Methodology for *in silico* mining of microsatellite polymorphic *loci*

Carlos M. Martínez Ortiz and Alejandro Rivero Bandínez
Department of Biochemistry, University of Medical Sciences, ICPB "Victoria de Girón", Havana, Cuba

Abstract

Background: Polymorphisms with variable number of tandem repeats (VNTR), particularly in microsatellites (SSR), are genetic markers widely used in multiple areas of genomics as evolutionary, epidemiological and population genetics studies. The exponential growth of genomic sequences in data banks and the development of computational tools for bioinformatics allow the mining of these markers without the need to use experimental methods, extending the analysis to non-model organisms of medical or economic importance. However, due to the low complexity of these sequences and the high number of candidates presented when inspecting one or several genomes in a scaled manner, serious difficulties arise in processing the enormous volume of data that is generated and the detection of polymorphisms by visual inspection in candidate markers.

Methods: A methodology and its algorithmic specificities are described, implemented in a software pipeline, which allow the fast and reliable identification of polymorphic SSRs *loci*. The global processing is done by the concatenation of the programs MIDAS, BLAST and the PSSR-Extractor script. The inputs are directory paths where multiple sequence files are found in FASTA or GBFF format and the outputs are the SSRs, access codes to the databases, positions in the genome, number of repetitions and the degree of polymorphism expressed as range of variation, allelic frequency, allele number and polymorphic information content (PIC). An optional script, SSRMerge, allows the identification of unique (non-redundant) *loci* in the set of processed genome sequences with taxonomically closed relationship.

Application: 23 complete genomes (RefSeq from NCBI) belonging to various isolates of *Mycobacterium tuberculosis* were processed. 4433 SSRs were detected and from them 414 non-redundant *loci* were extracted within the species. The polymorphisms for these SSRs were mined in the BLAST server outputs and different measures are reported that reflect *loci* variations.

Keywords: SSR
VNTR
Molecular marker
Data mining
Algorithm

E-mail: cmmo@infomed.sld.cu

Introduction

Microsatellites, or simple sequence repeats (SSRs), are small DNA motifs (from 1 to 6 nucleotides) repeated in tandem, present in all the genomes of prokaryotic and eukaryotic organisms [1]. These sequences forms tracts that can range from a few copies to hundreds of them. The molecular mechanism that explains these sequences is the so-called replication slippage, being the mechanism itself the cause of variability observed in the number of copies [2]. Microsatellites present high levels of polymorphism, which is translated as variable number of tandem repeats (VNTR) [3]. They present mutation rates between 10^{-2} and 10^{-5} per *locus* per generation, contrasting with other markers, for example, simple nucleotide polymorphisms (SNP) that has mutation rate around 10^{-9}. The variation in the mutation rates of VNTRs also produces a wide range of allelic diversity, making these markers very valuable to determine the degree of biological relationship between populations at the same species.

When, for a given study, microsatellites are chosen as markers, researchers have two options for their detection and characterization: generating sequence data or mining in repositories of sequences, whether public or private. The first option requires the preparation of genomic libraries, sequencing platforms and software for subsequent detection. The second option makes the first two steps of the first option unnecessary, eliminating its high costs, leaving only the stage assisted by software [4].

In silico mining of polymorphic SSRs comprises two stages: 1st detect SSRs, with a wide range of applications developed, which, despite having the same purpose, use diverse statistical and computational criteria that influences the results obtained [5, 6]; 2nd determine if these SSRs has polymorphisms in repeat copy number, for which it is necessary to make sequence comparisons against repositories of related sequences or the same species. The ideal applications for this purpose are those of BLAST type (Basic Local Alignment Search Tool, http://blast.ncbi.nlm.nih.gov/) [7]. This second stage, due the goal of BLAST of searching through local alignment homologous subsequences and the composition of the SSRs, which are regions of very low complexity, entails post-processing the outputs which presents serious difficulties when we want to do this mining in a scaled manner. This post-processing is usually done by manually editing the alignments and visually inspecting them when there are few SSRs candidates to explore, but when it comes to hundreds or thousands of candidates, the process necessarily requires a formalization and automation. In reviewed articles on the subject, the methodology for these purposes is not clarified, presenting itself in some cases in a low explicit manner and in others, omitting completely. Few references appear on this second stage and it is due to the usual procedure of pass directly to functional genotyping in experimental manner, for which an extensive and expensive detection system is required, which includes PCR, electrophoresis and sometimes sequencing.

In the case of microsatellites, the determination of polymorphism by variation in copy number, either experimentally or *in silico,* is supported in the specificity of the flanks at each *locus*. These 5'- and 3'-flanking sequences, which are normally reported by software that detect SSR with sizes around 20bp, are assumed to be conserved and unique in the genomes of each species, allowing the unambiguous location of the *locus* and becoming candidates for primer sequences in PCR techniques.

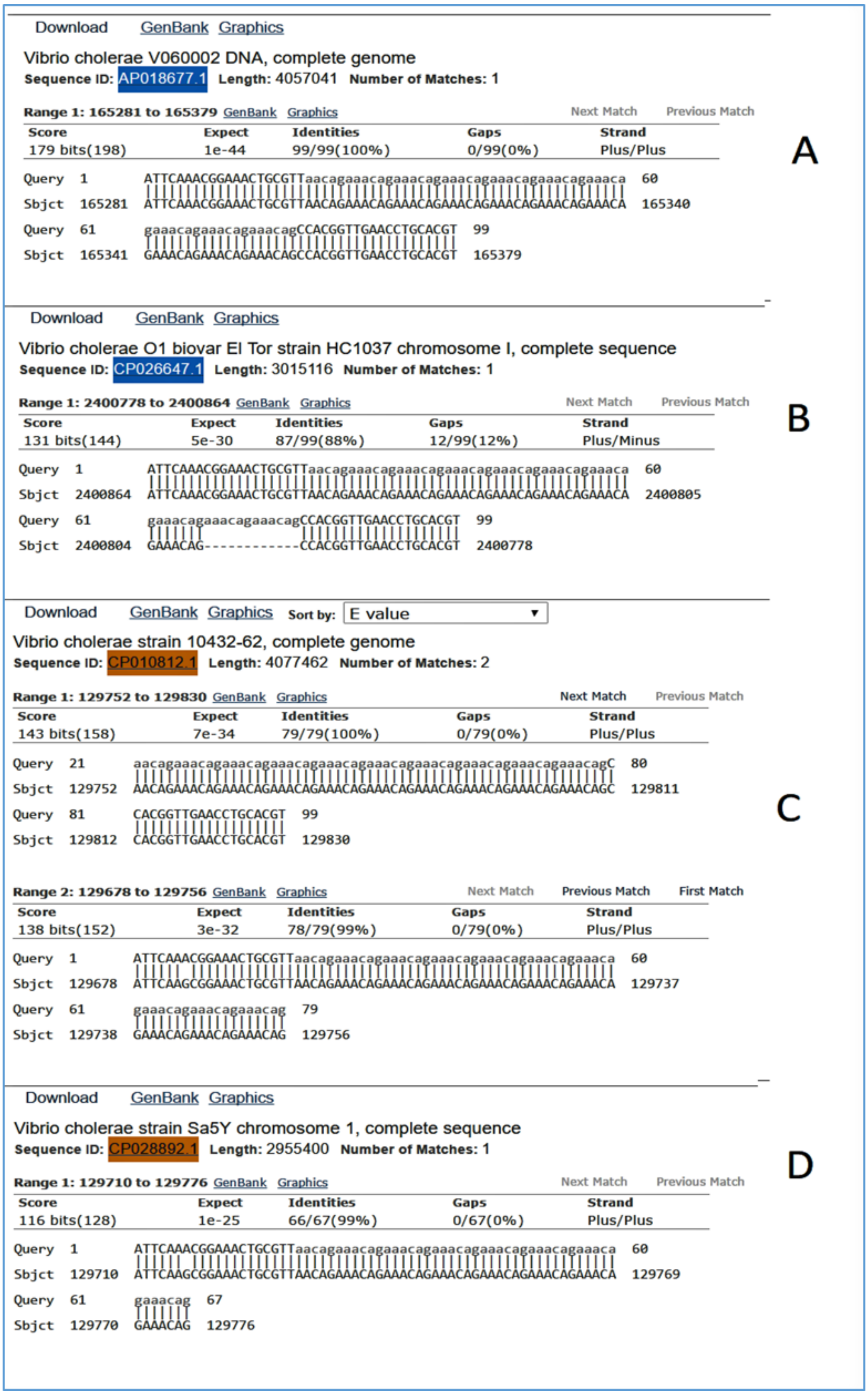
When we do an extensive search with BLAST (blastn, for nucleotides) and the query sequences are SSRs with their respective 5'and 3' flanks, we have several complications due to the low complexity or redundancy of these sequences and also the possibility that flanks are not properly preserved. We must remember that BLAST is a system designed to detect homologous sequences that precisely includes filters to eliminate the high scores that produce these repeated regions. This system does not have a specific design to detect non-redundant homologous sequences separated by a redundant region and therefore it can't capture the *locus* as a whole in order to compare the variations in copy number.

