

A Self-Tuning Method for One-Chip SNP Identification

Michael Molla^{1,2,*}, Jude Shavlik¹, Thomas Albert², Todd Richmond², and Steven Smith²

¹University of Wisconsin-Madison

²NimbleGen Systems, Inc.

{molla,shavlik}@cs.wisc.edu, {talbert,trichmond,ssmith}@nimblegen.com

Abstract

Current methods for interpreting oligonucleotide-based SNP-detection microarrays, SNP chips, are based on statistics and require extensive parameter tuning as well as extremely high-resolution images of the chip being processed. We present a method, based on a simple data-classification technique called nearest-neighbors that, on haploid organisms, produces results comparable to the published results of the leading statistical methods and requires very little in the way of parameter tuning. Furthermore, it can interpret SNP chips using lower-resolution scanners of the type more typically used in current microarray experiments.

Along with our algorithm, we present the results of a SNP-detection experiment where, when independently applying this algorithm to six identical SARS SNP chips, we correctly identify all 24 SNPs in a particular strain of the SARS virus, with between 6 and 13 false positives across the six experiments.

1. Introduction

To date, the genomes of hundreds of organisms have been sequenced. For each of these organisms, a consensus or reference sequence has been deposited into a public database. Though this sequence matches the particular individual whose genome was sequenced, other individuals of this species will differ slightly from this reference sequence. One way to identify these differences is to completely sequence, from scratch, the genomes of other individuals of this species and then do a comparison. However, this is very costly and generally impractical. Since most of the genetic variation between individuals is in the form of Single Nucleotide Polymorphisms (SNPs; Altshuler et al., 2000), a much more cost-effective approach is to use the reference sequence as scaffolding and identify variations from this sequence in various individuals.

This technique is known as resequencing (Saiki et al. 1989).

One method of resequencing that has shown significant results utilizes oligonucleotide microarray technology (Hacia, 1999). In particular, this type of resequencing chip consists of a complete tiling of the reference sequence – that is, a chip containing one probe corresponding exactly to each 29-mer in the reference sequence – plus, for each base in this sequence, three mismatch probes: one representing each possible SNP at this position (see the next section for a more detailed description of this method). In theory, any time a SNP is present, the mismatch probe representing this SNP will have a higher intensity signal than the corresponding probe that matches the reference sequence. However, due to unpredictability in signal strength, varying hybridization efficiency, and various other sources of noise, this method typically results in many base positions whose identities are incorrectly predicted. In other words, among all the cases where a mismatch probe has more signal intensity than that of the reference sequence's probe, we would like to accurately separate the true SNPs from the noisy, false positives.

Current approaches to this noise-reduction problem (Cutler et al., 2001) require extensive parameter tuning involving the analysis of very large amounts of data. This tuning needs to be re-run any time experimental conditions are changed. Another limitation of current methods is that, in order to have a single probe represented by a sufficient number of pixels, a very expensive high-resolution scanner must be used.

We present a technique that uses a simple data-classification technique to differentiate potential SNPs from chip noise. Unlike other methods, ours does not require such a high-resolution scanner and furthermore requires very little tuning outside of the single chip being analyzed. For the haploid SARS strain we use to evaluate our approach, our algorithm produces results similar to the published results in SNP identification

rate for the best known of the current statistical methods (Cutler et al., 2001). Our method uses only the mean signal intensity of each probe on the chip and no data from outside of the chip.

2. Task definition

Our task is to identify SNPs (Single Nucleotide Polymorphisms) in the context of oligonucleotide-microarray-based DNA resequencing (Nuwaysir et al. 2002; Singh-Gasson et al., 1999). This type of resequencing consists of fully tiling (making probes corresponding to every 29-mer in) the reference sequence of an organism's DNA through a region of interest. For each of these probes, another three *mismatch* probes are generated. Each of these has a different base in its center position. For example, if the organism's reference DNA includes the sequence:

3' -CTGACATGCAGCTATGCATGCATGAA-5'

the corresponding *reference probe* will be its reverse complement and, therefore, be the sequence:

5' -GACTGTACGTCGATACGTACGTACTT-3'

and the corresponding *mismatch probes* will be the sequences:

5' -GACTGTACGTCGAAACGTACGTACTT-3'

5' -GACTGTACGTCGACACGTACGTACTT-3'

5' -GACTGTACGTCGAGACGTACGTACTT-3'

We call a group of probes such as this that represent all possible SNPs at a given position a *quartet*.

Our approach does involve the creation of $4N$ probes, where N is the length of the organism's DNA sequence. Creating this many probes is currently not feasible for large genomes, such as human, rat, etc., but it is feasible for viruses, bacteria, and for limited regions of interest in a large genome.

We can summarize the task of interpreting such a resequencing chip as follows:

- Given:** The data from a single resequencing chip, representing either the complete genome of an organism, or some regions of interest in such a genome.
- Do:** Identify, from among the positions at which the sample sequence seems to differ from the reference sequence, which of these positions are likely to be real SNPs rather than noise and return these positions along with a confidence measure for each.

