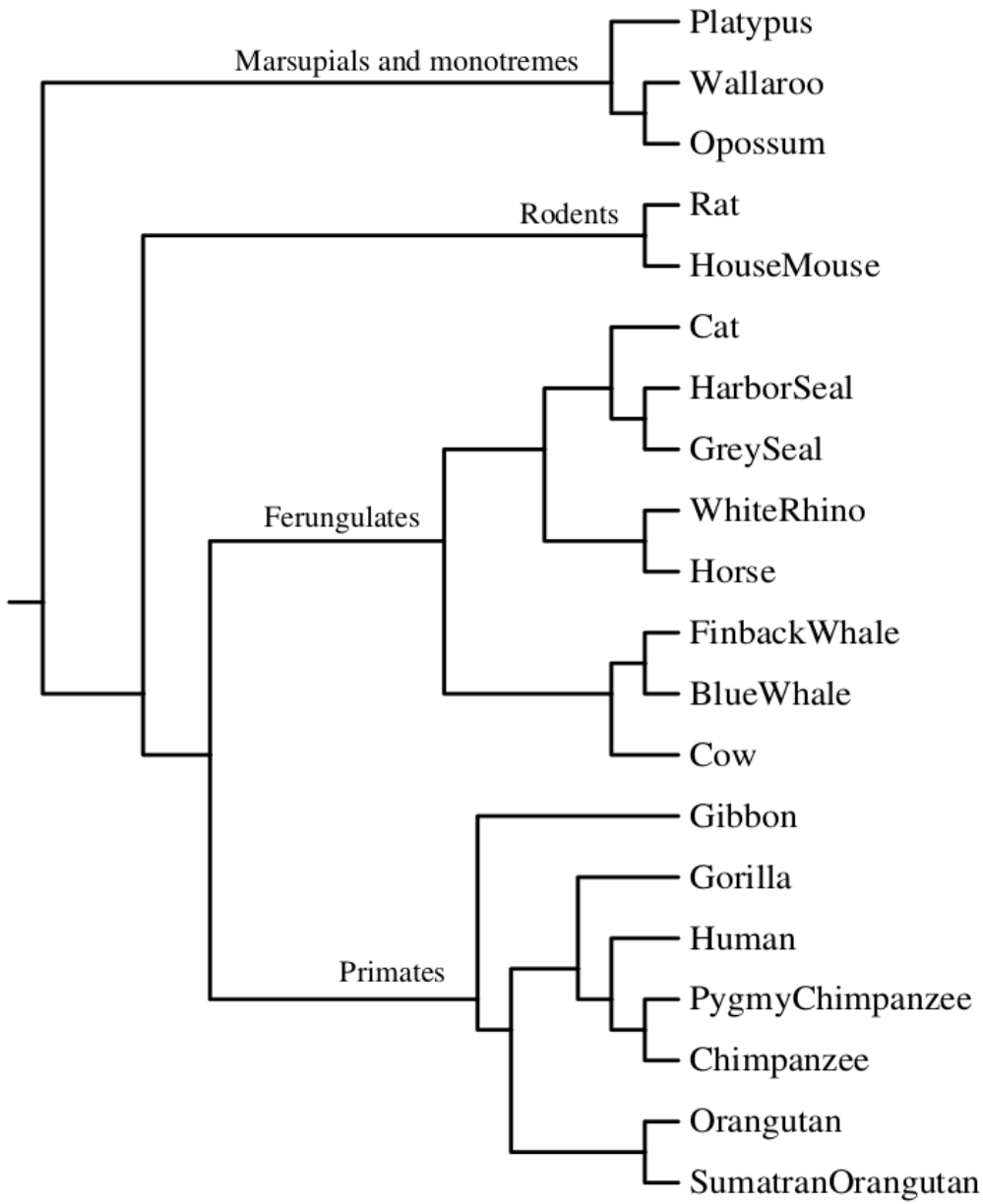
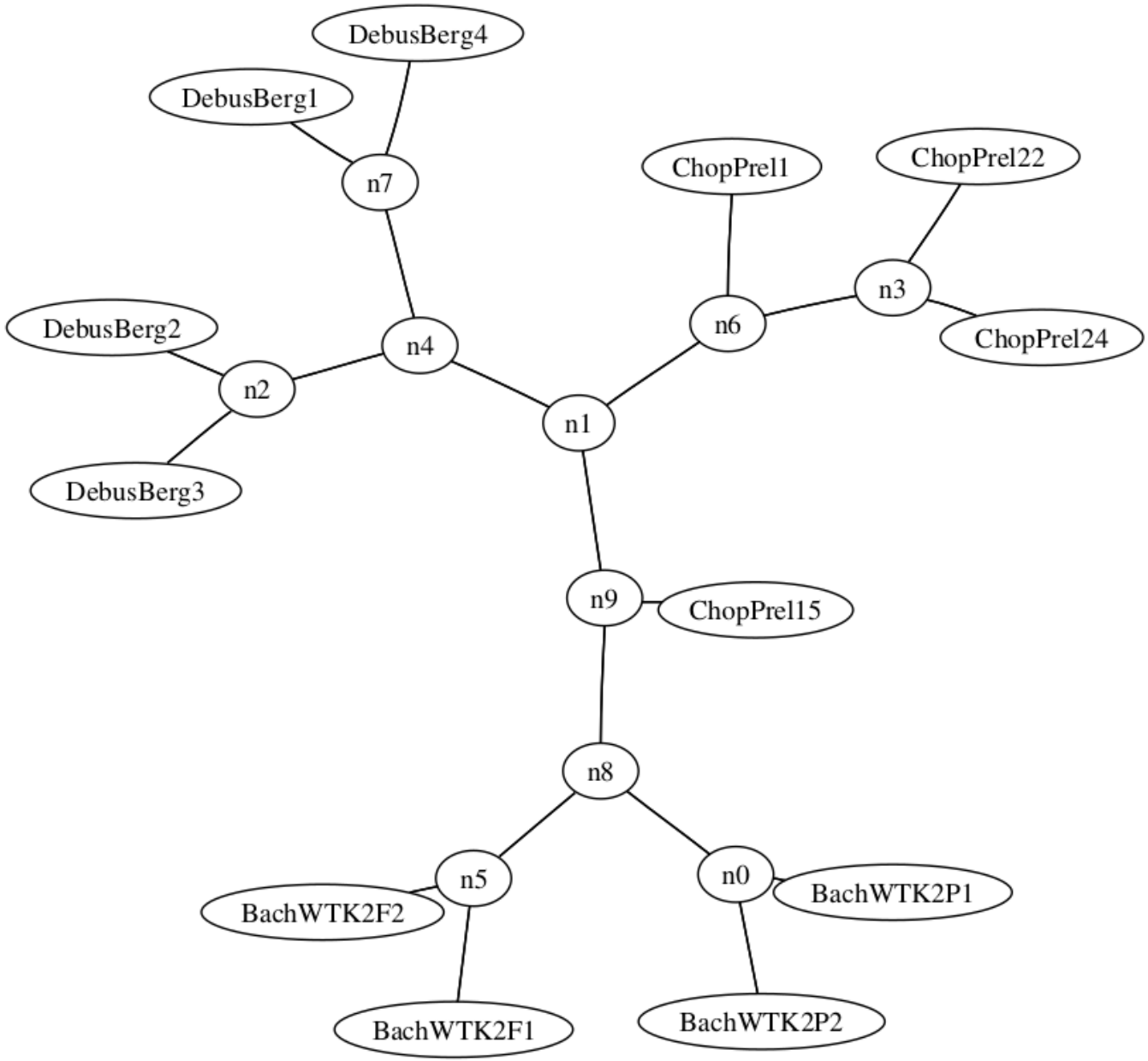
Distance d'Information Normalisée

