ABSTRACT

LOCATION OF FRAMESHIFT ERRORS VIA COMPARISON WITH CLOSELY RELATED GENOMES

Stephen Snow, M.S.
Department of Computer Science
Northern Illinois University, 2009
Reva Freedman, Director

Frameshift errors represent a significant problem in sequenced genomes. Pathfinder is a software tool written to locate these frameshift errors and facilitate their correction as accurately as possible by comparing an erroneous gene to a closely related and better sequenced template gene. Pathfinder is found to significantly improve poorly sequenced genomes while introducing only a minuscule amount of new error.

NORTHERN ILLINOIS UNIVERSITY
DEKALB, ILLINOIS

May 2009

LOCATION OF FRAMESHIFT ERRORS

VIA COMPARISON WITH CLOSELY

RELATED GENOMES

BY

STEPHEN SNOW

A THESIS SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE

MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

Thesis Director:
Reva Freedman

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

1. Introduction

In the following thesis, we will describe Pathfinder, software used for the purpose of locating and aiding in the correction of frameshift errors in already sequenced genomes. Various technologies are used to sequence genomes, or identify the sequence of base pairs that make up the DNA of an organism. Then, gene-calling software is used to identify and locate genes within the sequenced genome. If the original sequence is incorrect, genes will be incorrectly called, interfering with the ability of biologists to use the data for further research.

Frameshift errors, where one or more characters are added or deleted to a proposed DNA sequence, are one of the most serious types of errors in sequenced genomes. Frameshift errors can lead the gene calling software to report that there are two genes where there should only be one or cause the omission of a gene altogether. In particular, a massively parallel pyrosequencing system developed by 454 Life Sciences, known as 454 sequencing (Ronaghi et al., 1998), is known to cause a large number of frameshift errors; 454 sequencing is, however, becoming more and more prevalent due to its low cost and quick sequencing time (Margulies et al. 2005).

Numerous methods have been used to attempt correction of these frameshift errors during the gene's sequencing (Miller et al., 2008; Quinlan et al., 2008; Vacic et al., 2008). The 454 sequencing machines themselves come with built-in error correction software (Margulies et al., 2005). Pathfinder differentiates itself from these methods by locating and attempting correction of the genes after they have already been sequenced, not during the sequencing process. Pathfinder also relies heavily on data from very closely related genomes, not only the data from the genome being corrected.

Overall System

Pathfinder is one module of a larger system, seen in Figure 1. The end goal of the complete system is to repair frameshift errors and produce a higher quality genome. Pathfinder reads its data from the SEED database (Overbeek et al., 2005). It then attempts to locate the frameshift errors as described in this document. Finally, a corrector program takes the information generated by Pathfinder and attempts to correct the errors Pathfinder has found.

Pathfinder uses the SEED database developed by Ross Overbeek and FIG, the Fellowship for the Interpretation of Genomes (Overbeek et al., 2005), to compare closely related genes. The SEED contains a vast amount of up-to-date information about several hundred prokaryotic genomes and the genes contained within them, as well as an annotation system known as the RAST server. The SEED also keeps track of a gene's “SIMs,” similar genes in closely related organisms. Due to the fact that much DNA is conserved during evolution, these genes frequently are homologous (have a common ancestor) (Lesk, 2005). Pathfinder operates on the assumption that the DNA between these genes is well conserved enough for us to determine a region in which a frameshift error occurred during reading.

