L’empreinte environnementale des calculs de la bioinformatique

1/ Des outils disponibles pour estimer l’empreinte unitaire d’un calcul bioinformatique

2/ L’effet rebond

3/ Comment calculer l’empreinte de la bioinformatique

1/ Empreinte environnementale des outils bioinformatiques

The Carbon Footprint of Bioinformatics

Jason Grealey,[*,†,1,2] Loïc Lannelongue,[†,3,4,5] Woei-Yuh Saw,[1] Jonathan Marten,[‡,4] Guillaume Méric,[1,6] Sergio Ruiz-Carmona,[1] and Michael Inouye[*,1,3,4,5,7,8]

Mol. Biol. Evol. 39(3):msac034 doi:10.1093/molbev/msac034 Advance Access publication February 10, 2022

Embracing Green Computing in Molecular Phylogenetics

Sudhir Kumar[*,1,2]

Mol. Biol. Evol. 39(3): msac043 doi:10.1093/molbev/msac043

Many biological disciplines apply computational approaches to investigate evolutionary questions involving the origins of genes, evolutionary relationships of organisms, positive and negative selection, the evolution of biodiversity, and genotype–phenotype connections across the tree of life. The importance of these questions is reflected by the escalating use of software for molecular evolutionary analyses (fig. 1).

(Stamatakis 2014; Sharma and Kumar 2021). In the future, smarter software will avoid overcomputing, decreasing the carbon footprints of big data analyses.

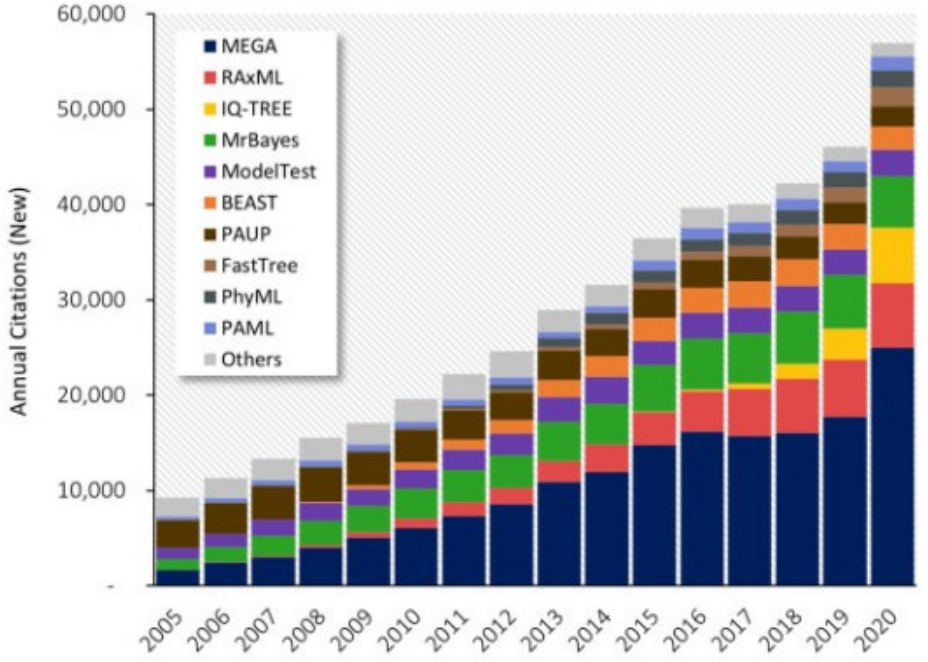


Fig. 1. The use of computational methods in molecular evolution has been increasing quickly, as seen in the annual counts of new research articles citing the use of major software packages for molecular evolutionary and phylogenetic analyses. Citation counts for software packages were obtained from Google Scholar (last accessed January 25, 2022) for 2005–2020. See supplementary material, Supplementary Material online for more details on software versions included.

Table 1. Carbon Footprint of a Range of Bioinformatic Tasks.