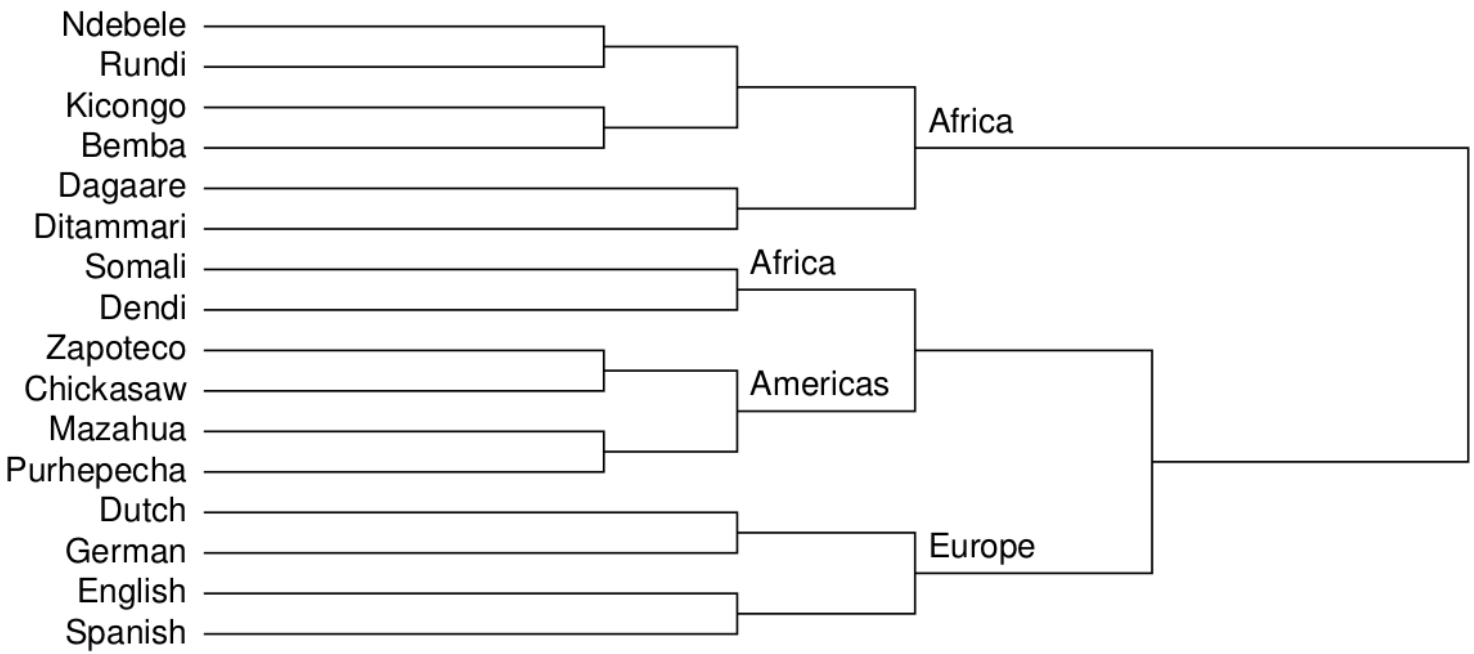
Ou: Comment avec un peu de théorie de la calculabilité on peut classifier des choses dont on ne sait rien

Meven Bertrand

7 Février 2018


Platypus
Marsupials and monotremes
Wallaroo
Opossum
Rodents
Rat
HouseMouse
Cat
HarborSeal
GreySeal
Ferungulates
WhiteRhino
Horse
FinbackWhale
BlueWhale
Cow
Gibbon
Gorilla
Human
Primates
PygmyChimpanzee
Chimpanzee
Orangutan
SumatranOrangutan


DebusBerg4
DebusBerg1
n7
ChopPrel1
ChopPrel22
n3
n6
DebusBerg2
n4
ChopPrel24
n2
n1
DebusBerg3
n9
ChopPrel15
n8
n5
n0
BachWTK2P1
BachWTK2F2
BachWTK2F1
BachWTK2P2


Ndebele
Rundi
Kicongo
Bemba
Dagaare
Ditammari
Somali
Dendi
Zapoteco
Chickasaw
Mazahua
Purhepecha
Dutch
German
English
Spanish
Africa
Africa
Americas
Europe

Plan

1. Un peu de complexité de Kolmogorov
2. Distance d'information
3. Distance d'information normalisée
4. Bonus : La distance du Web

Comment mesurer la complexité d'un mot ?

But du jeu : être capable de dire qu'un mot est « compliqué » ou « simple » :

111111111111111111111111111111111111111111111111111111111111

vs

100011111111011100010101100110100101110011000010111011011000 11

L'idée : ce qui est compliqué est ce qui est **dur à décrire**

Comment mesurer la complexité d'un mot ?

But du jeu : être capable de dire qu'un mot est « compliqué » ou « simple » :

111111111111111111111111111111111111111111111111111111111111

vs

100011111111011100010101100110100101110011000010111011011000 11

L'idée : ce qui est compliqué est ce qui est **dur à décrire**

Comment mesurer la complexité d'un mot ?

But du jeu : être capable de dire qu'un mot est « compliqué » ou « simple » :

111111111111111111111111111111111111111111111111111111111111

vs

100011111111011100010101100110100101110011000010111011011000 11

L'idée : ce qui est compliqué est ce qui est **dur à décrire**

La complexité de Kolmogorov

On va utiliser abondamment :

- un langage Σ (typiquement $\{0, 1\}$)
- des mots sur ce langage : des éléments de $\Sigma^* = \bigcup_{n\in\mathbb{N}} \Sigma^n$
- la longueur len d'un mot

Définition

La complexité $C_{lang}(x)$ du mot x dans le langage $lang$ est la longueur du plus petit programme écrit en $lang$ qui produit le mot x.

On définit aussi $C_{lang}(x|y)$ complexité de x sachant y, où on donne au programme y

La complexité de Kolmogorov

On va utiliser abondamment :

- un langage Σ (typiquement $\{0, 1\}$)
- des mots sur ce langage : des éléments de $\Sigma^* = \bigcup_{n\in\mathbb{N}} \Sigma^n$
- la longueur len d'un mot

Définition

La complexité $C_{lang}(x)$ du mot x dans le langage $lang$ est la longueur du plus petit programme écrit en $lang$ qui produit le mot x.

On définit aussi $C_{lang}(x|y)$ complexité de x sachant y, où on donne au programme y

La complexité de Kolmogorov

On va utiliser abondamment :

- un langage Σ (typiquement $\{0, 1\}$)
- des mots sur ce langage : des éléments de $\Sigma^* = \bigcup_{n\in\mathbb{N}} \Sigma^n$
- la longueur len d'un mot

Définition

La complexité $C_{lang}(x)$ du mot x dans le langage $lang$ est la longueur du plus petit programme écrit en $lang$ qui produit le mot x.

