6 Q-grams and filtration schemes

Recall the definition of approximate seeds. We consider two arbitrary strings x, y within edit distance k. The result then holds for any substring of the text within distance k from the pattern.

Lemma 1. *Let x, y be two strings with $d_E(x, y) = k$. If y is partitioned w.l.o.g. into s non-overlapping seeds such that $1 \leq s \leq k + 1$, then at least one seed occurs as a factor of x within distance $\lfloor k/s \rfloor$.*

Approximate seeds provide filtration schemes of variable specificity. The fastest but weakest filtration scheme is given by $s = k + 1$, while the most specific filtration is obtained for $s = 1$ i. e. perfect filtration scheme without any verification step.

6.1 Filtration schemes

Alternatively, filtration specificity is controlled by acting on the minimum seed length q. Fixing q yields $s = \lfloor m/q \rfloor$, or vice versa, fixing the number of seeds s gives $q = \lfloor m/s \rfloor$. Filtration specificity is expected to increase with seed length.

Lemma 1 assigns the same distance threshold to all seeds, yet this is not obligatory. Hence, we give a more general definition of *filtration scheme* for approximate seeds.

Definition 2. A seeds filtration scheme is an integer vector $\mathbf{t} = (t_1, \ldots, t_s)$, where integer $t_i \in \mathbb{N}_0$ represents the threshold assigned to the i-th seed.

Lemma 3. *Any filtration scheme $\mathbf{t} = (t_1, \ldots, t_s)$ such that*

$$s + \sum_{i=1}^{s} t_i > k \tag{6.1}$$

is full-sensitive for k-differences (and k-mismatches).

Example 4. The filtration schemes $(0, 0, 0, 0, 0)$, $(1, 1, 0)$, $(2, 1)$, (4) are full-sensitive for 4-differences. For instance, given a pattern of length $m = 100$, q is respectively $20, 33, 50, 100$.

How to choose a *good* filtration scheme in practice? Both Myers and Navarro carried out involved analysis to estimate the optimal parameterization. Navarro finds out that a number of seeds $s = \Theta(\frac{m}{\log_\sigma n})$ yields an overall time complexity sublinear for an error rate $\epsilon < 1 - \frac{e}{\sqrt{\sigma}}$.

Myers reports an analogous sublinear time when $q = \Theta(\log_\sigma n)$ is the seed length. Yet, these results do not necessarily translate into optimal filtration schemes in practice. The parameterization depends on the full-text index, the verification algorithm, the statistical properties of the text. Missing the optimal number of seeds by one often results in a runtime penalty of an order of magnitude.

Having established the number of seeds, or their length, thresholds have to be assigned. Lemma 3 allows to assign arbitrary distance thresholds.

In practice, it is convenient to distribute distance thresholds evenly, as seeds with the highest threshold dominate the overall filtration time. The most strict threshold assignment is to give distance $\lfloor k/s \rfloor$ to (k mod s)+1 seeds and distance $\lfloor k/s \rfloor - 1$ to the remaining seeds.

6.2 Filtration schemes for Q-grams

Before we will go to gapped q-grams (we will call them Q-grams), recall the q-gram counting lemma:

Lemma 5 (The q-gram lemma)**.** *Let x, y be two strings such that $d_E(x, y) = k$, and assume w.l.o.g. $|x| \leq |y|$ and $|x| = m$. Then x and y have q-gram similarity $\tau_q(m, k) \geq m - q + 1 - kq$.*

The first part of the threshold function τ_q counts the number of q-grams of x (i. e. $m - q + 1$), while the second part counts how many q-grams can be covered by k errors (i. e. at most q per error, hence kq in total).

The position of errors in the transcript solely determines which q-gram occurrences are affected or preserved. The q-gram lemma considers one *worst case* positioning of the errors that *minimizes* the threshold.

Lets denote by a pair (q,t) the filtration scheme counting q-grams with threshold t. According to lemma 5, if $t = \tau_q(m,k) \geq 1$, then (q,t) is full-sensitive for any k-differences instance where $|p| = m$. In this case, I say that (q,t) *solves* instance (m,k).