Task	Tool	Version	Details about the Experiments	Carbon Footprint		Tree-months	km in a Car (EU)	Running Time and Memory	Approximate Scaling (if known)
				Increase (%)	$kgCO_2e$				
Genome scaffolding	SSPACE	2.0	Scaffolding 2.4 million long reads from human chromosome 14 (Hunt et al. 2014).	—	0.0010	0.0011	0.01	3 min 21 s 30 GB	Linearly with number of reads.
	SOAPdenovo2	r223		+45%	0.0015	0.0016	0.01	4 min 52 s 30 GB	
	SGA	0.9.43		+2,752%	0.029	0.032	0.17	1 h 35 min 30 GB	
Genome scaffolding	SSPACE	2.0	Scaffolding 23 million short reads from human chromosome 14 (Hunt et al. 2014).	—	0.0027	0.0029	0.02	8 min 40 s 30 GB	
	SOAPdenovo2	r223		+34%	0.0036	0.0039	0.02	1 min 38 s 30 GB	
	SGA	0.9.43		+4,801%	0.13	0.14	0.74	7 h 05 min 30 GB	
Genome assembly	Abyss	2.0	De novo assembly of a human genome from Illumina sequencing reads (Jackman et al. 2017).	—	11	12	61	20 h 34 GB	
	MEGAHIT	1.0.6		+42%	15	16	86	26 h 197 GB	
Metagenome assembly	MetaVelvet k101	1.2.01	Metagenome assembly from 100 soil samples (Vollmers et al. 2017).	—	14	16	82	1 h 06 min 130 GB	
	MEGAHIT	1.0.3		+438%	77	84	439	15 h 36 min 12 GB	
	metaSPAdes	3.8.0		+1,206%	186	203	1,065	29 h 24 min 60 GB	
Metagenome classification (short read)	Kraken2	2.0.7	Metagenomic classification of 5 Gb of randomly sampled reads from Zymo mock community (batch ZRC190633), containing yeast, Gram-negative, and positive bacteria (Dilthey et al. 2019)	—	0.0052	0.0057	0.03	20 min 21 GB	Linearly with number of reads.
	Centrifuge	1.0.4		+141%	0.013	0.014	0.07	58 min 12 GB	
	Kraken/Bracken	0.10.5/1.0.0		+1,650%	0.092	0.10	0.52	1 h 40 min 154 GB	
Metagenome classification (long read)	MetaMaps	—		—	18.25	19.91	104.27	209 h 53 min 262 GB	
Phylogenetics	BEAST/BEAGLE	1.8.4/2.1.2	Codon substitution modeling of extant carnivores and a pangolin group. Nucleotide substitution and phylogeographic modeling of Ebola virus genomes. See supplementary table 2, Supplementary Material online, for detailed results (Baele et al. 2019).	—	0.012–0.30	0.013–0.33	0.069–1.72	3 min 30 s to 7 h 45 min 2–8 GB	Power law with number of loci.

(continued)

Green Algorithms

How green are your computations?

Details about your algorithm

To understand how each parameter impacts your carbon footprint, check out the formula below and the methods article

Runtime (HH:MM): 12 : 0

Type of cores: CPU

Number of cores: 12

Model: Xeon E5-2683 v4

Memory available (in GB): 64

Select the platform used for the computations

Local server

253.64 g CO2e
Carbon footprint

2.28 kWh
Energy needed

0.28 tree-months
Carbon sequestration

1.45 km
in a passenger car

0.51%
of a flight Paris-London

Share your results with this link!

Select the platform used for the computations

Local server

Select location

Europe

Austria

Do you know the real usage factor of your CPU?

○ Yes ◉ No

Do you know the Power Usage Efficiency (PUE) of your local data centre?

○ Yes ◉ No

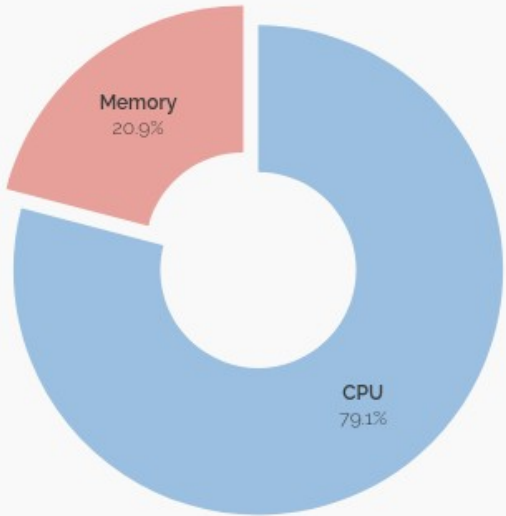
Do you want to use a Pragmatic Scaling Factor?

