Supplementary Material for "An inference method for global sensitivity analysis"

Gildas Mazo
Université Paris-Saclay, INRAE, MaIAGE, 78350, Jouy-en-Josas, France
and
Laurent Tournier
Université Paris-Saclay, INRAE, MaIAGE, 78350, Jouy-en-Josas, France

October 30, 2024

A Detailed calculations for the examples of Section 2

We have

$$\begin{pmatrix} \tau^*(\emptyset) \\ \tau^*(\{1\}) \\ \tau^*(\{2\}) \\ \tau^*(\{3\}) \\ \tau^*(\{1,2\}) \\ \tau^*(\{1,3\}) \\ \tau^*(\{2,3\}) \\ \tau^*(\{1,2,3\}) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \sigma^*(\emptyset) \\ \sigma^*(\{1\}) \\ \sigma^*(\{2\}) \\ \sigma^*(\{3\}) \\ \sigma^*(\{1,2\}) \\ \sigma^*(\{1,3\}) \\ \sigma^*(\{2,3\}) \\ \sigma^*(\{1,2,3\}) \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ \frac{1}{2} + \frac{b\pi^4}{5} + \frac{b^2\pi^8}{50} \\ \frac{a^2}{8} \\ 0 \\ \frac{1}{2} + \frac{b\pi^4}{5} + \frac{b^2\pi^8}{50} + \frac{a^2}{8} \\ \frac{1}{2} + \frac{b\pi^4}{5} + \frac{b^2\pi^8}{18} \\ \frac{a^2}{8} \\ \frac{1}{2} + \frac{b\pi^4}{5} + \frac{b^2\pi^8}{18} + \frac{a^2}{8} \end{pmatrix}$$

and

$$\begin{pmatrix} \tau(\emptyset) \\ \tau(\{1\}) \\ \tau(\{2\}) \\ \tau(\{3\}) \\ \tau(\{1,2\}) \\ \tau(\{1,3\}) \\ \tau(\{2,3\}) \\ \tau(\{1,2,3\}) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \sigma^*(\emptyset) \\ \sigma^*(\{1\}) \\ \sigma^*(\{2\}) \\ \sigma^*(\{3\}) \\ \sigma^*(\{1,2\}) \\ \sigma^*(\{1,3\}) \\ \sigma^*(\{2,3\}) \\ \sigma^*(\{1,2,3\}) \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ \frac{1}{2} + \frac{b\pi^4}{5} + \frac{b^2\pi^8}{18} \\ \frac{a^2}{8} \\ \frac{8b^2\pi^8}{225} \\ \frac{1}{2} + \frac{b\pi^4}{5} + \frac{b^2\pi^8}{18} + \frac{a^2}{8} \\ \frac{1}{2} + \frac{b\pi^4}{5} + \frac{b^2\pi^8}{18} \\ \frac{a^2}{8} + \frac{8b^2\pi^8}{225} \\ \frac{1}{2} + \frac{b\pi^4}{5} + \frac{b^2\pi^8}{18} + \frac{a^2}{8} \end{pmatrix}$$

and

$$\begin{pmatrix} \theta(\{1\}) \\ \theta(\{2\}) \\ \theta(\{3\}) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 1/2 & 1/2 & 0 & 1/3 \\ 0 & 0 & 1 & 0 & 1/2 & 0 & 1/2 & 1/3 \\ 0 & 0 & 0 & 1 & 0 & 1/2 & 1/2 & 1/3 \end{pmatrix} \begin{pmatrix} \sigma^*(\emptyset) \\ \sigma^*(\{1\}) \\ \sigma^*(\{2\}) \\ \sigma^*(\{3\}) \\ \sigma^*(\{1,2\}) \\ \sigma^*(\{1,3\}) \\ \sigma^*(\{2,3\}) \\ \sigma^*(\{1,2,3\}) \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{2} + \frac{b\pi^4}{5} + \frac{17b^2\pi^8}{450} \\ \frac{a^2}{8} \\ \frac{4b^2\pi^8}{225} \end{pmatrix}.$$

B A reminder about Shapley values in cooperative game theory and Shapley effects in sensitivity analysis

Shapley effects assess the global importance of each individual input (Owen, 2014). They result from an allocation method of cooperative game theory (Shapley, 1951; Winter, 2002). If D is seen as a set of players, then to each coalition $B \subset D$ there corresponds a real number, denoted by $\mathrm{val}(B)$, which represents the "value" of B. It is assumed that $\mathrm{val}(\emptyset) = 0$. The Shapley value of player j, $j = 1, \ldots, d$, is defined as

$$\theta_{\mathrm{val}}(\{j\}) = \frac{1}{d} \sum_{B \subset D \setminus \{j\}} \frac{1}{\binom{d-1}{|B|}} (\mathrm{val}(B \cup \{j\}) - \mathrm{val}(B)).$$

The value of player j is based on the change in value of the coalitions, should that player be added in them. In fact, it was shown by Shapley (Shapley, 1951) that this way of assigning values is the only possible way that satisfy certain axioms, among which $\mathrm{val}(D) = \sum_{j=1}^{d} \theta_{\mathrm{val}}(\{j\})$. See Shapley (1951); Winter (2002) for more details. The Shapley value can be re-expressed (Harsanyi, 1963) as

$$\theta_{\mathrm{val}}(\{j\}) = \sum_{B \cap \{j\} \neq \emptyset} \frac{\sum_{A \subset B} (-1)^{|B \setminus A|} \mathrm{val}(A)}{|B|},$$

a formula that will be useful later on. If val is replaced by its dual $\mathrm{val}^*(B) = \mathrm{val}(D) - \mathrm{val}(D \setminus B)$ then $\theta_{\mathrm{val}}(\{j\}) = \theta_{\mathrm{val}^*}(\{j\})$ for all $j = 1, \ldots, d$. We say that the Shapley value is self-dual, see Funaki (1998); Oishi et al. (2016).