The following question arises: which is the longest q-gram solving instance (m,k)? In order to satisfy lemma 5, the q-gram threshold must be greater than zero, i. e. it must hold $\tau_q(m,k) \geq 1$. Thus, by substituting $\tau_q(m,k)$, it follows that the q-gram length must be $q \leq \left\lfloor \frac{m}{k+1} \right\rfloor$, analogously to seed filters.

However, the longest q-gram does not yield always the most specific filtration scheme.

For instance, a threshold of 1 completely discards the counting argument of lemma 5 and makes filtration very unspecific in practice. Hence, on certain (m,k) instances, filtration schemes with non-optimal q-gram length yield more specific filtration. Example 6 shows alternative filtration schemes solving a given (m,k) instance.

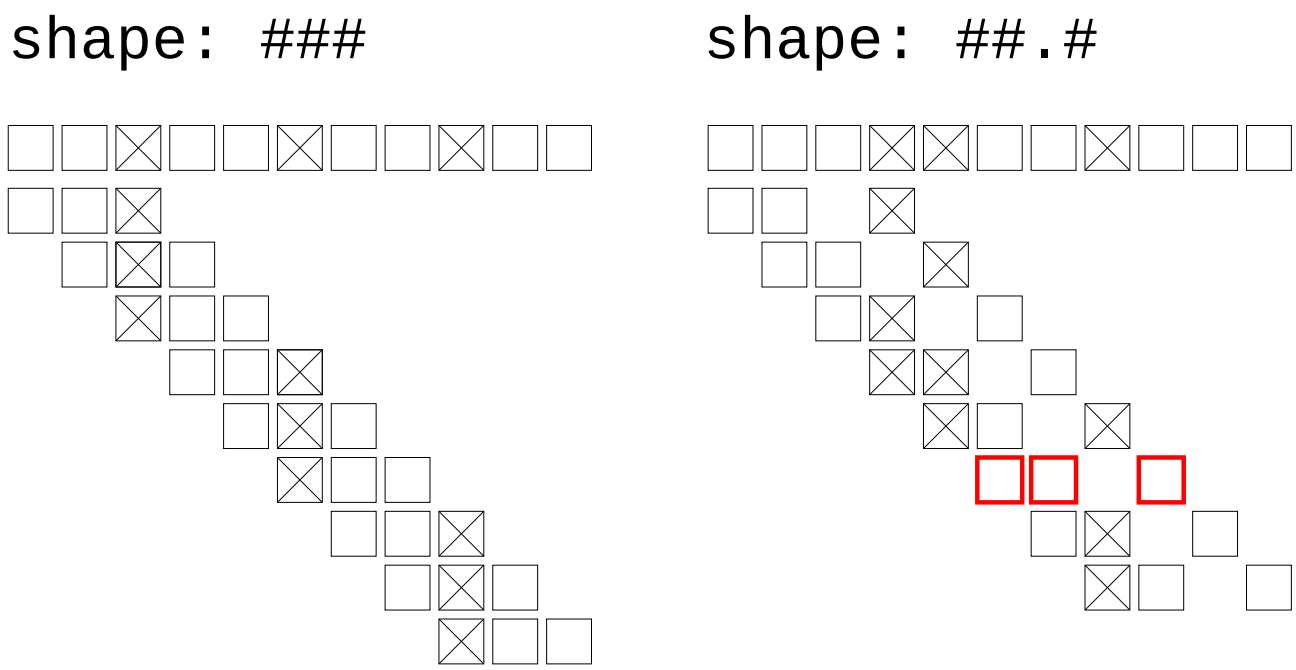
Example 6. The following (q,t) filtration schemes solve $(100,4)$-differences: $(20,1)$, $(19,6)$, $(18,11)$.

Lets turn our attention to Q-grams. A surprisingly high number of research papers have been published on this topic.

Almost all work focuses on lossy filtration for homology search, rather than full-sensitive filtration for approximate string matching. Here, we consider gapped q-grams only in the context of full-sensitive filtration for k-mismatches. This case has been first considered by Burkhardt and Kärkäinen.

Gapped q-grams introduce fixed *don't care positions* where text and pattern characters are ignored. The following example illustrates the advantage of such don't care positions.

