de génération

- .Génération sur un modèle de récurrence selon le dernier élément
 - Choix du premier élément sous loi uniforme
- .Choix des éléments suivants selon des tables de probabilités prédéfinies
- .Génération aléatoire : pas de procédural, pas d'IA

Résultats du générateur

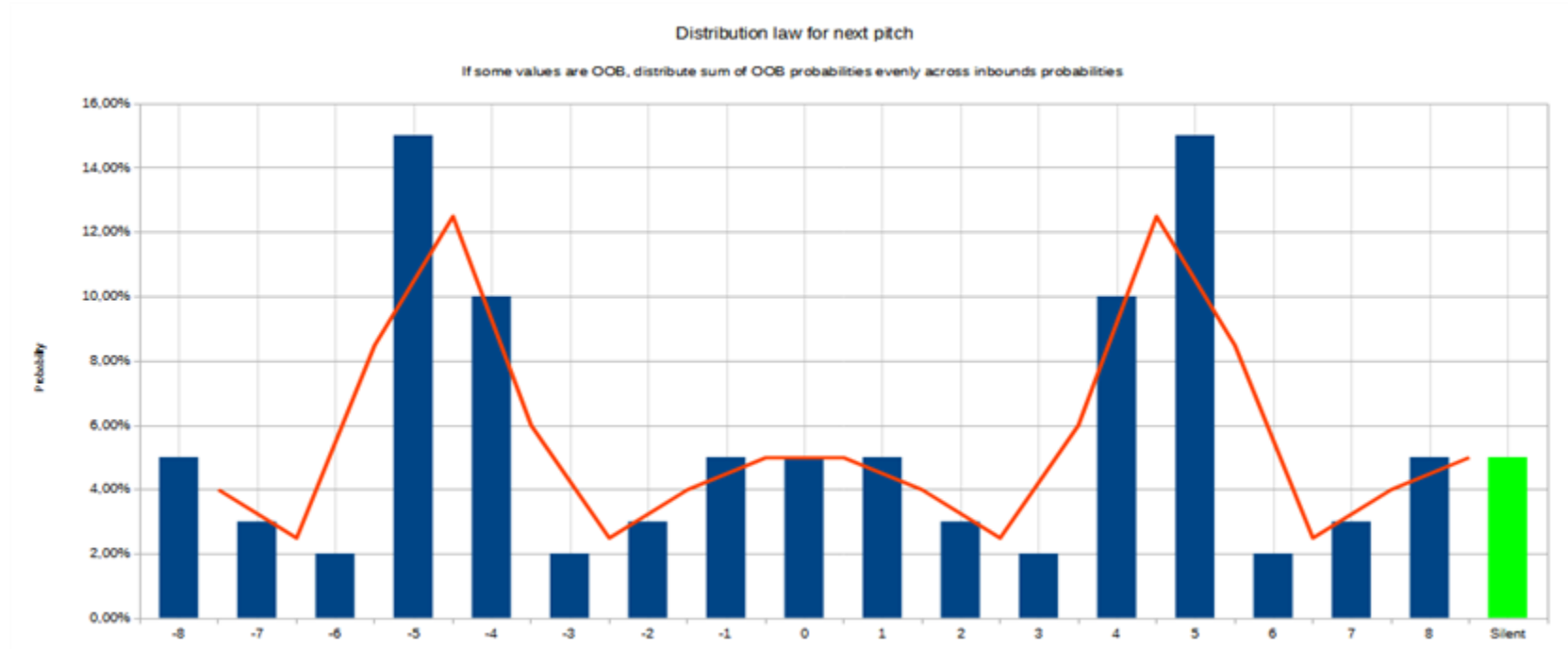


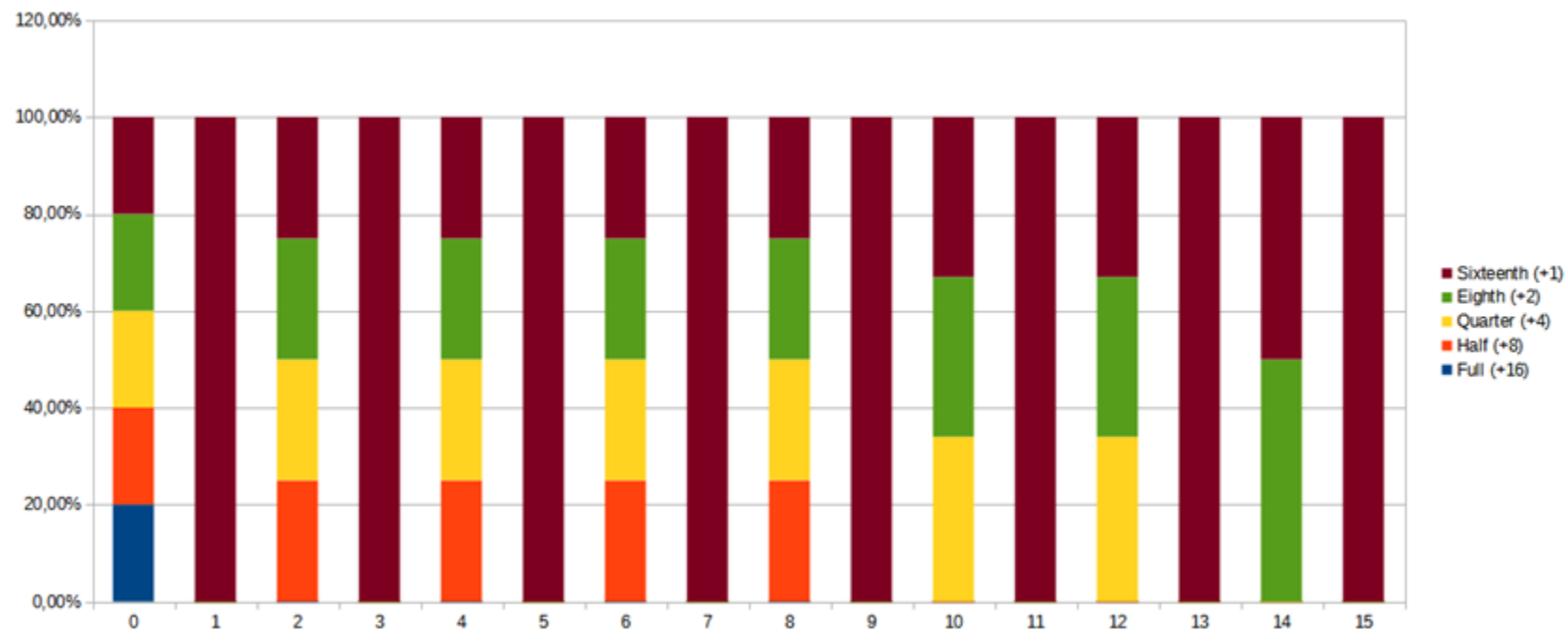
- C'est pas génial... Pourquoi ?
 - Principal coupable : le rythme
 - Mais les hauteurs y sont aussi pour quelque-chose