Figures 1A - 1D show BLAST outputs for an SSR with these characteristics and the different types of alignments that produce the findings (hits) for different entries in the chosen database. It is an SSR extracted from a genome of *Vibrio cholerae* faced with a database of nucleotide sequences of this bacterial species. The SSR query is shown with flanks of 20 bp in capital letters and the repeat region, which is the region that must vary in length in case of being a polymorphic *locus*, in lowercase letters, with repetition pattern **aacaga**. Figure 1A shows how the blastn finds a sequence that is identical to the query, producing 100% identity and an e-value of 1e-44. In this case it is not ruled out that the sequence found (ID: AP018677.1) is the same one from which the SSR was extracted, although it does not necessarily have to be that way. Figure 1B shows an ideal case where an entry is found in the database (ID: CP026647.1) with a *locus* that presents variation in the number of repetitions, in this case it is a suppression of $(aacaga)_2$. Figure 1C shows two hits for the same sequence (ID: CP010812.1), which represent two entries in the BLAST output file (hit table) where the *locus* could not be completely covered by the system and two alignments are presented that overlap in a certain region. This represents a case where it is complicated to detect, visually or by another type of method, the variation in the number of copies. Finally, Figure 1D shows an entry for a sequence (ID: CP028892.1) where the alignment was truncated, not reaching the other flank and not reporting any other hit for that sequence with the right flank.

These are some examples where one can observe possible complications to interpret and detect polymorphism in a computational manner. When this is scaled to hundreds of markers it is totally impossible to do by simple visual inspection of alignments, even editing these to solve truncated entries.

The methodology that we present describes the stages and the algorithmic bases for the computational detection of polymorphisms in SSRs. The general procedure is done by the concatenation of software ranging from the detection of the SSRs, processing these by BLAST system and interpretation of BLAST outputs for detection of polymorphic markers.

In the following section (**Methods**) the sequence of steps followed by this methodology and the software used with explanation of its specificities are described in detail. In the **Results** section, the output corresponding to the detection of polymorphisms of SSRs in *Mycobacterium tuberculosis* genomes is exposed and analyzed. It also describes input parameters, input and output formats and reported values.



```
Download     GenBank  Graphics

Vibrio cholerae V060002 DNA, complete genome
Sequence ID: AP018677.1  Length: 4057041  Number of Matches: 1

Range 1: 165281 to 165379 GenBank  Graphics                          Next Match    Previous Match
 Score               Expect    Identities          Gaps            Strand
 179 bits(198)       1e-44     99/99(100%)         0/99(0%)        Plus/Plus

Query  1       ATTCAAACGGAAACTGCGTTaacagaaacagaaacagaaacagaaacagaaacagaaaca  60
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  165281  ATTCAAACGGAAACTGCGTTAACAGAAACAGAAACAGAAACAGAAACAGAAACAGAAACA  165340

Query  61      gaaacagaaacagaaacagCCACGGTTGAACCTGCACGT  99
               |||||||||||||||||||||||||||||||||||||||
Sbjct  165341  GAAACAGAAACAGAAACAGCCACGGTTGAACCTGCACGT  165379
```

```
Download     GenBank  Graphics

Vibrio cholerae O1 biovar El Tor strain HC1037 chromosome I, complete sequence
Sequence ID: CP026647.1  Length: 3015116  Number of Matches: 1

Range 1: 2400778 to 2400864 GenBank  Graphics                        Next Match    Previous Match
 Score               Expect    Identities          Gaps            Strand
 131 bits(144)       5e-30     87/99(88%)          12/99(12%)      Plus/Minus

Query  1        ATTCAAACGGAAACTGCGTTaacagaaacagaaacagaaacagaaacagaaacagaaaca  60
                ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  2400864  ATTCAAACGGAAACTGCGTTAACAGAAACAGAAACAGAAACAGAAACAGAAACAGAAACA  2400805

Query  61       gaaacagaaacagaaacagCCACGGTTGAACCTGCACGT  99
                |||||||            ||||||||||||||||||||
Sbjct  2400804  GAAACAG------------CCACGGTTGAACCTGCACGT  2400778
```

```
Download     GenBank  Graphics   Sort by: E value ▼

Vibrio cholerae strain 10432-62, complete genome
Sequence ID: CP010812.1  Length: 4077462  Number of Matches: 2

Range 1: 129752 to 129830 GenBank  Graphics                          Next Match    Previous Match
 Score               Expect    Identities          Gaps            Strand
 143 bits(158)       7e-34     79/79(100%)         0/79(0%)        Plus/Plus

Query  21      aacagaaacagaaacagaaacagaaacagaaacagaaacagaaacagaaacagaaacagC  80
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  129752  AACAGAAACAGAAACAGAAACAGAAACAGAAACAGAAACAGAAACAGAAACAGAAACAGC  129811

Query  81      CACGGTTGAACCTGCACGT  99
               |||||||||||||||||||
Sbjct  129812  CACGGTTGAACCTGCACGT  129830


Range 2: 129678 to 129756 GenBank  Graphics              Next Match    Previous Match    First Match
 Score               Expect    Identities          Gaps            Strand
 138 bits(152)      3e-32     78/79(99%)          0/79(0%)        Plus/Plus

Query  1       ATTCAAACGGAAACTGCGTTaacagaaacagaaacagaaacagaaacagaaacagaaaca  60
               |||||| |||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  129678  ATTCAAGCGGAAACTGCGTTAACAGAAACAGAAACAGAAACAGAAACAGAAACAGAAACA  129737

Query  61      gaaacagaaacagaaacag  79
               |||||||||||||||||||
Sbjct  129738  GAAACAGAAACAGAAACAG  129756
```

```
Download     GenBank  Graphics

Vibrio cholerae strain Sa5Y chromosome 1, complete sequence
Sequence ID: CP028892.1  Length: 2955400  Number of Matches: 1

Range 1: 129710 to 129776 GenBank  Graphics                          Next Match    Previous Match
 Score               Expect    Identities          Gaps            Strand
 116 bits(128)       1e-25     66/67(99%)          0/67(0%)        Plus/Plus

Query  1       ATTCAAACGGAAACTGCGTTaacagaaacagaaacagaaacagaaacagaaacagaaaca  60
               |||||| |||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  129710  ATTCAAGCGGAAACTGCGTTAACAGAAACAGAAACAGAAACAGAAACAGAAACAGAAACA  129769

Query  61      gaaacag  67
               |||||||
Sbjct  129770  GAAACAG  129776
```

