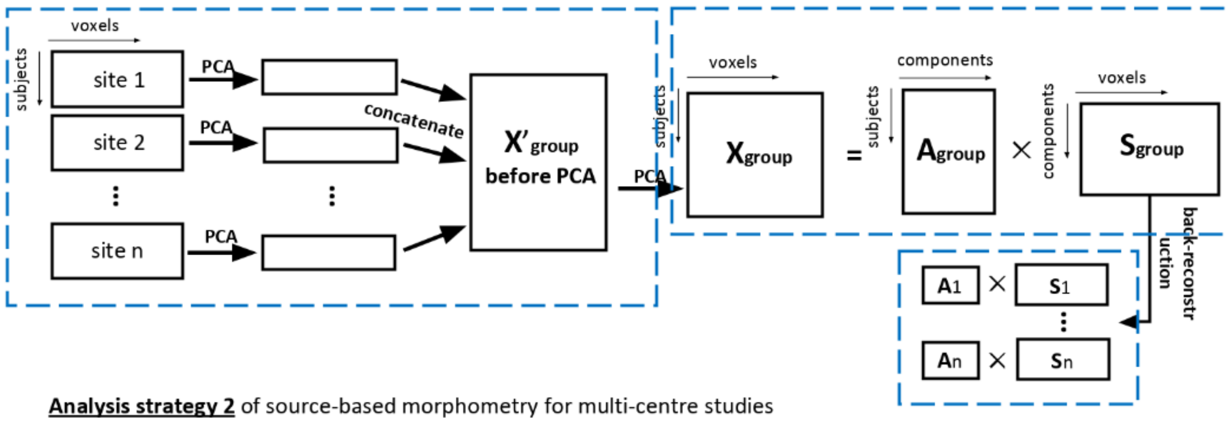
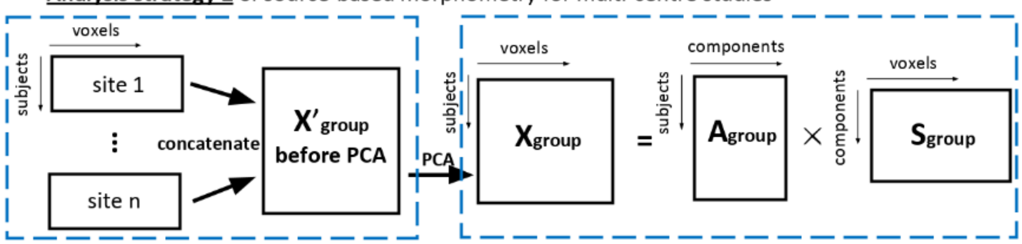
The purpose of this manual is to compare the results of the two analysis strategies.

Please note that for the spatial maps, the analysis strategy 1 would generate group-level as well as individual-level spatial maps (**S**$_n$), however, the 2nd strategy can only generate group-level spatial maps. For the loading vectors, the two strategies would generate individual-level vectors (**A**$_n$), but in different ways. Therefore, the following comparisons would include: comparing two strategies of their accuracy in 1) group-level spatial maps v.s. the ground truth (spatial map templates); 2) individual-level spatial maps v.s. the ground truth; 3) loading-vectors v.s. the ground truth (loading vector templates).

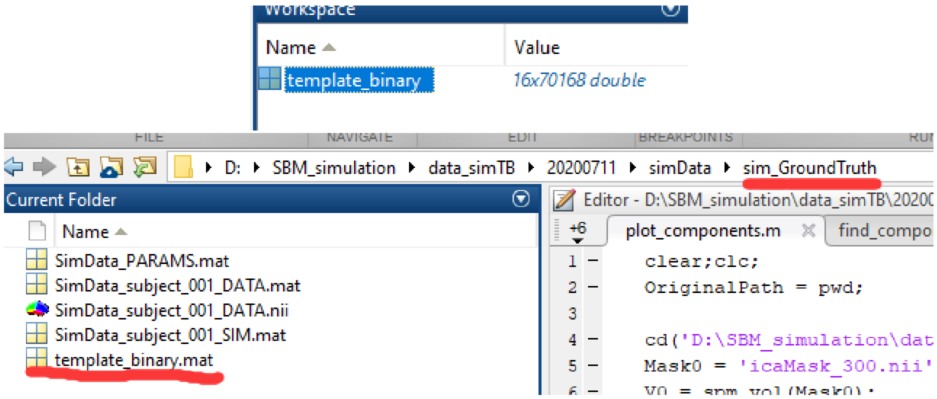
<u>**Analysis strategy 1**</u> of source-based morphometry for multi-centre studies



