Chimica computazionale

Loriano Storchi

loriano@storchi.org

http://www.storchi.org/

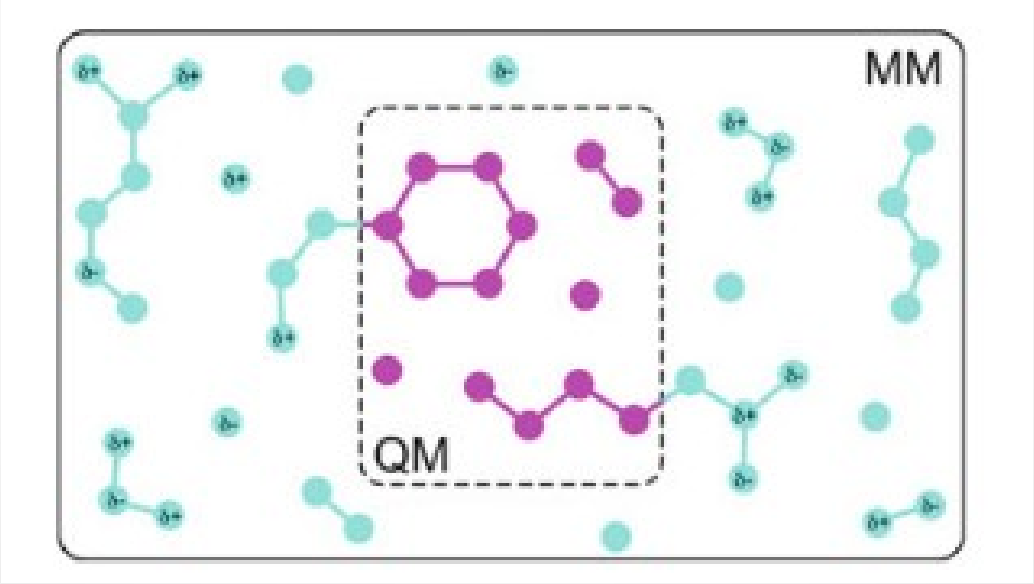
QM/MM CENNI

QM/MM

- La strategia ibrida QM / MM è stata originariamente introdotta da Warshel e Levitt
- **Si divide il sistema in regioni in cui si applica un livello di teoria diverso QM ed MM questo a causa del carattere locale della maggior parte delle reazioni chimiche.**
- Una distinzione solitamente può essere tra un "**centro di reazione" con atomi che sono direttamente coinvolti nella reazione** e **regione "spettatore”** in cui gli atomi non partecipano direttamente alla reazione.
- Ad esempio, la maggior parte delle reazioni in soluzione coinvolgono i reagenti e solo i primi gusci di solvatazione. Il solvente è difficilmente influenzato dalla reazione, ma può influenzare la reazione attraverso interazioni a lungo raggio.
- Lo stesso vale per la maggior parte degli enzimi, in cui **il processo catalitico è limitato a un sito attivo** il resto della proteina fornisce “solo” uno sfondo elettrostatico

QM/MM

- Una distinzione solitamente può essere tra un "**centro di reazione" con atomi che sono direttamente coinvolti nella reazione** e **regione "spettatore”** in cui gli atomi non partecipano direttamente alla reazione.



QM/MM

- L'energia potenziale ibrida QM / MM contiene **tre classi di interazioni:**

  1) interazioni tra gli atomi nella regione QM

  2) interazioni tra atomi nella regione MM

  3) interazioni tra QM e MM atomi

- **Le interazioni all'interno delle regioni QM e MM sono "ovvie" da descrivere, cioè sono descritte rispettivamente a livello di QM e MM**

- Le interazioni tra i due sottosistemi sono più difficili descrivere, e sono stati proposti **diversi approcci**

1 - Accoppiamento QM / MM

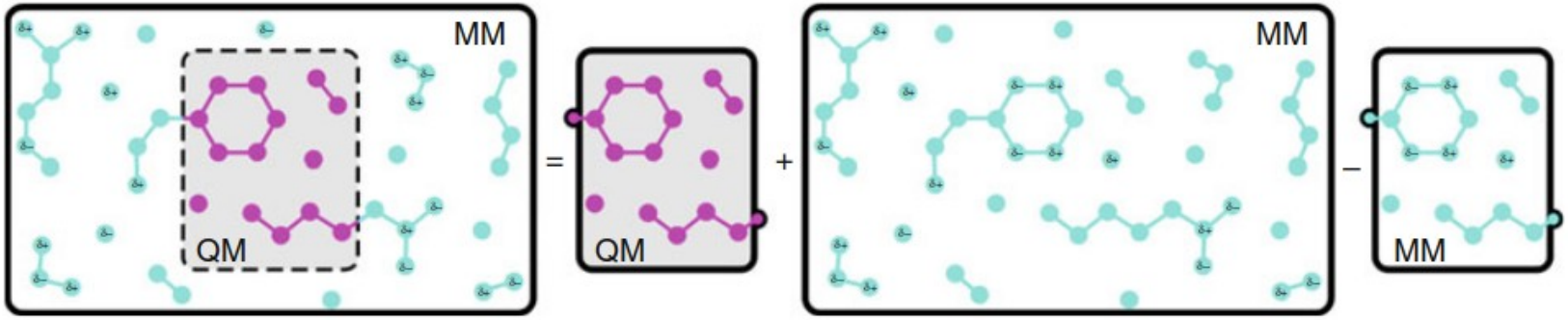
- Nel **primo tipo di accoppiamento** ad esempio:

$$V_{\mathrm{QM/MM}} = V_{\mathrm{MM}}(\mathrm{MM}+\mathrm{QM}) + V_{\mathrm{QM}}(\mathrm{QM}) - V_{\mathrm{MM}}(\mathrm{QM}).$$

- Quindi l'energia totale e' determinata in tre passi:
  - **Si calcola l'energia MM del sistema totale quindi subsystem MM e subsystem QM**
  - **Si calcola poi l'energia QM del susbsystem QM**
  - **Si sottrae l'energia MM del susbsystem QM**
- Non c'e' una "comunicazione" esplicita fra il sottosistema QM ed quello MM quindi sono "**disaccoppiati**" . **Chiaramente ho bisogno di un FF "buono" in grado di descrivere i cambiamenti che avvengono nel subsystem MM**

1 - Accoppiamento QM / MM

$$V_{\mathrm{QM/MM}} = V_{\mathrm{MM}}(\mathrm{MM} + \mathrm{QM}) + V_{\mathrm{QM}}(\mathrm{QM}) - V_{\mathrm{MM}}(\mathrm{QM}).$$



2 - Accoppiamento QM / MM

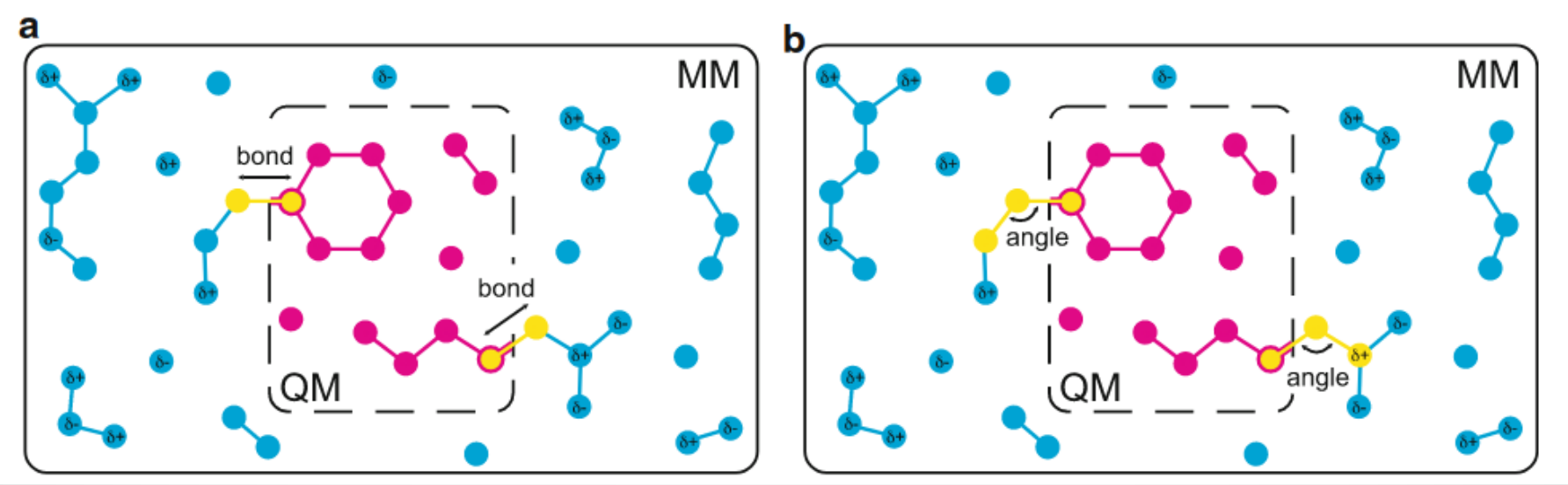
- Accoppiamento additivo:

$$V_{\mathrm{QM/MM}} = V_{\mathrm{QM}}(\mathrm{QM}) + V_{\mathrm{MM}}(\mathrm{MM}) + V_{\mathrm{QM-MM}}(\mathrm{QM} + \mathrm{MM}).$$

- In questo l'energia complessiva sara' la somma di tre contributi.
  - **Energia QM del subsystem QM**
  - **Energia MM del susystem MM**
  - **Termini di accoppiamento QM/MM**
- Come calcolo il contributo di accoppiamento ? Ho diversi schemi possibili tanto per dare **un'idea… (le interazioni fra subsystem QM ed MM sono trattate a livello MM)**

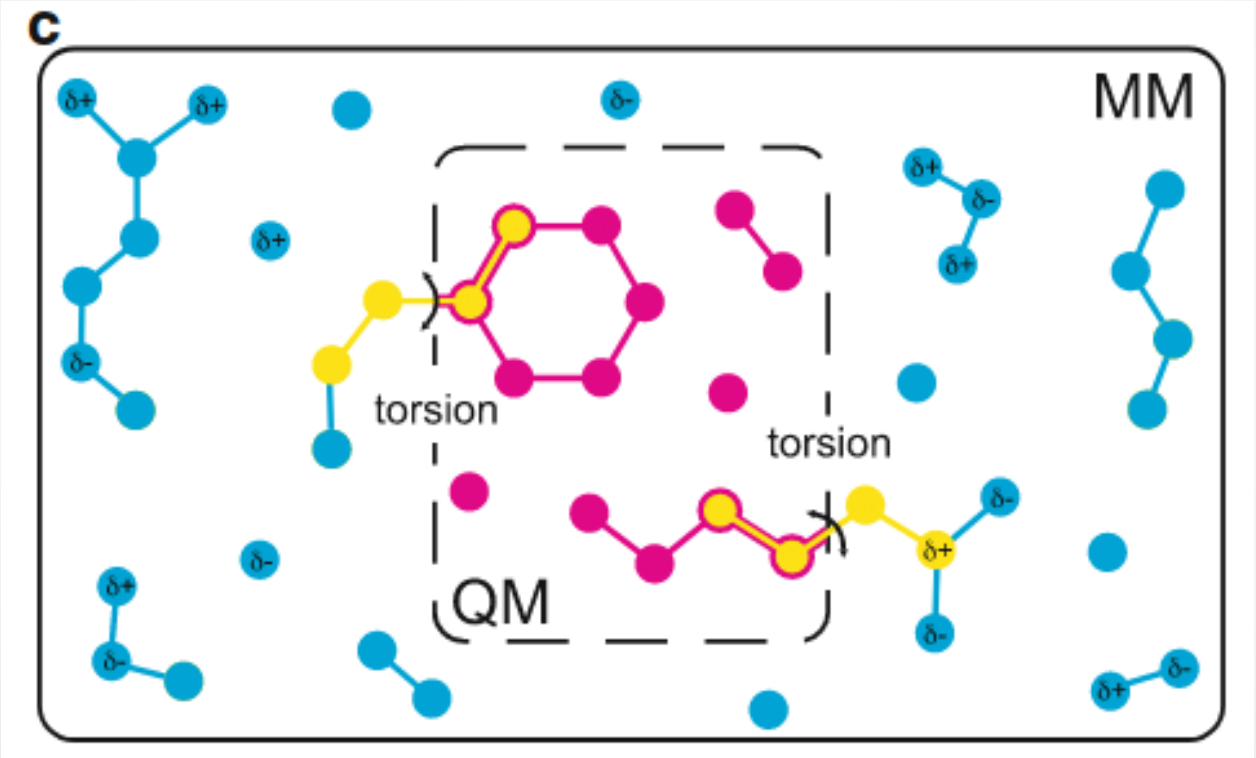
2 - Accoppiamento QM / MM

- I legami fra MM e QM sono modellati con un potenziale armonico cosi' come gli angoli