Example 7. Let $m = 11$ and $k = 3$ and consider the (gapped) q-grams (also called shape) `###` and `##.#`. The threshold for ungapped q-grams is $0 = 11 - 3 \cdot 4 + 1$ and for the gapped $-1 = 11 - 4 - 3 \cdot 3 + 1$ respectively. Thus neither q-gram would be useful for filtering. However, the real threshold for `##.#` is 1. This can be checked by a full enumeration of all combinations of 3 mismatches.



Worst-case mismatch positions

This example relaxes the full-sensitivity threshold and opens the door to more specific filtration schemes for k-mismatches.

The counting argument of gapped q-grams generalizes q-gram similarity from substrings to *subsequences*, i. e. from contiguous to *non-contiguous* sequences of symbols.

Filtration with gapped q-grams indeed counts the number of subsequences of length q common to two strings. An additional set Q determines which symbols are taken in the subsequences.

Definition 8. A gapped q-gram (abbreviated as Q-gram) is a finite sequence Q of natural numbers starting with 0, i. e. $Q \subset \mathbb{N}_0$ and $0 \in Q$. The cardinality $|Q|$ is called the *weight* of Q and denoted as $w(Q)$. The maximum element of Q+1 is named *span* and indicated by $s(Q)$.

Hence in the above example we had a Q-gram of weight 3 and of span 4. The filtration scheme $(Q, t) = (\{0, 1, 3\}, 1)$ solves instance $(11, 3)$.

As in lemma 5, the threshold for Q-grams still depends on the worst-case positioning of the errors in the transcript, which in turn depends on parameters (m, k).

Thus, we still consider filtration schemes (Q, t) solving a (m, k) instance. However, contrary to contiguous q-grams, function τ_Q now gives only a lower bound to the full-sensitivity threshold.

Which is the most specific filtration scheme (Q, t) solving a given instance (m, k)? This question turns out to be surprisingly hard to answer.

As discussed above, the most specific filtration schemes for contiguous q-grams are easily found. The choice falls on a few values of q that are close to the maximum and do not yield a threshold too close to 1.

Conversely, such choice is non-trivial for Q-grams, as the search space of Q is exponentially large in the span $s(Q)$ and a full-sensitivity threshold for arbitrary Q-grams is hard to compute. In addition, it is not easy to determine which filtration scheme is the most specific one in a set of full-sensitive candidates.

Given a specific Q-gram, we can consider the following problems:

Full Sensitivity	Does filtration scheme $(Q, 1)$ solve instance (m, k)?
Optimal Threshold	Which is the optimal threshold t^* such that (Q, t^*) solves (m, k)?
Specificity	Which is the expected specificity of scheme (Q, t)?

Nicolas et al. considered the decision problem Full Sensitivity associated to Optimal Threshold. Full Sensitivity is easy for contiguous q-grams, i. e. the answer is no iff $\tau_q(m, k) = 0$. Nicolas et al. show, by performing an indirect reduction from Exact Cover by 3-Sets, that Full Sensitivity is *strongly* NP-complete for arbitrary Q-grams. Strong NP-completeness implies that no *fully polynomial-time approximation scheme* (FPTAS) nor any *pseudo-polynomial* algorithm for Full Sensitivity exist, under the assumption that $P \neq NP$.

Burkhardt et al. first considered the optimization problem Optimal Threshold. They give a DP algorithm solving Optimal Threshold in time $O(m \cdot k \cdot 2^{s(Q)})$ for any Q-gram. Subsequently, they use their DP algorithm to explore the search space of full-sensitive Q-grams for some specific instances of (m, k).

Before we discuss the DP algorithm, I will show some results that illustrate how much stronger the thresholds for Q-grams are in comparison to q-grams.

The following table gives the exact thresholds for all Q-grams for $m = 50$ and $k = 5$. One can see that in many cases, especially for higher values of q, best Q-grams have higher thresholds than contiguous shapes of the same or even smaller size.

$s\downarrow$: $q\rightarrow$	4	5	6	7	8	9	10
5	26	21	–	–	–	–	–
6	25	20	15	–	–	–	–
7	24	19	14	9	–	–	–
8	23	18	13	8	3	–	–
9	22	**18** > 17	**14** > 12	**9** > 7	**5** > 2	0	–
10	21	**18** > 16	**13** > 11	**10** > 6	**6** > 1	**3** > 0	0
11	20	**16** > 15	**13** > 10	**10** > 5	**7** > 0	**4** > 0	**2** > 0
12	19	**16** > 14	**12** > 9	**9** > 4	**7** > 0	**4** > 0	**2** > 0

6.3 Computing the threshold

To define the DP recurrence we need some more notation. For a set I of integers let $I \oplus j = \{i + j \mid i \in I\}$. Also we define $\ominus$ similar and $[cond]$ be 1 if condition $cond$ is true and 0 otherwise. The DP is based on the following conditional threshold.