Figure 1. Examples of BLAST outputs for an SSR showing some complications that affect the identification of polymorphic *locus*. See explanations in the text for cases A, B, C, D.

Methods

Figure 2 shows the general sequence of stages for *in silico* mining of polymorphic SSRs. First a run of MIDAS [8] is done, entering as parameters a genomic sequence file that can be in FASTA or GBFF format (with simple or multiple entries), the minimum unit of the repeat to be detected and the alignment parameters for match, mismatch and indel. MIDAS detects all SSRs, exact or approximate, and generates a MultiFASTA file (extension .mfaa) with 20 bp flanking sequences added in capital letters and the repeat in lowercase letters. This format that marks the repeat region with lowercase letters is used by BLAST as masking procedure.

Then we run SSRMerge script, which allows extracting the set of non-redundant SSRs from multiple MIDAS output files applied to multiple genomes. Its first parameter is a path to a directory in the PC and it will process all the files with extension .mfaa. This is an optional stage and is applicable only when we are analyzing multiple genomes of related species. The greater the taxonomic link between these genomes, the greater the probability of finding similar conserved *loci* that are repeated in different genomes. The algorithmic principle of this script is an all against all flank comparison of SSRs present in all files processed. This comparison is made by Nedleman-Wunsch global sequence alignment. When comparing two SSRs and they present more than 90% identity, one of them is chosen and the other is rejected. The result is a MultiFASTA file containing a set of non-redundant SSRs, according to the defined parameters, and we assume that these SSRs belong to different *loci* in the set of genomes analyzed.

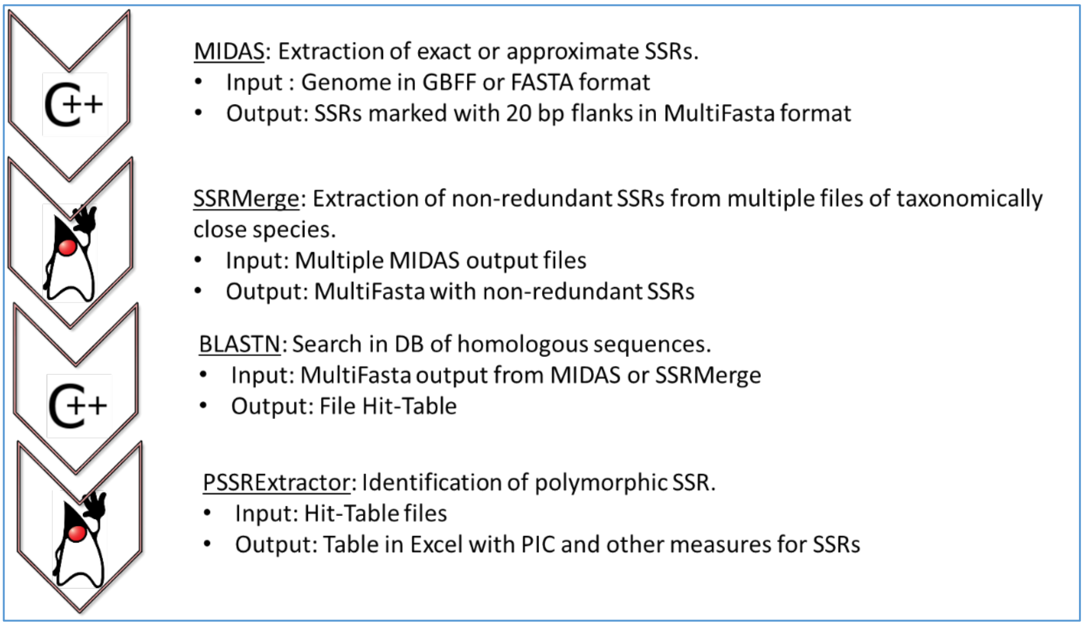

C++
MIDAS: Extraction of exact or approximate SSRs.
• Input : Genome in GBFF or FASTA format
• Output: SSRs marked with 20 bp flanks in MultiFasta format
SSRMerge: Extraction of non-redundant SSRs from multiple files of taxonomically close species.
• Input: Multiple MIDAS output files
• Output: MultiFasta with non-redundant SSRs
C++
BLASTN: Search in DB of homologous sequences.
• Input: MultiFasta output from MIDAS or SSRMerge
• Output: File Hit-Table
PSSRExtractor: Identification of polymorphic SSR.
• Input: Hit-Table files
• Output: Table in Excel with PIC and other measures for SSRs

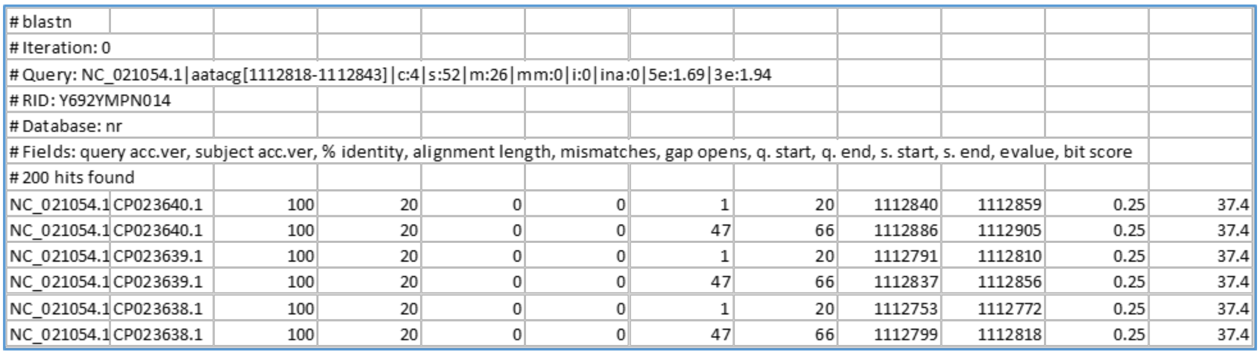
Figure 2. General sequence of steps for *in silico* mining of polymorphic SSRs.