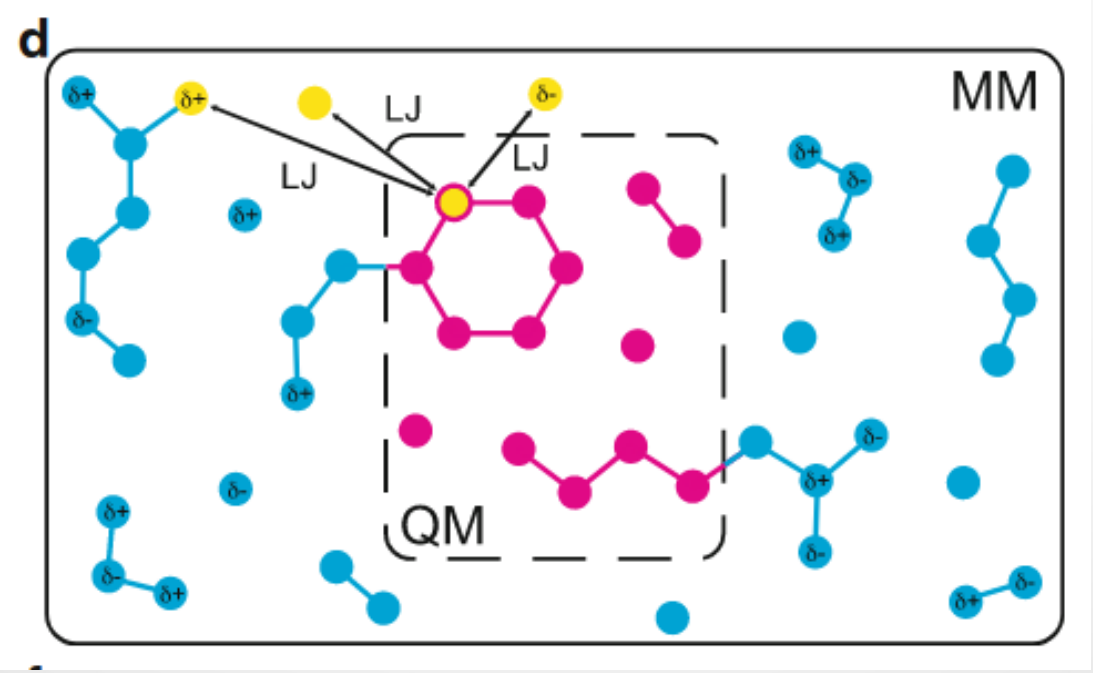
2 - Accoppiamento QM / MM

- I torsionali sono trattati con funzioni periodiche



2 - Accoppiamento QM / MM

- **Le interazioni non-bonded sono anch'esse trattate a livello MM. Nell'approccio più semplice la funzione d'onda del subsystem QM e' valutata considerando il sistema isolato quindi il subsystem MM non può indurre polarizzazione su quello QM**

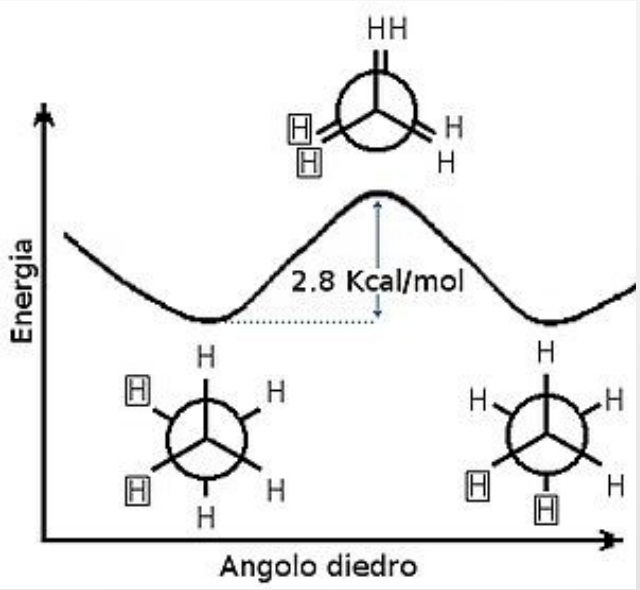


RICERCA CONFORMERI

Conformeri

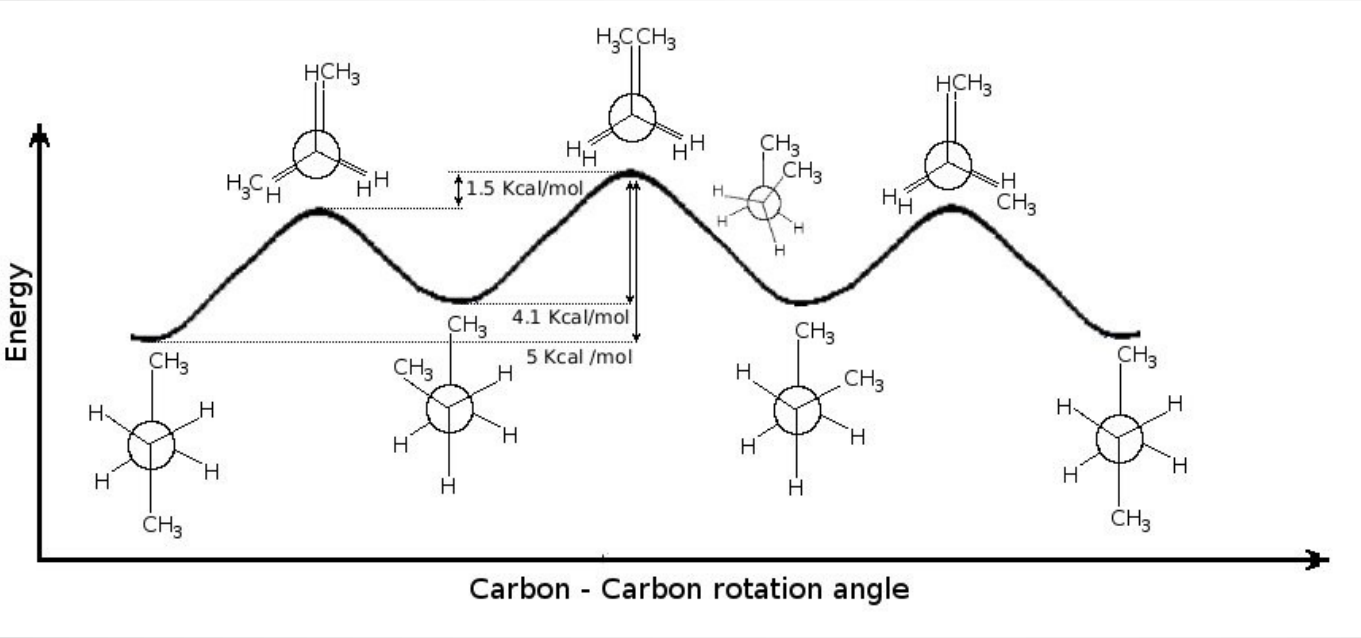
- In chimica, la conformazione è riferita alla forma della molecola, cioè alla geometria o disposizione spaziale degli atomi della molecola, che cambia in genere attraverso la rotazione interna (o torsione) attorno ai legami semplici o, talvolta, per inversione (o flipping) di particolari atomi tricoordinati (in genere di azoto) con geometria piramidale. (Wikipedia)

ETANO

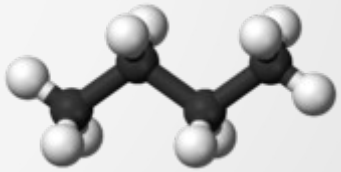


Conformeri

- Molte strutture hanno diversi conformeri
- Spesso separati da barriere di energie abbastanza basse tanto che temperatura ambiente le **sostanze sono una miscela di diverse configurazioni geometriche**



BUTANO



Conformeri

- Dunque abbiamo una PES con diversi **minimi locali della PES separati da barriere energetiche piccole**
- **I vari algoritmi di minimizzazione difficilmente arrivano a determinare il minimo assoluto**
- E' dunque spesso utile fare un ricerca di conformazionale anche perché **le proprietà di una struttura sono una media pesata**

$$p = \sum_{i=1}^{n} x_i p_i$$

- Dove p_i è la proprietà della i-esima conformazione mentre x_i è la frazione della conformazione

Conformeri

- L'insieme delle conformazioni ottenute mediante la ricerca può essere ulteriormente caratterizzato mediante metodi statistici. In questo caso la frazione x_i della conformazione i in una miscela all'equilibrio è data dalla **distribuzione di Boltzman**:

$$x_i = \frac{n_i}{N} = \frac{\exp\left(\frac{-E_i}{RT}\right)}{\sum_{j=1}^{N} \exp\left(\frac{-E_j}{RT}\right)}$$

R= costante dei gas
n_i = numero di molecole aventi conformazione i-esima
N = numero totale di molecole
E_i = energia molare della conformazione i-esima
T = temperatura

Conformeri

- Si possono usare diversi approcci nella ricerca dei conformeri. Si cerca di ottenere almeno tutti i conformeri energeticamente accessibili

- Ricerca sistematica
- Metodi stocastici (Metodi Monte Carlo)
- Algoritmi genetici
- Dinamica Molecolare

RICERCA SISTEMATICA

Ricerca sistematica

- In questo caso si esplora lo spazio conformazionale facendo cambiamenti strutturali sistematici e prevedibili. La più semplice ricerca conformazionale sistematica, chiamata anche **grid search (ricerca su griglia)**

1) **si identificano tutti i legami che possono ruotare nella molecola: le lunghezze di legame e gli angoli rimangono fissi.**
2) **ognuno di questi legami viene sistematicamente ruotato usando un incremento fisso fino a raggiungere i 360°.**
3) **tutte le conformazioni così generate sono soggette a minimizzazione dell'energia.**
4) **la ricerca termina quando tutte le possibili combinazioni degli angoli di torsione sono state generate e minimizzate**

Ricerca sistematica

- **Esplosione Combinatoriale**: chiaramente all'aumentare del numero di legami il numero delle combinazioni possibili aumenta esponenzialmente. Infatti se manteniamo lo stesso incremento θ_i per tutti gli angoli abbiamo

$$Numero\,delle\,combinazioni = \prod_{i=1}^{N} \frac{360\,°}{\theta_i} = \left(\frac{360\,°}{\theta_i} \right)^N$$

  dove N e' il numero di torsionali

- Ad esempio 5 legami con 30° di incremento avremo 248832 combinazioni , con 7 legami 36 milioni di strutture (immaginiamo 1 secondo a struttura)… facciamo "due conti"

