12-1-2015

Gene duplications during experimental evolution of Caenorhabditis elegans : duplication rates and evolutionary responses

James Charles Farslow

James Charles Farslow
Candidate

Department of Biology
Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Dr. Ulfar Bergthorsson, Chairperson

Dr. Donald Natvig

Dr. Christopher Witt

Dr. Philip Gerrish

GENE DUPLICATIONS DURING EXPERIMENTAL EVOLUTION OF *CAENORHABDITIS ELEGANS*: DUPLICATION RATES AND EVOLUTIONARY RESPONSES

BY

JAMES CHARLES FARSLOW

B.S., Biology, University of New Mexico, 2006

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy

Biology

The University of New Mexico
Albuquerque, New Mexico

December, 2015

DEDICATION

To my wife, Cindy, and our son, Cameron,
for their encouragement, understanding, and patience over all these years.

To my father, Gerald,
who taught me the importance of hard work and honesty.

To my mother, Charlene,
who believed I could do anything.

To my sister, Sandi,
for her encouragement and optimism.

Thank you. I love you all.

ACKNOWLEDGEMENT

I gratefully acknowledge Dr. Ulfar Bergthorsson, my advisor and dissertation chair, for his continued patience and encouragement during these many years of research. He has shared his knowledge of this subject enthusiastically, as well as the facilities of his laboratory, and I will always appreciate his sage advice.

I would like to thank those who have served on my dissertation committee, Dr. Vaishali Katju, Dr. Donald Natvig, Dr. Christopher Witt, and Dr. Philip Gerrish, for their guidance and assistance throughout this endeavor.

I thank Dr. George Rosenberg of the Molecular Biology Facility for his assistance in working with the ABI 7000 Sequence Detection System (qPCR), as well as better understanding the process of the ABI 3130xl Genetic Analyzer (DNA sequencing).

I would also like to thank the coauthors of the manuscripts included in this dissertation, without whose help this research would not have been possible.

Gene Duplications During Experimental Evolution of *Caenorhabditis elegans*: Duplication Rates and Evolutionary Responses.

by

James Charles Farslow

B.S., Biology, University of New Mexico, 2006
Ph.D., Biology, University of New Mexico, 2015

ABSTRACT

Copy-number variants (CNVs) are a ubiquitous form of genetic variation. How often this form of variation arises and its adaptive significance are active areas of contemporary research. This work presents evidence regarding both of these subjects. First, it demonstrates that gene duplications occur at a frequency two orders of magnitude greater than point mutations. Specifically, the gene duplication rate is estimated to be 1.2 × 10^{-7}/gene/generation, compared to a point mutation rate on the order of ~10^{-9}/site/generation. Second, it was found that populations in a low state of fitness due to mutation accumulation could recover some or all of their fitness over short spans of generations concurrent with an increase in frequency of duplications and deletions that arose during the recovery process. The pattern of frequency increase among CNVs over generations during recovery was consistent with the signature of positive selection. The median size of duplications that were identified after selection for ~200 generations were significantly larger (191.5 kb) than both duplications that occurred spontaneously (2 kb) in the absence of selection and deletions identified after selection for ~200 generations (12.5 kb). The median number of genes contained in the duplications during recovery was 38, evincing the ability of these events to increase the genetic information available for selection to act on. These results clearly demonstrate that gene duplication and deletion processes contribute significantly to the adaptability of populations.

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter 1

Introduction

Evolution requires heritable variation within populations for natural selection to act on (Fisher 1930; Haldane 1932; Mayr 1963; Dobzhansky 1970; Futuyma 1998). The genetic variation that gives rise to the phenotypic variation acted on by selection includes not only the single nucleotide variations and small indels that distinguish allelic variants, but also variation in the number of copies of a gene in a genome (paralogs) as well as subsequent variation among those copies.

The processes of gene duplication generate multiple copies of existing genes in a genome, providing an increase in the amount of genetic information available for mutation and selection to act on. While this was primarily thought to produce "more of the same", altering only gene dosage, it is now understood that duplication mechanisms can produce new genetic information in the form of novel genes, either immediately or through a process of relaxed selection followed by diversification (Ohno 1970; Bergthorsson *et al.* 2007; Katju 2012). While point mutations acting alone can be extremely slow at creating new genetic information, duplication, on the other hand, can provide new genes in a single mutational step, either functional copies of existing genes, or merged with other sequence creating new function immediately (Katju 2012). Gene duplication, then, can be a major source of new genetic information.

Gene duplication, in spite of the term, is often not the duplication of a single gene. Depending on the mechanism (Long *et al.* 2003; Liu *et al.* 2012), one gene or many may be duplicated. In some cases hundreds of genes, or even copies of the entire genome, may be duplicated. While a few mechanisms act on single genes, many duplication mechanisms do not target genes *per se*, but rather duplicate segments of DNA which may or may not contain genes or parts thereof.

In order to evaluate the potential contribution of gene duplications, or deletions, to the adaptability of populations, and hence to their evolution, we first sought to determine what the spontaneous rate of gene duplication was in the nematode *Caenorhabditis elegans*, and then, given populations with reduced fitness, whether duplications and deletions contributed to the populations' recovery of fitness, as exhibited by the signature of selection, an increase in frequency over generations.

This dissertation is comprised of five chapters, three of which represent manuscripts either published or currently in review for publication. This Introduction is the first chapter. Chapter 2 discusses our work to determine the rate of spontaneous duplications and deletions using mutation accumulation procedures via bottlenecking the experimental populations. It also provided data on the median size of spontaneous duplicates and deletions. My contribution to the research included analyzing the oaCGH array data, developing qPCR methods to corroborate the oaCGH array results, and performing PCR and DNA sequencing of CNV breakpoints. I also designed all of the primers used for the above procedures. Chapter 3 is the project to investigate whether CNVs provide a means of adaptation, as evinced by a pattern of frequency increase over generations. This research also revealed a different size distribution of CNVs under adaptation compared to the research in Chapter 2. My contributions included, again, oaCGH array analysis, qPCR, and PCR followed by DNA sequencing of CNV breakpoints, including primer design. Chapter 4 discusses the development of statistical techniques for the analysis of the qPCR data from Chapter 3. I developed Matlab programs to perform statistical simulations emulating the production of data from the qPCR process in order to evaluate the effectiveness of different statistical methods. Finally, Chapter 5 is a short conclusion summarizing the main points. Additionally, Chapters 2 – 4 have addendums of additional material not published in the manuscripts due to space constraints.

The references for all of these works are combined in the References section. The numbering of figures and tables is first by chapter, or appendix, then in numerical order (*e.g.*, Figure 3.1).

Chapter 2

High Spontaneous Rate of Gene Duplication in *Caenorhabditis elegans*

Kendra J. Lipinski,[1] James C. Farslow,[1] Kelly A. Fitzpatrick,[1] Michael Lynch,[2] Vaishali Katju,[1] and Ulfar Bergthorsson[1]

[1] Department of Biology, University of New Mexico, Albuquerque, NM 87131, USA

[2] Department of Biology, Indiana University, Bloomington IN 47405, USA

Reprinted from *Current Biology,* **21**(4), Lipinski, K.J., Farslow, J.C., Fitzpatrick, K.A., Lynch, M., Katju, V., and Bergthorsson, U., High Spontaneous Rate of Gene Duplication in *Caenorhabditis elegans*. 306-310, (2011), with permission from Elsevier.
The manuscript was edited to fit the dissertation format.

Summary

Gene and genome duplications are the primary source of new genes and novel functions and have played a pivotal role in the evolution of genomic and organismal complexity (Ohno 1970; Lynch and Conery 2000). The spontaneous rate of gene duplication is a critical parameter for understanding the evolutionary dynamics of gene duplicates; yet few direct empirical estimates exist and differ widely. The presence of a large population of recently derived gene duplicates in sequenced genomes suggests a high rate of spontaneous origin, also evidenced by population-genomic studies reporting rampant copy-number polymorphism at the intraspecific level (Iafrate *et al.* 2004; Sebat *et al.* 2004; Mayden *et al.* 2007; Emerson *et al.* 2008). An analysis of long-term

mutation-accumulation lines of *Caenorhabditis elegans* for gene copy-number changes using array Comparative Genomic Hybridization yields the first direct estimate of the genome-wide rate of gene duplication in a multicellular eukaryote. The gene duplication rate in *C. elegans* is quite high, on the order of 10^{-7} duplications/gene/generation. This rate is two orders of magnitude greater than the spontaneous rate of point mutation per nucleotide site in this species and also greatly exceeds an earlier estimate derived from the frequency distribution of extant gene duplicates in the sequenced *C. elegans* genome.

Results

Most of the recent progress in elucidating the role of gene duplications in the history of life has been the result of analyses of whole genomes using comparative genomics. Although genomes can provide a rich record of the history of gene duplications in a particular lineage, the population-genetic dynamics and selection pressures on duplicated genes remain poorly understood. The spontaneous gene duplication rate shapes the natural variance in gene copy-number and is an important parameter for understanding the early evolutionary dynamics of novel genes (Ohta 1988; Otto and Yong 2002). Ultimately, the frequency of gene copy-number polymorphisms in geomes as well as their rate of fixation is determined by a combination of the spontaneous duplication rate and the probabilities of preservation or elimination of these changes by evolutionary forces such as natural selection, genetic drift, and various mutations (Otto and Yong 2002; Zhang 2003).

Estimates of the spontaneous rate of gene duplication come primarily from three sources: (i) calculations based on the abundance of very recent gene duplications in sequenced genomes (Lynch and Conery 2000; Pan and Zhang 2007), (ii) caluclations assuming mutation-selection balance where the fitness consequences of the duplication are known (Van Ommen 2005), and (iii) direct measurements on individual loci where gene copy-number differences result in a distinct phenotype or genotype (Anderson and Roth 1977; Anderson and Roth 1981; Shapira and Finnerty 1986; Lam and Jeffreys 2007; Watanabe *et al.* 2009). With method (i), Lunch and Conery (2003) utilized the distribution of synonymous site divergence between duplicate genes in several sequenced genomes to estimate a duplication rate of 0.1×10^{-8}/gene/yr in *D. melanogaster*, 0.4×10^{-8}/gene/yr in *S. cerevisiae*, and 1.6×10^{-8}/gene/yr in *C. elegans*, among others. Translating these rate estimates into duplications/gene/generation requires knowledge of the number of generations/year. For *C. elegans*, the rate of gene duplication was calculated to be similar to the synonymous substitution rate, and because the frequency of base substitutions in *C. elegans* has been estimated to be 2×10^{-9}/site/generation in long term mutation-accumulation experiments (MA henceforth) (Denver *et al.* 2009), the gene duplication rate per generation based on the genomic data would then be on the order of 10^{-9} duplications/gene/generation. Method (ii) estimates the rate of gene duplications using the frequency of gene duplications in a population and population-genetic theory of mutation-selection balance. Using this approach, the rate of new gene duplications in the X-linked human dystrophin gene leading to Duchenne Muscular Dystrophy (DMD) was estimated to be $\sim10^{-5}$ duplications/gene/generation (Van Ommen 2005). Direct empirical measures of the gene duplication rate based on method (iii) generally yield much higher values than those generated from those based on extant duplicates in sequenced genomes. For example, reports of locus-specific duplication rates in bacteria, Drosophila, and humans range from 10^{-3} to 10^{-7}/gene/generation (Van Ommen 2005; Anderson and Roth 1977; Anderson and Roth 1981; Shapira and Finnerty 1986; Lam and Jeffreys 2007; Watanabe *et al.* 2009; Turner *et al.* 2008). These estimates are based on a handful of loci and may not be representative of all duplicated loci in these genomes. The discrepancy between the genome sequence estimates and empirical measures is particularly stark in yeast. Bioinformatic analyses of the sequenced yeast genome suggested that the rate of gene duplication in yeast is half that of the per nucleotide base substitution rate (Lynch and Conery 2000). However, whole-genome sequencing of *S. cerevisiae* MA strains has now revealed that the duplication rate per locus is ten thousand-fold higher than the base substitution rate (Lynch *et al.* 2008). The five orders of magnitude discrepancy in the rate of spontaneous gene duplication in preceding studies is likely due to a combination of the use of different gene loci, species, and approaches to quantification.

We used Comparative Genome Hybridization (CGH) to measure the spontaneous gene duplication and deletion rate in *C. elegans* using experimental evolution lines that were generated during a long-term MA experiment (Figure 2.1) by enforcing single-worm bottlenecks each generation to greatly reduce the efficacy of natural selection (Vassilieva and Lynch 1999). Under these conditions, nearly all mutations are able to accumulate in the genome largely independent of their fitness consequences, which enables an estimation of the rate of spontaneous mutations. Analyses of ten *C. elegans* MA lines (bottlenecked for an average of 432 generations) with NimbleGen CGH microarrays detected 14 duplicated and 11 deleted segments that were unique to particular MA lines (Tables 2.1 and 2.2, respectively). These duplications and deletions were verified by quantitative PCR (Tables A.1 and A.2, Appendix A). The 14 duplicated segments involved the *complete* and *partial* duplication (Katju and Lynch 2006) of 11 and 19 loci, respectively. The *C. elegans* genome contains approximately 20,400 protein coding genes (excluding alternative splice forms), so the probability that any given gene

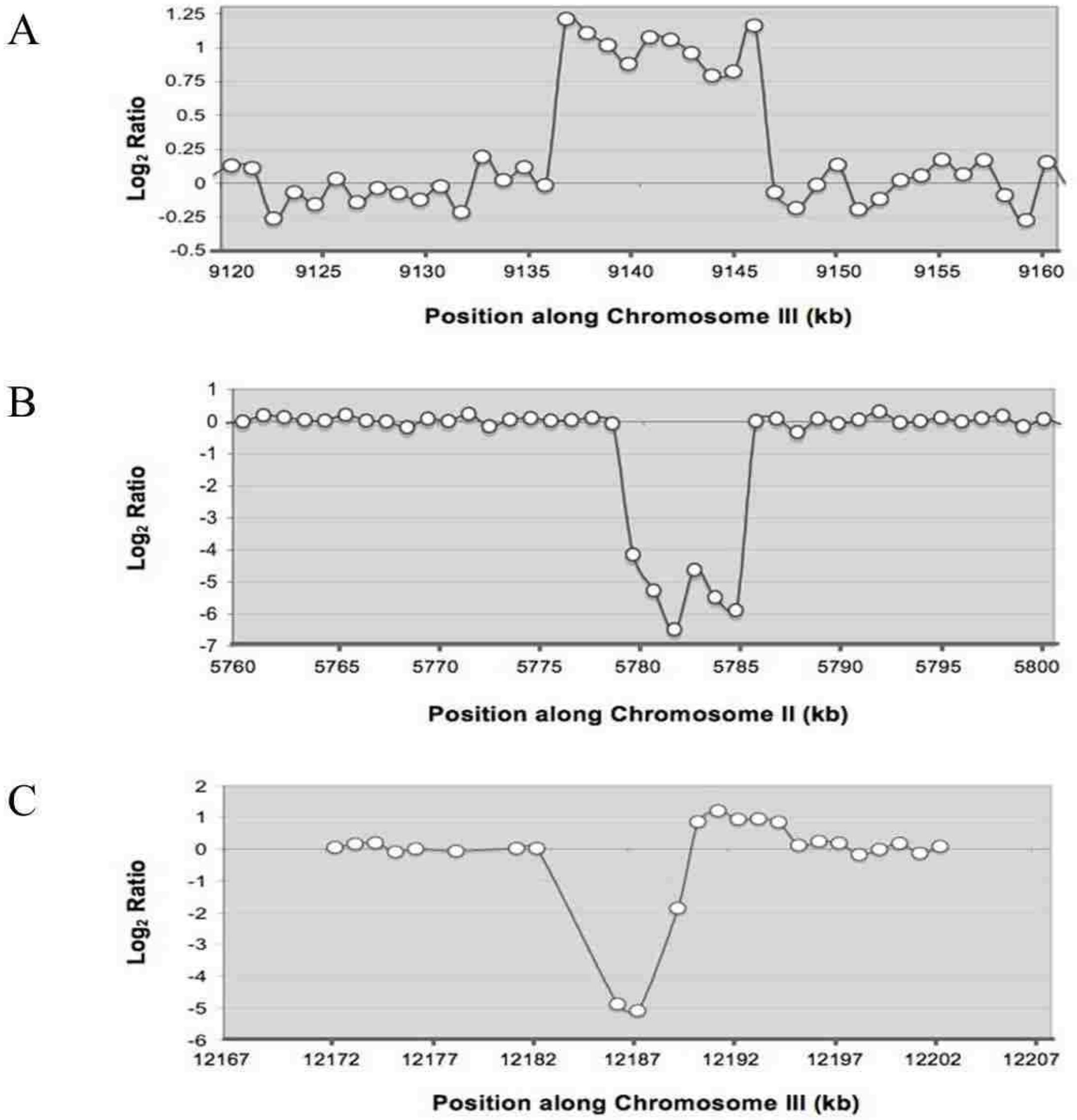


Figure 2.1. Nimblegen CGH array duplication and deletion. Each spot is a $\log_2$ ratio of the fluorescence of the experimental DNA and the control DNA, arranged in linear order according to position on the sequenced chromosome. **A.** Duplication on Chromosome III of MA line 78. **B.** Deletion on Chromosome II in MA line 18. **C.** Adjacent deletion and duplication on Chromosome III of MA line 99.

Table 2.1. Characterization of 15 duplication events detected in ten mutation accumulation lines of *C. elegans* using CGH microarray analysis.

MA Line ID	Bottleneck Generations	Chromosome	Start Position	Stop Position	Length of Duplication (bp)	No. of ORFs (*complete, partial*)
2	438	V	18,507,783	18,519,661	11,878*	3(1,2)
18	464	V	10,445,133	10,455,580	10,448	3(1,2)
18	464	V	17,847,927	17,858,066	10,140	1(0,1)
29	468	IV	17,482,852	17,490,972	8,121	2(1,1)
29	468	X	12,763,189	12,767,835	4,647	2(1,1)
41	438	–	–	–	–	–
63	425	V	4,893	18,375	13,483	2(2,0)
63	425	X	3,559,284	3,567,765	8,482	2(0,2)
78	428	I	6,682,405	6,688,767	6,361*	2(0,2)
78	428	III	9,135,580	9,145,930	10,351*	5(4,1)
78	428	X	7,609	11,592	3,984	1(0,1)
78	428	X	17,694,155	17,696,571	2,417	1(0,1)
83	385	IV	11,695,251	11,700,130	4,880	2(1,1)
84	465	–	–	–	–	–
94	367	III	813,463	819,305	5,843*	2(0,2)
99	464	I	10,716,364	10,721,038	4,675	2(0,2)
99	464	III	12,190,163	12,194,367	4,205	1(0,1)

Quantitative PCR results confirming these duplications are presented in Supplemental Table 1 of Appendix A. Duplication lengths with an asterisk are based on the DNA sequence of duplication breakpoints shown in Supplemental Figures 1A through D in Appendix A. Other length estimates are minimum estimates based on the location of probes included in the duplicated region. The numbers of ORFs were based on Wormbase sequence version WS219.

is duplicated at least partially is 30/(20,400 × 432 × 10) = 3.4 × 10^{-7}/gene/generation. The eleven deleted segments resulted in complete or partial deletions of 19 ORFs and a deletion rate of 2.2 × 10^{-7}/gene/generation.

If only *complete* duplicates are taken into consideration, the average duplication rate per gene becomes 1.2 × 10^{-7}/gene/generation (bootstrap 95% confidence interval = 0.6 – 2.1 × 10^{-7}/gene/generation). Both of these estimates of the gene duplication rate in *C. elegans* are quite high, about two orders of magnitude greater than the spontaneous rate of point mutation per nucleotide in this species (~ 10^{-9}/site/generation) (Denver *et al.* 2009). Additionally, our empirically determined rate of spontaneous gene duplication for experimental *C. elegans* MA lines is two orders of magnitude higher than that determined

Table 2.2. Characterization of 11 deletion events detected in ten mutation accumulation lines of *C. elegans* using CGH microarray analysis.

MA Line ID	Bottleneck Generations	Chromosome	Start Position	Stop Position	Length of Deletion (bp)	No. of ORFs (*complete, partial*)
2	438	–	–	–	–	–
18	464	II	5,779,858	5,784,774	4,917	1(0,1)
29	468	X	12,759,841	12,761,557	7,717	1(0,1)
41	438	–	–	–	–	–
63	425	V	1	3,147	3,147	1(1,0)
78	428	V	7,382,127	7,384,417	2,290*	2(0,2)
78	428	X	12,111	12,925	815	0
78	428	X	17,698,889	17,718,629	19,741	5(3,2)
83	385	II	184	4,901	4,718	1(1,0)
83	385	IV	8,582,021	8,613,791	31,771	5(5,1)
83	385	IV	15,187,709	15,187,923	215	0
84	465	X	6,449,100	6,451,323	2,224*	1(1,0)
94	367	–	–	–	–	–
99	464	III	12,186,190	12,189,700	3,511	1(0,1)

Quantitative PCR results confirming these deletions are presented in Supplemental Table 2 in Appendix A. Deletion lengths with an asterisk are based on the DNA sequence of deletion breakpoints shown in Supplemental Figures 1E and F in Appendix A. Other length estimates are minimum estimates based on the location of probes included in the deleted region. The numbers of ORFs were based on Wormbase sequence version WS219.

from the analyses based solely on the frequency distribution of extant duplicates of varying evolutionary ages in the sequenced N2 genome (Lynch and Conery 2000). Our direct gene duplication rate estimates may in fact be downwardly biased for two reasons, namely (i) that small duplications are likely to go undetected because the number of adjacent microarray probes signaling gene copy-number changes may not be sufficient for detection, and (ii) these CGH DNA microarrays are restricted to unique probes only and duplications of genes in recently duplicated regions, for instance by unequal crossing over, may not be detected. The genome-wide duplication and deletion rate reported here does not add much to the overall mutation rate per genome. The base substitution rate per genome in *C. elegans* is ≈ 0.1/genome/generation (Denver *et al.* 2009) and if we count each duplication and deletion as an independent mutation, then the duplication/deletion rate per genome/generation is 0.007, and 0.011 when the calculation is based on copy-number changes in individual ORFs.

If the duplication and deletion rates are homogeneous across MA lines, the number of copy-number changes per line is expected to be Poisson distributed. Two potential sources of bias in estimating the rate of gene duplication and deletion from MA experiments is that these rates might be subject to change, either due to mutations in recombination and repair genes or due to fitness-dependent differences in the rates (Agrawal and Wang 2008). These two sources of bias would result in a larger variance in gene copy-number changes than expected under the Poisson distribution. Nevertheless, the ratio of the variance to the mean in the number of gene duplications and deletions across different MA lines is close to random expectations (*F*-value = 1.13; $p>0.25$) suggesting the lack of a significant contribution from these two sources.

The duplication lengths ranged from 2.4 – 13.9 kb with a median duplication size of 7 kb. Deletions ranged in length from 0.8 – 31.7 kb with a median value of 3.5 kb. The difference in the length distributions of duplications and deletions are marginally significant (*Wilcoxon two-sample test*; $p = 0.05$). However, small deletions are more likely to be detected relative to small duplications and this may have influenced the difference in the median length of duplication and deletions. The median duplicon size of 7 kb in this data set is significantly greater than the median duplication size of 1.4 kb (Katju and Lynch 2003) for extant evolutionarily young gene duplicates with low synonymous divergence in the sequenced genome of the N2 laboratory strain of *C. elegans* (*Wilcoxon two-sample test*; p < 0.0001). This discrepancy can be due to either one or a combination of three possibilities, namely, (i) duplications are contracting in length due to internal deletions subsequent to their origin, (ii) there is purifying selection against larger duplicates, and/or (iii) CGH arrays are biased in favor of detecting larger duplications.

The spontaneous duplications and deletions in the ten MA lines were spread across all six chromosomes in the *C. elegans* genome (Figure 2.2a). Four duplications appear to be coupled with adjacent deletions and two of these are located at the ends of chromosomes. In addition, four duplications appear to involve more than a single copy addition, usually resulting in three to four copies, but I one case, perhaps as many as eight copies according to the qPCR results. Using divergent primers at the end of duplicons, we sequenced the breakpoints associated with four duplications and two deletions (Figures S1a-f). We were not successful in sequencing the coupled and high copy-number duplications using this strategy which is only expected to yield results when the duplicated segments are adjacent and there are no further rearrangements associated with the copy-number change. The breakpoints indicate direct tandem duplications with little or no sequence identity at the ends of the duplicons (Figures S1a-d). Moreover, in some instances, several additional nucleotides have been inserted at the breakpoint (Figures S1a,i, and j). One deletion appears to have been the result of unequal crossing-over (Figure S1e).