On définit aussi $C_{lang}(x|y)$ complexité de x sachant y, où on donne au programme y

Universalité

Théorème d'universalité

Il existe un langage $univ$ « universel » : si $lang$ est un autre langage, il existe $c_{lang} \in \mathbb{N}$ tel que

$$C_{univ}(w) \leq C_{lang}(w) + c_{lang}$$

On pose $K(x) = C_{univ}(x)$, c'est la **complexité de Kolmogorov** de x

De même, langage universel pour la complexité conditionnelle $\rightarrow K(x|y)$ complexité de x sachant y

Universalité

Théorème d'universalité

Il existe un langage $univ$ « universel » : si $lang$ est un autre langage, il existe $c_{lang} \in \mathbb{N}$ tel que

$$C_{univ}(w) \leq C_{lang}(w) + c_{lang}$$

On pose $K(x) = C_{univ}(x)$, c'est la **complexité de Kolmogorov** de x

De même, langage universel pour la complexité conditionnelle $\rightarrow K(x|y)$ complexité de x sachant y

Universalité

Théorème d'universalité

Il existe un langage $univ$ « universel » : si $lang$ est un autre langage, il existe $c_{lang} \in \mathbb{N}$ tel que

$$C_{univ}(w) \leq C_{lang}(w) + c_{lang}$$

On pose $K(x) = C_{univ}(x)$, c'est la **complexité de Kolmogorov** de x

De même, langage universel pour la complexité conditionnelle $\rightarrow K(x|y)$ complexité de x sachant y

Aparté : Mais comment diable construit-on ce truc ?

Définition

Si φ est une fonction récursive, on prend $C_\varphi(x) = \min_{\{p \in \Sigma^* | \varphi(p)=x\}} \text{len}(p)$.
Idem pour $C_\varphi(x|y) = \min_{\{p \in \Sigma^* | \varphi(p,y)=x\}} \text{len}(p)$

Pour trouver ψ universelle, on prend

- $(\varphi_n)_{n\in\mathbb{N}}$ énumération (calculable) des fonctions récursives
- ψ telle que $\varphi(n, y, p) = \psi_n(y, p)$
- alors pour $n \in \mathbb{N}$, $C_\psi(x) \leq C_{\varphi_n}(x|y) + c_n$ où c_n ne dépend que de n

Ce n'est pas tout à fait fini, avec ça on n'est pas sûr que $C_\psi(xy) \leq C_\psi(x) + C_\psi(y)$
$\rightarrow$ programmes autodélimitants (de la forme $1^{\text{len}(p)}0p$)

Aparté : Mais comment diable construit-on ce truc ?

Définition

Si φ est une fonction récursive, on prend $C_\varphi(x) = \min_{\{p \in \Sigma^* | \varphi(p)=x\}} \text{len}(p)$.
Idem pour $C_\varphi(x|y) = \min_{\{p \in \Sigma^* | \varphi(p,y)=x\}} \text{len}(p)$

Pour trouver ψ universelle, on prend

- $(\varphi_n)_{n\in\mathbb{N}}$ énumération (calculable) des fonctions récursives
- ψ telle que $\varphi(n, y, p) = \psi_n(y, p)$
- alors pour $n \in \mathbb{N}$, $C_\psi(x) \leq C_{\varphi_n}(x|y) + c_n$ où c_n ne dépend que de n

Ce n'est pas tout à fait fini, avec ça on n'est pas sûr que $C_\psi(xy) \leq C_\psi(x) + C_\psi(y)$
$\rightarrow$ programmes autodélimitants (de la forme $1^{\text{len}(p)}0p$)

Aparté : Mais comment diable construit-on ce truc ?

Définition

Si φ est une fonction récursive, on prend $C_\varphi(x) = \min_{\{p \in \Sigma^* | \varphi(p)=x\}} \text{len}(p)$.
Idem pour $C_\varphi(x|y) = \min_{\{p \in \Sigma^* | \varphi(p,y)=x\}} \text{len}(p)$

Pour trouver ψ universelle, on prend

- $(\varphi_n)_{n \in \mathbb{N}}$ énumération (calculable) des fonctions récursives
- ψ telle que $\varphi(n, y, p) = \psi_n(y, p)$
- alors pour $n \in \mathbb{N}$, $C_\psi(x) \leq C_{\varphi_n}(x|y) + c_n$ où c_n ne dépend que de n

Ce n'est pas tout à fait fini, avec ça on n'est pas sûr que $C_\psi(xy) \leq C_\psi(x) + C_\psi(y)$
$\rightarrow$ programmes autodélimitants (de la forme $1^{\text{len}(p)}0p$)

Aparté : Mais comment diable construit-on ce truc ?

Définition

Si φ est une fonction récursive, on prend $C_\varphi(x) = \min_{\{p \in \Sigma^* | \varphi(p)=x\}} \text{len}(p)$.
Idem pour $C_\varphi(x|y) = \min_{\{p \in \Sigma^* | \varphi(p,y)=x\}} \text{len}(p)$

Pour trouver ψ universelle, on prend

- $(\varphi_n)_{n\in\mathbb{N}}$ énumération (calculable) des fonctions récursives
- ψ telle que $\varphi(n, y, p) = \psi_n(y, p)$
- alors pour $n \in \mathbb{N}$, $C_\psi(x) \leq C_{\varphi_n}(x|y) + c_n$ où c_n ne dépend que de n

Ce n'est pas tout à fait fini, avec ça on n'est pas sûr que $C_\psi(xy) \leq C_\psi(x) + C_\psi(y)$
$\rightarrow$ programmes autodélimitants (de la forme $1^{\text{len}(p)}0p$)

Quelques inégalités bien utiles

Disclaimer

- les $\log$ sont en base 2
- les (in)égalités sont rarement vraiment vraies, mais elles sont moralement vraies

- $K(x) \leq \text{len}(x)$ (au pire, on réécrit tout x)
- $K(x,y) \leq K(x) + K(y|x) + O(1)$ (au pire, on construit tout x, puis tout y)
- $K(x,y) = K(x) + K(y|x) = K(y) + K(x|y)$ (à un terme $O(\log(K(xy)))$ près)

Quelques inégalités bien utiles

Disclaimer

- les $\log$ sont en base 2
- les (in)égalités sont rarement vraiment vraies, mais elles sont moralement vraies

- $K(x) \leq \text{len}(x)$ (au pire, on réécrit tout x)
- $K(x,y) \leq K(x) + K(y|x) + O(1)$ (au pire, on construit tout x, puis tout y)
- $K(x,y) = K(x) + K(y|x) = K(y) + K(x|y)$ (à un terme $O(\log(K(xy)))$ près)

Plan

1 Un peu de complexité de Kolmogorov

2 Distance d'information

3 Distance d'information normalisée

4 Bonus : La distance du Web

Comment mesurer la différence entre deux mots ?

