UPTEC X 03 017
JUN 2003

ISSN 1401-2138

ERIK ARNER

A quantitative study of the DNP repeat separation method in shotgun sequencing

Master's degree project

UPPSALA
UNIVERSITET

Molecular Biotechnology Programme

Uppsala University School of Engineering

UPTEC X 03 017

Date of issue 2003-06

Author

Erik Arner

Title (English)

A quantitative study of the DNP repeat separation method in shotgun seqeuncing

Title (Swedish)

Abstract

The DNP method for separation of repeats in shotgun sequencing was validated with regards to different parameters in the model, as well as different characteristics of the input data. Results show that the method is robust and flexible.

Keywords

Shotgun sequencing, repeats, DNA, DNP, multiple alignments

Supervisors

Björn Andersson
Karolinska Institutet

Scientific reviewer

Mats Gustafsson
Uppsala Universitet

Project name	Sponsors
Language **English**	Security
ISSN 1401-2138	Classification
Supplementary bibliographical information	Pages **21**

Biology Education Centre Biomedical Center Husargatan 3 Uppsala
Box 592 S-75124 Uppsala Tel +46 (0)18 4710000 Fax +46 (0)18 555217

A Quantitative Study of the DNP Repeat Separation Method in Shotgun Sequencing

Erik Arner

Sammanfattning

Alla organismer har en arvsmassa som består av DNA, en sträcka baser som kan tolkas som bokstäver. Det största problemet vid sekvensbestämning av DNA är att man med dagens teknik endast kan ta reda på ca 500 baser åt gången, medan sträckan man vill bestämma i allmänhet är mångfaldigt längre. För att lösa detta problem använder man sig av shotgunsekvensering. Tekniken innebär att DNA:t kopieras upp och slås sönder på ett slumpmässigt sätt. Då erhålls korta fragment som kan sekvensbestämmas. Fragmenten kommer att överlappa, och utifrån dessa överlapp kan man räkna ut i vilken ordning fragmenten ska sitta.

Dock blir det problem när det DNA man vill sekvensera innehåller repeterade sträckor. Dagens dataprogram klarar inte av att skilja åt snarlika fragment från olika delar av målsekvensen, även om det föreligger skillnader mellan de repeterade enheterna. Anledningen är att sekvensmaskinen gör felklassificeringar av baser i fragmenten emellanåt, och dessa är svåra att skilja från riktiga skillnader mellan repeterade enheter.

Examensarbetet har gått ut på att undersöka och validera DNP-metoden, som har utvecklats för att skilja sekvenseringsfel från skillnader mellan repeterade enheter. Undersökningen har visat att det visserligen finns utrymme för förbättringar av metoden, men att den samtidigt är robust och flexibel.

Examensarbete 20 p i Molekylär bioteknikprogrammet

Uppsala universitet juni 2003

Table of contents

Introduction

The Human Genome Project [1], and its competing private initiative [2], has been accompanied by a tremendous development of sequencing techniques and the computer methods that surround them. The sequencing of a bacterial genome can in these days be performed in a couple of hours, and the promising results of whole genome shotgun sequencing, which the private initiative used to sequence the human genome, has caused the sequencing community to turn its interests towards other higher eukaryotes such as mouse, horse, chicken and dog. In this process, the success of the shotgun method has become evident and it is thus nowadays the prevailing method in large-scale sequencing.

Despite the acknowledgment of the shotgun method as the method of choice when it comes to whole genome sequencing, a number of problems exist with the assembly methods that are commonly used for the processing of the sequenced data. These problems have to be solved if the goal is to produce correct sequences in a rapid fashion. One major problem is the failure of these methods to correctly assemble repeated regions in the target DNA when the repeats are nearly identical and longer than a shotgun fragment length. In particular, the common methods fail to distinguish single base differences between nearly identical repeats from erroneous base calls made in the sequencing stage. The resulting assemblies of such regions are in most cases erroneous and must be followed by manual finishing work. This work is tedious and time consuming, and in many cases it is impossible to determine a correct consensus sequence by hand.

As an example, this problem has become evident in the *Trypanosoma cruzi* sequencing project. *T cruzi*, a protozoan parasite, has a very complicated genome, which consists to more than a third of repeated regions. Uppsala Genome Sequencing Laboratory is one of three laboratories appointed to sequence *T cruzi*, and in addition to the sequencing taking place at the laboratory, an effort has been made to develop an in house shotgun sequencing assembly program, the Tandem Repeat Assembly Program (TRAP), that is capable of assembling nearly identical repeats.

The core of TRAP is a statistical method that is applied to multiple alignments consisting of a shotgun sequence fragment and all its overlaps with other fragments. The rationale for performing such an analysis is that in this way, all information available about a repeat region, and its similar counterparts in other repeat copies, is present in the alignment at the same time. Thus, a more qualified assessment can be made whether bases differing from the consensus in the alignments are due to sequencing errors, or represent single base differences between repeats. Previous methods, that compute statistics on pair-wise overlaps between sequence fragments, have so far been unsuccessful in separating nearly identical repeats.

The purpose of this study is to examine and quantify the performance of the TRAP repeat separation method for different choices of parameters in the model, as well as for different properties of the shotgun fragment data, on which the analysis is performed. A shotgun sequencing simulation program, designed to mimic the real life situation as close as possible, has been developed in order to perform this analysis. The results show that the TRAP method is robust and manages to separate repeats differing only 1%, which is close to the theoretical limit of repeat separation, in data sets containing up to 11% sequencing errors.

Theory

Large-scale sequencing using the shotgun method

The main obstacle when sequencing DNA is the fact that the best high-throughput sequencing methods are capable of sequencing a maximum number of roughly 500-800 bases with reasonable quality. This makes it impossible to sequence stretches of DNA longer than this maximum length in one run. One way of getting around this problem is the use of the shotgun method. In this method, the target DNA to be sequenced is first amplified, e.g. by growing transformed bacteria in culture. The DNA obtained is sheared in a random way, producing fragments that are cloned and subsequently sequenced. From the random shearing, the fragments overlap to different degrees. These overlaps can then be used to puzzle the fragments together in the most probable order. Figure 1 illustrates the schematics of shotgun sequencing.

