D6.5 Development and assessment of a validation method for assessment of plant trait extraction from the ROMI pipeline

Fabrice Besnard, Christophe Godin

ROBOTICS FOR MICROFARMS

Deliverable			**D6.5**	
Deliverable title: Development and assessment of a validation method for assessment of plant trait extraction from the ROMI pipeline **Accompanying report**				
Task	WP6 - Plant modelling T6.1 Modelling. T6.2 Extraction of plant parameters using neural networks. T6.3 Models of plant status for crop monitoring.			
Task Leader	**INRIA**	Planned Date	**12.07.2022**	
		Effective Date	**05.07.2022**	

	Written by	Reviewed and approved by	Authorized by
Name / Surname	**C. Godin (Inria)** **F. Besnard (CNRS)**	**David Colliaux**	**Peter Hanappe**

Executive Summary

Deliverable D6.5 is a demonstrator of the development and assessment of a validation method for assessment of plant trait extraction from the ROMI pipeline.

The demonstrator is provided as a **full standalone docker image** that contains pedagogical jupyter notebooks to run and discover our programs. It can be found at the following addresses:.

a docker image called “sm-dtw_demo”, hosted in the docker hub account of the romi project

To accompany this docker image, **we provide supports** to help a wider audience use our program:

- a readme explanation detailing the main procedures, hosted in the github repository of the project
- **a tutorial video** explaining both the context of use relevant for our programs and practical demos of the jupyter notebooks provided in the docker image ;
- another written help hosted by the documentation of the romi projet (https://docs.romi-project.eu/documentation/), with **helpful pointers to the different parts of the video**

This document accompanies the demonstrator.

1.1 Overview and description of the demonstrator content

The demonstrator notebooks are meant to present in a pedagogic manner the tools we developed to assess how good a phyllotaxis measure is and how good this assessment is.

In brief, it allows anybody to simulate phyllotaxis data (pair of sequences consisting of ground truth sequences and their error-prone measures), assess the measure performance with our new program 'sm-dtw' (detect errors and quantify precision) and control that this program correctly interpret the differences between the measure and its ground truth reference.

1.2 Partners involved

Leader: **CNRS**
Participants: CNRS, INRIA, IAAC (video)

1.3 Relation with other work packages and tasks

Relation to WP5 tasks.

The work presented in this deliverable is the quantitative method that was designed to assess the results of the computer vision algorithms developed in the Romi Plant Imager pipeline.

Relation to other ROMI work packages:

None.

1.4 WebLinks to videos, flyers …

- **D6.5 video**:

  https://zenodo.org/record/6793459#.YsQ3uC8itqu

- D6.5 SUPPLEMENTARY DOCUMENT1.pdf

1.5 Dissemination / IPR policy (since the beginning of the project)

Articles in peer-reviewed journals:

Two papers are in preparation based on the material of these reports:

- Paper 1: Technical presentation of the SM-DTW algorithm. Target: computer science journal journal. Contents: mostly an extension of D6.5 SUPPLEMENTARY DOCUMENT 1.
- Paper 2: Application of SM-DTW to sequences of divergence angles and internode length. Sensitivity analysis of the method and results.

Workshops, conferences :

Press Release:

Television coverage:

Outreach:

2 Main body

I. Assessment principle: comparing a phenotyping result with a ground truth control

A key objective of the Romi project was to demonstrate that low-cost technology used for automatic management and monitoring of plants in the field, could be adapted to automate as well the phenotyping of model plants used in research. Such a technology could be of tremendous interest for research labs throughout the world if sufficiently precise and robust. A first technological platform illustrating this new possibility was constructed by the ROMI project to perform high precision phenotyping of *Arabidopsis thaliana* (referred hereafter as *Arabidospsis*) inflorescence architecture.

The Romi Plant Imager and its software suite (plant-3d-vision) provides a pipeline to automatically execute a series of steps leading to the extraction of the traits of interest from the observed plant Fig 1.a. The first component takes pictures of a plant by moving a camera around the plant. Romi algorithms then produce a point cloud that is further interpreted by higher level software components. As an output, the pipeline produces main quantitative traits characterising the structure of the inflorescence architecture, Fig 1.b. In our case study, we selected traits that are essential to describe the inflorescence phyllotactic patterns, namely the sequences of divergence angle between consecutive siliques (*Arabidopsis* fruits) and their corresponding internodes, Fig 1.c.

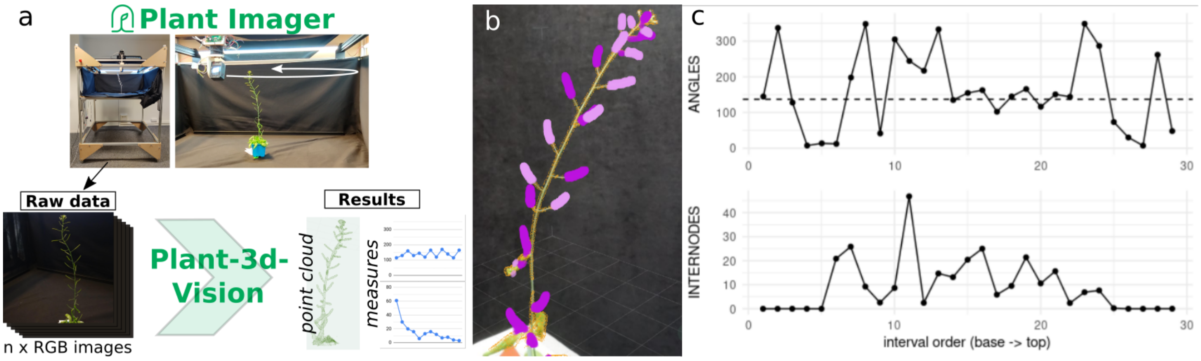

Fig 1. The Romi Plant Imager produces for each individual plant two sequences of botanical traits. (a) Outline of the Romi Plant Imager pipeline (b)This *Arabidopsis* main inflorescence stem was acquired and analyzed with the pipeline, automatically segmented fruits are highlighted on the picture (c) Divergence angles and internodes automatically measured on the above plant.

To be used in scientific investigation, the quality and accuracy of the divergence angle and internode sequences produced by our phenotyping pipeline should be quantitatively assessed. For this, one should first obtain ground truth data, and second, compare the pipeline output sequences with the ground truth ones. Although performing such comparison manually is in principle possible, this task turns out to be complex and is not doable on more than a few individuals in practice. This limits the possibility to analyse large datasets for a sound statistical analysis of the phenotyping performances, as well as it hinders rapid cycle of development where the measure accuracy (similarity of the pipeline result to the ground truth control) is the target objective to improve.

