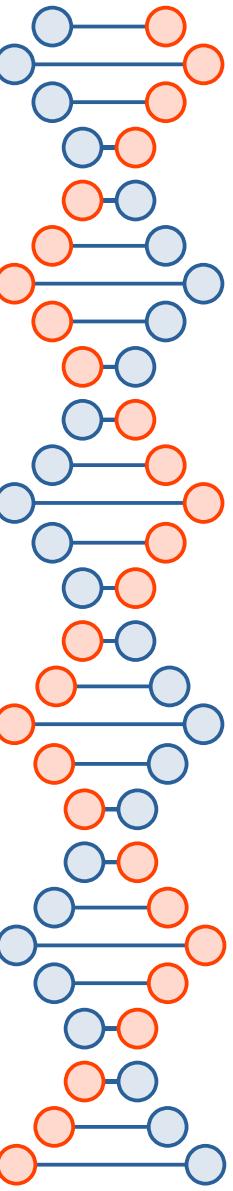


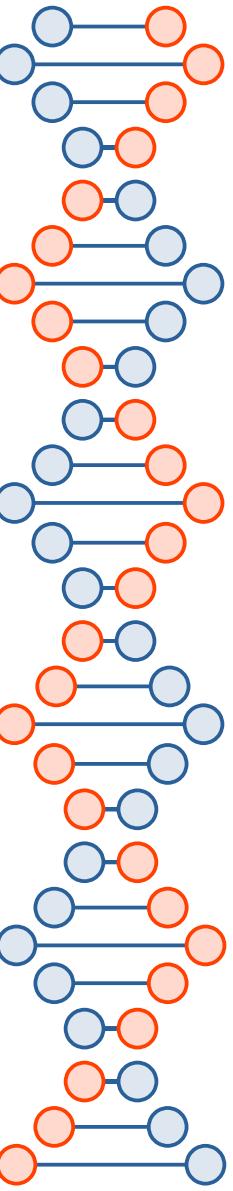
# Coleta e Preparação das Bases de Grafos

Atílio G. Luiz



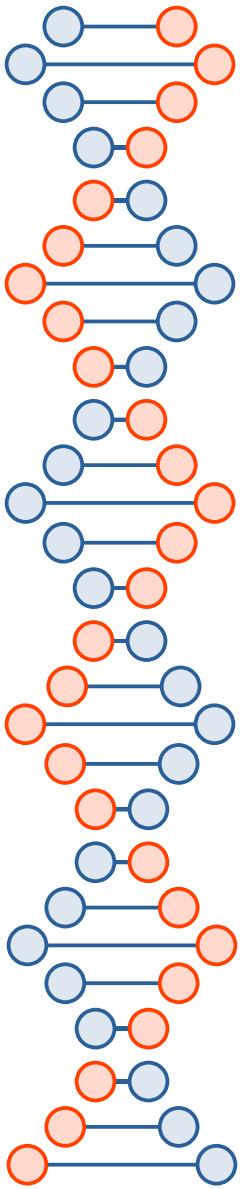
# Coleta de Grafos

- Meta-heurísticas em grafos são comumente testadas em uma base de tamanho considerável (300 ~ 400 grafos)
- Os grafos podem ser gerados aleatoriamente usando, por exemplo, o modelo de Erdös-Rényi.
- Podemos também usar grafos advindos de bases já existentes e comumente usadas na literatura.
- Às vezes temos que usar dados coletados do mundo real ou gerar exemplos desafiadores.



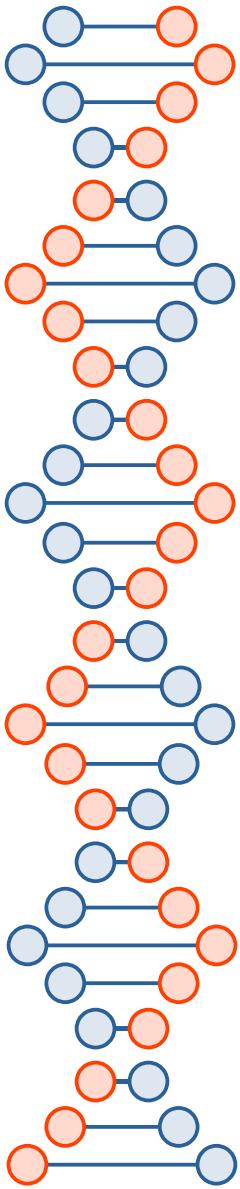
# Base DIMACS

- Criada nos anos 90 para uma série de competições científicas organizadas pelo *Center for Discrete Mathematics and Theoretical Computer Science* (DIMACS).
- As competições envolviam resolver problemas de: coloração de grafos, caixeiro viajante, fluxo máximo, caminho mínimo, dominação, etc.
- Os grafos estão armazenados em arquivos de texto simples, com um formato definido (o formato DIMACS).