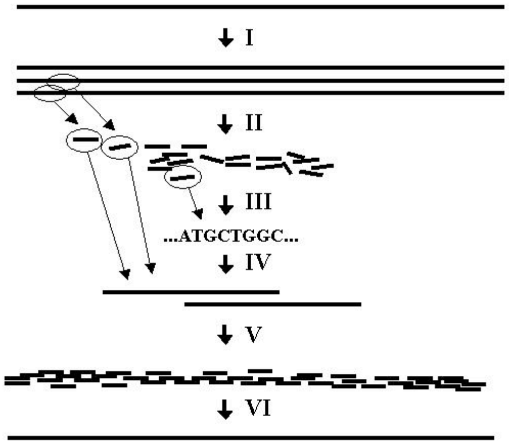


Figure 1. Schematics of shotgun sequencing. **I**. Amplification of target sequence. **II**. Random shearing. **III**. Cloning, sequencing and base calling. **IV**. Detection of pair wise overlaps. **V**. Assembly. **VI**. Computation of consensus sequence.

There are mainly two sequencing approaches that are used today when performing shotgun sequencing, clone-by-clone sequencing or whole genome shotgun. In the first approach, overlapping BAC or cosmid clones are sequenced one at a time, while the whole genome is sequenced at once in the second approach. The whole genome shotgun approach can be much faster than clone-by-clone sequencing provided that there is enough computer power to process the vast amounts of data generated using this method. On the other hand, some problems exist with whole genome shotgun that can lead to long finishing times, e.g. when the genome contains long stretches of repeats. In some cases it may be advantageous to use a combination of the two approaches.

Assembly of shotgun sequencing fragments

A number of algorithms and data structures have been developed over the years that are important to shotgun sequencing assembly. The most important algorithms are probably the Needelman-Wunsch algorithm [3] and its successor, the Smith-Waterman algorithm [4]. The development of these algorithms made it possible to compare two sequences in order to find the most optimal alignment between them in a relatively easy fashion. It is virtually impossible to imagine a computer program that deals with alignments between sequences that does not use some variant of these algorithms.

Two data structures worth mentioning are suffix trees and suffix arrays [5], which both are useful for all-against-all comparisons in a data set. These structures allow for rapid comparisons between strings of text, and for quick location of specific motifs in a data set. The suffix tree is more memory consuming than the suffix array, but allows for faster processing of the data added to the structure.

Another important factor in this context is the rapid development of computer power. Now, a BAC clone can readily be assembled on a plain PC within minutes, a task that would have required significant computer power only a few years ago.

Finally, a significant contribution to the development of algorithms for shotgun sequencing has been the introduction of error probabilities for the sequenced bases. The first assembly algorithms relied on the unrealistic assumption that the input data was error free. This is not the case, although developments in sequencing technology continuously lead to higher quality sequences. Nevertheless, state-of-the-art technology still produces erroneous sequences, which the assembly algorithms have to account for if the goal is to produce correct assemblies. The idea of error probabilities was initially discussed in [6-9], and Phred [10, 11] was the first base calling program to provide error probabilities for each base in the data set, by analyzing the shapes of the electropherograms obtained in the sequencing process.

The following section attempts to outline the main features of a generic assembly program. A typical fragment assembly procedure is divided into four stages: 1. Preparation, 2. Computation of pair-wise overlaps, 3. Contig construction, and 4. Consensus generation. In the first stage, preparation, the input data (i.e. the fragment sequences and corresponding quality files) is screened to enable masking of poor quality regions, known repeat elements, cloning vector sequences etc. Figure 2 shows an example of the quality profile of a shotgun fragment.

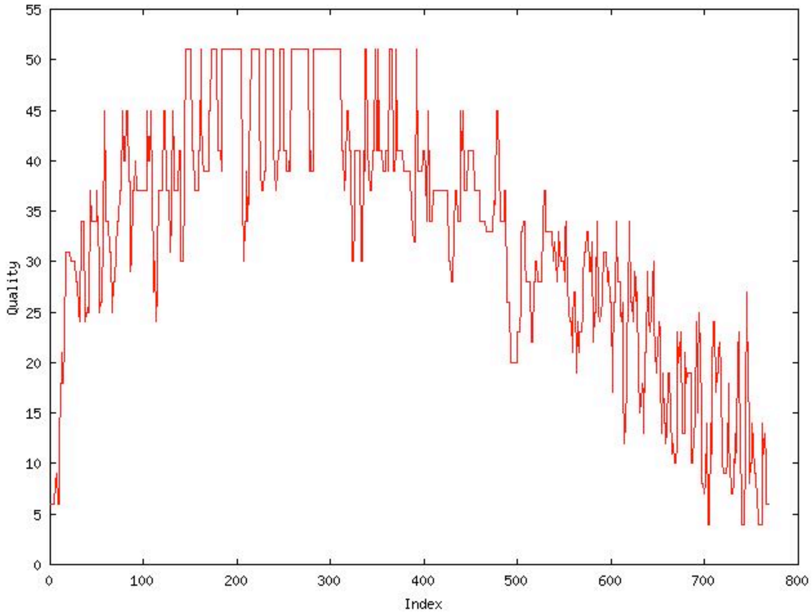


Figure 2. A quality profile of a sequence read from a shotgun project. A quality value *q* corresponds to an error probability $\varepsilon = 10^{-q/10}$.

In the second stage, all pair-wise overlaps between fragments in the data set are located and scored. The fragments are typically ordered into a suffix array or suffix tree, which enables rapid localization of exact matches between fragments. All exact matches longer than a minimum match length are subsequently evaluated as overlap candidates. Usually this is performed using a variant of the Smith-Waterman algorithm. If the score is too low, and/or an overlap criterion is not met, the overlap is discarded. One example of an overlap criterion is to require that the overlap starts in the beginning of one fragment and ends in the end of another (Figure 3).

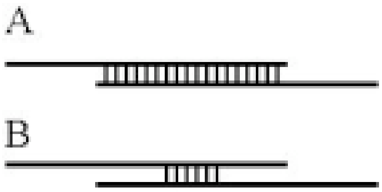


Figure 3. Overlap criterion: an overlap must start in one of the reads and end in one of them. Horizontal bars indicate reads; vertical bars indicate matching bases between the fragments. **A**. The overlap starts in the beginning of the lower read and ends in the end of the upper. Thus, the overlap criterion is met. **B**. The overlap criterion is not met.