3. Our approach

After the chip has been exposed to the sample, each of the probes will have a resulting intensity. We also call each quartet's set of four such intensities an *example* (we use this term taken from machine learning because our solution is built upon a technique from machine learning). For most of these examples, the highest of the four intensities will be the reference probe, i.e., the probe with no mismatch base. We call examples for which this is the case *conformers* (a table appearing later, Table 2, provides an illustration) since they conform with what we expect given the reference sequence. When one of the *mismatch* probes has the highest intensity, we call the quartet a *non-conformer*. Some of these non-conformers reflect actual SNPs in the DNA of the organism. However, most of them are the results of hybridization failures or other types of noise and do not represent an actual SNP in the sample. Note that, though the task of separating conformers from non-conformers is a trivial data-processing step, separation of the non-conformers that truly are SNPs from the non-conformers that arise from noise in the data is not.

We posit that one can perform the task of accurately separating the non-conformers that truly are SNPs from the noisy non-conformers by applying what is called the *nearest-neighbor* method (Mitchell, 1997). In this method one plots examples in an N -dimensional space, where the dimensions are features of the examples. In order to interpret an example in this *feature space*, one looks at the K examples nearest to it in this space and uses their classifications to interpret the example in question. Described in detail in the next section, the feature space for this task is defined by the intensities of the four probes in each quartet.

In the traditional manner for applying the nearest-neighbors method (which we do *not* follow in this work), one would manually label a "training set" of quartets as being either true SNPs – non-conformers that arise from a one-base difference between the sample sequence and the reference sequence – or false SNPs, non-conformers that arise from noise in the microarray experiment. The nearest-neighbors algorithm would then use these labeled examples when it needed to categorize future non-conformers.

In this case, however, this approach would not be feasible. It would require that someone laboriously collect each of these training examples. Worse still, whenever the chip chemistry or any other laboratory condition changed, one would need to collect an entirely new set of training examples. This is because

the underlying process that generated the noise would probably have changed.

Instead we apply the nearest-neighbor approach *without needing human-labeled examples*. Our key idea is that examples involving bad microarray hybridizations will tend to group together in different portions of feature space than examples from good hybridizations. Once we have separated “noisy” examples from good examples, we can identify SNPs by simply finding examples where the highest-scoring base is not the base in the reference sequence. This is possible because of the nature of our particular task. Specifically, we rely on the following three assumptions, which have held true in all of the data we have looked at so far, including the data used in the experimental section of this article:

- 1) Examples resulting from proper probe-target hybridizations will be much nearer to each other in feature space than to examples resulting from hybridization failures.
- 2) The majority of non-conformers are due to noise in the data rather than SNPs. (Even if this assumption is false for a given data set, one could also include data from other chips containing few or no SNPs.) Hence, we can safely ignore, when looking for SNPs, those areas in

feature space dense with non-conformers.

- 3) SNPs are relatively rare. Hence SNPs involved in successful hybridizations will fall in regions of feature space that are surrounded by conformers.

Figure 1 illustrates these assumptions. Non-conformers falling in areas of feature space dense with conforming examples can be predicted to be SNPs (i.e., the difference from the expected result that lead to this example being called a non-conformer is likely due to a base difference from the reference sequence rather than from a failed hybridization). Non-conformers surrounded by other non-conformers can be viewed as noisy data. In addition, the likelihood that any given example in an area is a hybridization error can be roughly estimated by the density of non-conformers in that area. By performing this estimation for each of the non-conformers, we find an approximate likelihood that it is the result of a hybridization error.

Note that, though our approach makes use of labeled examples, it does not require a human to label any examples as being SNPs or not. Instead, our possible labels are *conformer* and *non-conformer*, a distinction computed simply from the probe intensities in an example (i.e., group of four probes). In other words, our task is *not* to predict if an example is a conformer or not – that distinction can be made via a simple calculation. Instead, we use the idea of finding the

