

# Transició de fase i components connexes en grafs aleatoris

## Build

Per generar les binàries cal fer `make` i es generen a la carpeta `bin/`.

Tenim dues binàries `bin_Ncomp` i `geo_Ncomp` i reben els mateixos parametres:

1. `N` – nombre de vèrtex del graf
2. `rep` – nombre de repeticions per la mitja
3. `dy` – determina el nombre de dades que es calcula, com més petit més dades es calcularan

Treu per pantalla les dades calculades en un format que després podrem llegir en Python.

```
1 $ ./bin/geo_Ncomp 100 500 0.1
2 # N=100 REP=500 dy=0.1
3 # avg_Ncomp r
4 0 100
5 0.0015625 99.958
6 0.00234375 99.89
7 0.003125 99.83
8 0.00390625 99.796
9 0.0046875 99.66
```

## Càlcul

Per calcular les dades tenim un script en Python `compute.py`. Executa `geo_Ncomp` o `bin_Ncomp` per diferents valors de `N` passats com a arguments.

## Exemple

```
1 # genera dades de bin_Ncomp per valors de N 10, 20, 50 i 100 i les dades
  es guarden al directori data
2 python3 compute.py --repetitions 500 -d 0.1 --out-dir data ./bin/bin_Ncomp
  10 20 50 100
```

# Gràfics

Per generar els gràfics utilitzem un script de Python `plot.py` i els parametres es poden veure fent `python3 plot.py -h`.

## Exemple

```
1 # plot de BRG per totes les dades a la carpeta data
2 python3 plot.py -t "Binomial Random Graph \[p \leq 0.4\]" -x p -y "Nombre
  components connexos" --xmax 0.4 --show-legend -o plots/bin_mult_0.4.pdf
  data/bin*.dat
3
4 # plot de GRG per N=25
5 python3 plot.py -t "Geometric Random Graph \[N = 25\]" -x p -y "Nombre
  components connexos" -o plots/geo_Ncomp_0025.pdf data/binNcomp_0025.dat
```

## makePlots.sh

Es un script de bash que executa `compute.py` i `plot.py` per tal de automatitzar el càlcul de dades i generació de gràfics.

Guarda les dades a la carpeta `data/` i els gràfics a `plots/`.

```
1 bash makePlots.sh 10 25 50 60 80 100 150
```