In the next stage, an overlap graph is constructed using the computed overlaps from the previous stage and the fragments are assembled into a contiguous sequence (contig). Figure 4 shows a simplified example of this process. Finally, a consensus sequence is generated using one of several possible algorithms, the simplest being to choose the most abundant base in a column as the consensus base (Figure 5).

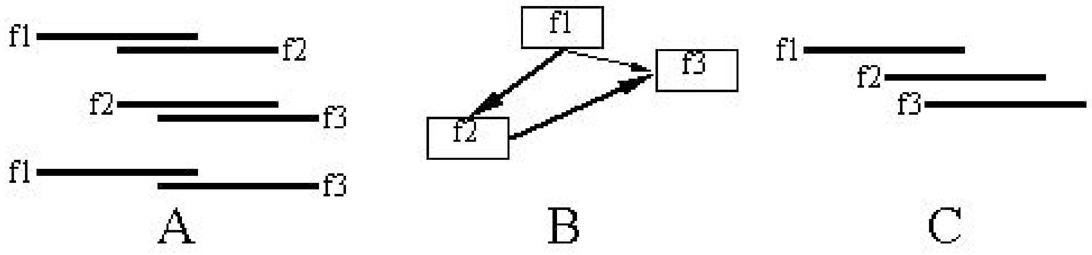
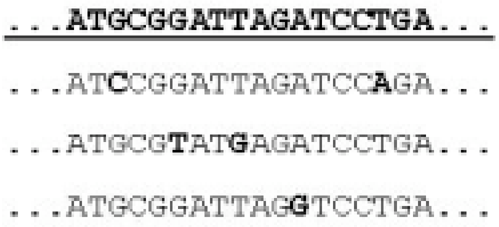


Figure 4. Schematic view of contig construction using an overlap graph. Horizontal bars indicate sequence reads. **A**. Pair wise overlaps between fragments f1, f2, and f3. **B**. Overlap graph constructed from the pair wise overlaps in A. Vertices represent fragments; edges represent pair wise overlaps. Boldface edges suggest a path through the graph following the longest overlaps. **C**. Contig resulting from following the path in B.



```
...ATGCGGATTAGATCCTGA...
...ATCCGGATTAGATCCAGA...
...ATGCGTATGAGATCCTGA...
...ATGCGGATTAGGTCCTGA...
```

Figure 5. Consensus computing. In this example, the consensus is chosen as the most abundant base in each column. The consensus sequence is shown on top, bold letters below indicate bases mismatching consensus.

Some programs that implement these ideas in various forms are PHRAP [12], CAP [13], ARACHNE [14], STADEN [15], TIGR [16], and CELERA [17].

Assembly of repeated DNA

When assembling fragments from a target sequence that contains repeated elements longer than a read length, all the previously developed assembly programs fail if the repeats differ to a low extent (typically 2 % or lower). The resulting assemblies often show repeat fragments merged together in big piles instead of evenly spread according to the sampled coverage of the target sequence (Figure 6). There are two reasons for this: firstly, the previously developed statistical methods to separate repeats are not sensitive enough to detect a small number of differences when there are few present, and distinguish them from sequencing errors. Even when low quality fragments are removed, the mean sequencing error after trimming is often higher than the rate of difference between repeat copies. Secondly, none of the previous algorithms consider the fact that the differing sites in the repeat fragments must be used if the objective is to produce a correct assembly. This is illustrated in Figure 7. For two sequence fragments to be joined together, a distinguishing base unique to one repeat copy must be present along the alignment in both sequences to ascertain that the sequences truly belong together. This also implies that it is impossible to assemble identical repeats with a high degree of confidence. In this case, one can at best guess the correct order of the fragments, e.g. by using some coverage based statistical method. It is hence necessary not only to determine which fragments do not belong together, but also to identify the positions that prove that two fragments are actually sampled from the same repeat copy in the target sequence. None of the previously developed assembly programs acknowledge this requirement.

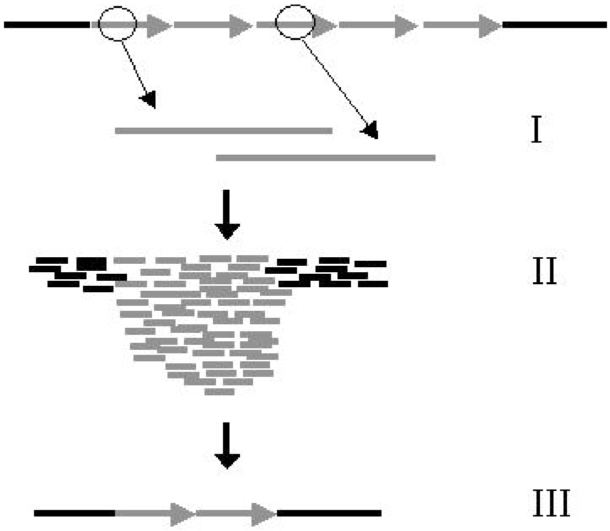


Figure 6. Misassembly of repeated sequences. Gray arrows indicate repeat copies in the target sequence, gray bars indicate sequence reads sampling the repeat region, black bars indicate reads sampling unique parts. **I**. Sequence reads sampling different repeat copies appear to overlap. **II**. The resulting assembly is erroneous, piling reads from different repeat copies. **III**. The consensus sequence is erroneously computed, with repeat copies merged.

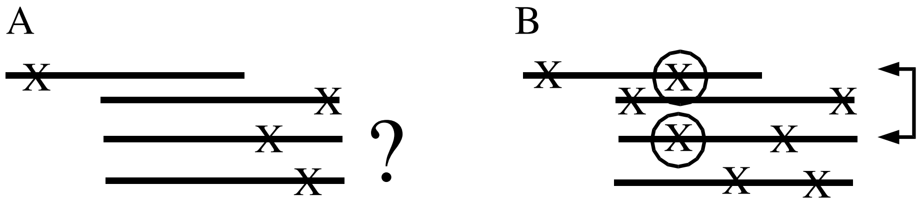


Figure 7. For correct assembly of repeats, detected differences must be explicitly used. Bars indicate reads; X:s indicate detected differences. **A**. If no difference is present along the alignment, it is impossible to determine which reads belong together. **B**. The first and third reads share a difference and can be joined.