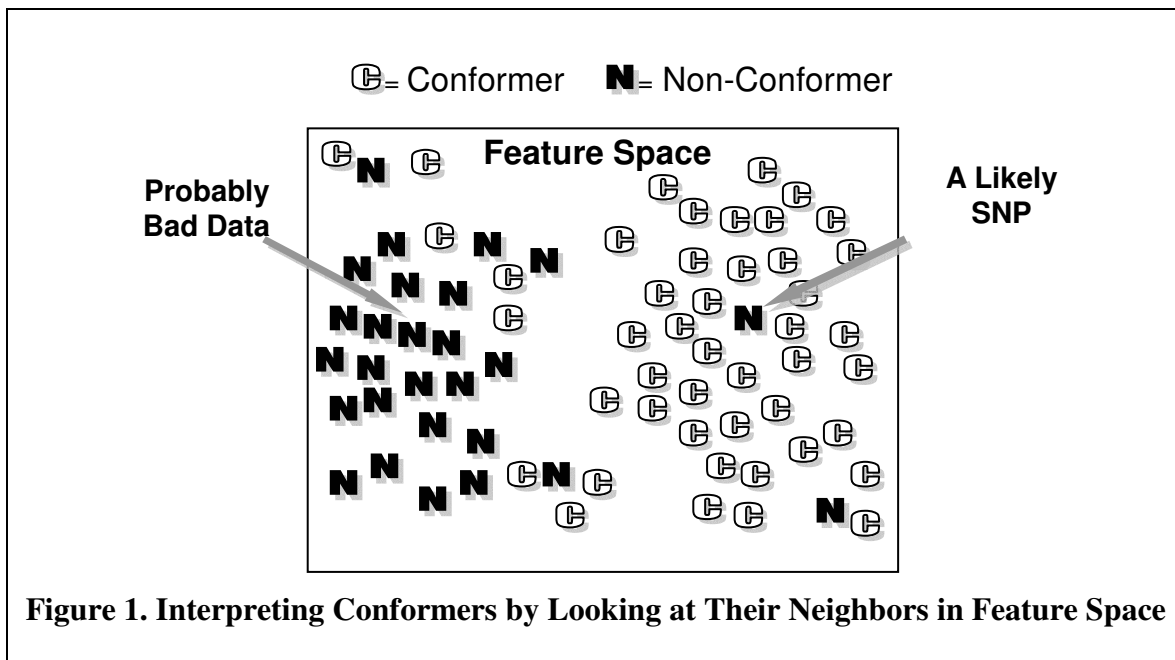


Figure 1. Interpreting Conformers by Looking at Their Neighbors in Feature Space

Table 1. Our Algorithm

Given K and <i>threshold</i> (In our experiments, except where otherwise noted, $K = 100$, <i>threshold</i> = .97) For each example Find the K examples closest to this example in feature space These are this example's K nearest neighbors Let P = the number of these K nearest neighbors that are conformers If $P / K > \textit{threshold}$ If the actual category of this example = <i>conformer</i> Classify this example as a <i>non-SNP</i> Else Classify this example as a <i>candidate SNP</i> Else Classify this example as a <i>non-call</i> (i.e., possibly bad data)

nearest neighbors in a feature space to separate (a) non-conformers produced by SNPs from (b) non-conformers resulting from hybridization failures. We hypothesize that the former are likely to be surrounded by conformers while the later are likely to be surrounded by other non-conformers.

4. Our algorithm

Table 1 contains our algorithm for SNP-detection in microarrays. This *K-nearest-neighbor* algorithm involves plotting each example in feature space and then, for each of these examples, finding the K other examples nearest to it in this space. The categories of these K neighbors determine the prediction. If greater than some *threshold* of these neighbors are conformers, we infer that the example is *not* the result of a failed hybridization. If such an example is a non-conformer, we thus classify it as a SNP. Otherwise, we infer that the sample sequence does match the reference sequence at this base position and explicitly classify it as a *non-SNP*. Should an insufficient number of neighbors be conformers, we view the example as being noisy and classify it as a *non-call* regardless of whether or not it is a conformer. The fraction of conformers among the K neighbors can be used as a measure of confidence in the prediction.

The appropriate value for K and *threshold* and appropriate definitions of *nearness* and *feature space* vary between learning tasks. The later two choices are of particular importance in this case since Assumption 1 from the previous section will clearly not hold unless *nearness* and *feature space* are defined properly. In

this task, our *feature space* – see Table 2 – is the 5-dimensional space of examples, where 4 of the dimensions correspond to the intensities of the 4 probes in the example and the 5th dimension is the identity of the base in the reference sequence. Instead of defining *nearness*, we define its inverse, *distance*. We define *distance* between two probes to be infinite in cases where the two examples differ in the 5th dimension. Otherwise, it is defined as the one-norm distance between the examples or:

$$\textit{distance}(\textit{example}_j, \textit{example}_k) = \sum_{i=1}^4 | \textit{feature}_i(\textit{example}_j) - \textit{feature}_i(\textit{example}_k) |$$

where $\textit{example}_j$ and $\textit{example}_k$ are two quartets, and $\textit{feature}_i(\textit{example})$ is the intensity of the i^{th} most intense probe in *example*.

In addition to the feature space described above, we tried two other slight variations that did not work nearly as well. The first unsuccessful variant only used the four features that represent the signal intensities; it ignored the identity of the reference base. Our best guess as to why this technique was not successful is that hybridization characteristics, the affinity between a given probe and the sample, vary slightly across the different nucleotides. As a result, the identity of the reference base carries with it some information about typical patterns of intensity. The second variant we tried did not sort the probe signals by their intensity. Rather, it compared neighbors' intensities on a nucleotide-by-nucleotide basis; that is, the two examples' intensities for the probe with an A in the middle were compared, then for the two probes with a G in the middle, etcetera for C and T.

