

# Self avoiding random walk applicati allo studio dei polimeri

Amurri Alessandro    Giardino Domenico

February 8, 2024

# Self avoiding random walk

## Definizione

Un self-avoiding random walk (SAW) di lunghezza  $N$  in un reticolo  $d$ -dimensionale  $\mathcal{L}$  è definito come una la sequenza  $\omega = (\omega_1 \dots \omega_N)$  tale che  $\omega_j \in \mathcal{L}$ ,  $\omega_0 = x$ ,  $x$  arbitrario,  $|\omega_j - \omega_{j-1}| = 1$ ,  $\forall j$  e  $\omega_i \neq \omega_j$  con  $0 \leq i < j \leq N$

# Self avoiding random walk

## Definizione

Un self-avoiding random walk (SAW) di lunghezza  $N$  in un reticolo  $d$ -dimensionale  $\mathcal{L}$  è definito come una la sequenza  $\omega = (\omega_1 \dots \omega_N)$  tale che  $\omega_j \in \mathcal{L}$ ,  $\omega_0 = x$ ,  $x$  arbitrario,  $|\omega_j - \omega_{j-1}| = 1$ ,  $\forall j$  e  $\omega_i \neq \omega_j$  con  $0 \leq i < j \leq N$

In altre parole un SAW è una sequenza di passi che non torna mai nello stesso punto.

# Parametri analizzati

- 1 Distanza testa-coda  $\langle R^2(N) \rangle \sim N^{2\nu}$

# Parametri analizzati

- 1 Distanza testa-coda  $\langle R^2(N) \rangle \sim N^{2\nu}$
- 2 Raggio d'inerzia  $\langle R_g(N) \rangle \sim N^\nu$

$$R_g = \sqrt{\frac{1}{N} \sum_i |r_i - r_{mean}|^2}$$

# Parametri analizzati

- 1 Distanza testa-coda  $\langle R^2(N) \rangle \sim N^{2\nu}$
- 2 Raggio d'inerzia  $\langle R_g(N) \rangle \sim N^\nu$

$$R_g = \sqrt{\frac{1}{N} \sum_i |r_i - r_{mean}|^2}$$

- 3 Distanza di intrappolamento  $\langle T_l(N, \epsilon) \rangle \sim N^{1.12\epsilon}$

# Algoritmi realizzati

*Beretti-Sokal* [2]

Questo algoritmo è un Monte Carlo dinamico, ovvero una catena di Markov. Il suo scopo è generare un SAW, con un'estremità fissata, nell'ensemble gran canonico con attività del monomero  $\beta$ .

L'algoritmo è basato su tre mosse elementari:

- 1 Mossa positiva
- 2 Mossa "nulla"
- 3 Mossa negativa

# Mosse elementari

## Probabilità

Le probabilità di tentare una mossa positiva o una mossa negativa sono:

$$\begin{cases} P(\Delta N = +1) = \frac{q\beta}{1+q\beta} \\ P(\Delta N = -1) = \frac{1}{1+q\beta} \end{cases}$$

# Mosse elementari

## Probabilità

Le probabilità di tentare una mossa positiva o una mossa negativa sono:

$$\begin{cases} P(\Delta N = +1) = \frac{q\beta}{1+q\beta} \\ P(\Delta N = -1) = \frac{1}{1+q\beta} \end{cases}$$

Ne segue che la probabilità di transizione da un walk  $\omega$  a un walk  $\omega'$  è data da:

$$p(\omega \rightarrow \omega') = \begin{cases} \frac{\beta}{1+q\beta} \chi(\omega') & \text{se } |\omega| < |\omega'| \\ \frac{1}{1+q\beta} & \text{se } |\omega| > |\omega'| \\ \frac{\beta}{1+q\beta} A(\omega) & \text{se } |\omega| = |\omega'| \neq 0 \end{cases}$$

# Bilancio dettagliato

Si può verificare che queste probabilità verificano il bilancio dettagliato:

$$\pi(\omega)p(\omega \rightarrow \omega') = \pi(\omega')p(\omega' \rightarrow \omega)$$

Dove

$$\pi(\omega) = \frac{\beta^{|\omega|}}{\Theta(\omega)} \chi(\omega) \quad \text{con} \quad \Theta(\omega) = \sum_{i=1}^{\infty} \beta^i C_N$$

# Catena di Markov

Il fatto che il bilancio dettagliato sia verificato assicura che la distribuzione di probabilità  $\pi(\omega)$  sia una distribuzione stazionaria per la catena di Markov:

$$\pi(\omega) = \frac{\beta^{|\omega|}}{\Theta(\omega)} \chi(\omega)$$

L'unicità di tale distribuzione è verificata dall'ergodicità del processo.