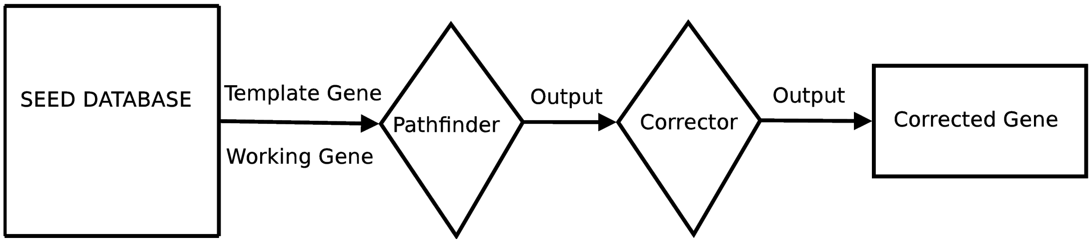

Figure 1: Pathfinder in Software Context

2. Background

The sequence of bases in DNA is represented as a long character string of A's, C's, G's and T's. Every three bases codes for one amino acid. These triplets are known as codons. Assuming that you are reading forward in the DNA string, there are three reading frames (Lewin, 2000), depending on where the translation starts with respect to a multiple of three. Each reading frame will be translated into a different amino acid string, of which probably only one is correct. Figure 2 below illustrates the three forward reading frames of a single stretch of DNA.

```
   Code: ATGGGACGTTATCCT
Frame 0:     ATG GGA CGT TAT CCT

Frame 1:   A TGG GAC GTT ATC CTX

Frame 2: AT GGG ACG TTA TCC TXX
```

Figure 2: Differing Reading Frames

Sequencing Methodology

Modern sequencing technologies can only sequence short strips of DNA at one time. In order to sequence large genomes, “shotgun sequencing” is used. In this method, the larger strip of DNA is broken up into many much smaller and overlapping pieces. Each of these pieces is then individually sequenced. These pieces are then reassembled into large contiguous sequences (henceforth referred to as contigs) (Staden, 1979).

The DNA is broken up in different ways numerous times. This is done in order to facilitate reassembly. The number of times the DNA of a particular genome has been broken and sequenced is known as its coverage. Usually, the higher the coverage, the higher quality the genome.

Often this process will introduce gaps into the DNA if a particular piece is misplaced, lost or otherwise erroneous. In this event, a single sequenced genome may be represented by many contigs in the database. Generally, the fewer contigs a gene has, the higher quality it is. A gene consisting of only one contig has been completely reassembled into one long contiguous sequence.

Frameshift Errors

A frameshift error occurs when the machine reading these bases from a strand of DNA inserts or deletes any number of bases not divisible by three. A single inserted or deleted base pair can throw off an entire sequence by shifting a portion of it into another reading frame. Figure 3 below illustrates a DNA sequence and the effect a single erroneous base pair will have on the translation.

DNA sequence before frameshift:

DNA Sequence:	ATG	GGA	CGT	TAT	CCT
	⇩	⇩	⇩	⇩	⇩
Amino Acid Sequence:	M	A	A	T	P

DNA sequence after frameshift (erroneous base pair is bold);

Erroneous DNA Sequence:	ATG	G**A**G	ACG	TTA	TCC	T
	⇩	⇩	⇩	⇩	⇩	
Amino Acid Sequence:	M	G	T	L	P	

Figure 3: Example of Frameshift Error

Scope

Pathfinder was designed with a specific data set in mind. To begin with, the gene must be in the SEED. It must also be in a genome that has a closely related genome (a genome in the same species) in the database. Furthermore it must also have a closely related SIM gene to which we may compare it. Ideally the SIM will be in a high-quality genome. Not all genomes have been sequenced equally well. Some genomes are sequenced using very expensive and time-consuming methods. These genes are much less error-prone and therefore much better to compare to than a more error-prone gene. In particular, genomes sequenced with 454 technology are more likely to contain errors (Gharizadeh et al., 2006; Goldberg et al., 2006).

3. Methodology

Pathfinder is invoked from the linux command line. The user inputs a desired gene number (or a range of genes). Pathfinder includes a myriad of command line options that display different levels of information throughout the process. The user also must specify a high quality template genome with which to compare the selected genes. Pathfinder processes each selected gene individually. It begins by selecting the most similar SIM gene within the template genome to the gene being worked on. If no such SIM gene exists, a warning is displayed and Pathfinder moves on to the next gene.

In general, frameshift errors will often cause a single gene to split into numerous small genes. In order to compensate for this problem, the length of the DNA of the working gene is extended to match that of the chosen SIM gene. The DNA of the working gene is then extended on either side by one hundred bases to ensure it is captured in its entirety.