The TRAP method for separating repeats

The TRAP method of separating repeats differs from the generic method by the addition of an analysis step before the contig construction stage, and by using the information obtained from the analysis when constructing the contigs. In this stage, all shotgun fragments are analyzed in the following fashion: a fragment is chosen and a multiple alignment consisting of the fragment and all its overlaps with other fragments is constructed. The multiple alignment is optimized locally using the ReAligner method [18], a round-robin algorithm that iteratively re-aligns all sequences until a stable optimum is reached. For each column in the optimized multiple alignment, the most abundant base is chosen as the consensus. The analysis is based on the assumption that deviations from the consensus resulting from sequencing errors are distributed randomly in the alignment, whereas deviations due to single base differences between repeat copies in the target sequence are not. The first part of the analysis consists of locating all columns that are candidates for containing deviations from consensus that result from differences between repeat elements rather than sequencing errors. Once the candidates have been identified, the second step is to evaluate them by performing pair wise comparisons between candidates. If two columns show a similar pattern regarding deviations from consensus in the same fragments, both columns are set to contain Defined Nucleotide Positions, DNPs. This procedure is repeated for all fragments in the data set.

The reason for the requirement of observing two columns with a similar pattern in order for DNPs to be assigned, is that it is necessary in order to get a good separation between the distributions of sequencing errors and real differences between repeats. An alternative strategy would be to look at one column at a time, comparing the number of observed deviations from consensus with what to expect from the error probabilities of the bases on the column. The problem with this approach is illustrated in Figure 8A. Depending on sequence quality, repeat copy number, shotgun coverage, and the difference between repeat copies, the distributions of errors and real differences will overlap to varying degrees. Thus a significant part of the differences will remain undetected if the error is to be maintained at a low level. In contrast, comparing coinciding deviations on column pairs with the expected number yields a better separation between the distributions as can be seen in Figure 8B.

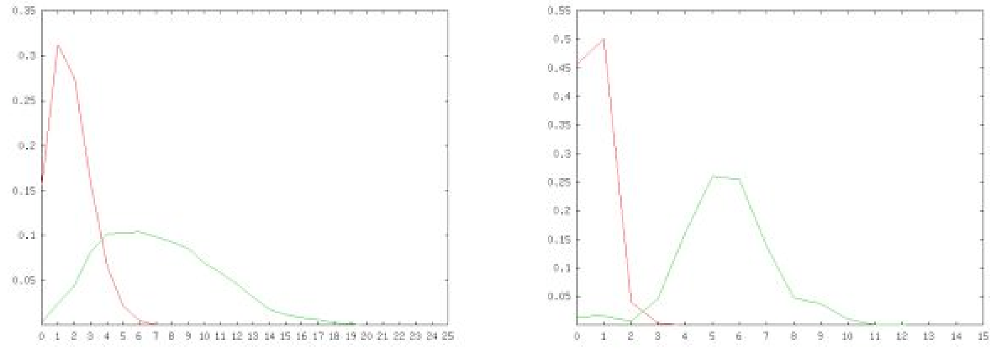


Figure 8. Examples of distributions of sequencing errors and real differences in multiple alignments. **A**. Distribution of deviations from consensus due to sequencing errors and real differences between repeat copies.

The distribution of sequencing errors is to the left. The two distributions overlap. **B**. Distributions of coinciding deviations due to errors and differences. The distribution of sequencing errors is to the left. A better separation between the distributions than in A is obtained.

The DNPs found in the analysis stage are later used in the contig construction stage. In addition to the usual criteria for adding a fragment into the contig - a reasonable pair-wise score with a fragment already in the assembly - two more requirements are imposed on the fragment to be added: 1. If there are DNPs present in the region around the position where it is to be placed, the read must contain at least two DNPs (Figure 7). 2. The read cannot mismatch at a DNP.

The following paragraph will describe the DNP assignment algorithm in somewhat more detail. Two different methods have been tested, the basic and the extended method. The basic method is fairly straightforward. In this method, a threshold D_{min} is set, which is the minimum number of coinciding deviations, n_{cd}, from the consensus that must be present on two columns simultaneously in order for those columns to be marked as containing DNPs (Figure 9). The deviations must be of the same base type within a column. In the extended method, the probability of observing the coinciding deviations by chance is calculated, and if the computed probability is lower than a threshold, $p_{\max}^{tot}$, the deviating bases are accepted as DNPs. The derivation of this probability measure uses base quality values in an approximation of the expected number of coinciding deviations; the interested reader is referred to [19] for a thorough explanation.

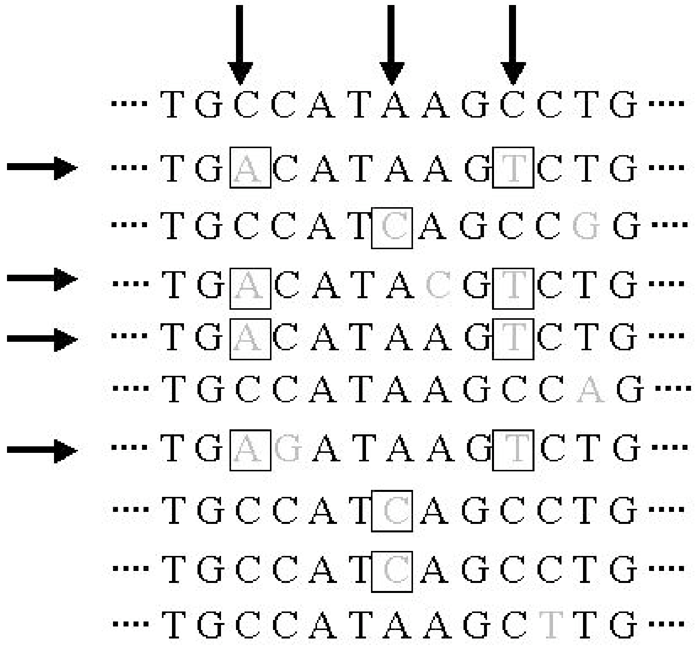

···· T G C C A T A A G C C T G ····
···· T G A C A T A A G T C T G ····
···· T G C C A T C A G C C G G ····
···· T G A C A T A C G T C T G ····
···· T G A C A T A A G T C T G ····
···· T G C C A T A A G C C A G ····
···· T G A G A T A A G T C T G ····
···· T G C C A T C A G C C T G ····
···· T G C C A T C A G C C T G ····
···· T G C C A T A A G C T T G ····

Figure 9. Vertical arrows indicate columns that are candidates for $D_{min} = 3$. Column 1 and 3 have $n_{cd} \geq D_{min}$, and the reads marked with horizontal arrows will thus have DNPs assigned.

