**Algorithm Specification for Automatic Strain Resolution**

**Input:**

1. The contig sequence in a FASTA file.

2. All aligned reads in a SAM file.

**Output:**

1. Strain blocks

2. Aligned reads to each strain block in SAM files

Usage:

This python script requires python 2.7, biopython (http://biopython.org/), and Pysam (<https://code.google.com/p/pysam/>);

Command:

python autostrainer.py -f input\_fasta\_file\_name (required), -m input\_sam\_file\_name (required) -o output\_file\_path (optional) -e error threshold (optional) -k number of variant paths (optional).

By default, the output\_file\_path will be the folder containing the input SAM file, error threshold is 2, while the number of variant paths is 3.

Command Sample:

python autostrainer.py -f test/GWA2\_AR10\_Chromosome\_circular\_complete-FINAL\_recirc.fasta, -m test/gwa2-vs-gwa2\_ar10\_final5\_shrunk.sam -o output/ -e 2 -k 3

Algorithm:

Prarameters list:

e: error threshold, specified by -e, default value is 2

k: number of variant paths, specified by -k, default value is 3

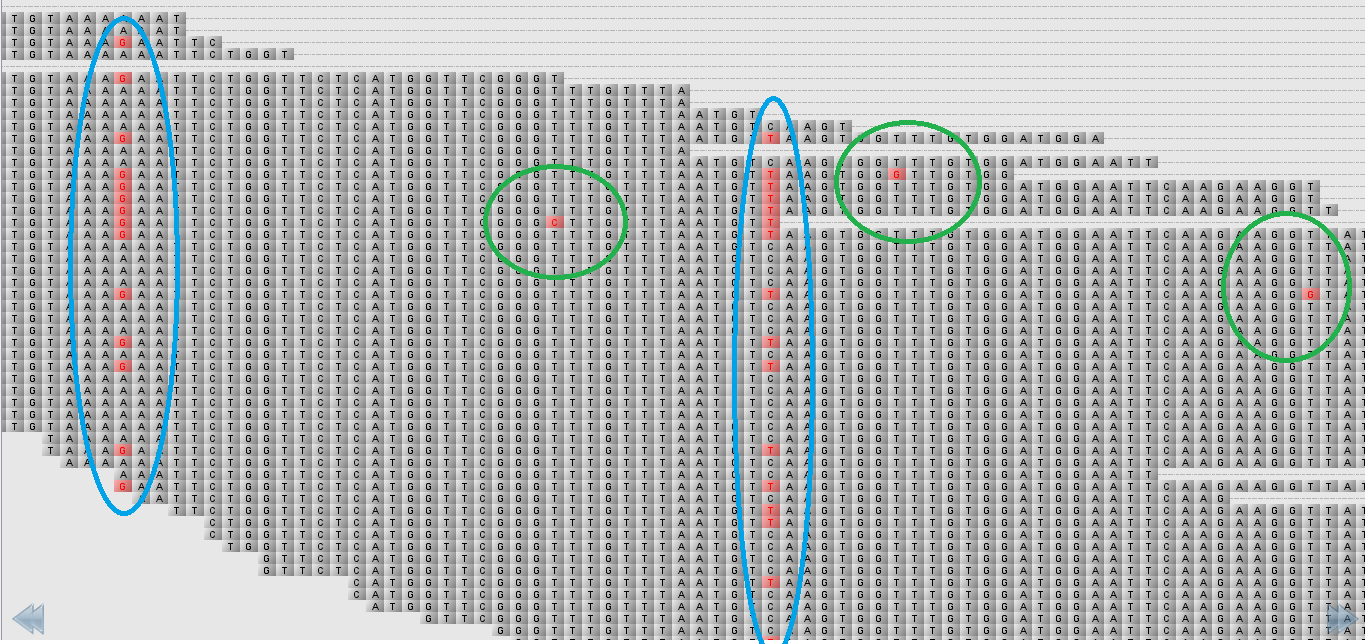
**Step1: Find variant positions**

On a position, the nucleotide on the reference genome is represented by NTref. In this step, our algorithm identifies all variant positions. Besides regular nucleotides A, C, G, and T, which we call regular nucleotides, this algorithm also considers wild card nucleotides denoted by symbols other than A, C, G, and T.

If NTref is a regular nucleotide, all read nucleotides with wild cards will be treated as same as NTref. Given NTref is A, if there are at least (*e+1*) C, G, or T from reads on this position, this is a variant position; otherwise, it is a consistent position. Similarly, this algorithm identifies variant positions given NTref is C, G, or T.

If NTref is a wild card, and there is at least one read nucleotide is a regular nucleotide, all wild card regular nucleotides are treated as same as the majority of regular read nucleotides. If there are at least (e+1) same regular nucleotides, this is a variant position; otherwise, this is a consistent position.

If NTref and all read nucleotides are wild card nucleotides, this is a consistent position



With error threshold e = 2, blue circles indicate variant positions, while green circles are just for errors, which will be ignored.

**Step 2: Output consistent reads**

All reads without being involved in a variant position are outputted to a file with extension name ref.sam. All other reads are stored in the **all-reads list**.