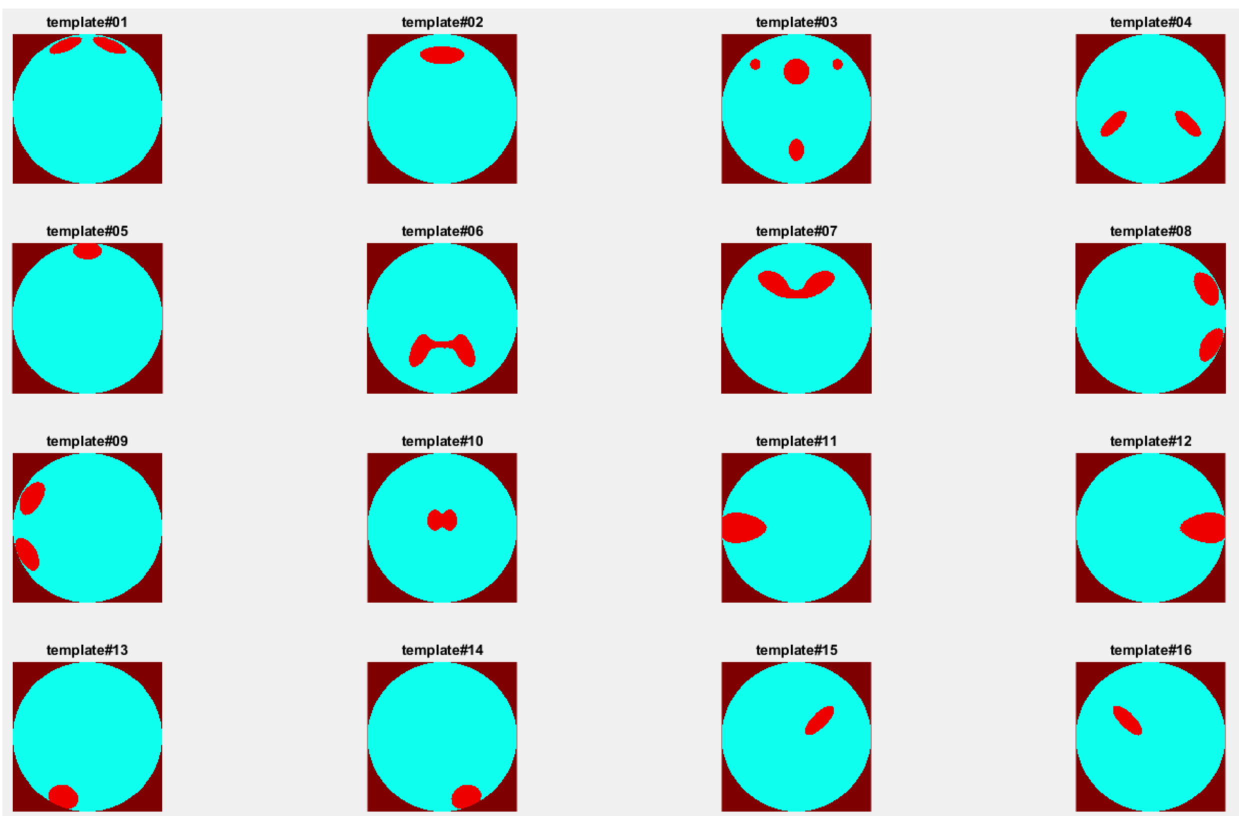
<u>**Analysis strategy 2**</u> of source-based morphometry for multi-centre studies