# Pseudocodice

## Pseudocodice dell'algoritmo Beretti-Sokal realizzando in linguaggio Matlab:

---

Inizializzazione delle variabili

Per ogni passo  $i$  da 1 a numero passi

  scelta della mossa

    se mossa=mossa positiva

      scelta di una possibile destinazione

      se la destinazione è accettabile

        aggiorna SAW

    altrimenti

      mossa nulla

  altrimenti mossa=mossa negativa

    se  $i$  maggiore di 1

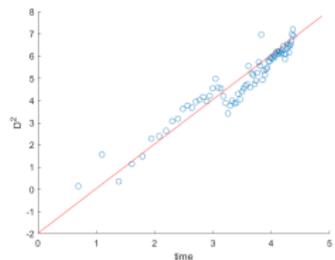
      cancella passo precedente

  aggiorna bit\_bap

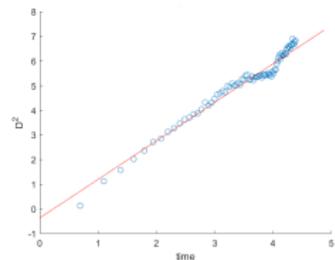
Output variabili

---

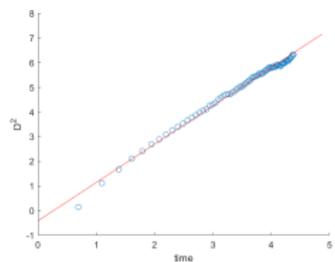
# Ruolo del numero delle run



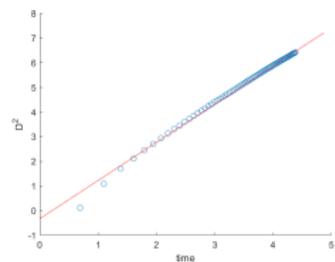
(a)  $N=1$



(a)  $N=10$



(b)  $N=100$



(b)  $N=10000$

## Media della distanza al quadrato, $\langle R^2 \rangle$

Il valore del parametro  $\nu$  può essere ricavato da

$$\langle R(N)^2 \rangle \sim N^{2\nu} \rightarrow \log(\langle R(N)^2 \rangle) = 2\nu \log(N) + \text{cost}$$

Valore ottenuto:

$$2\nu = 1.4985 \pm 0.0009$$

In accordo con quanto atteso per un reticolo bidimensionale.

# Algoritmi realizzati

## *GSAW algorithm* [3]

A ogni passo l'algoritmo sceglie in modo uniforme tra tutti i vicini non visitati precedentemente.

- A differenza del precedente questo algoritmo può rimanere intrappolato

# Algoritmi realizzati

## GSAW algorithm [3]

A ogni passo l'algoritmo sceglie in modo uniforme tra tutti i vicini non visitati precedentemente.

- A differenza del precedente questo algoritmo può rimanere intrappolato
- I self avoiding walk che restituisce non sono uniformemente distribuiti nell'insieme dei self avoiding walk di lunghezza  $n$  ( $SAW(n)$ ).



**Figure:** Il SAW a sinistra avrà probabilità  $\frac{1}{4} \cdot \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{1}{2}$  mentre quello a destra avrà una probabilità  $\frac{1}{4} \cdot \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{1}{3}$

# Importance sampling

Il problema può essere risolto se si pesa ogni walk ( $X_i$ ) con la sua probabilità di campionamento ( $w_i = \frac{1}{g(X_i)}$ ).

In questo modo si potrà, per esempio, calcolare il **valore di aspettazione** di una funzione sull'insieme  $U(SAW(n))$ .