○ Yes ◉ No

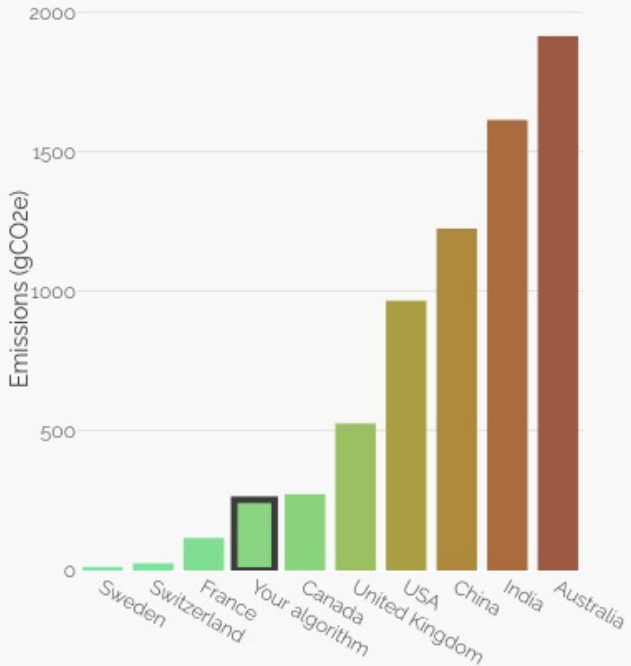
Reset *Change app version*

Share your results with this link!

Computing cores VS Memory

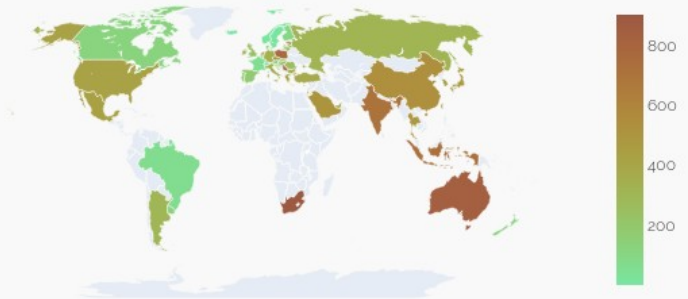


How the location impacts your footprint



More details about the methodology in the methods paper

Carbon Intensity across the world



About CO2e

"Carbon dioxide equivalent" (CO2e) measures the global warming potential of a mixture of greenhouse gases. **It represents the quantity of CO2 that would have the same impact on global warming** as the mix of interest and is used as a standardised unit to assess the environmental impact of human activities.

What is a tree-month?

It's the amount of CO2 sequestered by a tree in a month. **We use it to measure how long it would take to a mature tree to absorb the CO2 emitted by an algorithm.** We use the value of 11 kg CO2/year, which is roughly 1kg CO2/month.

What can you do about it?

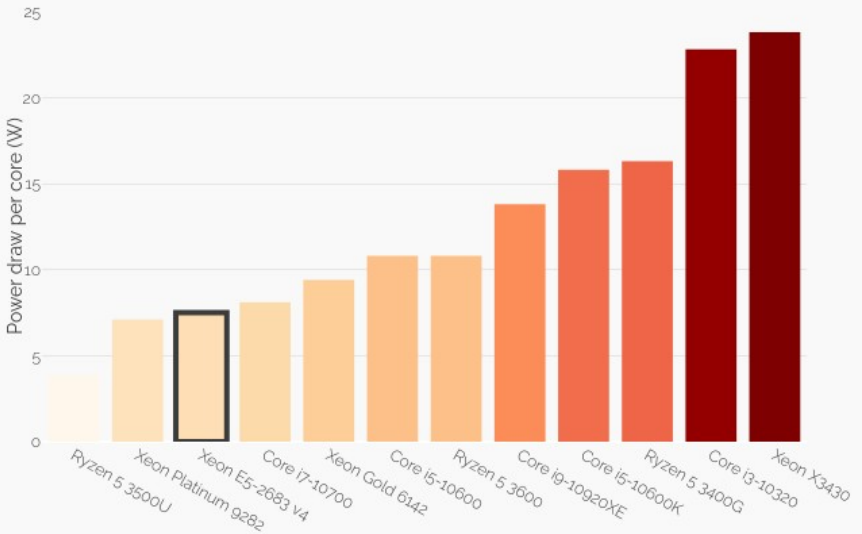
The main factor impacting your footprint is the location of your servers: the same algorithm will emit **74 times more** CO2e if ran in Australia compared to Switzerland. Although it's not always the case, many cloud providers offer the option to select a data centre.