The third step is a BLAST submitting (NCBI server, http://blast.ncbi.nlm.nih.gov/). The entry to the server is the MultiFASTA file of the previous stage. BLAST has many configuration parameters but the parameters that we must modify in that particular case are the following: In the search set (Choose Search Set) specify the organism for which we want to search (e.g. *Mycobacterium tuberculosis* (taxid: 1773)); in the program type select blastn, designed to find more remote homologies; in general parameters of the algorithm we are only interested in modifying the expected threshold (default threshold) which by default is 10. The expected threshold should be increased to > 30 allowing to find hits that although have large e-values they can be *loci* of interest. The next stage executed by PSSRextractor is responsible for debugging the wrong hits. Another parameter that needs to be changed is the masking procedure. We must mark the option of regions of low complexity and masking with lowercase letters. This guarantees that the blastn tries to recognize all *loci* including two flanks despite this is not always achieved, especially when the repeated region is large. The PSSRextractor script implements business rules to solve these cases. The output of this step is a hit table generated by the server.

Finally, the BLAST output is processed by PSSRextractor. It can process one or several outputs because its first input parameter is a path to a directory. It will process all the BLAST output files. This script first parses the hit table extracting all its information. The other two script parameters are identity and percentage of coverage that define criteria to consider the flanks in each entry of hit table. Subsequently it proceeds to evaluate the polymorphisms by variable number of repeat in the SSRs and it does so as follows:

For an SSR (query in hit-table) there will be one or many entries (subject in hit-table). Each SSR has a repeated unit size (RUS). Each entry in hit-table has the following values: subject access identifier (SAV), alignment length (AL), identity% (% I), initial position of the query (q.start), final position of the query (q.end), initial position of the subject (s.start) and final position of the subject (s.end). Then taking into account these data, the number of repetitions (RN) of each repeat unit is calculated by validating the following conditions:

1. If AL> 40 Then: RN = (AL - 40) / RUS. When the alignment is greater than or equal to 40 it means that it covered both sides of the query, since the repeated region is marked and not taken into account. Then the difference would give us the number of nucleotides that are in the repeat region that divided by RUS gives us the RN (Repeat Number). These cases occur with very low frequency.
2. If q.end <= 20 then the sequence found coincides with the left flank, otherwise, the sequence found coincides with the right flank. So for the same query, all the entries with the same SAV, will be classified to the left or to the right of the *locus* according to this condition. These cases are those that occur very frequently. See Figure 3.
3. If s.start left < s.end left AND s.start right < s.end right AND s.end left < s.start right, then RN = (|s.start right - s.end left| - 1) / RUS. This condition validates that the left flank is to the left of the subject sequence and that the right flank is to the right, being positive the direction (i.e. from 5' to 3'). Similar condition is validated when the address is negative, you just have to reverse the strict inequalities. This occurs because BLAST also analyzes the complementary sequence in the BD. See Figures 3 and 4.



# blastn											
# Iteration: 0											
# Query: NC_021054.1|aatacg[1112818-1112843]|c:4|s:52|m:26|mm:0|i:0|ina:0|5e:1.69|3e:1.94											
# RID: Y692YMPN014											
# Database: nr											
# Fields: query acc.ver, subject acc.ver, % identity, alignment length, mismatches, gap opens, q. start, q. end, s. start, s. end, evalue, bit score											
# 200 hits found											
NC_021054.1	CP023640.1	100	20	0	0	1	20	1112840	1112859	0.25	37.4
NC_021054.1	CP023640.1	100	20	0	0	47	66	1112886	1112905	0.25	37.4
NC_021054.1	CP023639.1	100	20	0	0	1	20	1112791	1112810	0.25	37.4
NC_021054.1	CP023639.1	100	20	0	0	47	66	1112837	1112856	0.25	37.4
NC_021054.1	CP023638.1	100	20	0	0	1	20	1112753	1112772	0.25	37.4
NC_021054.1	CP023638.1	100	20	0	0	47	66	1112799	1112818	0.25	37.4

Figure 3. Example of BLAST hit-table output. It is observed how for a query it finds two entries for the same subject with SAV CP023640.1. The first entry has q.start = 1 and q.end = 20 (left side of the query) and the second has q.start = 47 and q.end = 66 (right side of the query).

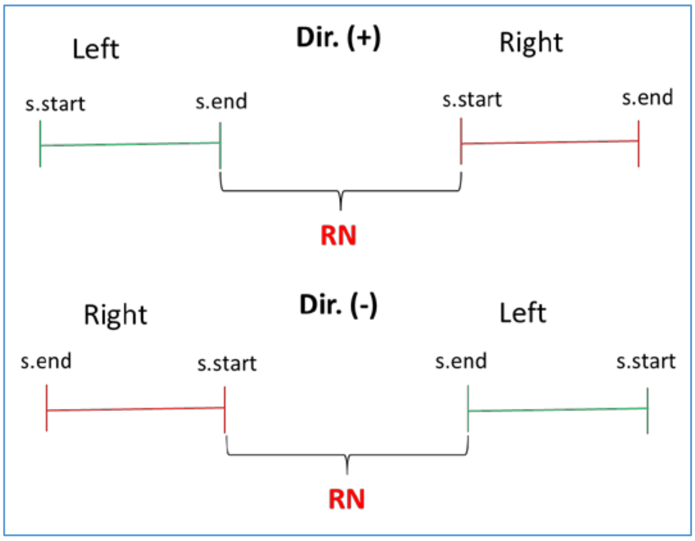
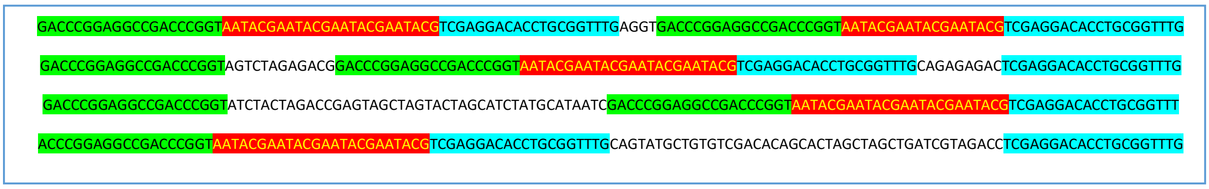


Figure 4. Validation of the address (positive or negative in query-subject comparison) and RN calculation.

4. RN chosen = *min* {RNs found}. This validation is done because doubtful cases can occur due to duplicate regions in the same genome. For example, for the query with the repeated unit AATACG (red zone) between the left flank (green zone) and the right flank (blue zone) the following cases can be found:



```
GACCCGGAGGCCGACCCGGTAATACGAATACGAATACGAATACGTCGAGGACACCTGCGGTTTGAGGTGACCCGGAGGCCGACCCGGTAATACGAATACGAATACGTCGAGGACACCTGCGGTTTG
GACCCGGAGGCCGACCCGGTAGTCTAGAGACGGACCCGGAGGCCGACCCGGTAATACGAATACGAATACGAATACGTCGAGGACACCTGCGGTTTGCAGAGAGACTCGAGGACACCTGCGGTTTG
GACCCGGAGGCCGACCCGGTATCTACTAGACCGAGTAGCTAGTACTAGCATCTATGCATAATCGACCCGGAGGCCGACCCGGTAATACGAATACGAATACGAATACGTCGAGGACACCTGCGGTTT
ACCCGGAGGCCGACCCGGTAATACGAATACGAATACGAATACGTCGAGGACACCTGCGGTTTGCAGTATGCTGTGTCGACACAGCACTAGCTAGCTGATCGTAGACCTCGAGGACACCTGCGGTTTG
```

5. Only entries where the alignments have a % I (second parameter of the script) and a % of coverage (third parameter of the script, (AL / 20) x 100) greater than 90 are taken into account. These two parameters guarantee that the flanks found are well preserved.