But du jeu : dire que des mots sont « proches » ou « lointains » :

1000111111110111000101011001101001011100110000101110110110011

vs

0111000000001000111010100110010110100011001111010001001011100

ou vs

1000111111110111000101011001101001011100110000101110110110011**1**11

ou vs

0101010000101000011101000101110101000101011101011100000000010

Quelques idées qui ne marchent pas

$K(x|y)$

Pas symétrique...
$K(x|\varepsilon) = K(x)$ mais $K(\varepsilon|x) = O(1)$

$K(x|y) + K(y|x)$

C'est trop gros : il y a de la redondance entre $x \to y$ et $y \to x$

Solution

$E(x, y) = \max(K(x|y), K(y|x))$

Quelques idées qui ne marchent pas

$K(x|y)$

Pas symétrique...
$K(x|\varepsilon) = K(x)$ mais $K(\varepsilon|x) = O(1)$

$K(x|y) + K(y|x)$

C'est trop gros : il y a de la redondance entre $x \to y$ et $y \to x$

Solution

$E(x, y) = \max(K(x|y), K(y|x))$

Quelques idées qui ne marchent pas

$K(x|y)$

Pas symétrique...
$K(x|\varepsilon) = K(x)$ mais $K(\varepsilon|x) = O(1)$

$K(x|y) + K(y|x)$

C'est trop gros : il y a de la redondance entre $x \to y$ et $y \to x$

Solution

$E(x, y) = \max(K(x|y), K(y|x))$

Aparté : Mais pourquoi diable ce truc est-il bien ?

Formellement, si φ est calculable, on définit

$$E_\varphi(x,y) = \min_{\{p\in\mathbb{N}|\varphi(p,x)=y\wedge\varphi(p,y)=x\}}(\text{len}(p))$$

comme pour C, il y a ψ universelle, i.e. telle que

$$E_\psi(x,y) = E_\varphi(x,y) + O(1)$$

Et on montre

Relation pas facile

$$E_\psi(x,y) = \max(K(x|y), K(y|x)) + O(\log\max(K(x|y), K(y|x)))$$

donc E n'est pas loin de la distance universelle E_ψ.

Universalité de E

Distance d'information

Une distance d'information admissible est une fonction (totale, pas forcément symétrique) $D : \Sigma^* \times \Sigma^* \to \mathbb{R}_+$ qui :

- vérifie $D(x, y) = 0$ ssi $x = y$
- vérifie $\sum_{x \neq y} 2^{-D(x,y)} \leq 1$ et $\sum_{y \neq x} 2^{-D(x,y)} \leq 1$
- est approximable par le haut

Universalité de E

Alors E est une distance d'information admissible, vérifie l'inégalité triangulaire, et est universelle :

$$E(x, y) \leq D(x, y) + O(1)$$

Universalité de E

Distance d'information

Une distance d'information admissible est une fonction (totale, pas forcément symétrique) $D : \Sigma^* \times \Sigma^* \to \mathbb{R}_+$ qui :

- vérifie $D(x, y) = 0$ ssi $x = y$
- vérifie $\sum_{x \neq y} 2^{-D(x,y)} \leq 1$ et $\sum_{y \neq x} 2^{-D(x,y)} \leq 1$
- est approximable par le haut

Universalité de E

Alors E est une distance d'information admissible, vérifie l'inégalité triangulaire, et est universelle :

$$E(x, y) \leq D(x, y) + O(1)$$

Plan

1 Un peu de complexité de Kolmogorov

2 Distance d'information

3 Distance d'information normalisée

4 Bonus : La distance du Web

Comment normaliser ?

Problème : $E(x,y) = \max(K(x|y), K(y|x))$ dépend de la taille... Ne mesure pas la similarité $\rightarrow$ il faut normaliser.

Par quoi diviser ?

- la longueur : tue l'inégalité triangulaire
- $K(x,y)$: si $K(x) \approx K(y) \approx K(x|y) \approx K(y|x)$, $\frac{\max(K(x|y),K(y|x))}{K(x,y)} \approx \frac{1}{2}$ et pas 1
- $\max(K(x), K(y))$ marche !

$$e(x,y) = \frac{\max(K(x|y), K(y|x))}{\max(K(x), K(y)}$$

est dans $[0;1]$, est une distance et est universelle

Comment normaliser ?

Problème : $E(x,y) = \max(K(x|y), K(y|x))$ dépend de la taille... Ne mesure pas la similarité $\rightarrow$ il faut normaliser.

Par quoi diviser ?

- la longueur : tue l'inégalité triangulaire
- $K(x,y)$: si $K(x) \approx K(y) \approx K(x|y) \approx K(y|x)$, $\frac{\max(K(x|y),K(y|x))}{K(x,y)} \approx \frac{1}{2}$ et pas 1
- $\max(K(x), K(y))$ marche !

$$e(x,y) = \frac{\max(K(x|y), K(y|x))}{\max(K(x), K(y)}$$

est dans $[0;1]$, est une distance et est universelle

Comment calculer ?

Compression : rendre un fichier le plus petit possible

$\rightarrow K$, mais en pratique ! Approximation de K par $Z =$ taille de x une fois compressé

Et $K(x|y)$?

Rappel : $K(xy) = K(x) + K(y|x)$, donc

$$e(x,y) = \frac{K(xy) - \min(K(x), K(y))}{\max(K(x), K(y))} \approx \frac{Z(xy) - \min(Z(x), Z(y))}{\max(Z(x), Z(y))}$$

On a gagné : c'est calculable !

Comment calculer ?

Compression : rendre un fichier le plus petit possible
$\rightarrow K$, mais en pratique ! Approximation de K par Z = taille de x une fois compressé

Et $K(x|y)$?
Rappel : $K(xy) = K(x) + K(y|x)$, donc

$$e(x,y) = \frac{K(xy) - \min(K(x), K(y))}{\max(K(x), K(y))} \approx \frac{Z(xy) - \min(Z(x), Z(y))}{\max(Z(x), Z(y))}$$

On a gagné : c'est calculable !

Comment calculer ?

Compression : rendre un fichier le plus petit possible
$\rightarrow K$, mais en pratique ! Approximation de K par Z = taille de x une fois compressé

Et $K(x|y)$?
Rappel : $K(xy) = K(x) + K(y|x)$, donc

$$e(x,y) = \frac{K(xy) - \min(K(x), K(y))}{\max(K(x), K(y))} \approx \frac{Z(xy) - \min(Z(x), Z(y))}{\max(Z(x), Z(y))}$$

On a gagné : c'est calculable !

Comment calculer ?

Compression : rendre un fichier le plus petit possible
$\rightarrow K$, mais en pratique ! Approximation de K par $Z =$ taille de x une fois compressé

Et $K(x|y)$?
Rappel : $K(xy) = K(x) + K(y|x)$, donc

$$e(x,y) = \frac{K(xy) - \min(K(x), K(y))}{\max(K(x), K(y))} \approx \frac{Z(xy) - \min(Z(x), Z(y))}{\max(Z(x), Z(y))}$$

On a gagné : c'est calculable !