In addition to the copy-number changes unique to individual MA lines, we also observed six copy-number differences that are shared among all the MA lines. These comprise five duplications and one deletion ranging from 634 to 19,358 bp (Tables 2.3 and S3, Figures 2.2b and S1g-j). These differences represent copy-number changes between different N2 laboratory isolates of *C. elegans*, specifically the N2 laboratory

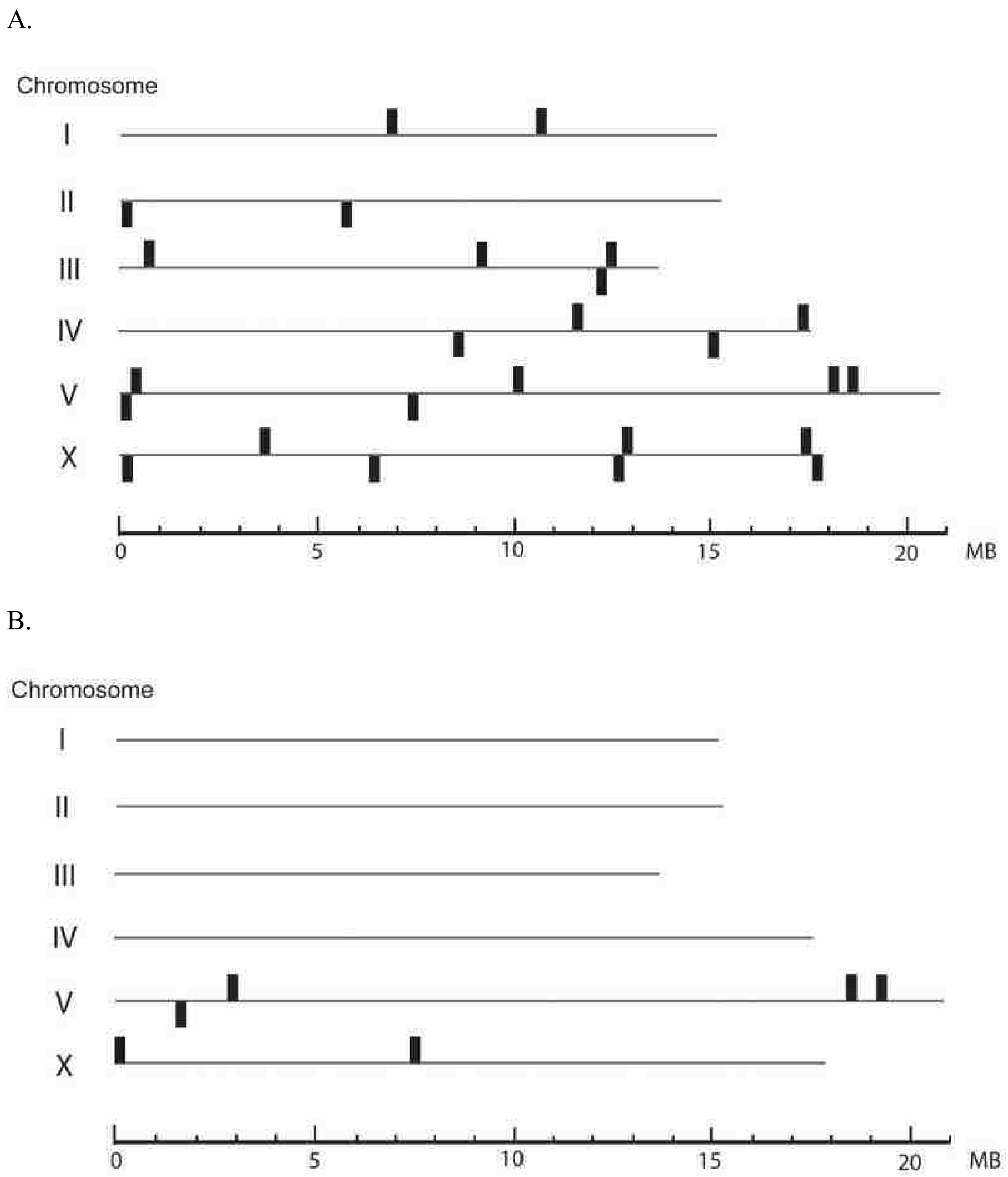


Figure 2.2. Chromosomal distribution of spontaneous duplications and deletions. The horizontal lines represent the six chromosomes comprising the *C. elegans* genome. A. Location of 14 duplications and 11 deletions across ten mutation accumulation (MA) lines derived from a single hermaphrodite of a N2 laboratory isolate of *C. elegans*. Black shaded rectangles above and below the line denote the location of duplications and deletions, respectively. B. Location of inferred duplications and deletions in the N2 laboratory isolate of *C. elegans* that was the source of reference DNA in the CGH microarray experiments.

Table 2.3. Characterization of duplication and deletion events detected in the common N2 ancestor of all MA lines and the reference strain of N2 used for hybridization against ten mutation accumulation lines of *C. elegans* for CGH microarray analysis.

Chromosome	Start Position	Stop Position	Length of Indel (bp)	No. of ORFs (*complete, partial)*
Duplications:				
V[a]	2,995,387	2,999,015	3,628*	2(0,2)
V[b]	18,706,963	18,726,320	19,358	3(2,1)
V[b]	19,428,007	19,431,266	3,260*	1(0,1)
X[b]	86,369	87,002	634	1(0,1)
X[b]	7,510,066	7,523,734	13,668*	1(0,1)
Deletions:				
V[c]	1,645,712	1,647,498	1,786*	1(0,1)

Quantitative PCR results confirming these duplications and deletions are presented in Supplemental Table 3, Appendix A. Duplication lengths with an asterisk are based on the DNA sequence of duplications and deletion breakpoints show in Supplemental Figures 1g through j, Appendix A. Other length estimates are minimum estimates based on the location of probes included in the duplicated region. The numbers of ORFs were based on Wormbase sequence version WS219.

[a,c] correspond to a duplication and deletion event in the common N2 ancestor of all MA lines.

[b] corresponds to duplication events in the N2 reference strain used for the CGH microarray analysis.

strain that was used as source of DNA in our CGH microarray experiments and the N2 laboratory strain that served as the ancestral stock for all the experimental MA lines established by Vassilieva and Lynch (1999). The deletion in the common N2 ancestor of all the MA lines was recently described as a common deletion found in strains that were subjected to mutagenesis with ethyl methanesulfonate and may in fact have been present in the genetic background of these strains prior to mutagenesis (Sarin *et al.* 2010).

Discussion

The rate of fixation of duplicated genes due to beneficial, neofunctionalizing mutations has been shown to be dependent on the species' effective population size as well as the rate of duplication (Ohta 1988; Lycnh *et al.* 2001). The direct estimates of gene duplication rates are two orders of magnitude greater than the per nucleotide point mutation rate. This may have important consequences for the role of adaptation in the evolution of duplicated genes. Theoretical and empirical work show that the mutation rate is an important determinant of the rate of fixation of adaptive mutations and that less-fit beneficial mutations can be fixed in the population earlier than the fittest mutation if the former are more frequent (Yampolsky and Stoltzfus 2001; Rokyto *et al.* 2005). For instance, if an adaptation to a novel environment requires an increase in the expression of a particular gene, and the gene duplication rate far exceeds the per nucleotide base substitution rate, advantageous duplications of the locus are more likely to occur and become fixed in the populations before beneficial point mutations. This may explain why recent adaptations in natural populations have often involved an increase in gene dosage through gene duplication and amplification rather than regulatory base substitutions (Bergthorsson *et al.* 2007; Nair *et al.* 2007; Perry *et al.* 2007). Once such adaptive duplications have become common or fixed, they become targets for mutations that increase the genetic repertoire of the organism. Were beneficial base substitutions more frequent than duplications, an increase in expression would more often be achieved by base substitutions rather than gene duplications. Hence, the relative rates of point mutations and duplications can play an important role in the evolutionary potential of genomes.

A large fraction of duplications do not span the coding sequence of genes in their entirety, and others are unlikely to capture the complete array of upstream regulatory sequences. This may predispose gene supplicates to subfunctionalization, as the first step in this process is the loss of an essential feature in one copy (Katju and Lynch 2006; Katju and Lynch 2003; Force *et al.* 1999; Lynch and Katju 2004). Moreover, failure to capture the full coding sequence or regulatory repertoire of the ancestral copy may predispose the duplicate copy to a different evolutionary trajectory wherein the ancestral copy is likely to retain its original function and the derived copy is more likely to be neofunctionalized, subfunctionalized, or pseudogentized. Indeed, recent analysis suggests that derived gene copies are evolving at faster rates relative to their ancestral counterparts (Cusack and Wolfe 2007; Han *et al.* 2009).

All empirically-derived estimates of the spontaneous duplication/deletion rates, be they locus-specific (Anderson and Roth 1977; Anderson and Roth 1981; Shapira and Finnerty 1986; Lam and Jeffreys 2007; Watanabe *et al.* 2009; Turner *et al.* 2008) or genome-wide (Lynch *et al.* 2008), are much greater than bioinformatically-derived estimates from extant duplicates in sequenced genomes for a diverse set of organisms across different kingdoms. This strongly suggests that most gene duplications are efficiently purged from the genome by purifying natural selection in their infancy, leaving a surviving observable pool dominated by duplicates with lower rates of loss. In fact, recent population-genetic analyses of gene copy-number polymorphism found an excess of rare duplications suggestive of purifying selection in *Drosophila melanogaster* (Emerson *et al.* 2008). Thus, prior genome-based estimates of the gene duplication rate may only reflect the birth rates of initially neutral or nearly neutral duplications. If this is the case, we predict that the discrepancy between bioinformatically- and empirically-derived estimates of the gene duplication rate will correlate positively with effective population size. In the case of the yeast *Saccharomyces cerevisiae*, the rate of spontaneous mutation has been measured as 0.7×10^{-9} substitutions/site/generation (Lynch *et al.* 2008) and the parameter $N_e\mu$ is approximately 0.023 (Lynch and Conery

2003), giving an estimated N_e of 3.3×10^7. This estimated N_e for *S. cerevisiae* is extremely similar to that measured for its close relative, *S. paradoxus* ($\approx 10^7$) (Tsai *et al.* 2008). In the case of *S. cerevisiae*, with a large effective population size, the discrepancy between the bioinformatics and empirical estimates of the gene duplication rate (Lynch and Conery 2000; Lynch *et al.* 2008) spans five orders of magnitude. In contrast, the discrepancy is only two orders of magnitude in the case of *C. elegans*, where the effective population size has been estimated as 9×10^4 individuals (Cutter 2006). However, it is possible that the present level of genetic variation in *C. elegans* and hence its small effective population size result from the recent evolution of hermaphroditism in this species (Cutter *et al.* 2009). For comparison, the estimated effective population size of *C. remanei*, an obligate outcrosser, is 1.6×10^6 (Cutter and Charlesworth 2006).

Most gene duplicates confer a slight penalty on the fitness of the carrier, possibly due to an initial dosage imbalance. Microorganisms and unicellular eukaryotes with their large effective population sizes and greater efficacy of selection may more effectively purge these newly arisen duplicates with their mildly deleterious effects. Conversely, the relatively smaller effective population sizes of many multicellular eukaryotes compromise their ability to efficiently rid their genome of the new entrants.

Supplementary Material

Supplementary information, including methods, is contained in Appendix A. Matlab program information is contained in Appendix B.

Acknowledgments

We thank Lijing Bu, Ushnik Gosh, Joshua M. Plank, Hallie S. Rane, George Rosenberg, and Jessica M. Smith for technical assistance. This work was supported by a National Institutes of Health National Center for Research Resources Grant P20-RR18754, a National Science Foundation grant DEB-0952342 (U.B. and V.K.), and a National Science Foundation Postdoctoral Fellowship in Biological Informatics DBI-0532735 (V.K.). M.L. was supported by a National Institutes of Health grant RO1-GM036827 and National Science Foundation grant EF-0827411. The authors declare that they have no competing financial interests.

Chapter 2 Addendum

The lead author of this manuscript, Kendra Lipinski, extracted DNA from the experimental populations and submitted it for CGH microarray hybridizations. She also had begun some of the analyses trying to identify duplications and deletions. I took over the project after she graduated from the master's degree program, and began by analyzing the CGH microarray data looking for duplications and deletions, specifically trying to identify their boundaries. I then performed qPCR on the suspected duplication or deletion regions to corroborate the CGH microarray data, and PCR and sequencing of the boundary regions to identify the specific breakpoints of the rearrangements when possible.

To look for duplications and deletions, our experimental lines were compared by CGH microarray to Bristol N2 populations which served as controls (Appendix A). These experimental populations of *C. elegans* were bottlenecked for over 400 generations, therefore they should have been fixed for any copy number variant (CNV).

This infers the signal level of the CGH microarray should reflect the copy number change in the haploid genome, and so the microarray signal levels should exhibit discrete differences from the single copy level. The program SnoopCGH (Alagro-Garcia *et al.* 2009) was initially used to identify CNVs. It uses levels of statistical significance and robustness based on permutations to identify CNVs. One of the issues that arose with the data is that there appeared to be a wave pattern along the chromosomes that may be an artifact of the process of scanning the CGH microarray. SnoopCGH occasionally identified the peak of the wave as a duplication. Visual confirmation was thus required to check what the program was identifying. Also, the probe signal levels in these CGH microarray results exhibited a large variance, which caused difficulty at times in trying to visualize or identify copy number changes and their boundaries. Additionally, there were issues with gaps in the probe sequence. This was because the CGH microarray design at the time did not include many probes for repetitive elements.

As the CNVs in this experiment were capable of presenting a visually identifiable image (Figure 2.1), provided the variance of the data was not excessive, another approach was done. A Matlab script was written (JCFreadCGH, Appendix B) which first performed a smoothing algorithm to reduce the variance of the data, then plotted both the raw data and the smoothed data for each chromosome, along with red reference lines representing plus and minus two standard deviations of the unsmoothed data in the smoothed data graph for visual reference (Figure 2.3). The smoothing algorithm took the average of the signal levels within a window that included a given probe and a specified number of probes up and downstream of it, and assigned this value to the given probe's position. For this analysis, the smoothing algorithm used a window of ± 10, which gives an average of 21 contiguous probes (contiguous by order, not necessarily by position because of gaps). This presentation of the data facilitated easier visual recognition of CNVs. The predicted boundary positions of the CNVs were judged according to the position of probes considered to be part of the CNV. For both duplications and deletions, the first probe inside the CNV is the predicted boundary. One of the potential issues that arose was a reduction of signal shift for small (< 1 Kb) duplications and deletions that was a result of the smoothing. With only a few probes in a CNV region, the smoothing removed the most variant probe signals, creating the need to compare the smoothed region with the original signal data.

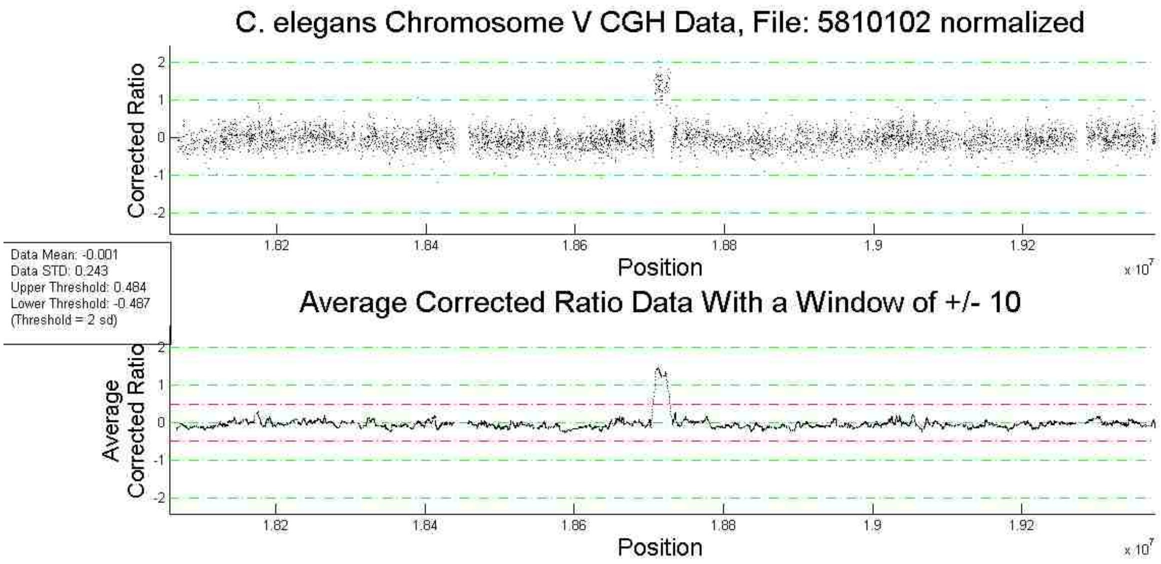


Figure 2.3. Visualization graph of CGH microarray data from JCFreadCGH. The onscreen graph titles provide information as to the chromosome, source data file, and window size. This specific graph represents the duplication in MA18 chromosome V. Only a subset of the chromosome is displayed for this figure.

As a means of confirming the microarray data, quantitative, or real-time, PCR (qPCR) was performed on all CNVs identified. DNA from an N2 population was used as the control for the qPCR experiments. The method is explained in Appendix A. Initially, SYBR Green without Rox was used as the methodology was developed.

However, the ABI 7000 Sequence Detection System has only one immobile bulb and sensor creating differences in measurements on different parts of the plate. The method was changed to SYBR Green with Rox, which has an internal reference dye (i.e., the Rox) that the instrument can use to compare with the SYBR Green signal. This solved the issue of high variance due to plate position. Also, qPCR is very susceptible to pipetting errors, including tiny droplets that can be pulled from the end of the pipette due to static charge between the pipette and the plate. Minimizing the distance traveled over the plate can reduce some of these errors. Thorough mixing of the DNA samples is crucial to producing sets of technical replicates with a low standard deviation.

Finally, it was desired to obtain the precise DNA sequences of the breakpoints (i.e., site of the rearrangement) of the CNVs, if possible. To do this, first PCR amplification was performed, followed by sequencing of the fragments. To PCR the breakpoints of a tandem duplication, primers must be designed to anneal just within the duplicated region pointing away from each other (Figure 2.4A; see Appendix A for more detail on the PCR methods). If a tandem duplication occurs, one pair of primers will orient facing each other. Provided they are close enough, then a PCR reaction can occur. Due to the aforementioned gaps and, in some cases, ambiguities of predicted boundaries, numerous attempts with primers in different positions were required to achieve PCR, and in many cases still failed. If a proximal inversion occurs, a primer will be paired with itself thus facilitating PCR (Figure 2.4B). If, however, a duplication results from translocation, the primers will not be able to align properly and will not be able to produce a correct PCR product (though spurious products can, and did, occur).

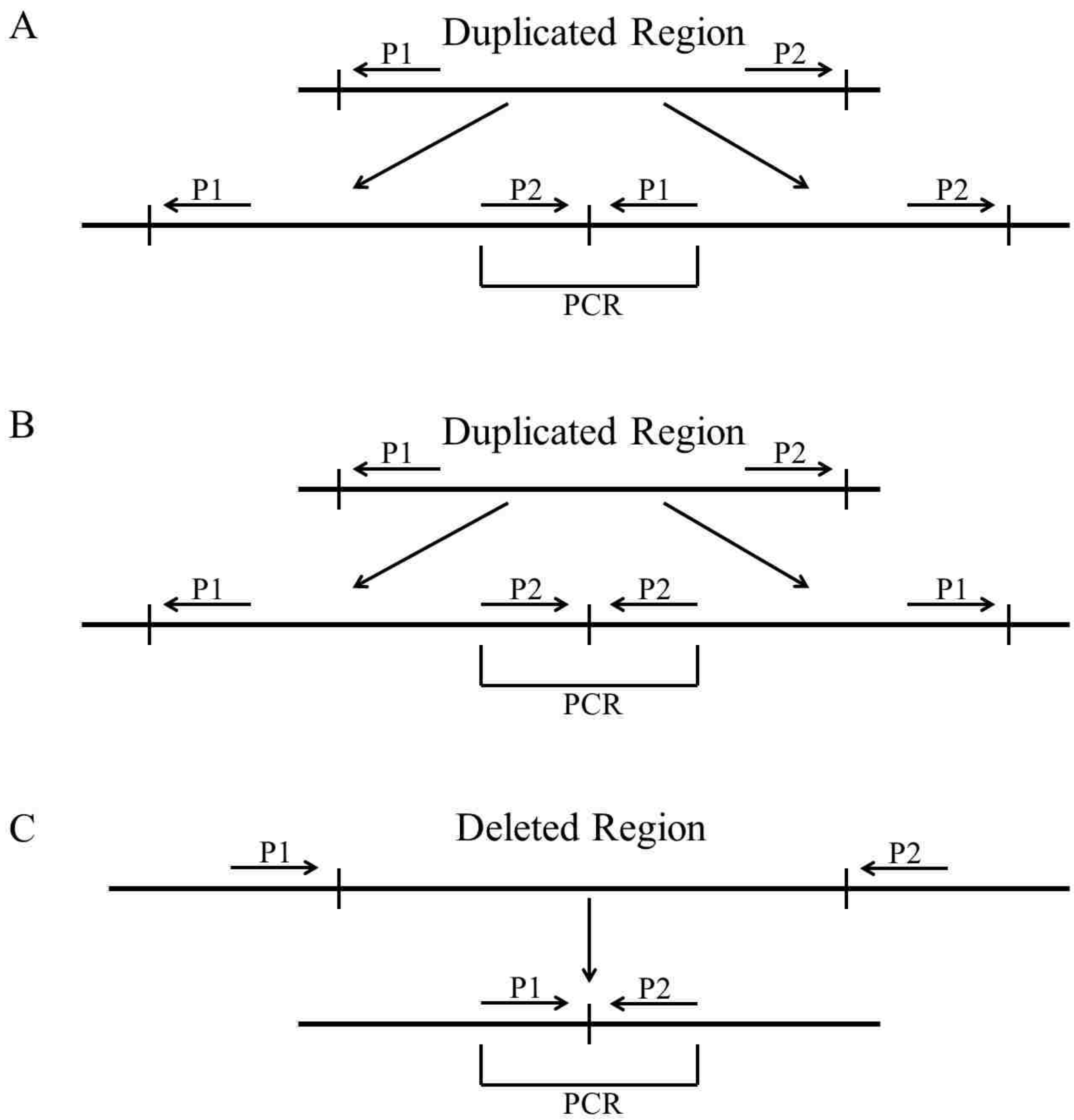


Figure 2.4. PCR amplification of CNV breakpoints. P1 and P2 represent PCR primers and their orientation. **A.** Tandem duplication. **B.** Inverted proximal duplication. **C.** Deletion.

For deletions, primers were designed to anneal to regions outside of the predicted breakpoints facing each other so that when the deleted region was removed, the primers were brought into close enough proximity to generate a PCR product (Figure 2.4C). Again, if the deletion was coupled with a translocation event, no valid PCR products were produced.

A distinct PCR product of approximately the predicted size itself can infer the existence of a CNV in a population, but having the sequence of the fragment and mapping it to the genome provides the best evidence for the rearrangement. When PCR products were successfully produced, they were sequenced (see Appendix A for details on the sequencing methods) and blasted (Altschul *et al.* 1990) against the *C. elegans* genome version WS219 (www.wormbase.org) for position information. The position of alignments to the reference genome revealed the precise point in many cases, or at least the narrow range where there were microhomologies, of the DNA rearrangements producing the CNVs (Figure A.1, Appendix A).

There were clearly cases in which no valid PCR products were produced, and thus no sequences for the CNV breakpoints were obtained. It is likely that these are the result of either translocations or more complex rearrangements than what we tested for. If duplications were the result of proximal inversion, then there should be PCR products from single primer reactions, and there were. However, none of these produced a PCR product that generated a clean DNA sequence. At one point, cloning was performed on some of these fragments (TOPO® TA Cloning® Kit for Sequencing, Invitrogen) to attempt to sequence them. After blasting against the NCBI nucleotide database (blast.ncbi.nlm.nih.gov), the results turned out to be fragments of bacterial contamination, some from *E. coli,* probably from the populations used to feed the worms. One cloning experiment did produce fragments aligning to *C. elegans*, but they failed to suggest a rearrangement.

The variation in the sequences of the breakpoint regions suggests the possibility of different mechanisms producing the rearrangements, but not necessarily the specific mechanism in a given case. Out of 15 duplications identified by CGH microarray and qPCR, only four produced PCR product and sequence for the breakpoints. Out of 11 deletions, only two produced PCR product and sequence for the breakpoints. In both cases either the PCRs weren't working, or the 20 CNVs without breakpoints may have involved translocations or complex rearrangements.

Figure A.1A (Appendix A) shows a duplication in chromosome V of experimental line MA2. There is no homology between the up and downstream ends of the duplicated region suggesting NAHR (Non-Allelic Homologous Recombination (Beckmann *et al.* 2007), also referred to as unequal crossover elsewhere in the text) is not likely the mechanism of duplication. What we do see is four As and four Ts, in palindromic arrangement, inserted at the breakpoint. There are a large number of As and Ts in the sequences at the ends, but no specific sequence that matches the insertion. While this might implicate a repair enzyme in the process such as NHEJ (Non-Homologous End Joining (Moore and Haber 1996)), at this point it is merely speculation. It should be pointed out that while we sequenced the breakpoints, we never performed PCR and sequencing on the end points into the adjacent sequence to look for alterations there.

In Figure A.1B (Appendix A), we see a duplication involving a microhomology, a short sequence shared between the up and downstream ends which remains at the breakpoint, in chromosome I of line MA78. This precludes the ability to identify a single point where the transition from one new paralog to the other occurs. Rather, the transition resides somewhere within or immediately adjacent to the microhomology. This does raise the question as to how small of a homology is required for NAHR, but it is difficult to accept that only two nucleotides would be sufficient.

Figure A.1C (Appendix A) illustrates a duplication in chromosome III of MA78 that is “clean”, meaning the sequence transitions smoothly from one paralog to the next with no added nucleotides. In this case again we see no homology between the ends, inferring NAHR is not involved. The same is true of the duplication in chromosome III of MA94 (Figure A.1D (Appendix A)).

NAHR not only generates duplications, but also deletions. Figure A.1E (Appendix A) presents a deletion in chromosome V of line MA78 of the intervening region between two pre-existing paralogs. The paralogs show high levels of homology for a span of 591 nucleotides oriented in the same direction, thus providing the conditions favorable for NAHR. It should also be noted that this deletion is clean, although where within the paralog region the unequal crossover occurred is undeterminable if the identity between the paralogs is 100%.

Deletions may also be caused by other mechanisms. Figure A.1F (Appendix A) illustrates a deletion in chromosome X of line MA84 that contains a microhomology at the breakpoint, but no other homology between the ends, suggesting something other than NAHR was responsible for this rearrangement.