After having the set of RN for all subjects, PSSRextractor generates two result files, one detailed and one generic, with names equal to the hit-table but with suffixes _specific.xls and _generic.xls respectively. The detailed one provides information on each subject processed and the generic provides the information related to the polymorphism for each query, i.e. for each SSR.

Among the information related to the polymorphism that generic report provides, are the following values:

I. min_RN, max_RN and range: These are three columns in the report that respectively mean the minimum RN, the maximum RN and the range (max_RN - min_RN).
II. frequency: Allelic frequency of the original SSR from which the search was made.
III. alleles: Number of alleles found for one *locus* (SSRs with different RNs).
IV. PIC: Polymorphic Information Content ($1 - \sum_i p_i^2$). This value is also known in other contexts as expected average heterozygosity or Nei genetic diversity, and gives a measure of the probability that, for a single *locus,* a pair of alleles chosen at random in the population are different.

The rest of values shown in generic report come from MIDAS (access_number, pattern, pattern_length (RUS), RN, inaccuracy, entropy_5, entropy_3), which provides valuable information when interpreting polymorphic SSRs. For example, the degree of inaccuracy found in SSRs or the compositional entropy of the flanks allow us to know, respectively, the degree of inaccuracy of the repeated tract and how informative the flanks may be in the characterization of the *locus*.

The values of the last column of the generic report (exceptions) show labels that correspond to exceptions in polymorphism validation. There are entries in the generic report where more than one of these labels can appear because the exceptions can occur simultaneous. When all the subject sequences have exceptions, the labels are place. Labels are the following:

i. **D** (degenerated): The subjects have a % I and/or a % coverage <90%.
ii. **NF** (not found): No subject was found in the database with similarity.
iii. **O** (outlier): The number of repeat units between the flanks of the subjects is doubtful because it is very large, being unlikely that there is a microsatellite between them. The cut-off values established for this exception were mono: 157 bp, di: 364 bp, tri: 109 bp, tetra: 45 bp, penta: 150 bp and hexa: 193 bp. These values were defined after processing all the SSR from more than 200 bacterial genomes, registering their sizes, and establishing the cut-off in 3 times the interquartile range.
iv. **U** (unpair): For the same subject sequence, one edge appears and not the other.

Results and Discussion

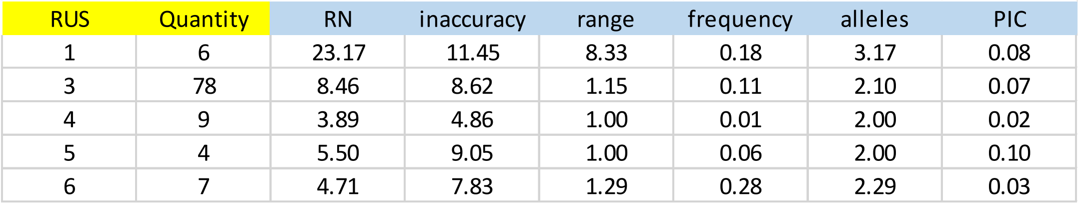
Mycobacterium tuberculosis, also known as Koch's bacillus, is a bacterial pathogen, causative agent of tuberculosis, a contagious infection that mainly affects the lungs but can spread to other organs. The prevalence of this disease is among the highest in the world, with approximately 12 million people infected. The bacillus is also object of concern in medical and scientific communities for presenting strains with resistance to multiple antibiotics.

The errors due to sliding in replication that cause microsatellites are normally repaired by three enzymes *mutL, mutS and mutH,* however some genomes such as mycobacteria suffer from this enzymatic system [9]. Due to this, these bacterial species constitute a valuable example to investigate the rates of microsatellite mutations and the existence of regulatory mechanisms that govern them [10].

M. tuberculosis is a pathogen with a genetic diversity emerged from more diverse strains, gaining in virulence mechanisms [11]. An example of medical importance in the expansion of microsatellites is that which occurs in proteins of mycobacteria with pentapeptide-2 (PP2) [12].

In the present study, 23 complete genomes were analyzed (RefSeq sequences), downloaded from the NCBI ftp site (ftp://ftp.ncbi.nlm.nih.gov/refseq/release/bacteria/), belonging to various *M. tuberculosis* isolates. MIDAS detected 4433 SSRs, and from them, SSRMerge script extracted 414 non-redundant *loci* from the species. The polymorphisms were mined from the BLAST server outputs with PSSRextractor script. The outputs of these two scripts can be found in the supplementary files (species_all.mfaa and Y692YMPN014-Alignment_generic_result.xls).

