Relazione TRMD

Valerio Tincani

22 Settembre 2025

1 Obiettivi

L'obiettivo di questa relazione è l'analisi dei dati della pandemia di COVID-19 nel periodo 2020-2025, ponendo particolare attenzione ai primi 35 giorni dall'inizio del contagio: nel dettaglio verrà illustrato come si è riusciti a fare un fit secondo due leggi differenti, quella logistica e quella di Gompertz. Fatto ciò verranno confrontate le previsioni così ottenute per i primi 100 giorni dell'epidemia con i dati realmente registrati. Il tutto è stato eseguito con un programma Python e in particolare con il ricorso a emcee.

2 Introduzione teorica

Per vincolare i parametri di un dato modello, alla luce di un certo set di dati raccolti, è necessario ricordare qualche nozione base; primo fra tutti è il Teorema di Bayes:

$$P(\Theta|x) = \frac{P(x|\Theta)P(\Theta)}{P(x)} \tag{1}$$

dove si considerano un certo modello parametrizzato da Θ e dei dati x. In ordine, da sinistra verso destra, dall'alto verso il basso, sono presenti il posterior, la likelihood, il prior e l'evidence: il primo indica la probabilità che i dati, una volta osservati, siano modellizzati attraverso l'uso dei parametri Θ, la seconda si riferisce alla probabilità che dato un modello con parametri fissati Θ i dati osservati siano effettivamente gli x, il terzo è la probabilità a priori che il modello sia descritto attraverso Θ e infine la quarta è la probabilità di misurare i dati x, qualunque sia il modello corretto.
Ciò che è interessante ai fini di questa discussione è il valore del posterior, in quanto idealmente l'obiettivo è trovare quel valore di Θ che massimizzi tale termine. Per farlo verrà generato un campione di parametri, più o meno probabili alla luce dei dati raccolti, che segua la distribuzione di probabilità data dal posterior: a quel punto si potrà assumere come punto sufficientemente vicino al picco (e quindi più probabile) la mediana del sample (non la media perchè non è detto che la distribuzione sia simmetrica).
Concretamente questo passaggio verrà eseguito con l'aiuto dell'algoritmo emcee,

che è una "rivisitazione" di quello di Metropolis-Hastings. In breve l'algoritmo esplora lo spazio dei parametri in cerca della zona che massimizza il posterior e per farlo si serve di un certo numero di walkers, ovvero di "entità" che "saltano" da un punto all'altro secondo una certa regola: essi si possono allontanare o avvicinare a un altro walker preso arbitrariamente con una probabilità data dal rapporto del posterior calcolato nelle due posizioni dei walkers coinvolti (al numeratore c'è il punto verso cui mi dovrei muovere). Ne consegue che per l'esecuzione dell'algoritmo l'evidence è ininfluente e quindi ci si può limitare al numeratore del teorema di Bayes.
Nel caso di emcee, per convenienza nei calcoli, si considerano i logaritmi naturali delle varie probabilità, che quindi invece di moltiplicarsi tra loro vanno ora sommate:

$$log\ posterior = log\ likelihood + log\ prior$$

Sia la likelihood che il prior sono determinati dal modello che scegliamo di adottare: nel nostro caso assumiamo che i contagi giornalieri siano una variabile casuale poissoniana in cui il parametro μ, ovvero i contagi attesi in media, sono dati dal valore calcolato con equazione logistica o di Gompertz. Di conseguenza il termine *log likelihood* si ricaverà nel seguente modo:

$$logP(x|\mu) = -\mu + xlog\mu - log(x!)$$

Nel nostro caso consideriamo una collezione di μ, uno per ogni giorno, quindi a questa espressione verrà aggiunta una sommatoria sui μ. Infine come prior ho scelto una distribuzione uniforme compresa tra dei valori di bordo che circoscrivano un intervallo verosimile per i parametri: questo perchè a priori non si è a conoscenza di molto riguardo a N e k.

