1 Confluence metric

1.1 Confluence for 2D slices

The input required for the proposed algorithm is a WMH segmentation image where each voxel's value is its probability of belonging to a WMH $p_{WMH} \in [0,1]$ or, in the case of binarised segmentations, a value of 0 or 1.

We propose the following function as a method for quantifying confluence of WMH:

For every 2D image slice $A \in [0,1]^{d\times d}$, i.e.

$$A = \begin{pmatrix} a_{11} & \dots & a_{1d} \\ \vdots & \ddots & \vdots \\ a_{d1} & \dots & a_{dd} \end{pmatrix}, \qquad \text{with each } a_{ij} \in [0,1],$$

we define the *confluence* as

$$\mathrm{Conf}(A) := \frac{1}{C\,\mathrm{vol}(A)} \sum_{i,j,m,n=1}^{d} a_{ij}\, a_{mn}\, k\,(i-m, j-n) \quad \in \ [0,1], \tag{1}$$

with kernel

$$k\,(i-m, j-n) = e^{-s((i-m)^2+(j-n)^2)} \in (0,1]$$

for some parameter $s > 0$, and *volume* of A given by

$$\mathrm{vol}(A) := \sum_{i,j=1}^{d} a_{ij}$$

and C being a normalization constant precisely ensuring that $\mathrm{Conf}(A)$ has minimal and maximal value equal to 0 and 1 respectively, i.e. $\mathrm{Conf}([0,1]^{d\times d}) = [0,1]$.

The function $\mathrm{Conf}(A)$ can be interpreted as a total two-body interaction between each pair of voxels in a slice, weighted by a kernel function of their distance: Voxels at positions (i,j) and (m,n) interact which each other through a multiplication of their WMH probabilities $p_{WMH} \in [0,1]$ with $p_{WMH} = a_{ij}$ and $p_{WMH} = a_{mn}$ respectively. The interaction strength is weighted by a function $k(i-m, j-n)$ which decreases with the distance between the two voxels, i.e. the further two voxels are apart, the less their interaction contributes to the confluence measure. We choose this kernel function to be a Gaussian $e^{-s((i-m)^2+(j-n)^2)}$ with $s = 0.04$. The value of s is chosen empirically ensuring that voxel interactions outside of a radius of 10 voxels only contribute with a weight of less than 0.01, which constitutes a plausible length scale in the context of brain images. The resulting interaction terms $a_{ij}\, a_{mn}\, k\,(i-m, j-n)$ are then summed over all pairs of voxels.

Normalization

As a next step, we normalize by dividing by the volume $\mathrm{vol}(A)$ of the image. The reason for that is that the confluence measure without normalization increases with an increasing number of WMH voxels in a slice regardless of their distribution. The normalisation convention chosen in (1) therefore leads to a measure reflecting only confluence itself.

Note that even after volume normalization the confluence metric can still slightly depend on volume in an indirect way: if the volume increases because clusters are larger and not because there are more clusters, then the confluence metric will increase slightly because there are more interactions of voxels within the larger clusters, which is a desirable outcome. However, the metric will not increase disproportionately as it would if no volume normalization was applied.

Scaling

Next, in order to ensure interpretability and comparability, the confluence metric is scaled so that it falls between 0 and 1. This is achieved by dividing it by the maximum possible confluence value C, i.e. the value for an image consisting entirely of WMH voxels.

$$C := \frac{1}{d^2} \sum_{i,j,m,n=1}^{d} k\left(i-m, j-n\right)$$

1.2 From slice confluence to 3D confluence

As a final step, in order to obtain a confluence metric for an entire image $A \in [0,1]^{d \times d \times d}$:, the volume-normalized and scaled $\mathrm{Conf}(A)$ values of all slices in a volume are summed up and divided by the number of slices containing WMH in order to obtain the average confluence metric for this image.

An alternative approach would be calculating confluence in 3D instead of slice by slice in 2D. This approach is discussed in more detail in the Supplementary Materials, however the 2D approach is preferable for multiple reasons: As shown in the Supplementary Materials, a high correlation between 3D and averaged 2D confluence metrics indicates that differences are negligible, however the 2D option has lower computational requirements. Computational complexity of the 2D slice formula is $\mathcal{O}(d^4)$, averaging over 2D slices then has computational complexity $\mathcal{O}(d^5)$. The 3D formula presented in the Supplementary Materials on the other hand has computational complexity $\mathcal{O}(d^6)$.

1.3 Proof of concept

The confluence metric increases with decreasing distance between WMH clusters and with increasing probability of voxels belonging to WMH. The function can also be applied to binarized images where voxels with a WMH above a certain threshold have a value of 1 and all other voxels have a value of 0. In the binarized case, the function will only depend on the number of and distance between WMH voxels.

As can be seen with simulated and real data, the confluence metric has desired characteristics as it does not rely on WMH volume or number of WMH clusters, and can distinguish between images which have identical volume or number of clusters.