Range Reporting for Time Series via Rectangle Stabbing*

Lotte Blank[1] and Anne Driemel[1]

1 Department of Computer Science, University of Bonn, Germany

Abstract

We study the Fréchet queries problem. It is a data structure problem, where we are given a set S of n polygonal curves and a distance threshold ρ. The data structure should support queries with a polygonal curve q for the elements of S, for which the continuous Fréchet distance to q is at most ρ. We study the case that the ambient space of the curves is 1-dimensional and show an intimate connection to the well-studied rectangle stabbing problem. Using known data structures for rectangle stabbing or orthogonal range searching this directly leads to a data structure with size in $\mathcal{O}(n \log^{t-1} n)$ and query time in $\mathcal{O}(\log^{t-1} n + k)$, where k denotes the output size and t can be chosen as the maximum number of vertices of either (a) the stored curves or (b) the query curves. Note that we omit factors depending on the complexity of the curves that do not depend on n.

Related Version arXiv:2401.03762

1 Introduction

The *Fréchet distance* is a popular measure of similarity of two curves q and s. We focus on a data structuring problem which we refer to as the *Fréchet queries problem*. Here, in the preprocessing phase, we are given a set S of n polygonal curves of complexity at most t_s, a distance threshold ρ, and the complexity t_q of the query time series. The task is to store this set in a data structure that can answer the following type of queries efficiently: For a polygonal curve q of complexity t_q, output all curves in S that have Fréchet distance at most ρ to q. We denote with the *complexity* of a curve the number of vertices that defines it. Afshani and Driemel [2] studied this problem in 2018 for 2-dimensional curves providing non-trivial upper and lower bounds for the exact case. Their data structure is based on multi-level partition trees using semi-algebraic range searching and has size in $\mathcal{O}\left(n(\log\log n)^{\mathcal{O}({t_s}^2)}\right)$ and uses query time in $\mathcal{O}\left(\sqrt{n} \cdot \log^{\mathcal{O}({t_s}^2)} n + k\right)$, where k is the output size and t_s and t_q are assumed to be constant. Recently, Cheng and Huang [6] have generalized their approach for higher dimensions. Other works on variants of this problem have focused on the approximate setting [4, 7, 8, 9, 10]. We study the exact setting and—following previous work by Bringmann, Driemel, Nusser and Psarros [4] and Driemel and Psarros [8]—we restrict the ambient space of the curves to be 1-dimensional, that is, they are time series.

Preliminaries For any two points $p_1, p_2 \in \mathbb{R}^d$, $\overline{p_1 p_2}$ is the directed line segment from p_1 to p_2. The linear interpolation of each pair of consecutive vertices of a sequence of vertices $s_1, \dots, s_{t_s} \in \mathbb{R}^d$ is called a polygonal curve. This curve is also denoted as $\langle s_1, \dots, s_{t_s} \rangle$. We can represent polygonal curves as functions $s : [1, t_s] \to \mathbb{R}^d$, where $s(i+\alpha) = (1-\alpha)s_i + \alpha s_{i+1}$

* This work was funded by 390685813 (Germany's Excellence Strategy – EXC-2047/1: Hausdorff Center for Mathematics); 416767905; and the Deutsche Forschungsgemeinschaft (DFG, German Research-Foundation) – 459420781 (FOR AlgoForGe)