The DNA of the working gene in each reading frame is translated into amino acids, and is then aligned with the SIM gene using BLAST (Altschul et al., 1990; Altschul et al. 1997). BLAST, or the Basic Local Alignment Search Tool, is an algorithm for comparing two DNA strings at either the base pair or amino acid level. Specifically, Pathfinder uses the version of BLAST known as BLASTP to align the amino acid sequences. By analyzing the amino acid sequences rather than the DNA directly, a great deal of noise is removed and the overall of efficiency of the program is increased. BLAST is one of the most commonly used and accepted alignment tools in the bioinformatics community.

BLAST returns a list of matched alignments. Each match identifies a position in the working gene, a related position in the SIM gene, the offset between them, and the reading frame in which the match occurred. If the relative positions between the working gene and the SIM gene are far apart, the match is likely very poor and spurious. To ensure only best quality matches are used, any match that has a significantly larger offset than the best match is thrown out. Matches with widely differing offsets are unlikely to be related to one another.

Once the list of good matches is acquired, they are lined up with the amino acid sequence of the SIM in an object known as the “framework”. The framework contains each match considered good, as well as information pertaining to the gene as a whole. Conceptually, the framework is the environment in which Pathfinder operates. Pathfinder is “inserted” into the framework and works its way through to the other end. Figure 4 below is a short sample of a framework. Each match is listed and numbered. The number in parenthesis beside the match number indicates the frame of that particular match. At each position, the relative position in the SIM is recorded, as well as the SIM's amino acid at that position. Also at each position, three pieces of information are displayed for each match: position in the original, amino acid at that position, and PAM10 score of that particular amino acid.

```
POS:SIM   Match 0 (1)     Match 1 (0)     Match 2 (2)
******************************************************
  0: 0   ( - , - ,-24)   ( - , - ,-24)   ( - , - ,-24)
  1: M   ( - , - ,-24)   ( 48, M , 12)   ( - , - ,-24)
  2: Q   ( - , - ,-24)   ( 49, Q ,  9)   ( - , - ,-24)
  3: K   ( - , - ,-24)   ( 50, K ,  7)   ( - , - ,-24)
  4: I   ( - , - ,-24)   ( 51, I ,  9)   ( - , - ,-24)
  5: D   ( - , - ,-24)   ( 52, D ,  8)   ( - , - ,-24)
  6: Y   ( - , - ,-24)   ( 53, Y , 10)   ( - , - ,-24)
  7: S   ( - , - ,-24)   ( 54, S ,  7)   ( - , - ,-24)
  8: T   ( - , - ,-24)   ( 55, T ,  8)   ( - , - ,-24)
  9: R   ( - , - ,-24)   ( 56, R ,  9)   ( - , - ,-24)
 10: K   ( 57, K ,  7)   ( 57, Q , -6)   ( - , - ,-24)
 11: K   ( 58, K ,  7)   ( 58, K ,  7)   ( - , - ,-24)
 12: I   ( 59, I ,  9)   ( 59, - ,-24)   ( - , - ,-24)
 13: D   ( 60, D ,  8)   ( 59, - ,-24)   ( - , - ,-24)
 14: L   ( 61, L ,  7)   ( 59, - ,-24)   ( - , - ,-24)
```

Figure 4: Sample of an Example Framework

Once inserted into a framework, Pathfinder attempts to identify the proper start position of the working gene. All genes begin with the DNA codon ATG, GTG or TTG. If the first matched position is not one of these start codons, Pathfinder looks in the region it is expected to be in based on where the SIM's start is. If no start is located there, Pathfinder steps backwards from the first matched position looking for any possible start codon within the same frame. If a start is found, Pathfinder updates the working gene's start location and proceeds from there. If no start is found, or a stop codon is found first, Pathfinder warns the user and proceeds on from the first matched position.