Table 2. The Features Used to Describe the Quartets

Reference Sequence: AGCGCTTTAAGCATATATCCATCCTAGCATACGATCTTTATACTTACATTACCCT...																																											
Resequencing probes (reference probes in bold)																																											
...	...																																										
Quartet 7:	TTTAAGCATATATCAATCCTAGCATACGA ← Probe 7A TTTAAGCATATATCCATCCTAGCATACGA ← Probe 7C TTTAAGCATATATCGATCCTAGCATACGA ← Probe 7G TTTAAGCATATATCTATCCTAGCATACGA ← Probe 7T																																										
Quartet 8:	TTAAGCATATATCGATCCTAGCATACGAT ← Probe 8A TTAAGCATATATCGCTCCTAGCATACGAT ← Probe 8C TTAAGCATATATCGGTCCTAGCATACGAT ← Probe 8G TTAAGCATATATCGTTCCTAGCATACGAT ← Probe 8T																																										
Quartet 9:	TAAGCATATATCGAACCTAGCATACGATC ← Probe 9A TAAGCATATATCGACCCTAGCATACGATC ← Probe 9C TAAGCATATATCGAGCCTAGCATACGATC ← Probe 9G TAAGCATATATCGATCCTAGCATACGATC ← Probe 9T																																										
...	...																																										
Resulting Intensities (obtained by exposing the chip to the sample)																																											
<table border="1"> <thead> <tr> <th>Probe</th> <th>Intensity</th> </tr> </thead> <tbody> <tr><td>...</td><td>...</td></tr> <tr><td>7A</td><td>1543</td></tr> <tr><td>7C</td><td>3354</td></tr> <tr><td>7G</td><td>342</td></tr> <tr><td>7T</td><td>737</td></tr> <tr><td>8A</td><td>1456</td></tr> <tr><td>8C</td><td>2432</td></tr> <tr><td>8G</td><td>212</td></tr> <tr><td>8T</td><td>334</td></tr> <tr><td>9A</td><td>332</td></tr> <tr><td>9C</td><td>456</td></tr> <tr><td>9G</td><td>232</td></tr> <tr><td>9T</td><td>2443</td></tr> <tr><td>...</td><td>...</td></tr> </tbody> </table>	Probe	Intensity	7A	1543	7C	3354	7G	342	7T	737	8A	1456	8C	2432	8G	212	8T	334	9A	332	9C	456	9G	232	9T	2443	<p>← The reference probe for quartet 7 is 7C. This is also the highest-intensity probe in this quartet. Hence, we call quartet 7 a <i>conformer</i>.</p> <p>← Note that, though the reference probe from quartet 8 is 8A, the highest intensity probe from this quartet is 8C. We call such a quartet a <i>non-conformer</i>.</p>												
Probe	Intensity																																										
...	...																																										
7A	1543																																										
7C	3354																																										
7G	342																																										
7T	737																																										
8A	1456																																										
8C	2432																																										
8G	212																																										
8T	334																																										
9A	332																																										
9C	456																																										
9G	232																																										
9T	2443																																										
...	...																																										
The Feature Set																																											
Each quartet produces one example. The features are the reference base and the four sorted intensities (note that the feature set contains no information about which actual probe has the highest intensity). The category of the example is either <i>conformer</i> or <i>non-conformer</i> , that is whether or not this quartet's highest intensity probe is the reference base.																																											
<table border="1"> <thead> <tr> <th>Example</th> <th>Reference Base</th> <th>Intensity 1</th> <th>Intensity 2</th> <th>Intensity 3</th> <th>Intensity 4</th> <th>Category</th> </tr> </thead> <tbody> <tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr> <tr><td>7</td><td>C</td><td>3354</td><td>1543</td><td>737</td><td>342</td><td>conformer</td></tr> <tr><td>8</td><td>A</td><td>2435</td><td>1456</td><td>334</td><td>212</td><td>non-conformer</td></tr> <tr><td>9</td><td>T</td><td>2443</td><td>456</td><td>332</td><td>232</td><td>conformer</td></tr> <tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr> </tbody> </table>	Example	Reference Base	Intensity 1	Intensity 2	Intensity 3	Intensity 4	Category	7	C	3354	1543	737	342	conformer	8	A	2435	1456	334	212	non-conformer	9	T	2443	456	332	232	conformer	
Example	Reference Base	Intensity 1	Intensity 2	Intensity 3	Intensity 4	Category																																					
...																																					
7	C	3354	1543	737	342	conformer																																					
8	A	2435	1456	334	212	non-conformer																																					
9	T	2443	456	332	232	conformer																																					
...																																					

This method seems to suffer from the fact that, on average, only one in $4! = 24$ training examples will have the same order of intensities as a given test example. If the features are not sorted, it is unlikely that two examples whose probe intensities are not similarly ordered will be close enough to each other in feature space to be considered “nearest neighbors”. If the feature space were more densely populated, this may not be a problem. However, in this case, there may not be enough training examples to support our method under these circumstances.

We could have used alternative distance measures as well, such as Euclidean distance, but the absolute-value approach we chose (sometimes called the *one-norm*), is more computationally efficient since a large number of calls to the squaring function are eliminated.

5. Evaluation

For purposes of evaluation, we compare our algorithm to a simple alternative, which we call our *baseline algorithm*. Table 3 contains this baseline algorithm, which simply compares the highest intensity probe to the second highest. If the ratio is above a threshold value, the algorithm assumes that the base represented by the highest intensity probe is the base in the sequence. If this quartet is a non-conformer, our baseline algorithm calls it a candidate SNP. It should be noted that this baseline algorithm is not the state of the art in SNP-finding software. That will be discussed later. Our baseline algorithm is simply a basic straightforward interpretation of the results of a resequencing chip.

Before turning to evaluating our approach on some real genomic data, we discuss the computational demands of our algorithm (Table 1). It is possible to implement clever data structures that support fast determination of the k nearest neighbors (Liu et al., 2003), e. g., logarithmic in the number of examples. However, we

have not implemented such data structures because we can process the data from one microarray in a matter of minutes with a simple linear-time algorithm (linear per example, so overall it is quadratic in the length of the DNA sequence). Our algorithm’s runtime is much less than the time it takes to run the “wet lab” phrase of a microarray experiment, so the algorithm is fast enough for our purposes. It takes approximately fifteen minutes to process a typical two-hundred-thousand-probe chip using a 1.5-gigahertz Pentium processor and 512 megabytes of RAM. Though this is longer than typical statistical methods, it is not a significant contributor to the time required for preparation and analysis of such a chip.