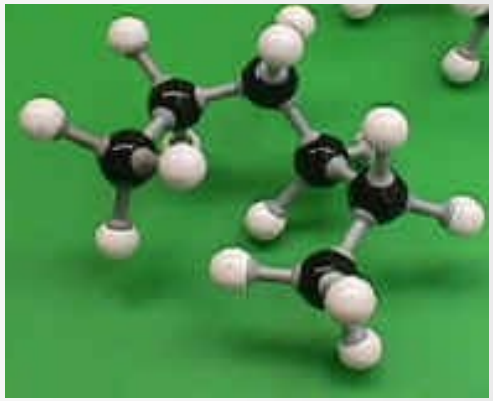
Ricerca sistematica

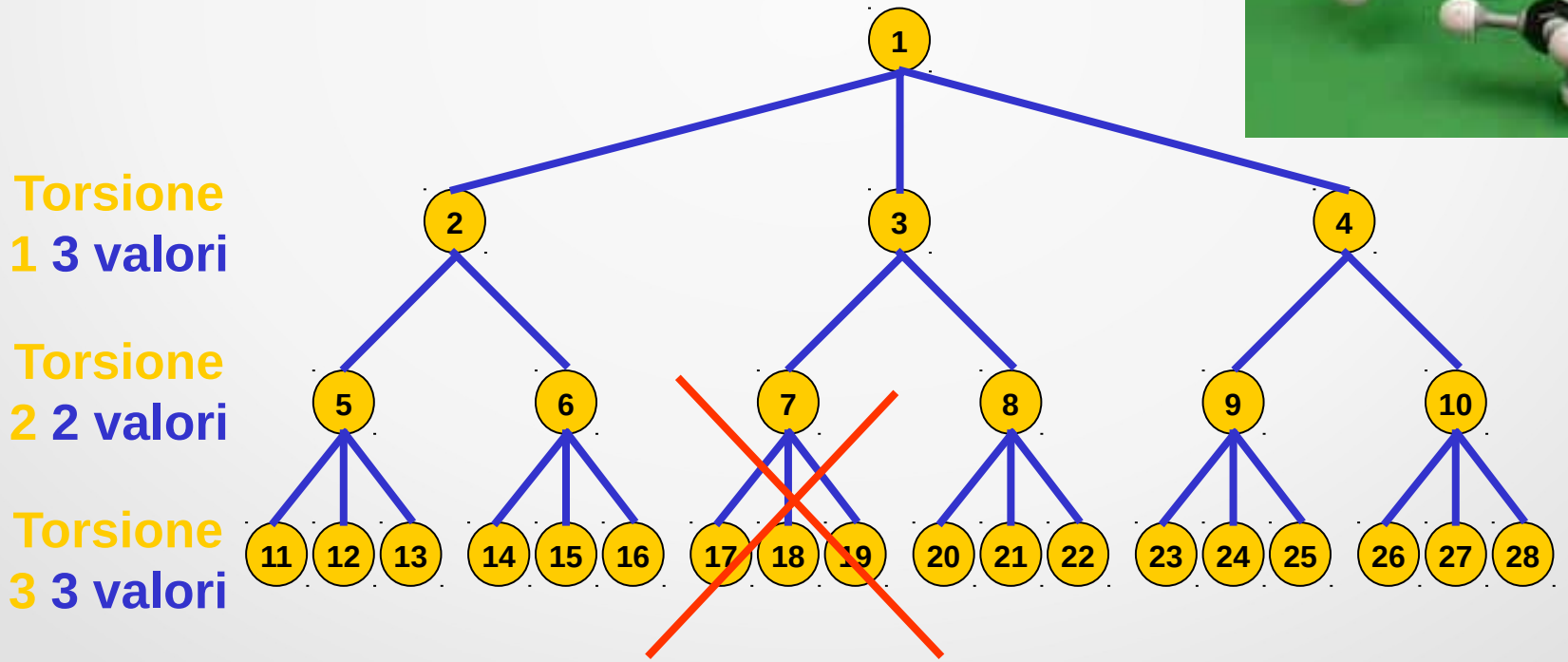
- Ad esempio 5 legami con 30° di incremento avremo 248832 combinazioni , con 7 legami 36 milioni di strutture (immaginiamo 1 secondo a struttura)… facciamo “due conti”

```
[redo@buchner 5]$ calc
C-style arbitrary precision calculator (version 2.12.5.0)
Calc is open software. For license details type:  help copyright
[Type "exit" to exit, or "help" for help.]

; 248832/3600
        69.12
; 36000000/3600
        10000
; 10000/24
        -416.66666666666666667
;
```

Ricerca sistematica

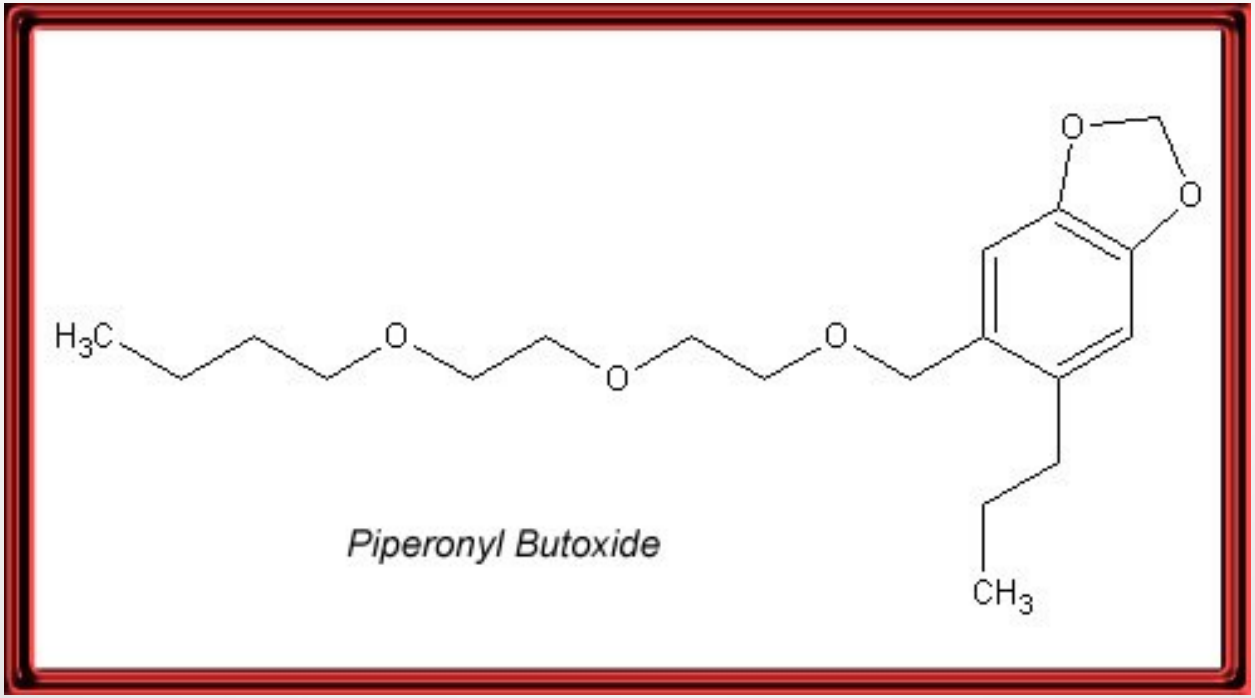
- Eliminiamo tutte le strutture problematiche. Ad esempio se la combinazione della variazione su due torsionali porta a strutture ad alta energia eliminano tutti i nodi corrispondenti





Ricerca sistematica

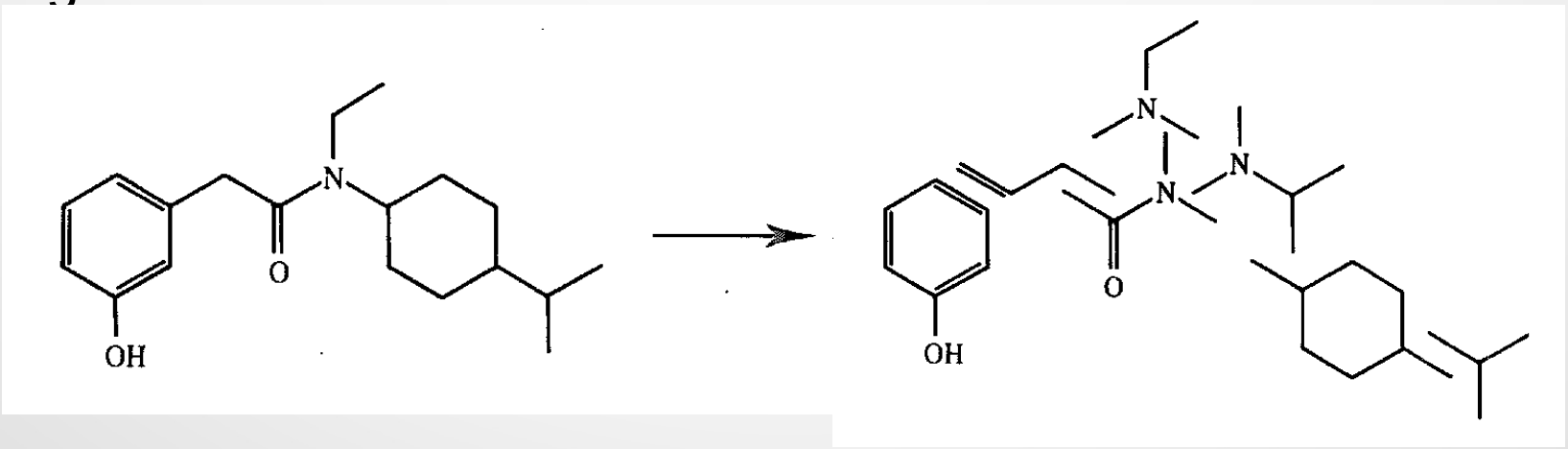
- Ad esempio qui abbiamo 13 torsionali anche considerando un incremento di 120° otteniamo 1.595.323 conformazioni
- Posso eliminare dunque tutte le conformazioni in cui la distanza fra atomi (o di idrogeno o pesanti) e' inferiori a certi valori soglia , cosi' da diminuire le combinazioni possibili



Piperonyl Butoxide

Ricerca sistematica

- Si possono poi usare metodi Model-building in cui si usano **frammenti molecolari a partire dai quali costruire le conformazioni**. Con gli approcci a “costruzione da modello” all’analisi conformazionale **si costruiscono le conformazioni di una molecola unendo insieme strutture tridimensionali di frammenti molecolari.** Tali metodi possono essere più efficaci poiché ci sono molte meno combinazioni di valori di angoli torsionali.