Pathfinder begins on whichever match the start was found in (if no start was found, Pathfinder begins on whichever match contained the first matched position). Pathfinder steps position by position, comparing the amino acid in the match to the amino acid in the SIM at that same position. If the two amino acids are not the same, a mismatch is identified. A mismatch may be evidence of a frameshift, but can also be a natural difference between the two genes. The mismatch is scored using a PAM matrix (Dayhoff et al., 1978), a matrix that gives the likelihood of one amino acid naturally mutating to another. The higher the score, the more likely the mutation (Lesk, 2005). For example, a mismatch of amino acids D and E will return a score of 0, while a mismatch of D and C will return a score of -21, which means that D is more likely to become E than C.

If, at some point, a match's PAM score becomes too low, Pathfinder analyses the next five positions in every available match. Each match is compared to the SIM gene, giving more weight to the positions closest to mismatch, and scored using the PAM10 matrix. If another match is found to have a higher score than the one Pathfinder is currently on, Pathfinder will switch matches. If Pathfinder does not switch matches, it marks the position as a mismatch and moves on. A mismatch may be useful information to a biologist studying the program's results, and may be related to a frameshift or another anomaly located near it.

If Pathfinder switches to a new match, this is very strong evidence of a frameshift error. By looking at the differences between frame and position caused by the jump, Pathfinder is able to come up with a suggested course of action needed to repair the gene. If Pathfinder chose to switch matches, but the frame of the matches does not change, this could be indicative of a more serious problem in the region, or numerous frameshift errors too close to one another for Pathfinder to reliably detect. In situation where Pathfinder is unable to determine the nature of the frameshift, it will list the problem area in the list of frameshifts and label it as "unknown."

Occasionally, due to very poor genomes or poorly matched SIMs, Pathfinder will come across large gaps in which there was no match between the working gene and the SIM gene. In these cases, Pathfinder locates the end of the gap, and uses the same position difference versus frame difference to determine a possible cause for the gap. In the output, gaps are listed differently than frameshifts.

Once Pathfinder has stepped through the matched gene in its entirety, it attempts to identify a stop. All genes must end with the DNA codons TAA, TAG, and TGA. If the gene does not end with a stop, Pathfinder employs the exact same method it used to attempt to identify a start.

In Figure 5 we see the path that Pathfinder chose through the example framework shown in Figure 4. The square brackets denote the match Pathfinder chose at that particular position, while the parenthesis represent a match that Pathfinder did not select. In this particular case, Pathfinder began on match number one, and reached a mismatch at SIM position 10 (orig position 57). At this point, Pathfinder looked at all available matches, and determined that match zero had the best score at that position.

```
POS:SIM   Match 0 (1)     Match 1 (0)     Match 2 (2)
********************************************************
  0: 0   ( - , - ,-24)   [ - , - ,-24]   ( - , - ,-24)
  1: M   ( - , - ,-24)   [ 48, M , 12]   ( - , - ,-24)
  2: Q   ( - , - ,-24)   [ 49, Q ,  9]   ( - , - ,-24)
  3: K   ( - , - ,-24)   [ 50, K ,  7]   ( - , - ,-24)
  4: I   ( - , - ,-24)   [ 51, I ,  9]   ( - , - ,-24)
  5: D   ( - , - ,-24)   [ 52, D ,  8]   ( - , - ,-24)
  6: Y   ( - , - ,-24)   [ 53, Y , 10]   ( - , - ,-24)
  7: S   ( - , - ,-24)   [ 54, S ,  7]   ( - , - ,-24)
  8: T   ( - , - ,-24)   [ 55, T ,  8]   ( - , - ,-24)
  9: R   ( - , - ,-24)   [ 56, R ,  9]   ( - , - ,-24)
 10: K   [ 57, K ,  7]   ( 57, Q , -6)   ( - , - ,-24)
 11: K   [ 58, K ,  7]   ( 58, K ,  7)   ( - , - ,-24)
 12: I   [ 59, I ,  9]   ( 59, - ,-24)   ( - , - ,-24)
 13: D   [ 60, D ,  8]   ( 59, - ,-24)   ( - , - ,-24)
```

Figure 5: Example Path Through Framework

Figure 6 is an example of Pathfinder's output. It begins with the identifier of the particular gene in question (the working gene). That is followed by the contig number and base pair location the gene had prior to correction. Then, the gene's new contig number (which typically will not change) as well as its newly corrected location on that contig are listed.