Additionally, CNVs were found in the N2 reference strain. These included a paralog amplification, likely via NAHR, from two copies to three in chromosome V (Figure A.1G (Appendix A)). There was also a complex duplication in chromosome V (Figure A.1I (Appendix A)) where part of the duplicated region is started (15 nucleotides), then the mechanism backed up a little further and started again. There was also a deletion in the N2 strain (Figure A.1H (Appendix A)) in chromosome V that contained a single nucleotide microhomology with no other homology between the ends.

Are the differences between these rearrangements the result of chance or are they caused by the specific mechanisms involved? It is difficult to say at this stage, though it does evince some of the variation possible among CNVs. The data presented in this chapter does not, however, provide a large enough sample size to make generalizations about patterns in CNVs.

As CNVs are really duplications, or deletions, of regions of DNA which may or may not contain genes or parts thereof, the mechanisms generating CNVs affect the number of genes duplicated, or deleted, as well as the rate at which it happens. Mechanisms such as retroposition (Long *et al.* 2003), which reverse transcribes mRNA into the genome, tend to be truly “gene duplication”, only making extra copies of single genes, though they are inserted without their promoter or regulatory elements, or any introns. They become pseudogenes unless they are inserted into the genome downstream of an existing promoter. Also, as a consequence of the mechanism, these duplicates should tend to be small, with a median size just slightly larger than the median gene size for the organism. Replication, recombination, and repair mechanisms, on the other hand, have the capacity to duplicate large tracts of the genome, including hundreds of genes with their regulatory elements in a single mutation. The relative rates of these different mechanisms should be reflected in the patterns of CNV types found in a genome.

The mechanisms are also responsible for the patterns of distribution of CNVs in the genome. Duplicate genes may be predominantly proximal to their ancestor, or they may be widely distributed in the genome occurring primarily on other chromosomes

(Katju *et al.* 2009). Mechanisms involving molecular intermediates should tend to disperse CNVs.

As mentioned above, the rates of mechanisms also have a profound effect on the evolutionary direction of a population. Mechanisms that occur more often than point mutations may provide adaptive advantages sooner, becoming fixed in the population and providing more genetic information for selection to work on. The combination of the type and rate of mechanisms affects the potential evolvability of the genome. It should be noted that the duplication rates determined here reflect a composite of the rates of all of the mechanisms involved.

The mechanisms also affect the type of CNVs produced. Duplication mechanisms can generate both complete and partial duplications (only a contiguous subset of the gene sequence is duplicated). Partial duplicates may insert into a region where they acquire novel sequence, producing a gene product with a new potential functionality (Katju and Lynch 2006). Additionally, partial sequence may be merged with the partial sequence of another gene, fusing different functional domains into a new combination (Long *et al.* 2003; Katju and Lynch 2006). Deletions, besides removing genes, may bring parts of genes together at the breakpoint, again generating novel constructs.

In summary, this work identified the high rate of spontaneous duplications and deletions. It presented information on the span of spontaneous CNVs. It also revealed variation in the breakpoints of CNVs, suggesting multiple mechanisms involved in the process of CNV formation.

Chapter 3

Rapid Increase in Frequency of Gene Copy-Number Variants During Experimental Evolution in *Caenorhabditis elegans*

James C. Farslow[1], Kendra J. Lipinski[1], Lucille B. Packard[1], Mark L. Edgley[2], Bin Shen[2], Jon Taylor[2], Stephane Flibotte[2], Donald G. Moerman[2], Vaishali Katju[1,3] and Ulfar Bergthorsson[1,3]*

[1]Department of Biology, University of New Mexico, Albuquerque, NM 87131, USA.

[2]Department of Zoology, University of British Columbia, Vancouver, British Columbia V6T 1Z4, Canada.

[3]Current address, Department of Veterinary Integrative Biosciences, Texas A&M University, College Station, TX 77843-4458, USA.

This manuscript is in review with *BMC Genomics*, and has been edited to fit the dissertation format.

Abstract

Background:

Gene copy-number variation (CNVs), which provides the raw material for the evolution of novel genes, is widespread in natural populations. We investigated whether CNVs constitute a common mechanism of genetic change during adaptation in experimental *Caenorhabditis elegans* populations. Outcrossing *C. elegans* populations with low fitness were evolved for >200 generations. The frequencies of CNVs in these

populations were analyzed by oligonucleotide array comparative genome hybridization, quantitative PCR, PCR, DNA sequencing across breakpoints, and single-worm PCR.

Results:

Multiple duplications and deletions rose to intermediate or high frequencies in independent populations. Several lines of evidence suggest that these changes were adaptive: (i) copynumber changes reached high frequency or were fixed in a short time, (ii) many independent populations harbored CNVs spanning the same genes, and (iii) larger average size of CNVs in adapting populations relative to spontaneous CNVs. The latter is expected if larger CNVs are more likely to encompass genes under selection for a change in gene dosage. Several convergent CNVs originated in populations descended from different low fitness ancestors as well as high fitness controls.

Conclusion:

We show that gene copy-number changes are a common class of adaptive genetic change. Due to the high rates of origin of spontaneous duplications and deletions, copy-number changes containing the same genes arose readily in independent populations. Duplications that reach high frequencies in these adapting populations were significantly larger in span. Many convergent CNVs may be general adaptations to laboratory conditions. These results demonstrate the great potential borne by CNVs for evolutionary adaptation.

Background

Gene and genome duplications are the primary source of new genes and have played a pivotal role in the evolution of genomic and organismal complexity (Ohno 1970; Zhang 2003; Innan and Kondrashov 2010; Katju 2012). The rates of spontaneous gene duplication and deletion are extraordinarily high and speak to the enormous potential of these structural variants for generating new adaptive variability (Anderson and Roth 1981; Shapira and Finnerty 1986; Lynch *et al.* 2008; Lipinski *et al.* 2011; Schrider *et al.* 2013; Katju and Bergthorsson 2013). However, most gene duplicates are eventually lost from populations due to a variety of reasons: genetic drift or natural selection, inherent instability of tandem duplications, and relaxed selection against detrimental mutations (Anderson and Roth 1981; Katju and Lynch 2003; Veitia 2004; Pettersson *et al.* 2009; Adler *et al.* 2014). Although, gene duplications and deletions contribute significantly to the immense standing genetic variation related to gene copy-number observed in natural populations (Emerson *et al.* 2008; Nair *et al.* 2008; Maydan *et al.* 2010; Mills *et al.* 2011), the relative importance of genetic drift versus natural selection in determining their evolutionary fate remains obscure.

Ohno (1970) theorized that newly duplicated genes were freed from the constraints of natural selection, implicating a dominant role of genetic drift in their early evolutionary dynamics. Likewise, genetic drift is assumed to be the dominant force in the early evolutionary history of duplicate genes under the DDC (duplication-degeneration-complementation) model (Force *et al.* 1999). In contrast, natural selection for increased gene expression may represent an important mechanism by which duplicate gene copies are maintained in populations (Adler *et al.* 2014). There is ample evidence for the preservation of multiple gene copies due to selection for increased gene dosage in diverse organisms (Bergthorsson *et al.* 2007). For example, adaptation to novel or resource-limited environments in laboratory populations frequently involves segmental duplications (Tlsty *et al.* 1984; Sonti and Roth 1989; Reams and Neidle 2003; Andersson and Hughes 2009). Likewise, natural populations harbor duplications that are clearly adaptive under novel environmental regimes (Maroni *et al.* 1987; Gonzalez *et al.* 2005; Newcomb *et al.* 2005; Perry *et al.* 2007; Kondrashov 2012). In addition, loss-of-function mutations can often be suppressed or compensated for by multiple copies, or increased transcription of another gene in the genome (Berg *et al.* 1988; Bender and Pringle 1989; Trempy and Gottesman 1989; Ueguchi and Ito 1992; Yamanaka *et al.* 1994; Serebrijski *et al.* 1995; Timms and Bridges 1998; Menez *et al.* 2001; Miller and Raines 2004; Patrick *et al.* 2007; Hughes *et al.* 2000; Riddle and Brenner 1978; Maruyama *et al.* 1989; Jones *et al.* 2012). The spontaneous rate of gene deletions is of a similar magnitude as that of duplications (Lipinski *et al.* 2011; Schrider *et al.* 2013). There is evidence that deletions tend to be more detrimental to fitness than duplications (Conrad *et al.* 2010). However, gene loss has also been associated with adaptation in diverse systems (Chan *et al.* 2010; Koskiniemi *et al.* 2012; Lee and Marx 2012).

We have previously established that the spontaneous, genome-wide rate of gene duplication in *C. elegans* is two orders of magnitude higher than the point mutation rate (Lipinski *et al.* 2011). In this study, we seek to determine if gene copy-number changes are a common class of genetic change during adaptation and what role, if any, natural selection plays in the maintenance and frequency increase of copy-number variants (CNVs henceforth) in experimental populations. Gene copy-number changes were analyzed in experimental lines of *C. elegans* which had been subjected to (i) fitness decline via mutation accumulation, and (ii) subsequent adaptive fitness recovery during population expansion for >200 generations. In addition, control lines maintained at large population sizes without having been subjected to mutation accumulation were also analyzed for copy-number changes. We used an obligately outcrossing strain of *C. elegans* to reduce the effects of genetic hitchhiking (Maynard Smith and Haigh 1974). These fitness-recovered populations were subsequently analyzed for copy-number changes to directly test if recovery lines display high rates of duplications and deletions, and to determine the role of these CNVs in adaptive evolution.

Results

Fitness decline during mutation accumulation (MA) and subsequent fitness increase following population expansion

This experimental evolution study comprised two distinct phases, (i) a mutation accumulation with a *msh-2* knockdown (MA) phase, followed by (ii) an adaptive recovery phase in the absence of *msh-2* knockdown (see Materials and Methods; Supplementary Figure C.1, Appendix C). Figure 3.1 displays the fitness trajectories of the five focal experimental lines via three fitness assays spanning both phases of the experiment (MA and population expansion), as measured by the life-history trait productivity. Ancestral pre-MA control lines had a mean productivity value of 464 progeny and were assigned a relative mean productivity value of 1.00. At 24 MA generations, the mean productivity of the five experimental lines ranged from 0.2 – 220 progeny (relative mean productivity of 0.004-47% compared to the ancestral control, Figure 3.1). The mean productivity of the five focal MA lines at the termination of the MA 1 phase (50 MA generations) was 31 offspring and the individual mean productivity of the five experimental MA lines ranged from 2 – 60 progeny (relative mean productivity of 0.43-13% compared to the ancestral control, Figure 3.1). ANOVA analyses found a significant variance component for productivity ($F = 40.1$; $p < 0.0001$) between the control and the five MA lines.

Following 150 generations of population expansion, we observed modest to substantial fitness recovery in the experimental lines (Figure 3.1). The mean productivity of the 25 adaptive recovery populations (that were descended from the five MA lines) ranged from 115 – 472 progeny, and relative productivity of 0.25-1.02 (25-102% relative to the ancestor). Populations 16A-E, descended from MA16, exhibited complete fitness recovery to ancestral levels with respect to productivity (average 472 progeny). Populations 66A-E, descended from MA66, exhibited substantial fitness recovery to 73% of ancestral levels with respect to productivity (average 341 progeny). Populations 7A-E, 19A-E, and 50A-E, descended from MA7, MA19, and MA50, respectively, had modest increases in productivity, ranging from 25-33% of ancestral levels (average productivity of 120, 153, and 115, respectively). The mean productivity of the five MA following 50 generations and the 25 recovery populations following ~150 generations was 31 and 274 offspring, respectively. ANOVA analyses found a significant variance component for productivity between the mutation accumulation lines and the recovery populations ($F = 16.9$; $p < 0.0001$).

CNVs comprise a common class of genetic change during adaptive recovery

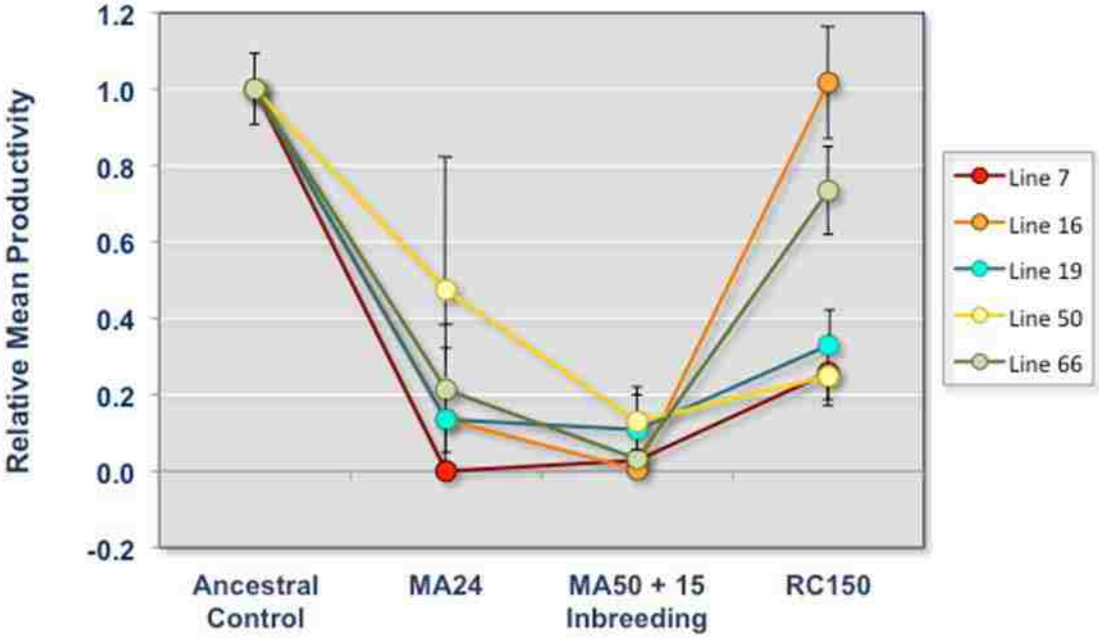


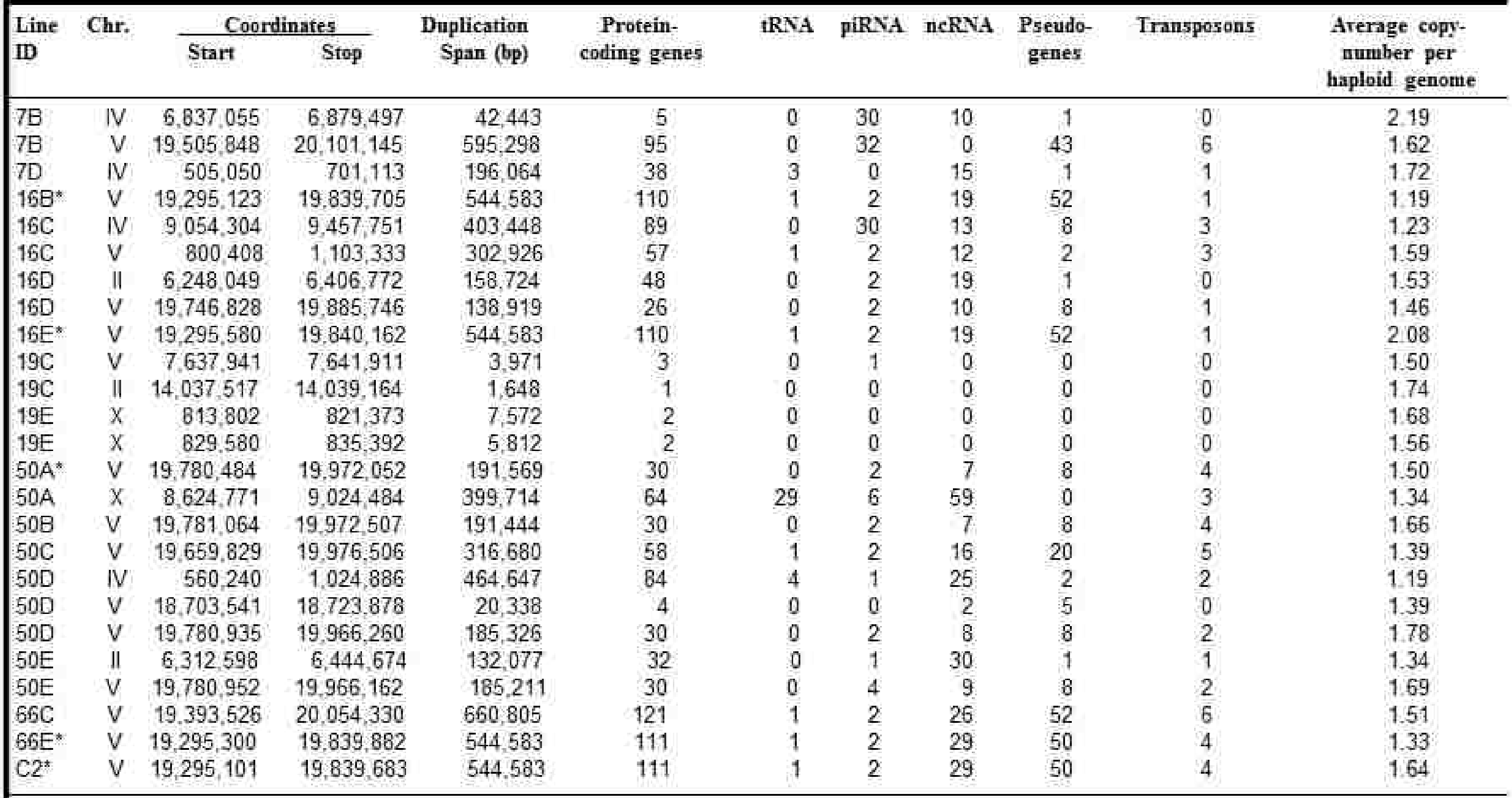
Figure 3.1. Decline in mean productivity of experimental lines during mutation accumulation with subsequent increase in productivity during population expansion. Fitness (productivity) trajectories of five experimental evolution lines of *C. elegans* during two experimental phases of (a) mutation accumulation, and (b) fitness recovery via population expansion. Two fitness assays were conducted during the mutation accumulation phase of the experiment — (i) following 24 consecutive generations of mutation accumulation with *msh-2* RNAi (MA24), and (ii) 50 consecutive generations of mutation accumulation with *msh-2* RNAi and an additional 15 additional generations of full-sib mating to promote homozygosity (MA50 + 15 Inbreeding). All five experimental lines displayed significant decline in productivity, a fitness-related trait during the MA phase, relative to the ancestral pre-MA control from which all lines were derived. Experimental lines exhibited moderate to strong fitness recovery following 150 consecutive generations of maintenance at large population sizes (RC150). Each point for the assay RC150 represents the mean productivity across five independently expanded sublines and within subline replicates (5 sublines × 5 replicates per subline). The mean productivity of the ancestral pre-mutation accumulation control has been scaled to a value of 1. Errors bars represent one standard error.

oaCGH detected 24 duplication events in 15 of the 25 experimental populations subjected to adaptive recovery via population expansion following mutation accumulation (Table 3.1). A single duplication event was identified in one of the five *fog-2* control populations (C2), which had been maintained at a large population size without having been subjected to a prior mutation accumulation phase. The duplication spans ranged from 1.6 to 660.8 kb in length, encompassing 1 to 121 protein-coding genes (Table 3.1; Supplemental Data S1, Appendix C). The median duplication span was 191.5 kb and the median number of protein-coding genes per duplication was 38. In addition, there were 18 deletions in 12 of the 25 adaptive recovery populations. An additional seven deletions were observed in the five *fog-2* control populations (one each in C1, C2 and C4; two each in C3, and C5). The length distribution of deletions was markedly different from that of duplications. The deletion spans ranged from 1.1 to 294.6 kb, resulting in the deletion of zero to 38 protein-coding genes (Table 3.2; Supplemental Data S2, Appendix C). The median deletion span was 12.5 kb and the median number of protein-coding genes deleted was one. None of these copy-number changes in the adaptive recovery phase were detected in the MA lines via (i) microarray analysis using the MA lines as the experimental lines and the common ancestor of all MA lines as a reference, (ii) qPCR, and (iii) PCR and sequencing of duplication and deletion breakpoints. Hence, they appear to have occurred and increased in frequency during the population expansion phase associated with adaptive recovery.

Duplications and deletions during adaptive recovery are significantly larger than those arising under mutation accumulation conditions

We further compared the size of CNVs originating in the adaptive recovery populations to spontaneously-occurring CNVs previously investigated in *C. elegans* lines comprising a long-term MA experiment with extreme bottlenecks of $N_e = 1$ (Lipinski *et al.* 2011). The duplication span in our adaptive recovery populations is significantly

Table 3.1. Summary of duplications in experimental *C. elegans* lines following approximately 200 consecutive generations of population expansion.



Line ID	Chr.	Coordinates		Duplication Span (bp)	Protein-coding genes	tRNA	piRNA	ncRNA	Pseudo-genes	Transposons	Average copy-number per haploid genome
		Start	Stop								
7B	IV	6,837,055	6,879,497	42,443	5	0	30	10	1	0	2.19
7B	V	19,505,848	20,101,145	595,298	95	0	32	0	43	6	1.62
7D	IV	505,050	701,113	196,064	38	3	0	15	1	1	1.72
16B*	V	19,295,123	19,839,705	544,583	110	1	2	19	52	1	1.19
16C	IV	9,054,304	9,457,751	403,448	89	0	30	13	8	3	1.23
16C	V	800,408	1,103,333	302,926	57	1	2	12	2	3	1.59
16D	II	6,248,049	6,406,772	158,724	48	0	2	19	1	0	1.53
16D	V	19,746,828	19,885,746	138,919	26	0	2	10	8	1	1.46
16E*	V	19,295,580	19,840,162	544,583	110	1	2	19	52	1	2.08
19C	V	7,637,941	7,641,911	3,971	3	0	1	0	0	0	1.50
19C	II	14,037,517	14,039,164	1,648	1	0	0	0	0	0	1.74
19E	X	813,802	821,373	7,572	2	0	0	0	0	0	1.68
19E	X	829,580	835,392	5,812	2	0	0	0	0	0	1.56
50A*	V	19,780,484	19,972,052	191,569	30	0	2	7	8	4	1.50
50A	X	8,624,771	9,024,484	399,714	64	29	6	59	0	3	1.34
50B	V	19,781,064	19,972,507	191,444	30	0	2	7	8	4	1.66
50C	V	19,659,829	19,976,506	316,680	58	1	2	16	20	5	1.39
50D	IV	560,240	1,024,886	464,647	84	4	1	25	2	2	1.19
50D	V	18,703,541	18,723,878	20,338	4	0	0	2	5	0	1.39
50D	V	19,780,935	19,966,260	185,326	30	0	2	8	8	2	1.78
50E	II	6,312,598	6,444,674	132,077	32	0	1	30	1	1	1.34
50E	V	19,780,952	19,966,162	185,211	30	0	4	9	8	2	1.69
66C	V	19,393,526	20,054,330	660,805	121	1	2	26	52	6	1.51
66E*	V	19,295,300	19,839,882	544,583	111	1	2	29	50	4	1.33
C2*	V	19,295,101	19,839,683	544,583	111	1	2	29	50	4	1.64

Line ID refers to the experimental line number. Columns 2, 3 and 4 display the chromosomal location of the CNV and the chromosomal coordinates based on WormBase version WS243. Column 5 provides estimates of the span (bp) of the duplication. Columns 6-11 represent the number of protein-coding genes, tRNAs, piRNAs, ncRNAs, pseudogenes and transposases, respectively, that are encompassed by the duplication event. Column 12 represents the copy-number of the duplicated region in the population following 180-212 generations of population expansion based on oaCGH results. Chromosomal coordinates of duplications are predicted based on oaCGH probes in all cases except for events marked by a * wherein exact duplication breakpoints were determined via direct sequencing.

Table 3.2. Summary of deletions in experimental *C. elegans* lines following approximately 200 consecutive generations of population expansion.