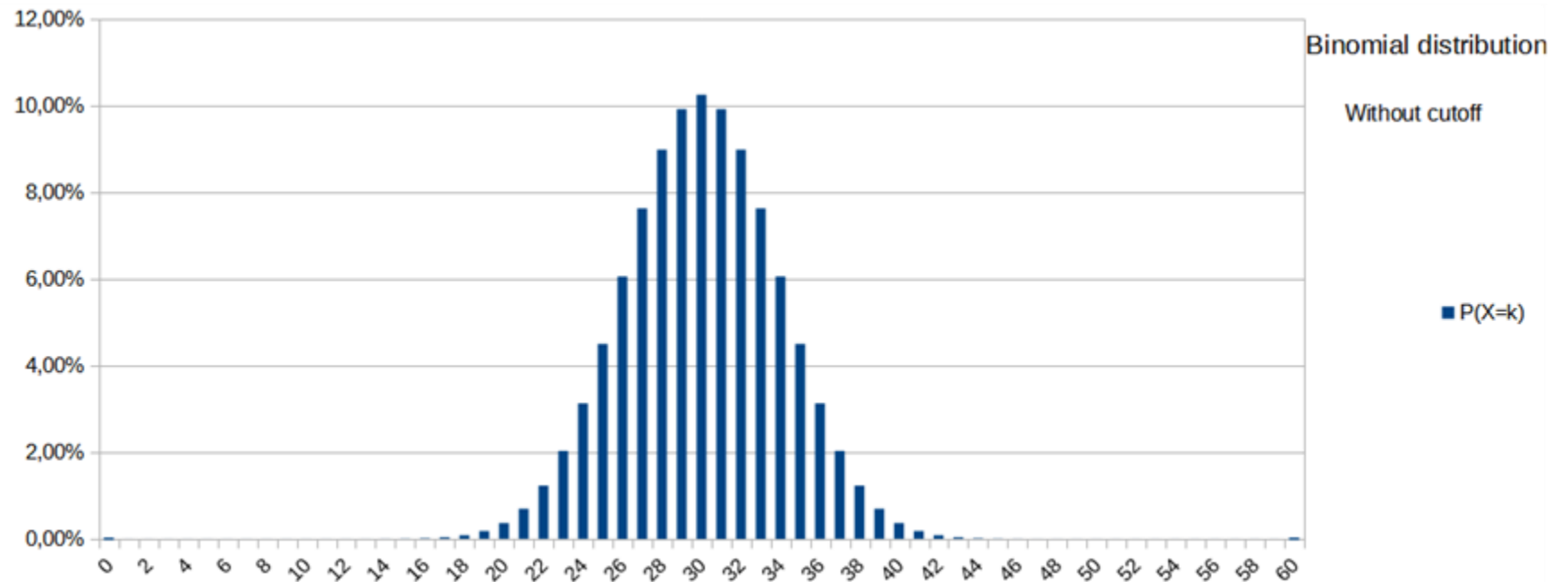
Révision du générateur

- .Aucune coordination entre harmoniques et mélodie
- .Génération mesure par mesure : chaque mesure est indépendante de l'autre
- .Contrairement aux exemples, aucune banque de données