Each frameshift found in listed with an FS in the first column. The following columns have the corresponding meanings: Frame before the frameshift, last matched position in the working gene before the frameshift, next matched position in the working gene after the frameshift, frame after frameshift, last matched position in the template before the frameshift, the next matched position in the template after the frameshift, and finally the recommended course of action to repair the frameshift.

Each mismatch detected is also listed with an MS in the first column. The remaining columns have the following meanings: position in the working gene, position in the template gene, amino acid in original gene, amino acid in the template gene.

```
fig|393125.3.peg.402
Orig_Position: 393125.3_fragment_17    81576     80825
Correct_Position: 393125.3_fragment_17 81477     81031
sim: fig|265669.1.peg.645
FS   0    52   1    54   19   21   delete 1
FS   1    146  0    149  113  115  delete 2
MS   150  116  Y    I
//
```

Figure 6: Pathfinder Output, Identifying Two Frame Shifts

Frameshift Correction

Once Pathfinder has generated a list of frameshift errors, that list is passed off to the correction program. The corrector takes a chunk of DNA from around the frameshift error. The corrector then attempts to match that DNA to the template genome exactly by attempting every simple insertion and deletion scenario. In the event that this method fails, the corrector will attempt to find a suitable match through substitution. For each substitution required for the match, a subtraction from an overall score is made. The change with the highest score is taken. If no change achieves a score of 80% or better, a third and final approach is attempted.

The DNA is translated into amino acids, and is scored using the PAM10 matrix. If no change receives a score better than 0 from the matrix, the correction process has failed.

The RAST annotation software is much more tolerant of a complete gene with a few incorrect amino acids than it is of truncated gene. As such, the corrector will attempt to fill in any significant gaps encountered. If a reading frame is not the same on either side of the gap (and thus a frame change is needed somewhere within the gap), the corrector will simply insert a C or CC to achieve the appropriate reading frame. These bases were chosen for being the least likely to generate a new stop codon. If no reading frame change is required within the gap, the corrector simply uses the original gene sequence from that region.

Finalization

Once correction has been completed, each corrected gene is tested for internal stop codons. An internal stop codon would prematurely terminate a gene, thereby truncating it. Any genes found to have an internal stop codon are removed, and the original uncorrected gene is used in its place. Any duplicate genes generated by merging two genes into one are removed, and any genes which were found to have no template match are included unaltered.

Implementation

Pathfinder is written in Perl version 5.8.7. It is implemented as three classes and is invoked from the command line using a driver script. Pathfinder relies on data access to the SEED database system. Pathfinder is freely available to the public. For further information, please refer to http://microbe.cs.niu.edu/biowiki.

4. Results

As Pathfinder itself is an early module in a larger system, we will henceforth speak of the system as a whole's performance and results. The overall system will be referred to as “Pathfinder” for simplicity. Pathfinder was tested on a series of sixteen Listeria genomes. Two of these genomes are very high quality genomes, and were chosen as the templates. The remaining fourteen genomes were sequenced using 454 sequencing and were very low quality. These were the target genomes Pathfinder attempted to improve.

Genome Quality

Frameshifts often cause gene calling software to find two genes where there should only be one. This causes the total number of genes in a heavily frameshifted genome to be inflated. This also causes a large number of genes in a heavily frameshifted genome to be short when compared to their full-sized counterparts in the template genome. As such, the number of total genes in relation to the template, as well as the percentage of genes that are of complete length when compared to their counterpart in the template genome are two measurements for the quality of a genome.

Between 5 and 15% of the genes in each genome had no match in the template genome. This may have been caused by a number of different factors. These genes may represent genetic divergence between the two species, or they may simply be anomalies created by the gene calling software. These "no match" genes are not correctable by Pathfinder and still inflate the number of genes in each genome.

There also exist some genes which have a match in the template, but are radically divergent from their template counterpart. To compensate for these genes, we report "good genes" as those that are within 5% of the length of their template, as well as maintaining at least 90% amino acid identity.