Methods

Implementation

In order to test the performance of TRAP, two simulation programs were developed. One of them, *gen_seq*, generates a target DNA sequence with an inserted random sequence repeated in tandem. The user can define parameters regarding repeat length, number of repeat copies, number of differences between any two repeat copies etc. The output is a file containing the sequence in FASTA format [20] and a log file describing the different properties of the sequence, such as the location of differing bases between the repeat copies. The *gen_seq* program was developed using perl [21].

The second simulation program, *sim_gun*, is a program that simulates the process of shotgun sequencing. It is loosely based on ideas presented in [22], with a few added features. The program takes the target sequence from *gen_seq*, the log file from *gen_seq*, and (optionally) a file containing sequence quality values as generated by Phred as input. This last feature was added in order to get more realistic simulations, as opposed to other shotgun simulation programs that use a flat error rate or the same error curve for all fragments. A number of other parameters can also be specified by the user, such as desired shotgun coverage, the rate of sequencing errors consisting of insertions and deletions instead of substitutions, and the rate of errors yielding undeterminable bases (an 'n' instead of 'a', 't', 'g', or 'c'). The program samples random locations in the target sequence provided by *gen_seq* until the desired mean coverage is obtained, producing a sequence file and a quality file in FASTA format as output. The *sim_gun* program was written in C++ [23].

A number of functions were added to TRAP for bookkeeping purposes. These functions generated statistics files that could later be parsed in order to conclude the results and generate data files for plotting and visualization of the input data.

Simulated sets

A total of 648 simulations were performed. These were divided into six sets, including one control set. The sets were constructed in the following fashion. Nine different real project quality files were chosen. For each such project quality file, projects consisting of repeat units of length 1000, 2000, and 3000 bases, each repeated 4, 6, 8 and 10 times in tandem, were constructed, resulting in 108 simulated target sequences per set. The sets differed from each other in terms of quality clipping, i.e. the minimum quality requirement on a fragment, and simulated shotgun coverage. The properties of the different sets are listed in Table 1.

Simulation	*Avg. err after trimming*	*Max error allowed in trimming*	*Coverage after trimming*	*Average read length*
Sim 1	4.3	11	8.7	494
Sim 2	3.3	8	7.6	472
Sim 3	2.6	6	6.8	459
Sim 4	2.6	6	10.2	457
Sim 5	2.6	6	3.5	463
Control	4.3	11	8.7	494

Table 1. Properties of the different simulation sets.

In all simulation sets except the control set, the difference between any two repeat copies was 1%. If the requirement of two DNPs along a sequence read described above is to be met, and the average length of a sequence read is 500 bases, the repeats must differ at least 0.4% to be theoretically separable. However, for DNPs to be detectable at this low rate of difference, the shotgun sequencing coverage would have to be very high in order to obtain a reasonable probability of finding reads that span two differences. The difference of 1% was chosen in order to test the DNP method near its limits, while still simulating shotgun sequencing at reasonable choices of coverage. The control set consisted of identical repeats.

Results and discussion

This investigation focused mainly on comparing the basic and extended method with respect to sensitivity, i.e. the amount of true differences detected in the sequence reads, and specificity, i.e. the amount of erroneous differences detected, at different choices of D_{min}. For the basic method, the effect of different D_{min} on the amount of detected differences in the target sequence was also examined. This analysis was not performed for the extended method, since the number of detected differences in the target sequence is directly dependent on the amount of differences detected in the reads. Thus, the results of this investigation should be readily applicable to the extended method. Furthermore, the effect of shotgun coverage was examined, as well as the effect of the repeat copy number in the target sequence. Finally, a brief investigation of the flexibility of the extended method was conducted, where the variation in specificity and sensitivity was recorded for different choices of $p_{\max}^{tot}$.

True DNPs detected

The results of the basic method regarding sensitivity, i.e. the amount of true positives detected, in simulations Sim 1, Sim 2 and Sim 3 are shown in Table 2. For convenience, the results of the basic method at $D_{min} = 2$ is set to 100% sensitivity and all other results, for the basic method as well as the extended method, are computed in relation to this. As expected, the sensitivity decreases with increasing D_{min}. The reason for this is the fluctuating coverage along the target sequence. The coverage at a specific base in the target sequence can be approximated by a Poisson distribution if the reads sample truly random locations in the target [24], which is often the case in real shotgun sequencing. In these simulations, generated by a program, the reads sample random locations within the limits of the random number generator of the computer. This results in an uneven coverage where some regions of the target will be sampled less frequently than the mean coverage, while other regions will be sampled more frequently. If, for instance, a region only has been sampled by three reads, no DNPs from that region will be detectable at $D_{min} = 4$ or higher.

Another expected observation is that the sensitivity decreases with increasing quality based trimming. In the trimming step, the sequence is scanned from left to right and right to left until a predefined number of bases with quality higher than a threshold have been located. If too few bases meeting the requirement are found, the read is discarded. A higher constraint on quality leads to shorter and fewer reads, which in turn means that fewer differences are sampled. Also, when differences are trimmed out of a read it can lead to a failure in the detection of other differences in the same read. If a read contains only two differences, and one is trimmed away, the other one will remain undetected since the DNP method requires two differences to be present.

	D_{min}				
	2	3	4	5	6
	S_R (%)	S_R (%)	S_R (%)	S_R (%)	S_R (%)
Sim 1	100	89	81	71	60
Sim 2	81	73	64	55	43
Sim 3	71	62	54	44	33

Table 2. Sensitivity results for the basic method. The sensitivity regarding true differences in reads, S_R, is measured for different D_{min}.

The same effects can be observed in the extended method at $p_{\max}^{tot}$ = 10^{-3} (Table 3). The sensitivity decreases with increasing D_{min} and trimming. Again, the sensitivities are computed in relation to the result of the basic method in Sim 1 for D_{min} = 2. The major difference between the basic and extended method can be observed at D_{min} = 2. The sensitivities are lower for the extended method under these conditions. This effect decreases for D_{min} = 3 and is not observed at all for higher values of D_{min}. The reason for the difference in sensitivity at D_{min} = 2, and the slight difference at D_{min} = 3, is that coinciding deviations from consensus due to real differences between repeats are sometimes undistinguishable from random observations at these levels of D_{min} with $p_{\max}^{tot}$ = 10^{-3}. In other words, the distribution of coincidences between real columns and random ones overlap at these positions (see Figure 8).