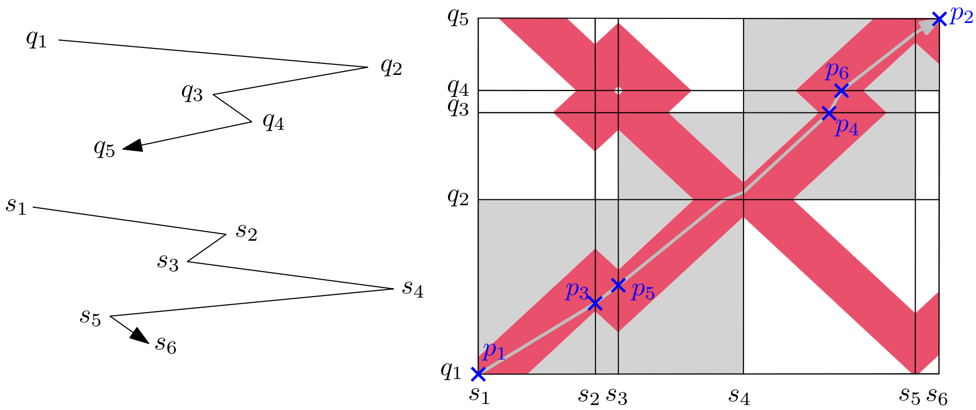

Figure 1 The free space diagram $F_\rho(q, s)$ of two time series with a feasible path trough a feasible sequence of cells $\mathcal{C} = ((1, 1), (1, 2), (1, 3), (2, 3), (2, 4), (3, 4), (4, 4), (4, 5))$, which is drawn in grey. Predicates $(P_1), (P_2), (P_3(1, 2)), (P_4(3, 4)), (P_5(1, 2, 3))$ and $(P_6(3, 4, 4))$ are true, because the points p_i are contained in the free space.

for $i \in \{1, \dots, t_s - 1\}$ and $\alpha \in [0, 1]$. The *(continuous) Fréchet distance* between polygonal curves $q : [1, t_q] \to \mathbb{R}^d$ and $s : [1, t_s] \to \mathbb{R}^d$ is defined as

$$d_{\mathrm{F}}(q, s) = \inf_{h_q \in \mathcal{F}_q, h_s \in \mathcal{F}_s} \max_{p \in [0,1]} \|q(h_q(p)) - s(h_s(p))\|_2,$$

where $\mathcal{F}_q$ is the set of all continuous, non-decreasing functions $h_q : [0, 1] \to [1, t_q]$ with $h_q(0) = 1$ and $h_q(1) = t_q$, respectively $\mathcal{F}_s$ for s.

We show an intimate connection of the Fréchet queries problem to the following classical problems studied in computational geometry. For *rectangle stabbing*, a set S of n axis-aligned d-dimensional rectangles in $\mathbb{R}^d$ needs to be preprocessed into a data structure so that all rectangles in S containing a query point q can be reported efficiently, ensuring that each such rectangle is reported exactly once. *Orthogonal range searching* is its dual. Here, a set S of n points in $\mathbb{R}^d$ is preprocessed into a data structure so that for a d-dimensional axis-aligned query rectangle R all points contained in S can be reported efficiently, ensuring that each such point is reported exactly once.

2 Predicates for Evaluating the Fréchet distance

In this section, we review the predicates used by Afshani and Driemel and how they enable the evaluation of the Fréchet distance in a data structure context. For this, we first recall the definition of the free space diagram from Alt and Godau [3]. For time series $q : [1, t_q] \to \mathbb{R}$ and $s : [1, t_s] \to \mathbb{R}$, the set $F_\rho(q, s) := \{(x, y) \in [1, t_q] \times [1, t_s] \mid |q(x) - s(y)| \leq \rho\}$ is called *free space diagram*. Refer to Figure 1 for an example. They showed that there exists a path in $F_\rho(q, s)$ from $(1, 1)$ to (t_q, t_s) which is monotone in both coordinates if and only if $d_F(q, s) \leq \rho$. For such a path, we say it is *feasible*.

We can decompose the rectangle $[1, t_q] \times [1, t_s]$ into $(t_q - 1) \cdot (t_s - 1)$ cells such that the cell $C_{ij} = [i, i+1] \times [j, j+1]$ corresponds to the part in the free space diagram defined by the edges $\overline{q_i q_{i+1}}$ and $\overline{s_j s_{j+1}}$. By definition of the free space diagram, it follows that $C_{ij} \cap F_\rho(q, s)$ lies between two parallel lines. Therefore, we focus on the boundary of the cells C_{ij}.

Our query algorithm will iterate over all possibilities of sequences of cells that a feasible path could traverse in the free space diagram. In light of this, we call a sequence of cells

$\mathcal{C} = ((i_1, j_1), \ldots, (i_t, j_t))$ *valid*, if $(i_1, j_1) = (1, 1)$, $(i_t, j_t) = (t_q - 1, t_s - 1)$, and $(i_{m+1}, j_{m+1}) \in \{(i_m, j_m + 1), (i_m + 1, j_m)\}$ for all $m < t$. The tuple (i, j) represents the cell C_{ij}. Further, a valid sequence of cells is called *feasible* in $F_\rho(q, s)$, if there exists a feasible path in $F_\rho(q, s)$ traversing exactly the cells in $\mathcal{C}$. The following predicates due to Afshani and Driemel [2] can be used to decide whether a valid sequence of cells is feasible in $F_\rho(q, s)$. See Figure 1 for an example.