Overall Results

Of the genes tested, group 1 was of the poorest quality. This can be seen in that they have a large number of contigs, low coverage, and a large number of genes (most of which are "bad") relative to the good template genomes. It was among this lower quality group that Pathfinder produced the greatest amount of corrections, making an average of 2499 inserts and deletes per genome. The percentage of good genes increased an average of 37.5%.

The genomes of group 2 were of significantly higher quality than those of group 1. The percentage of good genes was increased an average of 5.8% while the average number of insertions and deletions was 229.

Template Comparison

To gain a deeper insight, both template genomes were compared to one another. As can be seen in Table 1, very few changes were made between the two template genomes when 265699 was used at the template. This is consistent with the fact that they are both very high quality genomes in very related species. Another indicator of this is that they have very few "no match genes."

Using gene number 265699 as a template, 17 genes were corrected. When 169963 was used as the template, 13 genes were corrected. These 30 genes were studied thoroughly. One of the corrected was found to contain an intentionally programmed frameshift error (Craigen and Caskey, 1986). However, in most cases the corrected genes were so similar to the template genomes that they may represent natural mutations. They may also be dideoxy sequencing errors. In seven cases, genes were incorrectly truncated to match the template gene. These represent error introduced by Pathfinder.

454 Sequenced Genome

Changes to the most improved genome, gene number 393125 were also studied in depth. Initially this genome contained 5030 genes of which only 36.7%

Table 1

Summary of Genome Data Before and After Pathfinder Correction

					before correction:		after correction:				
RAST genome number	strain	approx. 454 coverage	contigs	bp	num genes	% good	num genes	% good	no match genes	inserts/ deletes	% substitu-tions
265669	L.m. 4b F2365	dideoxy	1	2905310	2855	--	--	--	--	--	--
169963	L.m. EGD-e	dideoxy	1	2944528	2865	90.3	2860	90.5	185	16	5.03
Group 1											
393125	L.m. FSL R2-503	8x	54	3001696	5030	36.7	3148	90.8	457	3848	0.58
393122	L.m. FSL J2-064	9x	327	2899431	3820	59.8	3100	86.4	275	1391	0.44
393120	L.m. FSL J2-003	8x	406	2878206	4584	36.2	3230	76.9	429	3016	4.87
393118	L.m. J1 175	9x	357	2902346	4475	41.7	3199	84.6	340	2926	0.31
393132	L.m. LO28	9x	529	2910810	4991	30.7	3457	73.9	552	3570	5.02
393117	L.m. FSL J1-194	10x	44	2986227	3692	66.9	3034	94.0	322	1182	0.42
393123	L.m. FSL N1-017	10x	46	2857865	3704	54.1	2895	82.3	289	1560	4.62
Group 2											
401650	L.m. Aureli 1997	20x	75	3006068	3219	88.5	3058	95.6	318	268	0.33
393121	L.m. FSL J2-071	21x	77	3149923	3370	87.1	3216	94.1	484	251	0.42
393124	L.m. FSL N3-165	22x	33	2886689	2881	87.4	2840	89.0	166	74	4.97
393131	L.m. J2818	24x	38	2971223	3176	81.5	3015	88.6	329	289	4.94
393128	L.m. F6900	26x	35	2958319	3262	76.5	2969	88.6	304	494	4.95
393130	L.m. J0161	29x	51	3051828	3179	84.6	3072	89.2	388	174	4.93
393133	L.m. 10403S	48x	32	2866709	2871	87.3	2845	88.6	190	57	5.01

were considered good by the above stated criteria. When corrected using gene 265669 as the template, the total gene number is reduced to 3148, with 90.8% of those genes considered good. A total of 1691 genes had frameshift errors corrected. 457 genes from 393125 had no match in 265669. These genes cannot be corrected, but are very likely frameshifted. Therefore, they inflate the total number of genes in this final genome. Taking this into account, the correction results bring the total gene number very close to the template genome's gene count. This represents a significant improvement to this genome.

Introduced Stop Codons

Pathfinder does introduce some new error into genomes, the most serious of which is the introduction of a new stop codon within a gene. Several genes are found to have a stop codon in the middle after correction. These genes represent an error made by Pathfinder, as the in-frame stop codon would effectively terminate the gene. Any gene found with a stop codon somewhere within the DNA is rejected in the final quality check, and the original uncorrected version is used.