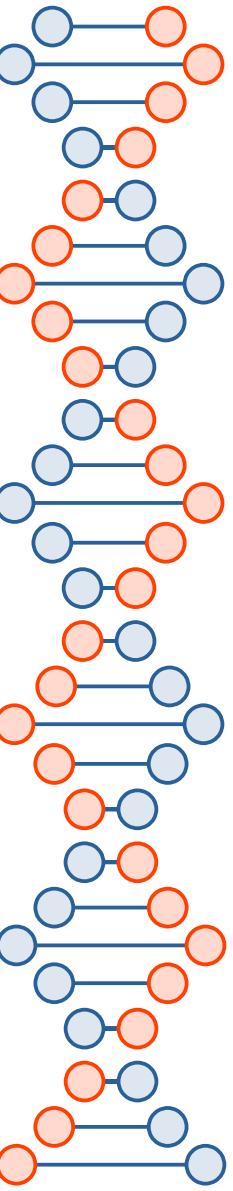
# Estrutura e formato dos arquivos DIMACS

- Os grafos são descritos em formato textual padronizado, com linhas que começam com letras-chave:
  - c Este é um comentário
  - p edge 5 7
  - e 1 2
  - e 1 3
  - e 2 3
  - e 2 4
  - e 3 5
  - e 4 5
  - e 1 5



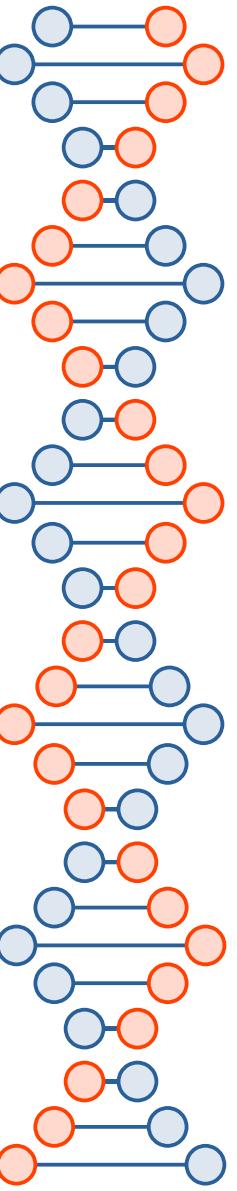
# Onde encontrar grafos da base DIMACS

- <http://archive.dimacs.rutgers.edu/pub/challenge/graph/>
- <http://archive.dimacs.rutgers.edu/pub/challenge/graph/benchmarks/color/>
- <http://archive.dimacs.rutgers.edu/pub/challenge/graph/benchmarks/clique/>
- <https://mat.tepper.cmu.edu/COLOR/instances.html>
- <https://networkrepository.com/dimacs.php>
- [https://iridia.ulb.ac.be/~fmascia/maximum\\_clique/](https://iridia.ulb.ac.be/~fmascia/maximum_clique/)
- <https://cedric.cnam.fr/~porumbed/graphs/>



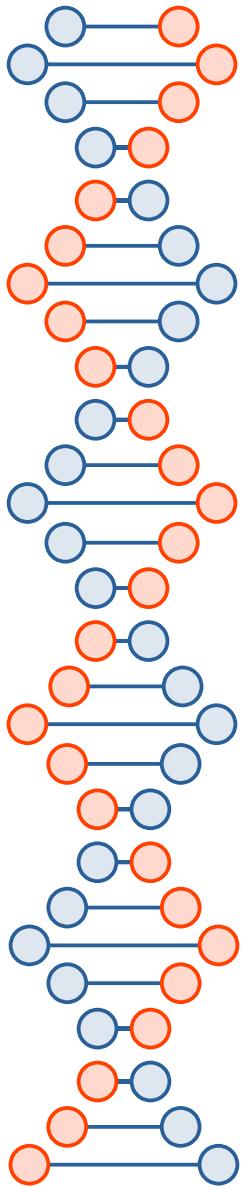
# Coleção de Matrizes Esparsas

- Matrizes esparsas podem ser vistas como grafos, desde que sejam matrizes quadradas e simétricas.
- Harwell–Boeing (HB) é uma coleção clássica de matrizes esparsas usada como referência para testar algoritmos numéricos e bibliotecas de álgebra linear.
- Essas matrizes não representam grafos diretamente, mas sistemas lineares esparsos que podem ser interpretados como grafos (por exemplo, a matriz de adjacência de uma malha, ou a matriz de rigidez de um sistema físico).