**Step 3: Build variant paths**

In this step, the algorithm handles reads involved in variant positions. A variant path contains reads without conflict on all variant positions. Three lists are used in this step. All available reads are stored in the **all-reads list**. Reads conflicting with current variant choice should be sent to the **black list**. Reads on the **black list** can’t be used in the current variant path**.** Once a read is put into the **black list**, the corresponding mate read will also be sent to the **black list**, and both of them have no chance to be deleted on the **black list**, until the current variant path is done**.** Reads chosen to form the current variant path is stored on the **white list**. Reads on the **white list** may be deleted from the **white list**, since some reads may conflict future choices. Reads in the *i*th variant path will be outputted to file var*i*.sam.

Preprocessing:

As for each variant position, if the number of a type of regular nucleotides is at most e, which is the user defined error threshold, all of them are treated as same as NTref (NTref is a regular nucleotide), or the majority of regular nucleotides (NTref is not a regular nucleotide).

For variant path i from 1 to k,

1. initialize an empty **black list** and an empty **white list**;

2. for each variant position from 5’ to 3’:

2.1 if a read involved on this position is available on the **black list**, this read won’t be considered in the following steps for building variant path *i*;

2.2 pick the reads with the majority nucleotides on this position;

2.3 add picked reads to the **white list**; if their mate reads are on **all-reads list**, also add them to the **white list**;

2.4 for each unpicked read:

if it has been added to the **white list**, delete it from the **white list**;

if its mate read is on the **white list**, delete the mate read from the **white list**;

add this read to the **black list**;

if its mate read is on the **all-reads list**, add this mate read to the **black list**;

3. Output all reads on the **white list** to a sam file with extension name var*i*.sam;

4. Delete all reads on the **white list** from the **all-reads list**;

5. Empty the **black list** and the **white list**.

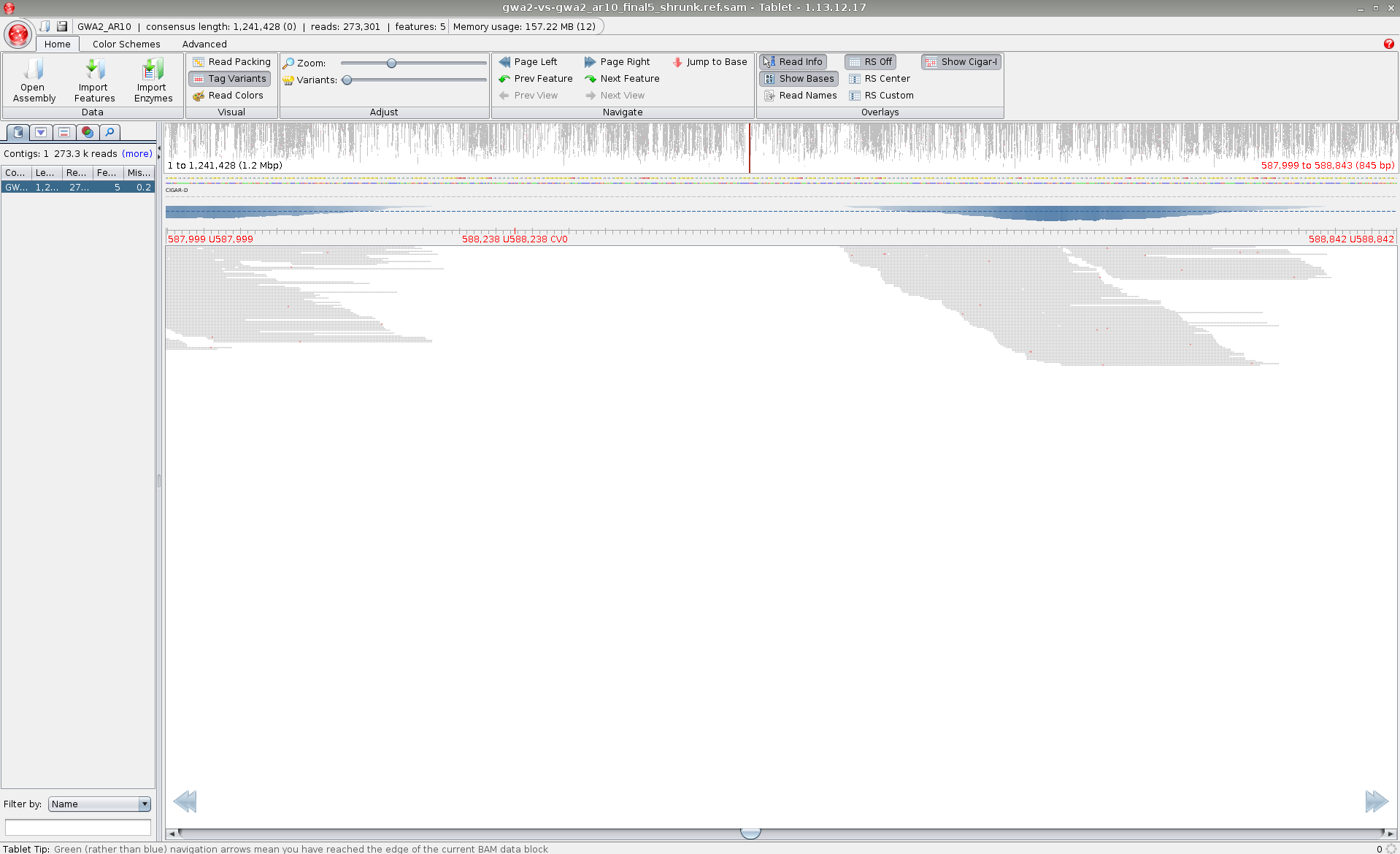
Basically, when a read is added to the **black list**, it can’t be on the **white list**. Once a read is added to the **black list**, it will stay there until the current variant path is done. A read and its mate read should be considered together.