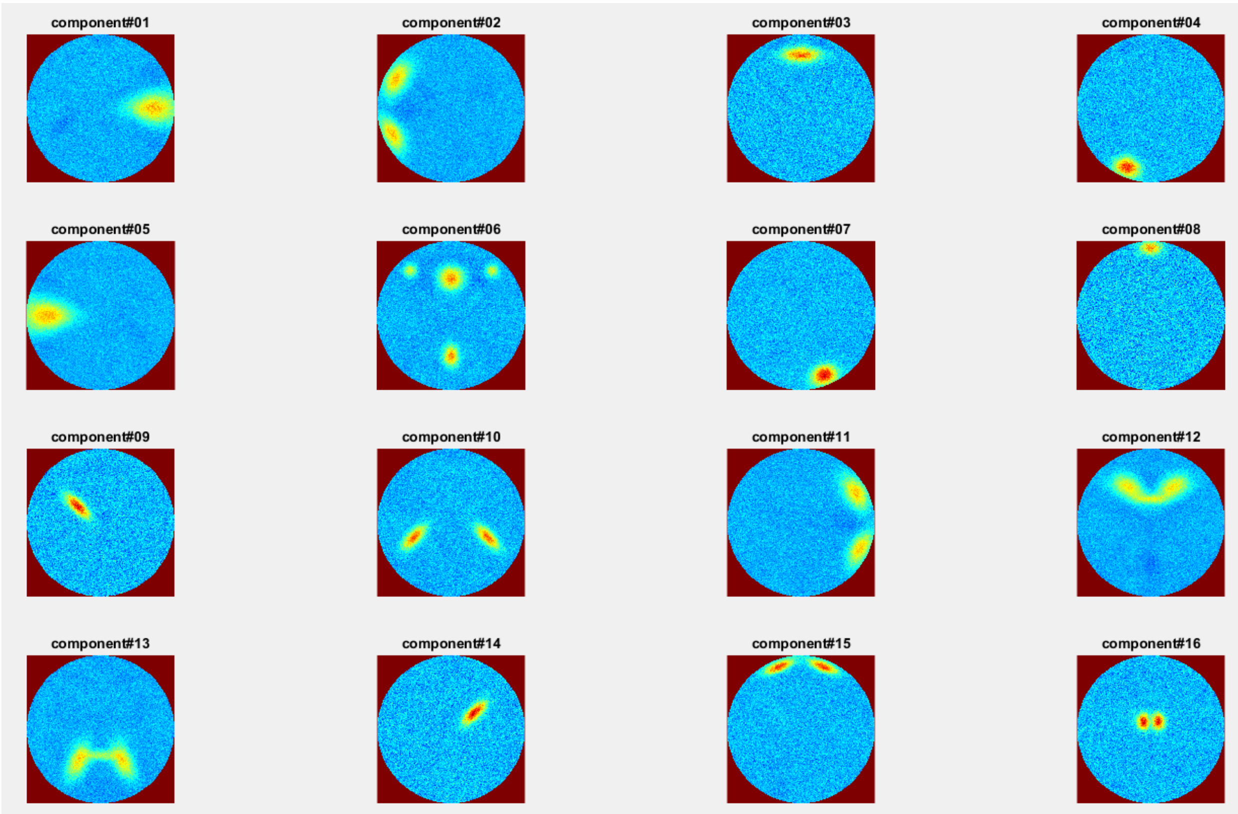
Step 1: load spatial map templates. ‘template_binary’ is a matrix with each row as a template of the simulated component, we will use these templates as ground truth to estimate the accuracy of each analysis strategy. There are 70168 entries in each row, indicating 70168 non-background pixels in the simulated data.



The 16 templates look like these:



Step 2: re-order the ICA outputs. Usually, the order of an ICA-generated components is random (e.g., component#01 actually corresponds to template#12). Therefor, one needs to find out how these 16 components correspond to the 16 templates.