Memory power draw is a huge source of waste, because **the energy consumption depends on the memory available, not the actual usage**, only requesting the needed memory is a painless way to reduce greenhouse gas emissions.

Generally, taking the time to write optimised code that runs faster with fewer resources saves both money and the planet.

And above all, **only run jobs that you need!**

Power draw of different processors



The formula

The carbon footprint is calculated by estimating the energy draw of the algorithm and the carbon intensity of producing this energy at a given location:

```
carbon footprint = energy needed * carbon intensity
```

Where the energy needed is:

```
runtime * (power draw for cores * usage + power draw for memory) * PUE * PSF
```

The power draw for the computing cores depends on the model and number of cores, while the memory power draw only depends on the size of memory available. The usage factor corrects for the real core usage (default is 1, i.e. full usage). The PUE (Power Usage Effectiveness) measures how much extra energy is needed to operate the data centre (cooling, lighting etc.). The PSF (Pragmatic Scaling Factor) is used to take into account multiple identical runs (e.g. for testing or optimisation).

The Carbon Intensity depends on the location and the technologies used to produce electricity. But note that **the "energy needed" indicated at the top of this page is independent of the location.**

How to report it?

It's important to track the impact of computational research on climate change in order to stimulate greener algorithms. For that, **we believe that the carbon footprint of a project should be reported on publications alongside other performance metrics**.

Here is a text you can include in your paper:

> *This algorithm runs in 12h on 12 CPUs Xeon E5-2683 v4, and draws 2.28 kWh. Based in Austria, this has a carbon footprint of 253.64 g CO2e, which is equivalent to 0.28 tree-months (calculated using green-algorithms.org v2.2 [1]).*

[1] Lannelongue, L., Grealey, J., Inouye, M., Green Algorithms: Quantifying the Carbon Footprint of Computation. Adv. Sci. 2021, 2100707.

Including the version of the tool is useful to keep track of the version of the data used.

Intensité carbone

$$\frac{CO_2}{Tasks} = \frac{CO_2}{Energy} \times \frac{Energy}{Tasks}$$

$$Energy = \frac{Energy}{Computations} \times Computations$$

Consommation d'énergie
Des processeurs

Faire un calcul unitaire, c’est décomposer un facteur et s’intéresser à un terme en supposant l’autre constant, et en négligeant non seulement les dépendances, mais en particulier l’effet de l’un sur l’autre

$$CO2 = \frac{CO2}{task} \times task$$

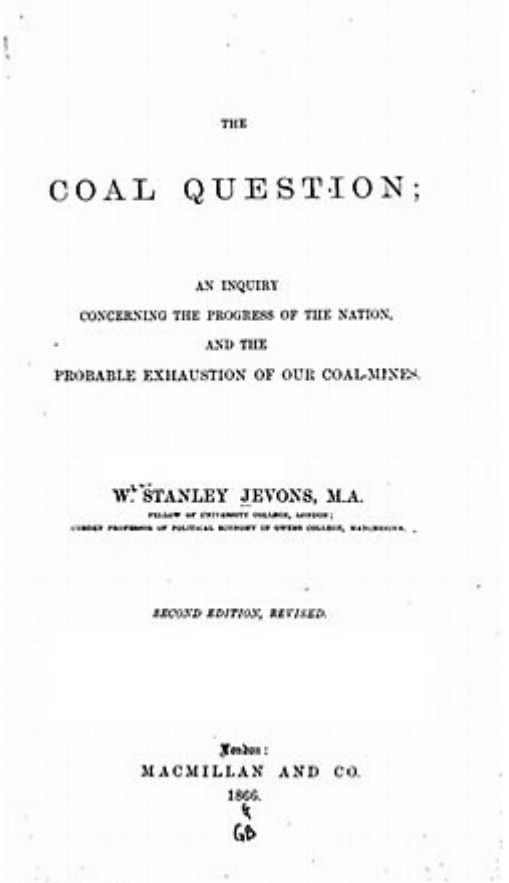
L’effet du terme unitaire sur l’autre est l’effet rebond : c’est à dire qu’il est susceptible de rendre les actions contre-productives

2/ Les effets rebond de l’optimisation

Un effet « paradoxal » : le « paradoxe de Jevons »
(économiste britannique du XIXe)

$$coal = \frac{coal}{machines} \times machines$$

« L'idée selon laquelle un usage plus économe du combustible équivaudrait à une moindre consommation est une confusion totale. C'est l'exact contraire qui est vrai. »



THE

COAL QUESTION;

AN INQUIRY
CONCERNING THE PROGRESS OF THE NATION,
AND THE
PROBABLE EXHAUSTION OF OUR COAL-MINES.

