1 Calculating the variance of $\epsilon$

The set $\mathcal{O}$ is a mixture of three types of TISs $T$, $F_u$ and $F_d$, we have

$$\hat{W}_i^{(O)} = \alpha_T \hat{W}_i^{(T)} + \alpha_{F_u} \hat{W}_i^{(F_u)} + \alpha_{F_d} \hat{W}_i^{(F_d)},$$  \hspace{1cm} (1)$$

where the superscript $(O)$ refers to PWMs obtained from the set $\mathcal{O}$ and three PWMs $\hat{W}_i^{(T)}$, $\hat{W}_i^{(F_u)}$ and $\hat{W}_i^{(F_d)}$ are virtually calculated from the three types of TISs in the set $\mathcal{O}$. Since we don’t know these three PWMs, we use three other PWMs obtained from the set $\mathcal{I}$ to replace them, and an error $\epsilon$ is generated

$$\hat{W}_i^{(O)} = \alpha_T \hat{W}_i^{(I)} + \alpha_{F_u} \hat{W}_i^{(I)} + \alpha_{F_d} \hat{W}_i^{(I)} + \epsilon,$$  \hspace{1cm} (2)$$

Here we arrange the $4 \times (l + r)$ matrices in row order to $4(l + r)$-dimension vectors. Consequently, $W_j(\mu)$ (in main text) becomes $W(4(j - 1) + \mu), j = 1, 2, ..., l + r$ and $\mu = 1, 2, 3, 4$. Thus the error term $\epsilon$ can be explicitly written as

$$\epsilon = \sum_{i=1}^{3} \alpha_i (\hat{W}_i^{(O)} - \hat{W}_i^{(I)}),$$  \hspace{1cm} (3)$$

where the index $i=1,2,3$ refer to the three sets $T$, $F_u$ and $F_d$, respectively. The “homogeneity assumption” (see the paper) says that $\hat{W}_i^{(O)}$ and $\hat{W}_i^{(I)}$ are independent finite-sample estimations of the same PWM $W_i$. Therefore,

$$E(\epsilon) = \sum_i \alpha_i (E(\hat{W}_i^{(O)}) - E(\hat{W}_i^{(I)})) = \sum_i \alpha_i (W_i - W_i) = 0,$$  \hspace{1cm} (4)$$

and the variance of $\epsilon$ can be written as

$$Var(\epsilon) = \sum_i \alpha_i^2 (Var(\hat{W}_i^{(O)}) + Var(\hat{W}_i^{(I)})).$$  \hspace{1cm} (5)$$

We further assume that the nucleotide frequencies at different positions in the PWM are independent (Staden, R. (1984) Computer methods to locate signals in nucleic acid sequences. Nucleic Acids Res, 12 :505-519). Thus, for position $j$ and position $k$ (where $j, k=1, 2, \ldots, l + r$), we have

$$Cov(\hat{W}_i^{(O)}(4(j - 1) + \mu), \hat{W}_i^{(O)}(4(k - 1) + \nu)) = W_i(4(j - 1) + \mu) \delta_{\mu,\nu} W_i(4(k - 1) + \nu) \Omega^{O} \delta_{j,k}$$  \hspace{1cm} (6)$$
and

\[ \text{Cov}(\hat{W}_i^{(0)}(4(j - 1) + \mu), \hat{W}_i^{(0)}(4(k - 1) + \nu)) = W_i(4(j - 1) + \mu)\delta_{\mu,\nu} - W_i(4(j - 1) + \mu)W_i(4(k - 1) + \nu)\delta_{j,k}, \]  

where \( \mu, \nu=1, 2, 3, 4, \) denoting nucleotide A, C, G, T, respectively. This yields

\[
\text{Var}(\hat{W}_i^{(0)}) = \frac{1}{\alpha_i\Omega_i}\sum_i
\]  

and

\[
\text{Var}(\hat{W}_i^{(0)}) = \frac{1}{\Omega_i}\sum_i
\]

where \( \Sigma_i \) is a block diagonal symmetric matrix and the number of blocks is determined by the number of positions of the PWM alignment. An block according to position \( j \) is shown as below:

\[
\begin{pmatrix}
W_i(4j - 3) - W_i^2(4j - 3) & -W_i(4j - 3)W_i(4j - 2) & -W_i(4j - 3)W_i(4j - 1) & -W_i(4j - 3)W_i(4j) \\
W_i(4j - 2) - W_i^2(4j - 2) & -W_i(4j - 2)W_i(4j - 1) & -W_i(4j - 2)W_i(4j) \\
W_i(4j - 1) - W_i^2(4j - 1) & -W_i(4j - 1)W_i(4j) \\
W_i(4j) & W_i^2(4j)
\end{pmatrix}
\]

With Eq. 5, Eq. 8 and Eq. 9, we finally obtain the variance of \( \varepsilon \)

\[
\text{Var}(\varepsilon) = \sum_i \left( \frac{\alpha_i^2}{\Omega_i} + \frac{\alpha_i}{\Omega_i}\Omega_i \right)\Sigma_i
\]


\[
\begin{cases}
V(3j - 2) = W(4j - 3) + W(4j - 2) - W(4j - 1) - W(4j) \\
V(3j - 1) = W(4j - 3) - W(4j - 2) + W(4j - 1) - W(4j) \\
\end{cases}
\]