(P_1) *(Endpoint (start))* This predicate is true iff $|s_1 - q_1| \leq \rho$.
(P_2) *(Endpoint (end))* This predicate is true iff $|s_{t_s} - q_{t_q}| \leq \rho$.
($P_3(i, j)$) *(Vertex of s - edge of q)* This predicate is true iff $\exists\, p_3 \in \overline{q_i q_{i+1}}$ s.t. $|p_3 - s_j| \leq \rho$.
($P_4(i, j)$) *(Vertex of q - edge of s)* This predicate is true iff $\exists\, p_4 \in \overline{s_j s_{j+1}}$ s.t. $|p_4 - q_i| \leq \rho$.
($P_5(i, j, k)$) *(Monotone in q)* This predicate is true iff $\exists\, p_3, p_5 \in \overline{q_i q_{i+1}}$ s.t. p_3 lies not after p_5 on the time series q and $|p_3 - s_j| \leq \rho$ and $|p_5 - s_k| \leq \rho$.
($P_6(i, l, j)$) *(Monotone in s)* This predicate is true iff $\exists\, p_4, p_6 \in \overline{s_j s_{j+1}}$ s.t. p_4 lies not after p_6 on the time series s and $|p_4 - q_i| \leq \rho$ and $|p_6 - q_l| \leq \rho$.

The following lemma verifies that the predicates can be used to test if the Fréchet distance between two curves is at most a given value.

▶ **Lemma 2.1** (Afshani and Driemel [2]). *Let $\mathcal{C} = ((i_1, j_1), (i_2, j_2), \ldots, (i_t, j_t))$ be a valid sequence of cells. Then $\mathcal{C}$ is feasible in $F_\rho(q, s)$ if and only if the following predicates defined by q, s and ρ are true: (P_1) and (P_2) and ($P_3(i, j)$) if $(i, j - 1), (i, j) \in \mathcal{C}$ and ($P_4(i, j)$) if $(i - 1, j), (i, j) \in \mathcal{C}$ and ($P_5(i, j, k)$) if $(i, j - 1), (i, k) \in \mathcal{C}$ for $j < k$ and ($P_6(i, l, j)$) if $(i - 1, j), (l, j) \in \mathcal{C}$ for $i < l$.*

3 Simplification of the Predicates

Given a sequence of cells $\mathcal{C}$ and a time series s, we want to find intervals $I_1, \ldots, I_{t_q}$ such that $\mathcal{C}$ is feasible in $F_\rho(q, s)$ if and only if $q_i \in I_i$ for all i, where $q = \langle q_1, \ldots, q_{t_q} \rangle$ is a time series with some additional properties. The intervals will be defined using the predicates. Lemma 2.1 shows which predicates need to be true such that $\mathcal{C}$ is feasible in $F_\rho(q, s)$.

▶ **Lemma 3.1.** *Let $q = \langle q_1, \ldots, q_{t_q} \rangle$ and $s = \langle s_1, \ldots, s_{t_s} \rangle$ be two time series. Then the following holds for the predicates in the free space diagram $F_\rho(q, s)$:*

(i) (P_1) *is true* $\Leftrightarrow q_1 \in [s_1 - \rho, s_1 + \rho]$,
(ii) (P_2) *is true* $\Leftrightarrow q_{t_q} \in [s_{t_s} - \rho, s_{t_s} + \rho]$,
(iii) ($P_3(i, j)$) *is true* $\Leftrightarrow$ *if* $q_i \leq q_{i+1}$: $q_i \leq s_j + \rho$ *and* $q_{i+1} \geq s_j - \rho$ *and*
if $q_i \geq q_{i+1}$: $q_i \geq s_j - \rho$ *and* $q_{i+1} \leq s_j + \rho$,
(iv) ($P_4(i, j)$) *is true* $\Leftrightarrow q_i \in [\min\{s_j - \rho, s_{j+1} - \rho\}, \max\{s_j + \rho, s_{j+1} + \rho\}]$,
(v) ($P_5(i, j, k)$ *is true* $\Leftrightarrow$ ($P_3(i, j)$) *and* ($P_3(i, k)$) *are true and one of the following holds:*
- $|s_j - s_k| \leq 2\rho$, *or*
- $|s_j - s_k| > 2\rho$ *and* $s_j \leq s_k$ *and* $q_i \leq s_j + \rho$ *and* $q_{i+1} \geq s_k - \rho$, *or*
- $|s_j - s_k| > 2\rho$ *and* $s_j > s_k$ *and* $q_i \geq s_j - \rho$ *and* $q_{i+1} \leq s_k + \rho$.