Interestingly, we found no algorithm in the literature that could carry out such a comparison between ground truth and computed sequences of tree structures, especially when they are affected by branching gains or losses. We then designed a new algorithm called SM-DTW (for Split-Merge Dynamic Time Warping), to carry out such comparison efficiently (section III).

II. Meta-assessment: digital twins and the gold-mine of synthetic ground truths.

Developing new algorithmic tools requires extensive testing, especially to achieve some genericity and for them to be useful in a wide range of real situations. Determining the robustness of an algorithm to different inputs and characterising its adapted range of use can require collecting a lot of different “test” data, whose production and/or accessibility can be a real bottleneck.

In the Romi project, we push forward the **use of synthetic data to test and assess our computer tools**. In theory, once a proper generator has been created, the production of test data can be virtually illimited, and according to the model design, a considerable parameter space can be efficiently explored. Furthermore, synthetic data provides an incomparable opportunity to generate all kinds of ground truth data, possibly free of any artefact or inaccuracy of measurement methods.

Hence, to test the image-based phenotyping pipelines, we developed a realistic virtual *Arabidopsis* model, accurately reproducing the architectural development of the plant (deliverable D6.4). Images of virtual *Arabidopsis* plants can be taken as our real Plant Imager would do, creating digital twin datasets that can be analysed with the same tools as real plants. The results are then compared to the initial ground truth encoded by the model (labelling of plant parts, counts of structures, topology metrics, length, areas, etc…), offering a fast, inexpensive, exhaustive and precise assessment of the phenotyping pipeline.

Likewise, we developed here a program to specifically test SM-DTW (described in section III). This program creates synthetic sequences of divergence angles and internodes with different sources of (controlled) perturbations. The program also records the perturbation transforming the initial sequence (reference) to the perturbed version (test), providing a **ground truth realignment solution** against which the SM-DTW prediction can be tested. Thanks to this tool, we present performance results of SM-DTW as a new quantitative assessment methodology. We show that it is highly reliable, and when applied to our phenotyping pipeline, it provides first insight on its output quality.

III. Design of a new algorithm for aligning ground truth and simulation sequences (SM-DTW)

Comparing a ground truth sequence with a modified test sequence can be formulated as a mathematical optimization problem. Due to the particular mathematical properties of segmentation errors in the sequences, this turned out to be a difficult combinatorial problem for which no algorithmic solution existed in the literature.

Briefly, the problem is to compare two stem segmentations into internodes and lateral organs, obtained from a common point cloud. These segmentations are meant to reflect the same observed organ sequence reality Fig 2.a. They are thus expected to be the same in many places but with possibly missed or inserted organs. Assume that a lateral organ in the reference sequence was missed in the output sequence. Then it is expected that, in the output sequence, the internode length is the sum of internodes length surrounding the missed organ in the reference sequence. Fig 2.b. Therefore, the optimization algorithm should construct a mapping between the reference and the output sequence by testing in principle all possible hypotheses of one or several organs being inserted or missed at every position while respecting the previous constraint that lengths (or divergence angles) must keep consistent between mapped aggregated segments in each stem. We call this problem the "*best aggregation mapping*" problem, Fig 2.c.

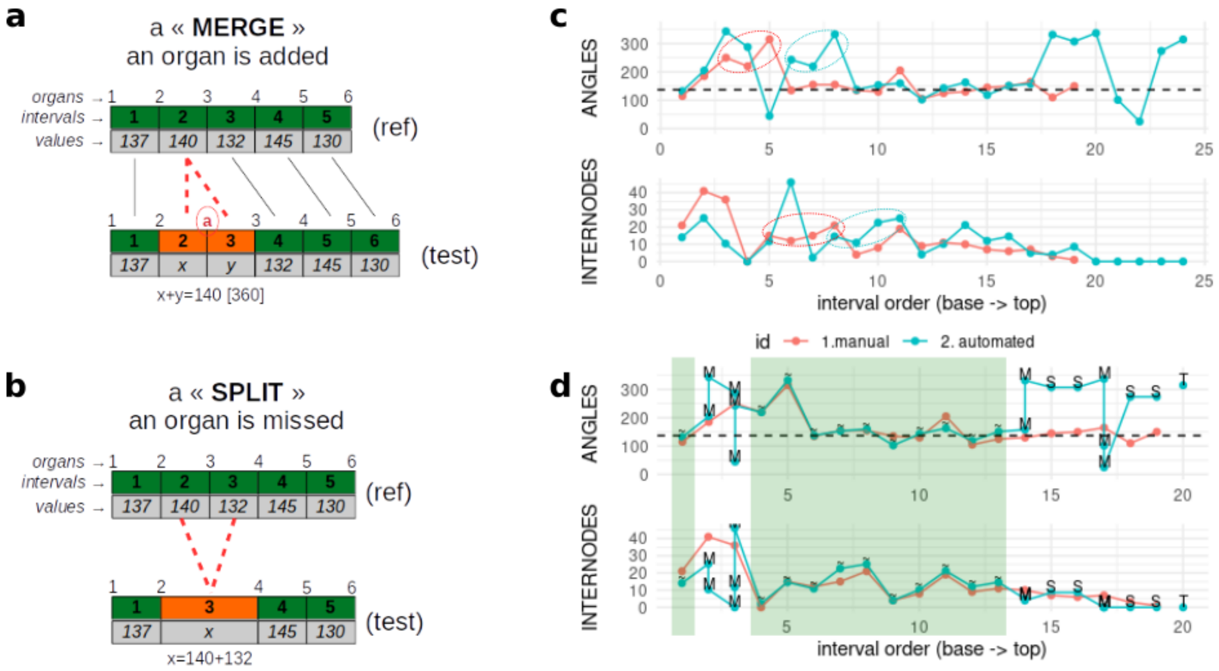

Fig. 2. *Best aggregation mapping* problem. (a) Effect of an organ addition into the test sequence: two intervals will be merged into a previous interval (b) Effect of an organ loss into the test sequence: a new interval will be splitted into two previous intervals. Merge and split impose a mathematical conservation of the attributes values between the locally concerned intervals (for a and b, the top two figure rows indicate organ and interval orders respectively while values in the grey lines are typical divergence angle values) (c) A real *Arabidopsis* reference sequence (20 organs, red) and an output sequence from the Romi Plant Imager pipeline (25 organs, blue). While shifts in identical patterns are easy to detect (outline by dotted circles), other divergent patterns are too complex to be interpreted by visual inspection (d) The solution of the optimal best aggregation realigns the previous two sequences

by proposing 7 organ additions (creating M-labelled dots aligning vertically) and 2 organ losses (creating S-labelled dots aligning horizontally). “~” labels indicate matching values that diverge only by measure precision (corresponding segments are shaded in green), “T” (*T*ail) indicate the particular addition on an organ after the last real organ.