METODI RANDOM

Ricerca sistematica

- Si esplora lo spazio conformazionale partendo da una data conformazione e **facendo cambiamenti casuali** su di essa in una serie di step successivi che portano alla generazione di nuove conformazioni in un ordine impredittibile. Uno schema generale di ricerca conformazionale casuale, consiste nei seguenti passi:

1) **Si sceglie una data conformazione** di partenza e si inizia un ciclo iterativo.
2) **Si varia in maniera casuale la geometria di tale conformazione**. Ciò può essere fatto sia variando le coordinate atomiche degli atomi che gli angoli di torsione dei legami che possono ruotare.
3) La nuova struttura è soggetta a **minimizzazione** dell'energia e genera una nuova conformazione
4) **Si confronta tale conformazione con quelle dei cicli precedenti (RMSD)** e se essa non è stata già trovata viene aggiunta alla lista delle conformazioni.
5) **Si sceglie una conformazione di partenza per il ciclo successivo e si torna al punto 2.**
6) La ricerca termina quando viene verificato un **opportuno criterio di terminazione.**

Metodo Random

Metodo Random

- Scelta della conformazione per l'iterazione successiva
  - Conformazione dell'ultimo ciclo
  - Scelta casualmente fra quelle generate nei cicli precedenti
  - La meno utilizzata fra quelle generate nei cicli precedenti
  - Quella di più bassa energia fra le generate nei cicli precedenti
  - Scelta col criterio Monte Carlo - Metropolis
- Criterio di terminazione
  - Dopo un numero di cicli predefinito
  - Fino a che nessuna nuova struttura viene generata

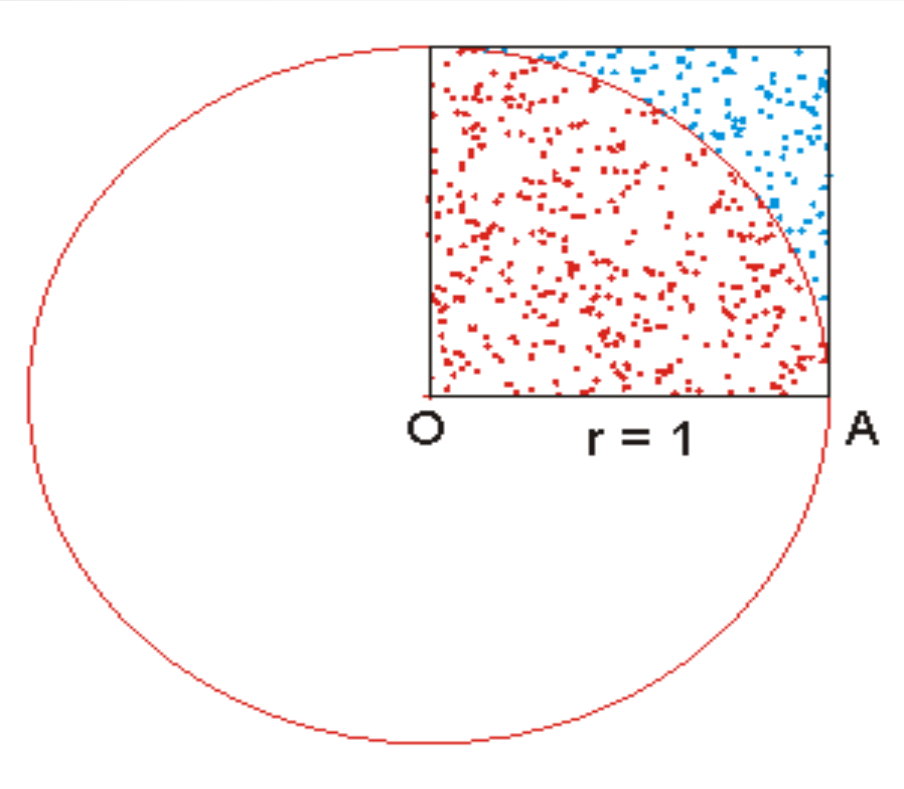
METODO MONTE-CARLO

Metodo Monte-Carlo

- Vista l’importanza dei Metodi-Montecarlo in diversi settori della Fisica-Chimica e Matematica vale la pena fare una breve introduzione generale
- Le sue origini risalgono alla metà degli anni 40 nell'ambito del **Progetto Manhattan**. I formalizzatori del metodo sono **Enrico Fermi, John von Neumann e Stanisław Marcin Ulam**, il nome Monte Carlo fu inventato in seguito da **Nicholas Constantine Metropolis** in riferimento al noto casinò situato a Monte Carlo, nel Principato di Monaco [Wikipedia]
- Quando la soluzione analitica e’ troppo complicata (le combinazioni possibili sono troppe) cercare la soluzione di un problema rappresentandola quale parametro di una ipotetica popolazione e nello stimare tale parametro tramite l’esame di un campione della popolazione ottenuto mediante sequenze di numeri casuali.

Metodo Monte-Carlo

- Ad esempio calcolo integrali o simili….



$$A_S = (2r)^2 = 4r^2$$

$$A_C = \pi r^2$$

$$\pi = 4 \times \frac{A_C}{A_S}$$

Metodo Monte-Carlo

- Cloniamo un repo:

```
git clone https://github.com/lstorchi/teaching
```

- E vediamo un po' di codice:

```
cd teaching/mcpi/
i (master)]$ python mcpi.py
```

- E proviamo ad eseguirlo:

```
[redo@buchner mcpi (master)]$ python mcpi.py 100
3.28
[redo@buchner mcpi (master)]$ python mcpi.py 10000
3.1456
```

- Come funziona ?

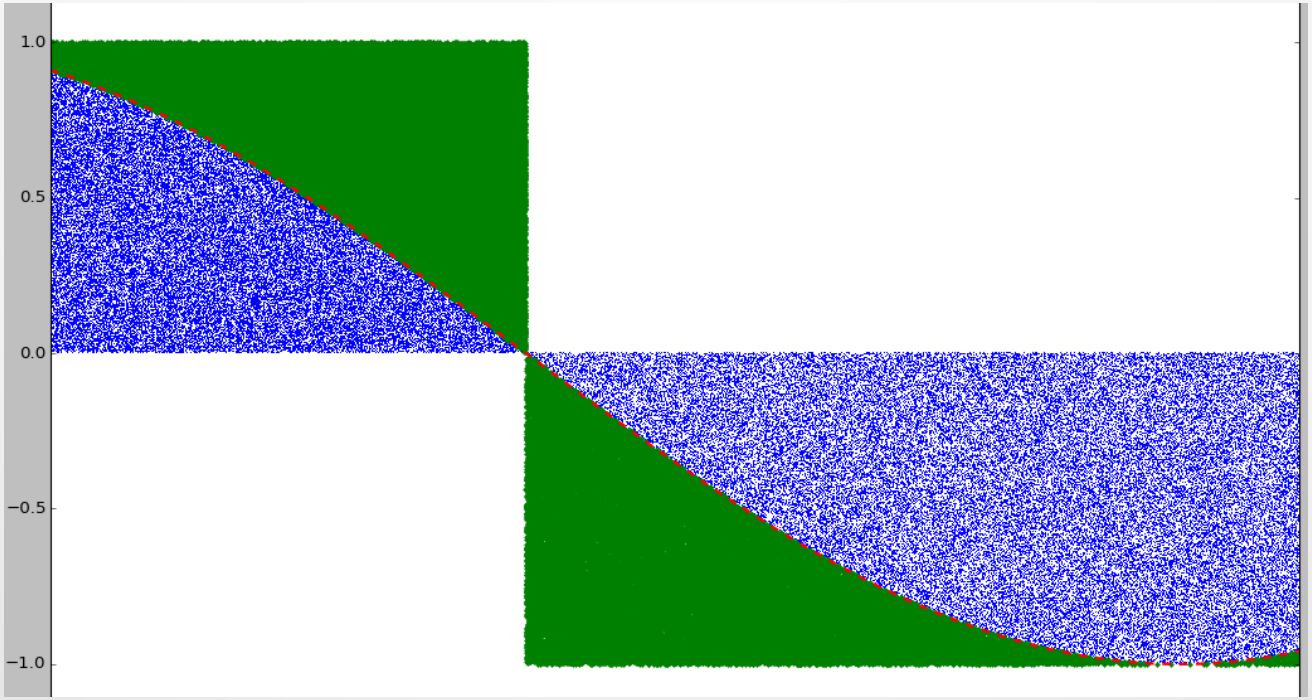
Metodo Monte-Carlo

```
import random
import math
import sys

DIM = 0

if len(sys.argv) != 2:
    print "usage: ", sys.argv[0] , " NUM "
    exit(1)
else:
    DIM = int(sys.argv[1])

circle_count = 0

for i in range(0,DIM):

   x = random.uniform(0.0, 1.0)
   y = random.uniform(0.0, 1.0)

   if ((math.pow(x, 2.0) + math.pow(y, 2.0)) < 1.0):
     circle_count = circle_count + 1

pi = float(circle_count) / float(DIM)

print 4.0 * pi
```

Metodo Monte-Carlo

- Calcolo di integrali stesso approccio

$$\int_2^5 \sin(x)\,dx = \cos(2) - \cos(5) = -0{,}69981$$



Metodo Monte-Carlo

- Anche qui vediamo il codice, e facciamo qualche test:

```
[redo@buchner teaching (master)]$ cd mcsin/
[redo@buchner mcsin (master)]$ ls
mcsin.py  mcsin_wplt.py
[redo@buchner mcsin (master)]$ python mcsin.py
usage:  mcsin.py  NUM
[redo@buchner mcsin (master)]$ python mcsin.py 100
-0.84
[redo@buchner mcsin (master)]$ python mcsin.py 1000
-0.834
[redo@buchner mcsin (master)]$ python mcsin.py 10000
-0.7614
[redo@buchner mcsin (master)]$ python mcsin.py 100000
-0.69114
[redo@buchner mcsin (master)]$
```