W. STANLEY JEVONS, M.A.

SECOND EDITION, REVISED.

London:
MACMILLAN AND CO.
1866.

Des paradoxes à expliquer :

- En 2008, le rapport « SMART 2020 » du Global e-Sustainability Initiative (GESI) estimait que les technologies numériques pouvaient permettre une réduction de 15 à 30% des émissions de gaz à effet de serre mondiaux d'ici 2020. Il ne mentionne pas les effets rebonds.

  2008 : 32 10^9 t CO2 numérique 0.6

  2020 : 35 10^9 t CO2 numérique 1.4

- De 1990 à 2012,

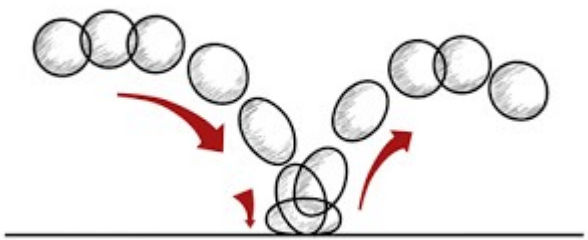
  consommation d'une voiture aux 100km : 8.3l → 6.7l

  consommation par habitant en France : 551l → 554l

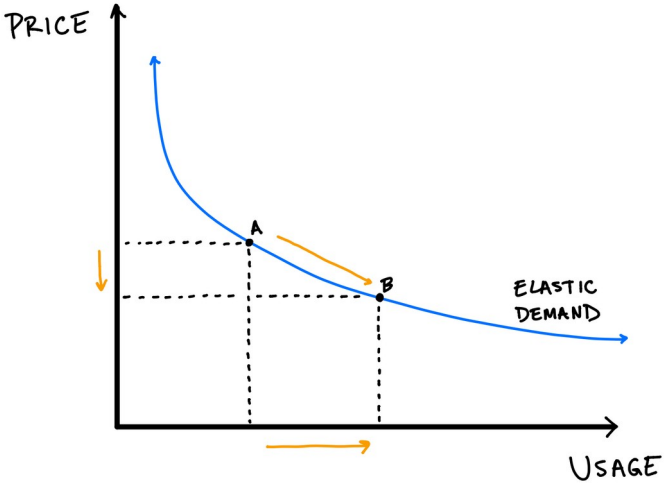
- De 1990 à 2005,

  la masse d'un téléphone est divisée par 4

  la masse des téléphones est multipliée par 8



De multiples variantes :



- On a accès à un article scientifique beaucoup plus rapidement, mais on passe plus de temps à faire la veille scientifique sur un sujet

- Plus les transports sont rapides, plus on passe de temps dans les transports

- Les écrans plats prennent moins de place, mais plus de place est consacrée aux écrans

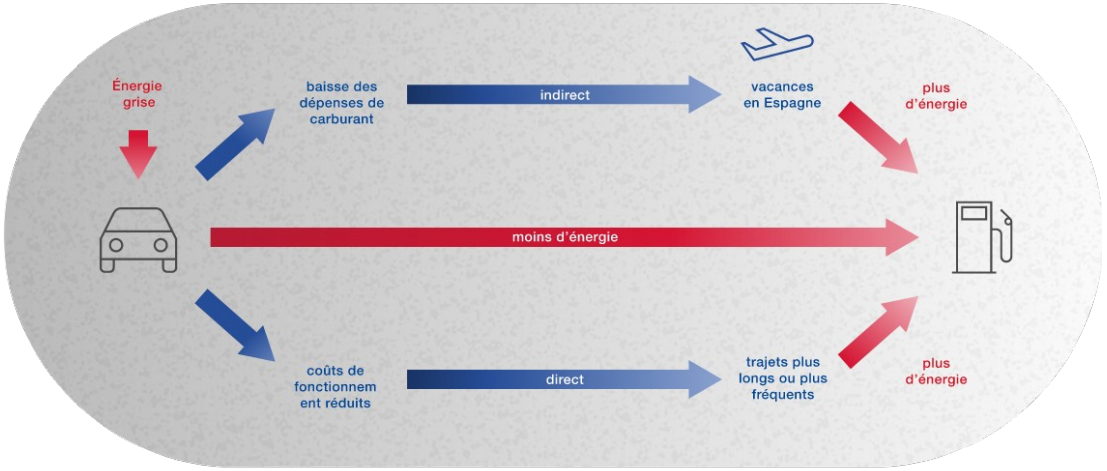
- Le prix d’un produit baisse, et son vendeur gagne plus

- Remplacer prématurément son équipement par un plus récent et plus performant peut augmenter la pression sur les ressources

Trois types d’effet rebond

- Directs : la baisse du coût d’une ressource déclenche une augmentation de la demande. Par exemple, si une machine à laver consomme moins d’énergie, les consommateurs peuvent se permettre de laver leur linge plus souvent.