Definition 9. Let Q be a $Q - gram$ with span s. For non-negative integers $m, k \leq m$ and a set $M \subseteq [1, s-1]$, $|M| \geq s - 1 - k$, let $t_Q(m, k, M)$ denote the optimal threshold for pattern length m and Hamming distance k under the additional condition that the matches in the last $s - 1$ positions correspond to M. That is:

$$t_q(m, k, M) = \min_{M' \subseteq [1,m], |M'| = m-k, (M' \ominus (m-s+1)) \cap [1,s-1] = M} |\{i \in [1, m-s+1] | Q_i \subseteq M'\}|$$

The actual optimal threshold can then be computed by minimizing over all possible M, i.e.

$$t_Q(m,k) = \min_{M\subseteq[1,s-1],|M|\geq s-1-k} t_Q(m,k,M).$$

6.4 Computing the threshold

The DP is based on the fact that the values $t_Q(i,*,*)$ can be computed form the values $t_Q(i-1,*,*)$. Note that it is not possible to compute $t_Q(i,*)$ from $t_Q(i-1,*)$; the matches in the last $s-1$ positions have to be recorded as well. The following lemma gives the details:

Lemma 10. *For $s \leq i \leq m, 0 \leq j \leq k$ and $M \subseteq [1, s-1], \mid M \mid \geq s-1-j$, we have*

$$t_Q(s-1,j,M) = 0 \tag{6.2}$$
$$t_Q(i,j,M) = \min\{(1),(2)\} \tag{6.3}$$

with

$$(1):\ t_Q(i-1, j-[s-1\notin M], (M\cup\{0\})\setminus\{s-1\})\oplus 1) + [Q\subseteq(M\cup\{0\})]$$

and

$$(2):\ t_Q(i-1, j-[s-1\notin M], (M\setminus\{s-1\})\oplus 1).$$

Proof in exercise (also see seminar paper).

We will now give an *integer linear program* (ILP) that solves exactly Optimal Threshold and is usually much quicker than the DP algorithm discussed above.

Lets start by modeling the decision problem Full Sensitivity before turning to the optimization problem Optimal Threshold.

We consider any Hamming distance transcript over $\{R, M\}$ as an m-dimensional vector $\mathbf{x} = (x_1, x_2, \ldots x_m)$ over $B = \{0, 1\}$. Accordingly, we denote by $m - \sum \mathbf{x}$ the Hamming distance of the transcript, and by $B_k^m \subset B^m$ the set containing all transcripts $\mathbf{x}$ such that $|\mathbf{x}|_0 = k$.

We now define the event of detecting transcript by a filtration scheme (Q, t).

Definition 11. A Q-gram *occurs* at position i in a transcript $\mathbf{x}$ iff $\forall j \in Q\ x_{i+j} = 1$.

Definition 12. A filtration scheme (Q, t) *detects* $\mathbf{x}$ iff the Q-gram occurs at least t times in $\mathbf{x}$.

We introduce a *boolean function* to characterize the set of transcripts detected by a filtration scheme of the form $(Q, 1)$.

Let $T_Q^m : B^m \to B$ denote the boolean function such that $T_Q^m(\mathbf{x})$ is true iff the Q-gram occurs in a transcript $\mathbf{x}$ of length m.

We define such boolean function as the disjunction

$$T_Q^m(\mathbf{x}) = \bigvee_{i=1}^{m-s(Q)} \bigwedge_{j\in Q} x_{i+j} \tag{6.4}$$

where each *clause* of T_Q^m represents a single possible occurrence of Q in $\mathbf{x}$. According to definition 12, filtration scheme (Q, t) detects $\mathbf{x}$ iff $\mathbf{x}$ satisfies at least t clauses of T_Q^m.

The formal definition of Full Sensitivity follows.