This is a huge combinatorial optimization problem. However, we showed (D6.5 SUPPLEMENTARY DOCUMENT 1) that this problem can be solved efficiently by using the dynamic programing principle (Bellman, 1957). This is the same principle that is used in BLAST algorithms to compare DNA sequences, or in speech recognition to align sequences of speech signal. However, our optimization problem being different, we had to conceive a new dynamic programming-based algorithm, named split-merge dynamic time warping (SM-DTW), detailed in D6.5 SUPPLEMENTARY DOCUMENT 1. This algorithm is inspired from dynamic time warping algorithms used in speech processing, e.g. (Sakoe & Chiba, 1978). Hence, SM-DTW solves the best aggregation mapping problem and basically proceeds as follows.

Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1..I}$ be the output sequence and $\mathbf{Y} = \{\mathbf{y}_j\}_{j=1..I}$ be the reference sequence (the one for providing the reference known segmentation). Note the lengths of these sequences can be different, i.e. $I \neq J$. SM-DTW compares the two sequences by scanning them from left to right, trying all possible aggregations at each step, and keeping track of optimal ones only (thanks to the dynamic programming principle). For each pair (i, j), $i = 1..I$ and $j = 1..J$, it computes the optimal solution to the best aggregation mapping problem between partial sequences $\mathbf{X}[i] = \{\mathbf{x}_{i'}\}_{i'=1..i}$ and $\mathbf{Y}[j] = \{\mathbf{y}_{j'}\}_{j'=1..j}$. Then, the optimal solution to our comparison problem is obtained when we get the optimal solution to aligning $\mathbf{X}[I]$ and $\mathbf{Y}[J]$. If K is the maximum length of sub-sequences that can be aggregated in either $\mathbf{X}$ or $\mathbf{Y}$, then we showed that the time complexity of the algorithm is in $O(I.J.K)$, meaning that the computation time will growth as a linear function of either I, J or K (K is a parameter of the algorithm). See details in the companion document D6.5 SUPPLEMENTARY DOCUMENT 1.

IV. Generation of paired sequences of divergence angles and internode lengths with controlled sources of perturbation as synthetic ground truth

Definition of paired phyllotactic sequences and of the source of perturbations.

To assess the reliability of the pipeline assessment method, we designed algorithms to produce paired, synthetic phyllotactic sequences. The sequence pair is made of a reference and a test. While the reference sequence can be considered as a ground truth, the test sequence derives from this latter by adding three types of perturbations: measurement noise on values, permutation in the order of close organs along the stem and segmentation errors. Segmentation errors correspond to either an addition of a false positive organ (over-segmentation) or to the loss of a true reference organ (under-segmentation).

Generation of realistic synthetic data.

To obtain realistic sequences close to our experimental data, we studied the major statistical properties of real sequences (obtained from manual measurements) to calibrate our synthetic phyllotaxis sequences. Divergence angles are cyclic data (angles being between 0 and 360 degrees). In previous studies (Guedon et al, 2013, Besnard et al 2014), we characterised the histograms of divergence angles encountered in wild-type (WT) *Arabidopsis* plants and some of its mutants. Here, we focus on WT distributions only, using the same method of manual measurements (Fig 3. a), but we add for the first time data for internode lengths too. We measured phyllotaxis for 15 new plants (almost 400 intervals). These distributions show substantial variation which can result both from measurement and biological noises. Concerning divergence angles, measured data show a typical asymmetrical distribution with a main peak around 137.5 degrees (the canonical golden angle) and secondary modes roughly centred on multiples of the golden angle. These non-canonical angles are due to the existence of permutations in the standard order of organs along the stem (Besnard, 2014). To mimic them, we modelled natural permutations of organs in our sequence generator. By controlling permutation frequency, we observed the apparition of a realistic asymmetrical angle distribution in our synthetic data. Although a proper statistical model would be required to fit to the mix of the different angles modes (Guedon et al, 2013), we fine-tuned the main parameters of our generator for divergence angles (natural standard deviation -sd- of angles, frequency of natural permutation) to obtain the best visual fit of the distribution and close global statistics (global mean and sd). We also proceeded by a global visual fit to reproduce a realistic internode length distribution, focusing on length decay along the stem, the length at the final plateau and length sd rather than natural permutations (which do not produce stereotypical pattern here as in divergence angles).

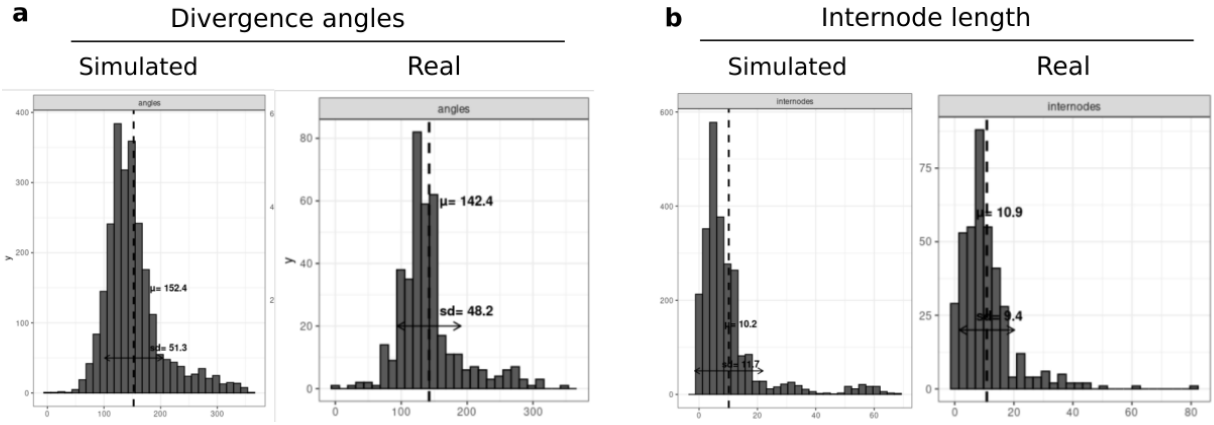

Fig 3. Reproduction of real phyllotaxis distribution in synthetic ground truth data (a) Histogram of divergence angles in simulated data (N=100) versus WT (N=15). Asymmetry can be observed in the distribution that corresponds to secondary modes due to “permutations”, which were modelled in the sequence generator (b) Histogram of internode lengths in simulated and real data of the same plants, showing also a skewed distribution. Descriptive statistical metrics are provided (µ: mean, vertical dotted line; sd: standard deviation, horizontal double-headed arrow)