	D_{min}				
	2	3	4	5	6
	S_R (%)	S_R (%)	S_R (%)	S_R (%)	S_R (%)
Sim 1	91	87	81	71	60
Sim 2	76	72	64	55	43
Sim 3	67	62	54	44	33

Table 3. Sensitivity results for the extended method at $p_{\max}^{tot}$ = 10^{-3}. The sensitivity regarding true differences in reads, S_R, is measured for different D_{min}.

Erroneously detected DNPs

The error of the basic method, i.e. the rate of erroneously assigned DNPs, was studied for Sim1, 2 and 3. The results are shown in Table 4. The error decreases with increasing quality constraints and with increasing D_{min}. The first effect is explained by the fact that more stringent trimming reduces the number of sequencing errors present in the data set and thus the number of opportunities for random effects to occur. In a completely error free data set there would of course be no erroneous DNP assignments, except for cases outside of the model, i.e. errors caused by alignment errors. The major decrease in error can be observed in the interval D_{min} = 2 to D_{min} = 4; with further increasing of D_{min} the drop in error rates is less significant. A closer investigation of errors remaining at D_{min} > 3 suggested that the majority resulted from either alignment errors or cases where two bases in a read had been erroneously "sequenced" as differences from another repeat, making them undistinguishable from true DNPs.

	D_{min}				
	2	3	4	5	6
	ε (%)	ε (%)	ε (%)	ε (%)	ε (%)
Sim 1	59	4.3	0.55	0.42	0.36
Sim 2	40	2.6	0.37	0.28	0.26
Sim 3	25	0.71	0.27	0.21	0.20

Table 4. Error results for the basic method. The error, ε, is measured for different D_{min}.

The results of the extended method (Table 5) follow the same pattern, i.e. decreasing error with increasing D_{min} and trimming. The major difference between the methods can be observed for D_{min} = 2 and D_{min} = 3. At higher values of D_{min}, virtually no difference between the two methods can be observed. The explanation is again that the distributions of coinciding deviations due to chance and due to actual differences between repeats do not overlap for D_{min} > 3 (again, see Figure 8 above). This means that the errors detected at D_{min} > 3 using the extended method are the same as for the basic method: alignment errors and coinciding sequencing errors.

	D_{min}				
	2	3	4	5	6
	ε (%)	ε (%)	ε (%)	ε (%)	ε (%)
Sim 1	8.9	1.6	0.53	0.42	0.36
Sim 2	6.6	0.92	0.37	0.28	0.21
Sim 3	5.5	0.52	0.27	0.21	0.20

Table 5. Error results for the extended method at $p_{\max}^{tot}$ = 10^{-3}. The error, ε, is measured for different D_{min}.

Detected differences in the target sequence

The effect of D_{min} on the ratio of detected differences in the target sequence was investigated (Table 6). The percentage of detected unique positions decreases with increasing D_{min} and quality clipping. This is expected, since the sensitivities in reads decrease under the same conditions (Table 2 above). Figure 10 shows that there is in principle a one to one relationship between detected differences in the template and in reads.

	D_{min}				
	2	3	4	5	6
	S_T (%)	S_T (%)	S_T (%)	S_T (%)	S_T (%)
Sim 1	97	87	78	65	53

Sim 2	94	82	70	57	43
Sim 3	90	76	62	48	34

Table 6. Detected differences in the target sequence using the basic method. The sensitivity, S_T, is measured for different D_{min}.

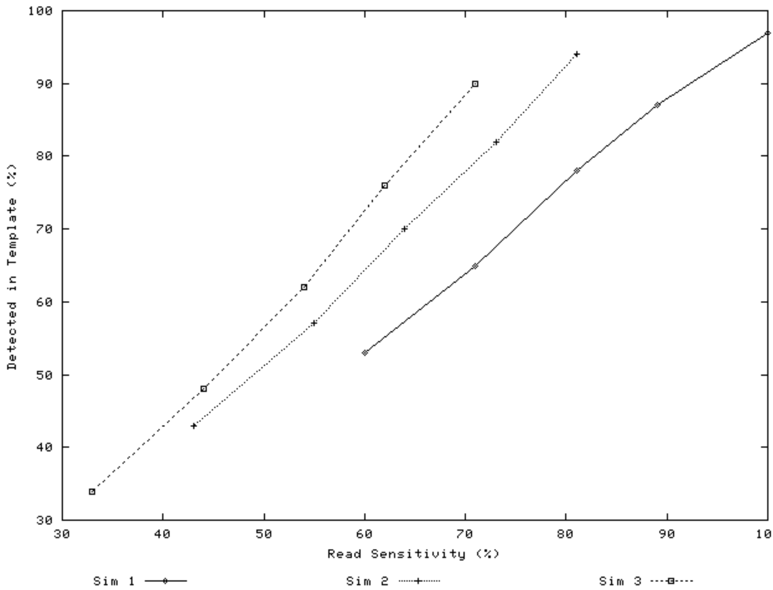


Figure 10. The relation between sensitivity in reads, S_R, and sensitivity in the target sequence, S_T, in Sim 1, 2 and 3 using the basic method.

Effect of shotgun coverage

The effect of shotgun coverage on error and sensitivity was studied under the same trimming conditions as Sim 3. Table 7 shows the results. Sim 4 has 50% higher shotgun coverage than Sim 3, and Sim 5 shows 50% lower coverage. These results show that an increase in shotgun coverage leads to an increase in sensitivity (see Table 2 above for comparison with Sim 3). Similarly, a decrease in coverage leads to a decrease in sensitivity. This is expected, since two adjacent DNPs are more likely to be spanned several times by sequence reads at higher coverage.

The effect on error rate is harder to interpret. In Sim 4, the error seems to increase for D_{min} = 2 and D_{min} = 3 compared to Sim 3, while the error for higher values of D_{min} is slightly lower. In Sim 5, with the lower coverage, the error seemingly decreases for D_{min} = 2 and D_{min} = 3 compared to Sim 3, and increases for higher values of D_{min}. Determining the cause of this effect will require further investigation.