To determine the truth value of the monotone in s predicates (P_6), we introduce the new concept of *forward* and *backward numbers* $f_i(q)$ and $b_i(q)$. Here, we take advantage of the fact that the direction of each edge of a time series can only be orientated forward or backward with respect to the x-axis. Refer to Figure 2 as an example.

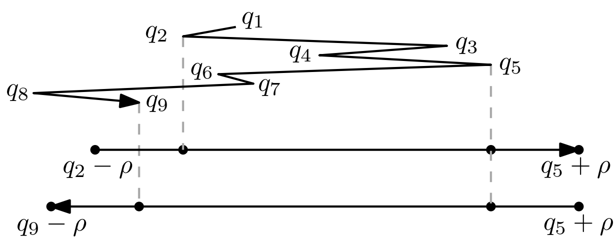

Figure 2 Illustration of the values $f_2(s) = 5$ and $b_5(s) = 9$ for a time series s.

▶ **Definition 3.2** (forward and backward numbers)**.** For a time series $q = \langle q_1, \dots, q_{t_q} \rangle$ and $i \in \{1, \dots, t_q\}$, we denote by the *forward number* $f_i(q) \leq t_q$ (resp. *backward number* $b_i(q) \leq t_q$) the highest number such that $\langle q_i - \rho, q_{f_i(q)} + \rho \rangle$ (resp. $\langle q_i + \rho, q_{b_i(q)} - \rho \rangle$) is oriented forward (resp. backward) and its Fréchet distance to the time series $\langle q_i, \dots, q_{f_i(q)} \rangle$ (resp. $\langle q_i, \dots, q_{b_i(q)} \rangle$) is at most ρ, i.e.,

$$f_i(q) := \max\{k \in \{i, \dots, t_q\} \mid d_{\mathrm{F}}(\langle q_i, \dots, q_k \rangle, \langle q_i - \rho, q_k + \rho \rangle) \leq \rho \text{ and } q_i - \rho \leq q_k + \rho\},$$
$$b_i(q) := \max\{k \in \{i, \dots, t_q\} \mid d_{\mathrm{F}}(\langle q_i, \dots, q_k \rangle, \langle q_i + \rho, q_k - \rho \rangle) \leq \rho \text{ and } q_i + \rho \geq q_k - \rho\}.$$

Note, that for all $i \leq x \leq f_i(q)$, it holds that $d_{\mathrm{F}}(\langle q_i, \dots, q_x \rangle, \langle q_i - \rho, q_x + \rho \rangle) \leq \rho$ and $q_i - \rho \leq q_x + \rho$. Respectively, for $b_i(q)$. The next lemma shows how the forward and backward numbers can be used to determine values of the predicates (P_6). To decide whether a valid sequence of cells is feasible or not in $F_\rho(q, s)$, predicate $(P_6(i, l, j))$ needs to be true only if all predicates $(P_6(x, y, j))$ need to be true with $i \leq x < y \leq l$ by Lemma 2.1.

▶ **Lemma 3.3.** *Let $q = \langle q_1, \dots, q_{t_q} \rangle$ and $s = \langle s_1, \dots, s_{t_s} \rangle$ be time series, $i, l \in \{1, \dots, t_q\}$ with $i < l$ and $j \in \{1, \dots, t_s - 1\}$. If $s_j \leq s_{j+1}$ (resp. $s_j \geq s_{j+1}$), then $(P_6(x, y, j))$ is true $\forall i \leq x < y \leq l$ if and only if $f_i(q) \geq l$ (resp. $b_i(q) \geq l$) and $(P_4(x, j))$is true $\forall i \leq x \leq l$.*

4 Data Structure

In this section, we present two data structures solving the Fréchet queries problem. We start with some assumptions, that can be made for the time series. Let $s = \langle s_1, \dots, s_t \rangle$ be a time series. Then, we can assume that either $s_{2j-1} \leq s_{2j} \geq s_{2j+1}$ for all j (*M-shaped*), or $s_{2j-1} \geq s_{2j} \leq s_{2j+1}$ for all j (*W-shaped*). Moreover, we can assume that the complexity of all time series in S is exactly t_s by simply adding dummy vertices in the end otherwise.