Simulating perturbations and creation of a groundtruth realignment between the paired sequences. Finally, we designed a program to perturb a phyllotactic sequence by the three types of perturbations defined above. In particular, the correspondence between each organ in the reference sequence and in the final test sequence is kept as a table, as well as the consequence in terms of successive intervals for both angles and internode length. Measurement noise, artificial permutations and segmentation errors can be defined and controlled separately by the users as input, in order to create a wide range of scenarios affecting the test sequence. We created dedicated plotting functions to align the two paired sequences and export functions to use generated synthetic data as input for our new alignment algorithm, SM-DTW.

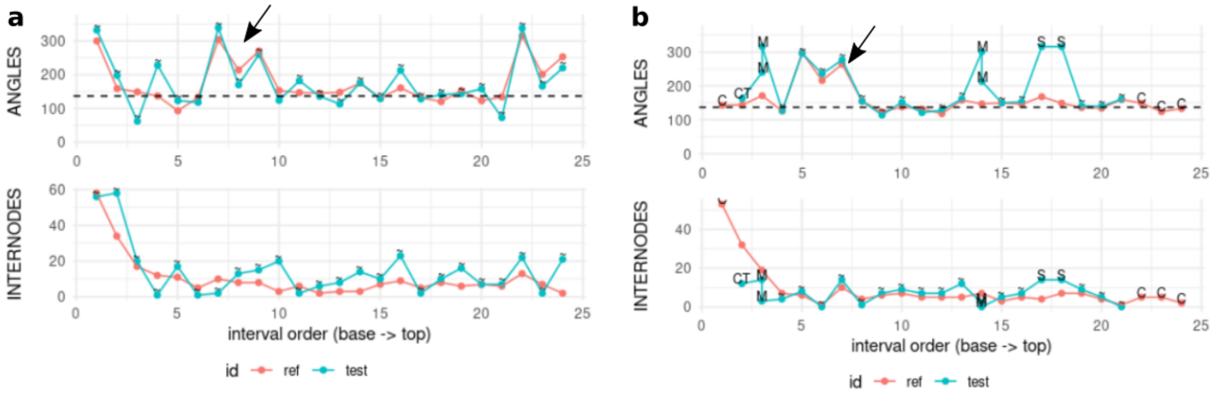

Fig 4. Creation of groundtruth alignment of synthetic reference and test phyllotaxis sequences. In the two plots, red and blue curves are reference and test sequences, respectively ; divergence angles of internode length, in phase. The black arrows indicate a typical M-shape motif, which is a consequence of the natural permutations modelled by the sequence generator. (a) Test sequence diverges only from the reference sequence by adding a high level of white noise (values randomly picked from a gaussian distribution centred on zero) to each values. The intensity of this noise is controlled by the standard deviation (sd) of the random gaussian noise: here it is 30° for angles and 10mm for internode length. No segmentation errors have been added: the sequence sizes are equal, all points are matching ("~" label) (b) In this second simulation, noise on test values has been reduced (sd=10° for angles and 3mm for internodes) but segmentation errors have been imposed: organs have been missed at both end of the true sequence (generating unmatched C=chops values at both end of the reference sequence) and within the sequence (generating S=split) , several organs have been added (T=Tail at the beginning, M=merge within the sequence).

Code is available for simulating synthetic paired phyllotactic sequences: Phyllotaxis-sim-eval

V. Assessment of the SM-DTW algorithms

To assess the SM-DTW algorithm, we evaluated its results on our ground truth synthetic data (see section IV). We assessed the impact of different sources of errors on the SM-DTW ability to recover the true unperturbed sequence. For each experiment, a hundred reference sequences were generated with an average length of 25 elements. At each rank, a sequence consists of two values, one for the divergence angle, the other of the internode length.

We first tested the effect of the white noise on both the divergence angles and internode length of the reference sequences. This white noise is modelled by the addition to each initial reference value of a random picked up from a gaussian distribution centred on zero. The intensity of this noise is controlled by the standard deviation of the noise valu distribution: it was in the range of ±[0,35] degrees for divergence angles and ±[0,10] mm for internode length (Fig 5.a).

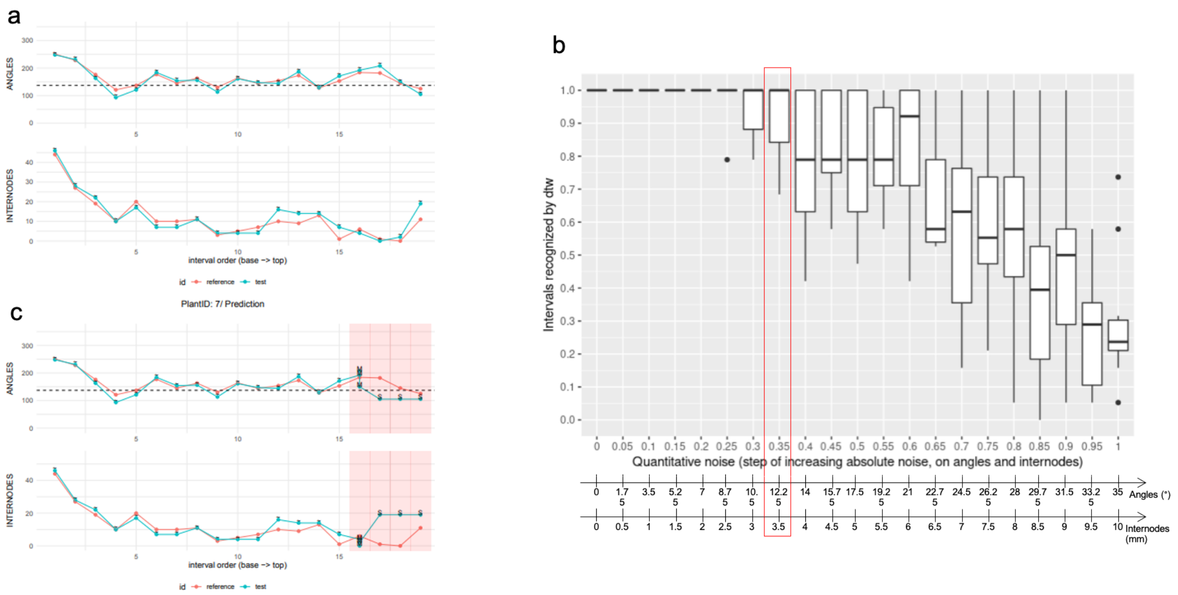

Fig 5. Assessment of noise level on the performance of SM-DTW. (a) Reference sequence (orange) and test sequence corresponding to the reference sequence with the addition of noise (green). (b) Variation of the percentage of perfectly recognized sequences by SM-DTW as a function of increasing noise level. (c) Typical error induced by noise, where noise can be interpreted occasionally as an inserted or missed organ by SM-DTW (last segments highlighted in pink).