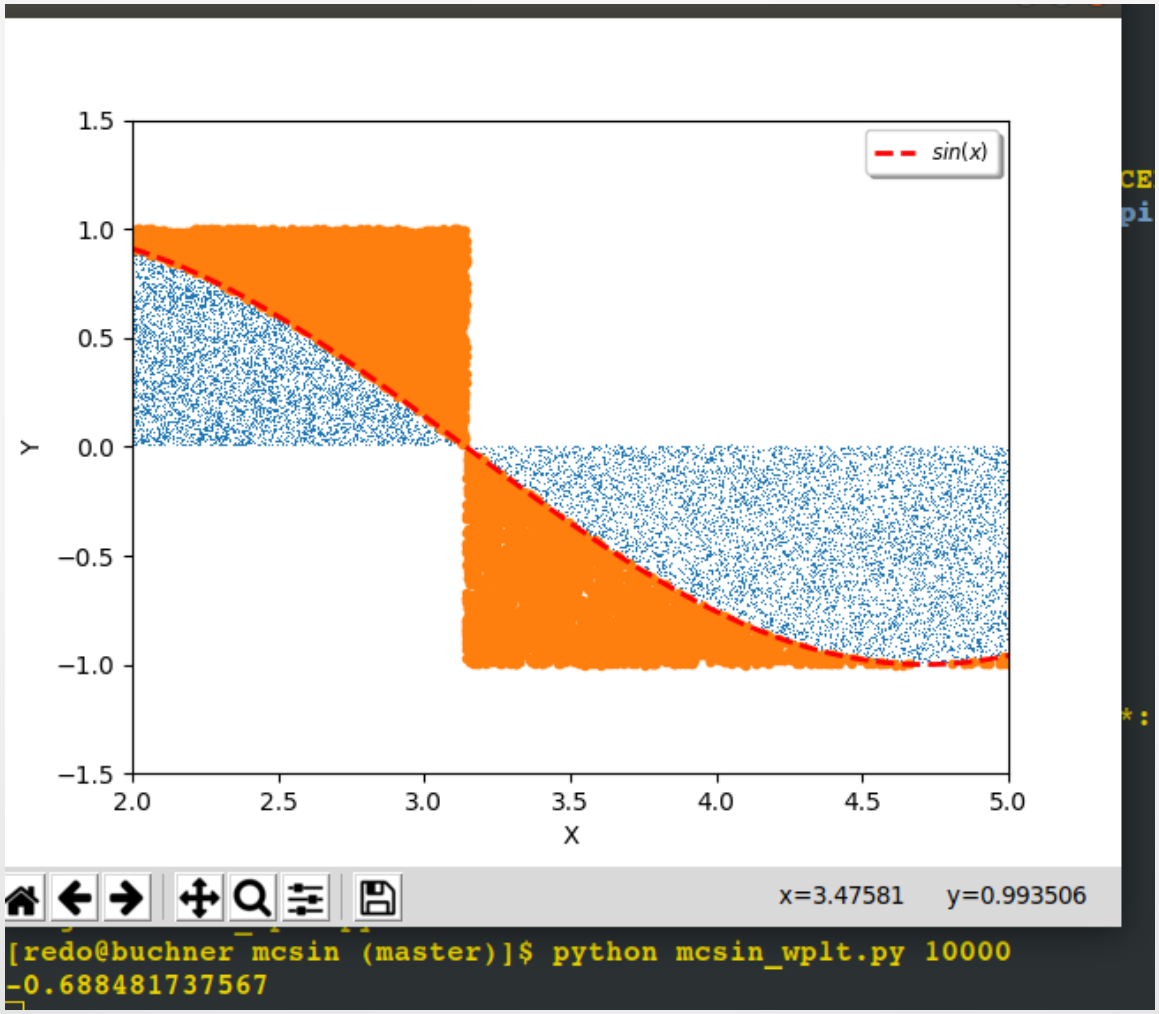
Metodo Monte-Carlo

- Anche qui vediamo il codice, e facciamo qualche test:

```
xmin = 2.0
xmax = 5.0
ymin = -1.0
ymax = 1.0

for i in range(0,DIM):

   x = random.uniform(xmin, xmax)
   y = random.uniform(ymin, ymax)

   #if x < math.pi :
   #  if ((y <= math.sin(x)) and (y >= 0.0)):
   #    rect_count = rect_count + 1
   #elif x > math.pi :
   #  if ((y >= math.sin(x)) and (y <= 0.0)):
   #      rect_count = rect_count - 1

   if math.sin(x) > 0.0 :
     if ((y <= math.sin(x)) and (y >= 0.0)):
       rect_count = rect_count + 1
   elif math.sin(x) < 0.0 :
     if ((y >= math.sin(x)) and (y <= 0.0)):
         rect_count = rect_count - 1

p = (xmax - xmin) * (ymax - ymin) * (float(rect_count) / float(DIM))
```

Metodo Monte-Carlo

- Anche qui vediamo il codice, e facciamo qualche test:



==========

Metodo Monte-Carlo

- Dopo questa breve digressione torniamo ai conformeri

1) Partendo da una geometria iniziale q^a si ottiene un minimo (minimizzazione di geometria) q^{a+k} ad energia $E(q^{a+k})$

2) Si fa una perturbazione “random”, variando ad esempio un legame, e si ottiene una nuova geometria q^b e si minimizza arrivando a q^{b+k} ed energia $E(q^{a+k})$

3) Per la perturbazione successiva da quale geometria si parte ? **Se la nuova energia e’ inferiore si parte da q^{b+k} per la nuova perturbazione. Se $E(q^{a+k}) > E(q^{a+k})$ la probabilita’ di partire dalla nuova geometria decide usando l’equazione di Boltzmann vista precedentemente**. Ad esempio nel caso di due geometrie:

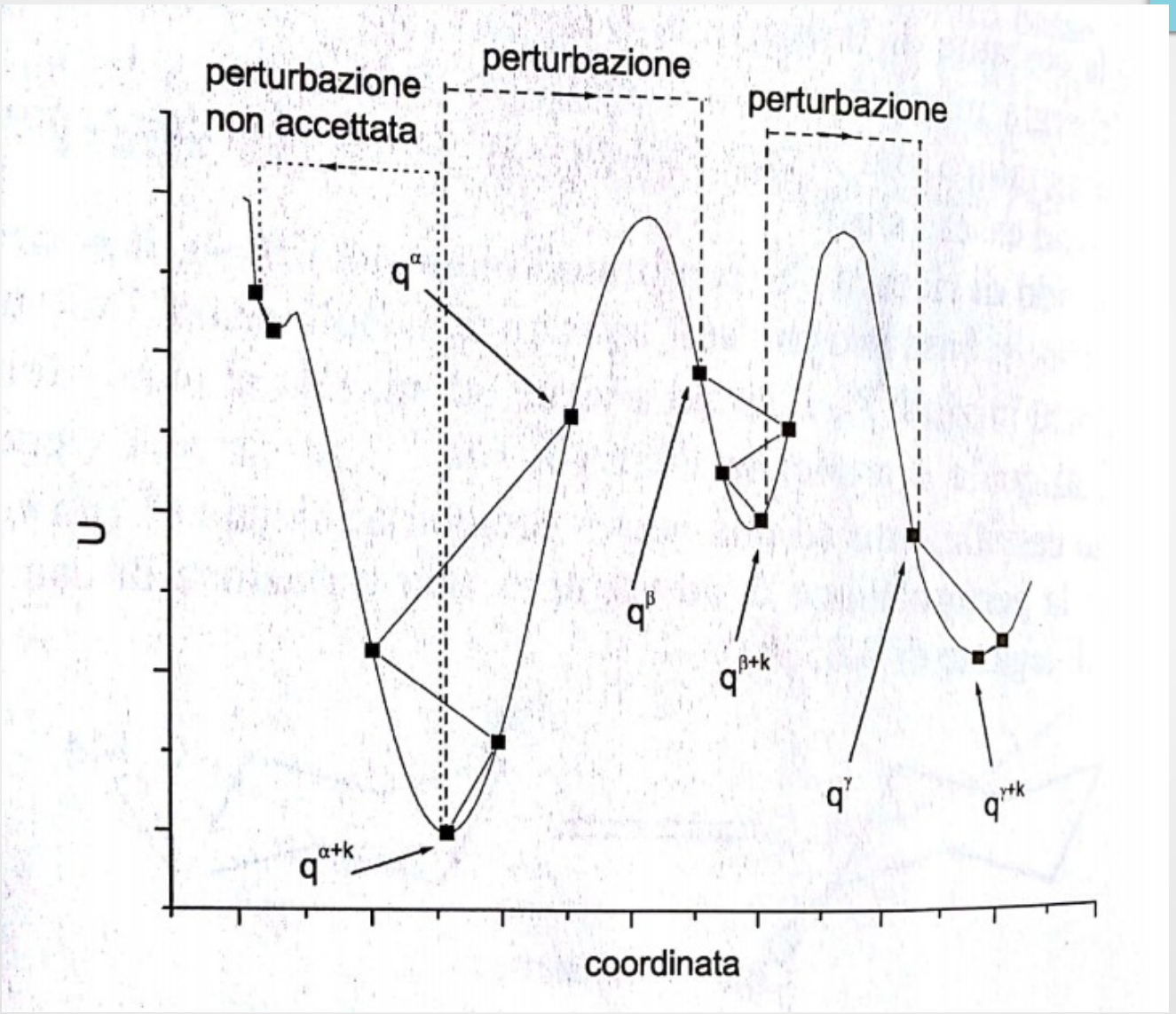
$$\frac{N(y)}{N(z)} = e^{-\frac{U(y)-U(z)}{K_B T}}$$

Metodo Monte-Carlo

Alcune considerazioni:

- **Se si imposta una temperatura maggiore la probabilita' di partire da una nuova geometria ad energia maggiore aumenta**
- Partire sempre da nuove geometrie potrebbe pero' portare a configurazioni, confomeri, chimicamente poco sensati
- **Non e' detto che ogni minimizzazione porti a nuovi minimi locali**
- Il numero di perturbazioni plausibili viene calcolato partendo da alcune considerazioni strutturali sulla molecola. Ad esempio numero di legami ed anelli nella struttura , numero di legami singoli , doppi o tripli.

Metodo Monte-Carlo



ALGORITMI GENETICI

Algoritmi Genetici

Anche qui facciamo una breve digressione (Evolutionary Computing and Machine Learning)

- Possiamo sempre usare il caso ? Il **teorema della scimmia instancabile** o **teorema delle scimmie infinite** afferma che una scimmia che prema a caso i tasti di una tastiera per un tempo infinitamente lungo quasi certamente riuscirà a comporre qualsiasi testo prefissato. **Ma in un tempo finito, come posso migliorare ?**
- Chiaramente come abbiamo visto nel caso dei conformeri il numero di combinazioni possibili in generale e' tale che si deve trovare un criterio per limitare il numero di tentativi
- Gli algoritmi genetici chiaramente si basano sulla teoria di **della selezione naturale di Darwin**

Algoritmi Genetici

Vediamo i punti fondamentali della teoria di Darwin. Perche' si verifichi l'evoluzione naturale necessitiamo di questi **tre "ingredienti" fondamentali**. Come vedremo questi saranno poi i tre ingredienti fondamentali degli algoritmi genetici

1) **Ereditarietà**: Deve esserci un processo attraverso il quale i bambini ricevono le proprietà dei loro genitori. **Se le creature vivono abbastanza a lungo da riprodursi, i loro tratti vengono trasmessi ai loro figli nella generazione successiva**.

Algoritmi Genetici

1)**Mutazioni / Variazioni:** Ci **deve essere una varietà di tratti presenti nella popolazione o un mezzo con cui introdurre variazion**i. Per esempio, diciamo che c'è una popolazione di coleotteri in cui tutti i coleotteri sono esattamente uguali: stesso colore, stesse dimensioni, stessa apertura alare, stesso tutto. Senza alcuna varietà nella popolazione, i bambini saranno sempre identici ai genitori e tra loro. **Senza mutazioni nuove combinazioni di tratti non possono mai accadere e nulla può evolversi.**

Algoritmi Genetici

1) **Selezione: Ci deve essere un meccanismo attraverso il quale alcuni membri di una popolazione hanno l'opportunità di essere genitori e trasmettere le loro informazioni genetiche e altri no**. Questo è in genere definito come "**sopravvivenza del più adatto**" o del "**migliore**" **in relazione al contesto**. Ad esempio, supponiamo che una popolazione di gazzelle sia inseguita dai leoni ogni giorno. Le gazzelle più veloci hanno maggiori probabilità di sfuggire ai leoni e quindi hanno maggiori probabilità di vivere più a lungo e hanno la possibilità di riprodursi e trasmettere i loro geni ai loro figli. Il termine più adatto, tuttavia, può essere un poco fuorviante. Generalmente, pensiamo che significhi più grande, più veloce o più forte. Mentre questo può essere il caso in alcuni casi, **la selezione naturale opera sul principio che alcuni tratti sono più adatti per l'ambiente della creatura e quindi producono una maggiore probabilità di sopravvivere e riprodursi**. **Non ha nulla a che fare con una determinata creatura che è "migliore" (dopotutto, questo è un termine soggettivo) o più "fisicamente in forma"**. **Nel caso delle nostre scimmie digitazione, per esempio, una scimmia più "in forma" è quella che ha digitato una frase più vicina a "essere o non essere".**

Algoritmi Genetici

Torniamo alla scimmia/scimmie che vogliono scrivere ad esempio anche solamente **“to be or not to be that is the question”**

- Se la scimmia inizia a digitare, la possibilità che ottenga il primo carattere a destra è 1 su 27. Poiché la probabilità che otterrà il secondo carattere è anche 1 su 27, ha un 1 su 27 * 27 possibilità di atterrare il primo due caratteri nell'ordine corretto, che segue direttamente dal concetto di probabilità eventi indipendenti. Pertanto, la probabilità complessiva che la scimmia digiti la frase completa è:

(1/27) moltiplicato per se stesso 39 volte, cioè $(1/27)^{39}$

E quindi:

1/665559370338678226078955492410964829530176158347352261 63

Algoritmi Genetici

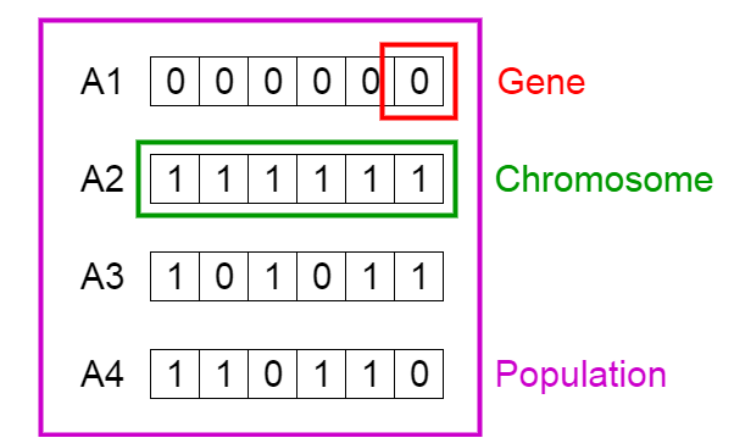
Ora, vale la pena notare che questo problema (arrivare alla frase "to be or not to be that is the question") è ridicolo solo un **toy model**. Dal momento che conosciamo la risposta, tutto ciò che dobbiamo fare è digitarlo. Ecco uno codice minimale che risolve al volo la questione:

```
[redo@buchner 5]$ python
Python 2.7.15rc1 (default, Apr 15 2018, 21:51:34)
[GCC 7.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print "to be or not to be that is the question"
to be or not to be that is the question
>>>
```

Algoritmi Genetici

Vediamo i tratti fondamentali di un algoritmo genetico immaginando di voler arrivare come obbiettivo ad una **stringa 11111** quindi 5 bit a 1.