$$\hat{R}^2 = \frac{\sum w_i R^2(X_i)}{\sum w_i}$$

Tuttavia si avrà un **bias** inevitabile dovuto al numero limitato di campioni

# $R^2$ calcolato in due modi

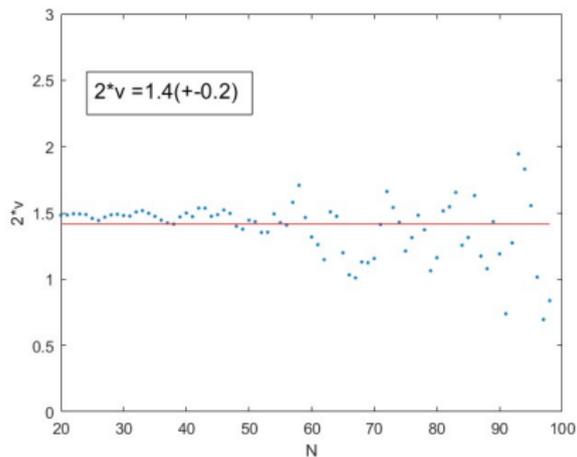


Figure: Approssimazione al primo ordine

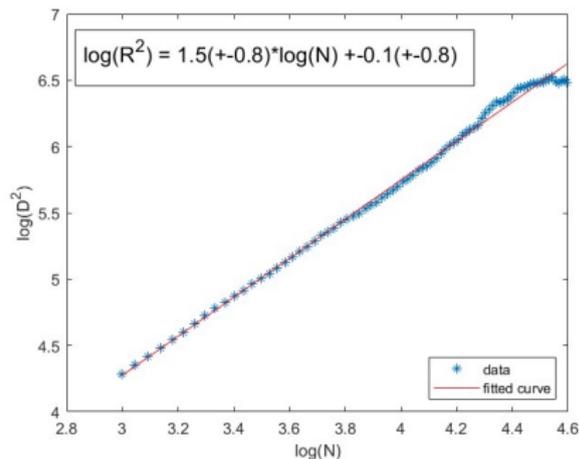


Figure: Regressione lineare

# Polimeri

Un polimero è un materiale costituito da una catena di macromolecole (**monomeri**).

Come descriviamo la struttura di un polimero amorfo?

Abbiamo bisogno di una media statistica sulle varie conformazioni possibili della catena → **random walk!**

# Polimeri

Un polimero è un materiale costituito da una catena di macromolecole (**monomeri**).

Come descriviamo la struttura di un polimero amorfo?

Abbiamo bisogno di una media statistica sulle varie conformazioni possibili della catena → **random walk!**

$$R^2(\hat{n}) \propto N^{\frac{3}{2}}$$

$$Rg(\hat{n}) \propto N^{\frac{3}{4}}$$

# Facciamo crescere un polimero

*Trapping in Self-Avoiding Walks with Nearest-Neighbor Attraction [4]*

Nel simulare il processo di crescita del polimero consideriamo **l'attrazione esercitata dai vicini più prossimi** ( $m_i$ ).

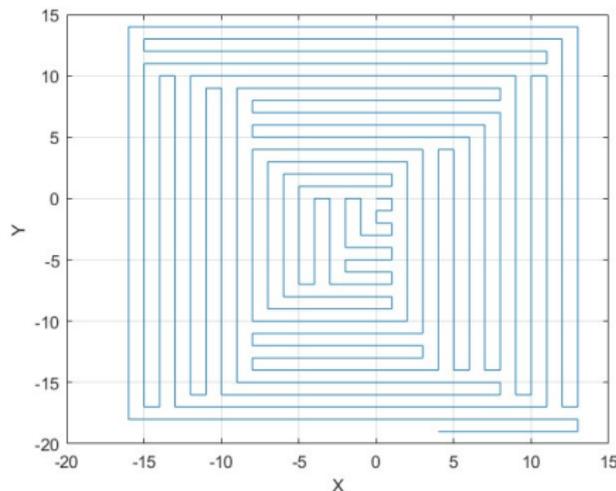


Figure: GSAW con energia di interazione alta

$$p_i = \frac{1}{Z} e^{\epsilon m_i}$$

# Facciamo crescere un polimero

*Trapping in Self-Avoiding Walks with Nearest-Neighbor Attraction [4]*

Nel simulare il processo di crescita del polimero consideriamo **l'attrazione esercitata dai vicini** più prossimi ( $m_i$ ).