In order to evaluate our algorithm, we chose a useful, realistic task. One strain of the SARS virus (Ruan, 2003) has been completely sequenced via standard capillary sequencing. We were supplied with a different sample strain. This sample differed from the reference sequence to an unknown degree. Our task was to identify candidate SNPs in this strain. Our predictions would subsequently be evaluated using further capillary sequencing and various other “wet” laboratory methods (Wong, 2004).

Using the reference sequence, we designed a resequencing chip including both the forward and reverse strands of this virus. We then exposed this chip to the sample. After that we used our algorithm to predict the SNPs on this chip. Once these results were obtained, we combined the forward and reverse predictions for each possible SNP position by averaging the two predictions.

Table 3. A Baseline Algorithm

<p>Given <i>threshold</i> For each example Let <i>MaxIntensity</i> = intensity of the highest intensity base in this example Let <i>SecondIntensity</i> = intensity of the 2nd highest intensity base in this example If (<i>MaxIntensity</i> / <i>SecondIntensity</i>) < <i>threshold</i> Classify this example as a <i>non-call</i> Else If the actual category of this example = <i>conformer</i> Classify this example as a <i>non-SNP</i> Else Classify this example as a <i>candidate SNP</i></p>

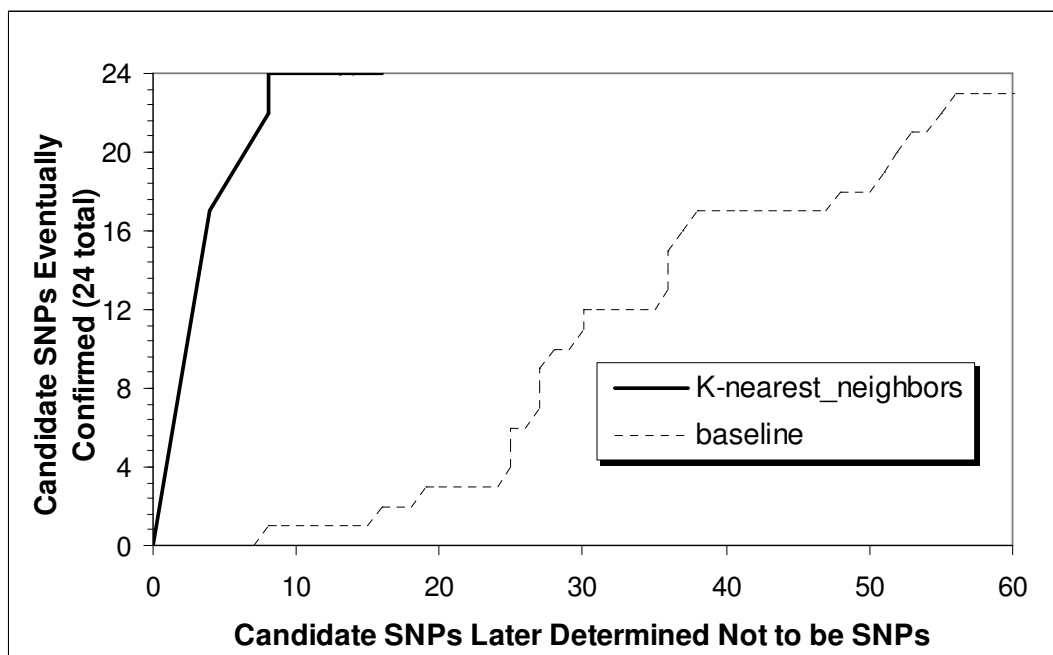


Figure 2: ROC Curve for SARS SNP Detection

Materials and Methods

Preparation and hybridization of SARS Sample. A detailed description of the methods used to prepare and analyze the SARS samples has been previously published (Wong, 2004). Briefly, total RNA is extracted from patient lung, sputum or fecal samples, or from Vero E cultured cells inoculated with SARS-CoV RNA. RNA is reverse-transcribed into double-stranded cDNA. Tissue samples are amplified using a nested-PCR strategy. For each sample, PCR-product fragments are pooled at an equimolar ratio, digested with DNase I (from Invitrogen, Carlsbad, CA) and end labeled with Biotin-N6 ddATP (Perkin Elmer, Wellesley, MA) using Terminal Deoxynucleotidyl Transferase (Promega, Madison, WI).

The arrays are synthesized as previously described (Nuwaysir 2002; Singh-Gasson 1999). The re-sequencing arrays are hybridized with biotinylated DNA overnight, then washed and stained with Cy3-Streptavidin conjugate (Amersham Biosciences, Piscataway, NJ). Cy3 signal is amplified by secondary labeling of the DNA with biotinylated goat anti-streptavidin (Vector Laboratories, Burlingame, CA).

Data extraction and analysis. Microarrays are scanned at 5 μm resolution using the Genepix® 4000b scanner (Axon Instruments, Inc., Union City, CA). The image

is interpolated and scaled up 2.5x in size using NIH Image software (<http://rsb.info.nih.gov/nih-image/>). Each feature on the microarray consists of 49 pixels; pixel intensities are extracted using NimbleScan™ Software (NimbleGen Systems, Inc. Madison, WI).