By Lemma 2.1, a sequence of cells $\mathcal{C}$ is feasible in the free space diagram $F_\rho(q, s)$ if and only if the predicates in Lemma 2.1 defined by $\mathcal{C}$ are true. The truth assignment of all predicates $(P_1), (P_2), (P_3), (P_4)$ and (P_5) can be determined using intervals defined by s and ρ. Furthermore, $\mathcal{C}$ can only be feasible in $F_\rho(q, s)$ if for all $(i-1, j), (l, j) \in \mathcal{C}$ with $i \leq l$, the monotone in s predicate $(P_6(i, l, j))$ is true. By Lemma 3.3, we can use the forward number $f_i(q)$ in the case that $s_j \leq s_{j+1}$ (i.e., j is odd if s is M-shaped) to determine whether $(P_6(i, l, j))$ is true. We define the *forward number* $f_i(\mathcal{C})$ as the highest such number l that is needed for $\mathcal{C}$ to be feasible in $F_\rho(q, s)$. Respectively, if $s_j \geq s_{j+1}$ (i.e., j is even if s is M-shaped) for $b_i(q)$ and we define the *backward number* $b_i(\mathcal{C})$. Formally, we get

$$f_i(\mathcal{C}) = \begin{cases} l \geq i, & \text{if } \exists\ (i-1, j), (l, j) \in \mathcal{C} \text{ s.t. } j \text{ is odd and } (l+1, j) \notin \mathcal{C}, \\ i, & \text{otherwise;} \end{cases}$$

$$b_i(\mathcal{C}) = \begin{cases} l \geq i, & \text{if } \exists\ (i-1, j), (l, j) \in \mathcal{C} \text{ s.t. } j \text{ is even and } (l+1, j) \notin \mathcal{C}, \\ i, & \text{otherwise.} \end{cases}$$

As $\mathcal{C}$ is valid there exists a unique j such that $(i-1,j),(i,j),\dots,(l,j) \in \mathcal{C}$. Hence, the numbers $f_i(\mathcal{C})$ and $b_i(\mathcal{C})$ are well-defined.

The Data structure. Let S_M be the set of stored time series that are M-shaped and S_W the set of those that are W-shaped. We will describe how S_M is stored. The time series in S_W are stored in the same way after they were mirrored at the origin. Consequently, for those the query algorithm mirrors the query time series q at the origin and is then the same as for the time series in S_M. For all valid sequences of cells $\mathcal{C}$, we build two associated rectangle stabbing data structures storing the time series in S_M as t_q-dimensional axis-aligned rectangles. One for the case that the query time series q is M-shaped and the other one for the case that q is W-shaped. Knowing the shape of q, Lemma 2.1 and 3.1 define for every $s \in S_M$ an interval for every vertex q_i of the query time series in which it must lie such that $\mathcal{C}$ can be feasible in $F_\rho(q,s)$. For a time series s, we store the Cartesian product of those t_q intervals in the associated rectangle stabbing data structure. Note that even if the complexity of the stored time series is greater than t_q, we store only a t_q-dimensional rectangle for it.

The Query Algorithm. Let q be a query time series of complexity t_q. The query algorithm starts with computing the numbers $f_1(q),\dots,f_{t_q}(q),b_1(q),\dots,b_{t_q}(q)$. For all valid sequences of cells $\mathcal{C}$, we check whether $f_i(\mathcal{C}) \leq f_i(q)$ and $b_i(\mathcal{C}) \leq b_i(q)$ for all i. If so, we do a query search in the rectangle stabbing data structure depending on $\mathcal{C}$ and the shape of q with the point $(q_1,q_2,\dots,q_{t_q})$ and output all time series associated with a rectangle containing this point.