- Indirects : les économies faites sur un secteur induisent des dépenses sur un autre. Par exemple, prendre son vélo tous les jours au lieu de la voiture fait réaliser une économie, qu’on peut dépenser en prenant des vacances plus loin.



- Structurels : la baisse du coût d’une ressource modifie les paramètres du système. Par exemple, un carburant moins cher permet d’habiter plus loin de son lieu de travail.

Mesurer et prévoir les effets rebond

L’effet rebond se mesure :

Si suite à une baisse du prix de l’énergie de 10 %, la consommation augmente de 2 %, l’effet rebond est de 20 %

Les effets rebond inférieurs à 100 % génèrent une vraie économie d’énergie

Les mesures sont parfois réalisées par des économistes dans des cas particuliers

(La causalité est difficile à établir)

3/ Comment mesurer un effet rebond pour la bioinformatique

1/ Ne pas isoler une variable et supposer que les autres sont indépendantes, comme « la vitesse de calcul ».

2/ Ne pas isoler son activité de celle des autres : si je fais mes calculs dans un pays à faible émission CO2/kWh, où les autres vont-ils faire leurs calculs ?

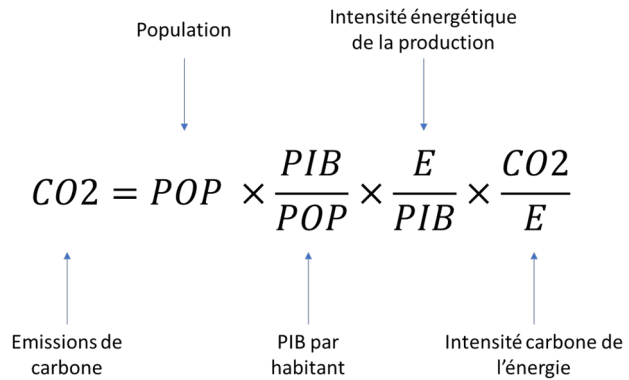
L’empreinte de la bioinformatique est globale :

$$CO2 = \frac{CO2}{task} \times task$$

Une décomposition et plusieurs effets rebond possibles

$$CO_2 = \frac{CO_2}{Energy} \times \frac{Energy}{Computations} \times \frac{Computations}{Data} \times \frac{Data}{Tasks} \times \frac{Tasks}{Users} \times Users$$

C’est une équation tautologique du type « Kaya » :



- Si on utilise du matériel moins consommateur, on consomme plus
- Si on utilise des logiciels plus efficaces, on augmente la taille des données ou le nombre d’analyses
- Si on programme des logiciels plus conviviaux, on augmente la taille de la communauté utilisatrice

$$CO_2 = \frac{CO_2}{Energy} \times \frac{Energy}{Computations} \times \frac{Computations}{Data} \times \frac{Data}{Tasks} \times \frac{Tasks}{Users} \times Users$$