6. Results and Discussion

Our algorithm performed very well on this task. Out of the 24,900 sequence positions represented by quartets on this chip, 442 are non-conformers. Of these 442, our algorithm identifies 36 as candidate SNPs. Subsequent laboratory experimentation that we performed identified 24 actual SNPs, all of which were among the 36 identified by our algorithm.

All 24 actual SNP's are non-conformers (i.e., quartets where the highest-intensity probe was not from the reference sequence). Note, though, that in general it is possible for a conformer to truly be a SNP; however, our algorithm will not call these as SNPs, at best it will label this quartet as suspicious data. Since the SARS strain we used did not contain any "conforming" SNP's, we are unable to evaluate how well our approach does at labeling such SNPs as non-calls. Of the 24,458 conformers, our algorithm (using the same parameter settings as used for categorizing the non-conformers) only marked 3% as bad data.

In order to verify this result, we generated five more identical SNP chips and exposed them to the same sample using the same values of K and $threshold$ (later in this section we discuss how we choose good values for K and $threshold$). The results varied only slightly. Our algorithm found *all* 24 SNPs in each of the five cases. The number of false positives ranged from 6 to 13.

Our algorithm is largely self-tuning, in that examples are compared to their neighbors in feature space and classifications are made according to the properties of the neighbors, as opposed to specific portions of feature space being pre-labeled as clean or noisy. However, we do have two parameters, K and $threshold$. We next describe some experiments that investigate the sensitivity of our algorithm to the particular settings of these parameters.

In order to choose an appropriate value for K , we tried various values between 1 and 250 to see how many false positives would result if one chose the largest threshold that allowed our algorithm to detect all 24 of the true SNPs. The results of this experiment appear in Figure 3. Fortunately our approach is not overly sensitive to the particular value of K ; we chose $K=100$

and hypothesize that this parameter setting will work well across a wide variety of organisms and strains.

Figure 4 presents the impact of varying $threshold$ (for $K=100$). It reports the number of true SNPs detected, as well as the number of false positives (non-SNPs incorrectly called SNPs). As can be seen, the algorithm's performance is not overly sensitive to the setting for $threshold$. We also anticipate that a single setting for $threshold$ (such as the 0.97 that we use) will work well across many organisms and strains, and hope that neither K nor $threshold$ need to be reset for each new dataset.

Remember, however, that our approach classifies some quartets as *non-calls*, namely those whose neighbors are predominantly non-conformers. The percentage of quartets that are called (either SNP or non-SNP) is typically known as the *call rate*. If this rate is too low, the procedure is of much less use since the algorithm only interprets a small fraction of the data. In order to increase the call rate, one can lower the $threshold$ value. Using our chosen parameter settings we achieve a call rate of over 97%, while still identifying all of the SNPs in the samples we tested and misclassifying only a small number of non-SNPs.

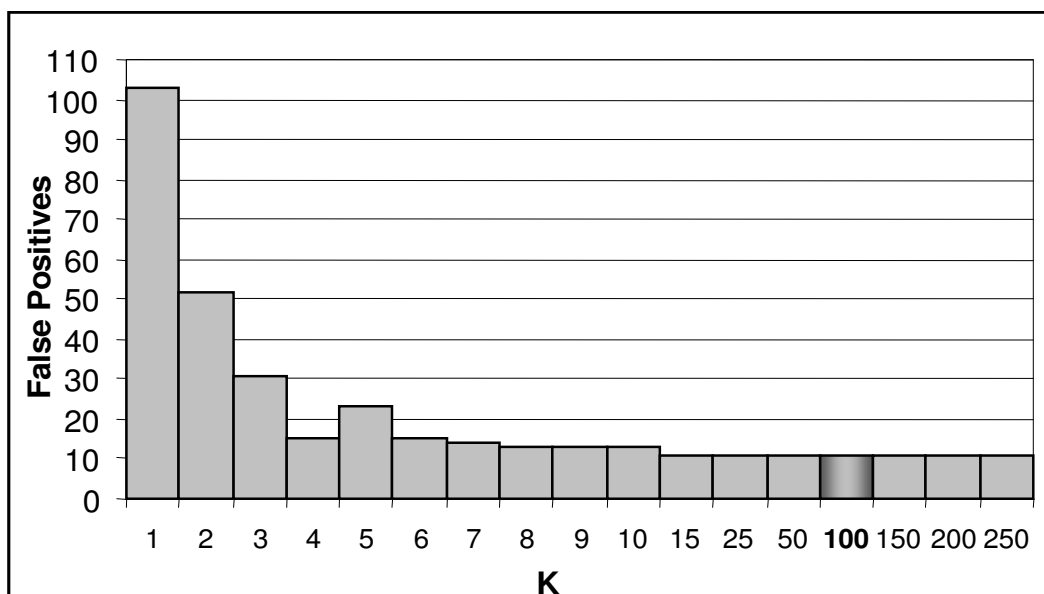


Figure 3. The Impact of K .

The Y-axis reports the number of false positives (noisy examples misclassified as SNPs) that result for the given value of K for the largest $threshold$ that allows our algorithm to detect all 24 true SNPs.