- Il processo inizia con un gruppo di individui che è chiamato **Popolazione**. Ogni individuo è una soluzione al problema che vuoi risolvere. Nel nostro caso ad esempio un set iniziale di stringhe di 1 e 0. Un **individuo** è caratterizzato da un insieme di parametri (variabili) noti come **geni**. I geni sono uniti in una stringa per formare un **cromosoma** (**soluzione**).

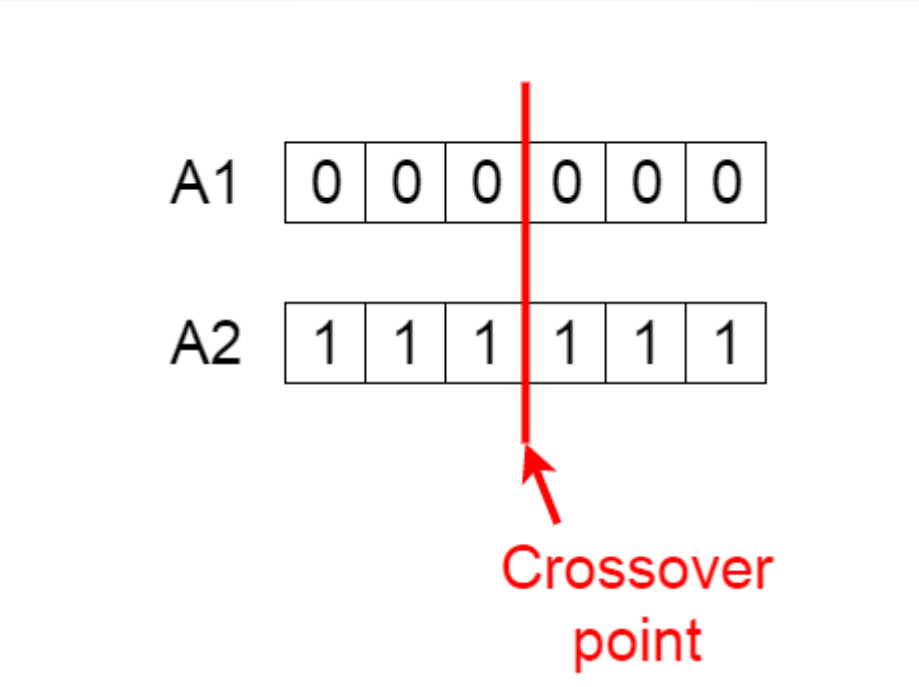


Algoritmi Genetici

- **Fitness Function** : **La funzione produrrà un punteggio numerico per descrivere l'idoneità di un dato membro della popolazione**. Questo, ovviamente, non è affatto come funziona il mondo reale. **Le creature non hanno un punteggio; semplicemente sopravvivono o no**. Ma nel caso del tradizionale algoritmo genetico, nel quale stiamo cercando di evolvere una soluzione ottimale per un problema, **dobbiamo essere in grado di valutare numericamente qualsiasi soluzione possibile data**.
- **Selezione**: L'idea della fase di selezione è selezionare gli individui più adatti e lasciarli passare i loro geni alla generazione successiva. Due coppie di individui (**genitori**) **sono selezionati in base ai loro punteggi di fitness**. Gli individui con alta idoneità hanno più possibilità di essere selezionati per la riproduzione.

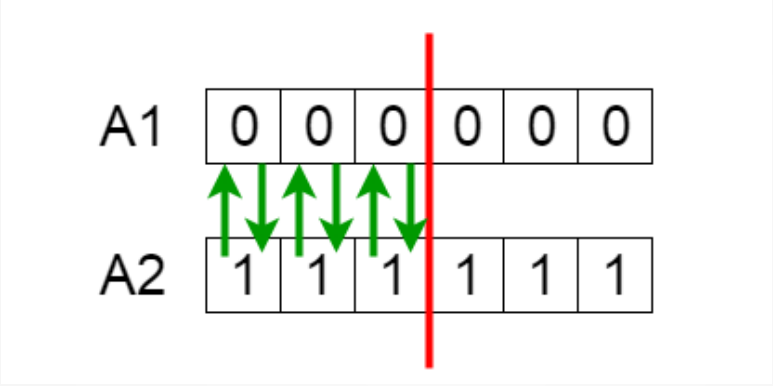
Algoritmi Genetici

- **Crossover**: Per ogni coppia di genitori da accoppiare, viene scelto un punto di crossover a caso all'interno dei geni.

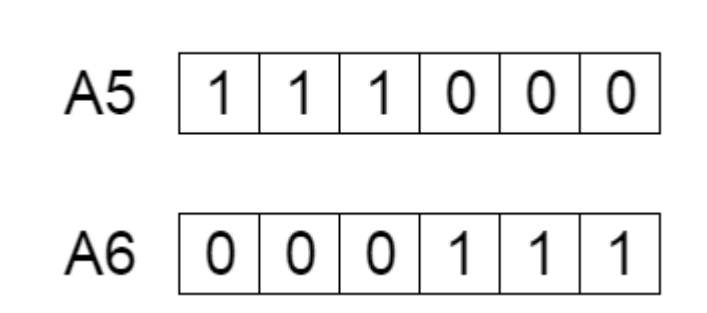


Algoritmi Genetici

- **I figli** vengono creati scambiando tra loro i geni dei genitori fino al raggiungimento del punto di **crossover**.

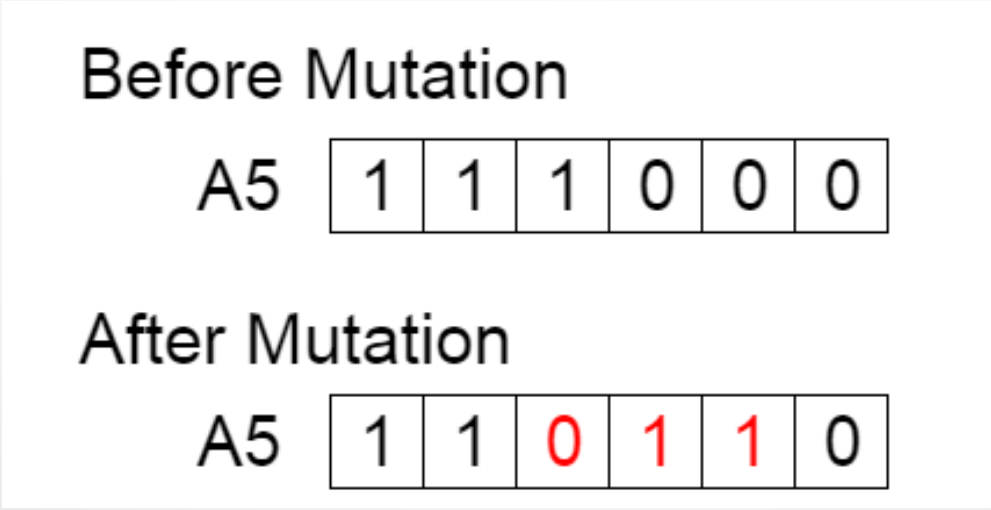


- A questo punto i figli sono aggiunti alla popolazione:



Algoritmi Genetici

- **Mutazione**: un altro aspetto fondamentale degli algoritmi genetici e' la ovvia possibilita' di avere mutazioni. Mutazioni che possono essere benefiche o meno. **In alcuni nuovi discendenti formati, alcuni dei loro geni possono essere sottoposti a una mutazione con una bassa probabilità casuale**. Ad esempio alcuni dei bit nella stringa di bit possono essere capovolti.



Algoritmi Genetici

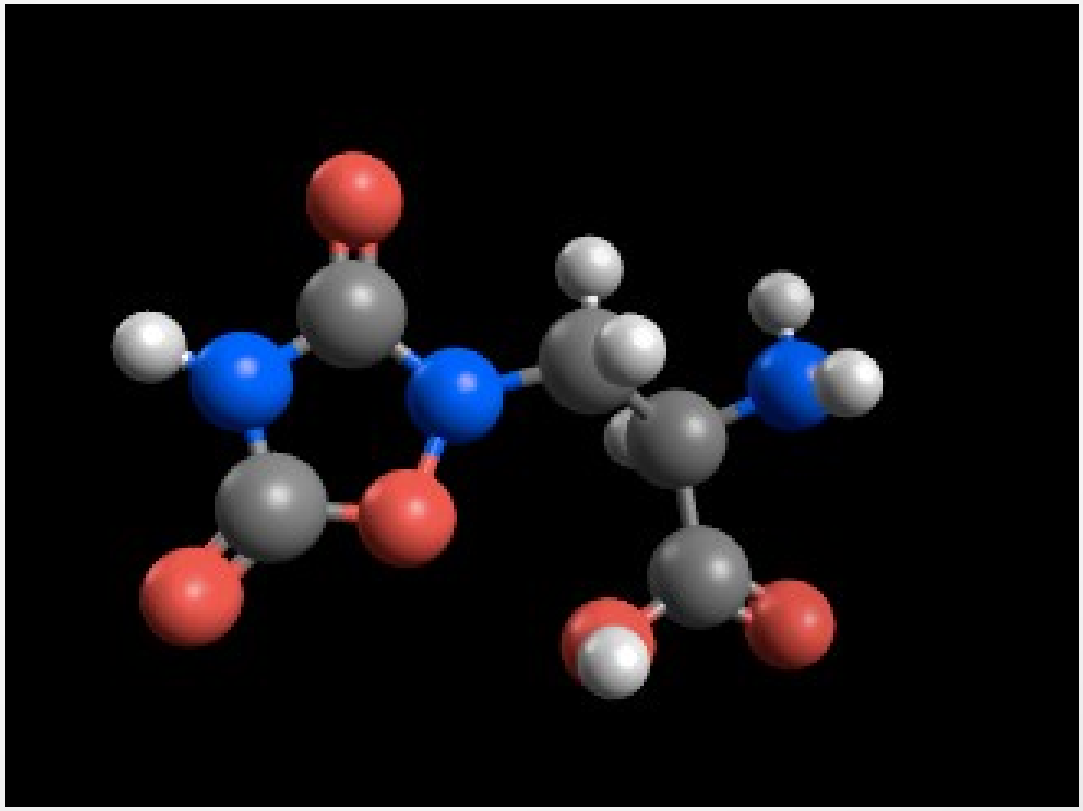
- **Terminazione**: L'algoritmo termina se la popolazione è a convergenza (**non produce figli che sono significativamente diversi dalla generazione precedente**). Quindi si dice che l'algoritmo genetico ha fornito una serie di soluzioni al nostro problema. Io non conosco la soluzione a monte.

```
START
Generate the initial population
Compute fitness
REPEAT
    Selection
    Crossover
    Mutation
    Compute fitness
UNTIL population has converged
STOP
```