	D_{min}				
	2	3	4	5	6
	ε / S_R (%)	ε / S_R (%)	ε / S_R (%)	ε / S_R (%)	ε / S_R (%)
Sim 4	30 / 95	1.3 / 89	0.21 / 83	0.17 / 78	0.14 / 69

Sim 5	18 / 64	0.69 / 41	0.47 / 23	0.47 / 21	0.77 / 11

Table 7. Effect of shotgun coverage using the basic method. Sensitivity in reads, S_R, and error, ε, is measured for different Dmin.

Effect of repeat copy number

To observe the effects of different repeat copy numbers in the target sequence on the error of the basic method, subsets of Sim 1, 2 and 3 were studied, containing 4 and 10 repeat copies in the target sequence respectively. The results (Table 8) are ambiguous. The subsets containing 10 repeat copies seem to follow the general trends observed above, i.e. decreasing error with increasing quality trimming and D_{min}. The subsets containing 4 repeats, however, seem to have an increase in error with increased trimming, contrary to all other results in this study. Whether this requires a modification of the model to handle repeats with few copy numbers or not remains to be investigated. The effect of increasing D_{min} in the 4 copy subsets follows the same trends as earlier with a decrease in error.

	D_{min}				
	2	3	4	5	6
	ε (%) 4/10	ε (%) 4/10	ε (%) 4/10	ε (%) 4/10	ε (%) 4/10
Sim 1	56 / 61	4.1 / 5.1	1.1 / 0.51	0.99 / 0.31	0.75 / 0.22
Sim 2	40 / 41	2.3 / 1.5	1.2 / 0.26	1.1 / 0.14	0.94 / 0.073
Sim 3	28 / 26	2.1 / 0.52	1.4 / 0.14	1.3 / 0.069	1.1 / 0.059

Table 8. Effect of repeat copy number using the basic method. The error, ε, is measured for 4 and 10 repeat copies in the target sequence, respectively.

Flexibility of the extended method

A brief investigation of the impact of different choices of $p_{\max}^{tot}$ was conducted. The experiment was set up as follows: D_{min} was set to 3, and the $p_{\max}^{tot}$ threshold was only used for observations of coinciding deviations $n_{cd} = 3$. All observations of $n_{cd} > 3$ were accepted as DNPs. In other words, the extended method was applied for $n_{cd} = 3$, otherwise the basic method was used. $p_{\max}^{tot}$ varied between 1 and 10^{-6}. The results are shown in Figure 11. At $p_{\max}^{tot} = 1$, all observations $n_{cd} \geq 3$ are accepted as DNPs which is effectively the same as running the basic method with $D_{min} = 3$. With a lowering of $p_{\max}^{tot}$, the sensitivity and error gradually decrease. However, the decrease in error starts before the decrease in sensitivity as can be seen in Figure 11. At $p_{\max}^{tot} = 10^{-3}$, the sensitivity has dropped 2% (from 89% to 87%), whereas the error has dropped 63% (from 4.3% to 1.6%). This shows that the extended method is a more flexible tool than the basic method, and that there is room for optimization of the extended method.

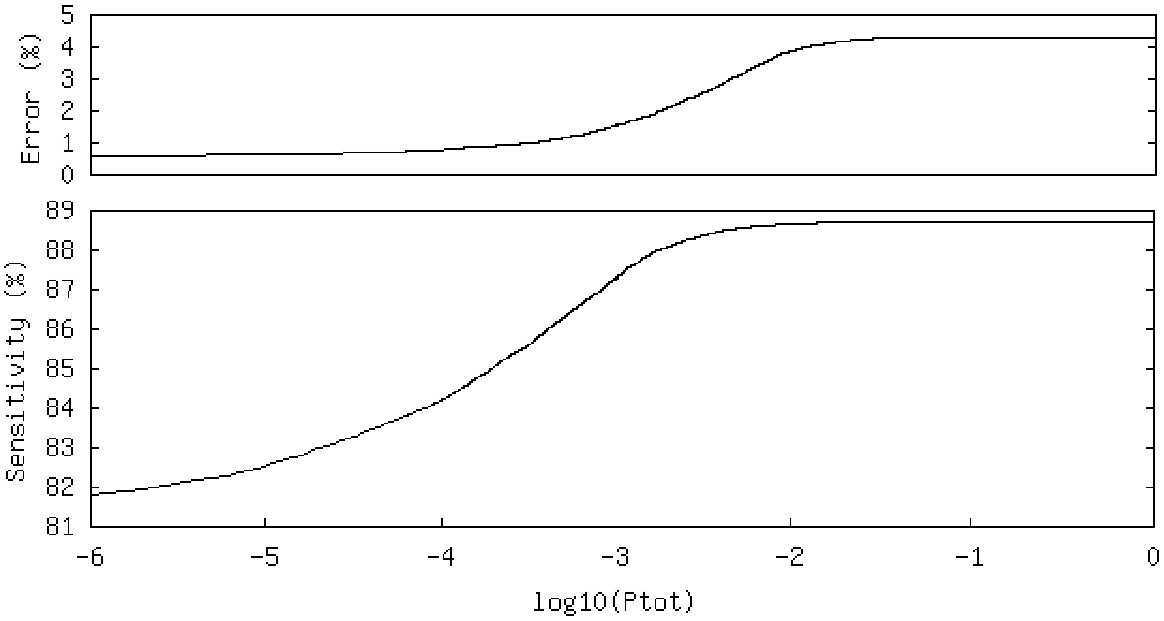


Figure 11. The variation in error (above) and sensitivity (below) for different $p_{\max}^{tot}$ at D_{min} = 3 in Sim 1.

The control set

The control set was identical to Sim 1, except that no different between repeat copies in the target sequence were present. Using the basic method, 58 298 DNPs were erroneously detected with D_{min} = 2. At D_{min} = 3, 693 errors remained. No DNPs were erroneously detected at higher levels of D_{min}. Using the extended method at D_{min} = 3 with $p_{\max}^{tot}$ = 10^{-3}, 162 errors were made. With D_{min} = 3 and $p_{\max}^{tot}$ = 10^{-5}, no DNPs were erroneously detected.