By an analogy between cooperative game theory and global sensitivity analysis (the individual inputs X_j play the role of the players), we can choose $\mathrm{val}(B)$ to be

$$\mathrm{Var}\,\mathrm{E}(f(X)|X_B) = \tau^*(B) \text{ or } \mathrm{E}\,\mathrm{Var}(f(X)|X_{D \setminus B}) = \tau(B),$$

leading *a priori* to two sets of indices $\theta_{\tau^*}(\{j\})$ and $\theta_{\tau}(\{j\})$. The first choice was made in Owen (2014) and the second in Song et al. (2016). But since τ and τ^* are duals of each other, it holds that $\theta_{\tau}(\{j\}) = \theta_{\tau^*}(\{j\})$ for every $j = 1, \ldots, d$. This property was noticed in Song et al. (2016) and is a direct

consequence of the self-duality of the Shapley value in cooperative game theory. Thus, it makes sense to define

$$\theta(\{j\}) := \theta_\tau(\{j\}) = \theta_{\tau^*}(\{j\}),$$

which are called Shapley effects in sensitivity analysis.

C Reminder about asymptotic confidence intervals and p-values

Let $\psi \in \mathbf{R}^{2^d}$ be some unknown vector of interest with components indexed by the subsets of D, that is, with components $\psi(A)$, $A \subset D$. Let $\widehat{\psi}$ be an estimator of ψ. Consider first the situation where it is known that $\widehat{\psi}$ follows a multivariate normal distribution with mean ψ and variance-covariance matrix Ψ/n. Then, for any given $A \subset D$, we have that $\widehat{\psi}(A)$ is normally distributed with mean $\psi(A)$ and variance $\Psi(A,A)/n$ and hence we could calculate a confidence interval of level, say $1-\alpha \in (0,1)$ under the form $[\widehat{\psi}(A) \pm q_{1-\alpha/2}\sqrt{\Psi(A,A)/n}]$, where $q_{1-\alpha/2}$ is the quantile of order $1-\alpha/2$ of a standard normal distribution.

In all but the most simple cases, however, it is unknown whether $\widehat{\psi}$ actually follows a multivariate normal distribution. But if we can establish that $\widehat{\psi}$ approximately follows a multivariate normal distribution, that is, if we can establish that

$$\sqrt{n}(\widehat{\psi} - \psi) \xrightarrow{\mathrm{d}} \mathrm{N}(0, \Psi)$$

for some asymptotic variance-covariance matrix Ψ, then we could do as above and get an approximate confidence interval.

The same goes for hypothesis testing, which is another way of making inferences. Remember that to decide between a null hypothesis H_0 and an alternative H_1, one computes a statistic S_n from the data and rejects H_0 if S_n belongs to some "rejection set". For simplicity, assume that rejection set is of the form (r, ∞), so that rejection occurs if $S_n > r$. The so-called type I and type II errors of the test are given by $\Pr_{\mathrm{H}_0}(S_n > r)$ (probability of rejecting H_0 while it is true) and $\Pr_{\mathrm{H}_1}(S_n \leq r)$ (probability of accepting H_0 while H_1 is true), respectively. The p-value of the test is given by $\Pr_{\mathrm{H}_0}(S_n > s_n^{(\mathrm{obs})})$ (probability under H_0 of observing a statistic beyond the one actually observed in the data), where $s_n^{(\mathrm{obs})}$ stands for the statistic actually observed

in the dataset at hand. Given some level of significance $1-\alpha \in (0,1)$, one seeks the least critical value r such that $\Pr_{\mathrm{H}_0}(S_n > r) \leq \alpha$. If it is known that S_n obeys a standard normal distribution then $r = q_{1-\alpha}$ and the test can be carried out. Otherwise, as for confidence intervals, one wants to establish that S_n is approximately normal and get approximate critical values.

To establish that $\widehat{\psi}$ and S_n are approximately normal, one uses limit theorems of asymptotic statistics. The obtained approximate confidence intervals and p-values are then sometimes said to be asymptotic. The approximation improves as the sample size n increases.

D Building better estimators of the vector of total indices

While the estimator $\widehat{\tau}$ in Section 3.1 is conceptually simple and widely used, we can construct more efficient estimators. Efficiency can be improved in two ways: one builds an estimator with a smaller variance or one builds an estimator with the same variance but with a smaller number of function evaluations. We shall do both.