Line ID	Chr.	Coordinates		Deletion Span (bp)	Protein-coding genes	tRNA/ rRNA	piRNA	ncRNA	Pseudo-genes	Transposons	Average copy-number per haploid genome
		Start	Stop								
16A*	X	817,573	830,086	12,514	1	0	0	0	0	0	0.05
16D*	V	7,663,133	7,687,447	24,315	7	0	0	0	0	0	0.05
19A*	X	800,773	827,100	26,328	5	0	1	0	0	0	0.47
19C	V	7,642,395	7,682,740	40,346	10	0	0	1	0	0	0.23
19E	X	821,499	829,454	7,956	1	0	0	0	0	0	0.19
50B	V	7,650,284	7,693,435	43,152	12	0	0	0	1	0	0.76
50C	V	7,647,125	7,696,096	48,972	14	0	0	0	1	0	0.71
50C	X	1,029	273,082	272,054	35	0	0	14	18	5	0.85
50D*	V	7,653,667	7,680,465	26,799	6	0	0	0	0	0	0.18
50D	X	1,029	295,671	294,643	38	0	0	15	20	6	0.81
50E*	V	7,652,044	7,682,914	30,871	8	0	0	0	0	0	0.47
66B	V	15,258,727	15,326,180	67,454	26	0	1	2	5	1	0.62
66B*	X	9,983,441	9,999,107	15,667	2	0	0	1	0	0	0.04
66D	V	18,665,661	18,670,354	4,694	1	0	0	0	0	0	0.54
66D	V	18,701,820	18,725,404	23,585	3	0	0	3	5	0	0.39
66D	X	961,361	963,014	1,654	1	0	0	0	0	0	0.09
66D	X	7,528,608	7,529,729	1,122	1	0	0	0	0	0	0.07
66E	X	7,528,608	7,529,729	1,122	1	0	0	0	0	0	0.06
C1	I	15,060,622	15,071,438	10,817	0	4	0	4	1	0	0.75
C2	I	15,060,388	15,071,427	11,040	0	4	0	4	1	0	0.66
C3	II	14,034,460	14,039,471	5,012	1	0	0	0	0	0	0.45
C3*	X	7,527,813	7,529,236	1,424	1	0	0	0	0	0	0.05
C4	I	15,060,388	15,071,427	11,040	0	4	0	4	1	0	0.60
C5	I	15,061,973	15,071,438	9,466	0	4	0	3	0	0	0.79
C5	X	823,167	827,286	4,120	1	0	0	0	0	0	0.38

Line ID refers to the experimental line number. Columns 2, 3 and 4 display the chromosomal location of the CNV and the chromosomal coordinates based on WormBase version WS243. Column 5 provides estimates of the span (bp) of the deletion. Columns 6-11 represent the number of protein-coding genes, tRNAs/rRNAs, piRNAs, ncRNAs, pseudogenes and transposases, respectively, that are encompassed by the deletion event. Column 12 represents the copy-number of the deleted region in the population following 180-212 generations of population expansion based on oaCGH results. Chromosomal coordinates of deletion are predicted based on oaCGH probes in all cases except for events marked by a * wherein exact deletion breakpoints were determined via direct sequencing.

greater than that of previously determined spontaneous duplications under mutation accumulation conditions (Lipinski *et al.* 2011) (*Wilcoxon two-sample test*, $Z = -3.85$, $p < 0.0001$, Figure 3.2A). Duplications in populations subjected to adaptive recovery had a median duplication span of 191.5 kb versus a median span of 7.2 kb in spontaneous mutation accumulation populations (Lipinski *et al.* 2011) under the influence of genetic drift. Similarly, we detected significantly larger deletion spans in the adaptive recovery populations compared to spontaneous deletions occurring under mutation accumulation conditions (*Wilcoxon two-sample test*, $Z = -2.4$, $p = 0.016$, Figure 3.2B). The median spans of deletions in our adaptive recovery and mutation accumulation populations (Lipinski *et al.* 2011) were 12.5 and 3.5 kb, respectively.

Gradual increase in the frequencies of CNVs during the adaptive recovery phase

Based on the oaCGH arrays, the average population wide copy-number of the 24 duplications ranged from 1.19 to 2.19 copies per haploid genome (Table 3.1). Assuming that individuals harboring duplications only contain one additional copy of the duplicated segment, the frequency of individual duplications in the populations ranged from 0.19 to 1 (or fixation). The average copy-number for the deleted segments ranged from 0.81 to 0.04, suggesting that the frequency of these deletions in the populations range from 0.19 to 0.96.

In light of the oaCGH results following >200 recovery generations, qPCR was used to analyze the frequencies of duplications and deletions following approximately 80, 140 and, 208 recovery generations. In the majority of the populations, duplications and deletions that had reached high frequencies by generations 180-212 were found in

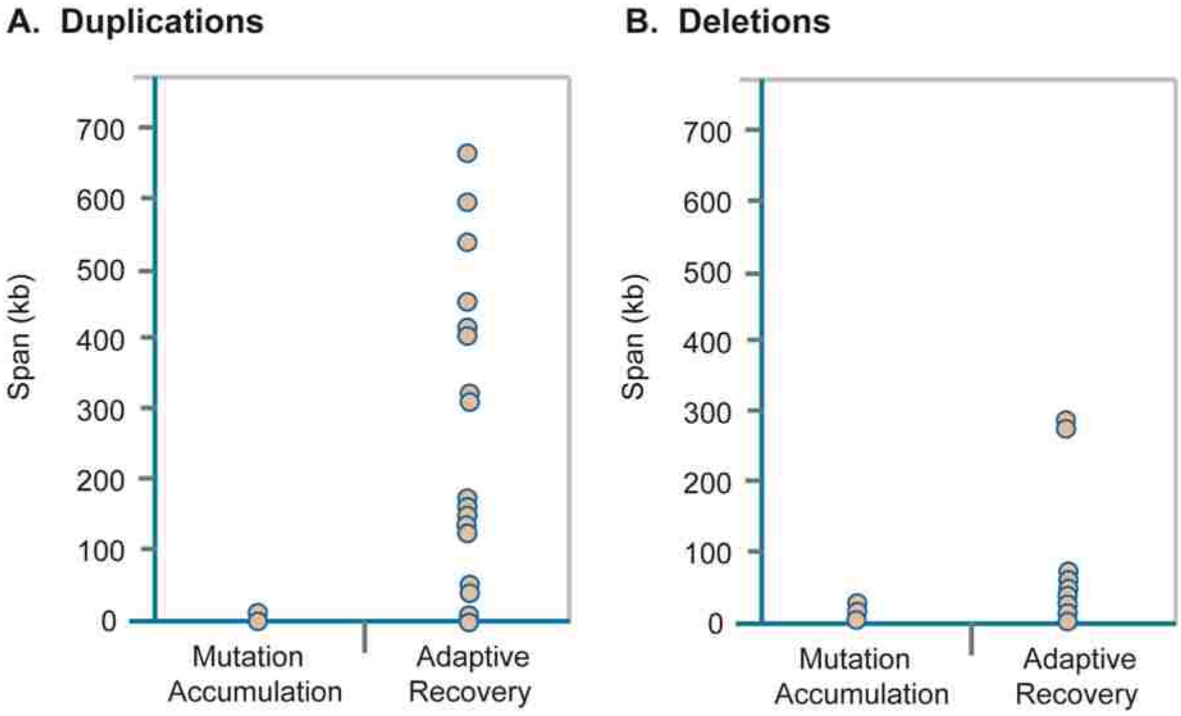


Figure 3.2. Comparison of duplication and deletion spans in adaptive recovery versus spontaneous mutation accumulation (MA) lines. A. The span of 24 independent duplication events in the adaptive recovery populations compared to the duplication span of spontaneous duplications during MA (Lipinski *et al.* 2011). The span of duplications during adaptive recovery is significantly larger than duplications detected under spontaneous MA conditions ($p < 0.0001$). **B.** The span of 18 deletion events in the adaptive recovery populations compared to the deletion span of spontaneous deletions during MA (Lipinski *et al.* 2011). The deletion span for 18 deletion events in the adaptive recovery populations was significantly greater than the span of spontaneous deletions during MA ($p = 0.032$).

intermediate frequencies at approximately 80 and 140 generations, providing evidence of a gradual increase in the frequencies of individual CNVs with time (Figures 3.3, 3.4; Supplemental Figures C.2-C.9, Appendix C). Based on the oaCGH results in Table 3.1, duplications in two populations had reached fixation by recovery generation 208 (7B:ChrIV, and 16E:ChrV). However, based on the qPCR results, three additional duplications appear to have reached fixation in their respective populations (19E:ChrX, 50B:ChrV, and 50D:ChrV). The pattern of increase in the frequency of CNVs is particularly striking in the case of several deletions (Table 3.2; Figure 3.4; Supplemental Figures C.6-C.9, Appendix C). The oaCGH results suggested that six deletions reached high frequency and that the deleted segment is only in 4-9% frequency in these populations (Table 3.2). Moreover, the qPCR results for these CNVs suggest that five deletions were already fixed by recovery generations 140-160 in these populations (Figure 3.4; Supplemental Figures C.6-C.8, Appendix C, corresponding to 16A:ChrX, 16D:ChrV, 2 deletions in 66D:ChrX, and 66E:ChrX) and one additional deletion (66B:ChrX; Supplemental Figure C.9, Appendix C) had reached fixation by recovery generation 208. In general, there was a good correlation between the oaCGH and qPCR estimates of the frequency of copy-number changes (duplications and deletions) in the populations at recovery generation 208 ($r = 0.95$, $p < 0.001$).

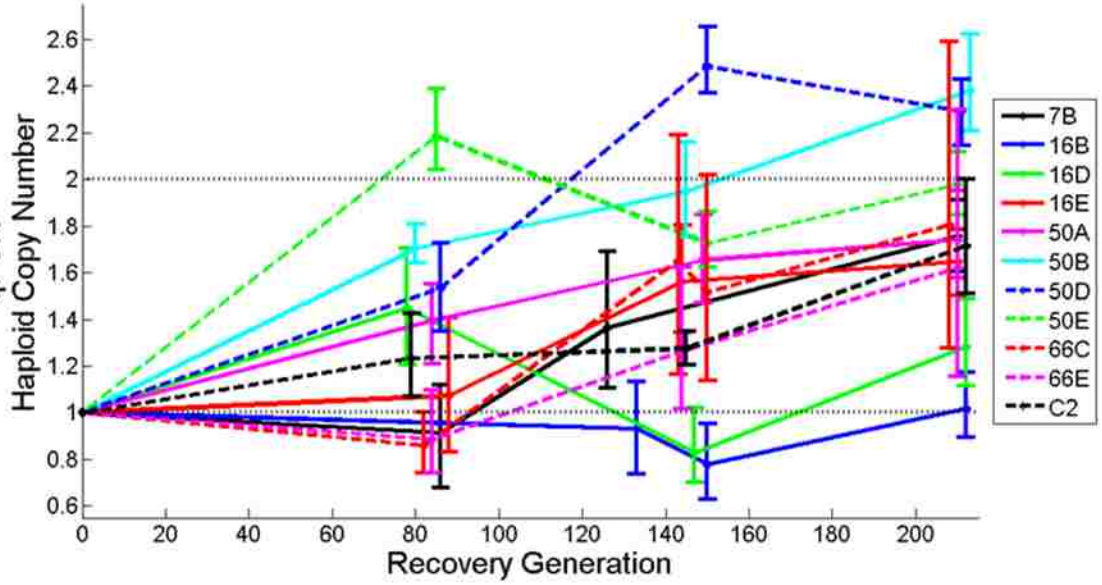


Figure 3.3. Increase in the frequency of parallel duplication events in 11 independent populations containing an overlapping region on Chromosome V. The average copy-number per haploid genome was calculated from qPCR results and is indicated on the vertical axis. The generation from which the copy-number was estimated is indicated on the horizontal axis. Error bars represent 95% CI.

Duplication and deletion breakpoints in independent populations occur within the same repetitive sequences

Our attempts to precisely map the duplication and deletion breakpoints with PCR and DNA sequencing yielded mixed results. We were able to sequence five duplication breakpoints from the set of 24 duplications in Table 3.1. In addition, we generated breakpoint sequences for seven deletion events in Table 3.2. Four duplication breakpoints on chromosome V, in populations 16B, 16E, 66E and control population C2, are located within the same 1,031 bp repeats flanking the duplications and appear to be the result of unequal crossing-over. The sequence identity between the two repeats is 96% and the point of unequal crossing-over within the repeats is different in all four cases, indicating independent events (Figure 3.5). The seven deletions with sequenced

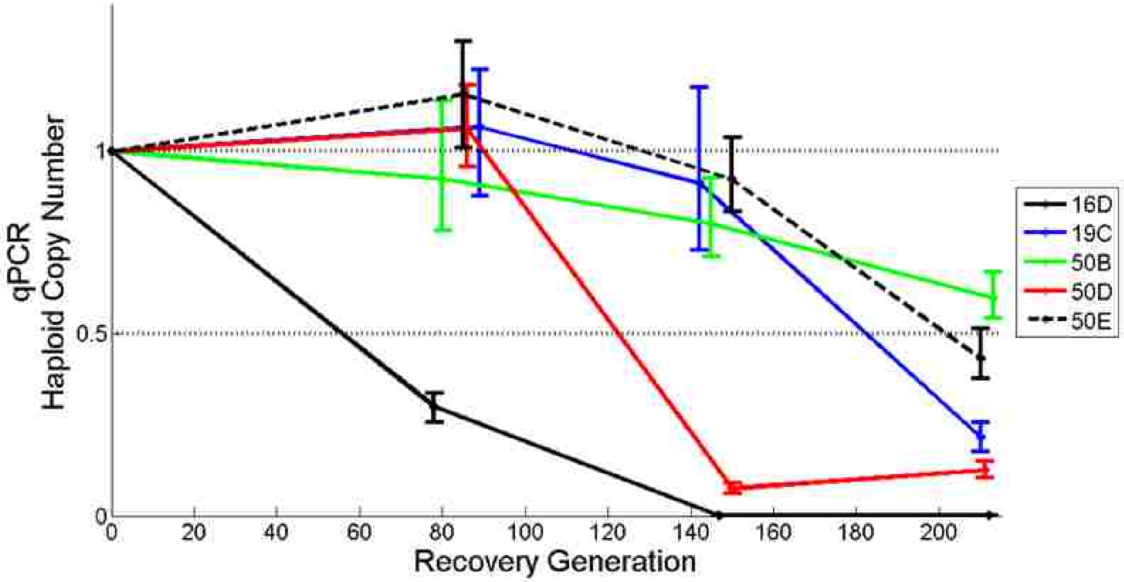
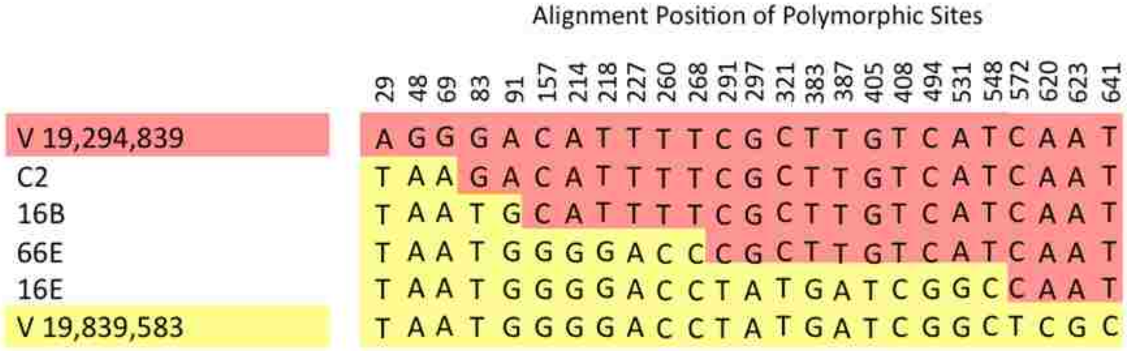


Figure 3.4. Copy-number decreases due to parallel deletion events in five adaptive recovery populations containing an overlapping region on Chromosome V. The average copy-number per haploid genome was calculated from qPCR results and is indicated on the vertical axis. The number of recovery generations is indicated on the horizontal axis. The deletions have reached fixation when the average copy-number has reached 0. Error bars represent 95% CI.

breakpoints are 16A:ChrX, 16D:ChrV, 19A:ChrX, 50D:ChrV, 50E:ChrV,66B:ChrX, and C3:ChrX (Table 3.2). These sequenced deletions do not appear to be associated with repeat motifs.

Extensive parallelism in copy-number changes of certain CNVs

Twelve duplications in 11 independent recovery populations and one control population span an overlapping region on chromosome V which extends up to ~59 kb and contains 11 protein-coding genes (Figure 3.6A; Supplemental Data S3, Appendix C). The range of duplication spans encompassing this overlapping region in the 12 populations range from ~139-661 kb. Gene Ontology (GO) annotations report the



Alignment Position of Polymorphic Sites

	29	48	69	83	91	157	214	218	227	260	268	291	297	321	383	387	405	408	494	531	548	572	620	623	641
V 19,294,839	A	G	G	G	A	C	A	T	T	T	T	C	G	C	T	T	G	T	C	A	T	C	A	A	T
C2	T	A	A	G	A	C	A	T	T	T	T	C	G	C	T	T	G	T	C	A	T	C	A	A	T
16B	T	A	A	T	G	C	A	T	T	T	T	C	G	C	T	T	G	T	C	A	T	C	A	A	T
66E	T	A	A	T	G	G	G	G	A	C	C	C	G	C	T	T	G	T	C	A	T	C	A	A	T
16E	T	A	A	T	G	G	G	G	A	C	C	T	A	T	G	A	T	C	G	G	C	C	A	A	T
V 19,839,583	T	A	A	T	G	G	G	G	A	C	C	T	A	T	G	A	T	C	G	G	C	T	C	G	C

Figure 3.5. Breakpoints of the four common duplications on chromosome V compared to their flanking repeats. Four independent populations contain a duplication of a region between positions 19,294,839 and 19,838,583 on chromosome V. These duplications are the product of unequal crossing-over between two 1,031 bp repeats that are 96% identical and flank the duplication. The figure shows polymorphic sites between the two repeats, and the nucleotides flanking the breakpoints of the four duplications. The sequences of the upstream and downstream repeats are displayed on the topmost (orange) and lowermost (yellow) rows, respectively. The sequence of the new repeat in the center of the tandem duplication is shown for strains 16B, 66E, 16E, and C2, and the correspondence to the original flanking repeats is indicated by color. The duplication breakpoint is inferred to be between the sequence that corresponds to the downstream repeat (yellow) and the upstream repeat (orange).

function of four of these 11 duplicated ORFs (*srt-45*, M162.7, Y116F11B.2, and Y116F11B.17) as unclassified with respect to biological process, cellular component and molecular function. Four of the 11 duplicated ORFs have their molecular function defined as protein-binding (*fbxa-118*, and *fbxa-194*) or carbohydrate-binding (*clec-258*, and *clec-259*). Duplicated gene *daf-28* is probably the best-characterized locus within this shared region on chromosome V. It encodes a beta-type insulin and inhibits dauer

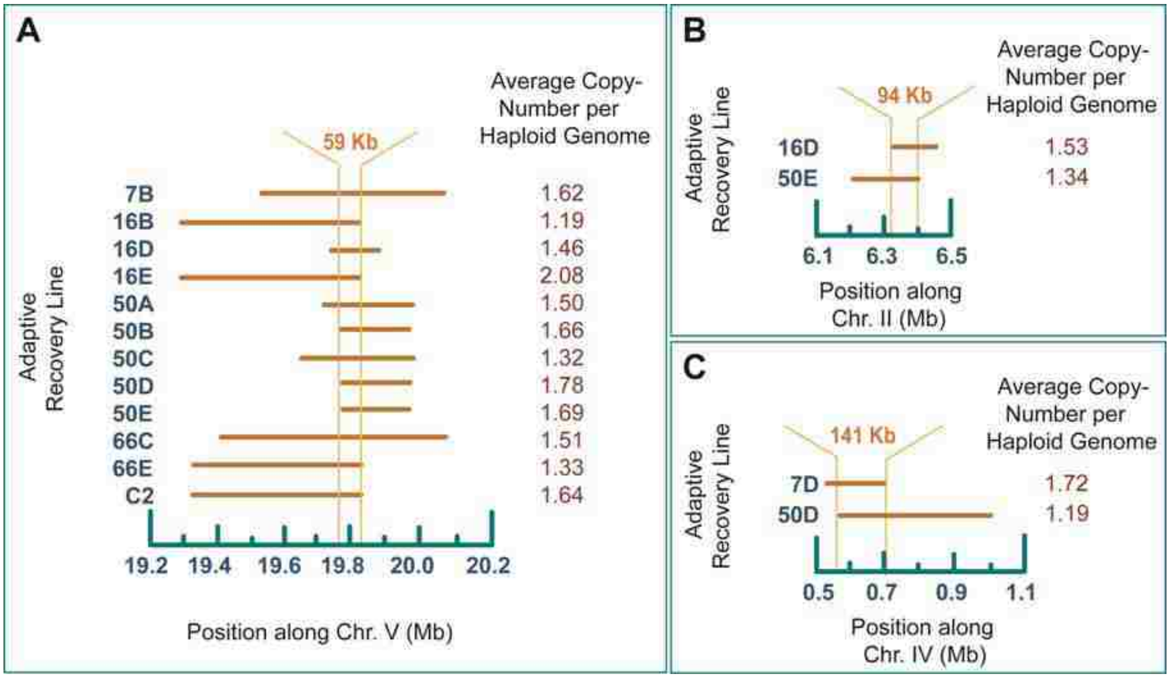


Figure 3.6. **Location and span of convergent duplication events.** The populations are indicated to the left, the chromosomal position is shown on the horizontal axis and the average haploid copy-number based on the oaCGH results from generation 208 is indicated on the right. The horizontal bars designate the regions that are duplicated in each of these populations. The vertical orange lines indicate the boundaries of the shared segment among these duplications. **A.** Overlapping duplications on chromosome V during the adaptive recovery phase of the experiment. The 59 kb region shared among all 12 populations is delineated by the vertical lines that run through the horizontal bars. **B.** Overlapping duplications on chromosome II during the adaptive recovery phase of the experiment. The 94 kb region shared among the two populations is delineated by the vertical lines that run through the horizontal bars. **C.** Overlapping duplications on chromosome IV during the adaptive recovery phase of the experiment. The 141 kb region shared among the two populations is delineated by the vertical lines that run through the horizontal bars.

formation (Li *et al.* 2003) and influences adult life-span, two potentially important life-history traits that could be under selection during the adaptive recovery regime of the experiment. *pcp-4* exhibits serine-type peptidase activity and is involved in proteolysis whereas *srw-38* codes for a protein product that serves as an integral component of membranes.

The convergent duplications on chromosome II (populations 16D and 50E), (Figure 3.6B; Supplemental Data S3, Appendix C) and chromosome IV (populations 7D and 50D), (Figure 3.6C; Supplemental Data S3, Appendix C) encompass larger overlapping regions (94 kb and 141 kb, respectively), and have lower average copy-numbers relative to the convergent duplications on chromosome V (Figure 3.6A). The convergent or overlapping duplications on Chromosome II are found in two populations and span 26 protein-coding ORFs of which 11 are unclassified with respect to biological process, cellular component and molecular function. For the remaining 15 ORFs, we note that ten ORFs (C32D5.3, *sma-6*, *set-4*, C32D5.8, *lgg-1*, C32D5.10, C32D5.12, *ani-2*, *lin-23*, and F58F12.1) have biological processes related to important life-history traits involving some combination of reproduction, dauer development, embryo development, determination of adult lifespan and oogenesis. The convergent duplications on chromosome IV occur in two populations and span 30 protein-coding ORFs of which 18 are unclassified with respect to biological process, cellular component and molecular function. Of the remaining 12 ORFs, six ORFs (*efn-4*, *gex-2*, F56A11.6, *rpl-15*, K11H12.3, and *cutl-28*) have biological processes related to the very same life-history traits observed for the overlapping duplication on chromosome II.

Additionally, we also observed five convergent deletion events that spanned overlapping regions in independent populations. Cumulatively, these five convergent deletions comprise 19 independent deletion events observed in 11 adaptive recovery populations and all five control populations. One convergent deletion in four control populations of the adaptive recovery phase (C1, C2, C4 and C5) spanned ~9.5 kb and resulted from a copy-number loss in four rDNA genes at the end of chromosome I (F31C3.7, F31C3.11, F31C3.9, and F31C3.8; Figure 3.7A; Supplemental Data S3, Appendix C). Our qPCR results suggest that the *fog-2* strain, ancestral to all of the populations in these experiments, possesses 86 copies of this repeat. In these four control populations, the number of rDNA repeats has been reduced by 21-40% (Table 3.2).

A second convergent deletion event was detected in six adaptive recovery populations (16D where it appears to have reached fixation, 19C, 50B, 50C, 50D, and 50E) and led to the loss of an overlapping 17,333 bp region on chromosome V encompassing four protein-coding ORFs (Figure 3.7B; Supplemental Data S3, Appendix C). Three of these ORFs are unclassified with respect to GO annotations. The last ORF, *Cyp-33A1* (C12D5.70), was partially deleted and is classified as a heme- and iron-ion binding protein involved in the oxidation-reduction process.

The third convergent deletion event occurred in three adaptive recovery populations (16A, 19A, 19E) and one control population (C5). This deletion entailed the loss of an overlapping 3,934 bp region partially encompassing a single protein-coding gene, *daf-3* (F25E2.5) on chromosome X (Figure 3.7C; Supplemental Data S3, Appendix C). *daf-3* is classified as an enhancer sequence-specific DNA-binding protein involved in dauer larval development among its biological processes.

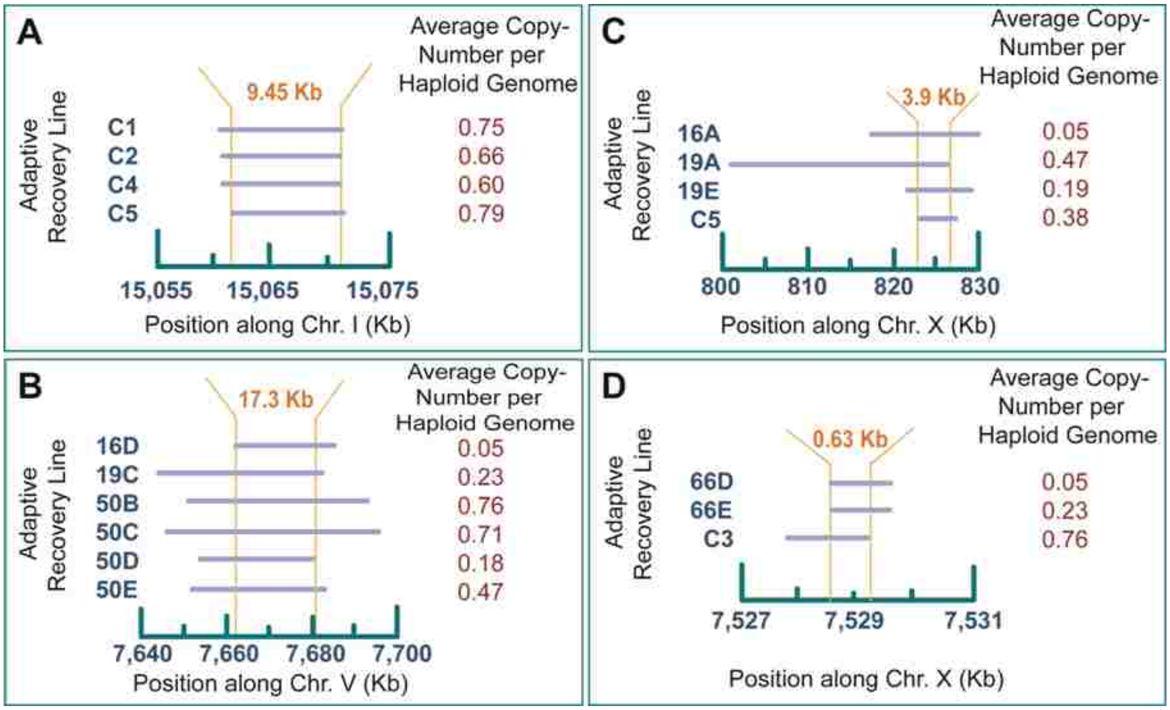


Figure 3.7. Location and span of convergent deletion events. The populations are indicated to the left, the chromosomal position is shown on the horizontal axis and the average haploid copy-number based on the oaCGH results from generation 208 is indicated on the right. The horizontal bars designate the regions that are deleted in each of these populations. The vertical orange lines indicate the boundaries of the shared segment among these deletions. **A.** Overlapping deletion on chromosome I during the adaptive recovery phase of the experiment. The ~9.5 kb region shared among four control populations (C1, C2, C4 and C5) is delineated by the vertical lines that run through the horizontal bars. **B.** Overlapping deletion on chromosome V during the adaptive recovery phase of the experiment. The 17.3 kb region shared among the six adaptive recovery populations is delineated by the vertical lines that run through the horizontal bars. **C.** Overlapping deletions on chromosome X during the adaptive recovery phase of the experiment. The 3.9 kb region shared among three adaptive recovery and one control population(s) is delineated by the vertical lines that run through the horizontal bars. **D.** Overlapping deletions on chromosome X during the adaptive recovery phase of the experiment. The 0.6 kb region shared among the two adaptive recovery and one control population(s) is delineated by the vertical lines that run through the horizontal bars.