We observe that SM-DTW is able to correctly interpret almost 100% of the noisy sequence up to a noise on the angles of ±12° and of ±3.5 mm on internodes (Fig. 5b). Beyond this noise level, noise starts to be interpreted as inserted or lacking organs, which is actually also impossible to sort out by an expert looking at the noisy sequences. This critical noise level is typically of the order of magnitude of the measurement noise, leading to the conclusion that SM-DTW can correctly recognize sequences when the quality of the ground truth is sufficient. However, too noisy measurements may lead to a degradation of the interpretation, both for humans and algorithms.

We then tested noise corresponding to simulated addition or removal or organs in the synthetic data (Fig 6.a), with additional different noise levels on the values of divergence angles and internode lengths. If the added noise remains weak, the algorithm is able to retrieve perfectly the organs inserted or missing in the sequence, corresponding to a detection of split and merge operations in either of the sequences, Fig. 6bc. The ability to correctly interpret the modified sequences start to be significantly affected for a noise of level 3, Fig. 6d, i.e. greater than ±20 degrees on divergence angles and ±5 mm on internode length (Fig. 5ef provides an illustration of such incorrect interpretations). Note that again, just based on the sequences, a human expert could not do much better, as the algorithm finds an optimal interpretation based on all the quantitative data available. This means again that, for reasonable measurement noises (i.e. reconstructed azimuthal positions of lateral organs and lengths of internodes) , the algorithm is able to automatically find the inserted and missing organs in the Romi Plant Imager pipeline output. First results corresponding to the application of SM-DTW to assess the ROMI pipeline have been obtained recently and are available in the Deliverable 5.5.

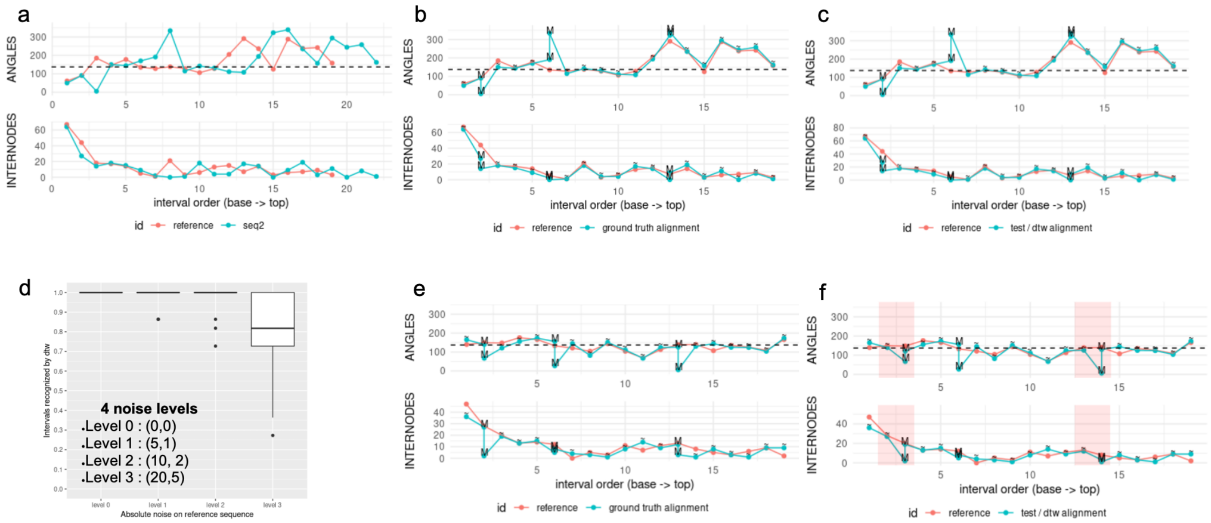

Fig 6. Assessment of the ability of SM-DTW to detect inserted and missing organs in synthetic sequences. (a) Synthetic ground truth reference (orange) and synthetic test sequence obtained from the ground truth by adding or removing organs from the reference sequence. These operations have been made on both divergence sequence and internode lengths consistently. (b) Ground truth interpretation explicitly showing the split and merge operations that have been made. (c) Result of the SM-DTW algorithm applied on data shown in a. The result is identical to the ground truth. (d) Percentage of synthetic modified sequence correctly interpreted by SM-DTW as a function of the noise level on angles and internodes, expressed as absolute values of the standard deviation of the applied white noise (in degree and mm for angles and internodes, respectively) . (e) Example of a sequence where SM-DTW interpretation differs from the ground truth (in pink zones areas).

3 Bibliography and technical annexes

Bellman, R., 1957. Dynamic programming. Princeton University Press, 340p.

Besnard, F., Rozier, F., Vernoux, T., 2014. The AHP6 cytokinin signaling inhibitor mediates an auxin-cytokinin crosstalk that regulates the timing of organ initiation at the shoot apical meristem. Plant Signal Behav 9, e28788-4. https://doi.org/10.4161/psb.28788

Guédon, Y., Refahi, Y., Besnard, F., Farcot, E., Godin, C., Vernoux, T., 2013. Pattern identification and characterization reveal permutations of organs as a key genetically controlled property of post-meristematic phyllotaxis. J. theor. Biol. 338, 94–110. https://doi.org/10.1016/j.jtbi.2013.07.026

Sakoe, H., Chiba, S., 1978. Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustic, Speech, and Signal processing ASSP-26, 43–49.

I. Demonstration video

The demonstration takes the form of a docker image with Jupyter Notebooks illustrating the full pipeline for assessing and using SM-DTW

D6.5 video

The docker encapsulates all the programs and third-party libraries required to run our tools, dispensing users from any complicated installations. A very simple set-up procedure allows to start 3 Jupyter notebooks that make a comprehensive workflow for testing the new SM-DTW assessment tool for phenotyping phenotyping. Indeed, we incorporated our program to simulate phyllotaxis data, generate perturbations and output a control ground truth alignment. Hence, users can explore by themselves the parameter spaces of sequence complexity and perturbation intensity and observe the performance of SM-DTW to realign a perturbed sequence with the original reference.