3 Struttura e commento del programma

Lo script Python ha richiesto l'importazione di alcune librerie specifiche: numpy per effettuare operazioni con array in modo immediato (senza ricorrere a loop), pandas per l'analisi dei dati basata su data frame, matplotlib per utilizzarne in particolare le funzioni relative al plot di grafici e al trattamento delle date, scipy per le funzioni minimize e gammaln (usata nel calcolo di *log likelihood*), e infine emcee per l'algoritmo omonimo.
Il programma è diviso in tre blocchi fondamentali: definizione delle funzioni, esecuzione dell'analisi dati, generazione dei grafici.
Per quanto riguarda la prima parte si è partiti dalle funzioni cumulative per i due modelli considerati (eq. logistica ed eq. di Gompertz) i cui argomenti sono i due parametri da stimare (N e k) e un array denominato t (che di fatto indica la differenza in giornate tra l'i-esimo giorno analizzato e il primo, che è $t_0 = 0$).
A queste funzioni sono state aggiunte:

- **daily from cum func** che restituisce direttamente un array che ha come i-esima entrata il numero di nuovi positivi dell $(i+1)$-esimo giorno tramite

la differenza tra due chiamate della funzione cumulativa traslate di una unità lungo il tempo

- **poisson loglike** che restituisce il logaritmo naturale della likelihood (float) dei dati osservati per un dato valore dei parametri N e k (per la formula completa vedere sezione precedente)

- **build priors from data** e **log prior** definiscono i priors di N e k come funzioni costanti con supporto $[N_{min}; N_{max}]$ e $[k_{min}; k_{max}]$ dove i bordi sono stati decisi con una "educated guess" basata sui contagi effettivamente registrati nei primi 35 giorni di pandemia e su valori verosimili di k (il logaritmo naturale sarà $-\infty$ fuori dal supporto del prior e 0 al suo interno)

- **log probability** calcola il logaritmo del posterior (float) per una data scelta dei parametri N e k facendo la somma dei di log prior e log likelihood, secondo il teorema di Bayes

- **fit with emcee N** è la funzione che vincola tramite algoritmo emcee i valori dei parametri N e k basandosi sui dati raccolti : i suoi argomenti sono l'array t, già presente nelle funzioni cumulate definite all'inizio, obs, ovvero l'array che riunisce i contagi osservati giorno per giorno, cum model func (corrisponderà alla funzione logistica o a quella di Gompertz a seconda dell'evenienza), priors, che è un dizionario che assegna i valori di bordo per N e k, e infine nwalkers, nsteps e discard burnin.
  Questa funzione propone un punto iniziale nello spazio dei parametri in cui N è il doppio dei contagi registrati fino al 30 marzo e k è 0.2, ma questo viene utilizzato solo nel caso in cui la funzione python "minimize" produca un errore: quest'ultima trova un minimo approssimativo del logaritmo della posterior (cambiato di segno) in modo da trovare un buon candidato come punto di partenza dei walkers.
  A questo punto vengono generati i punti di partenza dei vari walkers, applicando un rumore gaussiano al punto iniziale trovato con minimize: in particolare si applica un rumore proporzionale al valore di N e di k con in più un rumore di fondo costante per evitare che per valori troppo piccoli dei parametri i vari walkers partano eccessivamente vicini tra loro. Dopo un rapido controllo che tutti i walkers partano all'interno dei bordi, si inizializza il sampler e lo si esegue con funzione target la log probability. Infine vengono restituiti il sampler, i samples (con il burn-in già escluso) e il punto di partenza nello spazio dei parametri. In particolare i samples sono presentati come array 2D a 2 colonne, ognuna contenente il campione generato dai vari walkers per N e k (questo si ottiene appiattendo la chain resistuita dal sampler, che di default è 3D con shape nsteps*nwalker*ndim)