**Output example:**

Here the same location on the genomes are shown for the three output SAM files: ref, var1, and var2.

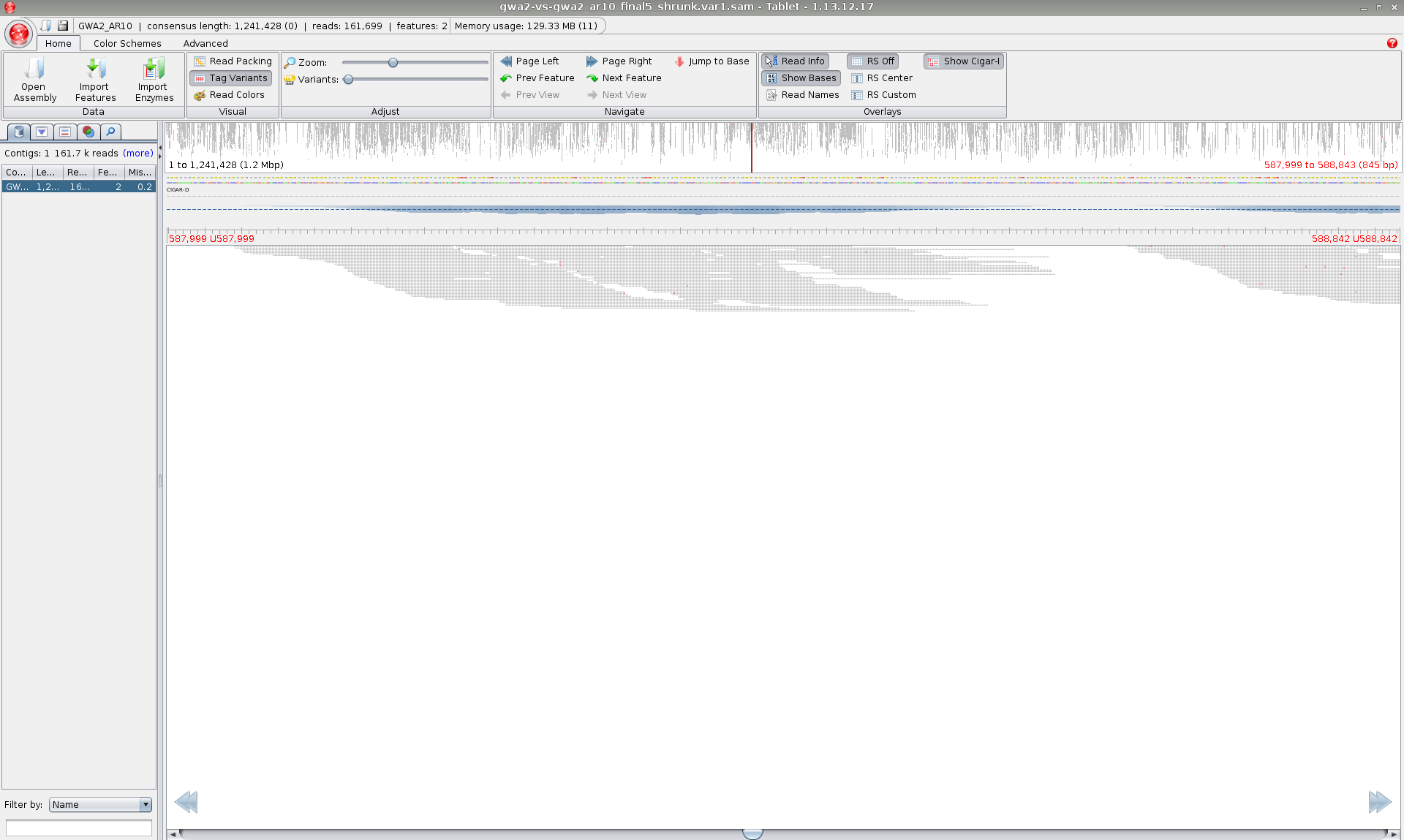
Ref with consistent reads:

The location with variant positions doesn’t have any reads remaining in the Ref SAM.



variant path 1:

Most of those reads on these variant positions are moved to var1 and these reads contained the same sequence as the reference sequence.



variant path 2:

Some of the reads are moved to var2, because they contained linked variants.