▶ **Theorem 4.1.** *The Fréchet queries problem for constant parameters $t_q \geq 2$ and t_s can be solved with a data structure of size in $\mathcal{O}(n \log^{t_q-2} n)$ and query time in $\mathcal{O}(\log^{t_q-1} n + k)$, where k is the size of the output (without duplicates).*

Proof. The number of valid sequences of cells is constant and the forward and backwards numbers can be computed in constant time because t_s and t_q are constant. So, the size and the query time of the data structure follow by using the rectangle stabbing data structure by Afshani, Arge and Larsen [1]. The correctness follows by the discussion above and the fact that there exists a feasible valid sequences of cells in $F_\rho(q,s)$ if and only if $d_F(q,s) \leq \rho$. ◀

Using an orthogonal range searching data structure, it is possible to store the time series in S as t_s-dimensional points and the query time series defines t_s-dimensional axis-aligned rectangles depending on $\mathcal{C}$. The data structure by Afshani, Arge and Larsen [1] leads to the following.

▶ **Corollary 4.2.** *The Fréchet queries problem for constant parameters t_q and $t_s > 2$ can be solved with a data structure of size in $\mathcal{O}\left(n(\log n/\log\log n)^{t_s-1}\right)$ that uses query time in $\mathcal{O}(\log n(\log n/\log\log n)^{t_s-3} + k)$, where k is the size of the output (without duplicates).*

Known lower bounds for rectangle stabbing and orthogonal range searching by Afshani, Arge and Larsen [1] and by Chazelle [5] can be applied to the Fréchet queries problem, because those problems can be transformed to it. Consider a data structure that solves the Fréchet queries problem and operates on a pointer machine. If it uses nh space, it must use query time in $\Omega(\log n(\log n/\log h)^{\lfloor t/2\rfloor-2} + k)$. And, if it uses query time in $\mathcal{O}(\log^c n + k)$, where c is a constant, it must use space in $\Omega(n(\log n/\log\log n)^{\lfloor t/2\rfloor-1})$. In both cases, k denotes the size of the output (without duplicates) and $t = \min\{t_q, t_s\}$.

References

1 P. Afshani, L. Arge, and K.G. Larsen. Higher-dimensional orthogonal range reporting and rectangle stabbing in the pointer machine model. In *Proceedings of the 2012 Symposuim on Computational Geometry*, pages 323–338, 2012. doi:10.1145/2261250.2261299.

2 P. Afshani and A. Driemel. On the complexity of range searching among curves. In *Proceedings of the 2018 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 898–917, 2018. doi:10.1137/1.9781611975031.58.

3 H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry and Applications*, 5(01& 02):75–91, 1995. doi:10.1142/S0218195995000064.

4 K. Bringmann, A. Driemel, A. Nusser, and I. Psarros. Tight bounds for approximate near neighbor searching for time series under the Fréchet distance. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 517–550, 2022. doi:10.1137/1.9781611977073.25.

5 B. Chazelle. Lower bounds for orthogonal range searching: I. the reporting case. *Journal of the ACM*, 37(02):200–212, 1990. doi:10.1145/77600.77614.

6 Siu-Wing Cheng and Haoqiang Huang. Solving Fréchet distance problems by algebraic geometric methods. *ArXiv*, abs/2308.14569, 2023. doi:10.48550/arXiv.2308.14569.

7 Mark de Berg, Atlas F. Cook, and Joachim Gudmundsson. Fast Fréchet queries. *Computational Geometry*, 46(6):747–755, 2013. doi:10.1016/j.comgeo.2012.11.006.

8 A. Driemel and I. Psarros. ANN for time series under the Fréchet distance. In *Algorithms and Data Structures*, pages 315–328, 2021. doi:10.1007/978-3-030-83508-8_23.

9 A. Driemel and F. Silvestri. Locality-sensitive hashing of curves. In *33rd International Symposium on Computational Geometry*, volume 77, pages 37:1–37:16, 2017. doi:10.4230/LIPIcs.SoCG.2017.37.

10 A. Filtser, O. Filtser, and M.J. Katz. Approximate nearest neighbor for curves: simple, efficient, and deterministic. *Algorithmica*, 2022. doi:10.1007/s00453-022-01080-1.