D.1 Estimators with a smaller variance

Let $\widehat{\tau}^{(\alpha)}(A) = \alpha\widehat{\tau}^{(1)}(A) + (1-\alpha)\widehat{\tau}^{(0)}(A)$, where $\alpha \in [0,1]$, $A \subset D$ and

$$\widehat{\tau}^{(1)}(A) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}(f(X^{(i)}) - f(X^{(i)\setminus A}))^2,$$
$$\widehat{\tau}^{(0)}(A) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}(f(X'^{(i)}) - f(X'^{(i)\setminus A}))^2.$$

Note that $\widehat{\tau}^{(1)}(A)$ is Jansen's estimator mentioned in Section 3.1, that is, $\widehat{\tau}^{(1)}(A) = \widehat{\tau}(A)$. Note also that $\mathrm{E}\,\widehat{\tau}^{(0)}(A) = \mathrm{E}\,\widehat{\tau}^{(1)}(A) = \tau(A)$ and $\mathrm{Var}\,\widehat{\tau}^{(0)}(A) = \mathrm{Var}\,\widehat{\tau}^{(1)}(A)$. Let $\widehat{\tau}^{(\alpha)}$ be the vector formed by stacking all $\widehat{\tau}^{(\alpha)}(A)$ with $A \subset D$. If M is a matrix then denote by $\mathrm{tr}(M)$ the trace of M, that is, the sum of its diagonal elements.

Proposition D.1. *The following statements hold:*

(i) $\mathrm{E}[\widehat{\tau}^{(\alpha)}(A)] = \tau(A)$ *and* $\mathrm{Var}[\widehat{\tau}^{(\alpha)}(A)] \leq \mathrm{Var}[\widehat{\tau}(A)]$ *for every* $\alpha \in [0,1]$*;*

(ii) the variance of $\widehat{\tau}^{(\alpha)}(A)$ *is minimum when* $\alpha = 1/2$ *and*

$$0 \leq \mathrm{Var}[\widehat{\tau}(A)] - \mathrm{Var}[\widehat{\tau}^{(1/2)}(A)] = \frac{\mathrm{Var}[\widehat{\tau}(A)] - \mathrm{Cov}[\widehat{\tau}(A), \widehat{\tau}^{(0)}(A)]}{2};$$

(iii) $\sqrt{n}(\widehat{\tau}^{(\alpha)} - \tau) \xrightarrow{\mathrm{d}} \mathrm{N}(0, T_\alpha)$*, with* $T_\alpha = (2\alpha^2 - 2\alpha + 1)T + 2\alpha(1-\alpha)C$*, where* T *was defined in Proposition 1 and* C *is the matrix of size* $2^d \times 2^d$ *given by*

$$C(A, B) = \mathrm{Cov}\left[\frac{1}{2}(f(X') - f(X'^{\setminus A}))^2, \frac{1}{2}(f(X) - f(X^{\setminus B}))^2\right].$$

Moreover, $0 \leq \mathrm{tr}(T) - \mathrm{tr}(T_\alpha) \leq \mathrm{tr}(T) - \mathrm{tr}(T_{1/2}) = \frac{1}{2}\mathrm{tr}(T - C)$ *for every* $\alpha \in [0, 1]$.

The proof of Proposition D.1 follows from elementary calculations and the standard delta method. Proposition D.1 implies that $\widehat{\tau}^{(1/2)}$ is componentwise more efficient—both at finite and infinite sample sizes—than any $\widehat{\tau}^{(\alpha)}$, including $\widehat{\tau}$ itself.

D.2 An estimator with a smaller number of function evaluations

Instead of looking for an estimator with a smaller variance as above, we can, in the spirit of Saltelli (2002), turn the problem around and look for an estimator that requires fewer evaluations of function f. Consider the following sampling scheme. For each $i = 1, \ldots, n$, draw two independent copies $X^{(i)} = (X_1^{(i)}, \ldots, X_d^{(i)})$ and $X'^{(i)} = (X_1'^{(i)}, \ldots, X_d'^{(i)})$ and then perform the steps below:

$$\left\{\begin{array}{l} \text{compute } f(X^{(i)}), f(X'^{(i)}) \\ \quad \text{for } k = 1, \ldots, \lfloor d/2 \rfloor \\ \quad\quad \text{for } A \in \mathbf{2}^D : |A| = k \\ \quad\quad\quad \text{compute } f(X^{(i)\setminus A}). \end{array}\right. \tag{D.1}$$

Then, define for every $A \subset D$,

$$\widehat{\tau}'(A) = \left\{\begin{array}{ll} \frac{1}{n}\sum_{i=1}^n \frac{1}{2}\left(f(X^{(i)}) - f(X^{(i)\setminus A})\right)^2 & \text{if } |A| \leq \lfloor d/2 \rfloor \\ \frac{1}{n}\sum_{i=1}^n \frac{1}{2}\left(f(X'^{(i)}) - f(X_i^{(i)\setminus A^{\mathrm{c}}})\right)^2 & \text{otherwise,} \end{array}\right. \tag{D.2}$$

where A^{c} stands for $D \setminus A$, the complement of A in D. To shorten the notation, denote $Y_{A,B} = (f(X^{\setminus A}) - f(X^{\setminus B}))^2/2$ for every $A, B \subset D$.

Proposition D.2. *The following statements are true:*

(i) it holds that

$$\sqrt{n}(\widehat{\tau}' - \tau) \xrightarrow{\mathrm{d}} \mathrm{N}(0, T'),$$

where T' is the variance-covariance matrix given by

$$T'(A,B) = \begin{cases} \mathrm{Cov}(Y_{\emptyset,A}, Y_{\emptyset,B}) & \text{if } |A| \leq \lfloor d/2 \rfloor, |B| \leq \lfloor d/2 \rfloor, \\ \mathrm{Cov}(Y_{\emptyset,A}, Y_{D,B^c}) & \text{if } |A| \leq \lfloor d/2 \rfloor, |B| > \lfloor d/2 \rfloor, \\ \mathrm{Cov}(Y_{D,A}, Y_{\emptyset,B}) & \text{if } |A| < \lfloor d/2 \rfloor, |B| \leq \lfloor d/2 \rfloor, \\ \mathrm{Cov}(Y_{D,A^c}, Y_{D,B^c}) & \text{if } |A| > \lfloor d/2 \rfloor, |B| > \lfloor d/2 \rfloor. \end{cases}$$

Moreover, it holds $\mathrm{tr}(T') = \mathrm{tr}(T)$, where T is defined in Proposition 1.