Focusing on gene number 393125, nineteen genes with an internal stop codon were found after correction. Nine of these erroneous stop codons were found in areas that BLASTP was unable to find any match, causing a gap in the DNA . These gaps likely contained numerous frameshifts that Pathfinder was unable to locate. Another seven cases occurred when two frameshifts (an insert and a delete) occurred within a very short time of one another, effectively canceling out reading frame changes. The remaining three stop codons were caused by isolated base changes, and may be naturally occurring. An improved gap handling algorithm would cut down on these introduced errors, and would be a good focus for future work.

Gene Functional Assignments

The RAST system attempts to give functional assignments to each gene. The quality of a gene plays an important role in this. If a gene is badly damaged and heavily frameshifted, the system may not be able to assign the gene to a functional role. Of the 1691 genes corrected in genome 393125, 1018 genes were placed given a role that at least one of their counterparts had previously been assigned to. Sixty-eight genes were not assigned any role before or after correction. 532 genes were assigned a functional role while none of their counterparts had previously been given a role. These 532 genes represent an overall improvement of the genome as a whole.

However, 48 genes that had a previously assigned functional role were assigned a different role after correction. In all but 8 of these cases, the new assignments were simply more specific roles. A further 24 genes effectively lost their functional roles after correction. These 24 genes, as well as the eight mentioned previously are potential errors that inhibit the functional assignment process. They represent 1.9% of the total genes corrected.

Comparison to RAST's Frameshift Correction

In processing gene number 393125, the RAST system changed 220 genes, making about 444 total changes. This is about 1/8 as many changes as Pathfinder made to the same genome. Of these changes, Pathfinder also modified all but 13 of these genes. All but one of those 13 genes had no match in either template, and therefore was uncorrectable by Pathfinder.

Of the changes made, Pathfinder and RAST often made different changes that resulted in the same reading frame change. Table 2 shows some of these different corrections. As can be seen, Pathfinder often matches the template genome much more closely than the RAST system's corrections. If it is assumed that the gene should match the template sequence as closely as possible, Pathfinder achieves a much better correction than the RAST system.

Table 2

Amino Acid Sequences of the Region Surrounding Frameshift Corrections

Suggested by RAST and Pathfinder

Gene number	Pathfinder change	RAST change	Pathfinder corrected sequence	Template sequence	RAST corrected sequence
263	delete 2	insert 1	N	N	KQ
400	insert 1	delete 2	IV	IV	L
434	delete 2	insert 1	QN	QN	TKI
565	delete 2	insert 1	GISAIM	GISAIM	WLFQLLW
900	delete 2	insert 1	FFA	FFA	IFCS
1107	delete 2	insert 1	IFW	IFW	YFFG
1565	delete 2	insert 1	G	G	LF
1633	delete 2	insert 1	QKKR	QKKR	TKKTL
1762	delete 2	insert 1	IKKK	IKKE	NKKRK

5. Discussion

The Pathfinder system successfully improves poorly sequenced genomes which have a well-sequenced analogous genome while introducing a very small amount of new errors. Through Pathfinder's correction of these frameshfit errors, the number of genes in a particular genome can be reduced to a reasonable number. These corrections facilitate the assignment of functions to these genes, which is a major goal of genomics.

Pathfinder's scope is limited to those genes which have closely related SIM genes in a well sequenced related genome. However, within that scope, analysis finds that Pathfinder is not only an improvement upon the RAST system's corrections, but also detects and corrects a great deal more errors.

Pathfinder begins to introduce new errors in regions that are poorly covered by the BLASTP tool, and regions in which there are many frameshifts very close together. These are areas in which Pathfinder can be improved upon in the future. However, Pathfinder's improvements are far greater and more numerous than errors introduced.

References

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. (1990). Basic local alignment search tool. *Journal of Molecular Biology* 215: 403-410.

Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. (1997). Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res*. 25: 3389-3402.