Intensité carbone de l'énergie
moyenne 500g/kWh

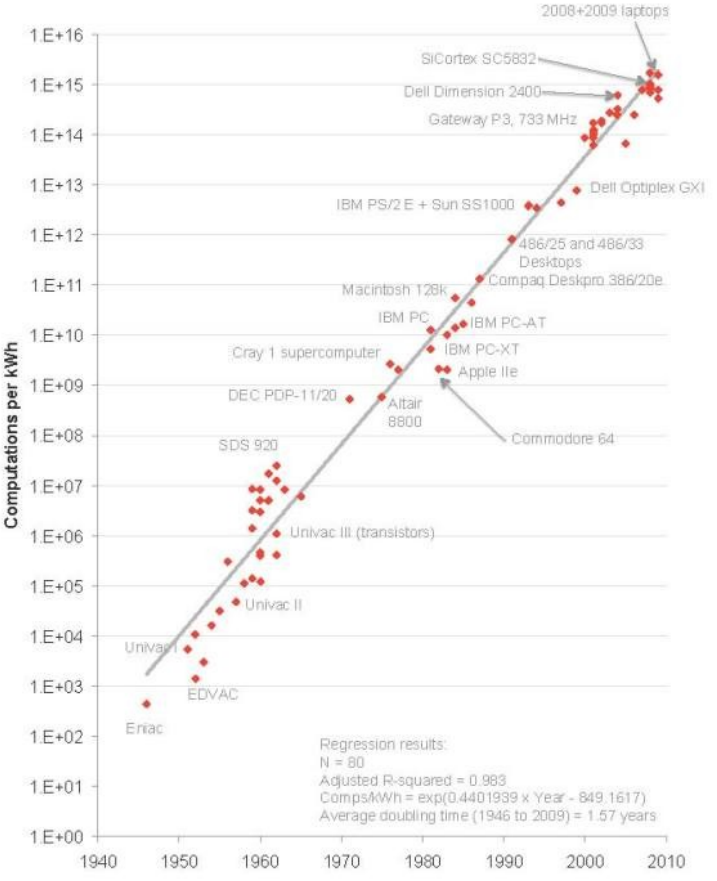
Moyens d'actions : fournisseurs d'énergie
au niveau mondial

Moins 0.25 % par an entre 1980 et 2020

On peut estimer l'intensité énergétique du matériel, c'est la loi de Koomey

(moins 20 % par an entre 1940 et 2010)

Dépend du progrès technologique extérieur à la bioinformatique



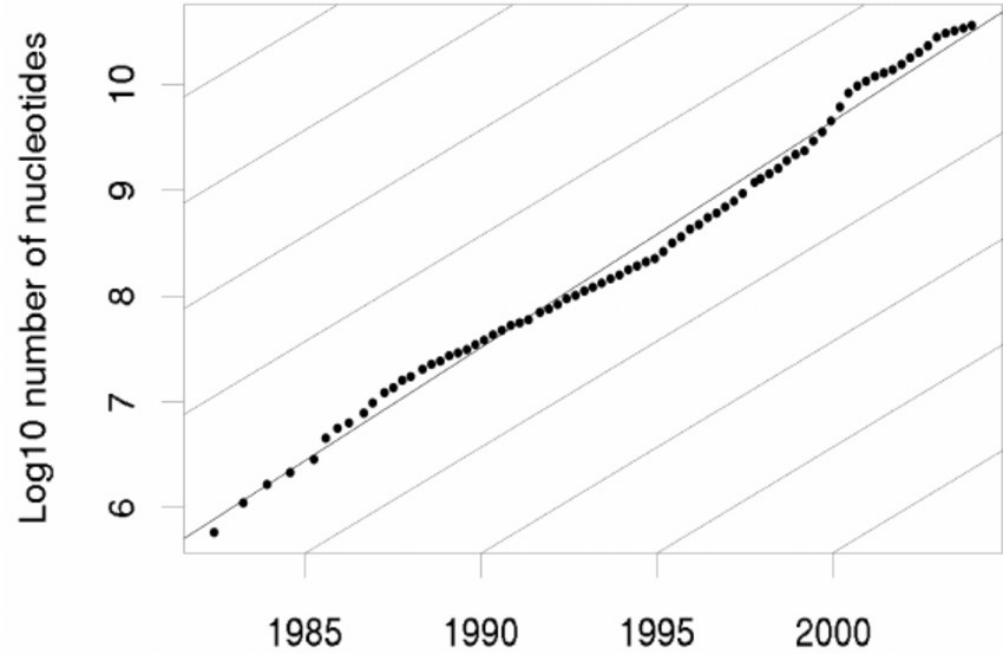
$$CO_2 = \frac{CO_2}{Energy} \times \frac{Energy}{Computations} \times \frac{Computations}{Data} \times \frac{Data}{Tasks} \times \frac{Tasks}{Users} \times Users$$

$$CO_2 = \frac{CO_2}{Energy} \times \frac{Energy}{Computations} \times \frac{Computations}{Data} \times \frac{Data}{Tasks} \times \frac{Tasks}{Users} \times Users$$

On peut jouer sur l'efficacité des algorithmes et des programmes ici

Par exemple, BoltLMM 2.3 (2018) est 2.68 fois plus efficace que BoltLMM 1.0 (2015), soit un gain annuel de 40 %

$$CO_2 = \frac{CO_2}{Energy} \times \frac{Energy}{Computations} \times \frac{Computations}{Data} \times \frac{Data}{Tasks} \times \frac{Tasks}{Users} \times Users$$