- **posterior predictive** esegue una veloce statistica sui valori dei parametri ricavati con emcee: in particolare l'utente può scegliere quanti dei samples ottenuti utilizzare; essi verranno scelti casualmente attraverso rng.choice. In seguito la funzione costruisce un array 2D (shape: nsamples*len(t)) che

riunisce le previsioni di contagi giornalieri per ognuno dei modelli ottenuti ponendo N e k uguali ai valori restituiti da rng.choice. Infine vengono restituiti i valori al 16esimo, 50esimo e 84esimo percentile (servirà per i grafici), in modo da avere il modello "mediano" e i modelli che si trovano a una deviazione standard da esso (assumendo che i modelli siano distribuiti normalmente, altrimenti i percentili delimitano comunque una "fascia di confidenza" accettabile)

- **summarize parameters** stima, sempre attraverso i percentili, dei valori "centrali" per i parametri e le relative incertezze, stampando a schermo le informazioni ricavate

Il corpo centrale del programma comincia con il definire la regione di interesse, il path da cui ricavare il file CSV contenente i dati, le date che circoscrivono il periodo di fit (nel nostro caso 24 febbraio 2020 e 30 marzo 2020) e il numero di giorni seguenti all'inizio del fit con cui vogliamo confrontare le previsioni dei nostri modelli.
In seguito il programma legge i dati nel file CSV e crea un dataframe che seleziona solo le righe corrispondenti alla regione scelta, le ordina in base alla data e pone sempre la data come indice di riga, assicurandosi che questo sia datetime in modo da poterci lavorare con più facilità. A questo punto viene fatta un'ulteriore selezione che restituisce solo le righe del dataframe nell'intervallo di 100 giorni da noi deciso. La stessa cosa viene fatta per l'intervallo di 35 giorni utilizzato per il fit, ottenendo infine i due array "obs daily 100" e "obs daily fit" che raccolgono i contagi giornalieri nei due periodi considerati per la stima dei parametri e per il confronto con le previsioni.
Viene dunque costruito l'array t già nominato basandosi sulla lunghezza di "obs daily fit", per poi impiegare le funzioni "build priors from data", "fit with emcee N" e "summarize parameters" sia con la funzione logistica che con quella di Gompertz; infine viene implementata anche "posterior predictive" per calcolare i modelli da graficare con relativo "intervallo di confidenza" e vengono stampate tutte le informazioni principali a schermo.
Da ultimo il programma si dedica alla costruzione dei grafici: un grafico che riassume tutti i contagi giornalieri registrati (da febbraio 2020 a gennaio 2025), un confronto tra previsioni del modello logistico e di Gompertz per i contagi giornalieri e per quelli cumulativi (sempre nella finestra di 100 giorni), i trace plots per i vari walkers utilizzati per l'emcee e uno schema riassuntivo dei movimenti di questi nello spazio dei parametri. Apparte questo, l'unico dettaglio degno di nota è la definizione di una funzione dedicata "plot walkers" per disegnare i grafici relativi ai walkers.

4 Risultati e conclusioni

Prima di passare ai risultati veri e propri tengo a precisare che l'equazione utilizzata per il modello logistico è stata leggermente modificata rispetto alla consegna, è stata impiegata infatti la seguente legge:

$$N_{(<t)} = \frac{N_{tot}}{1 + N_{tot} \cdot e^{-k(t-t_0)}} \tag{2}$$

ho apportato questa modifica per avere un inizio di curva molto basso che meglio si adattasse ai contagi registrati; al contrario la formula originale partiva da un numero troppo alto e il fit non veniva eseguito correttamente.
Passando ai risultati, i parametri sono stati vincolati correttamente per entrambi i modelli:

- *FIT LOGISTICA*

$$N_{50\%} = 1,68 \cdot 10^3$$

$$N_{16\%} = 1,63 \cdot 10^3$$

$$N_{84\%} = 1,72 \cdot 10^3$$

$$k_{50\%} = 0,265$$

$$k_{16\%} = 0,263$$

$$k_{84\%} = 0,267$$

- *FIT GOMPERTZ*

$$N_{50\%} = 5,20 \cdot 10^3$$

$$N_{16\%} = 4,78 \cdot 10^3$$

$$N_{84\%} = 5,71 \cdot 10^3$$

$$k_{50\%} = 0,0550$$

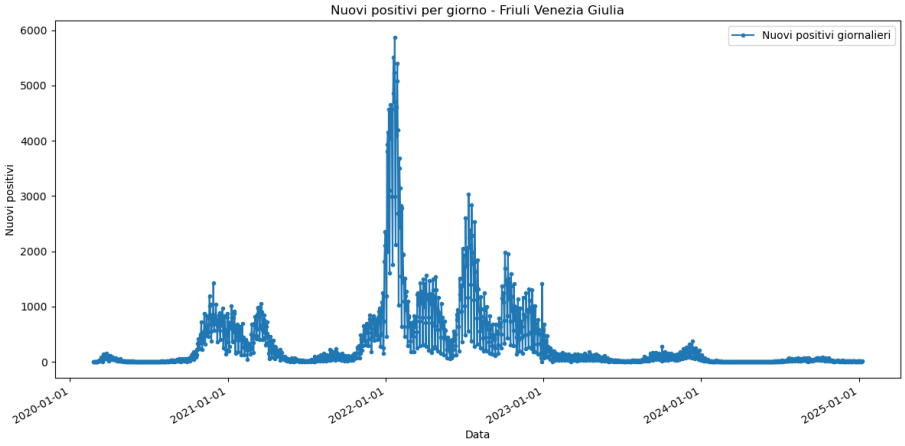
$$k_{16\%} = 0,0533$$

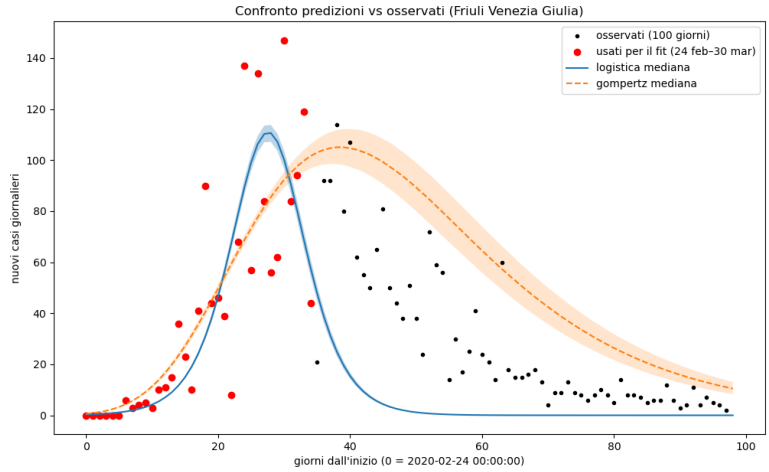
$$k_{84\%} = 0,0567$$

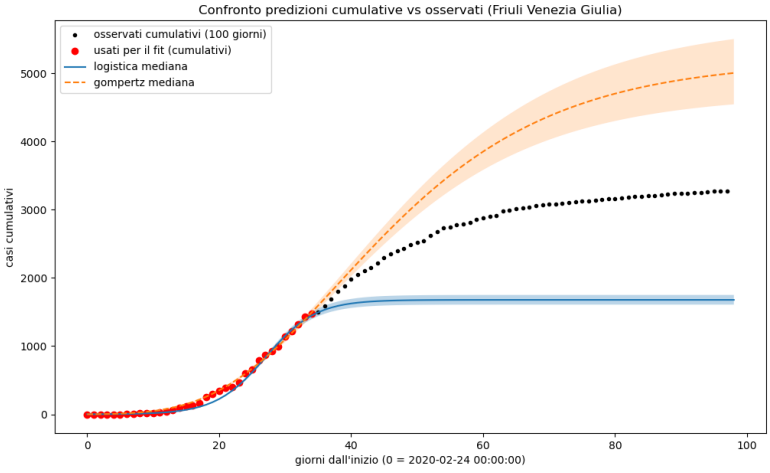
Ciò che salta subito all'occhio è la differenza notevole nella previsione sui contagi totali, comunque prevedibile se si considera il fatto che si parla di due modelli differenti; risulta però evidente anche una maggiore incertezza sul valore di questo parametro nel modello di Gompertz, in cui anche l'incertezza relativa su k è maggiore ma comunque contenuta.
Osservando i grafici dei contagi giornalieri e cumulativi si può ricondurre questo comportamento alla natura del modello di Gompertz di prevedere una diminuzione nel tasso di trasmissione della malattia più docile e graduale rispetto a quanto descritto dall'equazione logistica: nel caso di quest'ultima il modello "migliore" indica che al 35esimo giorno il contagio dovrebbe essere praticamente arrivato alla fase stazionaria e dunque il numero di positivi previsto non si discosta molto da quello effettivamente registrato fino al 30 marzo e l'incertezza sulla curva non ha modo di propagarsi a lungo.
Al contrario il modello di Gompertz suggerisce che il 30 marzo si trova addirittura prima del punto di flesso della curva cumulativa e quindi la previsione dei contagi totali a fine epidemia è molto maggiore; inoltre essendo in questo caso l'ondata di COVID più "lunga" le incertezze sulla curva hanno "più tempo" per propagarsi visto che al 100esimo giorno si è ancora visibilmente lontani da un definitivo appiattimento.