(ii) The number of evaluations of f needed to compute the estimator (D.2) is equivalent to $n2^{d-1}$, asymptotically as d goes to infinity.

Proof. The proof of the first statement is a direct application of the central limit theorem. To show the "moreover" part, put $Z(A) = \frac{1}{2}(f(X) - f(X^{\setminus A}))^2$, $Z'(A) = \frac{1}{2}(f(X') - f(X^{\setminus A^c}))^2$ and

$$\widetilde{Z}(A) = \begin{cases} Z(A) & \text{if } |A| \leq \lfloor d/2 \rfloor, \\ Z'(A) & \text{otherwise,} \end{cases}$$

for every $A \subset D$. With this notation, $T(A,B) = \mathrm{Cov}(Z(A), Z(B))$ and $T'(A,B) = \mathrm{Cov}(\widetilde{Z}(A), \widetilde{Z}(B))$. If $|A| \leq \lfloor d/2 \rfloor$ and $|B| \leq \lfloor d/2 \rfloor$ then obviously $T(A,B) = T'(A,B)$. If $|A| > \lfloor d/2 \rfloor$ and $|B| > \lfloor d/2 \rfloor$ then the equality is also true because X and X' play the same role.

To show the second statement, notice that the outputs needed in the top row of (D.2) are precisely those computed in (D.1). The outputs needed in the bottom of (D.2) also have been computed in (D.1), since $|A| \geq \lfloor d/2 \rfloor + 1$ implies $|A^c| \leq d - \lfloor d/2 \rfloor - 1 \leq \lfloor d/2 \rfloor$. But the number of simulations in (D.1) is clearly 2^{d-1}, which has to multiplied by n to get the final number of evaluations.

□

According to Proposition D.2, the estimator $\widehat{\tau}'$ is as efficient as $\widehat{\tau}$ but requires only half the number of function evaluations of $\widehat{\tau}$.

E Further details for the fast and extended fast Möbius transform algorithms

E.1 Implementation based on "bit-shift" manipulations

Let us introduce some notation first. If σ^* denotes a vector of size 2^d then let $\sigma^*(a_{i,1}\cdots a_{i,d})$ denote its component corresponding to the subset of D represented by the Boolean vector $(a_{i,1}\cdots a_{i,d})$. The components of σ^* are assumed to be arranged in the lexicographical order of the Boolean vectors $(a_{i,1}\cdots a_{i,d})$, as in Table 1.

To compute $\sigma^{*(d)}$ in (22), proceed as follows:

for each $j = 1, \ldots, d$
  for each $i = 1, \ldots, 2^d$
    if $a_{i,d-j+1} = 1$
      $\sigma^{*(j)}(a_{i,1}\cdots a_{i,d}) \leftarrow \sigma^{*(j-1)}(a_{i,1}\cdots a_{i,d-j}, 1, a_{i,d-j+2}\cdots a_{i,d})$
      $\qquad -\sigma^{*(j-1)}(a_{i,1}\cdots a_{i,d-j}, 0, a_{i,d-j+2}\cdots a_{i,d})$
    else
      $\sigma^{*(j)}(a_{i,1}\cdots a_{i,d}) \leftarrow \sigma^{*(j-1)}(a_{i,1}\cdots a_{i,d})$.

This implementation saves more memory as storing the matrices $(I_{2^{d-j}} \otimes M_1^* \otimes I_{2^{j-1}})$ becomes unnecessary. A flow diagram with $d = 3$ is depicted in Figure 1 of this supplementary file.