Full Sensitivity
Instance A Q-gram, an (m, k) instance.
Question $\exists\, \mathbf{x} \in B_k^m$ such that $T_Q^m(\mathbf{x}) = 0$?

Lets now consider the *pseudo-boolean function*, counterpart of the above function, that associates a filtration threshold to any transcript. Let the function $t_Q^m : B^m \to \mathbb{N}_0$ be the boolean function T_Q^m acting on $\mathbb{N}_0$. Here, $t_Q^m(\mathbf{x})$ *counts* how many times a Q-gram occurs in a transcript $\mathbf{x}$ of length m.

We define such pseudo-boolean function as

$$t_Q^m(\mathbf{x}) = \sum_{i=1}^{m-s(Q)} \prod_{j\in Q} x_{i+j} \tag{6.5}$$

The formal definition of Optimal Threshold follows:

Optimal Threshold
Instance A Q-gram, an (m, k) instance.
Solution min $t_Q^m(\mathbf{x})$ subject to $\mathbf{x} \in B_k^m$

The following ILP solves Optimal Threshold. We first define binary variables $x_i, \forall 0 \leq i \leq m-1$ and $t_j, \forall 0 \leq j \leq m - s(Q)$. The variable $x_i = 1$ if and only if we have a *mismatch* and similarly we define $t_j = 1$ if and only if the $Q-gram\ Q_j$ is *not* matching. Note that this variable definitions are "flipped." Then we can formulate the ILP as follows:

$$\begin{aligned} &\max && \textstyle\sum \mathbf{t} \\ &\text{subject to} \\ &&& \mathbf{t} \in B^{m-s(Q)+1} \\ &&& \mathbf{x} \in B^m_{m-k} \\ &&& t_i \leq \textstyle\sum_{j \in Q} x_{i+j} \quad \forall\, 0 \leq i \leq m - s(Q) \end{aligned} \tag{6.6}$$

Vector $\mathbf{t}$ represents function T_Q^m and its sum function t_Q^m; each t_i indicates the truthfulness of the i-th clause of T_Q^m, i.e. $t_i = 0$ if the i-th clause is true. Vector $\mathbf{x}$ represents any hamming transcript; each $x_j = 0$ represents a matching position and is subject to an integer linear constraint such that the hamming distance of $\mathbf{x}$ is within k.

The set of inequalities of the form $t_i \leq \sum_{j \in Q} x_{i+j}$ binds the satisfiability of each clause t_i to its associated transcript values x_j. The solution $\mathbf{t}^*$ to the above ILP provides the optimal threshold $t = m - s(Q) + 1 - \sum \mathbf{t}^*$ for a Q-gram on instance (m, k).

Input		Threshold			Runtime [s]	
Q	k	LEMMA	EXACT	APX	ILP	DP
(i)	8	0	3	4	0.30	0.03
(i)	9	0	2	3	1.24	0.05
(i)	10	0	0	2	0.02	0.07
(ii)	8	0	7	9	0.20	31.60
(ii)	9	0	5	6	0.39	93.77
(ii)	10	0	1	3	0.32	244.83
(iii)	8	0	4	6	0.06	223.52
(iii)	9	0	3	4	0.08	885.37
(iii)	10	0	2	2	0.11	–

6.5 Q-gram families

To obtain even more specific filtration, Kucherov et al. propose *Q-gram families* (also known as *multiple gapped q-grams*). Filtration with a Q-gram family adopts disjunctively a set of multiple distinct gapped q-grams. The generalized counting argument now adds all occurrences of all gapped q-gram in the set.

Definition 13. A Q-gram family (abbreviated as $\mathbb{F}$-gram) is a finite set $\mathbb{F} = \{Q_1, \ldots, Q_f\}$ of Q-grams. Its counting threshold $\tau_{\mathbb{F}}$ is defined as:

$$\tau_{\mathbb{F}}(m, k) = \sum_{Q_i \in \mathbb{F}} \tau_{Q_i}(m, k) \tag{6.7}$$

All design problems introduced in the previous section and their solutions naturally generalize to Q-gram families. Lets define a boolean function for an $\mathbb{F}$-gram as the disjunction

$$T_{\mathbb{F}}^m(\mathbf{x}) = \bigvee_{Q_i \in \mathbb{F}} T_{Q_i}^m(\mathbf{x}) \tag{6.8}$$

and a pseudo-boolean function as the sum

$$t_{\mathbb{F}}^m(\mathbf{x}) = \sum_{Q_i \in \mathbb{F}} t_{Q_i}^m(\mathbf{x}). \tag{6.9}$$

6.6 Choosing a specific Q-gram

The filtering efficiency of a Q-gram clearly depends on the threshold $t_Q(m,k)$. Burkhardt et al. proposed the following criterion which correlates to a good specificity. This criterion is called *minimum coverage*.