Anche direttamente dai grafici si vede che il fattore soggetto a maggiore incertezza (nel caso di Gompertz) è proprio il numero di contagi, in quanto quello che cambia nella "fascia di confidenza" è più "l'altezza" della curva (legata ad N) che non la forma o la larghezza della stessa (regolata invece da k). Tale osservazione è corroborata anche dai grafici che mappano le posizioni dei walkers nello spazio dei parametri: nel caso del modello logistico essi si dispongono in un intorno più o meno circolare e quindi con incertezze relative sui due parametri comparabili. Per quanto riguarda Gompertz invece, si può osservare che l'intorno è decisamente schiacciato lungo la direzione corrispondente ad N, che quindi si dispone secondo una distribuzione ben più ampia di k.
In ogni caso proprio i grafici relativi ai walkers confermano la bontà del fit eseguito, in quanto è chiaro che essi "convergano" a un intorno del punto nello spazio dei parametri che meglio modellizza i contagi registrati; allo stesso tempo nel traceplot si vede come i walkers siano sempre localizzati in un certo range di valori e non si stabiliscano in punti stazionari (auspicabilmente si dovrebbe osservare qualcosa di simile a un rumore bianco, che corrisponde a quanto mostrato qui dai grafici).
In definitiva i due modelli non tengono conto di alcuni fattori che nella realtà dei fatti hanno prodotto una curva dei contagi visibilmente diversa da quelle predette: mi sento tuttavia di affermare che in questo caso il modello di Gompertz sia da ritenere quello più adatto, in quanto la decrescita dei contagi è sicuramente più vicina a quella graduale da esso descritta che non da quella repentina prevista dal modello logistico. Inoltre l'equazione di Gompertz fornisce una previsione piuttosto accurata nel breve termine (fino a 10 giorni circa dopo il 30 marzo) e verrebbe da pensare che il limite principale del modello in questo caso sia il ristretto sample di dati su cui è stato fatto il fit.
In ogni caso credo sia utile interpetare le due curve trovate come il miglior e il