ESERCITAZIONE

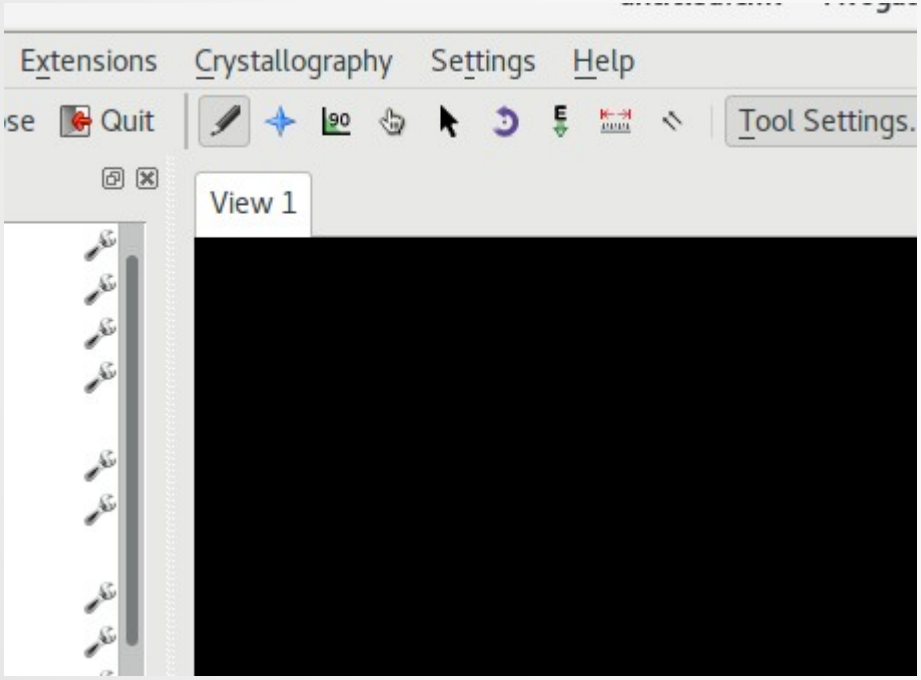
Generazione conformeri

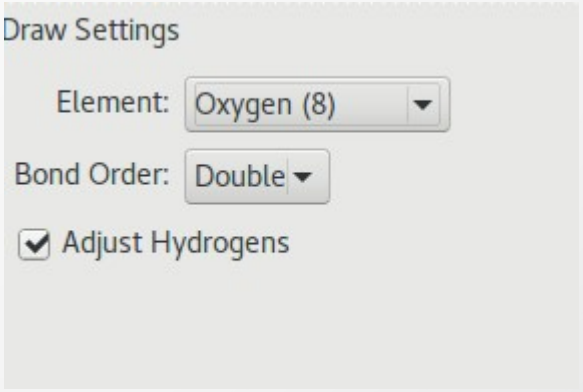
- Disegniamo quest struttura usando Avogadro:



Generazione conformeri

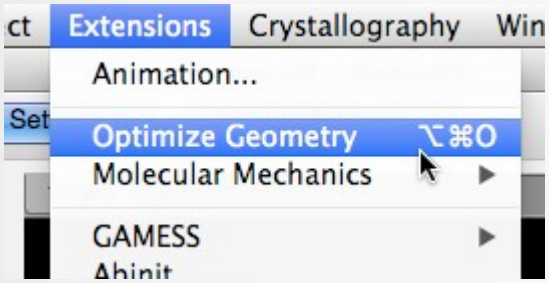
- Disegniamo questa struttura usando Avogadro:





Generazione conformeri

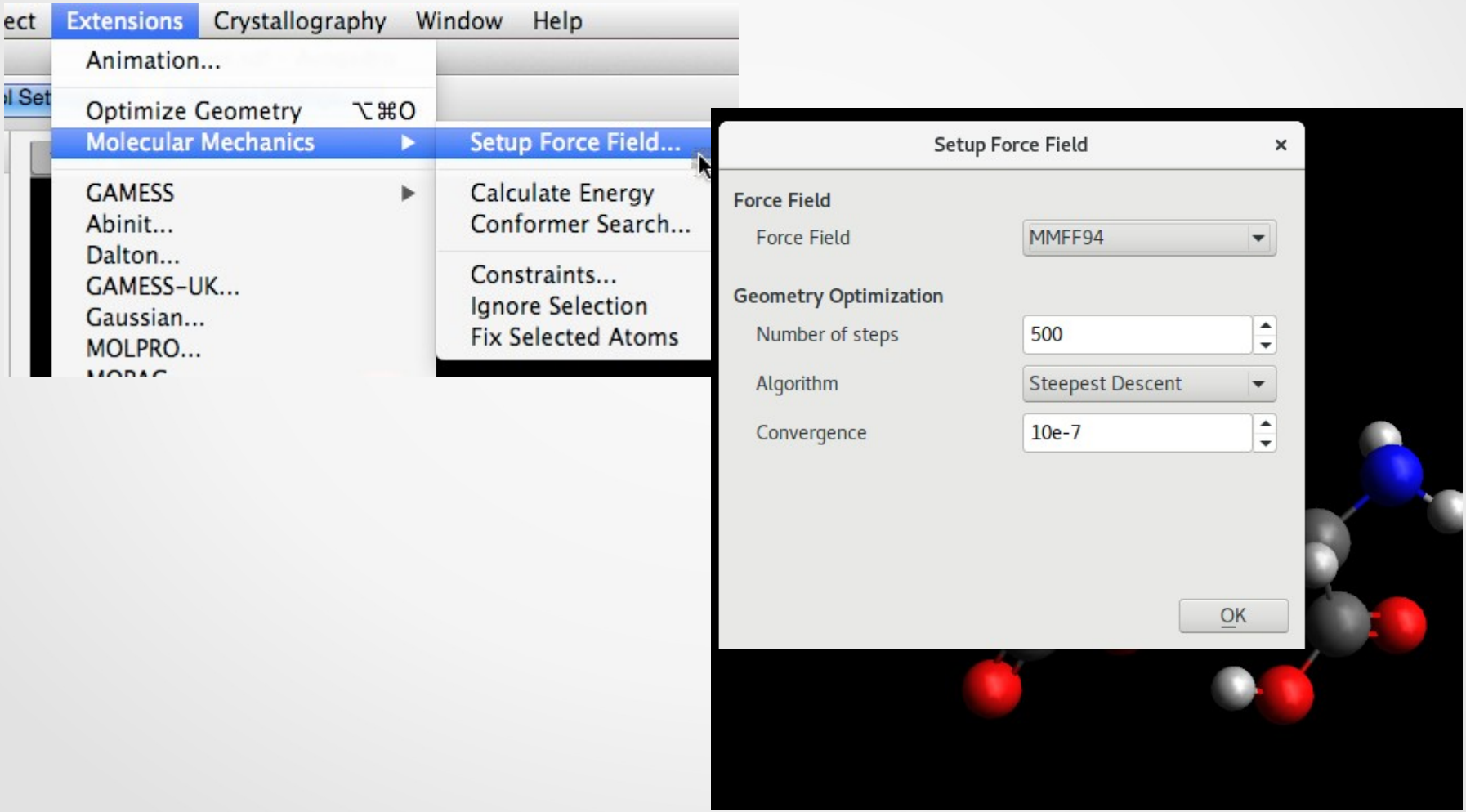
- Primo passaggio una minimizzazione usando un metodo MM.
- Primo passaggio una rapida “pulizia” della struttura:



- Poi una minimizzazione per essere certi di partire da una struttura “buona” nella ricerca di conformeri.

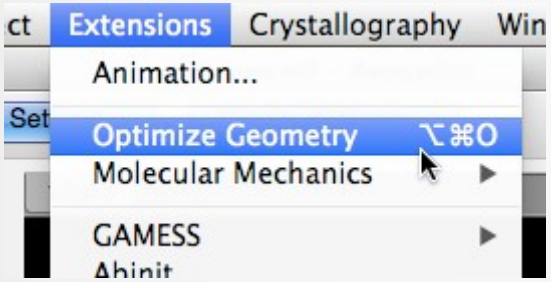
Generazione conformeri

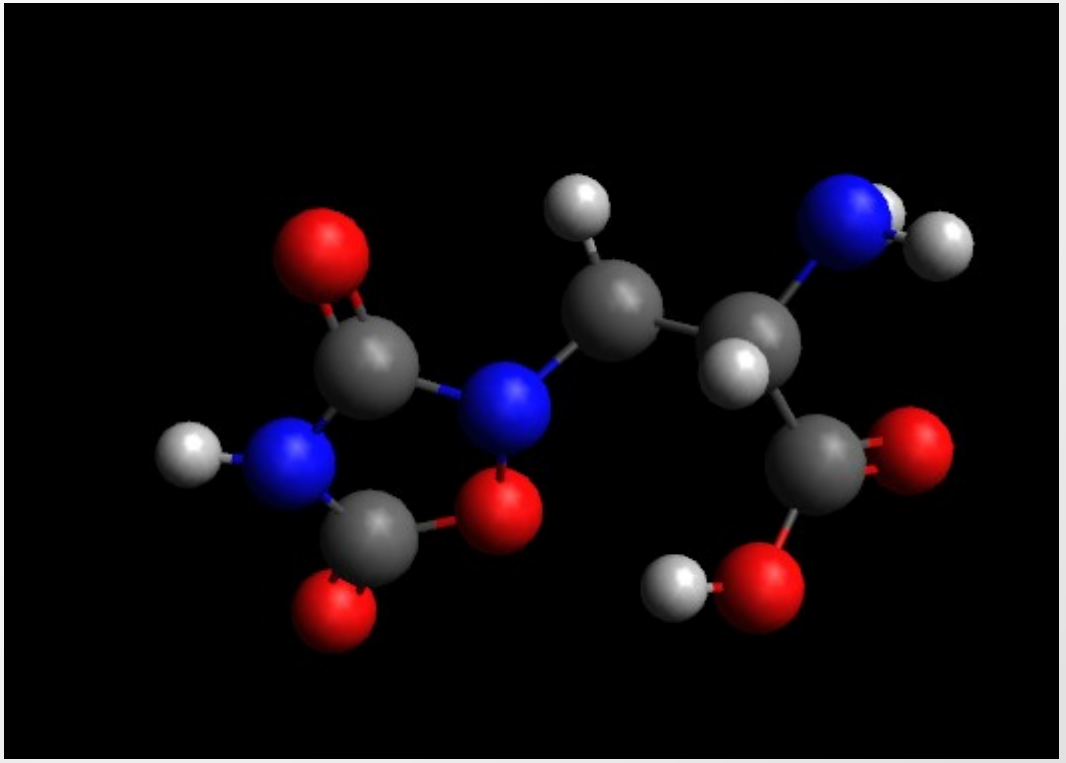
- Poi una minimizzazione MM, dopo il setup dell'FF



Generazione conformeri

- Poi volendo una minimizzazione MM, dopo il setup dell'FF e salviamo la struttura:





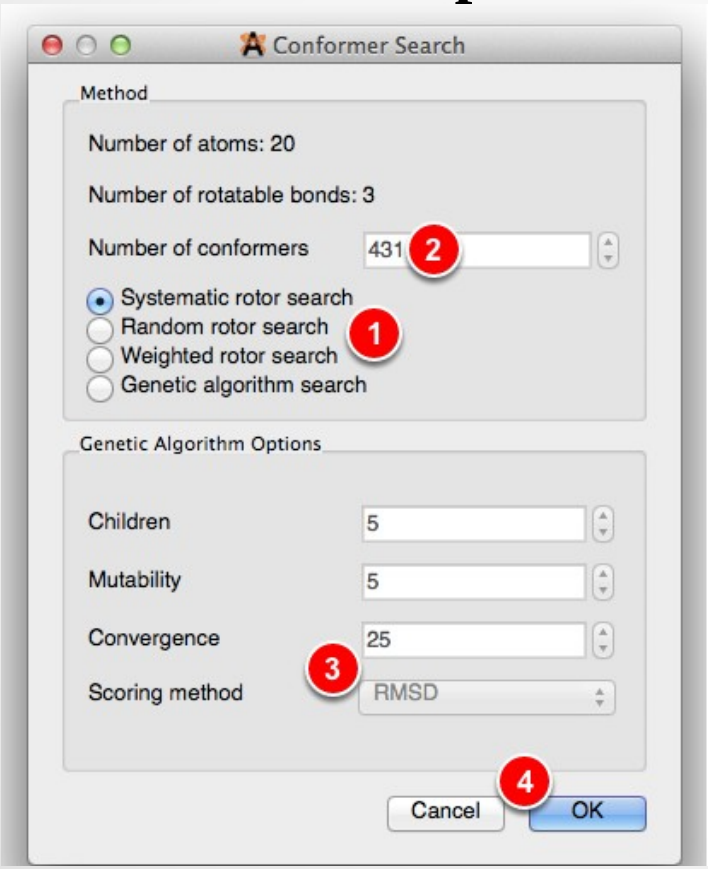
Generazione conformeri

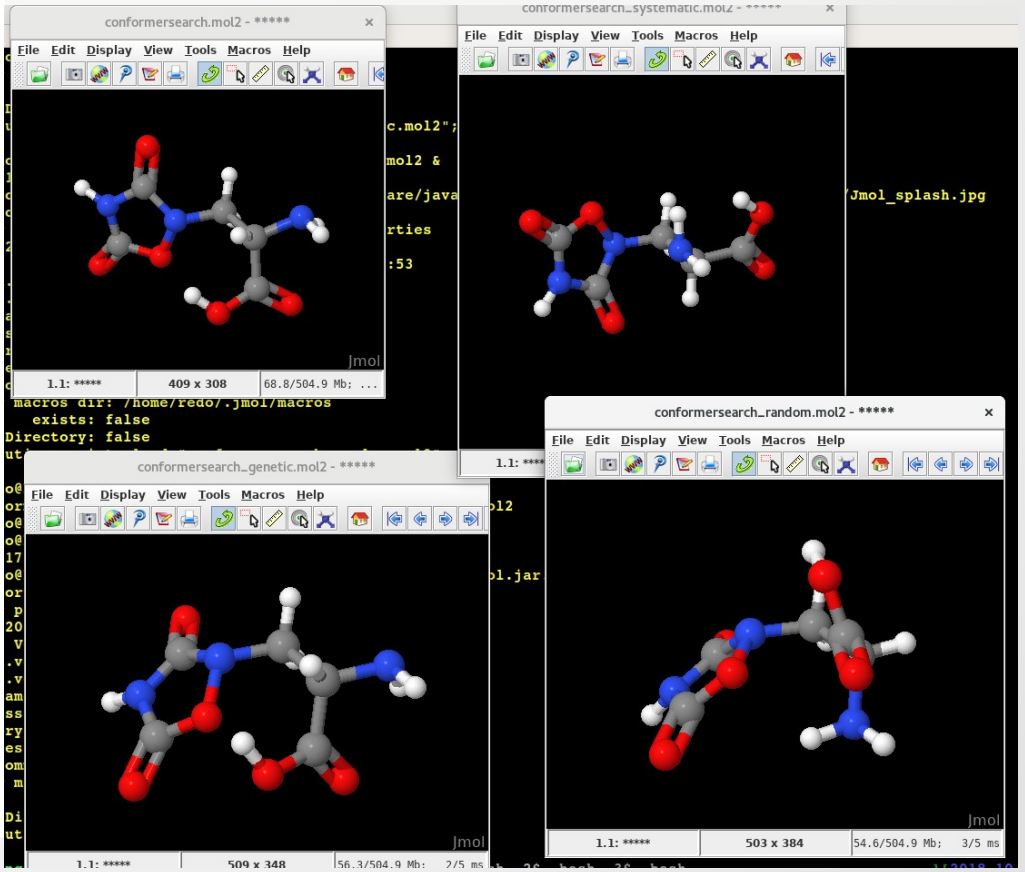
- Cambiamo i parametri del FF per gestire la minimizzazione dei conformeri

Numero di step: 20/30 tanto per essere un pochino piu' rapidi ma evitare Geometria non corrette (clash di atomi e legami)

Generazione conformeri

- Proviamo i tre metodi ed a seguire salviamo il minimo globale, **ogni volta dovremo ripartire dalla struttura originale**

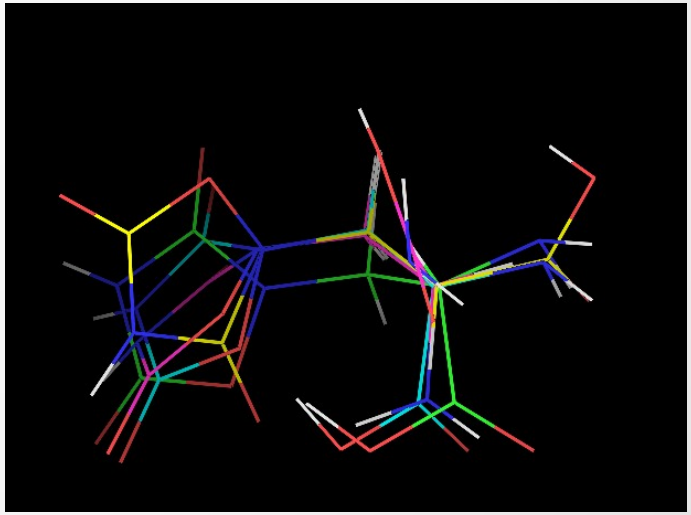




Generazione conformeri

- Calcoliamo l’energia dei 4 conformeri usando ad esempio MM da avogadro: Extensions → Molecular Mechanics → Calculate Energy
- E confrontiamo l’RMSD:

	Energy MMFF94	RMSD
starting	Energy = -139,521 kJ/mol	--
Systematic	Energy = -138,183 kJ/mol	2.154
Random	Energy = -96,4954 kJ/mol	1.543
Genetic	Energy = -41,3904 kJ/mol	0.761



==========

ESERCITAZIONE

Generazione conformeri

- **obabel**

  - `--systematic` - systematically (exhaustively) generate all conformers
  - `--random` - randomly generate conformers
  - `--weighted` - weighted rotor search for lowest energy conformer
  - `--ff <name>` - select a forcefield (default = MMFF94)

  Genetic algorithm based methods (default):

  - `--children #` - number of children to generate for each parent (default = 5)
  - `--mutability #` - mutation frequency (default = 5)
  - `--converge #` - number of identical generations before convergence is reached
  - `--score #` - scoring function [rmsd|energy] (default = rmsd)

Generazione conformeri

- Iniziamo con un algoritmo genetico:

  obabel conformersearch.mol2 -O obabel_GA_conformers.sdf --conformer --nconf 20 --score rmsd --writeconformers

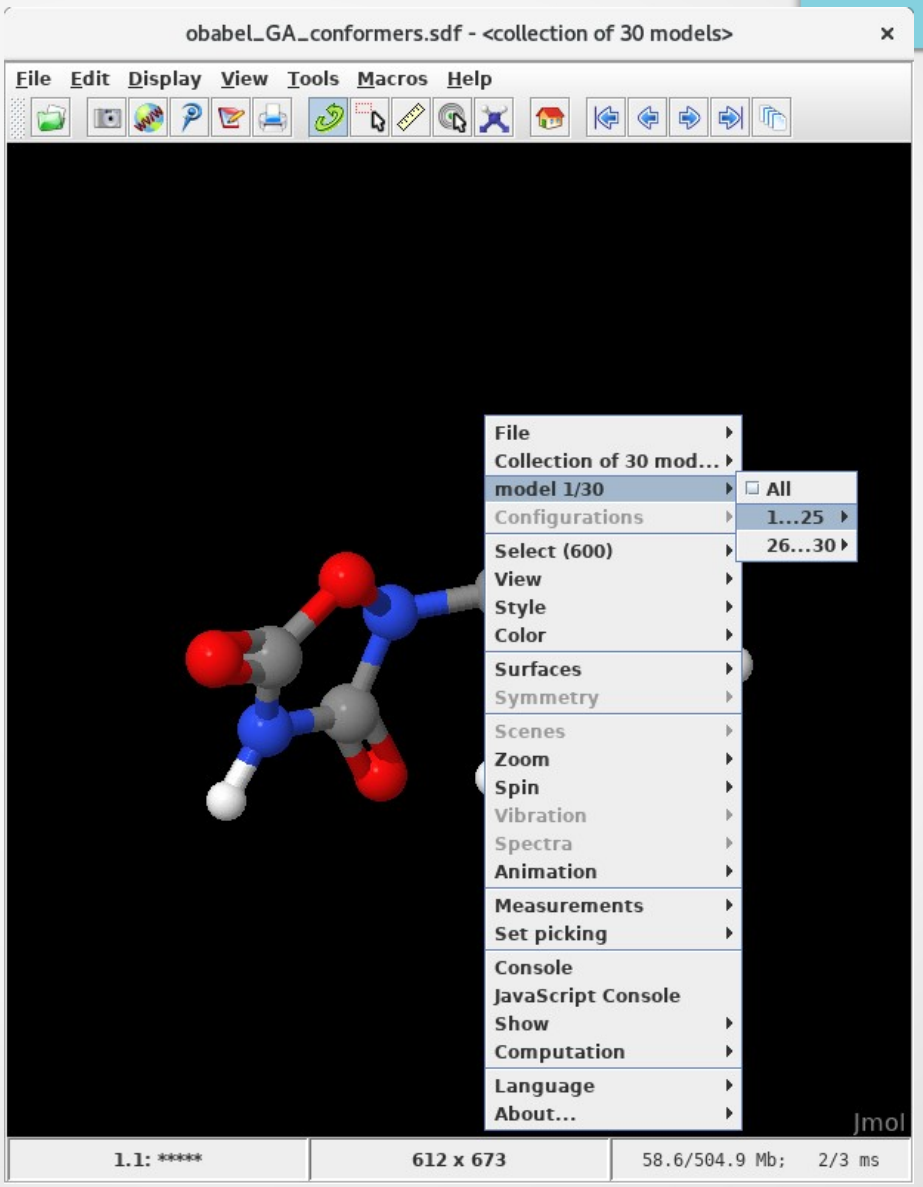
- Proviamo anche un search random volendo:

  obabel conformersearch.mol2 -O obabel_random_conformers.sdf --conformer --random --nconf 20 --log --writeconformers

- I files risultati saranno ovviamente un set di strutture. Con –log vi scrive anche quale conformero ha l’energia minore (solo con random e versione 2.4.X)

Generazione conformeri

- Con jmol potete ad esempio visionare i risultati



- Calcoliamo l’energia con metodo RHF (set a scelta dello studente) della struttura originale e del confomero che viene dato a piu’ bassa energia di **obabel_random_conformers.sdf**

  potrebbe esservi utile:

  babel -m -isdf obabel_random_conformers.sdf -omol2 obabel_random_conformers.mol2

==========