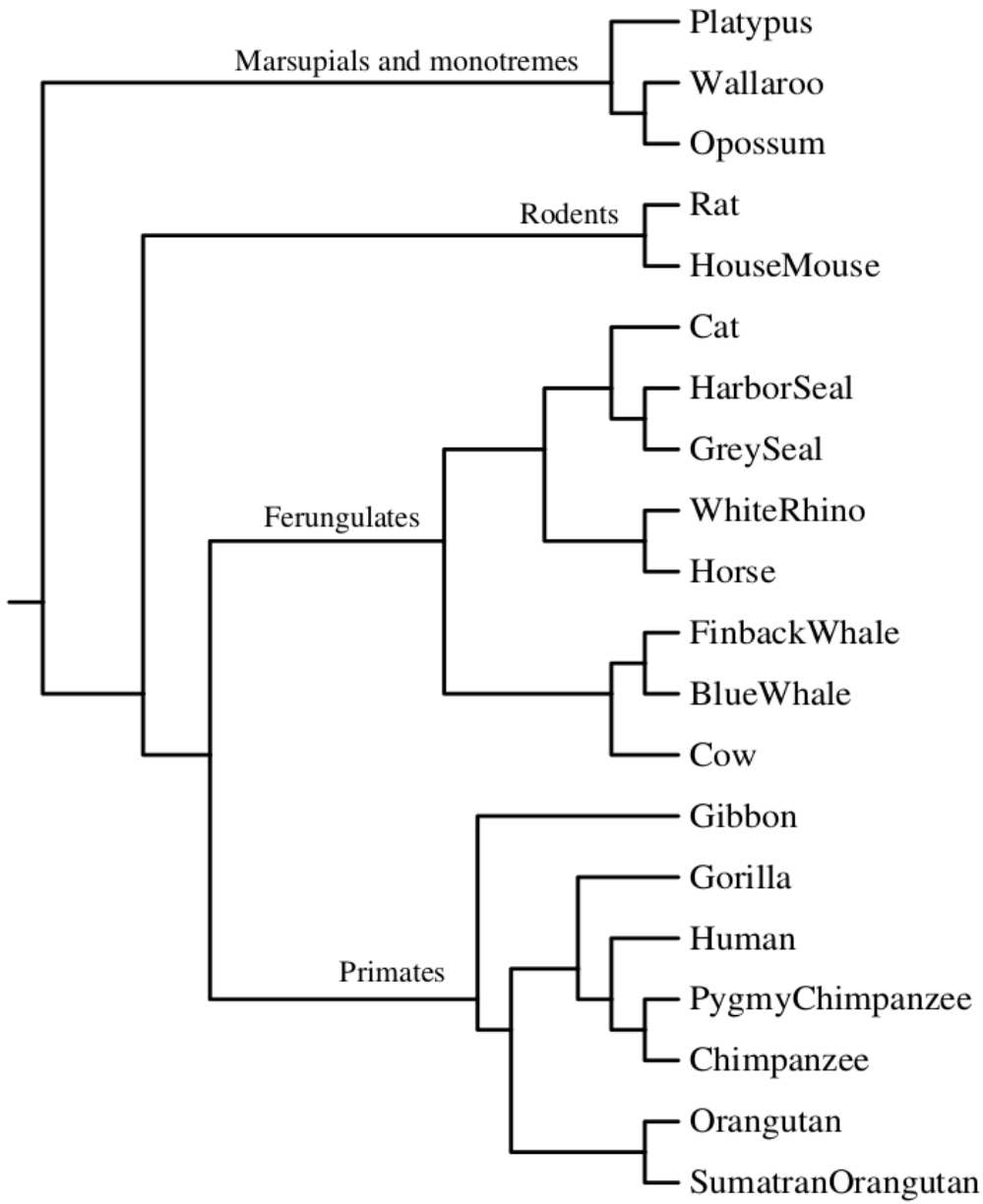
Comment calculer ?

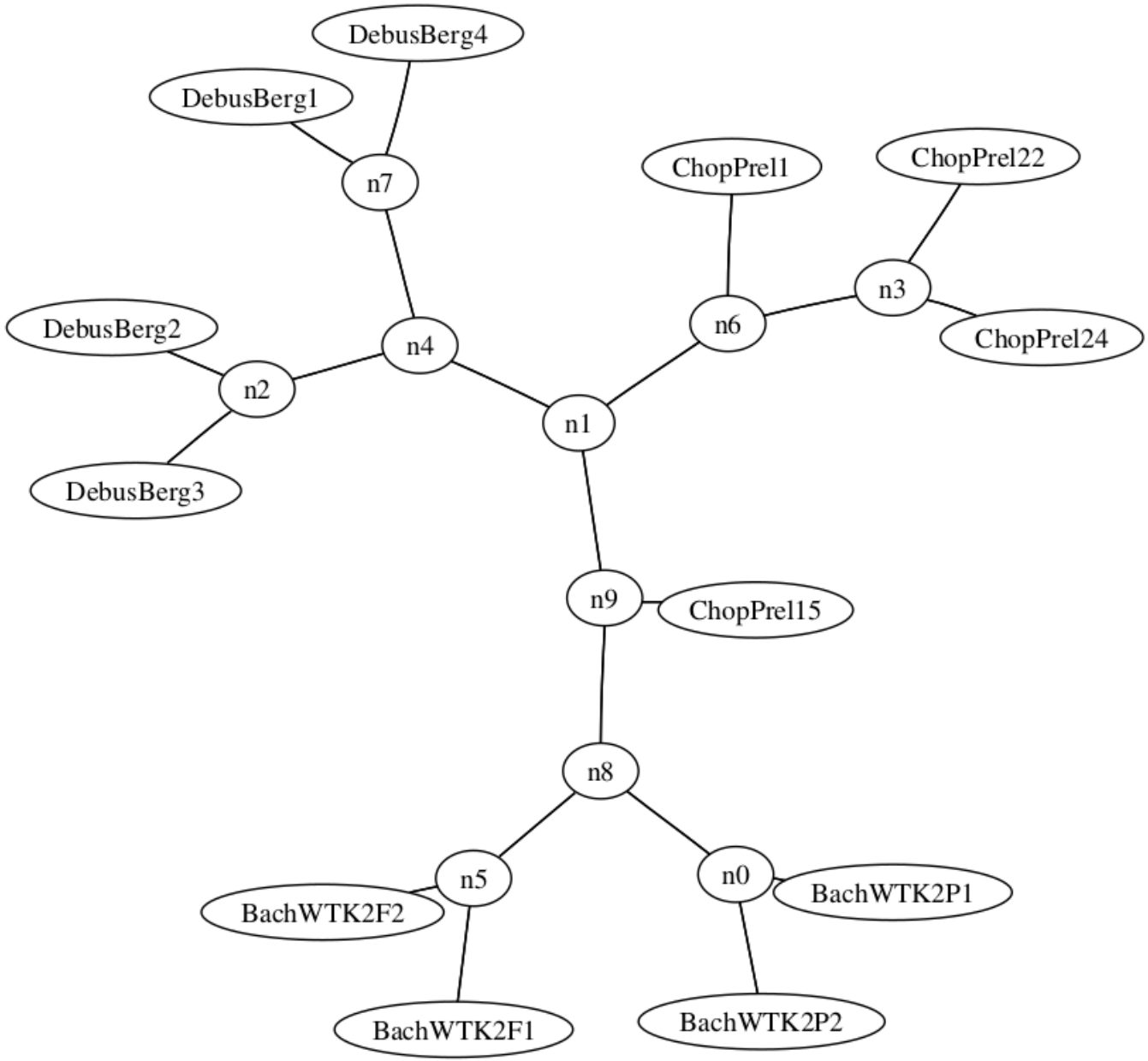
Compression : rendre un fichier le plus petit possible
$\rightarrow K$, mais en pratique ! Approximation de K par $Z =$ taille de x une fois compressé