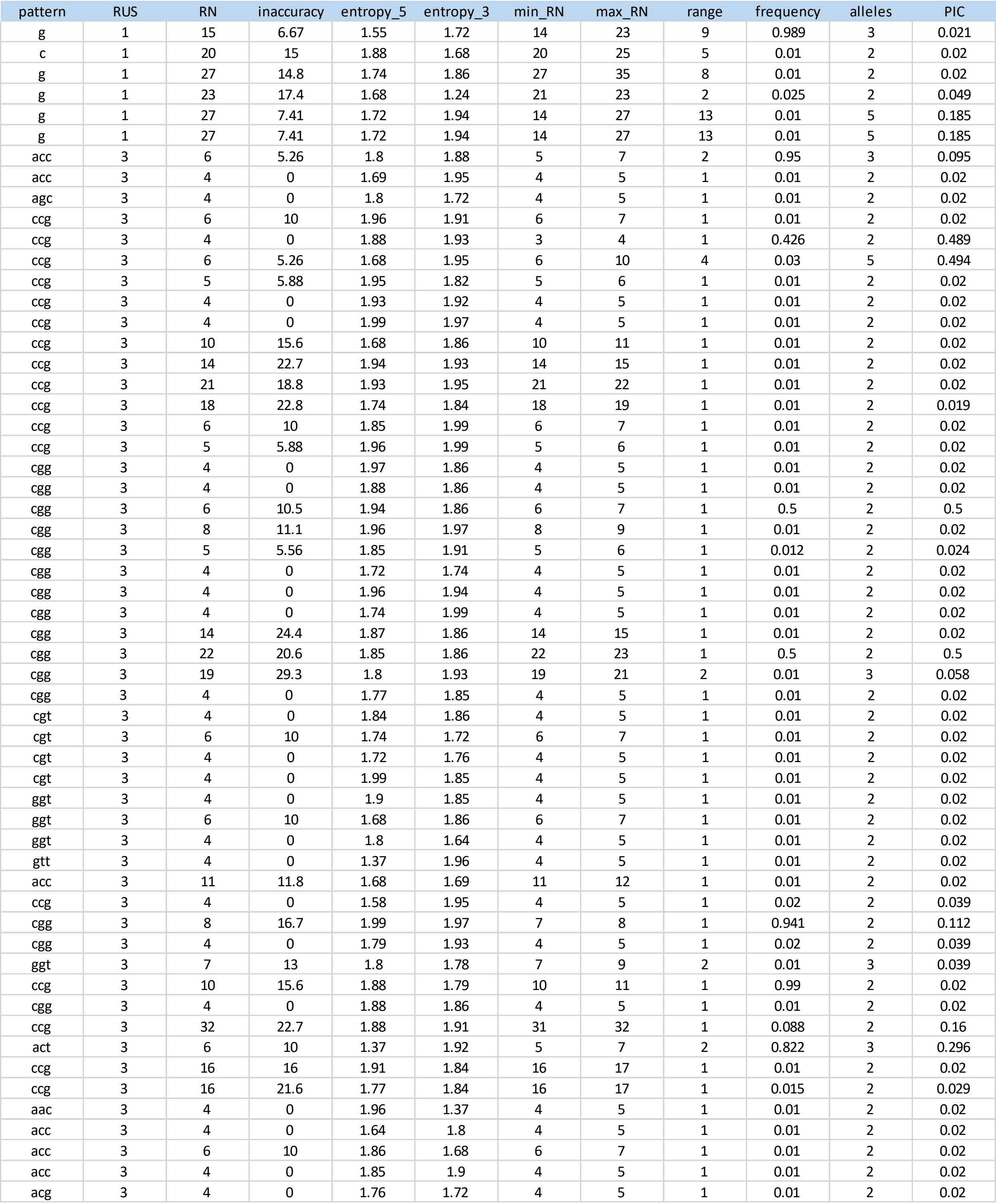
From 414 SSRs, 288 did not show any type of exception, and of these, 104 showed PIC> 0 (36.1%). Figure-5 shows average values obtained for the different types of SSRs classified by the size of the repeat unit (RUS).



RUS	Quantity	RN	inaccuracy	range	frequency	alleles	PIC
1	6	23.17	11.45	8.33	0.18	3.17	0.08
3	78	8.46	8.62	1.15	0.11	2.10	0.07
4	9	3.89	4.86	1.00	0.01	2.00	0.02
5	4	5.50	9.05	1.00	0.06	2.00	0.10
6	7	4.71	7.83	1.29	0.28	2.29	0.03

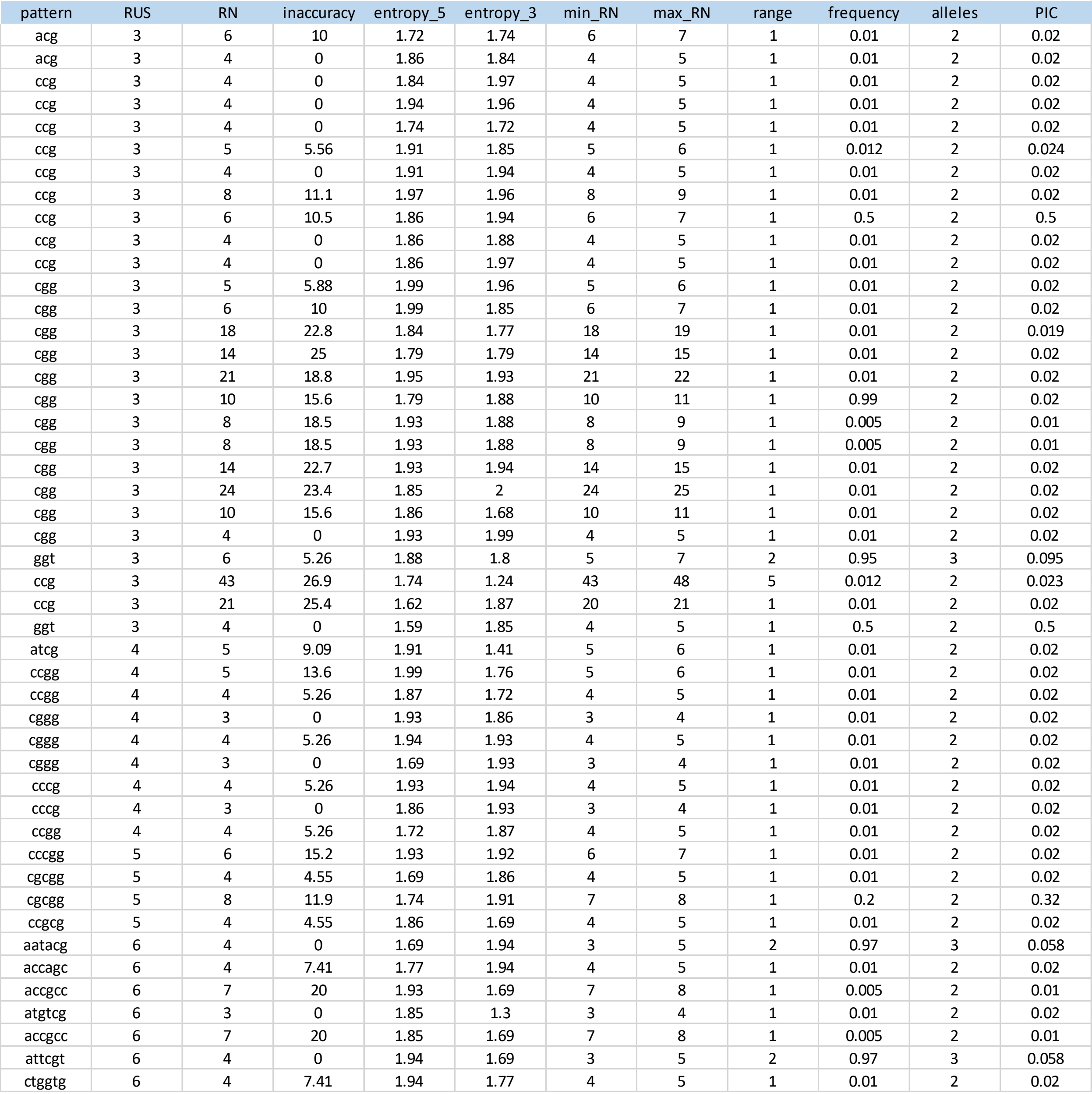
Figure 5. Average values for some measurements taken from the generic report for *M. tuberculosis.*

The number of SSRs with 3bp repeat unit is significant. This is because most of the bacterial genomes are coding regions, which have a bias in the use of codons. A single SSR was detected for dinucleotide and was excluded from the result due to exceptions in the polymorphism validation. On average, SSRs of 1bp were more inaccurate and showed greater polymorphism with more alleles and elevated PIC (3.17 and 0.08 respectively). The repetition tracts (RN) were also significantly greater than the rest.