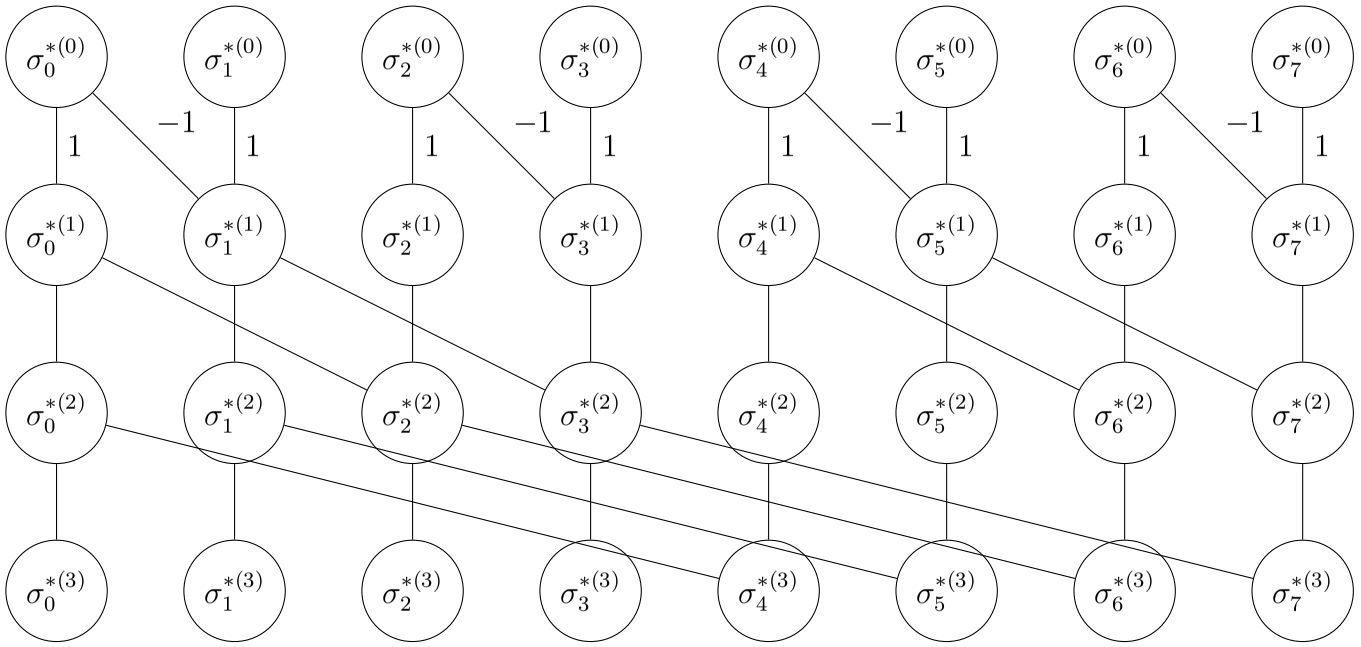

Figure 1: Flow diagram of the algorithm (22) with $d = 3$ and M_1^* as in (20). Here $\sigma_i^{*(j)}$ is the ith element of the vector $\sigma^{*(j)}$ and hence coincides with $\sigma^{*(j)}(a_{i,1} \cdots a_{i,d})$ in the main text. Although not shown for improved readability, all vertical and diagonal lines have weights 1 and -1, respectively.

To compute $\Sigma^{*(d)}$, proceed in two steps:

S1. For each $j = 1, \ldots, d$
for each $i = 1, \ldots, 2^d$
if $a_{i,d-j+1} = 1$
$$\text{row}^{(j)}(a_{i,1} \cdots a_{i,d}) \leftarrow \text{row}^{(j-1)}(a_{i,1} \cdots a_{i,d-j} 1 a_{i,d-j+2} \cdots a_{i,d}) - \text{row}^{(j-1)}(a_{i,1} \cdots a_{i,d-j} 0 a_{i,d-j+2} \cdots a_{i,d})$$
else
$$\text{row}^{(j)}(a_{i,1} \cdots a_{i,d}) \leftarrow \text{row}^{(j-1)}(a_{i,1} \cdots a_{i,d});$$
here $\text{row}^{(j)}(a_{i,1} \cdots a_{i,d})$ stands for the row of $\Sigma^{*(j)}$ that corresponds to the index $(a_{i,1} \cdots a_{i,d})$.

S2. Repeat S1 above but with "col" in place of "row", where $\text{col}^{(j)}(a_{i,1} \cdots a_{i,d})$ stands for the column of $\Sigma^{*(j)}$ that corresponds to the index $(a_{i,1} \cdots a_{i,d})$.

The above algorithm coincides with (24) when M_1^* is the matrix (20). As above, this implementation saves memory usage as there is no need to store the matrices in (24) anymore.

E.2 Computation times

To test in practice the fast Möbius and the extended fast Möbius transform algorithms with bit-shift implementation, and to compare them with straightforward matrix multiplications, a version was coded in MATLAB (2023b). Computational times were monitored, and the results are depicted in Figure 2 below. These times are computed from random datasets with increasing d; they encompass the computation time of Sobol indices σ^* from total indices τ^* (left panels) and the additional computation time of the corresponding $2^d \times 2^d$ covariance matrices Σ^* from T^* (right panels).

The top panels show clear gains of the fast Möbius algorithm in terms of computation time. These gains have to be put into perspective, however. They remain modest for the computation of σ^* (mere seconds, left panel). They are higher when including the computation of covariance matrices (right panel), with a gain of around 100 seconds for $d = 15$. Nevertheless, going beyond about $d = 15$ becomes excessively expensive because of the manipulation of enormous $2^d \times 2^d$ matrices (in practice, considering $d = 16$ involves manipulating > 30 Gigabyte matrices). In any case, it is important to note that, all in all, most of the computational expense of a global sensitivity