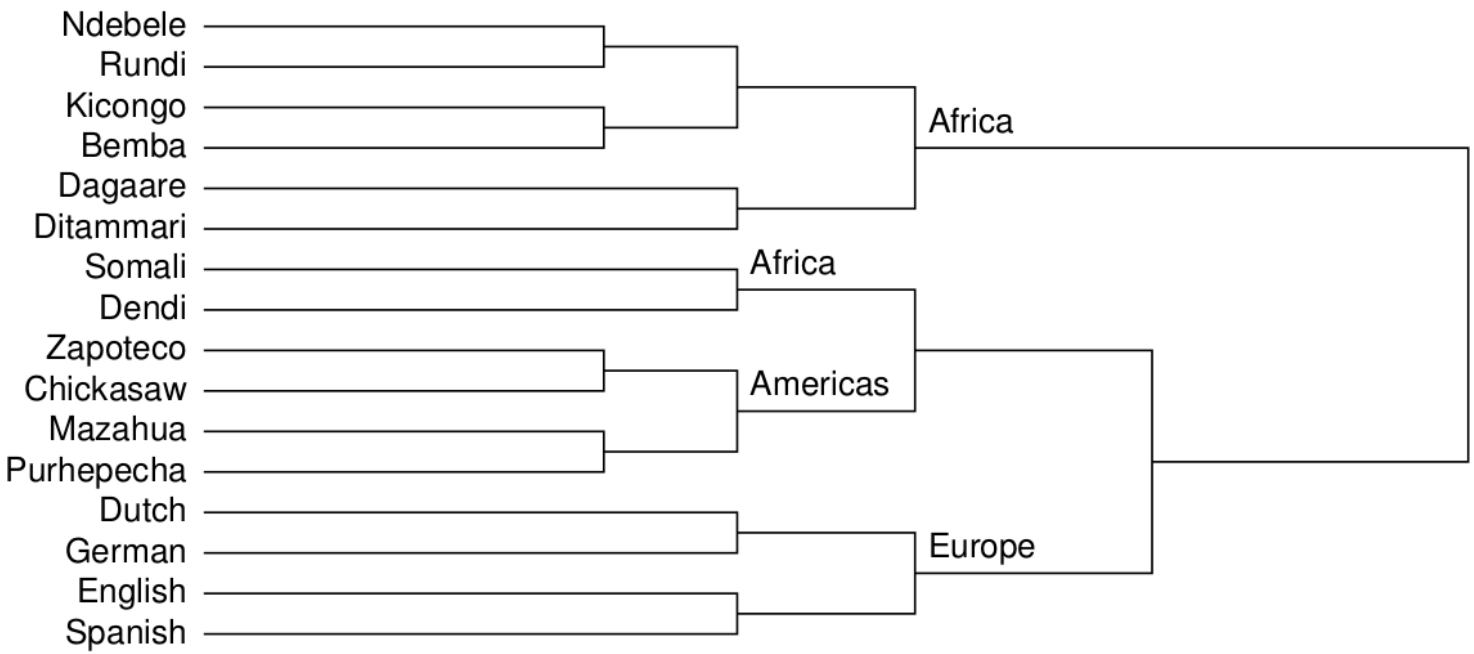
Et $K(x|y)$?
Rappel : $K(xy) = K(x) + K(y|x)$, donc

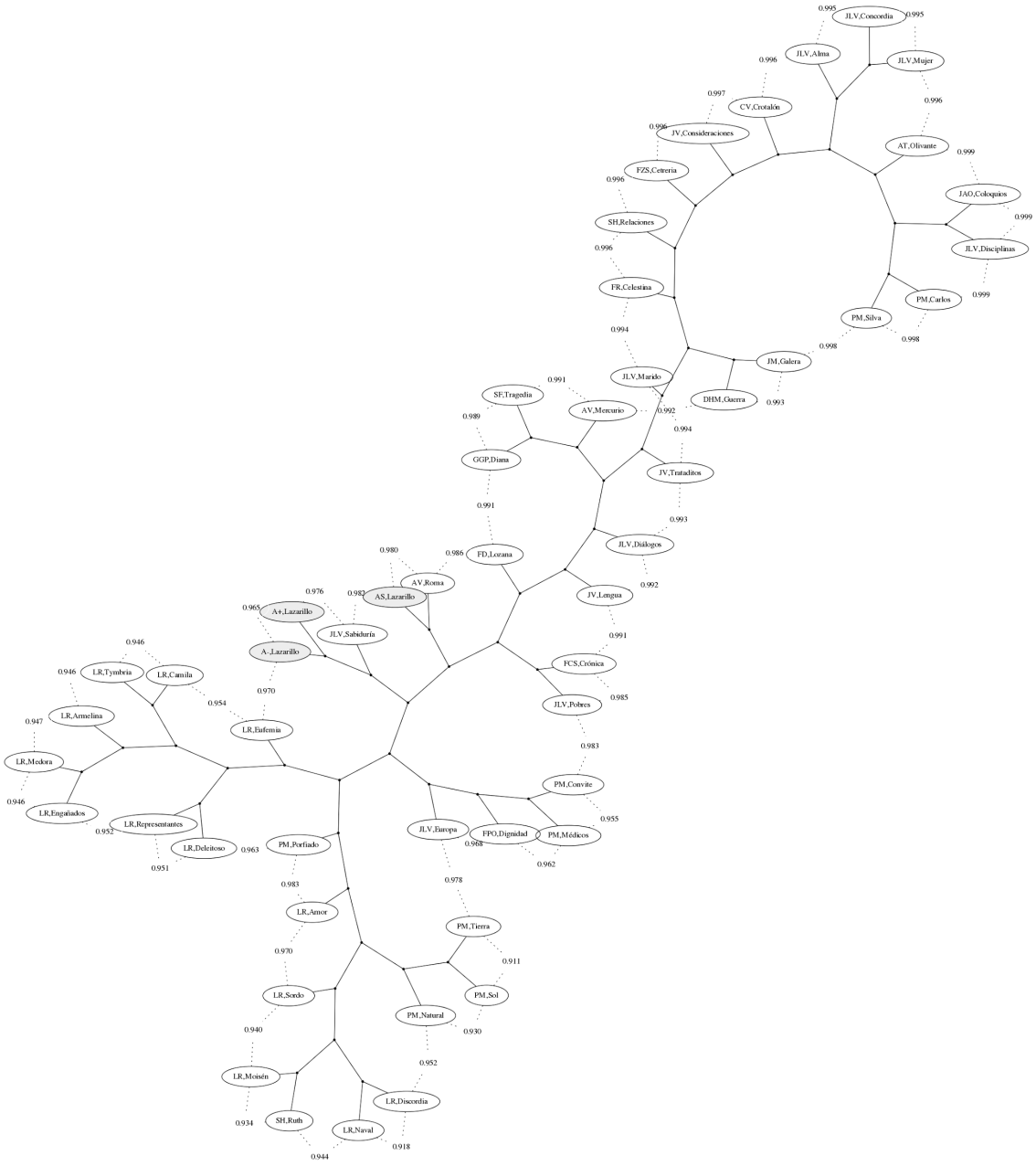
$$e(x,y) = \frac{K(xy) - \min(K(x), K(y))}{\max(K(x), K(y))} \approx \frac{Z(xy) - \min(Z(x), Z(y))}{\max(Z(x), Z(y))}$$