Jupyter notebooks are a really intuitive and pedagogic support to discover a code and its use. In addition, we provide sound documentation to assist a wider audience to the use of this docker, as a written online manual and as a tutorial video showing live screen recordings of the entire procedure, especially during the three Jupyter Notebooks.

3 Bibliography and technical annexes

Bellman, R., 1957. Dynamic programming. Princeton University Press, 340p.

Besnard, F., Rozier, F., Vernoux, T., 2014. The AHP6 cytokinin signaling inhibitor mediates an auxin-cytokinin crosstalk that regulates the timing of organ initiation at the shoot apical meristem. Plant Signal Behav 9, e28788-4. https://doi.org/10.4161/psb.28788

Guédon, Y., Refahi, Y., Besnard, F., Farcot, E., Godin, C., Vernoux, T., 2013. Pattern identification and characterization reveal permutations of organs as a key genetically controlled property of post-meristematic phyllotaxis. J. theor. Biol. 338, 94–110. https://doi.org/10.1016/j.jtbi.2013.07.026

Sakoe, H., Chiba, S., 1978. Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustic, Speech, and Signal processing ASSP-26, 43–49.

Split-Merge Dynamic Time Warping

Christophe Godin

June 30, 2022

Abstract

In this paper we are interested in the problem of comparing discrete sequences of vectors, possibly of different lengths, with the possibility to aggregate vectors prior to compare them. Each aggregated sub-sequence is replaced by a single vector corresponding to the sum of the aggregated vectors. Once aggregations have been made, the sequences are compared element-wise using a classical Euclidean distance as a local distance. Our aim is to find an aggregation that minimizes the total cumulated distance obtained in this way over the entire sequences. As the possibility to aggregate vectors in all different ways in both sequences is quite large, efficient algorithms are needed. Here, we propose a new dynamic programming-based algorithm to solve this optimization problem. The time complexity of the algorithm is in $O(IJK)$ where I and J are respectively the lengths of the two compared sequences and K is the maximum size of aggregated blocks.

1 Introduction

Comparing discrete sequences of symbols or vectors is a classical problem in computer science and pattern matching applications. A large family of algorithms relies on edit-distance mapping between sequences [3]. This consists of finding the minimum number of elementary edit operations that make it possible to transform progressively one sequence into the other. The distance between the two sequences is then defined by the minimal number of edit operations needed to make this transformation. It can be shown that under some conditions, this problem is equivalent to that of finding a mapping between the sequences respecting specific constraints. Various algorithms discussing different types of constraints and conditions have been proposed. This is at the origin of dynamic time warping algorithms. These algorithms make it possible to align two sequences in a non-linear manner by warping time so that the progression on one signal matches the progression on the other.

The efficiency of these techniques usually rely on the use of the dynamic programming principle [1] to cut down the complexity of the initial combinatorial problem. This principle has been extensively and successfully used in speech recognition [4, 2], in genome and molecular analysis, and in biology.

Here, we consider a new sequence comparison problem and we apply this principle to reduce the complexity of a new sequence comparison problem. Consider two discrete sequences of vectors, possibly of different lengths. We want to compare these two sequences with the possibility to aggregate vectors prior to compare them. Each aggregated sub-sequence is replaced by a single vector corresponding to the sum of the aggregated vectors. Once aggregations have been made, the sequences are compared element-wise using a classical Euclidean distance as a local distance. Our aim is to find an aggregation (viewed as either a split or merge operation of the sequences) that minimizes the total cumulated distance obtained in this way over the entire sequences. As the possibility to aggregate vectors in all different ways in both sequences is quite large, efficient algorithms are needed. Here, we show that dynamic programming-based algorithm can be designed to solve this optimization problem in an efficient way in a time essentially proportional to the product of the two input sequence lengths.

2 Formalization

We consider $\mathbb{R}^Q$ as a metric space with a euclidean distance D. Let $\mathbf{X} = \{\mathbf{x}_i\}_{1\leq i\leq I}$ be a sequence of vectors in $\mathbb{R}^Q$, where $\mathbf{x}_i = [\,x_i^1 \;\cdots\; x_i^q \;\cdots\; x_i^Q\,]^T$ is the component representation of vector $\mathbf{x}_i$ in some reference basis.

We denote $\mathbf{X}_{i_1}^{i_2}$ the subsequence $\{\mathbf{x}_i\}_{i_1 \leq i \leq i_2}$ extracted from $\mathbf{X}$ and $\bar{\mathbf{X}}_{i_1}^{i_2}$ the vector obtained by aggregating the components of sequence $\mathbf{X}_{i_1}^{i_2}$ in a new vector:

$$\bar{\mathbf{X}}_{i_1}^{i_2} = \sum_{i=i_1}^{i_2} \mathbf{x}_i = [\sum_{i=i_1}^{i_2} x_i^1 \ \cdots \ \sum_{i=i_1}^{i_2} x_i^q \ \cdots \ \sum_{i=i_1}^{i_2} x_i^Q\,]^T. \tag{1}$$

Note that with this aggregative definition, the aggregation of two consecutive aggregated sequences $\bar{\mathbf{X}}_{i_1}^{i_2}$ and $\bar{\mathbf{X}}_{i_2+1}^{i_3}$ is the aggregation of the concatenated sequence:

$$\bar{\mathbf{X}}_{i_1}^{i_2} + \bar{\mathbf{X}}_{i_2+1}^{i_3} = \bar{\mathbf{X}}_{i_1}^{i_3}. \tag{2}$$

Mappings: valid and partial valid mappings Let $\mathbf{Y} = \{\mathbf{y}_j\}_{1 \leq j \leq J}$ be a reference sequence. We consider mappings $M \subset [1 \cdots I] \times [1 \cdots J]$ between the indexes of $\mathbf{X}$ and $\mathbf{Y}$ such that the following constrains are verified:

i *Preservation of ancestrality*: if (i_1, j_1) and (i_2, j_2) are both in M, then,

$$i_1 \leq i_2 \Leftrightarrow j_1 \leq j_2.$$

ii *Proper split and merge*: if (i_1, j_1) and (i_2, j_2) are both in M, then both,

$$i_1 = i_2 \Rightarrow j_1 \neq j_2,$$

$$j_1 = j_2 \Rightarrow i_1 \neq i_2.$$

iii *Block consistency*:

a if (i, j_1) and (i, j_2) are both in M, then, $\forall j'$, such that $j_1 \leq j' \leq j_2$,

$$(i, j') \in M.$$

b if (i_1, j) and (i_2, j) are both in M, then, $\forall i'$, such that $i_1 \leq i' \leq i_2$,

$$(i', j) \in M.$$

The first constraint ensures that the order between elements in the sequences is preserved by the mapping. The second constraint ensures that the mapping can contain only $1 \times n$ or $m \times 1$ local associations, but not $m \times n$ when m and n both > 1. The last constraint ensures that elements in one sequence are mapped onto contiguous group of elements in the other sequence.

