[Supplementary Material]

Bayesian analysis and free market trade within the Roman Empire

Xavier Rubio-Campillo[1,2,*], María Coto-Sarmiento[2], Jordi Pérez-Gonzalez[3] & José Remesal Rodríguez[3]

[1] School of History, Classics and Archaeology, University of Edinburgh, William Robertson Wing, Old Medical School, 4 Teviot Place, Edinburgh EH8 9AG, UK

[2] Barcelona Supercomputing Center, Carrer de Jordi Girona 29–31, 08034 Barcelona, Spain

[3] CEIPAC, Departament de Prehistòria, Història Antiga y Arqueologia, Universitat de Barcelona, Carrer de Montalegre 6, 08001 Barcelona, Spain

[*] Author for correspondence (Email: xavier.rubio@ed.ac.uk)

Received: 23 June 2016; Accepted: 23 September 2016; Revised: 12 October 2016

1 Integrating dataset uncertainty

The analyzed dataset contains 7520 stamps collected from Dressel 20 amphorae found in Monte Testaccio. The exact alphabetic code of 5743 is fully readable, while 1777 stamps contain codes where part of the information has been lost to erosion or fragmentation. The lost fragments have been defined with simple regular expressions of two types: a) a single missing character (symbol '.') and b) a group of 1 or more characters that cannot be read (symbol '+').

These incomplete stamps form a substantive corpus of evidence that should be added to the analysis. The uncertainty has been integrated using a probabilistic approach. First, for each incomplete stamp a list of candidates is created with all the complete stamps matching the regular expression. Second, a Monte Carlo simulation is executed to randomly assign one of the candidates each incomplete stamp. Third, the aggregation of the complete and the reconstructed stamps is used as final dataset. The method is summarized as follows:

1. Initialize set C containing all complete codes, set U with the incomplete stamps, and the final set R as empty
2. For each incomplete stamp S in dataset U:
   (a) Create a list L of possible candidates from C matching the regular expression contained in S
   (b) If L is empty then add S to R
   (c) If L is not empty then with uniform probability select one complete stamp L_i from L and add it to R
3. Create the final dataset $F = \cup(C, R)$

The method is illustrated by two examples in Figure 1.

Results show that the outcome of different Monte Carlo runs generate similar datasets with some degree of variability, as can be seen in Figure 2.

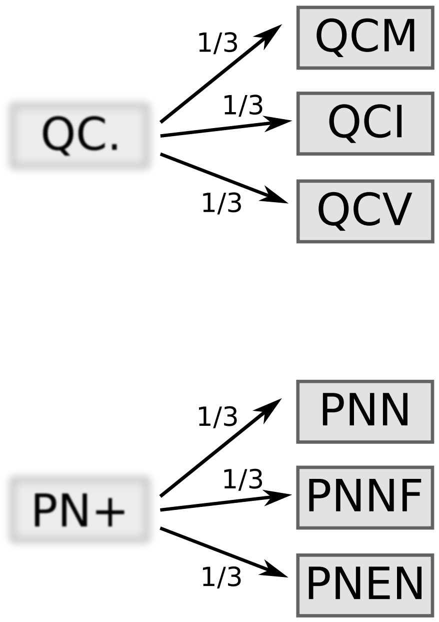

Figure 1: Examples of regular expressions used in the reconstruction algorithm. In the first case the last letter was unreadable while in the second case it is not even clear if there were one or more letters. Three candidates have been identified for both cases and each of the incomplete stamps will be assigned as one of them with $P = 1/3$.

We have also assessed the impact of the integrated uncertainty by replicating the analysis with different datasets. Table 1 shows DIC values computed just using the dataset C of complete codes. The order of preference of the four models is the same while the distance between $M4$ and the rest has increased.

Table 1: DIC measures using the dataset of complete codes and discarding the entire set of incomplete stamps.

Model	Mean Deviance	Penalty	Penalized deviance	ΔDIC
M1	14158	1.026	14159	10447
M2	6290	1.113	6291	2579
M3	5327	1.988	5329	1617
M4	3711	1.015	3712	0