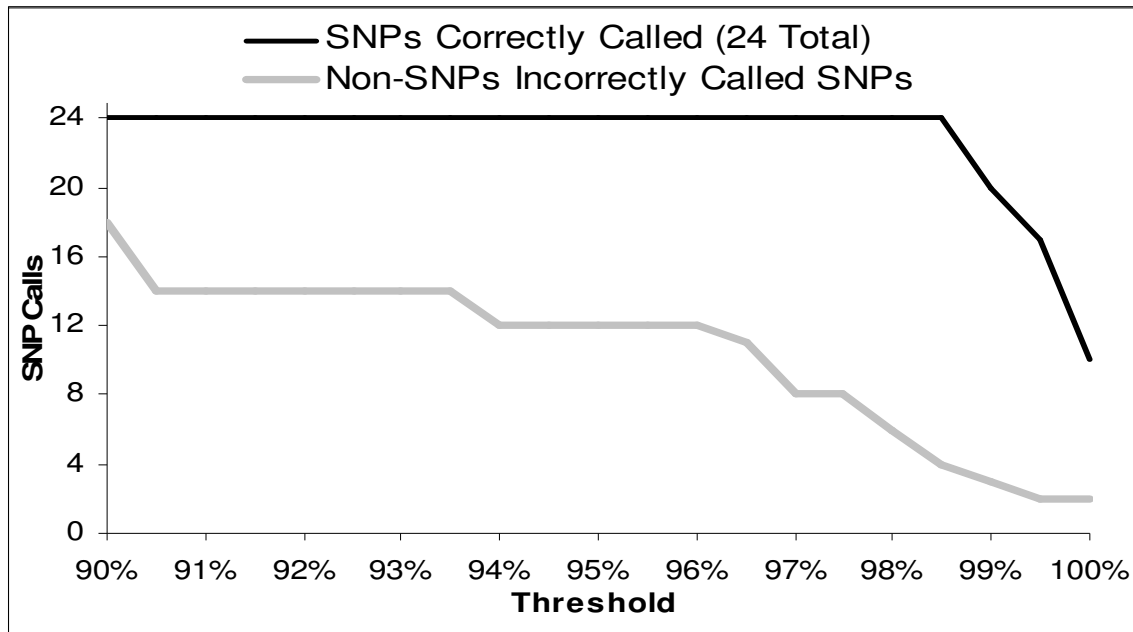


Figure 4. The Impact of the *Threshold* value.

The Y-axis reports the number of SNPs found and the number of false positives that result for the given *threshold* with the value of *K* fixed at 100.

We are unable to directly compare against the haploid SNP calling accuracy of the current standard algorithm, ABACUS, from the Cutler group in conjunction with Affymetrix Corp. However, we believe our results to be comparable to those published by Cutler et al. (2001), while our approach has much less overhead due to tuning and does not require high-resolution scanning. Their published results indicate an emphasis on high-confidence SNPs, at the cost of having a low call rate. The Cutler group's reported accuracy is very good. Of the 108 SNPs they predicted in the human X chromosome, all 108 were verified to be real. However, they report their call rate on the chip as a whole to only be approximately 80%. Though our method is currently geared more toward a high sensitivity to SNPs, we can change this by increasing our threshold from 97% to 99%. Our call rate drops from 97% to 81% and, though we only make 22 SNP calls at that level, only 2 of them are false positives (hence we only detect 20 of the 24 known SNPs). Of course, one should not closely compare results across species, but these numbers do at least suggest the accuracy of our algorithm is on par with that of the Cutler group.

7. Related Work

Several approaches to this problem have been previously tried (Wang et al., 1998, Hirschhorn et al., 2000, Cutler et al., 2001). The most successful to date has been that of the Cutler group in conjunction with Affymetrix. They use parametric statistical techniques that take into account the distribution of pixel intensities within each probe's scanned signal pattern. However, this approach presents a number of limitations. Principal among them is the fact that this method is very sensitive to changes in chemistry, scanner type, and chip layout. In order to overcome some of these problems, extensive parameter tuning is required. This involves the analysis of large amounts of data and needs to be re-run any time chemistry, light-gathering technology, or virtually any other experimental condition is changed. Another limitation is that, in order to have a single probe represented by a sufficient number of pixels, a high-resolution scanner must be used.

8. Future and current work

Efforts are currently underway to apply this method to various other genomes. Evaluation of its performance on larger genomes with varying degrees of complexity and SNP density are of great interest and will, perhaps, lead to further refinement of our algorithm.

We are also in the process of applying this method to the identification of heterozygote SNPs. These are SNPs where two different alleles are present on the sample. Statistical methods for this type of SNP identification exist as well (Cutler et al., 2001). However, they require comparison between multiple individuals of the species and require the same type of high-performance hardware and tuning as previously mentioned. Our method simply uses the same mean-signal intensities and intra-chip self tuning as the homozygote or haploid method already mentioned.

We would also like to decrease the number of probes needed to do such an analysis. Though four probes per base position per DNA strand is much more efficient than other standard methods, the process will need to become still more efficient if it is to handle large genomes. One possible approach would employ a resequencing chip which contains only base positions deemed to have a high probability of being SNPs. Though the resulting chip could be analyzed in the manner described here, it may not yield very good results. This is because our SNP-calling algorithm relies on there being a large number of non-SNPs in the sample along with the SNPs. We plan to extend this method so that it can work with fewer non-SNPs.