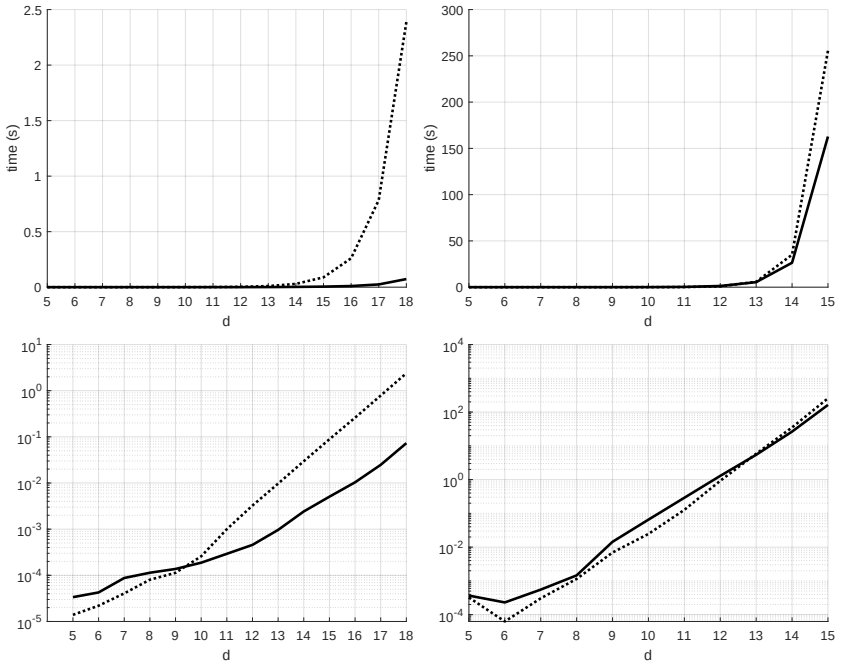

Figure 2: Comparison of execution times of the fast and extended fast Möbius transform algorithms (plain lines) and straightforward sparse matrix multiplications (dotted lines). Left panels: computation times for σ^*; right panels: computation times for σ^* and Σ^*. Top row: linear scale; Bottom row: logarithmic scale.

analysis is likely to come from the generation of the dataset of the model outputs.

F Generation of datasets for the estimation of sensitivity indices

Cell fate Boolean model of Section 5.1. The reduced version of 11 variables $V_1, \ldots, V_{11}$ presented in Calzone et al. (2010) was used. This model is taken as the reference, where each Boolean variable has the same relative speed. In the reference model, the probabilities of all transitions involving the activation of variable V_j (*i.e.* V_j going from 0 to 1) are multiplied by some weight w_j^+, and the probabilities of all transitions involving the deactivation of variable V_j (*i.e.* V_j going from 1 to 0) are multiplied by some weight w_j^-. Each row of the probability matrix is then divided by its sum, ensuring the process is still a Markov chain. In the global sensitivity analysis, the inputs are the weights w_j^+ and w_j^- for six chosen variables (C8, RIP1, NFkB, cIAP, MOMP and MPT), leading to the 12 inputs listed in Table 1. Since there is no a priori information about the relative speed of the regulatory processes involved in the cell, all weights are drawn independently. Each input is drawn according to the log-uniform distribution $\log \mathcal{U}(10^{-0.5}, 10^{0.5})$. Thus, for each $i = 1, \ldots, n$ and $j = 1, \ldots, d$, we independently draw $X_j^{(i)} \sim \log \mathcal{U}(10^{-0.5}, 10^{0.5})$, meaning that $\log X_j^{(i)} \sim \mathcal{U}(\log 10^{-0.5}, \log 10^{0.5})$.

The last step of the simulation consists in the execution of the updated Markov chain from a given initial condition, until it reaches one of the three steady states: apoptosis, necrosis or survival. The chosen initial condition is the same as in Calzone et al. (2010): all variables at 0 except for ATP, cIAP and TNF at 1.

ODE model of Section 5.2. The inputs and their distributions are listed in Table 2. Here again the inputs are drawn independently, since we do not have any a priori information.

We performed $n = 5000$ simulations; among them 13 presented at least one numerical anomaly (for instance the explosion of one variable) and were taken out, leaving a sample of size $n = 4987$. The simulations were carried out with the routine `ode45` of MATLAB. To determine

Input	Description: relative probability of
$X_1 = w_1^+$ (C8+)	activation of caspase 8
$X_2 = w_1^-$ (C8−)	deactivation of caspase 8
$X_3 = w_2^+$ (RIP1+)	activation of RIP1
$X_4 = w_2^-$ (RIP1−)	deactivation of RIP1
$X_5 = w_3^+$ (NFkB+)	activation of NFkB
$X_6 = w_3^-$ (NFkB−)	deactivation of NFkB
$X_7 = w_4^+$ (cIAP+)	activation of cIAP
$X_8 = w_4^-$ (cIAP−)	deactivation of cIAP
$X_9 = w_5^+$ (MOMP+)	activation of MOMP
$X_{10} = w_5^-$ (MOMP−)	deactivation of MOMP
$X_{11} = w_6^+$ (MPT+)	activation of MPT
$X_{12} = w_6^-$ (MPT−)	deactivation of MPT

Table 1: Description of the inputs of the Boolean network (part 5.1). Each input is drawn according to the log-uniform distribution $\log \mathcal{U}(10^{-0.5}, 10^{0.5})$.

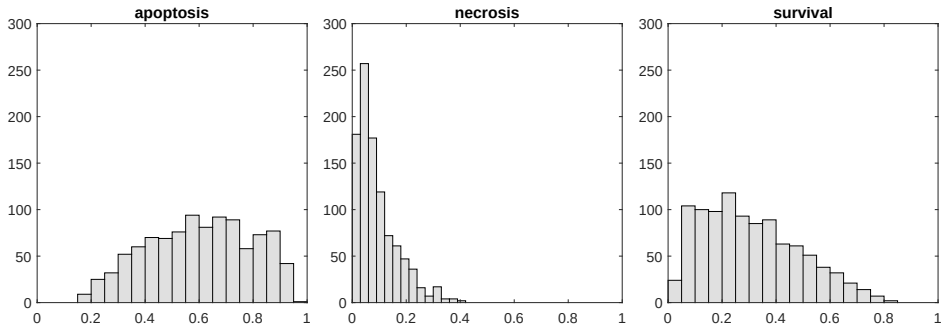

Figure 3: Histograms of 1000 independent model simulations.

when a steady state had been reached, we tested whether $\Delta N/N$ was below the threshold value 10^{-10} (recall that N denotes total bacterial population: $N = L + S + S^l$). The last step consisted in extracting the logarithms of steady state populations, expressed in CFU *colony forming unit*. If Y^* designates one of the five populations in steady state, we applied the following transformation:

$$Y = \begin{cases} \log_{10}(Y^*) & \text{if } Y^* > 1, \\ 0 & \text{if } Y^* \leq 1, \end{cases}$$

in order to obtain populations on a logarithmic scale. Because of the logarithmic transformation, very small populations ($Y^* \leq 1$ CFU) were truncated in order to ignore unwanted effects (such low population counts are below detection thresholds anyway).