The fourth convergent deletion event occurred in three populations (66D, 66E, C3) resulting in the loss of an overlapping 629 bp region partially encompassing a single protein-coding gene, *ceh-14* (F46C8.5) on chromosome X (Figure 3.7D; Supplemental Data S3, Appendix C). *ceh-14* is classified as a DNA- and protein-binding protein involved in the regulation of transcription and thermosensory behavior, with *ceh-14* mutants exhibiting lack of thermotaxis. In all cases, the deletion appears to have reached fixation within the populations. Although two of these deletions occurred in populations undergoing adaptive recovery following MA, one occurred in a control population that had not been subjected to MA and adaptive recovery. Interestingly, a lone deletion event in another gene on the X chromosome implicated in thermotaxis (Gomez *et al.* 2001), *ncs-1*, also reached fixation in strain 66D (Table 3.2).

Lastly, a fifth convergent deletion event occurred in two adaptive recovery populations, 50C and 50D. This deletion resulting in the loss of one end of the X chromosome reached a significant frequency in both populations. The deletion span in 50D was approximately 22 kb larger than the deletion in 50C. The average haploid copy-number of this segment was 0.85 and 0.81 in 50C and 50D, respectively, which translates into 15% and 19% of the X chromosomes bearing this segmental deletion in populations 50C and 50D, respectively. The overlapping 272 kb region in these two deletions contains 35 protein-coding genes (Supplemental Data S3, Appendix C). 20 of these 35 ORFs are unclassified with respect to GO annotations. For the remaining 15 ORFs, six ORFs (Y73B3A.18, Y73B3A.3, *elk-2*, *cad-6*, Y73B3A.10 and *set-33*) have biological processes related to important life-history and developmental traits involving some combination of reproduction, embryo development ending in birth or egg hatching, nematode larval development, hermaphrodite genitalia development and negative regulation of vulval development.

Single-worm PCR suggests simple duplications rather than higher-level amplifications

Independent estimates of CNV frequencies via single-worm PCR of CNV breakpoints confirmed the gradual increase of CNVs and are strongly correlated with the copy-number estimates from qPCR (*r* = 0.9; Table 3.3). There was one instance where the single-worm PCR results deviated significantly from the qPCR results, in line 16B following 212 generations of adaptive recovery. Both the qPCR and oaCGH data suggest that the duplication was present in low frequency in generation 212. In contrast, single-worm PCR estimated the duplication to exist at an intermediate frequency of 0.48 in the population. It is possible that some of the copy-number increases in these populations are due to a higher level of amplification (more than two copies per chromosome) than a single duplication. If the copy-number is frequently >two per haploid genome, we expect that the copy-number calculated from qPCR would systematically exceed the estimates from single-worm PCR. However, this is not the case, and the generally good agreement between the different methods suggests that higher-level amplification is not widespread for the three duplications with single-worm PCR estimates.

Discussion

In the last decade, analysis of gene copy-number variation has shown that CNVs are surprisingly widespread in natural populations. Like other classes of mutations, these variants can be beneficial, neutral or deleterious. However, gene copy-number increases are unique among mutations in that they can facilitate the evolution of novel genes. The population dynamics of gene copy-number variation in populations are therefore important for understanding both the adaptation and evolution of novel genes. In this study, we investigated whether gene copy-number changes (duplications and deletions)

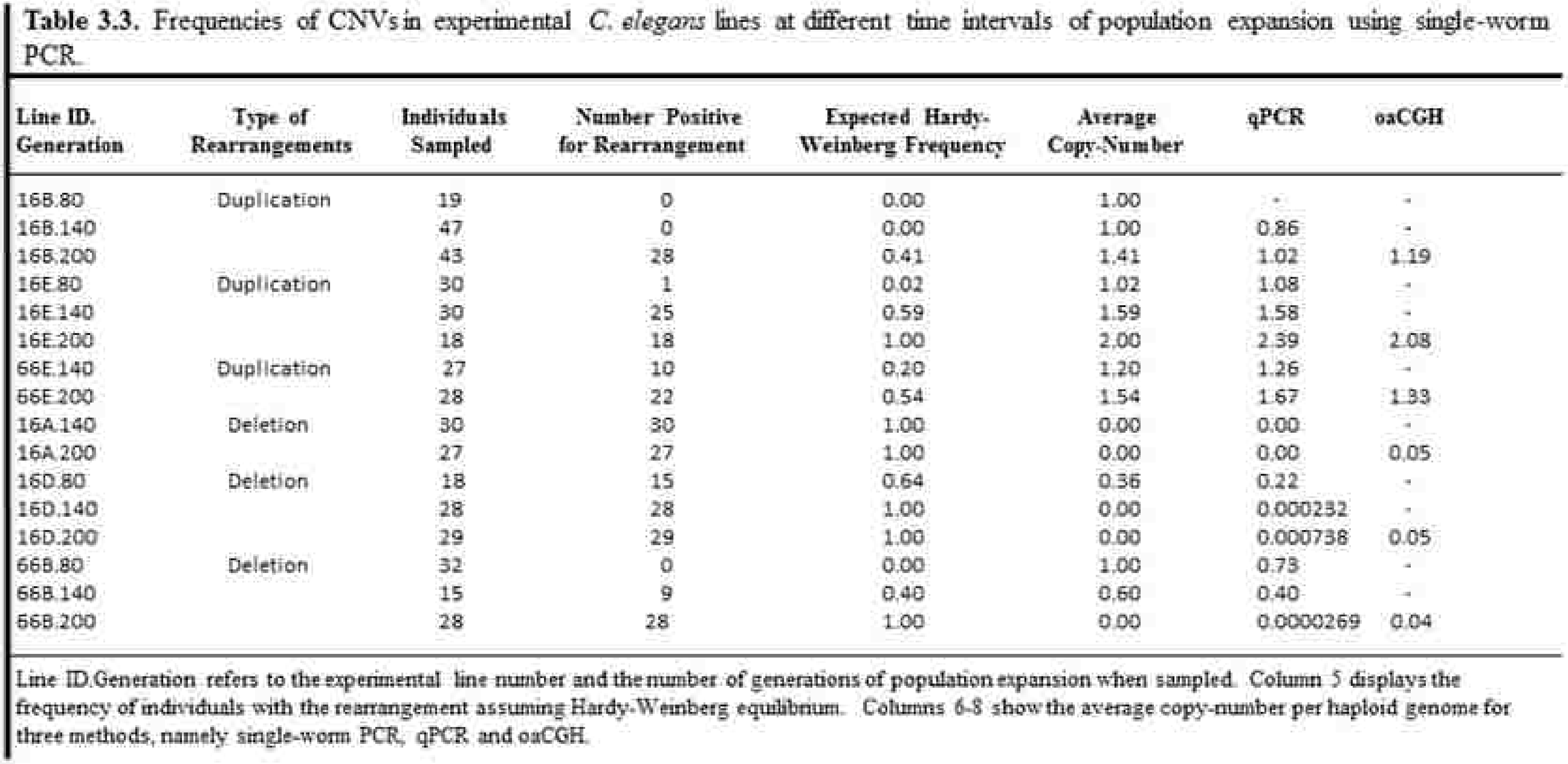


Table 3.3. Frequencies of CNVs in experimental *C. elegans* lines at different time intervals of population expansion using single-worm PCR.

Line ID. Generation	Type of Rearrangements	Individuals Sampled	Number Positive for Rearrangement	Expected Hardy-Weinberg Frequency	Average Copy-Number	qPCR	oaCGH
16B.80	Duplication	19	0	0.00	1.00	-	-
16B.140		47	0	0.00	1.00	0.86	-
16B.200		43	28	0.41	1.41	1.02	1.19
16E.80	Duplication	30	1	0.02	1.02	1.08	-
16E.140		30	25	0.59	1.59	1.58	-
16E.200		18	18	1.00	2.00	2.39	2.08
66E.140	Duplication	27	10	0.20	1.20	1.26	-
66E.200		28	22	0.54	1.54	1.67	1.33
16A.140	Deletion	30	30	1.00	0.00	0.00	-
16A.200		27	27	1.00	0.00	0.00	0.05
16D.80	Deletion	18	15	0.64	0.36	0.22	-
16D.140		28	28	1.00	0.00	0.000232	-
16D.200		29	29	1.00	0.00	0.000738	0.05
66B.80	Deletion	32	0	0.00	1.00	0.73	-
66B.140		15	9	0.40	0.60	0.40	-
66B.200		28	28	1.00	0.00	0.0000269	0.04

Line ID.Generation refers to the experimental line number and the number of generations of population expansion when sampled. Column 5 displays the frequency of individuals with the rearrangement assuming Hardy-Weinberg equilibrium. Columns 6-8 show the average copy-number per haploid genome for three methods, namely single-worm PCR, qPCR and oaCGH.

constituted a common form of genetic change during the adaptation of low-fitness experimental populations of *C. elegans*.

Several lines of evidence suggest that the high frequency of copy-number changes in the adaptive recovery and control populations are primarily due to natural selection. Both deletions and duplications increased in frequency with time, and some rearrangements had already reached fixation by 145 generations of population expansion. The theoretical expectation for the average number of generations until fixation of a neutral mutation under conditions of genetic drift is $4N_e$ generations (Kimura and Ohta 1969). Assuming a lower-bound conservative estimate of N_e = 1,000 individuals in the adaptive recovery populations each generation, neutral CNVs in our experimental populations would take, on average, more than 4,000 generations to reach fixation. Five duplications and eight deletions in our adaptive recovery and control populations originated and reached fixation within 212 generations alone. Moreover, the majority of other CNVs that had not yet reached fixation by the end of the recovery phase still exhibited a steady increase in population frequency with time. Furthermore, both duplications and deletions contained striking examples of parallelism or convergent evolution. Certain duplications and deletions contained overlapping regions, *i.e.* the same region was duplicated or deleted independently in different populations (Figures 3.6 and 3.7).

Duplications of parts of chromosome V contained the same 59 kb region in eleven independent adaptive recovery populations and one control population (Figure 3.6A). If these duplications had been experiencing selection for higher dosage, one or more of these genes could be under selection in all 12 strains. One of the best-characterized genes within this overlapping duplication was *daf-28*, a pleiotropic gene influencing several life-history traits such as adult lifespan and suppression of dauer formation. For instance, if a copy-number increase entails greater *daf-28* expression, the incidence of dauer formation may be further suppressed. In another example of convergence, *daf-3* is deleted in three independent adaptive recovery populations and one control population (Figure 3.7C). *daf-3* promotes dauer formation and the deletion is expected to suppress dauer. Hence, we have convergent duplications and deletions in 16 independent populations that are expected to reduce the incidence of dauer formation. We hypothesize that both the duplication of *daf-28* and deletion of *daf-3* may be adaptations to a predictable and frequent availability of a food source, in this case a fresh lawn of *Escherichia coli*. Other examples of convergence in these populations include the partial deletion of a gene, *ceh-14,* in three populations as detected by oaCGH (Figure 3.7D). The *ceh-14* gene contributes to thermosensing and thermotaxis in *C. elegans* (Cassata *et al.* 2000). Another gene implicated in thermotaxis, *ncs-1*, is also deleted in strain 66D (Gomez *et al.* 2001).

This form of parallel evolution is best explained by selection for increased gene dosage in the case of duplications (Maroni *et al.* 1987; Sonti and Roth 1989; Newcomb *et al.* 2005; Nair *et al.* 2008), and selection against a gene in the case of the deletions (Chan *et al.* 2010; Koskiniemi *et al.* 2012; Lee and Marx 2012). Parallel molecular evolution is frequently observed in experimental population studies, particularly in microbial systems (Bull *et al.* 1997; Bergthorsson and Ochman 1999; Riehle *et al.* 2001; Wood *et al.* 2005). In large microbial populations, the chance that the same beneficial mutation will occur in independently-evolving lineages is apparently reasonably high. Compensatory evolution experiments with hermaphroditic *C. elegans* populations have also found parallel nucleotide substitutions at two sites in two independent populations (Denver *et al.* 2010). The high frequency of parallel gene copy-number changes during the population expansion phase in this study is likely due to the high rates of spontaneous copy-number mutations in concert with natural selection (Lynch *et al.* 2008; Lipinski *et al.* 2011; Schrider *et al.* 2013). Because spontaneous gene duplications and deletions originate at rates that are orders of magnitude higher than point mutations, the probability that copy-number changes in the same genes occur in independent populations is much greater than the same point mutation occurring in independent populations. Furthermore, higher mutation rates improve the probability that new variants increase in frequency or reach fixation (Lipinski *et al.* 2011; Yamplosky and Stolzfus 2001).

There is a striking difference in the size distribution of spontaneous duplications and deletions detected in MA studies and their size distribution in these populations undergoing adaptive recovery. In a preceding *C. elegans* spontaneous mutation accumulation experiment with minimal influence of natural selection, the spontaneous duplications ranged from 1-30 kb in length, with a median duplication span of 2 kb (Lipinski *et al.* 2011). In this study of duplications and deletions in adapting *C. elegans* populations following an experimental phase of fitness decline, the size range of duplications originating in the adaptive recovery phase with population expansion was 1.6-661 kb with a median duplication span of 191.5 kb. A similar trend was observed in the case of deletions originating in the adaptive recovery phase. The spontaneous deletions originating during the mutation accumulation experiment ranged from 0.2-32 kb in length, with a median deletion span of 3.5 kb (Lipinski *et al.* 2011). During the adaptive recovery phase in this study, the size range of deletions was 1.1-295 kb and the median deletion span was ~12.5 kb. Admittedly, we are comparing the size distributions of CNVs in two different strains, the selfing laboratory strain N2 (Lipinski *et al.* 2011) and the obligately outcrossing loss-of-function *fog-2* strain in this study. However, there is no evidence or theoretical grounds to suggest that the size distribution of CNVs should be influenced by the mode of reproduction (selfing *vs.* ourcrossing) in these two different strains. The large difference in the size distribution can be explained by selection for gene dosage in the recovery populations. The larger the CNV span, the greater the chance that a gene (or several genes) under selection for altered gene dosage will be contained within the duplication or deletion. This may be a general phenomenon and we predict that recent copy-number variants that are being maintained in natural populations are, on average, larger than the average spontaneous duplication or deletion.

The appearance and increase in the frequency of gene duplications and deletions in large adaptive recovery populations is unlikely to be a direct consequence of the *msh-2* treatment during mutation accumulation. First, following the completion of the MA phase, the experimental lines were inbred for 15 additional generations in the absence of *msh-2* knockdown via RNAi, so it is unlikely that there are any residual effects of the RNAi treatment *per se*. Moreover, all the copy-number changes reported here were not detected in the post-MA ancestor and appear to have arisen during the adaptive recovery phase of the experiment.

Four of 12 populations that contained a large overlapping duplication on chromosome V (Figure 3.6A) possessed duplication breakpoints in the same 1 kb repeats (Figure 3.5). These repeats appear to be duplication hot-spots. However, this type of duplication was not detected in our previous study of the spontaneous duplication and deletion rate in the *C. elegans* genome, nor in the MA populations within this study. Although this region may experience a higher than average duplication rate, this alone does not appear to account for the high frequency of individuals possessing this duplication within these independent populations. Mutation pressure (in this case, the spontaneous rate of CNV origin) is a very weak force in changing the frequency of alleles (or CNVs) (Haldane 1932). The spontaneous duplication and deletion rates in *C. elegans* are on the order of 10^{-7}/gene/generation (Lipinski *et al.* 2011). Even after allowing for a 1,000-fold higher rate of origin of a particular duplication than the best estimate of the spontaneous gene duplication rate, only 1 of 10,000 worms would incur that particular duplication in each generation and the expected frequency of a CNV containing a particular gene would reach 2% by mutational input alone after 200 generations. Moreover, the spontaneous rate of duplication loss can be higher than the rate of origin of duplications and if we take the duplication loss rate into account, the rate of increase of a particular duplication in a population would be even slower and reach equilibrium rather than going to fixation or near fixation. Therefore, the rate of origin of CNVs alone cannot explain the observed increase in frequencies of CNVs in these populations.

Conclusions

Our results demonstrate that gene copy-number changes can be a common class of adaptive genetic change to novel challenges in multicellular eukaryotes. Although the nature of the benefit that the CNVs provide in our experiments is still unknown, we note that these changes can arise frequently and sweep rapidly through populations. Some of these copy-number changes may be compensatory, serving to ameliorate the negative fitness consequences of deleterious mutations accrued during the mutation accumulation phase of the experiment. However, we note that many of these copy-number changes in our experimental populations may represent adaptations to the experimental laboratory conditions for the following reasons: (i) the presence of copy-number changes in control populations subjected to population expansion (adaptive recovery phase) without having undergone a previous fitness decline during mutation accumulation, (ii) convergent copy-number changes shared among adaptive recovery and control populations, and (iii) convergent copy-number changes in adaptive recovery populations descended from independent mutation accumulation lines. These results demonstrate the great potential that gene copy-number changes have for both adaptation *per se* as well as the potential for adaptive duplications as raw material for novel genes.

Materials and Methods

Base strain

The MA lines in this study were created with an obligately outcrossing, loss-of-function *fog-2* mutant strain of *C. elegans*. This strain was maintained as a frozen stock prior to the experiment. The *fog-2* locus in *C. elegans* is required for the initiation of spermatogenesis in hermaphrodites (Schedl and Kimble 1988). XX individuals homozygous for *fog-2* are transformed from self-fertile hermaphrodites to females whereas XO *fog-2* mutant males are indistinguishable from wild-type males. Therefore, a homozygous *fog-2* strain is fully competent as an outcrosser but not as a self-fertilizing hermaphroditic strain. The choice of outcrossing, rather than selfing, hermaphroditic populations to test if fitness recovery lines have high rates of duplications, was based on avoiding the effects of genetic hitch-hiking to the greatest extent possible (Maynard Smith and Haigh 1974).

Creation of mutation accumulation lines by repeated bottlenecks and targeted RNAi knockdown of the mismatch repair gene msh-2

The MA experiment was initiated with a single male-female pair derived from the *fog-2(lf)* mutant line, kindly provided by the *Caenorhabditis* Genetics Center (St. Paul, MN). Four generations of single pair sib-matings were allowed from the resultant offspring to remove any freezer effects. From the F_5 descendants of the base individual pair, 74 *fog-2(lf)*MA lines were initiated using a single female and two male siblings (Supplemental Figure C.1, Appendix C). The lines were assigned identification numbers 1 through 74, respectively. The presence of two males increased the probability of mating. The remaining siblings were expanded into thousands of worms and stored frozen at -80°C for future use as a pre-MA ancestral control (Lewis and Fleming 1995). This pre-MA ancestral control served as a reference population to demonstrate potential fitness decline after MA.

The rate of spontaneous deleterious mutations in *C. elegans* is relatively low (Vassilieva *et al.* 2000; Katju *et al.* 2015), and it can take multiple years to see a significant fitness decline in the MA lines. In lieu of a spontaneous MA experiment, MA was independently accelerated in the experimental lines by simultaneously (i) bottlenecking populations, and (ii) reducing the functionality of the mismatch repair (MMR henceforth) gene *msh-2* by RNAi knockdown (Kamath *et al.* 2001). Silencing of the *msh-2* gene elevates mutation rates in the germline and somatic tissue of both sexes (Degtyareva *et al.* 2002; Tijsterman *et al.* 2002). A bacterial strain containing the feeding vector with the *msh-2* gene was obtained from Julie Ahringer at the University of Cambridge.

Each experimental line was subjected to 50 generations of MA, with bottlenecking and RNAi treatment at each generation. To ensure that mutations accumulated in the MA phase of the experiment were fixed within each line and not capable of segregation as wild-type alleles, each MA line was subjected to fifteen additional generations of full-sib mating without RNAi treatment. Treating the last MA generation as the reference population, fifteen generations of full-sib mating yields an inbreeding coefficient of 0.961 (*i.e.* 96.1% reduction in heterozygosity relative to a random-mating subpopulation with the same allele frequencies) (Falconer 1989). Thereafter, all extant MA lines were frozen at -80°C.

Population expansion of lines following mutation accumulation

After the MA phase, five MA lines with the greatest decline in fitness (MA7, 16, 19, 50, and 66) were each expanded into five populations (labeled A-E) and independently maintained at large population sizes under standard laboratory conditions (Sulston and Hodgkin 1988). To enable populations to expand to large sizes, the worms were housed on large 100×15 mm Petri dishes. Large population sizes were maintained across generations by transferring agar chunks to fresh plates with a sterilized scalpel every four days (equivalent to approximately one generation). This time period was adequate to ensure highly competitive conditions, as population sizes had reached several thousands of individuals prior to each transfer, with the animals being starved to the extent that egg-laying had ceased. To avoid cross-contamination between independent populations, petri plates were spaced apart on fiberglass trays and wrapped in parafilm. Populations were continually maintained at large population sizes for 180-212 generations (Supplemental Figure C.1B, Appendix C). These large-population treatment adaptive recovery (RC) populations were frozen at -80°C following ~80, ~130, ~ 180, and ~212 generations of large population treatment. For comparison, five control populations (C1 – C5) of *fog-2* were maintained at large population sizes for 208 generations without any prior MA treatment.

Fitness Assays During Mutation Accumulation and Population Expansion

During the MA phase, one fitness assay was conducted after 24 MA generations and the second after the termination of the MA phase (50 MA generations and 15 subsequent generations of full-sib mating without RNAi treatment). The fitness assay largely followed previous protocols for hermaphroditic MA lines (Vassilieva *et al.* 2000) with minor modifications suited to outcrossing lines. The assays were conducted simultaneously on all extant MA lines, 25 adaptive recovery (RC) populations and five control populations (C1-C5) that had not been subjected to MA, but had been maintained at large populations sizes for the same period as the RC populations. The ancestral *fog-2* pre-MA ancestral population maintained as a frozen stock prior to the initiation of the MA experiment served as the control. The frozen ancestral control was thawed and 20 control lines were established independently from the surviving worms.

For fitness assays during the MA phase, a single sib-pair from each extant line was randomly chosen to enter the fitness assay. At the start of each assay, the 20 control and extant MA lines were expanded into five replicates (five individual sib-pair progeny of the ancestral pair), yielding 450 lines across both treatments. These 450 lines were maintained by transferring a sib-pair for two generations in the absence of RNAi to remove maternal effects. Additionally, because gene inactivation by RNAi does not appear to extend beyond the F1 generation (Fire *et al.* 1998), any decline in fitness in the MA lines should reflect mutation load due to heritable, germline mutations accumulated under the *msh-2* RNAi regime. Nonheritable, somatic mutations should not contribute to fitness decline once *msh-2* function is restored by RNAi termination, as these should not be inherited by the assayed individuals.

Productivity (the number of offspring produced) was measured using third generation individuals of the replicated control and experimental (MA, RC or C) populations. For each line, twelve L1 (first larval stage) F_3 progeny were randomly selected upon hatching. After 36 hours, surviving individuals had reached the L3-L4 larval stage at which they could be sexed. One male-female pair was randomly selected and transferred to a new petri dish for measuring productivity. Every 24 hr ± 30 min thereafter, the focal sib-pair is transferred to a fresh plate. Daily transfers were terminated under the following conditions: (i) the female had not produced any eggs by day 8, or (ii) female mortality. Plates with eggs were placed at 20°C for an additional 24hr period to enable hatching, then stored at 4°C to kill the larvae for progeny counts. In order to score the number of offspring, the plates with dead progeny were stained with 0.0175% Toluidine Blue to enable visualization of worms against the media. Productivity was calculated as the total number of progeny produced. The procedure was the same for the assay of adaptive RC and control (C1-C5) populations except that a random male-female pair was selected from each recovery population and control population to enter the fitness assay.