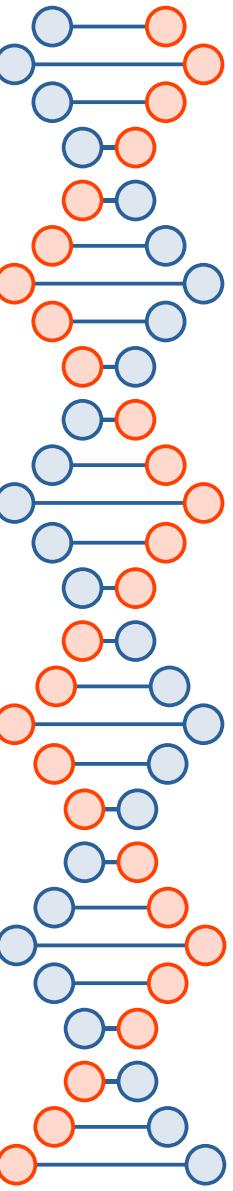
# Harwell-Boeing

- A coleção foi criada pelos pesquisadores Iain S. Duff, Roger G. Grimes, John G. Lewis
- Esses cientistas trabalhavam nos Harwell Laboratory (Reino Unido) e Boeing Computer Services (EUA) — daí o nome Harwell–Boeing.



# Onde encontrar a base Harwell-Boeing?

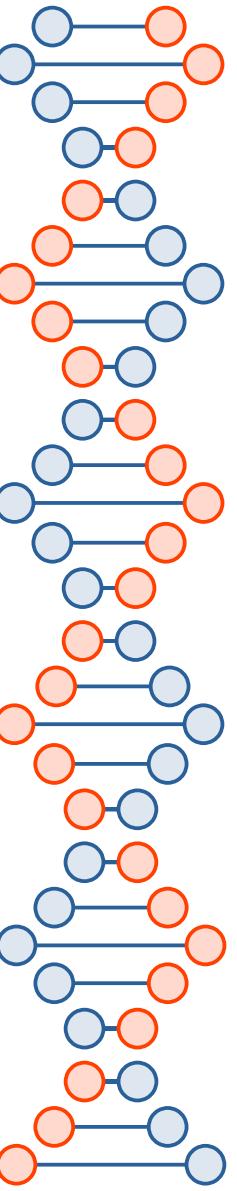
- Matrix Market (NIST):  
<https://math.nist.gov/MatrixMarket/matrices.html>
- Suite Sparse Matrix Collection: <https://sparse.tamu.edu/>
  - Versão ampliada e atualizada, mantida por Timothy A. Davis, com milhares de matrizes (incluindo as HB originais e muitas outras).



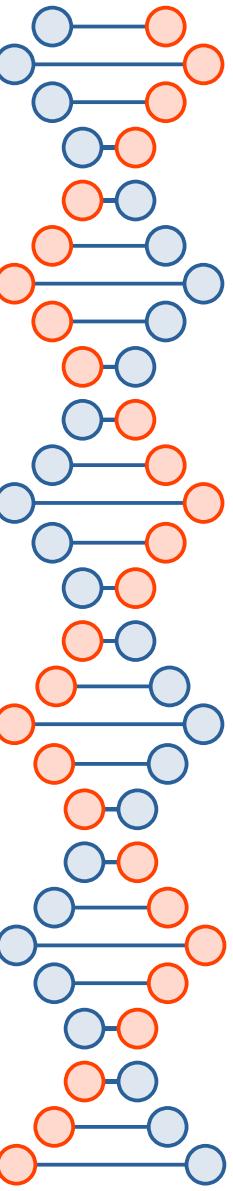
# HB: Exemplo de uso prático

- Em Python, com **scipy.io.mmread**, você pode carregar uma matriz HB. Depois, usando NetworkX você pode convertê-la em grafo (se ela for simétrica, claro).

```
1 import networkx as nx
2 from scipy.io import mmread
3
4 A = mmread("bcsstk01.mtx").tocsr()
5 G = nx.from_scipy_sparse_array(A)
```