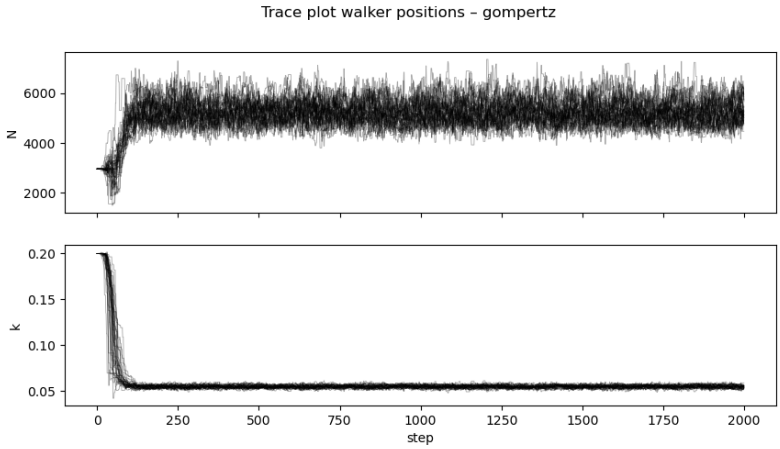
peggior scenario possibile per quanto riguarda le previsioni del contagio: salta all'occhio come la curva reale sia quasi perfettamente a metà tra quella logistica e quella di Gompertz, che in questa ottica servirebbero a circoscrivere il range delle varie possibilità nell'evoluzione dell'epidemia.
Infine, per quanto riguarda la seconda ondata di COVID (giugno 2020-giugno 2021) si può tentare un'analisi qualitativa: sono distinguibili due curve, di cui la prima ha uno "strascico" di contagi più prolungato mentre la seconda risulta più localizzata e simmetrica. Alla luce di questo la seconda ondata potrebbe essere suddivisa in due sotto-ondate di cui la prima è modellizzabile da una curva di Gompertz, mentre la seconda da una logistica (oppure sempre con Gompertz, ma con un k più elevato che "schiaccia" lateralmente il grafico).

