# Hadamard-Walsh Transform: A Method for Detecting Non-Linearity in Binary Encoded Datasets

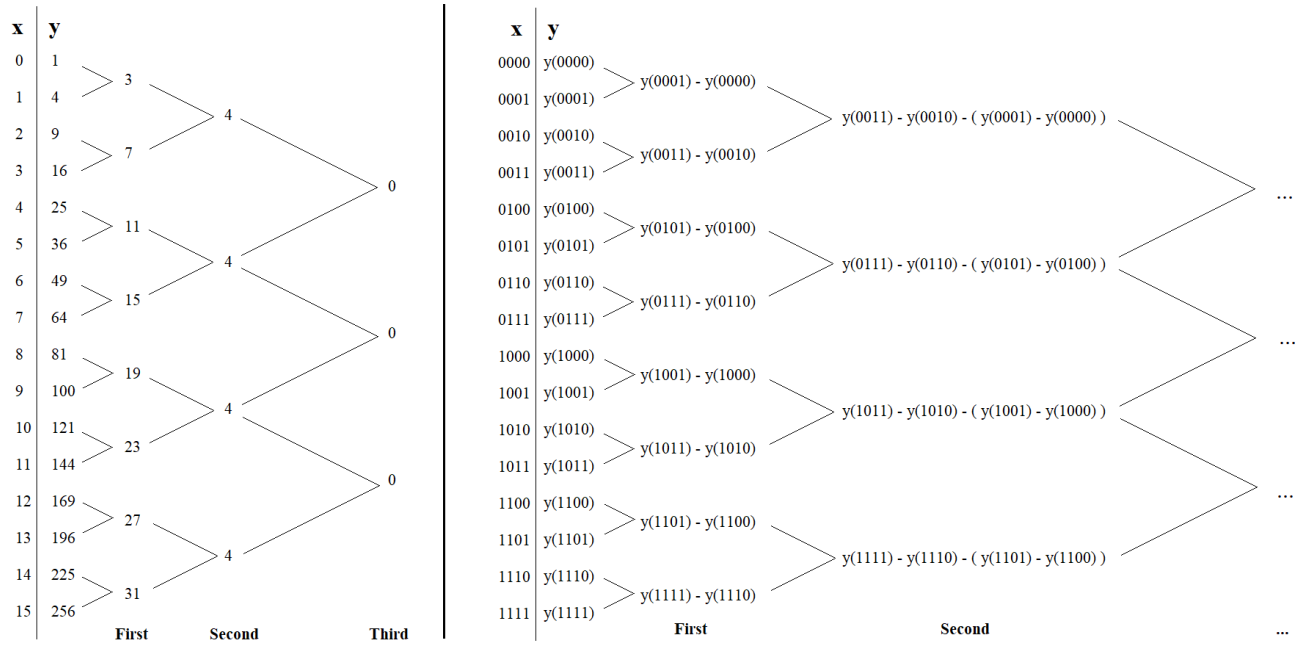
Miles Miller-Dickson, Victor A. Meszaros, C. Brandon Ogbunugafor

Department of Ecology and Evolutionary Biology, Brown University

## 1 Introduction

A popular means of measuring non-linearity between binary parcels is through the Hadamard-Walsh transform. The idea is that you collect the fitness (or any phenotypic measure) values into a vector and then reformat it into a Hadamard matrix (and is then scaled by a diagonal matrix). The output is a collection of coefficients which measure the degree to which the map is linear, of second order, third, and so forth. We have developed our own motivation for the transform (taken as an analogy to functions on real numbers) which I will outline briefly below.

Consider the function on real numbers, $y = 1 + 2x + x^2 = (1 + x)^2$. This is a generic quadratic function. Below, we take first order differences between the outputs. Then, we take second order differences (differences of first order differences). Lastly, we take third order differences (differences of second order differences). The figure below shows this graphically:




The left panel shows a table of input values, $x$, and output values, $y$, for the quadratic function above. The column labelled "First" is the collection of differences between the outputs, $y$. The "Second" column is the same but for differences between the first order differences, and so on. Notice that the "Third" column is zero. This is a generic property of quadratic functions (nothing special about the function I chose). If we had picked a cubic function, then the column with zeros would have been the "Fourth" column, and so on. Clearly, higher order differences are zero as well. So the columns (more precisely, the averages of the column values) give a sense for the order of the function. Loosely, it is the extent to which a given order dominates in the function. In particular, the coefficient of each polynomial term $(1, x, x^2, ...)$ contributes to the $nth$ order column: the larger

the coefficient the larger the contribution (in magnitude). These coefficients are analogous to the regression coefficients that one may use to describe the phenotype function, $y$.