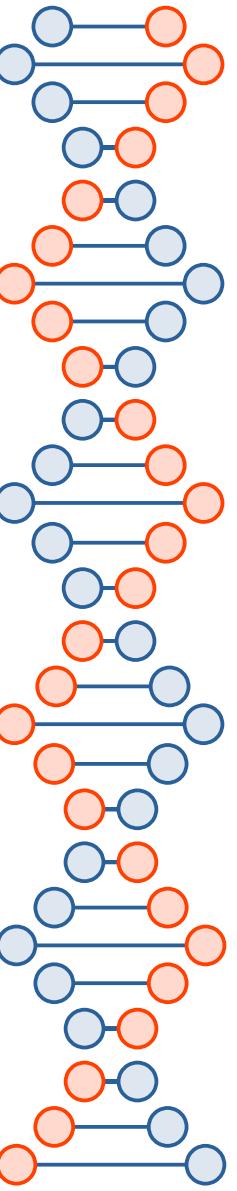


# Graph Cleaning (Limpeza dos Grafos)



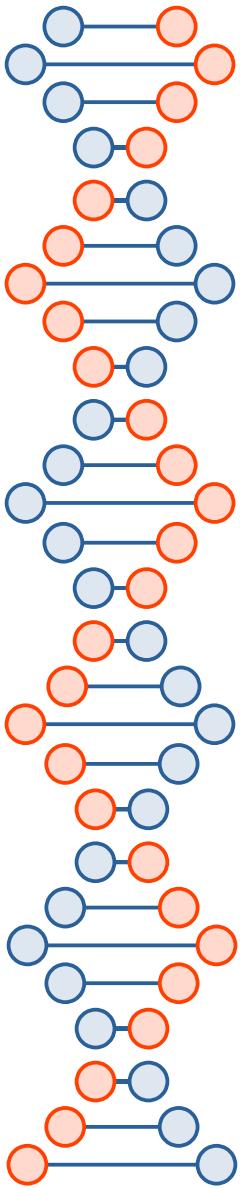
# Graph Cleaning

- Na maioria das aplicações, é preciso tratar os grafos antes de usá-los.
- Se for o caso, a matriz que se pretende utilizar para obter o grafo é quadrada e é simétrica?
- O grafo possui laços e arestas múltiplas?
- Os vértices do grafo estão numerados de 0 a  $n-1$ ?
- Os vértices do grafo são inteiros consecutivos ou não são consecutivos?



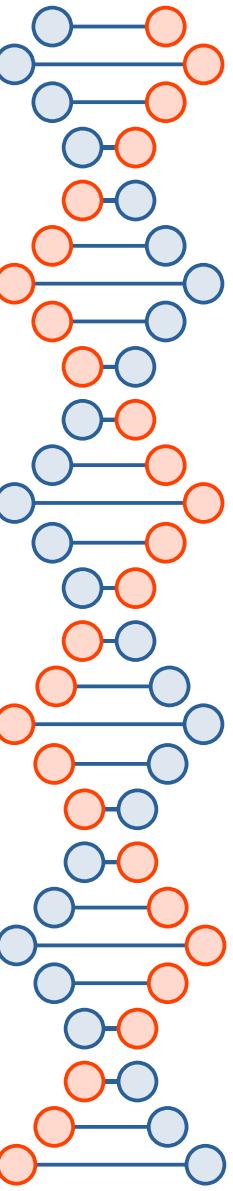
# Graph Cleaning

- Após baixar todos os grafos e fazer o pré-processamento (limpeza) de toda a base, é comum salvar os grafos em um formato já adequado para ser processado pelo seu programa.



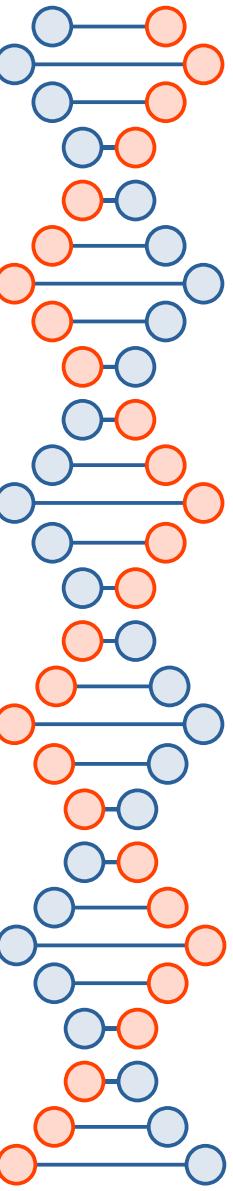
# Como encontrar outras bases de grafos?