Analyse des lois de probabilités

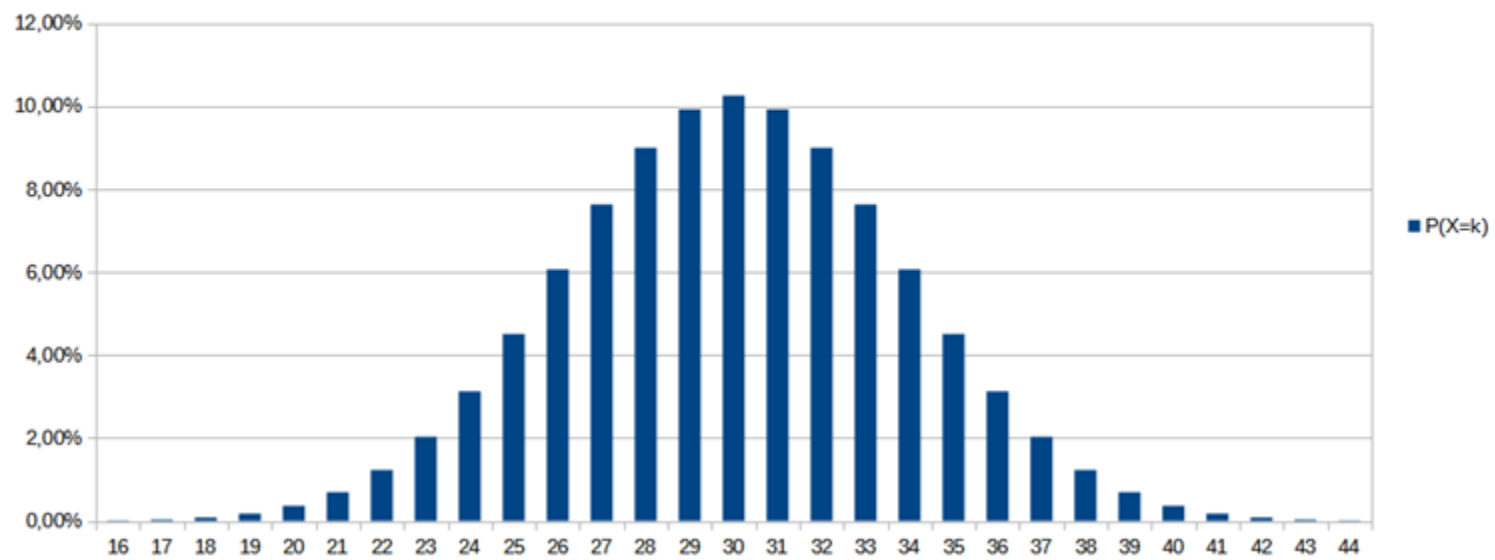


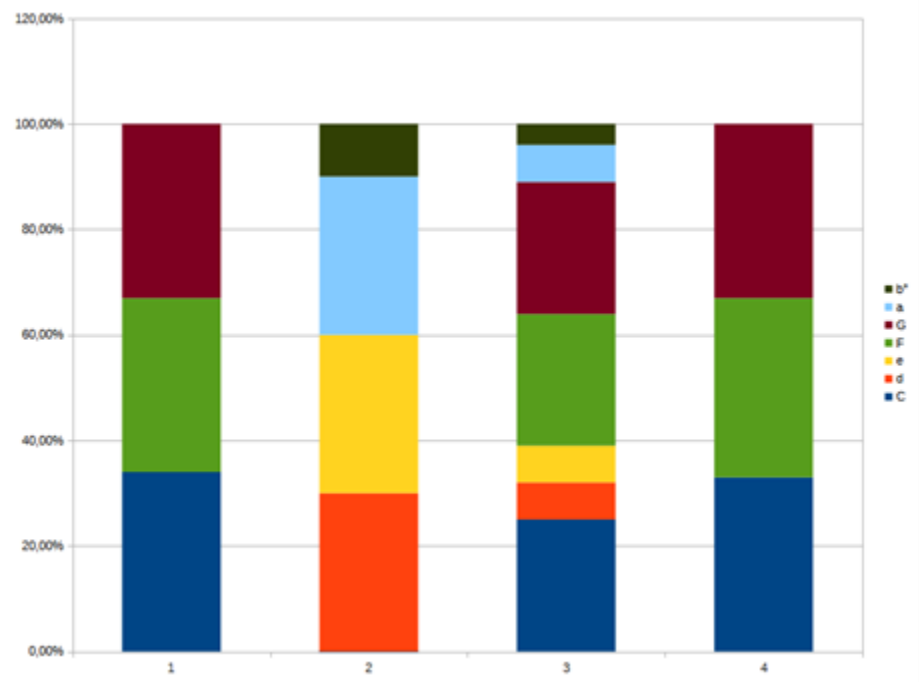




Binomial distribution

With cutoff





Synthèse

Matrix			Note vector	Affine	Next note	Variables	Addendums
1	0	0	Pitch	P	Pitch + P	P = next Pitch offset	P goes from -8 to 8 as long as the next pitch stays inbounds
0	0	0*	Rhythm	+ R	= R	R = next Rhythm	Next rhythm does not depend on the previous note, but rather the position in the channel
0	-10	1	Velocity	X	Velocity + X - 10*Rrhythm	X = velocity mod	X goes from 0 to 60 as long as the next velocity stays inbounds ~ Bin(60, 0.5)
							All variables in the affine vector are independent ; joint law easily computable
							2 main types of laws : inductive (Pitch and Velocity) and positional (Rhythm and Chord)
							For inductive laws, the first pick always follows uniform law