G Proofs

Proof of Proposition 2

We only show the second statement of the proposition. Let $A_1, \ldots, A_{2^d-1}$ be some enumeration of the elements of $\mathbf{2}^D \setminus \emptyset$. Since $\operatorname{Var} f(X) = \sum_{i=1}^{2^d-1} \sigma^*(A_i)$, the null hypothesis is equivalent to "$\mathrm{H}_0 : \sum_{i=1}^{2^d-1} c_i \sigma^*(A_i) \leq 0$", which is further rewritten as "$\mathrm{H}_0 : c^\top \sigma^* \leq 0$". Because of Proposition 1, it is clear that under H_0, it holds that $\lim_{n\to\infty} P_{\mathrm{H}_0}(S_n > r) \leq \lim_{n\to\infty} P(N > q_{1-\alpha}) \leq \alpha$, where here N is a standard normal random variable.

Proof of Proposition 3

We show that $M^* = \otimes^d M_1^*$. Let ω be the inverse of the one-to-one map that with each $A \in \mathbf{2}^D$ associates $\sum_{i=1}^d b_i \, 2^{d-i} \in \{0, \ldots, 2^d - 1\}$, where $b_i = 1$ if $i \in A$ and $b_i = 0$ otherwise. (Note that ω depends on d. For instance, for $d = 2$, $\omega(3) = \{1, 2\}$ but $\omega(3) = \{2, 3\}$ for $d = 3$.) The arrangement of the components of τ^* described at the beginning of Section 4 implies that

$$\tau_i^* = \tau^*(\omega(i)),$$

for all $i = 0, \ldots, 2^d - 1$, where in the left-hand side τ_i^* denotes the ith component (starting at zero) of the dual total index vector and in the right-hand side $\tau^*(\omega(i))$ denotes the dual total index of the set $\omega(i)$. The same holds for σ^*.

We show that M^* in (14) coincides with $\otimes^d M_1^*$ where M_1^* is as in (20). In this proof, d, and hence ω, since it depends on d, are fixed. A few results are collected in the following lemma.

Lemma 1. *Let $1 \le n \le d$ and $i, j \in \{0, \dots, 2^{n+1}-1\}$. If $0 \le i \le 2^n - 1$ and $2^n \le j \le 2^{n+1} - 1$ then the following statements are true:*

(i) $\omega(i) \subset \omega(j)$ if and only if $\omega(i) \subset \omega(j - 2^n)$.

(ii) $|\omega(j - 2^n) \setminus \omega(i)|$ is even if and only if $|\omega(j) \setminus \omega(i)|$ is odd.

(iii) $\omega(i) \not\subset \omega(j)$

We need to show that

$$(\otimes^d M_1^*)_{ij} = \begin{cases} -1 & \text{if } \omega(j) \subset \omega(i) \text{ and } |\omega(i) \setminus \omega(j)| \text{ is odd,} \\ 1 & \text{if } \omega(j) \subset \omega(i) \text{ and } |\omega(i) \setminus \omega(j)| \text{ is even,} \\ 0 & \text{otherwise.} \end{cases} \tag{G.1}$$

Let us show a slightly more general result. Let M_n^*, $n = 1, \dots, d$, be a finite sequence of matrices of increasing size defined by

$$M_1^* = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}, \qquad M_n^* = \underbrace{M_1^* \otimes \cdots \otimes M_1^*}_{n \text{ times}} \quad (n = 1, \dots, d).$$

Let us show that M_n^* satisfies (G.1) for every $i, j \in \{0, \dots 2^n - 1\}$ and all $1 \le n \le d$. This will show in particular that $M_d^* = M^*$. The proof is by mathematical induction. If $n = 1$ then it is clear that (G.1) holds because $\omega(0) = \emptyset$ and $\omega(1) = \{d\}$. Let $n \ge 1$ and suppose that M_n^* satisfies (G.1) for every $i, j \in \{0, \dots 2^n - 1\}$. Let us show that M_{n+1}^* satisfies (G.1) for every $i, j \in \{0, \dots 2^{n+1} - 1\}$. By definition,

$$M_{n+1}^* = M_1^* \otimes M_n^* = \begin{pmatrix} M_n^* & 0 \\ -M_n^* & M_n^* \end{pmatrix}. \tag{G.2}$$

Case $i, j \in \{0, \dots, 2^n - 1\}$: We have that $(M_{n+1}^*)_{i,j}$ is equal to $(M_n^*)_{i,j}$, which satisfies (G.1) by assumption.