Detection of CNVs via oligonucleotide array Comparative Genome Hybridization (oaCGH)

We analyzed copy-number changes in five MA lines (MA7, MA16, MA19, MA50 and MA66), 25 adaptive recovery populations (7A-E, 16A-E, 19A-E, 50A-E, 66A-E), and five additional control populations (C1-C5) that were propagated for the same period as the adaptive recovery populations but had not undergone a prior MA phase. In the microarray experiments, the MA lines and the C1-C5 populations were compared to their *fog-2* ancestor, and the adaptive recovery populations were compared to their post-MA ancestor (50 generations of MA and 15 generations of inbreeding). For example, copy number changes in recovery populations 7A-E were compared to MA7 after termination of the MA phase of the experiment. oaCGH analysis was performed as previously described (Maydan *et al.* 2007). We used oaCGH arrays manufactured by Roche NimbleGen Inc.: design 071114_CE2_WG_CGH_T, and new custom designed microarrays named 120618_Cele_WS230_JK_CGH. The new arrays are 3-plex microarrays with each individual sub-array comprising 720k 50-mer oligonucleotide probes synthesized at random positions on the arrays. The filters used to select the probes primarily followed Maydan *et al.* (2007) without focusing on coding regions in order to provide a more uniform coverage of the genome (Wormbase release WS230). In regions where unique probes could not be designed, selection filters were slightly relaxed in order to allow the inclusion of probes with possible cross-hybridization to at most one other location in the genome. The extraction of fluorescence intensity ratios and subsequent segmentation analysis followed Maydan *et al.* (2007) closely except that a quantile normalization was applied on the $\log_2$ ratios. The segmentation algorithm used a bottom-up approach, adjacent segments being merged until no neighboring segments reach a user-defined similarity threshold, the similarity being calculated with a *t*-test. At the end of the segmentation procedure each remaining segment was analyzed and labeled as amplified/deleted if the $\log_2$ ratio values within the segment passed two user-defined filters, one for the average and one for the *p*-value (calculated with a *t*-test). Visual inspection of the $\log_2$ ratios was used to guide the selection of the three user-defined parameters applied to the automated segmentation procedure. Additional analyses were performed with JCFread_cgh (Matlab script), and SnoopCGH (Almagro-Garcia *et al.* 2009).

The minimum length of these CNVs was calculated based on the distance between the first and last probe inside the region that had been duplicated or deleted. The breakpoint of the CNVs is expected to be located between the first or last internal probe and the adjacent flanking probe. However, in some cases the distance between the adjacent flanking probes and the probes contained in the CNV was fairly large, up to 40 kb, resulting in uncertainty about the location of the breakpoints.

Additionally, we used (i) qPCR, (ii) PCR and DNA sequencing of breakpoints, and (iii) single-worm PCR to independently verify the presence of CNVs identified by oaCGH as well as quantify the frequency of the CNVs in earlier generations of the adaptive recovery phase.

Quantitative PCR (qPCR)

We used qPCR as a means to independently verify the presence of CNVs identified by oaCGH as well as quantify the frequency of the CNVs in earlier generations of the adaptive recovery phase. The qPCR was performed and analyzed as described previously (Lipinski *et al.* 2011). Briefly, qPCR was performed using FastStart SYBR Green with Rox (Roche) and the reactions were run on an ABI Prism 7000 Sequence Detection System. qPCR was done by testing population DNA of specified generations against their post-MA, pre-adaptive recovery ancestor.

A modification of the ΔΔCt method (Ferreira *et al.* 2006) was used for measurement of copy-number changes in genomic DNA from populations. The efficiency of the reference was determined by a dilution series for each qPCR plate. Each "run" was comprised of four groups of three unpaired technical replicates, one group for each combination of template and primers (reference DNA with reference primers (R/R'), reference DNA with test primers (R/T'), test DNA with reference primers (T/R') and test DNA with test primers (T/T')), resulting in 12 cycle threshold measurements (Cts) per run. The average of each group was used to calculate copy-number. The mean copy-number was determined from $(1\text{+efficiency})^{-\Delta\Delta Ct}$ where ΔΔCt = (T/T' – T/R') – (R/T' – R/R') (Pfaffl 2001). Statistical analysis was performed as recommended by MIQE standards (Bustin *et al.* 2009). 95% confidence intervals for the mean copy-numbers were determined through bootstrapping (10,000 iterations) by random resampling of individual Ct values within each group to produce an array of sorted copy-numbers. The confidence interval bounds were the 2.5 and 97.5% quantiles of the sorted bootstrap array.

PCR and DNA sequencing across duplication and deletion breakpoints

For PCR and sequencing duplication breakpoints, we designed primers oriented in opposite directions within the predicted boundaries of the duplication event. In genomes bearing only a single gene-copy, the forward and reverse primers are divergent and would fail to initiate PCR amplification. However, in the event of gene duplication resulting in two adjacent paralogs (tandem or inverted), the primers are rendered convergent, enabling PCR amplification and subsequent DNA sequencing. For deletions, primers were designed to DNA sequences flanking the deleted sequence. This approach would fail to detect gene duplications and deletions with additional local rearrangements or those that have been rendered genomically distant via translocations. The PCR products were either gel-extracted and cleaned up using QIAquick Gel Extraction Kit (Qiagen) or prepared directly for sequencing using ExoSAP-IT (GE HealthCare Life Sciences). The PCR products were subsequently sequenced using Big Dye Terminator v3.1 Cycle Sequencing Kits (AB Applied Biosystems) on an ABI 3130xl Genetic Analyzer.

Single-Worm PCR

Single-worm PCR was additionally performed to confirm the accuracy of both the oaCGH and qPCR methods in estimating the frequency of existing deletions and duplications. Because adaptive recovery populations were cryogenically frozen at multiple time-intervals approximating generations 80, 140, and 200, it was possible to resurrect *C. elegans* populations at different generation times and collect individual worms from the thawed populations. Populations at varying generation times were removed from -86°C and thawed on regular NGM plates. Upon reaching maturity, worms were sexed and adult males were collected in lysis buffer and frozen in individual PCR tubes at -86°C. It was necessary to use adult males because outcrossing adult females may contain nonclonal eggs; hence a PCR band of DNA extracted from a mother and her eggs would not be an accurate representation of the genotype of an individual worm. Using primers designed to detect duplications and deletions, PCR was performed on 30 individual worms, when possible, using the single-worm PCR protocol developed by Williams *et al.* (1992). Frozen males were thawed and incubated at 65°C for 90 min, followed by incubation at 95°C for 15 min to deactivate proteinase K. After worms were lysed and DNA released from cells, PCR tubes were spun down to separate worm protein from solution. The DNA solution was removed from the tubes and divided between two PCR tubes, 2.5µl per tube.

We obtained single-worm PCR data at varying generation times for rearrangements for which duplication/deletion breakpoints had previously been sequenced. On average, 30 individuals for each population at each time-point were analyzed. To test the frequency of a deletion in a population, two separate reactions were prepared, (i) namely using deletion primers external to the deleted sequence, and (ii) primers internal to the deleted sequence. A positive result for the reaction containing the internal primers was evidence that the deletion was not present in the genome of the individual. A positive result for the reaction with primers external to the deleted sequence was evidence that the deletion had occurred in the genome of the individual. The presence of both deletion single worm PCR products indicated an individual that was heterozygous for the deletion of interest. To estimate the frequency of duplication in a population, two reactions were prepared for each individual. One reaction was prepared with divergent primers designed from sequencing the breakpoints of the duplication in question and yields a product of a known size when the duplication is present, and the second reaction contained positive control primers. All reactions were run with a touchdown thermocycling protocol with the following profile: 10 cycles of 30s @ 94°C, 30s @ 60°C – 1°C/cycle, and 2' @ 72°C followed by 30 cycles of 30s @ 94°C, 30s @ 50°C, and 2' @ 72°C. The products were analyzed by gel electrophoresis.

If the rearrangement resides on chromosome X, then the frequency of individuals showing a positive PCR result for the rearrangement should be a direct estimate of the frequency in the population since males are hemizygous for the X chromosome. If the rearrangement was present on any of the remaining five autosomes (I-V), the frequency of rearrangements was calculated under the assumption that the population was in Hardy-Weinberg equilibrium. The frequency of individuals that test negative for the rearrangement is therefore expected to be the frequency of individuals homozygous for the absence of the rearrangement (non-carriers). The frequency of individuals positive for the rearrangement is the frequency of individuals that are homozygous or heterozygous for the rearrangement. The frequency of the rearrangement is then estimated as 1 – square root of the frequency of non-carriers.

Data Access

The microarray data have been deposited in NCBI's Gene Expression Omnibus (Edgar *et al.* 2002) and are accessible through GEO Series accession number GSE67871.

Acknowledgements

This work was supported by a National Science Foundation grant DEB-0952342 to UB and VK and a Canadian Institutes of Health Research Grant CIHR MOP-97921 to DGM. DM is a Senior Fellow of the Canadian Institute for Advanced Research. The *fog-2* mutant strain used in this work was provided by the Caenorhabditis Genetics Center, which is funded by the NIH National Center for Research Resources (NCRR). We thank Gregory Campbell, Hallie Rane, and Alia Saad for technical assistance.

Chapter 3 Addendum

The oaCGH array data obtained during this investigation exhibited much less variance than the oaCGH array data from our previous research (Lipinski *et al.* 2011). This made it easier to identify CNVs and their breakpoints. The increased probe density also facilitated better breakpoint identification. There was a distinct difference between the type of CNV signal displayed in the two projects. In the previous work, CNVs were fixed in the population due to the bottlenecking process. This meant that the oaCGH array signal showed discrete levels of change reflecting the copy-number shift per haploid genome. In this work, because of the large populations where the CNV frequency in the population can fluctuate, the array signal could take on any intermediate value reflecting the CNV frequency.

Of the five duplications for which we sequenced breakpoints, the four common chromosome V duplications (Figure 3.5) all appear to be the result of NAHR, as mentioned in Chapter 2. In all four of these duplications, the recombination mechanism produced clean transitions between the sequences homologous to the downstream and upstream ends, respectively. There were no nucleotides added, nor deleted (no gaps), at the transition point. This provides more evidence for the hypothesis that crossover events, even if they occur in the middle of an exon, will not themselves disrupt the gene.

Summarizing, multiple experimental lines with reduced fitness were able to regain most or all of their fitness during recovery in large populations. CNVs arose in these populations, increasing quickly in frequency over a relatively small number of generations, suggesting the CNVs were adaptive and contributed to the fitness increase. Adaptive CNVs, as opposed to CNVs that arose in the absence of selection, are markedly larger, including many more genes per duplication event thereby increasing the chances of a favorable duplication. While there may be a question about this also increasing the probability of unfavorable duplications, it does appear that large duplications in this study, in many cases, were clearly adaptive under the experimental conditions.

Chapter 4

Statistical Methods for Technical Replicates and Combined Data in Quantitative PCR

James C. Farslow[1]

[1]Department of Biology, University of New Mexico, Albuquerque, NM 87131, USA

Manuscript is in review with *Biotechniques*, and is edited to fit the dissertation format.

Keywords: Quantitative PCR statistics; real-time PCR statistics; technical replicates.

Abstract

Effective statistical methods for analyzing technical replicates and combined data in quantitative, or real time, PCR (qPCR) are still lacking in the literature. In this analysis, computer simulations were used to analyze four bootstrap methods for technical replicates and calculated methods for both technical replicates and combined data. These simulations revealed that the calculated methods presented here had a confidence interval capture rate of approximately 95% regardless of the ddCt value or standard deviation of the source data. The bootstrap methods displayed various behaviors. The two methods that emulated the actual methods for determining ddCt tended toward a 95% capture rate with large sample sizes ($N_{Ct} > 25$), but tended to type I errors with smaller sample sizes. The other two bootstrap methods had confidence interval capture rates between 90.0% and 96.9% for small sample sizes ($N_{Ct} = 3$), but quickly tended to type II errors as sample size increased. These results suggest calculated methods work effectively provided the

correct confidence intervals are determined, and bootstrap methods may work for certain data sets but should be evaluated *in silico* to determine their effectiveness.

Methods Summary

This work presents an analytical method for determining the confidence interval for technical replicates in quantitative PCR by using a standard deviation derived from the standard deviation of the four groups of Ct values. Additionally, a method for determining the confidence interval for combined data (technical and biological replicates) is presented.

Introduction

Quantitative, or real-time, PCR (qPCR) has become a valuable tool for the measurement of relative quantities of DNA and/or RNA. While the calculations to determine *N*-fold copy change between experimental and control samples have been extensively discussed (Pfaffl 2001; Livak and Schmittgen 2001), the appropriate statistics to perform hypothesis testing are not so straightforward. Researchers often use biological replicates, and while the statistical methods for these are straightforward, there is some disagreement on the statistic to be analyzed (Yuan *et al.* 2006; Schmittgen and Livak 2008). Statistical methods for technical replicates of single biological samples (replicate tests on the same sample), or combinations of technical replicates and biological replicates (combined data), have been all but ignored in the literature. Specifically, qPCR methods that use technical replicates cannot be analyzed using biological replicate statistical methods because the measurements of technical replicates are unpaired, creating a difference in determining the standard error of the mean.

Bootstrap procedures may provide a solution to this problem, but these may be inadequate if sample sizes are small, and may need to be modified to provide an effective means of hypothesis testing. This work compares methods for calculating 95% confidence intervals (CI) for technical replicate and combined qPCR data as well as four different bootstrap algorithms. The metric of comparison in this analysis is the rate at which the true ddCt value, determined from the means of the cycle threshold value (Ct) source distributions used for random sampling, is contained within (or captured by) the CI for that method. With a 95% CI, the capture rate should be approximately 95% on average with a large number of trials if that method is optimal (Ott 1993; Samuels *et al.* 2012). The question is which methods, and for what sample sizes, provide a CI capture rate close to 95%? It is predicted that among bootstrap methods, those emulating the process of ddCt calculation will perform best, especially with large numbers of samples.

Methods

The simulations were Matlab scripts (see Appendix D) developed to measure the frequency at which the true ddCt value was contained within a method's CI, which will be referred to as the capture rate. The ddCt is the cycle difference measure between the experimental and reference samples. The programs performed 1000 trials per sample size (N_{Ct}) for each level of standard deviation of the source population (σ) and each target value of ddCt specified in the simulation for a total of 15,000 trials for each sample size except N_{Ct} = 3 and 25 which had 25,000 trials each. 10,000 bootstraps were done for each bootstrap algorithm in each trial. All methods were evaluated based on a 95% CI.

The Data

The programs pseudorandomly generated a set of simulated Ct values from normal distributions with specified means and standard deviations. Analysis of actual qPCR Ct values (Lipinski *et al.* 2011) from technical replicates showed the Ct distribution was not significantly different from normal with a sample size of $N = 72$ (*Jarque-Berra normality test* $p = 0.1790$; *Lilliefors normality test* $p = 0.4129$). Simulations were run with source population σ of 0.05, 0.15, 0.25, 0.35 and 0.45, and target ddCt values of −4, 0, and 4. The ddCt value was the statistic of interest (Yuan *et al.* 2006) and is a linear measure that maps via negative exponential transformation (base 2 if PCR efficiency is 100%) to the relative *N*-fold copy change or relative expression level. The ddCt is derived as the difference of Ct measures reported by the instrument. The program created four groups of measurements all containing the same number of Ct values specified for that simulation designed to generate a distribution of ddCt values. The true ddCt value was calculated from the four source distribution means using the standard equation:

$$\text{ddCt} = (T_{T'}\text{-}T_{R'})\text{-}(R_{T'}\text{-}R_{R'}), \qquad \text{(eq. 1)}$$

where *T* and *R* were the test and reference DNA samples respectively, and *T'* and *R'* were the target and reference primers respectively (Pfaffl 2001). The four combinations of the DNA and primers represented the four groups within which Ct measurements were made. Three of the means remained fixed for all simulations, while the fourth mean could be shifted for comparison of the results under different target ddCt values. Each trial iteration, all of the bootstrap and calculated methods, except the combined data method, were performed on the same set of simulated data. Sample sizes analyzed were: N_{Ct} = 3–10, 15, 20, 25, 30, 35, 40, 45, and 50.

The Bootstrap Methods

There were four bootstrap algorithms compared in this study:

Group Means Method: This method emulated the process for determining technical replicate ddCt, which assumed the data was unpaired. The normal process takes the mean of the Ct measurements of each of the four groups and plugs them into equation 1 to determine the ddCt value (Ferreira *et al.* 2006). Each bootstrap iteration, each of the four groups was resampled with replacement to form four new groups, which were then averaged and the four new means used to calculate a new ddCt value.

Paired Means Method: This method treated the data as if it were paired between the tests within each DNA type (test or reference), as in biological replicates. This produced two sets of dCt values, $\mathrm{dCt_T} = T_{T'}\text{-}T_{R'}$ and $\mathrm{dCt_R} = R_{T'}\text{-}R_{R'}$, which were then resampled with replacement within each set and averaged. These two new means were then used to determine ddCt using the equation $\mathrm{ddCt} = \mathrm{dCt_T} - \mathrm{dCt_R}$ during each bootstrap iteration. The reason for including this process, in spite of the data being generated in an unpaired manner, is that most methods described in the qPCR literature deal only with paired data and it was desired to look at how those methods handle unpaired data.

Single Random Resampling: This method randomly selected one of the Ct values from each of the four groups and used those single values instead of the means to calculate ddCt as described above during each bootstrap iteration. This method did not emulate the normal process for calculating ddCt. The range of possible ddCt values was the same as the group means method, but the distribution was less kurtotic thus increasing the CI by forcing its limits outward.

Single Paired Resampling: This method paired the data as in the paired means method above, but then randomly selected one dCt value from each set and used these single values to calculate ddCt during each bootstrap iteration. This method did not emulate the normal process for calculating ddCt.

The Technical Replicate Calculated Method

This method calculated a CI based on a standard error of the mean ($\mathrm{SE}_{\mathrm{mean}}$) determined from the standard deviation and number of Ct values within each group and assumed the data was unpaired. The $\mathrm{SE}_{\mathrm{mean}}$ for each group was determined by $\mathrm{SE}_{\mathrm{mean}} = s/\sqrt{n}$, where *s* = standard deviation of each group's measurements and *n* = number of Ct values within each group (Ott 1993; Samuels *et al.* 2012). The $\mathrm{SE}_{\mathrm{mean}}$ for the ddCt ($\mathrm{SE}_{\mathrm{ddCt}}$), which is the SE of the difference of the means of the groups, should be calculated as:

$$\mathrm{SE}_{\mathrm{dCtT}} = \mathrm{SE}_{T_{T'} - T_{R'}} = \sqrt{\mathrm{SE}^2_{T_{T'}} + \mathrm{SE}^2_{T_{R'}}}$$

$$\mathrm{SE}_{\mathrm{dCtR}} = \mathrm{SE}_{R_{T'} - R_{R'}} = \sqrt{\mathrm{SE}^2_{R_{T'}} + \mathrm{SE}^2_{R_{R'}}}$$

$$\mathrm{SE}_{\mathrm{ddCt}} = \mathrm{SE}_{\mathrm{dCtT-dCtR}} = \sqrt{\mathrm{SE}^2_{\mathrm{dCtT}} + \mathrm{SE}^2_{\mathrm{dCtR}}} \qquad \text{(eq. 2)}$$

(Samuels *et al.* 2012) where $\mathrm{SE}_{T_{T'}}$and $\mathrm{SE}_{T_{R'}}$= $\mathrm{SE}_{\mathrm{mean}}$ of Ct measures from test (or experimental) DNA with test and reference (or control) primers, respectively, $\mathrm{SE}_{R_{T'}}$and

$\mathrm{SE}_{R_{R'}}$= $\mathrm{SE}_{\mathrm{mean}}$ of Ct measures from reference DNA with test and reference primers, respectively, and $\mathrm{SE}_{\mathrm{dCtT}}$ and $\mathrm{SE}_{\mathrm{dCtR}}$ = $\mathrm{SE}_{\mathrm{mean}}$ of the difference between the mean Ct results of the test DNA and reference DNA, respectively. Note that the $\mathrm{SE}_{\mathrm{ddCt}}$ was calculated from the standard deviations of the Ct measurements, and not from σ of the source distributions. The degrees of freedom equals the total number of Ct values in all groups minus one for each group, df = $\sum(N_i - 1)$, where N is the number of Ct values in the ith group (Ott 1993). The number of Ct values were equal for all groups in these simulations, and there were four groups, thus df = $4(N_{Ct} - 1)$.

Additionally, two tests were made, at sample sizes of N_{Ct} = 3 and 25 with 15,000 trials each, pairing the technical replicates in the order they were generated to see if there was any difference in the capture rate. After pairing, the sets were treated as biological replicates and the standard deviation, $\mathrm{SE}_{\mathrm{mean}}$, and CI were determined from the resulting sets of ddCt values. The CIs of each iteration were tested for true mean capture.

The Combined Data Calculated Method

This final method was run independently of the other simulations as the arrangement of the data was different. First, data was generated as above to produce technical replicates within each of three biological replicates. The ddCt value for each biological replicate was determined as described above for technical replicates. Then, the biological replicate ddCts were averaged, the standard deviation and $\mathrm{SE}_{\mathrm{mean}}$ determined, and the CI estimated using standard statistical methods (Ott 1993; Samuels *et al.* 2012). 1000 trials per combination of conditions were performed using the same σ and target ddCt values, for a total of 15000 trials per sample size.

Results and Discussion

Various statistics have been reported with qPCR results including standard deviation and coefficient of variation (CV) (Schmittgen and Livak 2008). However, CV is not useful for hypothesis testing, and standard deviation needs to be adjusted for N_{Ct} to get the SE_{mean}. Additionally, the standard deviation of paired biological replicate data is calculated differently than the standard deviation of unpaired technical replicate data. Accurate determination of the standard deviation is paramount to accurate estimation of the SE_{mean}. On top of this is the necessity of an accurate determination of the degrees of freedom required to obtain the appropriate *t*-critical value for the estimation of CIs. Bootstrap methods circumvent the need for estimating standard deviation and degrees of freedom, but present some difficulties in the estimated capture rates of their CIs depending on N_{Ct} and the algorithm employed.

To be effective, methods should be invariant with respect to ddCt and σ. Table 4.1 shows a comparison of the results across different ddCt values and σ of the source distributions, and supports the hypothesis that the CI capture rate of all methods considered here is not correlated with ddCt or σ (all two-way ANOVA $p > 0.05$, thus fail to reject null hypothesis of no effect).

For biological replicates, the final ddCt of each paired set can be determined from the standard equation 1 above, and the mean ddCt, SE_{ddCt}, and the 95% CI calculated using standard statistical methods (Ott 1993; Samuels *et al.* 2012). The negative of the mean ddCt is then exponentiated ($2^{-\text{ddCt}}$, at 100% PCR efficiency) to provide an estimate for the *N*-fold change (expression or copy number) relative to the reference (Pfaffl 2001).

Table 4.1. Two-way ANOVA Results of the Comparison of Capture Rate Versus Population ddCt (-4, 0, 4) and σ (0.05 to 0.45).

Method	N_{Ct}	*p*-values ddCt	σ
Group Means Bootstrap	3	0.8215	0.7694
	25	0.9704	0.6634
Paired Means Bootstrap	3	0.6812	0.6119
	25	0.7096	0.4099
Single Random Resampling Bootstrap	3	0.7253	0.7639
	25	N/A	N/A
Single Paired Resampling Bootstrap	3	0.7243	0.7419
	25	N/A	N/A
Technical Replicate Calculated	3	0.9132	0.4446
	25	0.9699	0.3020
Combined Data Calculated	3	0.5521	0.6533
	25	0.1464	0.3738

Values marked as N/A were not computable as all of the data for those entries had a capture rate of 1, therefore there was no variance.

The confidence interval for the *N*-fold change, upon negative exponentiation, will be asymmetric about the mean (Livak and Schmittgen 2001). Also, the lower CI bound for the ddCt produces the upper bound for the *N*-fold change, and *vice versa*.

For technical replicates, when measurements are made on the reference DNA using both test and reference primers, no single measurement using the test primers is specifically paired to a particular measurement using the reference primers. Instead, the mean of each group is used to determine the difference (Ferreira *et al.* 2006). And while it is true that the difference of means equals the mean of differences regardless of how they are paired, the standard deviation of that mean difference changes according to if and how the data is paired. That standard deviation affects the hypothesis test.

The results do show that treating the unpaired data as paired produced an approximate 95% capture rate (95.13% on average regardless of the sample size). However, treating it as unpaired data using group means, there is only one ddCt value with one $\mathrm{SE}_{\mathrm{ddCt}}$ determined from the $\mathrm{SE}_{\mathrm{mean}}$ of the Ct groups (eq. 2). But if the data is treated as paired (though generated in an unpaired manner), then the standard deviation of the mean ddCt varies depending on how the data is arranged, producing a bias (or potential bias) in any single determination of the CI dependent on the arrangement (i.e., pairing) of the data.

An issue with small sample sizes (N_{Ct} = 3 or 4) is the possibility of failing to capture the true mean of the population within the range of the samples. For one group of three measurements, the average data range (instead of CI) capture rate is estimated to be only 75%. With four groups of measurements, however, the range capture rate increases. The simulations exhibited an estimated average range capture rate of 98.1% (in simulations with N_{Ct} = 3) or higher from all simulations, and 100% in all simulations with $N_{Ct} > 4$. While small sample sizes might be considered a problem, the simulations show that with four groups of measurements the probability of capturing the true mean within the range of the data is actually high.

Analysis of the calculated methods (Figure 4.1) for both technical replicates and combined data provided an estimated CI capture rate of approximately 95% regardless of the sample size, demonstrating their effectiveness.