We also plan to experiment with a richer feature set. It is possible, for example, that the intensities of probes in quartets representing bases near the genome position represented by a given test example could be of use.

9. Conclusion

Identifying SNPs is an important task. The emerging field of microarray technology has provided us with the tools to identify SNPs in a straightforward way through the use of SNP chips. We have presented here an alternative to the standard method for the interpretation of these SNP chips. Our empirical results on the SARS strains are encouraging, as are the prospects for future SNP detection via this method. Besides its effective SNP-detection ability, additional strengths of our algorithm are (a) that its simplicity means that less calibration is needed, (b) it does most of its calibration on a single chip due to the use of the nearest-neighbor

approach to classification, (c) that training examples for our nearest-neighbor approach are created via our simple-to-implement definitions of conformers and non-conformers (see Table 2), avoiding the need for a human to laboriously label examples, and (d) it does not require the use of high-resolution scanners.

10. Acknowledgements

We would like to thank Edison Liu, Christopher Wong, and Lance Miller of the Genome Institute of Singapore for the SARS samples. This research was partially supported by grants NIH 5 T32 GM08349 and NLM 1 R01 LM07050-01.

11. References

- Altshuler, D., Pollara, V., Cowles, C., Van Etten, W., Baldwin, J., Linton, L. & Lander, E., (2000). An SNP Map of the Human Genome Generated by Reduced Representation Shotgun Sequencing. *Nature* 407:513-516.
- Cutler, D., Zwick, M., Carrasquillo, M., Yohn, C., Tobin, P., Kashuk, C, Mathews, D., Shah, N., Eichler, E., Warrington, J., & Chakravarti, A. (2001). High-Throughput Variation Detection and Genotyping Using Microarrays. *Genome Research* 11:1913-1925.
- Hacia, J. G. (1999). Resequencing and Mutational Analysis using Oligonucleotide Microarrays. *Nature Genetics* 21(1 Suppl):42-7.
- Hirschhorn, J., Sklar, P., Lindblad-Toh, K., Lim, Y., Ruiz-Gutierrez, M., Bolk, S., Langhorst, B., Schaffner, S., Winchester, E., & Lander, E. (2000). SBE-TAGS: An array-based method for efficient single-nucleotide polymorphism genotyping. *Proc. Natl. Acad. Sci.* 97:12164-12169.
- Liu, T., Moore A., & Gray, A. (2003). Efficient exact k-NN and nonparametric classification in high dimensions. *Proceedings of Neural Information Processing Systems*.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, New York.
- Nuwaysir, E., Huang, W., Albert, T., Singh, J., Nuwaysir, K., Pitas, A., Richmond, T., Gorski, T, Berg, J. P., Ballin, J., McCormick, M., Norton, J., Pollock, T., Sumwalt, T., Butcher, L., Porter, D., Molla, M., Hall, C., Blattner, F., Sussman, M., Wallace, R., Cerrina, F., & Green, R. (2002). Gene Expression Analysis Using Oligonucleotide Arrays Produced by Maskless Photolithography. *Genome Research* 12:1749-1755.

Ruan, Y.J., Wei, C.L., Ee, A.L., Vega, V.B., Thoreau, H., Su, S.T., Chia, J.M., Ng, P., Chiu, K.P., Lim, L., Zhang, T., Peng, C.K., Lin, E.O., Lee, N.M., Yee, S.L., Ng, L.F., Chee, R.E., Stanton, L.W., Long, P.M., & Liu, E.T., (2003). Comparative Full-Length Genome Sequence Analysis of 14 SARS Coronavirus Isolates and Common Mutations Associated with Putative Origins of Infection. *Lancet* 361:1779-1785.

Saiki, R., Walsh, P., Levenson, C. & Erlich, H. A., (1989). Genetic Analysis of Amplified DNA with Immobilized Sequence-Specific Oligonucleotide Probes. *Proc. Natl. Acad. Sci. USA* 86:6230-6234

Singh-Gasson, S., Green, R., Yue, Y., Nelson, C., Blattner, F.R., Sussman, M.R., & Cerrina, F. (1999). Maskless Fabrication of Light-Directed Oligonucleotide Microarrays using a Digital Micromirror Array. *Nature Biotechnology*. 17,:974-978.

Wang,D., Fan,J., Siao,C., Berno,A., Young,P., Sapolsky,R., Ghandour,G., Preking,N., Winchester,E., Spencer,J., Kruglyak,L., Stein,L., Hsie,L., Topaloglou,T.,Hubbell,E., Robinson,E., Mittmann,M., Morris,M., Shen,N., Kilburn,D., Rioux,J., Nusbaum,C., Rozen,S., Hudson,T., Lipshutz,R., Chee,M., & Lander,E.(1998) Large-Scale Identification, Mapping, and Genotyping of Single-Nucleotide Polymorphisms in the Human Genome *Science*280:1077-1082.

Wong C., Albert, T., Vega V., Norton, J., Cutler D., Richmond, T., Stanton, L., Liu, E. & Miller, L. (2004). Tracking the Evolution of the SARS Coronavirus Using High-Throughput, High-Density Resequencing Arrays. *Genome Research*, accepted for publication.