pattern	RUS	RN	inaccuracy	entropy_5	entropy_3	min_RN	max_RN	range	frequency	alleles	PIC
g	1	15	6.67	1.55	1.72	14	23	9	0.989	3	0.021
c	1	20	15	1.88	1.68	20	25	5	0.01	2	0.02
g	1	27	14.8	1.74	1.86	27	35	8	0.01	2	0.02
g	1	23	17.4	1.68	1.24	21	23	2	0.025	2	0.049
g	1	27	7.41	1.72	1.94	14	27	13	0.01	5	0.185
g	1	27	7.41	1.72	1.94	14	27	13	0.01	5	0.185
acc	3	6	5.26	1.8	1.88	5	7	2	0.95	3	0.095
acc	3	4	0	1.69	1.95	4	5	1	0.01	2	0.02
agc	3	4	0	1.8	1.72	4	5	1	0.01	2	0.02
ccg	3	6	10	1.96	1.91	6	7	1	0.01	2	0.02
ccg	3	4	0	1.88	1.93	3	4	1	0.426	2	0.489
ccg	3	6	5.26	1.68	1.95	6	10	4	0.03	5	0.494
ccg	3	5	5.88	1.95	1.82	5	6	1	0.01	2	0.02
ccg	3	4	0	1.93	1.92	4	5	1	0.01	2	0.02
ccg	3	4	0	1.99	1.97	4	5	1	0.01	2	0.02
ccg	3	10	15.6	1.68	1.86	10	11	1	0.01	2	0.02
ccg	3	14	22.7	1.94	1.93	14	15	1	0.01	2	0.02
ccg	3	21	18.8	1.93	1.95	21	22	1	0.01	2	0.02
ccg	3	18	22.8	1.74	1.84	18	19	1	0.01	2	0.019
ccg	3	6	10	1.85	1.99	6	7	1	0.01	2	0.02
ccg	3	5	5.88	1.96	1.99	5	6	1	0.01	2	0.02
cgg	3	4	0	1.97	1.86	4	5	1	0.01	2	0.02
cgg	3	4	0	1.88	1.86	4	5	1	0.01	2	0.02
cgg	3	6	10.5	1.94	1.86	6	7	1	0.5	2	0.5
cgg	3	8	11.1	1.96	1.97	8	9	1	0.01	2	0.02
cgg	3	5	5.56	1.85	1.91	5	6	1	0.012	2	0.024
cgg	3	4	0	1.72	1.74	4	5	1	0.01	2	0.02
cgg	3	4	0	1.96	1.94	4	5	1	0.01	2	0.02
cgg	3	4	0	1.74	1.99	4	5	1	0.01	2	0.02
cgg	3	14	24.4	1.87	1.86	14	15	1	0.01	2	0.02
cgg	3	22	20.6	1.85	1.86	22	23	1	0.5	2	0.5
cgg	3	19	29.3	1.8	1.93	19	21	2	0.01	3	0.058
cgg	3	4	0	1.77	1.85	4	5	1	0.01	2	0.02
cgt	3	4	0	1.84	1.86	4	5	1	0.01	2	0.02
cgt	3	6	10	1.74	1.72	6	7	1	0.01	2	0.02
cgt	3	4	0	1.72	1.76	4	5	1	0.01	2	0.02
cgt	3	4	0	1.99	1.85	4	5	1	0.01	2	0.02
ggt	3	4	0	1.9	1.85	4	5	1	0.01	2	0.02
ggt	3	6	10	1.68	1.86	6	7	1	0.01	2	0.02
ggt	3	4	0	1.8	1.64	4	5	1	0.01	2	0.02
gtt	3	4	0	1.37	1.96	4	5	1	0.01	2	0.02
acc	3	11	11.8	1.68	1.69	11	12	1	0.01	2	0.02
ccg	3	4	0	1.58	1.95	4	5	1	0.02	2	0.039
cgg	3	8	16.7	1.99	1.97	7	8	1	0.941	2	0.112
cgg	3	4	0	1.79	1.93	4	5	1	0.02	2	0.039
ggt	3	7	13	1.8	1.78	7	9	2	0.01	3	0.039
ccg	3	10	15.6	1.88	1.79	10	11	1	0.99	2	0.02
cgg	3	4	0	1.88	1.86	4	5	1	0.01	2	0.02
ccg	3	32	22.7	1.88	1.91	31	32	1	0.088	2	0.16
act	3	6	10	1.37	1.92	5	7	2	0.822	3	0.296
ccg	3	16	16	1.91	1.84	16	17	1	0.01	2	0.02
ccg	3	16	21.6	1.77	1.84	16	17	1	0.015	2	0.029
aac	3	4	0	1.96	1.37	4	5	1	0.01	2	0.02
acc	3	4	0	1.64	1.8	4	5	1	0.01	2	0.02
acc	3	6	10	1.86	1.68	6	7	1	0.01	2	0.02
acc	3	4	0	1.85	1.9	4	5	1	0.01	2	0.02
acg	3	4	0	1.76	1.72	4	5	1	0.01	2	0.02

Figure 6. SSRs from the *M. tuberculosis* with PIC> 0.



pattern	RUS	RN	inaccuracy	entropy_5	entropy_3	min_RN	max_RN	range	frequency	alleles	PIC
acg	3	6	10	1.72	1.74	6	7	1	0.01	2	0.02
acg	3	4	0	1.86	1.84	4	5	1	0.01	2	0.02
ccg	3	4	0	1.84	1.97	4	5	1	0.01	2	0.02
ccg	3	4	0	1.94	1.96	4	5	1	0.01	2	0.02
ccg	3	4	0	1.74	1.72	4	5	1	0.01	2	0.02
ccg	3	5	5.56	1.91	1.85	5	6	1	0.012	2	0.024
ccg	3	4	0	1.91	1.94	4	5	1	0.01	2	0.02
ccg	3	8	11.1	1.97	1.96	8	9	1	0.01	2	0.02
ccg	3	6	10.5	1.86	1.94	6	7	1	0.5	2	0.5
ccg	3	4	0	1.86	1.88	4	5	1	0.01	2	0.02
ccg	3	4	0	1.86	1.97	4	5	1	0.01	2	0.02
cgg	3	5	5.88	1.99	1.96	5	6	1	0.01	2	0.02
cgg	3	6	10	1.99	1.85	6	7	1	0.01	2	0.02
cgg	3	18	22.8	1.84	1.77	18	19	1	0.01	2	0.019
cgg	3	14	25	1.79	1.79	14	15	1	0.01	2	0.02
cgg	3	21	18.8	1.95	1.93	21	22	1	0.01	2	0.02
cgg	3	10	15.6	1.79	1.88	10	11	1	0.99	2	0.02
cgg	3	8	18.5	1.93	1.88	8	9	1	0.005	2	0.01
cgg	3	8	18.5	1.93	1.88	8	9	1	0.005	2	0.01
cgg	3	14	22.7	1.93	1.94	14	15	1	0.01	2	0.02
cgg	3	24	23.4	1.85	2	24	25	1	0.01	2	0.02
cgg	3	10	15.6	1.86	1.68	10	11	1	0.01	2	0.02
cgg	3	4	0	1.93	1.99	4	5	1	0.01	2	0.02
ggt	3	6	5.26	1.88	1.8	5	7	2	0.95	3	0.095
ccg	3	43	26.9	1.74	1.24	43	48	5	0.012	2	0.023
ccg	3	21	25.4	1.62	1.87	20	21	1	0.01	2	0.02
ggt	3	4	0	1.59	1.85	4	5	1	0.5	2	0.5
atcg	4	5	9.09	1.91	1.41	5	6	1	0.01	2	0.02
ccgg	4	5	13.6	1.99	1.76	5	6	1	0.01	2	0.02
ccgg	4	4	5.26	1.87	1.72	4	5	1	0.01	2	0.02
cggg	4	3	0	1.93	1.86	3	4	1	0.01	2	0.02
cggg	4	4	5.26	1.94	1.93	4	5	1	0.01	2	0.02
cggg	4	3	0	1.69	1.93	3	4	1	0.01	2	0.02
cccg	4	4	5.26	1.93	1.94	4	5	1	0.01	2	0.02
cccg	4	3	0	1.86	1.93	3	4	1	0.01	2	0.02
ccgg	4	4	5.26	1.72	1.87	4	5	1	0.01	2	0.02
cccgg	5	6	15.2	1.93	1.92	6	7	1	0.01	2	0.02
cgcgg	5	4	4.55	1.69	1.86	4	5	1	0.01	2	0.02
cgcgg	5	8	11.9	1.74	1.91	7	8	1	0.2	2	0.32
ccgcg	5	4	4.55	1.86	1.69	4	5	1	0.01	2	0.02
aatacg	6	4	0	1.69	1.94	3	5	2	0.97	3	0.058
accagc	6	4	7.41	1.77	1.94	4	5	1	0.01	2	0.02
accgcc	6	7	20	1.93	1.69	7	8	1	0.005	2	0.01
atgtcg	6	3	0	1.85	1.3	3	4	1	0.01	2	0.02
accgcc	6	7	20	1.85	1.69	7	8	1	0.005	2	0.01
attcgt	6	4	0	1.94	1.69	3	5	2	0.97	3	0.058
ctggtg	6	4	7.41	1.94	1.77	4	5	1	0.01	2	0.02