However, the bootstrap methods (Figure 4.1) exhibited various behaviors. Though bootstrap methods can be effective under certain circumstances, the bootstrap algorithm may not perform at an optimal level under the experimental conditions and should be simulated *in silico* to evaluate the method's capture rate. The group means method begins at an estimated CI capture rate of 84.5% at N_{Ct} = 3 and asymptotes to the

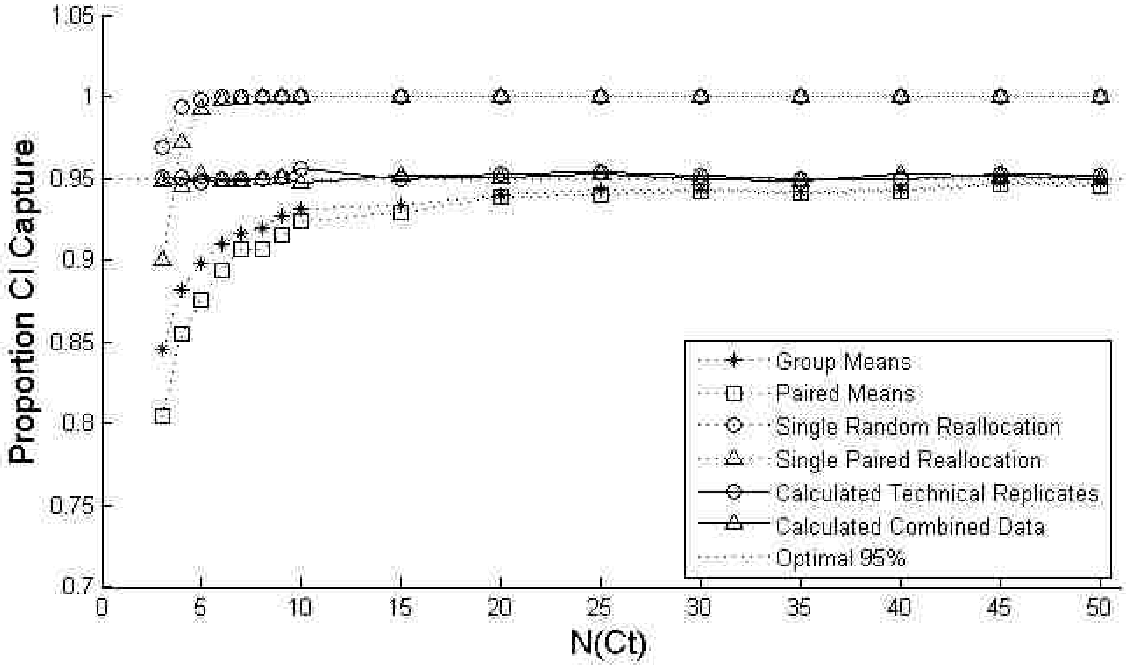


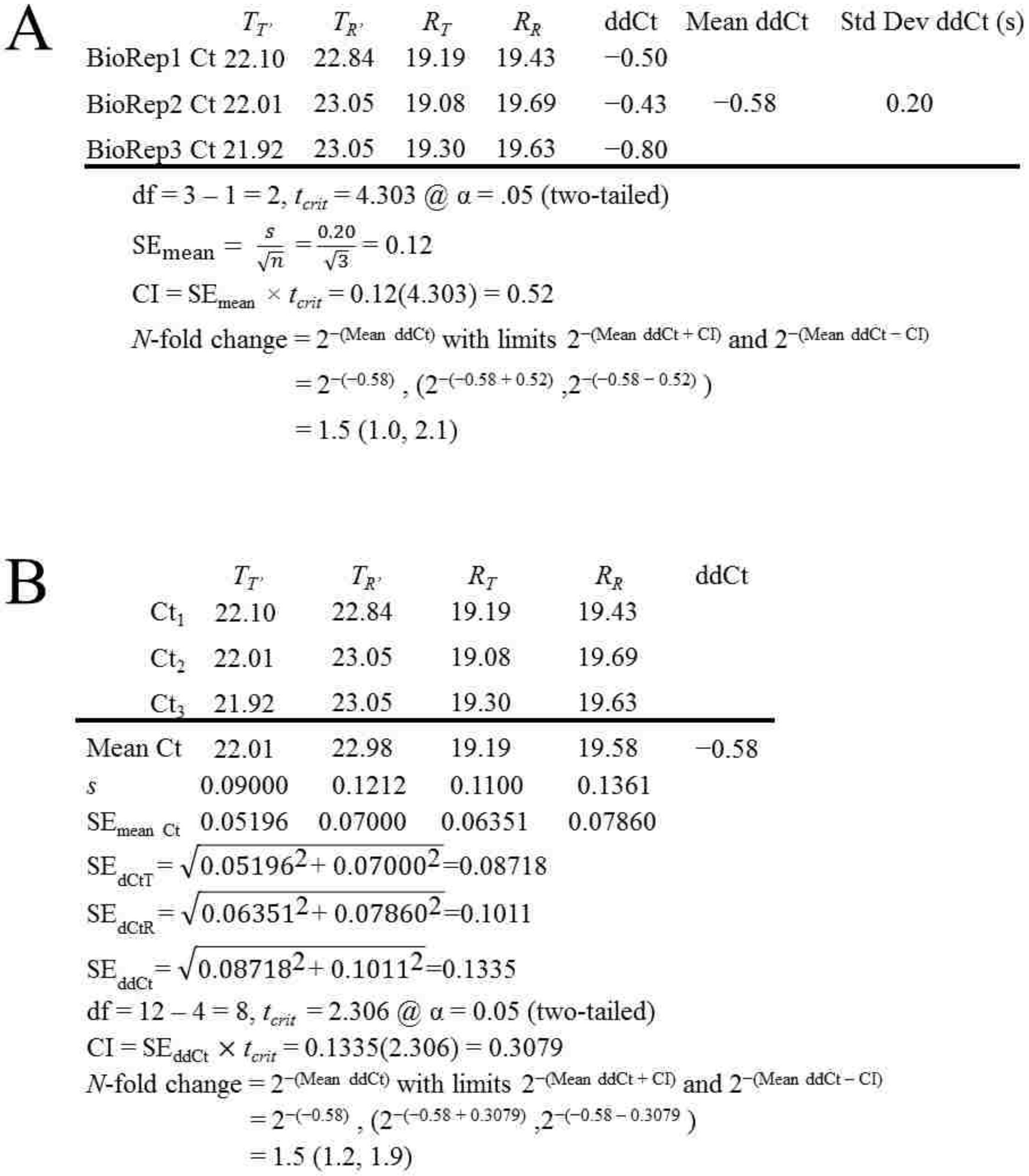
Figure 4.1. Proportion of True ddCt Capture by CI. Optimal capture rate is 0.95. Data points for N_{Ct} = 3 and 25 are based on 25,000 trials. All other data points are based on 15,000 trials.

95% capture rate as N_{Ct} increases. This method is close to the 95% capture rate when N_{Ct} > 25, but has an increasing tendency of type I error as N_{Ct} decreases to 3. The paired means method starts at an estimated CI capture rate of 80.5% at N_{Ct} = 3 and also asymptotes to the 95% capture rate as N_{Ct} increases, getting close to the 95% rate when N_{Ct} > 25. This method also has an increasing tendency of type I error as N_{Ct} decreases to 3. Both of these methods emulate the actual process of determining ddCt, and approach an optimal CI capture rate with large sample sizes. However, many qPCR experiments have small sample sizes, either because of limits on biological samples or budgets.

The single random resampling method has an estimated CI capture rate of 96.9% at N_{Ct} = 3, which is just slightly conservative of the 95% rate. As N_{Ct} increases, this method quickly approaches an estimated 100% capture rate, indicating a distinct bias toward type II error. The single paired resampling method starts with an estimated CI capture rate of 90.0% at $N_{Ct} = 3$, and an estimated capture rate of 97.3% at $N_{Ct} = 4$, the second being nearer the target CI capture rate of 95%. Increasing N_{Ct} produces estimated CI capture rates approaching 100%, again displaying a bias toward type II error. These two methods did not emulate the actual process of determining ddCt. They used the same data as the previous methods, keeping the same range but altering the distribution by reducing kurtosis. The result spreads the limits of the CI, bringing the capture rate closer to the optimal CI capture rate for small sample sizes, but quickly approaches a 100% capture rate as N_{Ct} increases.

The examples provided in Figures 4.2 and 4.3 demonstrate the differences in the determination of ddCt, $\mathrm{SE}_{\mathrm{ddCt}}$, and CI. With biological replicates (Figure 4.2A), statistics are performed on the ddCt values of the individual biological replicates, which is why df = number of biological replicates – 1 (Ott 1993). With technical replicates (Figure 4.2B), statistics are performed on the ddCt value as a difference of means of the groups of Ct values, therefore df = total number of Ct measurements – the number of Ct groups (Ott 1993). As shown in Figure 4.2, the mean *N*-fold change is the same in both methods, but the confidence bounds are different (larger in this case) in the biological replicate method than in the technical replicate method with the same data. The $\mathrm{SE}_{\mathrm{ddCt}}$ of technical replicates will be higher than the highest $\mathrm{SE}_{\mathrm{mean}}$ of any of the four groups of Ct measurements. It only takes one group with a high $\mathrm{SE}_{\mathrm{mean}}$ to cause the final ddCt measure to have a high $\mathrm{SE}_{\mathrm{ddCt}}$.

With the combined data shown in Figure 4.3, the technical replicates of each biological replicate are evaluated to produce ddCt values for each biological replicate.



A

	$T_{T'}$	$T_{R'}$	R_T	R_R	ddCt	Mean ddCt	Std Dev ddCt (s)
BioRep1 Ct	22.10	22.84	19.19	19.43	−0.50		
BioRep2 Ct	22.01	23.05	19.08	19.69	−0.43	−0.58	0.20
BioRep3 Ct	21.92	23.05	19.30	19.63	−0.80		

$$\text{df} = 3 - 1 = 2,\ t_{crit} = 4.303 \text{ @ } \alpha = .05 \text{ (two-tailed)}$$

$$\text{SE}_{\text{mean}} = \frac{s}{\sqrt{n}} = \frac{0.20}{\sqrt{3}} = 0.12$$

$$\text{CI} = \text{SE}_{\text{mean}} \times t_{crit} = 0.12(4.303) = 0.52$$

$$N\text{-fold change} = 2^{-(\text{Mean ddCt})} \text{ with limits } 2^{-(\text{Mean ddCt} + \text{CI})} \text{ and } 2^{-(\text{Mean ddCt} - \text{CI})}$$

$$= 2^{-(-0.58)},\ (2^{-(-0.58+0.52)}, 2^{-(-0.58-0.52)})$$

$$= 1.5\ (1.0,\ 2.1)$$

B

	$T_{T'}$	$T_{R'}$	R_T	R_R	ddCt
Ct_1	22.10	22.84	19.19	19.43	
Ct_2	22.01	23.05	19.08	19.69	
Ct_3	21.92	23.05	19.30	19.63	
Mean Ct	22.01	22.98	19.19	19.58	−0.58
s	0.09000	0.1212	0.1100	0.1361	
$\text{SE}_{\text{mean Ct}}$	0.05196	0.07000	0.06351	0.07860	

$$\text{SE}_{\text{dCtT}} = \sqrt{0.05196^2 + 0.07000^2} = 0.08718$$

$$\text{SE}_{\text{dCtR}} = \sqrt{0.06351^2 + 0.07860^2} = 0.1011$$

$$\text{SE}_{\text{ddCt}} = \sqrt{0.08718^2 + 0.1011^2} = 0.1335$$

$$\text{df} = 12 - 4 = 8,\ t_{crit} = 2.306 \text{ @ } \alpha = 0.05 \text{ (two-tailed)}$$

$$\text{CI} = \text{SE}_{\text{ddCt}} \times t_{crit} = 0.1335(2.306) = 0.3079$$

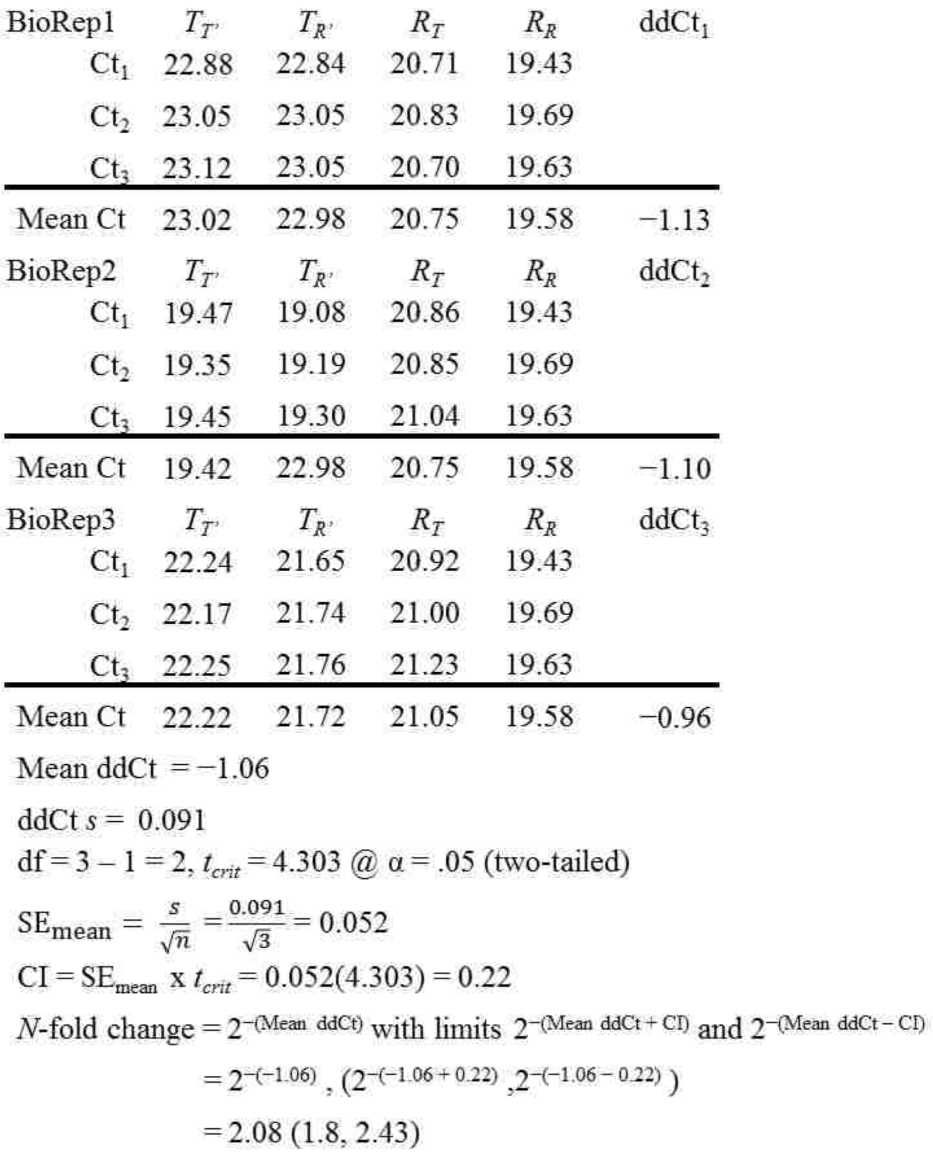
$$N\text{-fold change} = 2^{-(\text{Mean ddCt})} \text{ with limits } 2^{-(\text{Mean ddCt} + \text{CI})} \text{ and } 2^{-(\text{Mean ddCt} - \text{CI})}$$

$$= 2^{-(-0.58)},\ (2^{-(-0.58+0.3079)}, 2^{-(-0.58-0.3079})$$

$$= 1.5\ (1.2,\ 1.9)$$

Figure 4.2. Biological and technical replicate examples for calculating ddCt, SE$_{\text{mean}}$, df, and CI. (A) Biological replicates (paired data). (B) Technical replicates (unpaired data).

We are not interested in the standard deviations within the groups. The biological replicate ddCt values are treated as any other biological replicates. Simulations trying to combine the variance of the technical replicates with the variance of the biological



BioRep1	$T_{T'}$	$T_{R'}$	R_T	R_R	ddCt_1
Ct_1	22.88	22.84	20.71	19.43	
Ct_2	23.05	23.05	20.83	19.69	
Ct_3	23.12	23.05	20.70	19.63	
Mean Ct	23.02	22.98	20.75	19.58	−1.13
BioRep2	$T_{T'}$	$T_{R'}$	R_T	R_R	ddCt_2
Ct_1	19.47	19.08	20.86	19.43	
Ct_2	19.35	19.19	20.85	19.69	
Ct_3	19.45	19.30	21.04	19.63	
Mean Ct	19.42	22.98	20.75	19.58	−1.10
BioRep3	$T_{T'}$	$T_{R'}$	R_T	R_R	ddCt_3
Ct_1	22.24	21.65	20.92	19.43	
Ct_2	22.17	21.74	21.00	19.69	
Ct_3	22.25	21.76	21.23	19.63	
Mean Ct	22.22	21.72	21.05	19.58	−0.96

Mean ddCt = −1.06

ddCt s = 0.091

df = 3 – 1 = 2, t_{crit} = 4.303 @ α = .05 (two-tailed)

$$\text{SE}_{\text{mean}} = \frac{s}{\sqrt{n}} = \frac{0.091}{\sqrt{3}} = 0.052$$

$$\text{CI} = \text{SE}_{\text{mean}} \text{ x } t_{crit} = 0.052(4.303) = 0.22$$

$$N\text{-fold change} = 2^{-(\text{Mean ddCt})} \text{ with limits } 2^{-(\text{Mean ddCt} + \text{CI})} \text{ and } 2^{-(\text{Mean ddCt} - \text{CI})}$$

$$= 2^{-(-1.06)}, (2^{-(-1.06+0.22)}, 2^{-(-1.06-0.22)})$$

$$= 2.08\ (1.8, 2.43)$$

Figure 4.3. Combined data example for calculating ddCt, SE_{mean}, df, and CI.

replicates resulted in capture rates that were not optimal and varied with *σ* of the source distributions (data not shown). The variance among Ct measurements is only one of several independent arguments that sum to produce the observed variance of the biological replicate ddCt values (Kitchen *et al.* 2010). Therefore, the observed variance of the ddCt values in combined data is all that is needed. The simulations reveal (Figure 4.1) that the estimated capture rate of CIs produced by the biological replicate method on the ddCt values of combined data (15,000 trials per N_{Ct}) is approximately 95%, and it appears invariant with respect to source distribution σ and ddCt (Table 4.1).

The combined method can be used for biological replicates on different plates, as it only compares the value of each biological replicate ddCt relative to the other ddCt values, provided PCR efficiency is similar or accounted for. As the ddCt is the result of differences between measurements and not a direct Ct measure itself, calibrators may not be necessary between plates provided all the reactions for any biological replicate are on the same plate.

It is the author's hope that the above examples will help those trying to cope with the statistical analysis of qPCR data. Small sample sizes are common in qPCR experiments, raising questions about the effectiveness of bootstrap algorithms which haven't been tested for capture rate efficiency. With more researchers adopting the MIQE quidelines (Bustin *et al.* 2009), the need for applying appropriate statistical methods increases. We should always question whether an analytical method is appropriate for the research question and data at hand, and not just use the same method as everyone else out of convenience.

Acknowledgements

The author would like to thank Dr. Ulfar Bergthorsson for his support, suggestions, and review of the manuscript.

Competing Interests Statement

The author declares no competing interests.

Funding

The author declares no funding provided.

Chapter 4 Addendum

One of the few bootstrap methods that describes the algorithm is the REST© program (Pfaffl *et al.* 2002). The bootstrap resamples dCt values from paired sets and treats all of the data as if it came from the same distribution. It then evaluates how often it obtains a bootstrapped ddCt value as great or greater than the one observed. From this it provides a p-value rather than a CI. This process, as opposed to comparing two distributions to see if they are significantly different, might tend to be conservative. However, like most of the other methods considered, it is designed for analysis of paired data of biological replicates. Various software packages for analyzing qPCR data, including Q-Gene (Muller 2002), LightCycler Relative Quantification Software (Roche Diagnostics), qBase (Hellemans *et al.* 2007), SoFar (Metralabs), qCalculator (Gilsbach *et al.* 2006), Dart-PCR (Peirson *et al.* 2003), Gene Expression Macro (BioRad), and qPCR-DAMS (Jin *et al.* 2006), were reviewed (Pfaffl *et al.* 2009), and about half of them offered little or no statistical analysis. Those that did offer statistical methods did not appear to effectively describe how they were determining the standard deviation or the degrees of freedom, though it may be possible this information was simply not covered in the review.

As mentioned in the manuscript, the search to develop a method for determining the standard deviation based on both the variance of the biological replicates and the variance of the technical replicates was not successful. The equation used for this attempt (Headrick 2010) is shown below:

$$V = \frac{m^2V_1 + n^2V_2 - nV_1 - nV_2 - mV_1 - mV_2 + mnV_1 + mnV_2 + mn(M_1 - M_2)^2}{(n + m - 1)(n + m)}$$

where V is the observed variance, V_1 and V_2 are the variances of two sets of measurements, m and n represent the number of measurements in each set, and M_1 and M_2 are the means of the two sets. The simulations indicated that while this method is invariant with respect to ddCt, it is not invariant with respect to σ. The reason this approach was unsuccessful was simply that what the equation provides is an observed variance given variances both within and among populations of measurements. The variance of the biological replicates does not reflect merely differences between the sets of measurements, but includes the variance within the sets of measurements. Therefore, the variance of the biological replicates is the observed variance, which is derived from the standard deviation.

In summary, statistical methods to deal with hypothesis testing of technical replicates or combinations of technical and biological replicates were not available in the qPCR literature and had to be developed. Bootstrap methods may work for specific data sets, but clearly do not work for all. Evaluation of bootstrap methods in silico against the definition of confidence intervals provides a means of judging the effectiveness of that method. Calculated methods appear to be the best option, though one has to be careful of the pitfalls in determining the correct standard deviation and degrees of freedom.

Chapter 5

Conclusions

The results of this work demonstrate that the spontaneous rate of gene duplications, as measured on a per gene per generation basis, is approximately two orders of magnitude higher than the point mutation rate. Thus if both a duplication and a point mutation confer an increase in fitness, the probability is that the duplication will occur first. And whereas a point mutation changes only a single nucleotide, duplications can span hundreds of thousands of nucleotides and multiple genes, dramatically raising the amount of genetic material subject to selection in a single mutational event.

This work also demonstrates that duplications can clearly be adaptive. Most of the duplications, as well as most of the deletions, demonstrated clear evidence of selection, i.e. a constant increase in frequency of the CNV in the population over generations. That so many CNVs would have exhibited this pattern is highly unlikely to have occurred by genetic drift alone.

While the focus here was primarily on the adaptive nature of duplications, deletions were also investigated. While duplications represent an increase of genetic information, deletions are usually perceived as information loss. However, if the unit of information is the gene, deletions can be viewed in certain circumstances as information change, the alteration of a gene without actual loss. If a deletion removes an exon, it doesn't remove the entire gene or alter its expression. From a DNA information perspective, however, deletions can only be a loss of information.

As to what selection pressures these CNVs are specifically adapting, we do not know for certain. Some possibilities have been presented in the manuscripts, but a detailed functional analysis of specific CNVs and the individual genes within them would be an important next step in this research. As well as the genes themselves, variations in the mechanisms producing these CNVs also need to be investigated. Apparent differences in mechanisms have been shown between very different organisms such as *C. elegans* and the yeast *Saccharomyces cerevisiae* (Katju *et al*. 2009). These differences can also manifest themselves even among closely related taxa, such as *Homo sapiens* and *Pan troglodytes* (Bu 2015), suggesting that factors affecting duplication mechanisms, such as viral prevalence, can influence the frequency of duplication and deletion mechanisms in a population, thus altering the population's evolutionary trajectory.

This work demonstrates the importance of the contributions of gene duplications and deletions to the adaptability of populations. More research elucidating the degree of contribution of the various mechanisms would contribute to our understanding of their effect on evolutionary trajectories. Rapid evolutionary changes, including increases in genome information content, are clearly possible via this process.

LIST OF APPENDICES

Appendix A

Supplemental Material for Chapter 2

Supplemental Experimental Procedures

Worm MA Lines

Independent *C. elegans* lines were maintained at small population sizes via single-progeny descent for up to 465 generations in order to preserve the vast majority of spontaneous non-lethal and non-sterile mutations (Vassilieva and Lynch 1999). The duplications and deletions reported here are homozygous due to multiple generations of selfing.

Preparation of genomic DNA from MA lines

Ten MA lines were grown to large population sizes (>5000 individuals) and starved. The worms were harvested and the genomic DNA extracted using standard procedures (Sulston and Hodgkin 1988).

CGH Arrays

Genome-wide duplications/deletions in the MA lines were detected using NimbleGen CGH arrays. Each array contains 385,000 unique probes (50—75 nts in length) that span both coding and noncoding regions. The design of the array is based on Wormbase version WS120 but all the data coordinates provided in this manuscript are based on version WS219.

For each DNA microarray hybridization, 1 μg genomic DNA each from an MA line and Bristol N2 (the common reference for all hybridizations) were labeled using Cy-3 or Cy-5 labeled random primers (TriLink Biotechnologies) and New England Biotech Klenow fragment, 50U/μl (M0212M) according to the manufacturer's specifications. 6 μg of labeled DNA from each control and experimental line was added to 18 μl Hybridization Solution Master Mix, and hybridized to a NimbleGen CGH array, design 071114_CE2_WG_CGH_T. Slides were hybridized for 16-h at 42°C in a BioMicro MAUI® Hybridization System, washed with NimbleGen wash buffers and dried for 1 min on an ArrayIt™ slide drier. The slides were scanned on an Axon GenePix 4000B scanner and the data analyzed in Nimblescan and SignalMap. Arrays of Bristol N2 DNA hybridized against itself served as baseline controls. Gene identities for regions with copy-number variation were obtained from Wormbase sequence version WS120. Bootstrap confidence intervals for the duplication and deletion rates were calculated using Matlab 2009b built-in "bootci" function.