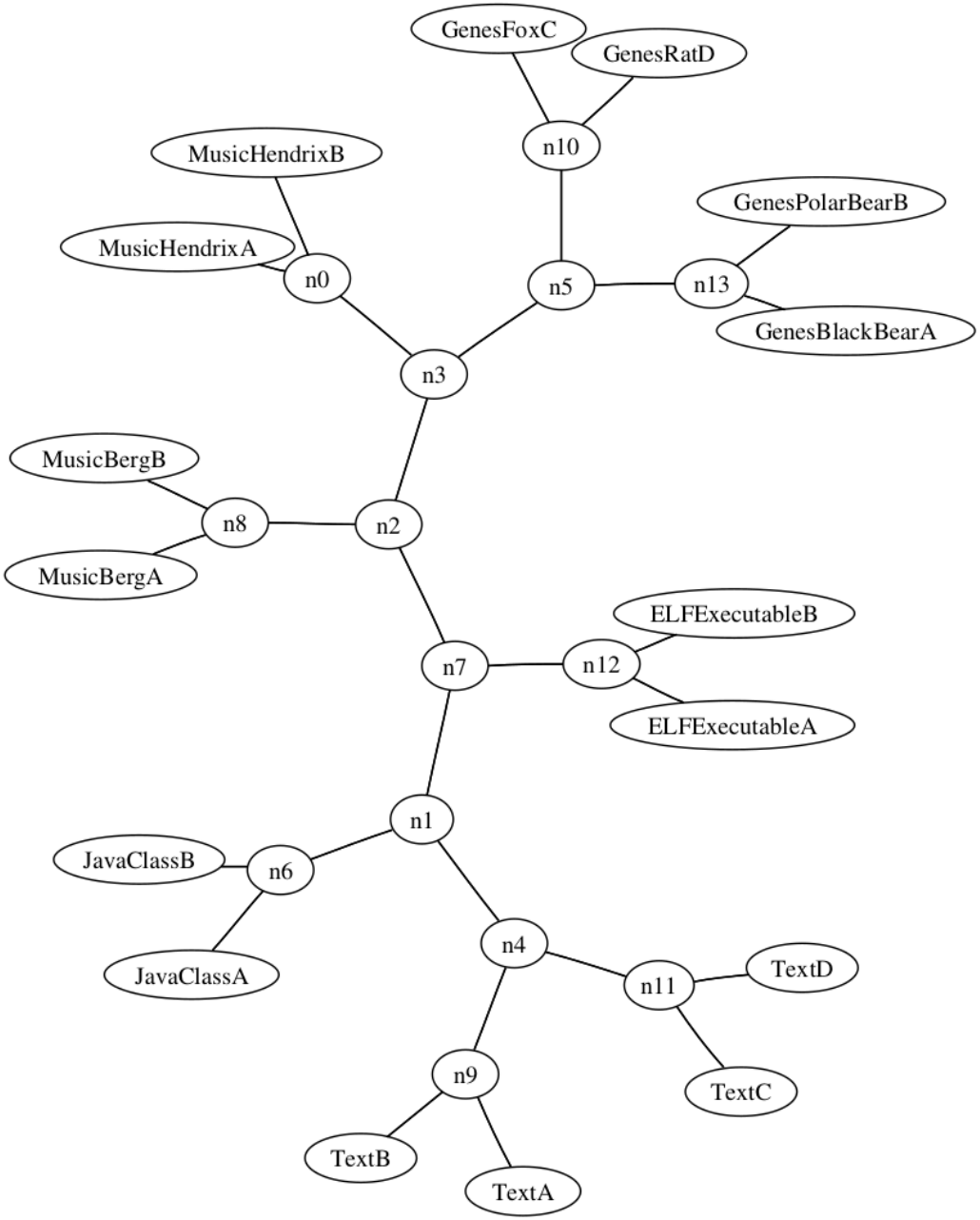
On a gagné : c'est calculable !


Marsupials and monotremes
Platypus
Wallaroo
Opossum
Rodents
Rat
HouseMouse
Ferungulates
Cat
HarborSeal
GreySeal
WhiteRhino
Horse
FinbackWhale
BlueWhale
Cow
Primates
Gibbon
Gorilla
Human
PygmyChimpanzee
Chimpanzee
Orangutan
SumatranOrangutan


DebusBerg4
DebusBerg1
n7
ChopPrel1
ChopPrel22
n3
n6
DebusBerg2
n4
ChopPrel24
n2
n1
DebusBerg3
n9
ChopPrel15
n8
n5
n0
BachWTK2P1
BachWTK2F2
BachWTK2F1
BachWTK2P2


Ndebele
Rundi
Kicongo
Bemba
Dagaare
Ditammari
Somali
Dendi
Zapoteco
Chickasaw
Mazahua
Purhepecha
Dutch
German
English
Spanish
Africa
Africa
Americas
Europe


JLV,Concordia
JLV,Alma
JLV,Mujer
CV,Crotalón
JV,Consideraciones
AT,Olivante
FZS,Cetrería
JAO,Coloquios
SH,Relaciones
JLV,Disciplinas
FR,Celestina
PM,Carlos
PM,Silva
JM,Galera
JLV,Marido
SF,Tragedia
DHM,Guerra
AV,Mercurio
GGP,Diana
JV,Trataditos
JLV,Diálogos
FD,Lozana
AV,Roma
AS,Lazarillo
JV,Lengua
A+,Lazarillo
JLV,Sabiduría
A-,Lazarillo
FCS,Crónica
LR,Tymbria
LR,Camila
JLV,Pobres
LR,Armelina
LR,Eufemia
PM,Convite
LR,Medora
LR,Engañados
LR,Representantes
LR,Deleitoso
JLV,Europa
FPO,Dignidad
PM,Médicos
PM,Porfiado
LR,Amor
PM,Tierra
LR,Sordo
PM,Sol
PM,Natural
LR,Moisén
LR,Discordia
SH,Ruth
LR,Naval


GenesFoxC
GenesRatD
n10
MusicHendrixB
GenesPolarBearB
MusicHendrixA
n0
n5
n13
GenesBlackBearA
n3
MusicBergB
n8
n2
MusicBergA
ELFExecutableB
n7
n12
ELFExecutableA
n1
JavaClassB
n6
n4
JavaClassA
n11
TextD
n9
TextC
TextB
TextA

Plan

1 Un peu de complexité de Kolmogorov

2 Distance d'information

3 Distance d'information normalisée

4 Bonus : La distance du Web

Inégalité de Kraft et interprétation probabiliste

Ensemble préfixe $P \Rightarrow$ inégalité de Kraft : $\sum_{p \in P} 2^{-\operatorname{len}(p)} \leq 1$

En particulier, $\sum_{x \in \Sigma^*} 2^{-K(x|y)} \leq 1$ donc $m(x|y) = 2^{-K(x|y)}$ ressemble à une **probabilité** !

