Contents

1 Notes on implementation

This document serves as a look-up for the implementation of the package. It is useful to collect all the necessary information that guides the development.

1.1 Module MetricMeasureSpaces.jl

This contains the struct and functions related to metric measure spaces. For this project we initially consider only **finite** metric measure spaces. We summarize everything here.

1.1.1 struct MetricMeasureSpace

```
struct MetricMeasureSpace:
    C::SMatrix
    \mu::StaticArray
    constructor1(C, mu=nothing)
    constructor2(array, dist_function, mu=nothing)
end
```

1. C:
   - static matrix (Float64)
   - dimension NxN
   - It contains the dissimilarities between elements of the metric space. Since it is not a distance, it can contain any value.
   - It must be of type Float64 for implementation purposes (we have to apply gradient based optimization).
2. μ
   - static array (Float64), dimension N.
   - It contains the relative importance of each element in the space.
   - it must be positive

- it must sum to 1 (represents a probability)
- it must have the same dimension as matrix C

3. constructor1: inner constructor. Arguments:

   - C matrix of numeric type. Dimension NxN.
   - μ. Optional. array of {float, int, rational, uint}.

   Requirements:

   - If μ has a negative entry –> raise error
   - If μ is not normalized –> renormalize and raise warning
   - If μ and C have different dimensions –> raise error
   - Force μ to be Float64
   - Force C to be Float64
   - if μ is not provided, initialized as uniform.

4. constructor2 : second inner constructor

   (a) **TODO** decide that maybe it can be an outer constructor Arguments:

   - elements. Array of any type (T for example).
   - dist. Function of type dist(T, T): Float. It takes two elements of type T and determines their dissimilarity. Not necessarily a distance.
   - μ. Optional. array of {float, int, rational, uint}.

   Requirements:

   - It calculates C according to elements and dist.
   - It must check that the elements are all of the same type.
   - The it calls constructor1.