Before we define it formally lets have a look at an example.

Example 14. Let $m = 13$ and $k = 3$. Then both shapes `###` and `##.#` have a threshold of two. If two strings have four consecutive characters then they have two common 3-grams of shape `###`. In contrast, in order to have two common 3-grams of shape `##.#`, two strings need at least 5 matching characters.

This means, that the gapped 3-gram would have a lower count of common q-grams on strings that have only four consecutively matching characters although it has the same threshold.

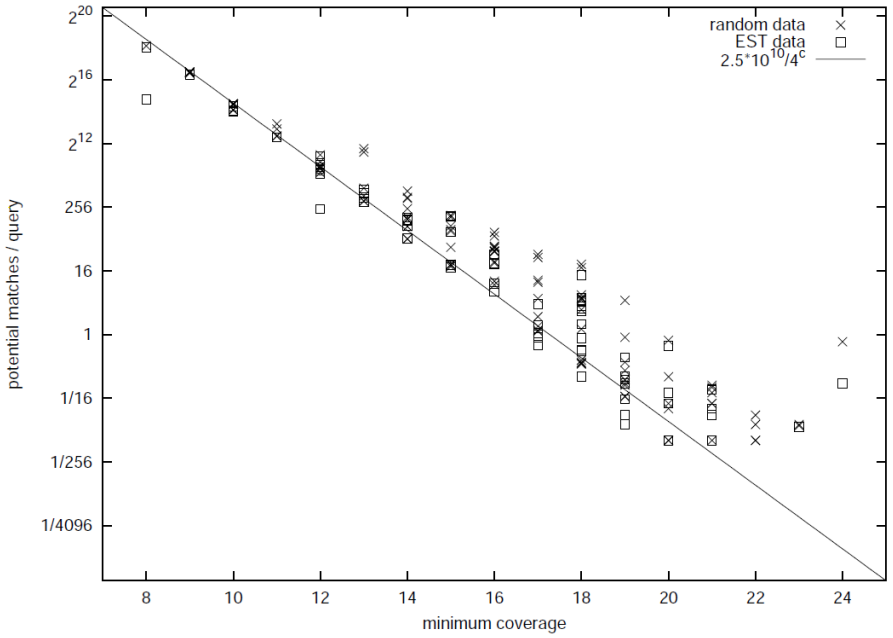
Obviously this makes the gapped $Q - gram$ more specific.

Definition 15. Let Q be a $Q - gram$ and t be a non-negative integer. The *minimum coverage* of Q for threshold t is:

$$c(Q,t) = \min_{C \subset \mathbb{N}, |C|=t} | \cup_{i \in C} Q_i | .$$

Hence the minimum coverage is the minimum number of characters that need to match between a pattern and a text substring for there to be t matching Q-grams.

Computational experiments indicate that there is a strong correlation between the minimum coverage $c(Q, t_Q(m,k))$ and the filter efficiency.



Correlation between expected and actual number of potential matches.

The following table shows different shapes for $k = 5$. The column *best* shows the shape with the highest minimum coverage (ties are broken using the threshold). The column *median* shows the median shape ordered by minimum coverage. If one chooses a random shape, the chance is 50% to be better (or worse) than this one. The last column show the best *one-gapped* shape. (The details of the tie breaking used here can be read in the paper.)

q	best	median	1-gapped
6	`##.....#..#..#.#`	`#.###.....#.#`	`#####....#`
7	`#.##.....##..#.#`	`##..#..#..##`	`#####...##`
8	`##..#.#.............##..#.#`	`#.#..####...#.#`	`######...##`
9	`###..#..#.#...#.##`	`######..#.#.#`	`#######..##`
10	`###..#..#.#..###.#`	`##.##..#.#.###.#`	`#######`