Fig. 6 (cont.) SSRs from the *M. tuberculosis* with PIC> 0.

Figure 6 shows the complete list of 104 extracted SSRs that showed levels of polymorphism. The entire sequences of the markers, including the flanking sequences, the access numbers and positions in the genome can be obtained from the MIDAS outputs.

The methodology described has distinctive features with respect to other *in silico* procedures reported in the literature:

(I) It has been experimentally demonstrated that not all SSRs *loci* show polymorphism. Among other aspects, this is because the *locus* within the analyzed population is not subject to a particular changing dynamic. In this sense, the methodology allows us to select those *loci* that do show polymorphisms in the selected sequence data banks, reducing the costs when it is done experimentally.

(II) Determination of *locus* polymorphism is fully automated. The common procedures to detect polymorphisms, not the experimental ones, use visual inspection of multiple alignments for markers. In this sense, the procedure is ideal for large-scale analysis.

(III) Polymorphism is defined strictly as variation in copy number (PIC> 0), and not as simple insertions or deletions present in markers that do not correspond to the size of the repeat unit.

(IV) The procedure can be done starting from SSRs detection in one sequence, but optionally, it can also start from multiple sequences without the need to assemble a consensus one. This allows a SSRs detection that do not belong to the same *locus* despite being in highly related genomes. SSRMerge script allows to eliminate the redundancy of common *loci* in all sequences.

The methodology is excellent for purely computational analysis of SSR *loci*, applied to evolutionary studies, genotypic identification or functional studies with the genes involved. For experimental analysis using PCR, the methodology provides all the necessary information (sequence access identifier, positions in the genome, flanking sequences, etc.) that allows the design of primers using applications available on internet.

Availability

All the software are available in supplementary material. MIDAS (binary distribution midas_v1.1.exe). SSRMerge and PSSRExtractor are compressed in zip format. Both are *Java NetBeans Projects* for JDK 1.8 platform or above. The binary jar files, for command line execution, are in \dist folder once decompressed.

Conclusions

A methodology for the fully computational detection of microsatellite polymorphic *loci* is described. As example of use, 23 complete genomes belonging to various isolates of *M. tuberculosis* were processed. 4433 SSRs were detected and from them 414 non-redundant *loci* were extracted within the species. The polymorphisms were mined from BLAST outputs and 104 SSRs showed polymorphisms. The methodology is intuitive and comes with software for application. Its main advantage lies in the levels of scaling it allows and the reduction of costs when experimental analyzes are made, allowing preselection of markers showing polymorphism in chosen genomic sequence data banks.

References

1. Li YC, Korol AB, Fahima T, Beiles A, Nevo E. Microsatellites: Genomic distribution, putative functions and mutational mechanisms: A review. Molecular Ecology 2002; 11: 2453–2465.
2. Ellegren, H. Microsatellites: Simple sequences with complex evolution. Nature Reviews. Genetics 2004; 5: 435–445.
3. Xu, J.S.,Wu,Y.T.,Ye,S.J.,Wang,L.,and Feng,Y.Z. SSR primer screening and assessment on pear germplasm resources. J. Central South Univ. Forest.Technol. 2012; 32, 80–85.
4. Hodel et al. Using microsatellites in the 21st century. Applications in Plant Sciences 2016 4(6)
5. Leclercq, S., Rivals, E., Jarne, P. Detecting microsatellites within genomes: significant variation among algorithms. BMC Bioinformatics 2007, 8:125.
6. Grover A, Aishwarya V, Sharma PC. Searching microsatellites in DNA sequences: approaches used and tools developed. Physiol Mol Biol Plants (January–March 2012) 18(1):11–19
7. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J. Mol. Biol. (1990) 215:403-410.
8. Martínez CM. MIDAS: Computer application for the identification of exact and inaccurate microsatellites in genomic sequences. *Revista Cubana de Informática Médica*, Volúmen 18, No. 2 (2018).
9. Fleischmann RD, Alland D, Eisen JA, Carpenter L, White O, Peterson J, DeBoy R, Dodson R, Gwinn M, Haft D, Hickey E, Kolonay JF, Nelson WC, Umayam LA, Ermolaeva M, Salzberg SL, Delcher A, Utterback T, Weidman J, Khouri H, Gill J, Mikula A, Bishai W, Jacobs Jr WR, Venter JC, Fraser CM: Whole-genome comparison of Mycobacterium tuberculosis clinical and laboratory strains. J Bacteriol 2002, 184(19):5479-5490.
10. Sreenu V, Kumar P, Nagaraju J, Nagarajaram H. Microsatellite polymorphism across the M. tuberculosis and M. bovis genomes: Implications on genome evolution and plasticity. BMC Genomics 2006, 7:78.
11. Supply P, Marceau M, Mangenot S, Roche D, Rouanet C, Khanna V, et al. Genomic analysis of smooth tubercle bacilli provides insights into ancestry and pathoadaptation of Mycobacterium tuberculosis. Nat Genet. 2013 Feb; 45(2):172–179. doi: 10.1038/ng.2517 PMID: 23291586.
12. Warholm P, Light S. Identification of a Non-Pentapeptide Region Associated with Rapid Mycobacterial Evolution. PLoS ONE (2016), 11(5): e0154059. doi:10.1371/journal.pone.0154059