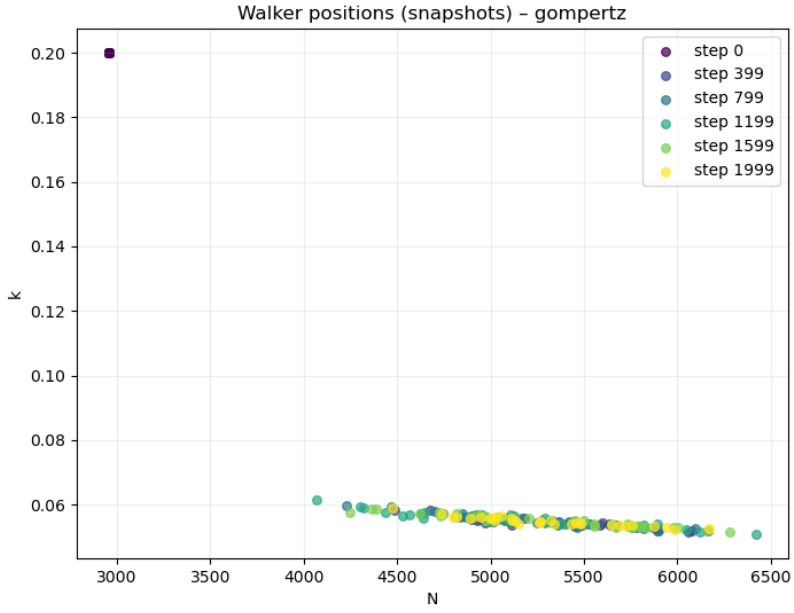
5 Grafici

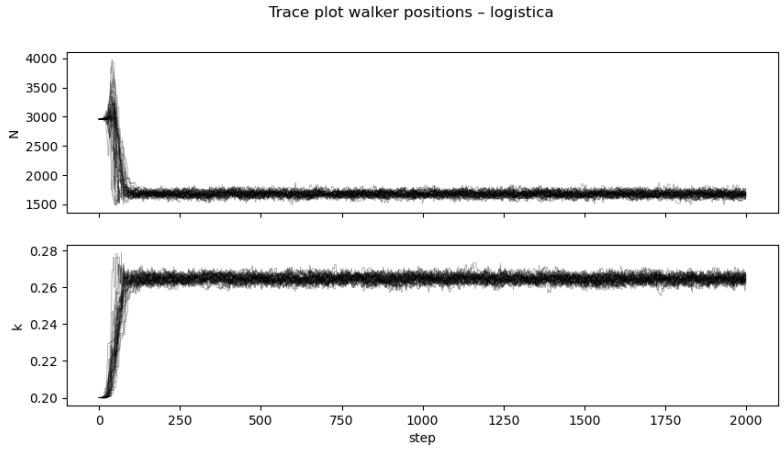

Nuovi positivi per giorno - Friuli Venezia Giulia
Nuovi positivi giornalieri
Nuovi positivi
6000
5000
4000
3000
2000
1000
0
2020-01-01
2021-01-01
2022-01-01
2023-01-01
2024-01-01
2025-01-01
Data

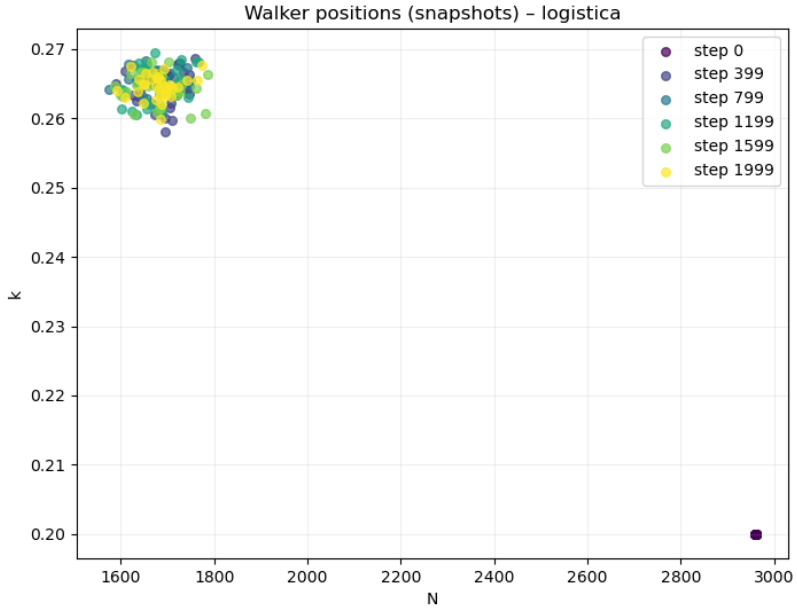

Confronto predizioni vs osservati (Friuli Venezia Giulia)
osservati (100 giorni)
usati per il fit (24 feb–30 mar)
logistica mediana
gompertz mediana
nuovi casi giornalieri
giorni dall'inizio (0 = 2020-02-24 00:00:00)


Confronto predizioni cumulative vs osservati (Friuli Venezia Giulia)
osservati cumulativi (100 giorni)
usati per il fit (cumulativi)
logistica mediana
gompertz mediana
casi cumulativi
giorni dall'inizio (0 = 2020-02-24 00:00:00)


Trace plot walker positions – gompertz
N
6000
4000
2000
k
0.20
0.15
0.10
0.05
0
250
500
750
1000
1250
1500
1750
2000
step


Walker positions (snapshots) – gompertz
step 0
step 399
step 799
step 1199
step 1599
step 1999
k
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
3000
3500
4000
4500
5000
5500
6000
6500
N


Trace plot walker positions – logistica
N
k
step
4000
3500
3000
2500
2000
1500
0.28
0.26
0.24
0.22
0.20
0
250
500
750
1000
1250
1500
1750
2000


Walker positions (snapshots) – logistica
step 0
step 399
step 799
step 1199
step 1599
step 1999
k
N
0.27
0.26
0.25
0.24
0.23
0.22
0.21
0.20
1600
1800
2000
2200
2400
2600
2800
3000