Consequently,

\[
\hat{\nu} = \sum_i \alpha_i\hat{V}_i + \varepsilon'.
\]

The variance of \( \varepsilon' \) has a similar form with \( \varepsilon \)

\[
\text{Var}(\varepsilon') = \sum_i \left( \frac{\alpha_i^2}{\Omega_i} + \frac{\alpha_i}{\Omega_i}\Omega_i \right)\Sigma_i',
\]
where $\Sigma'_i = H \Sigma_i H^T$ and $H$ is a block diagonal matrix with each block being

$$
\begin{pmatrix}
1 & 1 & -1 & -1 \\
1 & -1 & 1 & -1 \\
1 & -1 & -1 & 1
\end{pmatrix}.
$$

$\hat{W}_i$ is used as an estimate of $W_i$ to calculate $\Sigma'_i$. The effect of this approximation is of high order in our model if the samples are sufficient, for example over 50. In the following part, $Var(\varepsilon')$ will be denoted by $\Sigma'$ for convenience.

2 Minimizing the weighted sum of squared errors $\varepsilon'^T \Sigma' \varepsilon'$

In the main text we mentioned that $\alpha$ is estimated by minimizing the weighted sum of square errors

$$
f(\alpha_1, \alpha_2, \alpha_3) = (\hat{V}_O - \sum_{i=1}^{3} \alpha_i \hat{V}_i)^T \Sigma' (\hat{V}_O - \sum_{i=1}^{3} \alpha_i \hat{V}_i) \quad (13)
$$

, where the index $i=1,2,3$ refer to the three sets $T$, $F_u$ and $F_d$ respectively.

Substitute $\alpha_3$ with

$$
\alpha_3 = 1 - \frac{1}{2} \sum_{i=1}^{2} \alpha_i \quad (14)
$$

and Eq. (13) can be written as

$$
f(\alpha_1, \alpha_2) = (S - \sum_{i=1}^{2} \alpha_i T_i)^T \Sigma' (S - \sum_{i=1}^{2} \alpha_i T_i) \quad (15)
$$

where

$$
S = \hat{V}_O - \hat{V}_3 \quad (16)
$$

and

$$
T_i = \hat{V}_i - \hat{V}_3, \; i = 1, 2 \quad (17)
$$

To minimize $f(\alpha_1, \alpha_2)$, we let the partial derivatives be zero

$$
\frac{\partial f}{\partial \alpha_j} = -2 T_j^T \Sigma' (S - \sum_{i=1}^{2} \alpha_i T_i) - (S - \sum_{i=1}^{2} \alpha_i T_i)^T \Sigma' \frac{\partial \Sigma'}{\partial \alpha_j} (S - \sum_{i=1}^{2} \alpha_i T_i) = 0 \quad (18)
$$

Eq. (18) can be simplified to

$$
\sum_{i=1}^{2} K_{ij} \alpha_i = L_j, \; j = 1, 2 \quad (19)
$$

where

$$
K_{ij} = T_j^T \Sigma' T_i, \; i, j = 1, 2 \quad (20)
$$


\[ L_j = T_j^T \Sigma' S + \frac{1}{2} (S - \sum_{i=1}^{2} \alpha_i T_i)^T \Sigma' \frac{\partial \Sigma'}{\partial \alpha_j} \Sigma' (S - \sum_{i=1}^{2} \alpha_i T_i). \quad j = 1, 2 \]  

(21)

There are 2 equations and 2 variables. The equations can be solved iteratively. First we set

\[ \alpha_i^{(0)} = 1/3, \quad i = 1, 2 \]  

(22)

Then we calculate \( \Sigma^{(0)} \) and \( \partial \Sigma^{(0)}/\partial \alpha_j \) by Eq. 12, \( K_{ij}^{(0)} \) and \( L_j^{(0)} \) by Eq. 20 and Eq. 21, and then obtain \( \alpha^{(1)} \) by solving

\[ \sum_{i=1}^{2} K_{ij}^{(0)} \alpha_i^{(1)} = L_j^{(0)}, \quad j = 1, 2 \]  

(23)

Then \( \alpha_i^{(1)} \) is used to calculate \( \alpha_i^{(2)} \) and the process is repeated until \( \sum_{i=1}^{2} |\alpha_i^{(n)} - \alpha_i^{(n-1)}| < 10^{-6} \).

It’s difficult to prove that this iteration process will always converge, but in practice it converges quite fast. For instance, when to estimate the accuracy of RefSeq annotation for \( E. coli \) K12, the algorithm converge in less than 10 steps (Fig. 1); we also show the iteration process of another 12 randomly selected genomes in Fig. 2.

[Fig. 1 about here.]

[Fig. 2 about here.]
Fig. 1. The convergency of the optimization algorithm (shown on *E. coli* K12).
Fig. 2. The convergency of the optimization algorithm (shown on 12 randomly selected genomes).