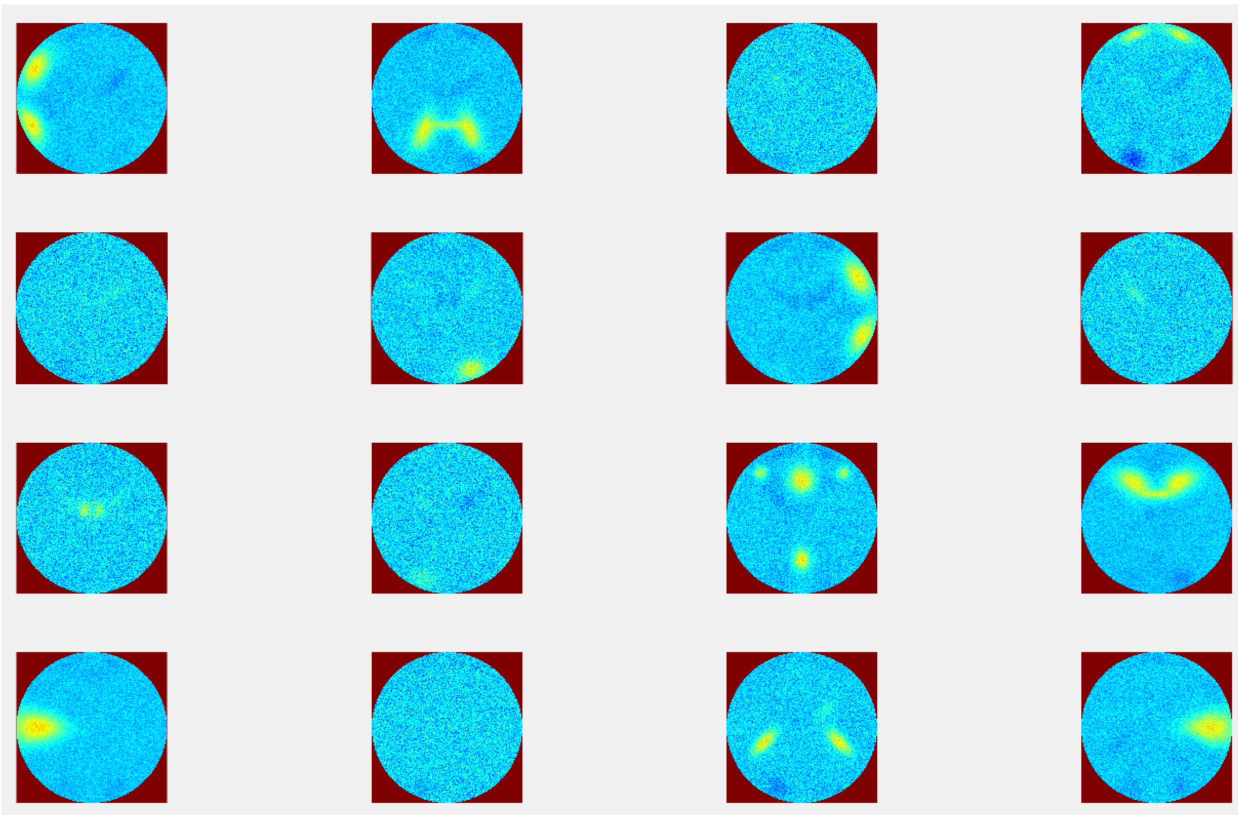
results of strategy 1

Please execute "component_order.m" (**remember change the folder names in the code!!!**), and you'll get results like the following:

```
>>
order_strategy1'

ans =

    15     3     6    10     8    13    12    11     2    16     5     1     4     7    14     9

>> order_strategy2'

ans =

     4    11    11    15     4     2    12     7     1     9    13    16    10     6     9     8
```

"order_strategy1" "order_strategy2" are the orders of the ICA outputs which are correspond with the order of the templates. For instance, for the "order_strategy1", 15 indicates that the 15th component corresponds to the 1st spatial template.

For the "order_strategy2", you will see the 2^{nd} and 3^{rd} elements were the same (i.e., 11), but you will find the 11^{th} component should correspond to the 2^{nd} template, not the 3^{rd} template, and the you can hardly find a component corresponds well with the 3^{rd} template. Therefore, the 3^{rd} 11 in the "order_strategy2" was not accurate, this is because with the strategy 2, there are several "random noise"-like components. If this condition happens (2^{nd} and 3^{rd}, 1^{st} and 5th, 10^{th} and 15^{th}), you can assign other numbers to these "random noise" components. In this example, I've assigned 3, 5 and 14 to the 3^{rd}, 5^{th} and 15^{th} components. Therefore, the "order_strategy2" would be [4 11 **3** 15 **5** 2 12 7 1 9 13 16 10 6 **14** 8].



results of strategy 2

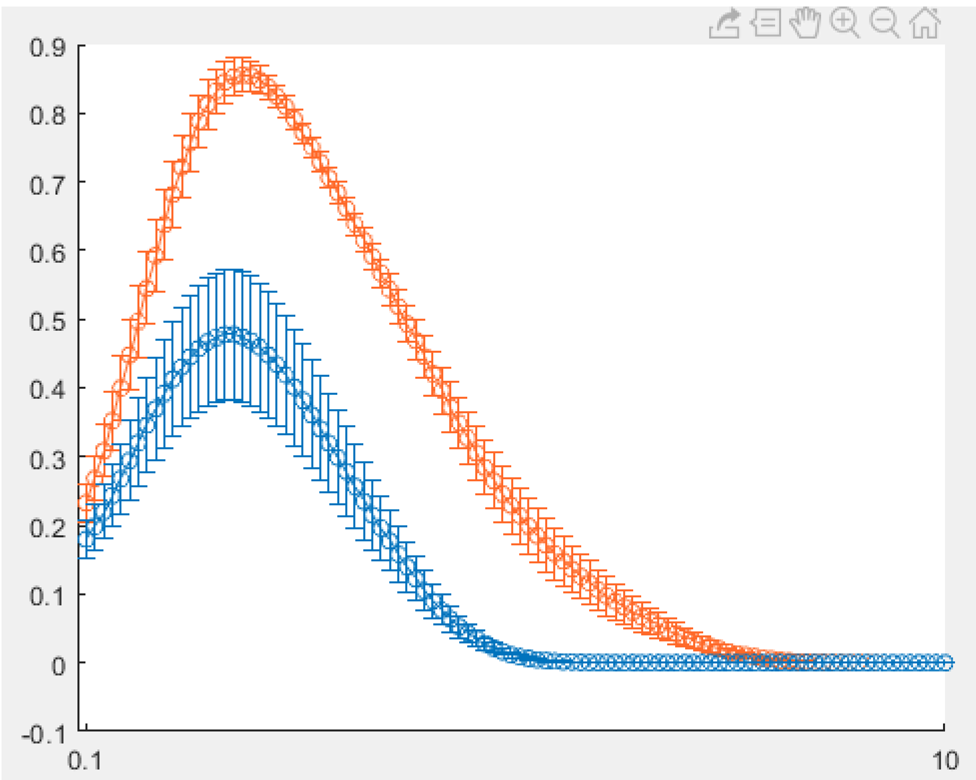
Step 3: Compare the two strategies by quantifying their spatial map similarities with the templates.

