VarMatch: robust matching of small variant datasets using flexible scoring schemes

Chen Sun and Paul Medvedev

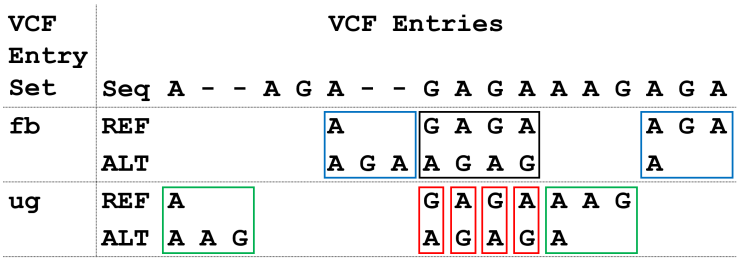
1 Supplemental Material

1.1 Exact branch and bound algorithm

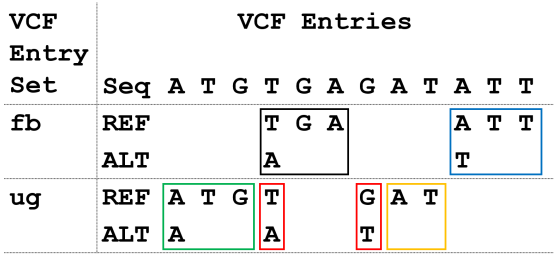
In this section, we present an exact branch and bound algorithm for VarMatch program. It takes as input two variant sequences V, W and reference genome R. For $0 \leq i \leq |V|, 0 \leq j \leq |W|$, a (i,j)-*partial solution* is a pair of selection sequences Φ_V and Φ_W, such that Φ_V has all the positions after the i^{th} one set to 0 and Φ_W has all the positions after the j^{th} one set to 0. Our algorithm maintains a queue Q of partial solutions, initialized with a $(0,0)$-partial solution. The main body of our algorithm is a loop where in each step, we pop a partial solution s from Q, pick from V or W the closest variant v that is not in s, and create and push onto Q three new partial solutions corresponding to the three selection options for v. The loop stops when Q contains only $(|V|,|W|)$-partial solutions, and we output the highest scoring one.

We employ two main strategies to prune the search space. First, consider an (i,j)-partial solution (Φ_V, Φ_W) and the shortest genome sequence R' that is affected by the first i variants of V and the first j variants of W. We can apply the partial solution to R' to get two pairs of donor sequences v_0, v_1 and w_0, w_1. We call these *partial donors* corresponding to the partial solution. We can safely discard this partial solution if there exists an $i \in \{0,1\}$ such that v_i is not the prefix of w_i, and w_i is not the prefix of v_i. In such cases, no matter how the partial solution is extended, the donor strings will never be identical. For the second pruning strategy, fix i and j and consider a set of (i,j)-partial solutions with the same partial donor sequences lengths. We can discard all partial solutions in such a set except for one with the highest score.

The algorithm's running time and memory usage is $\Omega(3^{|V|+|W|})$, since this is the number of possible solutions. However, the pruning strategies make the algorithm fast in practice, since it is applied only on small subproblems generated by the LinearClustering algorithm discussed in main text Section 3.2. Our algorithm is based on and similar to the algorithm of RTG Tools discussed in main text. The novelty here is to properly formalize it into a branch and bound framework, to branch the search tree on variants instead of on nucleotides, and to optimize the pruning operations.



Supplementary Figure 1: An example where different numbers of matches are made in the total scoring scheme then in the baseline scoring scheme. VCF entries are represented by boxes but are grouped together into entry sets from fb as baseline and ug as query. We put dashes into the reference (Seq) to space it out for the purposes of illustrating insertions. Under the total scoring scheme, the fb entry represented by a black box (with GAGA as the reference allele) matches the four ug entries in red boxes (all SNVs). The total number of variants in this match is five but only one is from the baseline. At the same time, all the three fb variants match the two green ug variants (indels). The total number of variants is five but there are three from the baseline.



Supplementary Figure 2: An example where different matches are made in unit vs. edit distance cost model (UGT vs. EGT). The fb entry represented by a black box (on the left) matches the ug entries in red and orange boxes (the right three), giving a match with four total variants and an edit distance of seven. At the same time, all the fb entries match the ug entries represented by green and orange boxes, giving a match with four total variants and an edit distance of eight.

Main text ref	Dataset	Unit Cost Model		Edit Distance Cost Model	
		Mode	# Matches	Mode	# Matches
Table 1	fb	UGT	2,843,396	EGT	+0
	hc		2,912,641		+0
Table 2	fb	UGT	4,197,138	EGT	+8
	ug		4,322,083		-15
Table 4	bwa-fb	UGB	402,552	EGB	-4
	pt		532,856		-20
Table 5 (genome-wide)	fb	UGB	2,896,841	EGB	+0
	pt		2,891,848		-1
Table 5 (dense regions)	fb	UGB	24,188	EGB	+0
	pt		24,522		-1

Supplementary Table 1: Difference of VarMatch in the unit cost and edit distance cost model. Datasets are used from the corresponding tables in the main text. The number of matches in the edit distance cost model is shown as an offset to the number of matches in the unit cost model.

	# Matched Benchmark Entries (Recall)	
	fb	pt
genome-wide	2,896,841 (99.35%)	2,891,759 (99.17%)
dense regions	24,188 (84.69%)	24,486 (85.73%)

Supplementary Table 2: The number and recall of benchmark variants matched by RTG Tools in fb and pt, from the whole genome (first row) and just the dense regions (second row).