Variando l'energia di attrazione si ricavano informazioni sul processo, tra cui per esempio la distanza media di intrappolamento ( $T_I(\epsilon)$ )

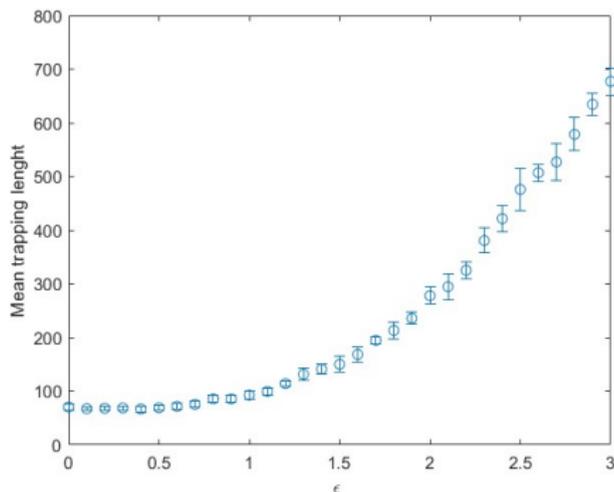


Figure: Distanza di intrappolamento media

$$p_i = \frac{1}{Z} e^{\epsilon m_i}$$

## La figura mostra il minimo di $T_I(\epsilon)$

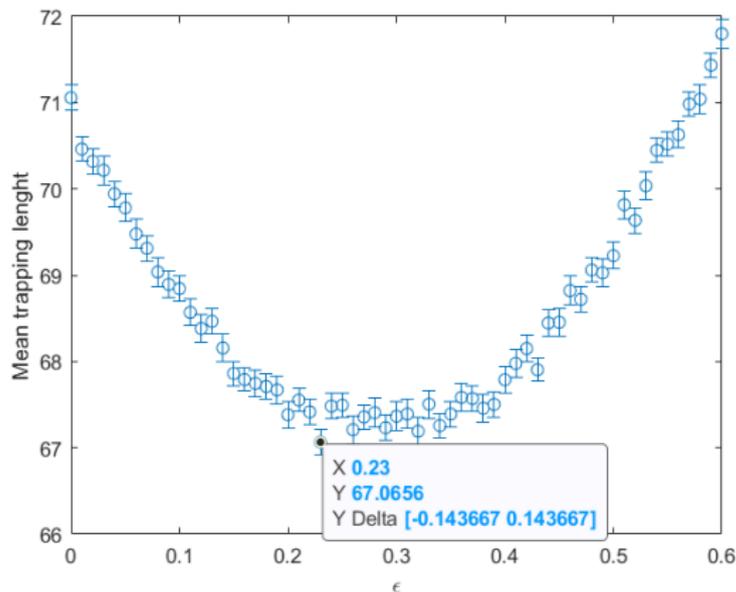


Figure:  $T_I(\epsilon)_{min} = 67$  con  $\epsilon = 0.25$

[Quando l'attrazione reciproca è forte, è più difficile si crei un vuoto in cui il walk può rimanere intrappolato, mentre se è debole non c'è nessun bias per cui debba finirci dentro]

# Gsaw interagenti

## Definizione

Un "buon solvente" è tale se l'interazione di *volume escluso* tra parti diverse della stessa catena fa assumere al polimero una **configurazione sparsa**.

Un "cattivo solvente" è tale se dominano le *attrazioni reciproche* fra i monomeri, in tal caso il polimero assume una **forma a spirale**

### theta point

Il passaggio dalla configurazione sparsa a quella globulare avviene in quello che è chiamato "theta solvent".

# Rg al variare della temperatura

Theta point ( $\theta_p$ ) diverso da un semplice SAW

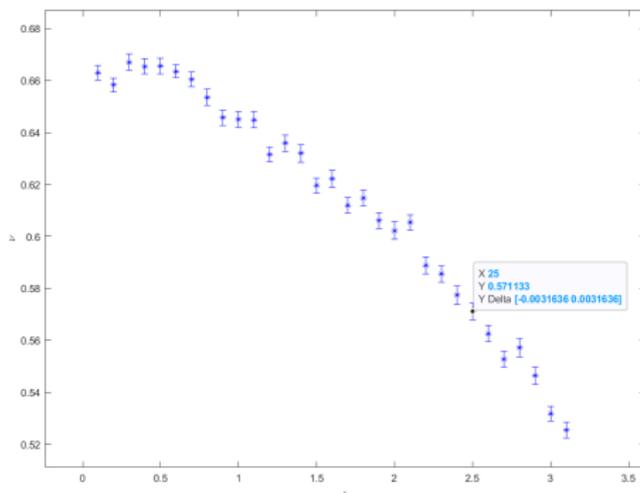


Figure: GSAW: ( $\theta_p = 2.5$ )?