The right panel is the same idea but for our genotype to phenotype function $y$. It turns out that this is how the Hadamard transform works. Each output from the Hadamard matrix is a sum of one of these columns, and then one generally divides out by a power of two so that the output can be interpreted as an average (this is given by multiplication by the diagonal matrix from earlier). More precisely, we have,

$$E = VHy$$

where $y$ is our vector of fitness values (representing the phenotype), $H$ is a Hadamard matrix, and $V$ is a diagonal matrix with the scaling coefficients to interpret the outputs, $E$, as averages. For instance, $E_{0000}$, the first element of $E$, is simply the average of the fitness values, $y$ (one may refer to this as a *zeroth* order). $E_{0001}$ is the sum of what's labelled the "First" column (scaled by $1/8$ from the diagonal matrix). $E_{0001}$ can be interpreted as an average over all configurations (of $0's$ and $1's$) in the slots in 0001 which are zero (i.e. the first three). This is why there are 8 of them. Similarly, $E_{0010}, E_{0100}$, and $E_{1000}$ are averages of first order differences over all configurations of the slots with zeros. $E_{0011}$ is the average of the column labelled "Second". In this case, it is an average of second order differences, averaged over all configurations that the zeros can take on (i.e. $00, 01, 10$, and $11$). $E_{0101}, E_{0110}, E_{1010}, E_{1001}$, and $E_{1100}$ are the same, but averaged over the configurations that their zeros can take on. The pattern continues for higher order terms. $E_{1111}$ is the $4th$ order term since there are no zeros for to average over.

Rather than speaking in terms of *linearity* or otherwise, we can also interpret the Hadamard-Walsh transform in terms of an average *effect* of the presence of mutations. For instance, $E_{001}$ is explicitly given by,

$$E_{001} = \frac{1}{4}[\,(y_{111} - y_{110}) + (y_{101} - y_{100}) + (y_{011} - y_{010}) + (y_{001} - y_{000})\,].$$

If $y_{111}$ represents the effect of having all three mutations present and $y_{110}$ is the effect of just the first two, then $(y_{111} - y_{110})$ represents the effect of a mutation in the third locus when the first two mutations are present. The same interpretation can be applied to the other three differences but with the presence of the mutations $10, 01$, and $00$ in the first two loci. Since we divide out by 4, this represents an average effect of a mutation in the third locus (averaged over all other relevant possibilities). Similarly, the explicit form of $E_{011}$ is,

$$E_{011} = \frac{1}{2}[\,(\,(y_{111} - y_{110}) - (y_{101} - y_{100})\,) + (\,(y_{011} - y_{010}) - (y_{001} - y_{000})\,)\,].$$

In this case, $(y_{111} - y_{110})$ is the effect of a mutation in the third locus when mutations are present in the second (there's a mutation in the first but that's not the important part since the first index is the one being averaged over here). $(y_{101} - y_{100})$ is the same except for when the the second locus is not mutated. Under this grouping, the difference between these two, $(y_{111} - y_{110}) - (y_{101} - y_{100})$, can be interpreted as how much more (or less) effective the third mutation is when there is a mutation in the second locus (all while the first locus is mutated). The same interpretation can be made for the second collection but when the first locus is not mutated. Then, summing and dividing by 2 gives it the interpretation of the average effect of the third mutation in the presence of the second. But there should be nothing special about considering the effect of the third mutation in the presence of the second. Notice that $y_{110}$ and $y_{101}$ can be swapped and it won't change $E_{011}$. So we could have just as well considered $(y_{111} - y_{101}) - (y_{110} - y_{100})$ and interpreted it as how much more effective the presence of the second mutation is when there is a mutation in the third.

Generally speaking, $E_{011}$ is symmetric about the interchange of the second and third index, and we would want this property to hold since $E_{011}$—if it is to be interpreted as an average effect of mutations in the second and third loci (averaged over all cases for the 0)—should not distinguish between these two loci, given that the state of their mutation is the same. Similarly, $E_{001}$, for instance, should have a symmetry between the first and second loci since the state of mutation for these two loci is the same (namely, both non-mutated), and one can easily check this.