Séquençage et mise à disposition,

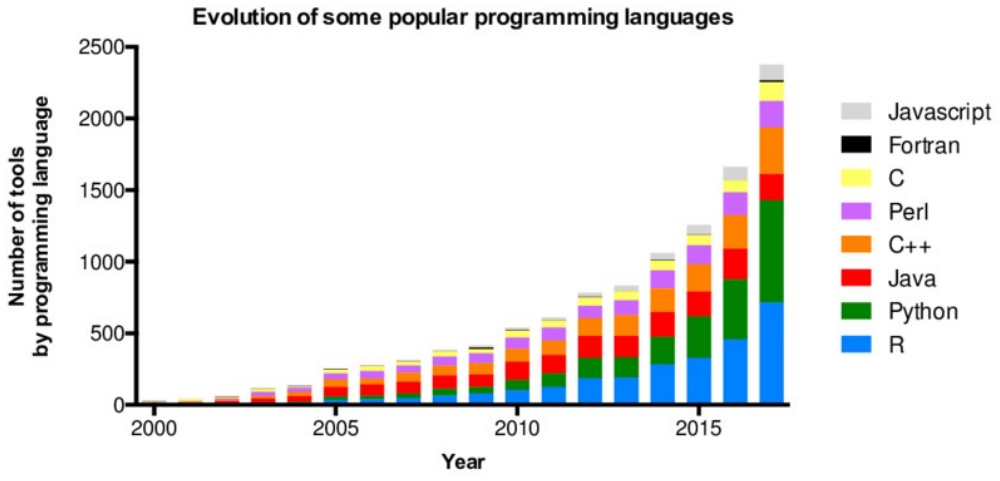
Plus 100 % tous les ans

$$CO_2 = \frac{CO_2}{Energy} \times \frac{Energy}{Computations} \times \frac{Computations}{Data} \times \frac{Data}{Tasks} \times \frac{Tasks}{Users} \times Users$$

Approximable par le nombre d'outils bioinformatiques

Plus 30 % par an

c



$$CO_2 = \frac{CO_2}{Energy} \times \frac{Energy}{Computations} \times \frac{Computations}{Data} \times \frac{Data}{Tasks} \times \frac{Tasks}{Users} \times Users$$

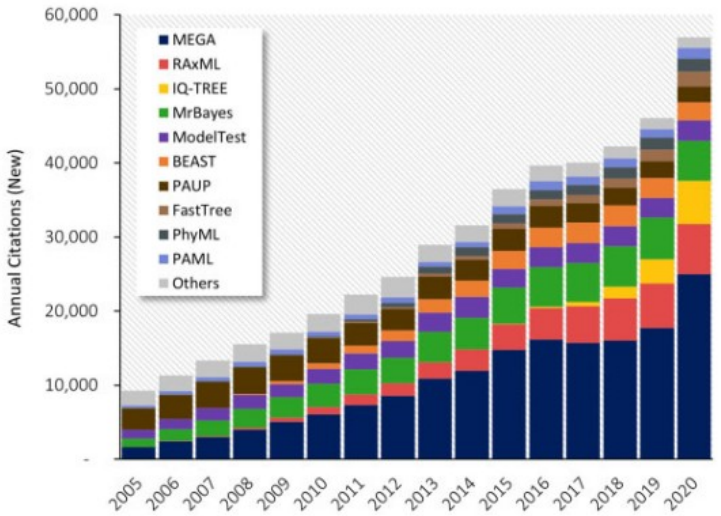


Fig. 1. The use of computational methods in molecular evolution has been increasing quickly, as seen in the annual counts of new research articles citing the use of major software packages for molecular evolutionary and phylogenetic analyses. Citation counts for software packages were obtained from Google Scholar (last accessed January 25, 2022) for 2005–2020. See supplementary material, Supplementary Material online for more details on software versions included.

Augmentation du nombre d'utilisateurs
(plus 25 % tous les ans)

Les leviers :

Maîtriser les usages, et pas seulement son propre usage

l’exemplarité peut compter, et donner un sens à son action (déontologisme) mais peut être frustrante et contre-productive (conséquentialisme)

L’efficacité d’une action individuelle exemplaire se heurte à l’effet rebond et au dilemme du prisonnier (équilibre de Nash)

Une règle collective peut proposer un moratoire sur les capacités de calcul ou sur le séquençage