Mappings between $\mathbf{X}$ and $\mathbf{Y}$ respecting properties (*i*), (*ii*) and (*iii*) are said to be valid and they form the set of valid mappings $\mathcal{M}(\mathbf{X}, \mathbf{Y})$ (or simply $\mathcal{M}$ when there is no confusion possible). We say that a mapping M between $\mathbf{X}$ and $\mathbf{Y}$ is K-valid if M verifies constraints (*i*), (*ii*) and (*iii*) and the maximum size of the blocks induces by M on both sequences is less than or equal to K Fig. 1.

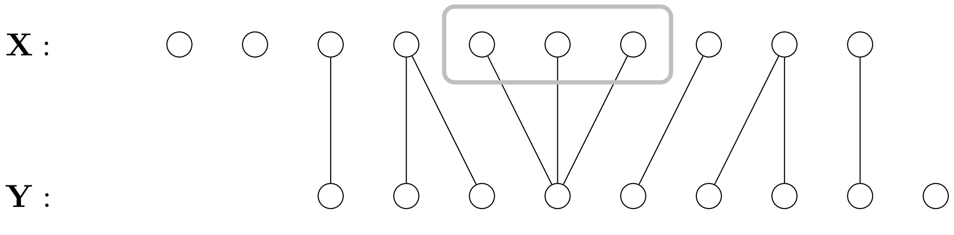

Figure 1: Valid mapping M between two sequences $\mathbf{X}$ and $\mathbf{Y}$. The mapping shows split and merge operations. Merge operations on one sequence correspond to split operations on the other. A merge block of 3 elements is highlighted in $\mathbf{X}$.

For any index i in the sequence $\mathbf{X}$, we note:

- $M(i) = \{j\}$ if $(i, j) \in M$ and i and j both appear only once in M,

- $M(i) = \{j_1, \cdots, j_n\}$ if i appears n times in M, i.e. $(i, j_1), (i, j_2), \cdots, (i, j_n)$ are all in M.
- $M(i) = \emptyset$ if i does not appear in M.

And a symmetrical notation holds for any index j in the sequence $\mathbf{Y}$:

- $M(j) = \{i\}$ if $(i, j) \in M$ and i and j both appear only once in M,
- $M(j) = \{i_1, \cdots, i_n\}$ if j appears n times in M, i.e. $(i_1, j), (i_2, j), \cdots, (i_n, j)$ are all in M.
- $M(j) = \emptyset$ if j does not appear in M.

The set of partial mappings $\mathcal{M}[i, j]$ is the set of valid mappings $\mathcal{M}(\mathbf{X}_1^i, \mathbf{Y}_1^j)$ between subsequences $\mathbf{X}_1^i$ and $\mathbf{Y}_1^j$ of $\mathbf{X}$ and $\mathbf{Y}$ respectively.

The transpose of a mapping M is an element of $\mathcal{M}(\mathbf{Y}, \mathbf{X})$ defined as:

$$M^T = \{(j, i) | (i, j) \in M\}.$$

Let L be a non-negative integer. We denote $\mathcal{M}^L(\mathbf{X}, \mathbf{Y}) \subset \mathcal{M}(\mathbf{X}, \mathbf{Y})$ the set of mapping of size $\#M$ greater or equal to L.

Split and merge Let M be a mapping between $\mathbf{X}$ and $\mathbf{Y}$. We define the set of splits of M (resp. the set of aligned and the set of unmatched) as the set of indexes:

$$\begin{aligned} s_M(\mathbf{X}) &= \{i \in I |\ \#M(i) > 1\}, \\ a_M(\mathbf{X}) &= \{i \in I |\ \#M(i) = 1\}, \\ u_M(\mathbf{X}) &= \{i \in I |\ \#M(i) = 0\}. \end{aligned} \tag{3}$$

Using the following definition of mapping subsets:

$$\begin{aligned} M_s &= \{(i, j) \in I |\ i \in s_M(\mathbf{X}), \} \\ M_a &= \{(i, j) \in I |\ i \in a_M(\mathbf{X})\}, \end{aligned} \tag{4}$$

and merges are defined as splits of the transpose mapping:

$$M_m = {M^T}_s.$$

With these notations, M can be decomposed into split (s), merge (m), aligned (a) sets:

$$M = M_s \cup M_m \cup M_a. \tag{5}$$

The set of elements unmatched by M in $\mathbf{X}$ (resp. $\mathbf{Y}$) is $u_M(\mathbf{X})$ (resp. $u_M(\mathbf{Y})$)

Cost of mappings Based on the above decomposition, a cost can be associated with each mapping:

$$C(M) = \underbrace{\sum_{i \in s_M(\mathbf{X})} C(i, M(i))}_{split} + \underbrace{\sum_{j \in s_{M^T}(\mathbf{Y})} C(M(j), j)}_{merged} + \underbrace{\sum_{i \in a_M(\mathbf{X})} C(i, M(i))}_{aligned} + \underbrace{\sum_{i \in u_M(\mathbf{X})} C(i)}_{deleted} + \underbrace{\sum_{j \in u_M(\mathbf{Y})} C(j)}_{inserted}, \tag{6}$$

where:

$$\begin{aligned} C(i, M(i)) &= C(i, [j_1, \cdots, j_n]) \\ &= D(\mathbf{x}_i, \bar{\mathbf{Y}}_{j_1}^{j_n}) \\ &= D(\mathbf{x}_i, \sum_{k=1}^{n} \mathbf{y}_{j_k}), \end{aligned} \tag{7}$$

and

$$C(i) = 0. \tag{8}$$

Comparison problem Our aim is to compare a test sequence $\mathbf{X}$ with a reference sequence $\mathbf{Y}$ by allowing local aggregations of variables in the mapping between $\mathbf{X}$ and $\mathbf{Y}$ and consider in such case that the aggregated elements are equivalent to a unique element with attributes corresponding to the aggregation of the aggregated elements' attributes. The local cost between aggregated elements and their mapped element is then defined as the local distance between the aggregated attributes and the attributes of the mapped element Equ. 7. We define the total cost of such a mapping as the cumulated score of local distances between mapped elements Equ. 6