Universalité

Si $\mu : \Sigma^* \times \Sigma^* \to \mathbb{R}_+$ est telle que :

- $\mu(\cdot, y)$ est une mesure sur Σ^* de masse ≤ 1
- μ est approximable par le bas

alors $\mu(x|y) = O(m(x|y))$

Inégalité de Kraft et interprétation probabiliste

Ensemble préfixe $P \Rightarrow$ inégalité de Kraft : $\sum_{p \in P} 2^{-\operatorname{len}(p)} \leq 1$

En particulier, $\sum_{x \in \Sigma^*} 2^{-K(x|y)} \leq 1$ donc $m(x|y) = 2^{-K(x|y)}$ ressemble à une **probabilité** !

Universalité

Si $\mu : \Sigma^* \times \Sigma^* \to \mathbb{R}_+$ est telle que :

- $\mu(\cdot, y)$ est une mesure sur Σ^* de masse ≤ 1
- μ est approximable par le bas

alors $\mu(x|y) = O(m(x|y))$

Inégalité de Kraft et interprétation probabiliste

Ensemble préfixe $P \Rightarrow$ inégalité de Kraft : $\sum_{p \in P} 2^{-\operatorname{len}(p)} \leq 1$

En particulier, $\sum_{x \in \Sigma^*} 2^{-K(x|y)} \leq 1$ donc $m(x|y) = 2^{-K(x|y)}$ ressemble à une **probabilité** !

Universalité

Si $\mu : \Sigma^* \times \Sigma^* \to \mathbb{R}_+$ est telle que :

- $\mu(\cdot, y)$ est une mesure sur Σ^* de masse ≤ 1
- μ est approximable par le bas

alors $\mu(x|y) = O(m(x|y))$

La distance du Web

En approximant $m(x)$ par $\frac{N(x)}{N_0}$ (N : nombre de résultats dans une recherche, N_0 : nombre de résultats totaux) :

$$e(x,y) \approx \frac{\max(\log N(x), \log N(y)) - \log N(x,y)}{\log N_0 - \min(\log N(x), \log N(y))}$$

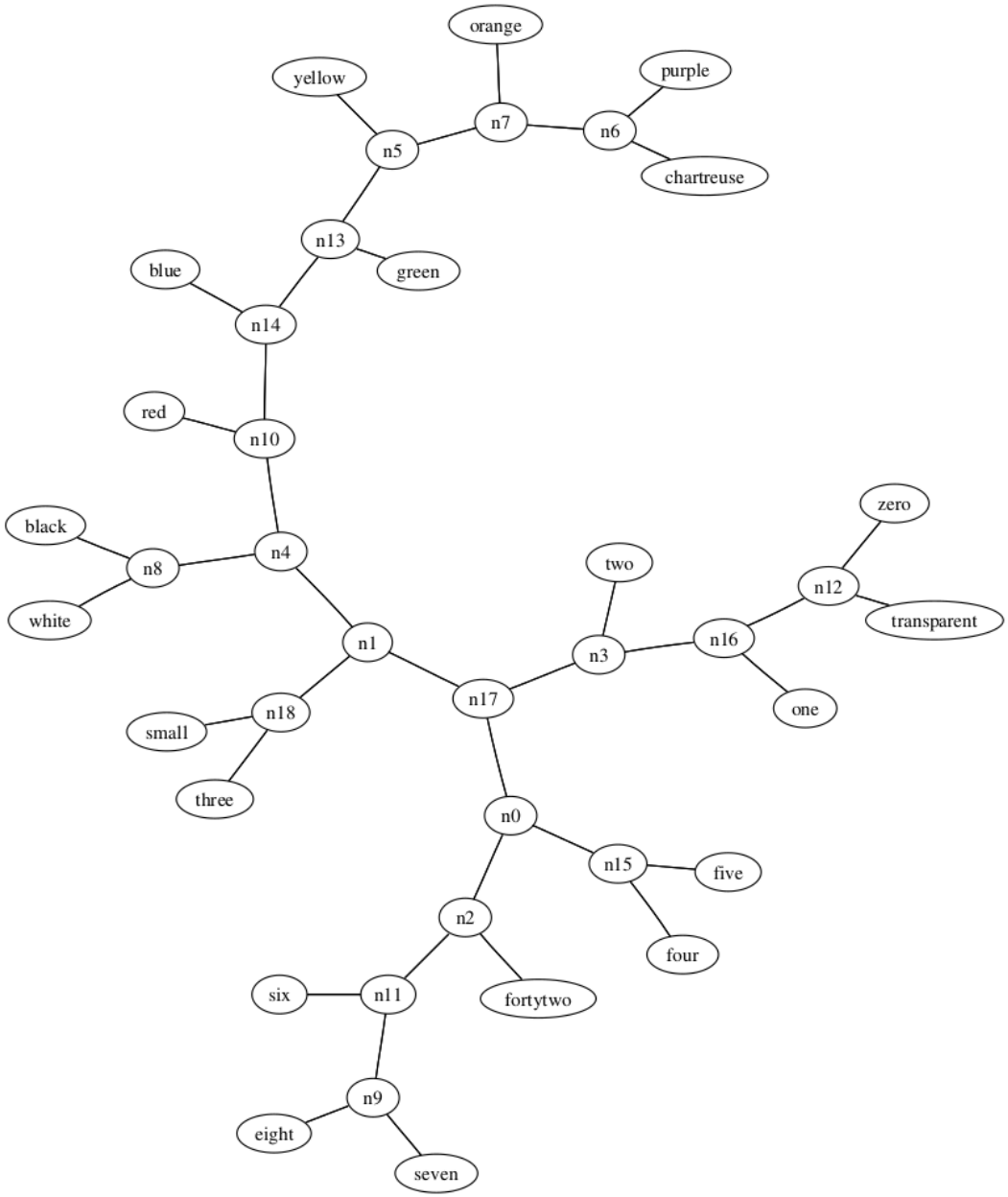
Ce n'est plus du tout une distance ! Mais c'est quand même bien pratique...

La distance du Web

En approximant $m(x)$ par $\frac{N(x)}{N_0}$ (N : nombre de résultats dans une recherche, N_0 : nombre de résultats totaux) :

$$e(x,y) \approx \frac{\max(\log N(x), \log N(y)) - \log N(x,y)}{\log N_0 - \min(\log N(x), \log N(y))}$$

Ce n'est plus du tout une distance ! Mais c'est quand même bien pratique...


orange
yellow
purple
n7
n6
n5
chartreuse
n13
blue
green
n14
red
n10
zero
black
n4
two
n12
n8
transparent
white
n1
n3
n16
n18
n17
one
small
three
n0
n15
five
n2
four
six
n11
fortytwo
n9
eight
seven

Training Data

Positive examples (21 cases)

11	13	17	19	2
23	29	3	31	37
41	43	47	5	53
59	61	67	7	71
73				

Negative examples (22 cases)

10	12	14	15	16
18	20	21	22	24
25	26	27	28	30
32	33	34	4	6
8	9			

Anchors (5 dimensions)
composite, number, orange, prime, record

Testing Results

	Positive tests	**Negative tests**
Positive Predictions	101, 103, 107, 109, 79, 83, 89, 91, 97	110
Negative Predictions		36, 38, 40, 42, 44, 45, 46, 48, 49

Accuracy: 18/19 = 94.74%