Conclusions

The sensitivity and specificity of the DNP method have been thoroughly investigated for different parameters in the model, as well as for different shotgun sequencing conditions. The results indicate that the DNP method is robust and flexible. The method is robust, since the results vary as expected with varying parameters and sequencing conditions. Almost all simulations yielded the expected results, except for some deviations in the experiments regarding shotgun coverage and repeat copy number. These areas will require further study, and perhaps minor modifications to the model will be necessary in order for the DNP method to behave in a predictable fashion under these conditions. Secondly, the introduction of the extended method makes the DNP method flexible. The basic method is a rather blunt instrument where there is a clear trade-off between sensitivity and specificity. The extended method is a means to achieve a better resolution and maintain high sensitivity without the increase in error seen with the basic method.

It is clear that the extended method has the greatest effect when sequence reads are trimmed less stringently. Under stringent trimming conditions, fewer sequencing errors are present in the data set. Multiple coincidences between columns due to chance are rare, and the errors that remain are mainly caused by alignment errors and cases where several bases in a read have simultaneously been erroneously sequenced as the unique elements of another repeat copy, making them undistinguishable from true DNPs. The distributions of coincidences due to chance and real differences do not overlap when the quality constraints are set highly enough. In contrast, the distributions will overlap under less stringent trimming conditions, which makes the extended method is more useful in these cases. For the same reasons, the extended method has the highest impact for low values of D_{min}. At higher values, the distributions do not overlap.

A possible strategy for a sequencing project would be to favor the basic method in combination with stringent trimming over using the extended method. This is a perfectly feasible strategy, but it has economical consequences that have to be considered. Using this strategy, a high specificity is obtained at the cost of a low sensitivity. The low sensitivity leads to more finishing, since fewer differences in the target are detected and can be utilized, resulting in a higher cost for completing the project. The decrease in sensitivity can be compensated for by an increase in shotgun coverage, which is also costly. Compare, for instance, the results of Sim 1 and Sim 4 (Tables 2 and 7). In order to achieve comparable results for the basic method, with $D_{min} = 3$, in Sim 4 compared to the extended method in Sim 1, the shotgun coverage has to be increased by as much as 50%.

To conclude, the DNP method will allow for faster finishing of complicated sequencing projects. There is also room for improvement of the model. In the current version, known parameters such as shotgun coverage and distance between DNP candidate columns are not used in the model. The effect of including these parameters in the model remains to be seen; it is clear that the DNP method in its current state is already a powerful tool for separating repeats in shotgun sequencing.

Acknowledgements

First I would like to thank my supervisor Björn Andersson for excellent support and valuable insights during the course of this project. A special thank you goes out to my friend, mentor and ”partner in crime” Martti Tammi, to whom I owe most of the knowledge gained during this time. I would also like to acknowledge Daniel Nilsson, who, aside of being a good friend has been of tremendous help in my process of learning about computers in general, and perl and Linux in particular. Thanks also to all the other nice people in the Uppsala Genome Sequencing Laboratory group for providing a nice and relaxed atmosphere to work in.

References

1. Lander, E.S. et al. Initial sequencing and analysis of the human genome. *Nature* **409**, 860-921. (2001).
2. Venter, J.C. et al. The sequence of the human genome. *Science* **5507**(291). 1304-51. (2001).
3. Needleman, S. & Wunsch, C. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* **48**, 443-453. (1970).
4. Smith, T. & Waterman, M. Identification of common molecular subsequences. *Journal of Molecular Biology* **147**. 195-197. (1981).
5. Gusfield, D. *Algorithms on strings, trees and sequences: computer science and molecular biology*. (Press Syndicate of the University of Cambridge, 1997).
6. Churchill, G. & Waterman, M. The accuracy of DNA sequences: estimating sequence quality. *Genomics* **14**, 89-98. (1992).
7. Giddings, M.C., Brumley, R.L., Haker, M., & Smith, L.M. An adaptive, object oriented strategy for base calling in DNA sequence analysis. *Nucleic Acids Research* **19**. 4530-40. (1993).
8. Lawrence, C. & Solovyev, V. Assignment of position-specific error probability to primary DNA sequence data. *Nucleic Acid Research* **22**, 1272-1280. (1994).
9. Lipshutz, R., Taverner, F., Hennessy, K., Garzell, G. & Davis, R. DNA sequence confidence estimation. *Genomics* **19**, 417-424. (1994).
10. Ewing, B., Hiller, L., Wendl, M. & Green, P. Base-calling of automated sequencer traces using Phred. I. Accuracy assessment. *Genome Research* **8**, 175-185. (1998).
11. Ewing, B. & Green, P. Base-calling of automated sequencer traces using Phred. II. Error probabilities. *Genome Research* **8**, 186-194. (1998).
12. Green, P. http://www.phrap.org. (1996)
13. Huang, X. & Madan, A. CAP3: a DNA sequence assembly program. *Genome Research* **9**, 868-877. (1999).
14. Batzoglou, S. et al. ARACHNE: a whole-genome shotgun assembler. *Genome Research* **12**, 177-189. (2002).
15. Staden, R. A new computer method for the storage and manipulation of DNA gel reading data. *Nucleic Acids Research* **8**, 3673-94. (1980).
16. Sutton, G.G., White, O., Adams, M.D. & Kerlavage, A.R. TIGR assembler: a new tool for assembling large shotgun sequencing projects. *Genome Science & Technology* **1**, 9-19.
17. Myers, E.W. et al. A whole-genome assembly of Drosophila. *Science* **287**, 2196-2204. (2000).
18. Anson, E.L. & Myers, E.W. ReAligner: a program for refining DNA sequence multi-alignments. *Journal of Computational Biology* **23**, 262-272. (1997).
19. Tammi, M.T., Arner, E., Britton, T. & Andersson, B. Separation of nearly identical repeats in shotgun assemblies using defined nucleotide positions, DNPs. *Bioinformatics* **3**(18), 379-388. (2002).

20. Pearson, W. & Lipman, D. Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci.* **85**, 2444-8. (1988).
21. Wall, D. *Programming Perl, Second Edition*. (O'Reilly, 1996).
22. Engle, M.L. & Burks, C. GenFrag 2.1: new features for more robust fragment assembly benchmarks. *Comput. Appl. Biosci.* **10**, 567-8. (1994).
23. Stroustrup, B. *The C++ Programming Language, Third Edition*. (Addison Wesley, 1997).
24. Lander, E. & Waterman, M. Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics* **2**, 231-9. (1988).