We thus want to find a mapping M^* with a given minimal size L and with minimal cost among all the possible valid mappings between $\mathbf{X}$ and $\mathbf{Y}$ of minimal size L:

$$M^* = \underset{M \in \mathcal{M}^L(\mathbf{X},\mathbf{Y})}{\text{argmin}} \; C(M) \tag{9}$$

Theorem 1. *Problem 9 can be solved in time complexity $O(IJK)$, where I and J are respectively the length of the two compared sequences and K is the maximum block size of an aggregation.*

Proof. Let us first remark that the set $\mathcal{M}^L[i,j]$ of valid mappings between the truncated sequences $\mathbf{X}_1^i$ and $\mathbf{Y}_1^j$ can be recursively expressed as a mapping between the last nodes $\mathbf{X}_1^i$ and $\mathbf{Y}_1^j$ and a mapping from sets $\mathcal{M}^L[i-1, j-k]$ or $\mathcal{M}^L[i-k, j-1]$ for $k = 1 \cdots K$. For this, let us define:

$$\begin{aligned} \mathcal{M}^L_{1,k}[i,j] &= \big\{(i,j),(i,j-1),\cdots,(i,j-k+1)\big\} \cup \mathcal{M}^L[i-1,j-k] \\ \mathcal{M}^L_{k,1}[i,j] &= \big\{(i,j),(i-1,j),\cdots,(i-k+1,j)\big\} \cup \mathcal{M}^L[i-k,j-1]. \end{aligned} \tag{10}$$

The union of these sets for $k = 1 \cdots K$ defines a partition of $\mathcal{M}^L[i,j]$:

$$\mathcal{M}^L[i,j] = \bigcup_{k=1}^{K} \mathcal{M}^L_{1,k}[i,j] \cup \bigcup_{k=1}^{K} \mathcal{M}^L_{k,1}[i,j] \tag{11}$$

The cost of a mapping M in $\mathcal{M}^L_{1,k}[i,j]$ is thus:

$$\begin{aligned} C(M) &= C(\big\{(i,j),(i,j-1),\cdots,(i,j-k+1)\big\} \cup M') \\ &= C(\big\{(i,j),(i,j-1),\cdots,(i,j-k+1)\big\}) + C(M') \\ &= D(\mathbf{x}_i, \sum_{h=1}^{k} \mathbf{y}_{j-h+1}) + C(M'), \end{aligned} \tag{12}$$

where M' is a mapping from $\mathcal{M}^L_{1,k}[i-1,j-k]$. The passage from the first to the second line comes from the fact that the cost of the mapping is simply the sum over the split/merge and exactly aligned elements (Equ. 6).

Likewise for a mapping M in $\mathcal{M}^L_{k,1}[i,j]$, there exists a mapping M' in $\mathcal{M}^L_{k,1}[i-k,j-1]$ such that:

$$\begin{aligned} C(M) &= C(\big\{(i,j),(i-1,j),\cdots,(i-k+1,j)\big\}) + C(M') \\ &= D(\sum_{h=1}^{k} \mathbf{x}_{i-h+1}, \mathbf{y}_j) + C(M'). \end{aligned} \tag{13}$$

Let us find a mapping M in $\mathcal{M}^L[i,j]$ with minimal cost and show that this computation can be done efficiently in a recursive manner by exploiting the dynamic programming principle.

$$\begin{aligned} M^*[i,j] &= \underset{M \in \mathcal{M}^L[i,j]}{\text{argmin}} \; C(M) \\ &= \underset{\substack{M \in \mathcal{M}^L_{k,1}[i,j] \cup \mathcal{M}^L_{1,k}[i,j] \\ k=1..K}}{\text{argmin}} \; C(M) \end{aligned} \tag{14}$$

This problem can thus be reduced to finding first minimal mappings in each of the subsets $\mathcal{M}^L_{k,1}[i,j]$ and $\mathcal{M}^L_{1,k}[i,j]$ for every $k = 1 \cdots K$ and then take the mapping with minimal cost:

$$M^*[i,j] = \text{argmin} \begin{cases} \min_{\substack{M \in \mathcal{M}^L_{k,1}[i,j] \\ k=1..K}} C(M) \\ \min_{\substack{M \in \mathcal{M}^L_{1,k}[i,j] \\ k=1..K}} C(M), \end{cases} \tag{15}$$

leading to:

$$M^*[i,j] = \operatorname{argmin} \begin{cases} \min_{k=1..K}\{D(\sum_{h=1}^{k}\mathbf{x}_{i-h+1},\mathbf{y}_j) + \min_{M'\in\mathcal{M}^L[i-k,j-1]} C(M')\} \\ \min_{k=1..K}\{D(\sum_{h=1}^{k}\mathbf{x}_i,\mathbf{y}_{j-h+1}) + \min_{M'\in\mathcal{M}^L[i-1,j-k]} C(M')\}, \end{cases} \tag{16}$$

and finally:

$$M^*[i,j] = \operatorname{argmin} \begin{cases} \min_{k=1..K}\{D(\sum_{h=1}^{k}\mathbf{x}_{i-h+1},\mathbf{y}_j) + C(M^*[i-k,j-1])\} \\ \min_{k=1..K}\{D(\sum_{h=1}^{k}\mathbf{x}_i,\mathbf{y}_{j-h+1}) + C(M^*[i-1,j-k])\}. \end{cases} \tag{17}$$

showing that the computation of $M^*[i,j]$ can be carried out recursively.

Our result then derives from the fact that:

$$\begin{aligned} M^* &= \operatorname*{argmin}_{M\in\mathcal{M}^L(\mathbf{X},\mathbf{Y})} C(M) \\ &= M^*[I,J] \end{aligned} \tag{18}$$

□

References

[1] Bellman, R. *Dynamic programming*, vol. 42. Press, Princeton Univ., 1957.

[2] Levinson, S. E. Structural methods in automatic speech recognition. *Proceedings of the IEEE 73*, 11 (11 1985), 1625 – 1650.

[3] Polanski, A., and Kimmel, M. *Bioinformatics.* Springer-Verlag, 2007.

[4] Sakoe, H., and Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustic, Speech, and Signal processing ASSP-26*, 1 (1978), 43 – 49.