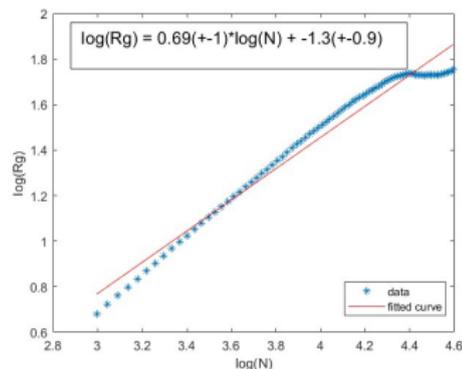


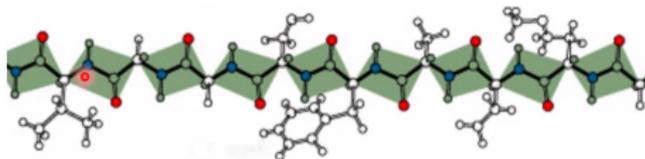
Figure:  $\nu$  a  $\epsilon = 0.66$

# Proteine

Come sono fatte?

## Definizione

Le proteine sono dei polimeri di amminoacidi legati tra loro da legami peptidici.

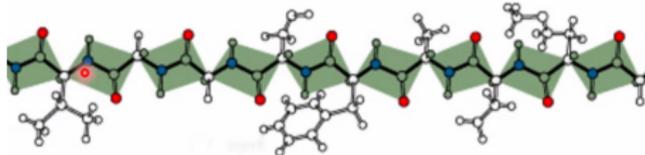


# Proteine

Come sono fatte?

## Definizione

Le proteine sono dei polimeri di amminoacidi legati tra loro da legami peptidici.

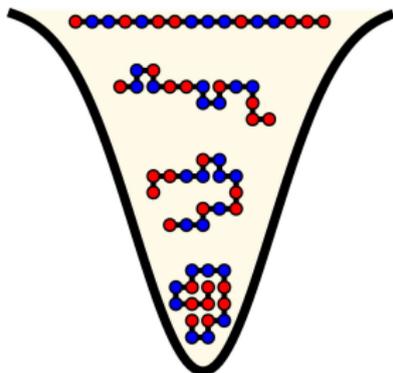


Gli amminoacidi si dividono in quattro gruppi:

- Acidi
- Basici
- Polari neutri
- Idrofobici

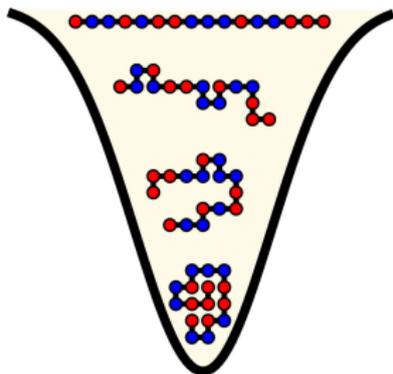
# Folding di una proteina

Una proteina nasce come un polimero disordinato per poi evolversi verso una struttura più ordinata.



## Folding di una proteina

Una proteina nasce come un polimero disordinato per poi evolversi verso una struttura più ordinata.



Questo processo è energeticamente favorevole: la creazione di interazioni tra gli amminoacidi diminuisce l'energia interna della proteina.

# Energie di interazione

Monte Carlo for Protein Structures[5]

Ci sono diversi tipi di **interazioni** che si instaurano, ognuna con un'energia propria. Queste sono rappresentate secondo l'ordine di grandezza relativo:

|            | Acido | Basico | Polare | Idrofobico |
|------------|-------|--------|--------|------------|
| Acido      | ++    | --     | -      |            |
| Basico     | --    | ++     | -      |            |
| Polare     | -     | -      | -      |            |
| Idrofobico |       |        |        | -          |

# Metropolis

A cosa serve?

## Problema n.1

Generare dei campioni  $x_i$  da una distribuzione di probabilità  $P(x)$

## Problema n.2

Siamo interessati al valore di una funzione che segua questa distribuzione di probabilità:

$$\langle \phi(x) \rangle \equiv \int d^N x P(x) \phi(x)$$

In particolare la distribuzione che vogliamo campionare è :

$$P(r) = \frac{1}{Z} e^{-\beta f(r)}$$

# Metropolis

## Come funziona?

Il metodo si basa sulla generazione di valori "proposti" che vengono accettati o rigettati. Si procede in questo modo:

- Si estrae un campione  $x'$ , secondo la probabilità  $g(x'|x)$

# Metropolis

## Come funziona?

Il metodo si basa sulla generazione di valori "proposti" che vengono accettati o rigettati. Si procede in questo modo:

- Si estrae un campione  $x'$ , secondo la probabilità  $g(x'|x)$
- Questo campione viene accettato con probabilità

$$\omega = \min \left\{ 1, \frac{p(x')}{p(x)} \right\}$$

# Metropolis

## Come funziona?

Il metodo si basa sulla generazione di valori "proposti" che vengono accettati o rigettati. Si procede in questo modo:

- Si estrae un campione  $x'$ , secondo la probabilità  $g(x'|x)$
- Questo campione viene accettato con probabilità

$$\omega = \min \left\{ 1, \frac{p(x')}{p(x)} \right\}$$

- In caso venga rifiutato si campiona di nuovo il campione precedente

# Parametro di idrofobicità

Gli amminoacidi idrofobici formano un "core"?

Per stimare quanto gli amminoacidi idrofobici ( $r_i^{idro}$ ) siano vicini al centro di massa della molecola definiamo:

$$I \equiv \frac{N}{\sum |r_i^{idro} - r_{mean}|^2}$$

# Sampling del landscape energetico

Campionamento di 100000 SAW casuali [1]

|               | <b>E</b>        | <b>I</b>        |
|---------------|-----------------|-----------------|
| <b>T=0</b>    | -9700           | 0.35            |
| <b>T=500</b>  | $-7200 \pm 100$ | $0.27 \pm 0.02$ |
| <b>T=1000</b> | $-4600 \pm 100$ | $0.25 \pm 0.04$ |

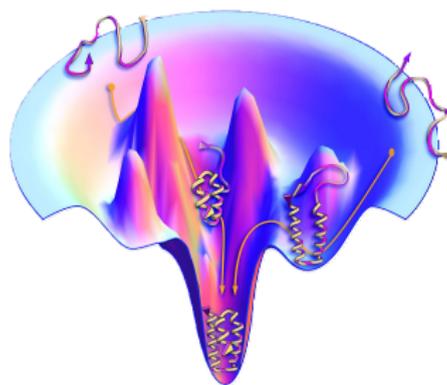


Figure: Landscape energetico

# Plot configurazioni raggiunte

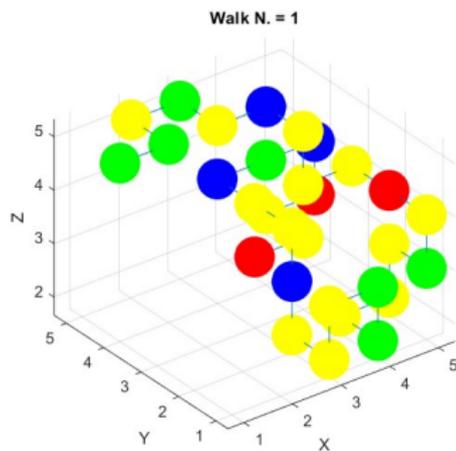


Figure: Plot configurazione iniziale

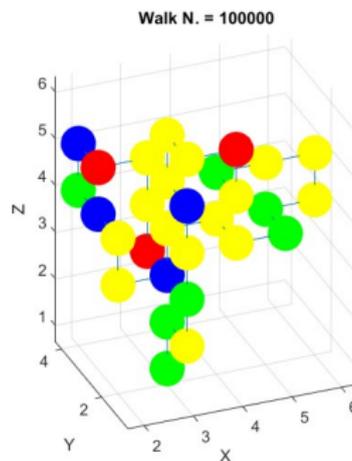


Figure: Plot configurazione finale

# Bibliografia I

-  Alessandro Amurri Domenico Giardino.  
Random walk proteine, 2023.  
[https://github.com/dodo13235/Proteine\\_randomwalk.git](https://github.com/dodo13235/Proteine_randomwalk.git).
-  Alberto Berretti and Alan D Sokal.  
Critical behavior of the self-avoiding random walk: a numerical study.  
*Journal of Statistical Physics*, 1 August 1985.
-  Gabriel Romon.  
Monte carlo project at ensae, spring 2018, 2018.  
<https://github.com/gabsens/SelfAvoidingWalk>.
-  Wyatt Hooper and Alexander R. Klotz.  
Trapping in self-avoiding walks with nearest-neighbor attraction.  
*Physical Review Letters*, E102(3), 2020.



Juami H. M. van Gils, Maurits Dijkstra, Halima Mouhib, Arriën Symon Rauh, Jocelyne Vreede, K. Anton Feenstra, and Sanne Abeln.

Monte carlo for protein structures.

*Journal of Computational Chemistry*, Capitolo 15.