- Pergunte ao google, chat gpt, gemini, etc.
- Se você está estudando um problema específico de otimização em grafos, procure por artigos recentes e antigos que tratem o seu problema pelo viés experimental. Esses artigos com certeza usam alguma base ou foram pioneiros na geração de uma base para o problema.



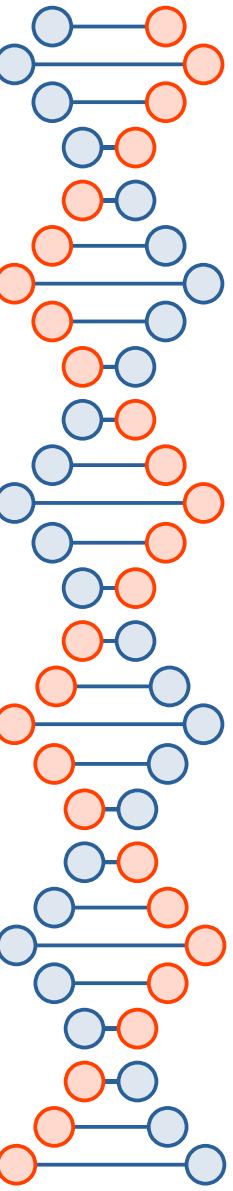
## Outras bases

- Problemas de atribuição de canais de frequência:  
[https://fap.zib.de/problems/?utm\\_source=chatgpt.com](https://fap.zib.de/problems/?utm_source=chatgpt.com)
- Caixeiro Viajante (tsplib):  
<http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>
- Network Repository:  
[https://networkrepository.com/?utm\\_source=chatgpt.com](https://networkrepository.com/?utm_source=chatgpt.com)
- Suite Sparse Matrix:  
[https://sparse.tamu.edu/?utm\\_source=chatgpt.com](https://sparse.tamu.edu/?utm_source=chatgpt.com)
- Base de redes sociais, químicas, neurais:  
[https://github.com/TrustAGI-Lab/graph\\_datasets/tree/master](https://github.com/TrustAGI-Lab/graph_datasets/tree/master)



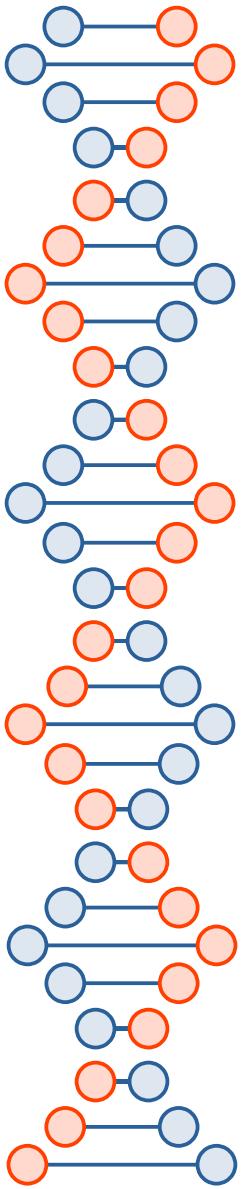
# Geração de grafos aleatórios

- A biblioteca **NetworkX** é a mais usada para lidar com grafos em Python e já tem funções prontas para gerar grafos aleatórios.
- O comando abaixo em python cria um grafo não direcionado com 10 vértices, onde cada par de vértices tem probabilidade 0.3 de estar conectado:
  - $G = nx.erdos\_renyi\_graph(10, 0.3)$



# Geração de grafos aleatórios

- A fim de gerar a base de grafos  $G(n,p)$  seguindo o modelo de Erdös-Rényi, podemos escolher uma faixa para o tamanho  $n$  dos grafos. Por exemplo,  $n = 50, 100, 150, 200, 250, \dots, 500$ .
- E para cada valor de  $n$ , geramos três probabilidades:
  - $P = 0.2, 0.5, 0.8$
- Assim, teremos tanto grafos esparsos quanto densos para todos os tamanhos.



FIM