

```
In[162]:= ClearAll["Global`*"]
```

Similarity metric

In this folder you will find the strength maps of the COBEv2 and HadISST datasets and the file land.txt. The file land.txt contains, well, the land...specifically values in the land are set to Indeterminate and values in the ocean to 0.

We need the land file for the following reason: we want to focus only on points in the ocean.

Also there could be small differences between the land in HadISST and COBEv2 so we apply in both the land found in COBEv2.

In the folder you will also find a pdf with the printed code.

author: Fabri Falasca

email: fabrifalasca@gmail.com

```
In[163]:= (*Paths to strength maps*)
```

```
pathStrengthMap1 = StringJoin[NotebookDirectory[], "strengthMap_COBEv2.txt"];
```

```
pathStrengthMap2 = StringJoin[NotebookDirectory[], "strengthMap_HadISST.txt"];
```

```
(*path to land*)
```

```
pathLand = StringJoin[NotebookDirectory[], "land.txt"];
```

function

```

In[166]:= (*Function to download the 2 strength maps*)
importing[path1_, path2_, pathLand_] :=
Module[
  {sMap1, sMap2, dim, result, timeSeries, land, sMapLand1, sMapLand2},
  sMap1 = Import[path1, "Data"];
  (*Importing the first Strength Map*)
  sMap2 = Import[path2, "Data"];
  (*Importing the second Strength Map*)
  (*Dimensions?*)
  dim = Dimensions[sMap1];
  (*Importing a matlab file with the lands stored in it*)
  land =
    Partition[
      ToExpression[
        Import[pathLand, "Words"]],
      {dim[[2]]}];
  sMapLand1 = Evaluate[sMap1 + land];
  sMapLand2 = Evaluate[sMap2 + land];
  result = {sMapLand1, sMapLand2};
  (*Result as a list with the 2 strength maps*)
  result
]

```

```

In[167]:= (*Function to compute the Distance between the two strength maps*)
distance[x_List, y_List] :=
Module[
  {preFlattenX, preFlattenY, flattenX,
   flattenY, num, den, notNormalizedResult, norm, distance},
  preFlattenX = Flatten[x]; (*From a matrix to a vector*)
  preFlattenY = Flatten[y]; (*From a matrix to a vector*)
  (*So instead of working with a matrix of dimensions (m,n) I work with
   a vector of length mxn. I then remove the Indeterminate (coming from the mask)
   values and work with the remaining part of the vector.*)
  flattenX = Select[preFlattenX, NumberQ[#] &];
  flattenY = Select[preFlattenY, NumberQ[#] &];
  (*now we compute the distance between two strength maps (now vectors) A, B*)
  (*Towards Distance(N1,N2)*)
  (*Step 1: Computing numerator: num = Abs[a_ij - b_ij] *)
  num = Total[Abs[flattenX - flattenY]];
  (*Step 2: Computing denominator:
   den = | randomized(a_ij) + randomized(b_ij) | *)
  den = Total[Abs[RandomSample[flattenX] - RandomSample[flattenY]]];
  distance = num/den;
  distance
]

In[168]:= similarity[path1_, path2_, pathLand_] :=
Module[
  {imp, strengthMap1, strengthMap2, dist, sim},
  (*Step 1
   Import the two strength maps*)
  imp = importing[path1, path2, pathLand];
  (*Step 2
   Define the two matrices*)
  strengthMap1 = imp[[1]];
  strengthMap2 = imp[[2]];
  (*Step 3
   Take the distance between them*)
  dist = distance[strengthMap1, strengthMap2];
  (*The similarity will just be S = 1 - d *)
  (*As a result give me the correlation*)
  sim = 1 - dist;
  sim
]

```

run

```
In[169]:= similarity[pathStrengthMap1, pathStrengthMap2, pathLand]
```

```
Out[169]= 0.722069
```