Figure 3 displays density estimates of DIC values for 100 different runs

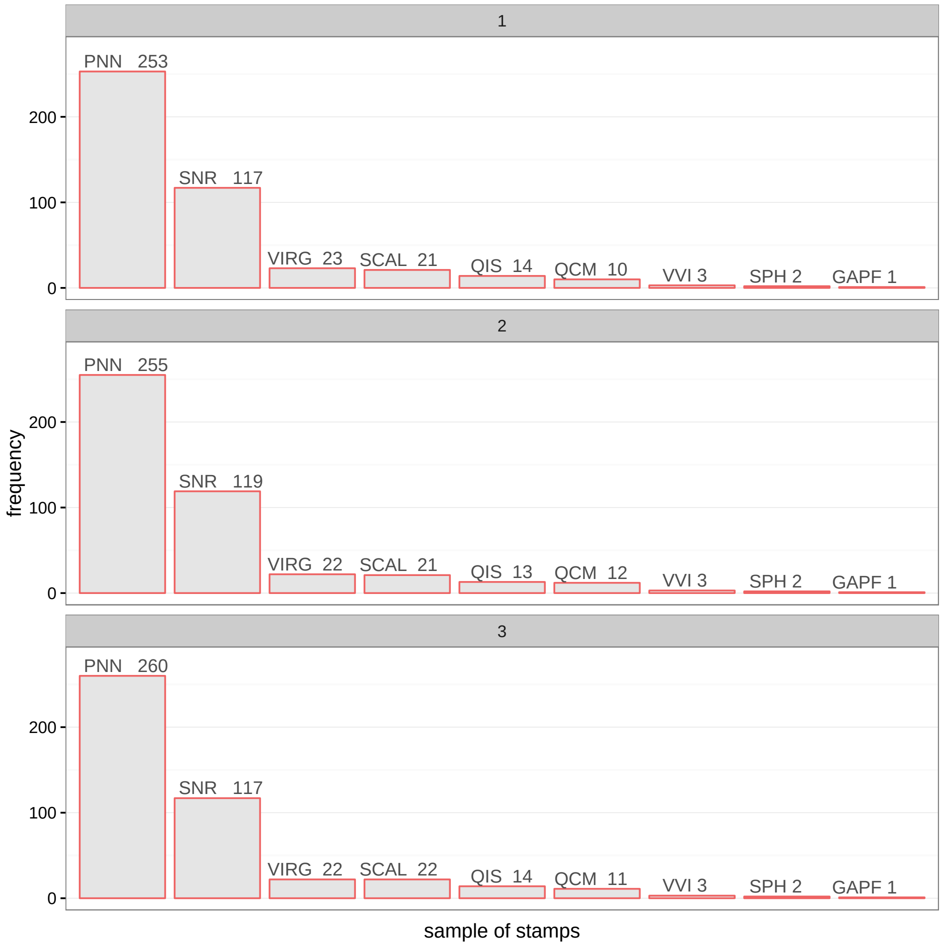

Figure 2: Sample of stamps from 3 different Monte Carlo runs. The frequency of 9 representative codes is shown for illustrative purposes.

integrating reconstructed stamps with the previously defined Monte Carlo method. This test shows how variation in the final measure is always lower than 2% and the order of preference is constant. The best model *M4* is the one with highest sensitivity to the uncertainty of the dataset. Slight variations on the frequency distribution have higher impact on the fat tail of the power-law than on the rest of models.

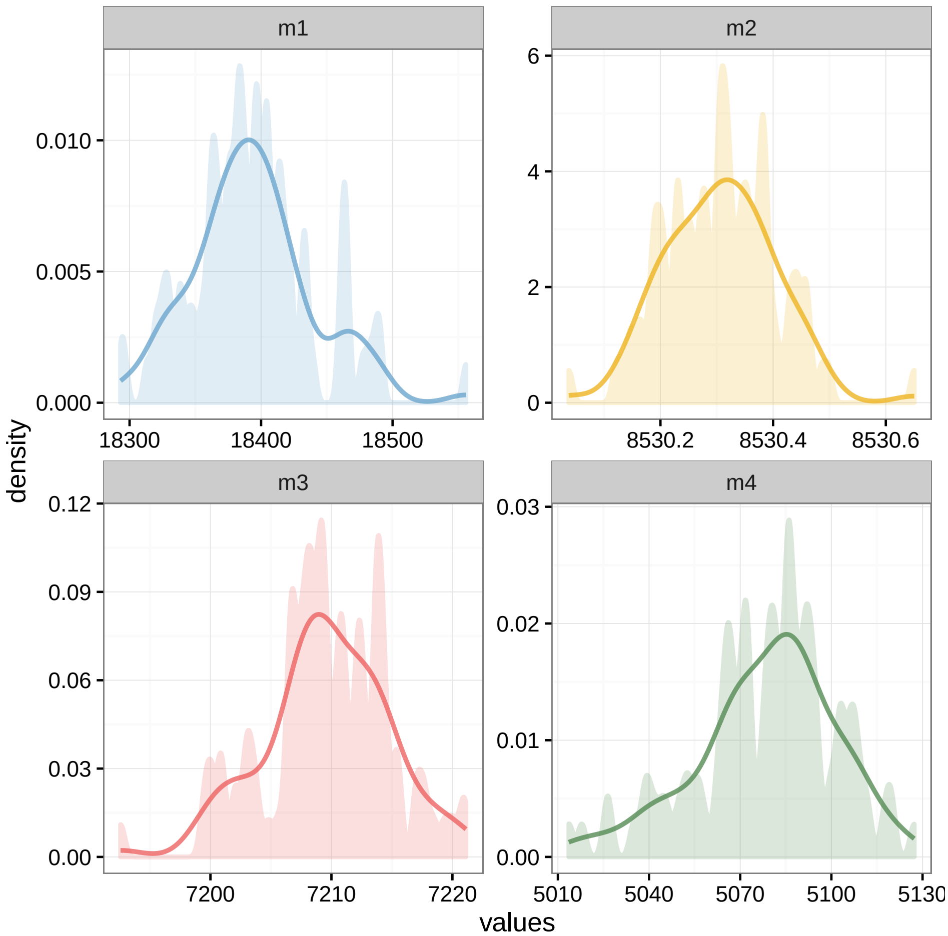

Figure 3: Kernel Density Estimates of penalized DIC for each model.

2 Markov Chain Monte Carlo execution

Table 2 summarizes the parameters of the MCMC used to compute the posterior distributions.

Table 2: Parameters for the Markov Chain Monte Carlo simulation

Parameter	Value
Number of chains	3
Burn-in period	500
Number of iterations	15500
Thinning	1