Compute the Dice's coefficient between the ICA components and spatial templates. Since Dice's coefficient (https://en.wikipedia.org/wiki/S%C3%B8rensen%E2%80%93Dice_coefficient) quantifies the similarity between two binary images (only have values of 0 and 1), and the templates are 16 binary images, we have to threshold the ICA components to binarize them. We'll use the threshold $\in$ [0.1,10].

Please execute "compare_spatialMaps.m" (**remember change the folder names in the code!!!**). This is a matlab function, the inputs of this function should be two vectors. You can execute this function in the command window:

```
>> order1=[15 3 6 10 8 13 12 11 2 16 5 1 4 7 14 9];
order2=[4 11 3 15 5 2 12 7 1 9 13 16 10 6 14 8];
compare_spatialMaps(order1,order2)
```

The output should be one figure:



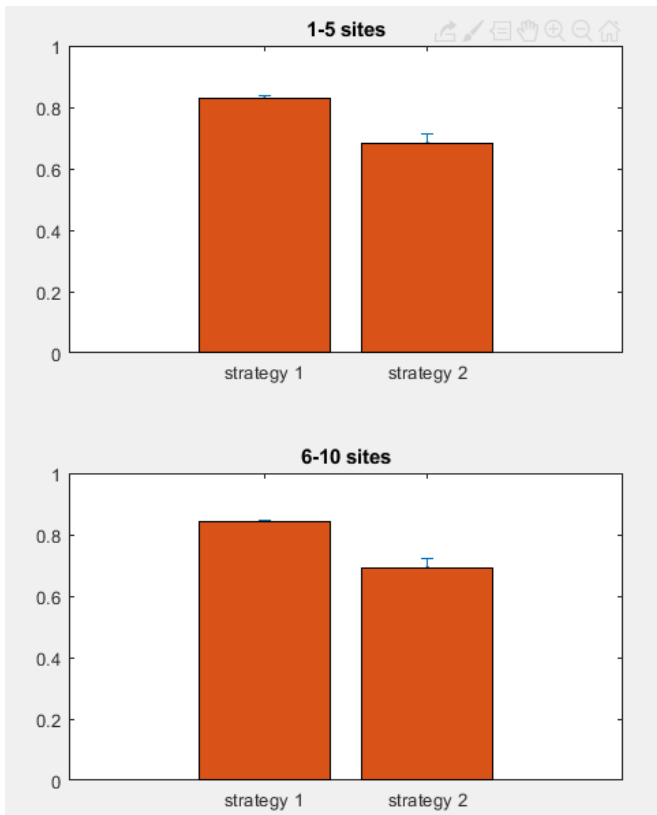
The x-axis is the z-score threshold $\in$ [0.1,10], and the y-axis is the value of Dice's coefficient. Orange curve indicates results of strategy 1, and blue curve indicates results of strategy 2.

Step 4: Compare the two strategies by quantifying their loading vector similarities with the templates.

Please execute "compare_spatialMaps.m" (**remember change the folder names in the code!!!**). This is a matlab function, the inputs of this function should be two vectors. You can execute this function in the command window:

```
Command Window
>> order1=[15 3 6 10 8 13 12 11 2 16 5 1 4 7 14 9];
order2=[4 11 3 15 5 2 12 7 1 9 13 16 10 6 14 8];
compare_loadingVectors(order1,order2)
```

The outputs should be one figure:



The y-axis is the value of average correlation coefficients between the 16 ICA loading vectors and the 16 templates.