Case $i \notin \{0, \dots, 2^n - 1\}$ and $j \in \{0, \dots, 2^n - 1\}$: We have $(M_{n+1}^*)_{i,j} = (-M_n^*)_{i-2^n,j}$. By assumption,

$$(-M_n^*)_{i-2^n,j} = \begin{cases} 1 & \text{if } \omega(j) \subset \omega(i - 2^n) \text{ and } |\omega(i - 2^n) \setminus \omega(j)| \text{ is odd,} \\ -1 & \text{if } \omega(j) \subset \omega(i - 2^n) \text{ and } |\omega(i - 2^n) \setminus \omega(j)| \text{ is even,} \\ 0 & \text{otherwise.} \end{cases}$$

Notice that, compared to (G.1), "1" and "-1" have been interchanged. By Lemma 1, it holds that $\omega(j) \subset \omega(i)$ is equivalent to $\omega(j) \subset \omega(i - 2^n)$. Thus, it remains to show that $|\omega(i - 2^n) \setminus \omega(j)|$ is even if and only if $|\omega(i) \setminus \omega(j)|$ is odd. But again this is true from Lemma 1.

Case $i \notin \{0, \ldots, 2^n - 1\}$ and $j \notin \{0, \ldots, 2^n - 1\}$: Here $(M^*_{n+1})_{i,j} = (M^*_n)_{i-2^n, j-2^n}$. That $(M^*_n)_{i-2^n, j-2^n}$ satisfies (G.1) follows from the same considerations as in the previous case.

Case $i \in \{0, \ldots, 2^n - 1\}$ and $j \notin \{0, \ldots, 2^n - 1\}$: We know from (G.2) that $(M^*_{n+1})_{i,j} = 0$. To show that (G.1) is satisfied, it suffices to see that $\omega(j) \not\subset \omega(i)$. But this is true from Lemma 1, since $j > i$.

Proof Proposition 4

We show that $\sigma^{*(d)} = (\otimes^d M_1^*)\sigma^{*(0)}$.

We shall show that, in general,

$$\sigma^{*(j)} = (I_{2^{d-j}} \otimes (\otimes^j M_1^*))\sigma^{*(0)}, \tag{G.3}$$

for all $j = 1, \ldots, d$. The proof is by mathematical induction. Equation (G.3) is true for $j = 1$ because of (22). Suppose that it is true for a given j and let us show it is true for $j + 1$. From (22) and (G.3), we have

$$\begin{aligned}
\sigma^{*(j+1)} &= (I_{2^{d-j-1}} \otimes M_1^* \otimes I_{2^j})\, \sigma^{*(j)} \\
&= ((I_{2^{d-j-1}} \otimes M_1^*) \otimes I_{2^j}) \left(I_{2^{d-j}} \otimes (\otimes^j M_1^*)\right) \sigma^{*(0)} \\
&= ((I_{2^{d-j-1}} \otimes M_1^*) I_{2^{d-j}}) \otimes \left(I_{2^j} (\otimes^j M_1^*)\right) \sigma^{*(0)} \\
&= (I_{2^{d-j-1}} \otimes M_1^*) \otimes \left(\otimes^j M_1^*\right) \sigma^{*(0)} \\
&= (I_{2^{d-j-1}} \otimes (\otimes^{j+1} M_1^*))\sigma^{*(0)}.
\end{aligned}$$

To show that the computation cost is $\Theta(d2^d)$, notice that each matrix $I_{2^{d-j}} \otimes M_1^* \otimes I_{2^{j-1}}$ is a block-diagonal matrix of the form

$$\begin{bmatrix} M_1^* \otimes I_{2^{j-1}} & & \\ & \ddots & \\ & & M_1^* \otimes I_{2^{j-1}} \end{bmatrix}.$$

Let m_{ij} denote the element at the ith row and jth column of M_1^* ($i, j \in \{1, 2\}$). There are 2^{d-j} block-rows of the form

$$M_1^* \otimes I_{2^{j-1}} = \begin{bmatrix} m_{11} I_{2^{j-1}} & m_{12} I_{2^{j-1}} \\ m_{21} I_{2^{j-1}} & m_{22} I_{2^{j-1}} \end{bmatrix},$$

and, if multiplied by a column vector to the right, each of them leads to $\Theta(\text{nnz}(M_1^* \otimes I_{2^{j-1}})) = \Theta(\text{nnz}(M_1^*)2^{j-1}) = \Theta(2^{j-1})$ arithmetic operations, yielding $2^{d-j}\Theta(2^{j-1}) = \Theta(2^{d-1}) = \Theta(2^d)$ operations per update. Since there are d updates, the proof is complete.

Input (unit)	Description	Distribution
d (h^{-1})	dilution rate of the mouse gut	$\mathcal{U}(0.1, 0.4)$
r (h^{-1})	maximal growth rate of bacteria	$\mathcal{U}(0.2, 1.5)$
k (CFU/g)	carrying capacity of bacteria	$\log \mathcal{U}(3.5\,10^9, 10^{10})$
x (h^{-1})	lysis induction rate	$\log \mathcal{U}(10^{-4}, 0.1)$
l (h^{-1})	lytic cells' mortality rate	$\mathcal{U}(0.2, 1.5)$
y $(\emptyset)$	burst size	$\mathcal{U}(1, 50)$
a $([\text{PFU}/g]^{-1}h^{-1})$	adsorption constant	$\log \mathcal{U}(10^{-10}, 10^{-7})$
g $(\emptyset)$	probability of lysogeny	$\log \mathcal{U}(10^{-4}, 0.7)$
L_0 (CFU/g)	initial lysogens	$\log \mathcal{U}(10^5, 10^7)$
S_0 (CFU/g)	initial susceptible	$\log \mathcal{U}(10^5, 10^7)$

Table 2: Description of the inputs of the ODE system (part 5.2).