Craigen WJ, Caskey, CT. (1986). Expression of peptide chain release factor 2 requires high-efficiency frameshift. *Nature* 322: 273-275.

Dayhoff MO, Schwartz RM, Orcutt BC. (1978). A model of evolutionary change in proteins. In M. O. Dayhoff (ed.), *Atlas of Protein Sequence and Structure* 5: 345-352. Washington, DC: National Biomedical Research Foundation

Gharizadeh B, Herman ZS, Eason RG, Jejelowo O, Pourmand N. (2006). Large-scale pyrosequencing of synthetic DNA: A comparison with results from Sanger dideoxy sequencing. *Electrophoresis* 27: 3042-3047.

Goldberg SM, Johnson J, Busam D, Feldblyum T, Ferriera S, Friedman R, Halpern A, Khouri H, Kravitz SA, Lauro FM, Li K, Rogers YH, Strausberg R, Sutton G, Tallon L, Thomas T, Venter E, Frazier M, Venter JC. (2006). A Sanger/pyrosequencing hybrid approach for the generation of high-quality draft assemblies of marine microbial genomes. *Proc Natl Acad Sci*. 103: 11240-11245.

Lesk AM. (2005). *Introduction to Bioinformatics(2nd ed.)*. New York: Oxford University Press.

Lewin B. (2000). *Genes VII*. New York: Oxford University Press.

Margulies M, Egholm M, Altman WE, Attiya S, Bader JS, Bemben LA, Berka J, Braverman MS, Chen YJ, Chen Z, Dewell SB, Du L, Fierro JM, Gomes XV, Godwin BC, He W, Helgesen S, Ho CH, Irzyk GP, Jando SC, Alenquer MLI, Jarvie TP, Jirage KB, Kim JB, Knight JR, Lanza JR, Leamon JH, Lefkowitz SM, Lei M, Li J, Lohman KL, Lu H, Makhijani VB, McDade KE, McKenna MP, Myers EW, Nickerson E, Nobile JR, Plant R, Puc BP, Ronan MT, Roth GT, Sarkis GJ, Simons JF, Simpson JW, Srinivasan M, Tartaro KR, Tomasz A, Vogt KA, Volkmer GA, Wang SH, Wang Y, Weiner MP, Yu P, Begley RF, Rothberg JM. (2005). Genome sequencing in microfabricated high-density picolitre reactors. *Nature* 437: 376-380.

Miller JR, Delcher AL, Koren S, Venter E, Walenz BP, Brownley A, Johnson J, Li K, Mobarry C, Sutton G. (2008). Aggressive assembly of pyrosequencing reads with mates. *Bioinformatics* 24: 2818-2824.

Overbeek R, Begley T, Butler RM, Choudhuri JV, Chuang HY, Cohoon M, de Crécy-Lagard V, Diaz N, Disz T, Edwards R, Fonstein M, Frank ED, Gerdes S, Glass EM, Goesmann A, Hanson A, Iwata-Reuyl D, Jensen R, Jamshidi N, Krause L, Kubal M, Larsen N, Linke B, McHardy AC, Meyer F, Neuweger H, Olsen G, Olson R, Osterman A, Portnoy V, Pusch GD, Rodionov DA, Rückert C, Steiner J, Stevens R, Thiele I, Vassieva O, Ye Y, Zagnitko O, Vonstein V. (2005). The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes. *Nucleic Acids Res*. 33: 5691-5702.

Quinlan AR, Stewart DA, Stromberg MP, Marth GT. (2008). Pyrobayes: An improved base caller for SNP discovery in pyrosequences. *Nature Methods* 5: 179-181.

Ronaghi M, Uhlen M, Nyren P. (1998). DNA sequencing: A sequencing method based on real-time pyrophosphate. *Science* 281: 363-365.

Staden R. (1979). A strategy of DNA sequencing employing computer programs. *Nucleic Acids Research* 6 (7): 2601-10.

Vacic V, Jin H, Zhu JK, Lonardi S. (2008). A probabilistic method for small RNA flowgram matching. *Pacific Symposium on Biocomputing* 13: 75-86.