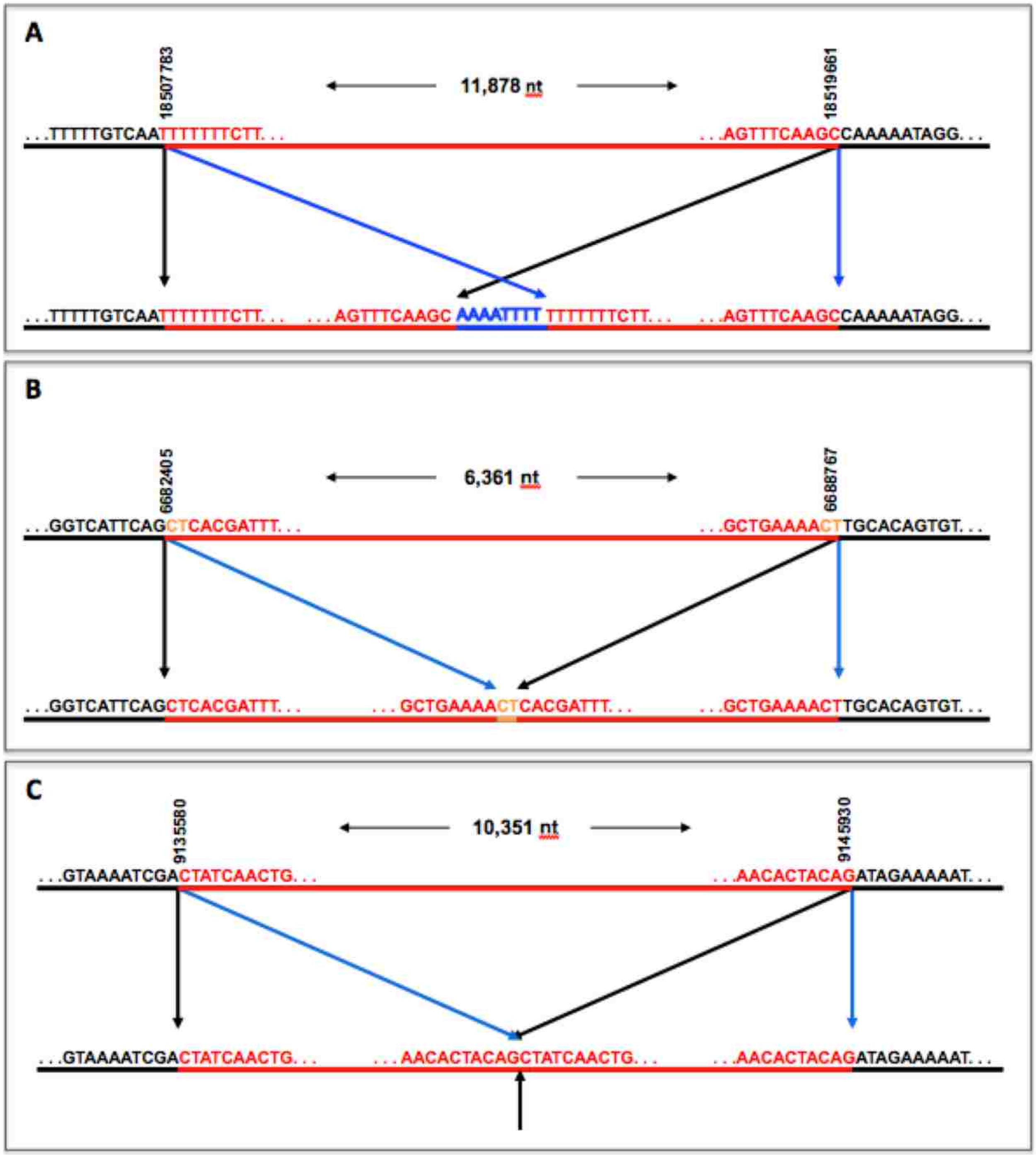
Quantitative PCR

qPCR was performed using FastStart SYBR Green with Rox (Roche) and the reactions were run on an ABI Prism 7000 Sequence Detection System. Total gDNA template per 25 uL reaction was 20 ng (5 uL @ 4 ng/uL) and the primer concentration in the reaction was 200 nmol/ primer. Serial dilutions of 10x, 1x, 0.1x, 0.01x, and 0.001x were performed for standardization. No template controls (NTCs) were also utilized to evaluate spurious products and contamination. Four sets of reactions were performed in triplicate for each locus: reference DNA with single copy reference primers, reference

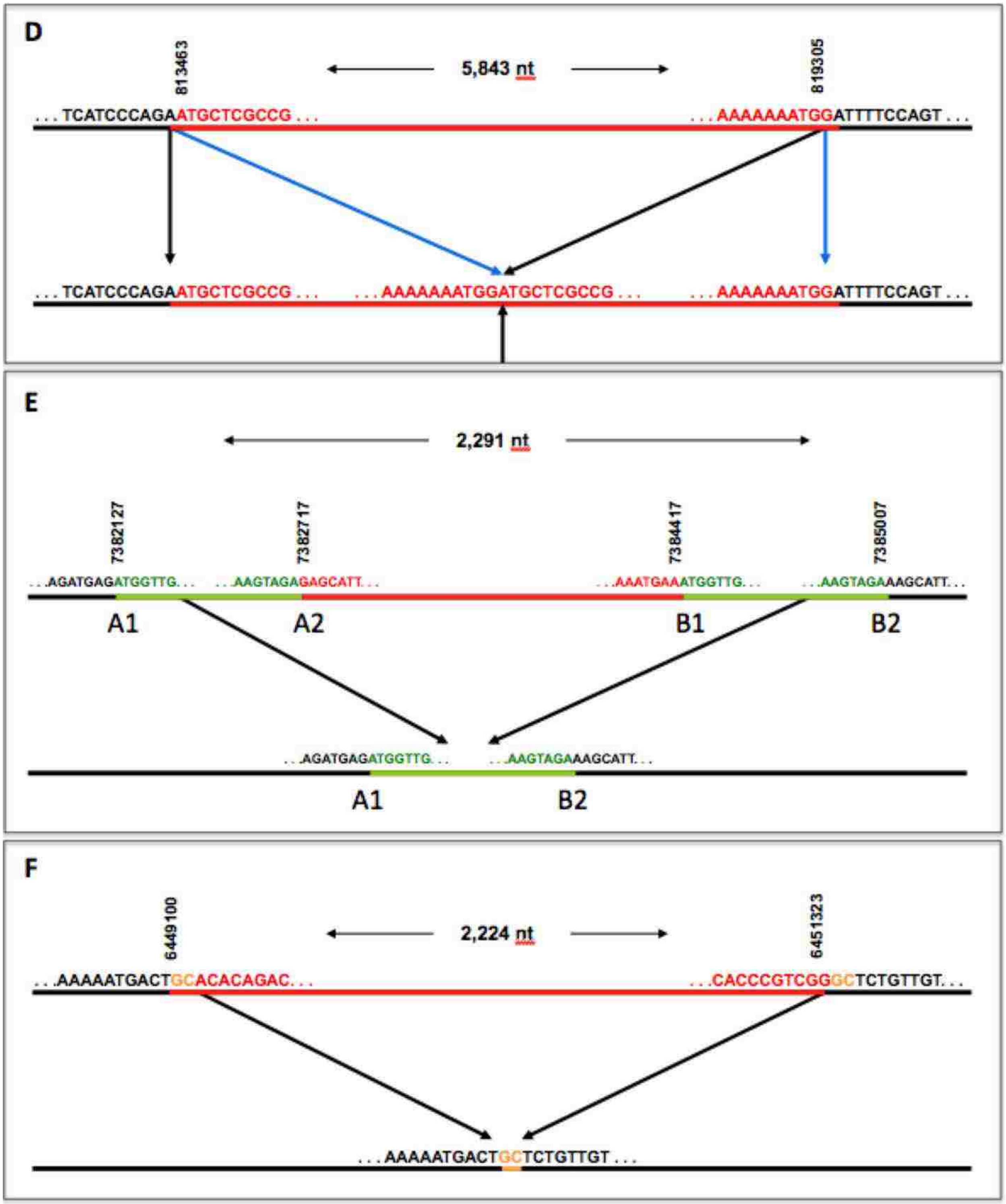
DNA with test primers (primers in region of interest), test DNA with reference primers, and finally test DNA with test primers. The primers were usually 20 nt long, with a GC content between 40 and 60 %, and generated products close to 100 nt long. The primers were designed using Primer3 (Rozen and Skaletsky 1998). Dissociation peaks were evaluated for primer performance and agarose gel electrophoresis (2%) was used to confirm the products.

PCR and DNA sequencing across duplication and deletion breakpoints

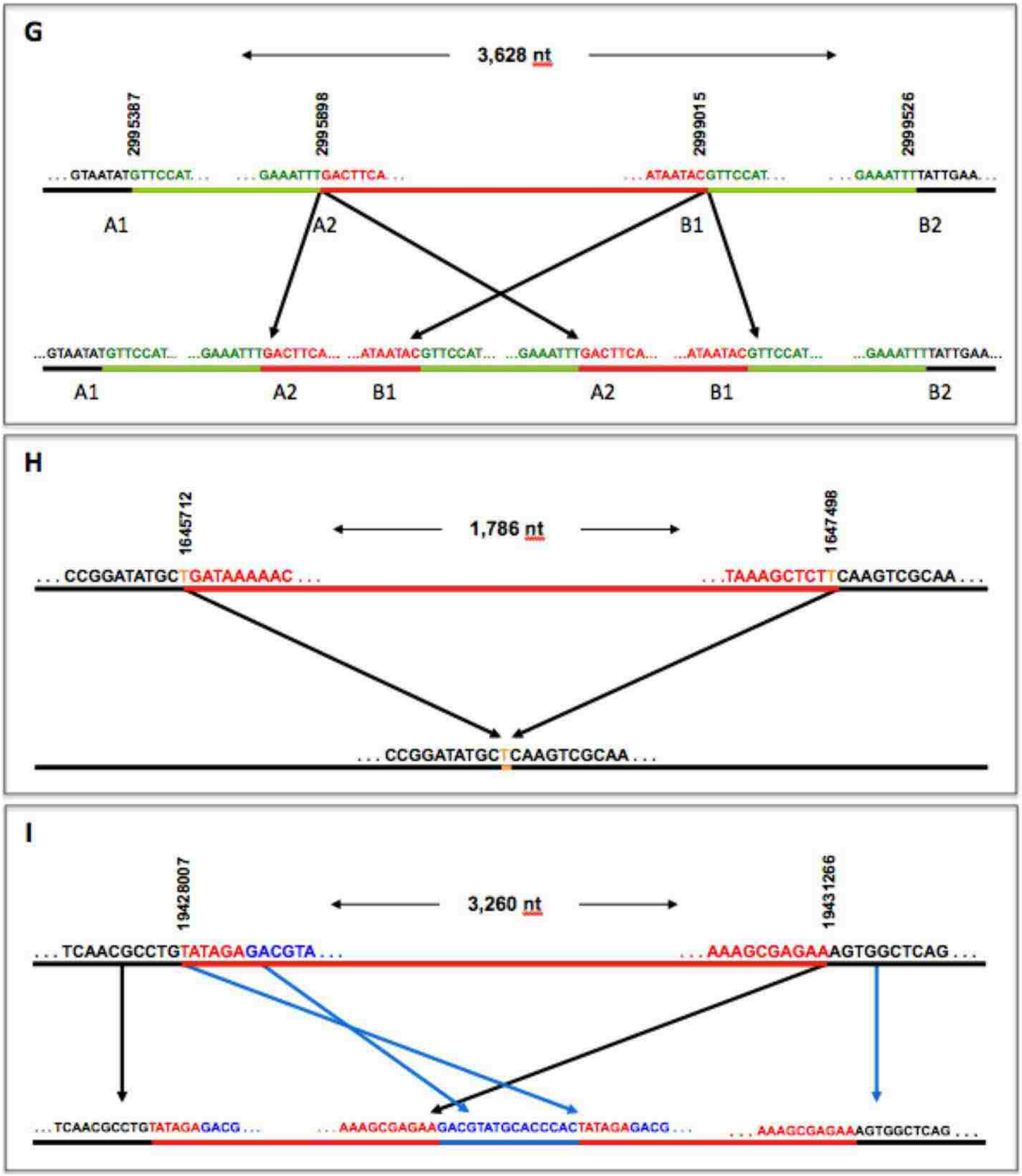
For sequencing duplication breakpoints, we designed primers oriented in opposite directions within the predicted boundaries of the duplication event. In genomes bearing only a single gene-copy, the forward and reverse primers are divergent and would fail to initiate PCR amplification. However, in the event of gene duplication resulting in two adjacent paralogs (tandem or inverted), the primers are rendered convergent and PCR amplification and subsequent DNA sequencing. For deletions, primers were designed to DNA sequences flanking the deleted sequence. This approach would fail to detect gene duplications and deletions with additional local rearrangements or have been rendered genomically distant via translocations. The PCR products were either gel-extracted and cleaned up using QIAquick Gel Extraction Kit (Qiagen) or prepared directly for sequencing using ExoSAP-IT (GE HealthCare Life Sciences). The PCR products were subsequently sequenced using Big Dye® Terminator v3.1 Cycle Sequencing Kits (AB Applied Biosystems) on an ABI 3130x Genetic Analyzer. The sequence information has been provided to Wormbase (www.wormbase.org) for annotation.


A
18507783
11,878 nt
18519661
. . .TTTTTGTCAATTTTTTTTCTT. . .
. . .AGTTTCAAGCCAAAAATAGG. . .
. . .TTTTTGTCAATTTTTTTTCTT. . .
. . . AGTTTCAAGC AAAATTTT TTTTTTTCTT. . .
. . .AGTTTCAAGCCAAAAATAGG. . .
B
6682405
6,361 nt
6688767
. . .GGTCATTCAGCTCACGATTT. . .
. . .GCTGAAAACTTGCACAGTGT. . .
. . .GGTCATTCAGCTCACGATTT. . .
. . . GCTGAAAACTCACGATTT. . .
. . .GCTGAAAACTTGCACAGTGT. . .
C
9135580
10,351 nt
9145930
. . .GTAAAATCGACTATCAACTG. . .
. . .AACACTACAGATAGAAAAAT. . .
. . .GTAAAATCGACTATCAACTG. . .
. . . AACACTACAGCTATCAACTG. . .
. . .AACACTACAGATAGAAAAAT. . .

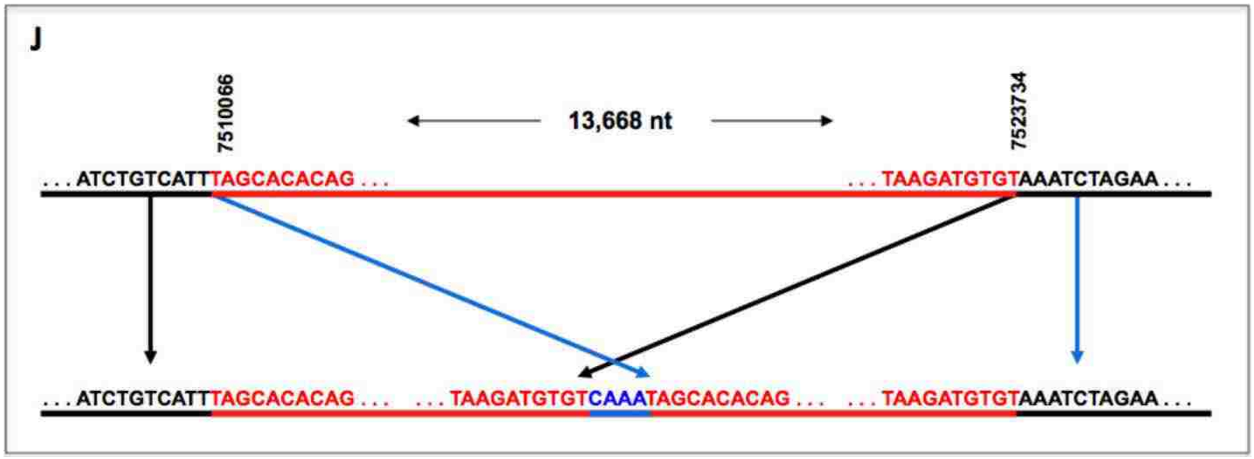
Supplemental Figure A.1. Lipinski *et al.* (1/4)


D
813463
5,843 nt
819305
. . . TCATCCCAGAATGCTCGCCG . . .
. . . AAAAAAATGGATTTTCCAGT . . .
. . . TCATCCCAGAATGCTCGCCG . . .
. . . AAAAAAATGGATGCTCGCCG . . .
. . . AAAAAAATGGATTTTCCAGT . . .
E
2,291 nt
7382127
7382717
7384417
7385007
. . .AGATGAGATGGTTG. . .
. . .AAGTAGAGAGCATT. . .
. . .AAATGAAATGGTTG. . .
. . .AAGTAGAAAGCATT. . .
A1
A2
B1
B2
. . .AGATGAGATGGTTG. . .
. . .AAGTAGAAAGCATT. . .
A1
B2
F
6449100
2,224 nt
6451323
. . .AAAAATGACTGCACACAGAC. . .
. . .CACCCGTCGGGCTCTGTTGT. . .
. . . AAAAATGACTGCTCTGTTGT . . .

Supplemental Figure A.1. Lipinski *et al.* (2/4)


G
3,628 nt
2995387
2995898
2999015
2999526
. . . GTAATATGTTCCAT. . .
. . . GAAATTTGACTTCA. . .
. . . ATAATACGTTCCAT. . .
. . . GAAATTTTATTGAA. . .
A1
A2
B1
B2
...GTAATATGTTCCAT...
...GAAATTTGACTTCA...
...ATAATACGTTCCAT...
...GAAATTTGACTTCA...
...ATAATACGTTCCAT...
...GAAATTTTATTGAA...
A1
A2
B1
A2
B1
B2
H
1645712
1,786 nt
1647498
. . . CCGGATATGCTGATAAAAAC . . .
. . .TAAAGCTCTTCAAGTCGCAA . . .
. . . CCGGATATGCTCAAGTCGCAA . . .
I
19428007
3,260 nt
19431266
. . . TCAACGCCTGTATAGAGACGTA . . .
. . . AAAGCGAGAAAGTGGCTCAG . . .
. . . TCAACGCCTGTATAGAGACG . . .
. . . AAAGCGAGAAGACGTATGCACCCACTATAGAGACG . . .
. . . AAAGCGAGAAAGTGGCTCAG . . .

Supplemental Figure A.1. Lipinski *et al.* (3/4)


J
7510066
13,668 nt
7523734
. . . ATCTGTCATTTAGCACACAG . . .
. . . TAAGATGTGTAAATCTAGAA . . .
. . . ATCTGTCATTTAGCACACAG . . .
. . . TAAGATGTGTCAAATAGCACACAG . . .
. . . TAAGATGTGTAAATCTAGAA . . .

Supplemental Figure A.1. Lipinski *et al.* (4/4)

Figure A.1. Duplication and Deletion Breakpoints in MA lines and N2 isolates. A-F and G-J represent duplication/deletion breakpoints in MA lines and the N2 isolate that was used as a common reference in the CGH experiments, respectively. The upper line and coordinates indicate the affected region in the sequenced genome. The nucleotides at the ends of the duplicated or deleted region are shown in red and the flanking sequence in black. The length of the duplicated or deleted sequence is also indicated. On the lower line, the nucleotides in blue indicate sequence that has been inserted at the breakpoint and nucleotides in orange indicate microhomology at the ends. All positions are based on Wormbase version WS219.

(A) Duplication on Chromosome V in *C. elegans* MA2 (Related to Table 2.1).

(B) Duplication on Chromosome I in *C. elegans* MA78 (Related to Table 2.1).

(C) Duplication on Chromosome III in *C. elegans* MA78 (Related to Figure 2.1A and Table 2.1).

(D) Duplication on Chromosome III in *C. elegans* MA94 (Related to Table 2.1).

(E) Deletion on Chromosome V in *C. elegans* MA78, probably due to unequal crossing-over. The green nucleotides represent the two paralogous copies in the ancestral N2 strain and the red sequence denotes unique sequence inserted between the two original paralogs and deleted in MA strain (Related to Table 2.2).

(F) Deletion on Chromosome X in *C. elegans* MA84 (Related to Table 2.2).

(G) Duplication on Chromosome V in the *C. elegans* ancestral N2 strain, probably due to unequal crossing over. The green nucleotides represent the two paralogous copies in the reference N2 strain and the red sequence denotes unique sequence inserted between the two original paralogs. The duplication results in three copies of the green region and two copies of the red (Related to Table 2.3).

(H) Deletion on Chromosome V in the *C. elegans* N2 reference strain (Related to Table 2.3).

(I) Duplication B on Chromosome V in the *C. elegans* N2 reference strain. The 15 nt region in blue gets inserted at the breakpoint as well as duplicated as a part of a tandem duplication (Related to Table 2.3).

(J) Duplication B on Chromosome X in the *C. elegans* N2 reference strain (Related to Table 2.3).

Supplemental Tables

Table A.1. Comparison of copy-number estimates for putative duplications in *C. elegans* MA lines based on qPCR and aCGH methods (Related to Table 2.1)

MA Line ID Number	Chromosome	Start Position	qPCR Copy Number	aCGH Copy
2	V	18,507,783	2.18	2.24
18	V	10,445,133	3.32	3.40
18	V	17,847,927	7.93	4.12
29	IV	17,482,852	2.03	1.83
29	X	12,763,189	2.18	1.70
63	V	4,893	2.14	1.71
63	X	3,559,284	2.29	1.90
78	I	6,682,405	1.71	2.04
78	III	9,135,580	1.55	2.05
78	X	17,694,155	3.05	2.98
83	IV	11,695,251	4.22	3.36
94	III	813,463	3.15	1.74
99	I	10,716,364	2.55	1.62
99	III	12,190,163	1.95	1.96

The qPCR copy number is 2^{-ddCt}, where ddCt = (Tt-Tr)-(Rt-Rr). Tt = test DNA with test primers, Tr = test DNA with reference primers, Rt = reference DNA with test primers, Rr = reference DNA with reference primers. The microarray (aCGH) copy number is 2^{x} where x = mean of the $\log_2$ ratios within the predicted region of the rearrangement.

Table A.2. Comparison of copy-number estimates for putative deletions in *C. elegans* MA lines based on qPCR and aCGH methods (Related to Table 2.2)

MA Line ID Number	Chromosome	Start Position	qPCR Copy Number	aCGH Copy
18	II	5,779,876	1.14×10^{-7}	0.03
29	X	12,759,852	6.84×10^{-7}	0.03
63	V	1,300	3.32×10^{-5}	0.07
78	V	7,382,127	4.74×10^{-5}	0.11
78	X	12,111	1.83×10^{-5}	0.03
78	X	17,698,905	6.07×10^{-4}	0.05
83	II	539	2.95×10^{-5}	0.09
83	IV	8,582,020	8.06×10^{-6}	0.06
83	IV	15,187,709	5.06×10^{-6}	0.10
84	X	6,449,100	6.14×10^{-4}	0.05
99	III	12,186,218	1.76×10^{-5}	0.04

The methods are the same as outlined in Table A.1.

Table A.3. Comparison of copy-number estimates based on qPCR and aCGH methods between the reference N2 strain and the N2 strain that serve as the ancestor for all *C. elegans* MA lines used in this study (Related to Table 2.3)

Chromosome	Start Position	qPCR Copy Number	aCGH Copy Number
V	1,645,712	9.44×10^{-6}	0.07
V	2,995,387	1.18	1.96
V	18,706,963	4.38	2.50
V	19,430,653	2.20	1.89
X	86,369	4.36	3.03
X	7,510,066	2.71	1.92

The methods are the same as outlined in Tables A.1 and A.2.

Table A.4. Open reading frames in spontaneous duplications and deletions in the Mutation Accumulation lines (Related to Tables 2.1 and 2.2)

Duplications	Location	Complete / Partial ORFs
MA2V	[18507783:18519661]	Y51A2D.1 / Y51A2D.19, Y51A2D.4
MA18VA	10445133:10455580	Y32F6A.5 / F22E12.1, Y32F6A.3
MA18VB	17847927:17858066	Y59A8A.3 / none
MA29IV	17482852:17490972	4R79.1, 4R79.5 / 4R79.2
MA29X	12763189:12767835	none / F22E10.5, T22H6.1
MA63V	4893:18375	B0348.5, B0348.6 / none
MA63X	3559284:3567765	none / F59D8.1, F59D8.2
MA78I	[6682405:6688767]	none / C17F3.3, T23B3.4
MA78III	[9135580:9145930]	ZK512.6, ZK512.7, ZK512.2, ZK512.11, ZK512.4 / none
MA78XA	7609:11592	none / CE7X_3.2
MA78XB	17694155:17696571	none / F20B4.6
MA83IV	1169251:11700130	T22B2.1 / ZK792.7
MA94III	[813463:819305]	none / B0412.2, B0412.1
MA99I	10716364:10721038	none / B0205.1, B0205.9
MA99III	12190163:12194367	none / Y75B8A.12

Deletions	Location	Complete / Partial ORFs
MA18DII	5779876:5784792	none / K05F1.6
MA29DX	12759852:12761568	none / F16H9.2 (intron only)
MA63DV	1300:3319	cTe13X.1 / none
MA78DV	[7382127:7385007]	none / C03G6.18, C03G6.19
MA78DXA	12111:12925	none
MA78DXB	17698905:17718646	cTe155X.1, 6R55.2, H11L12.1, F20B4.2 / F20B4.6
MA83DII	539:4901	2L52.1 / none
MA83DIVA	8582020:8613790	Y59H11AR.2, Y59H11AR.3, Y59H11AR.5, Y59H11AR.4, F42A9.6 / F42A9.5
MA83DIVB	15187709:15187923	none
MA84DX	[6449100:6451323]	K10C2.5 / none
MA99III	12186218:12189728	none / Y75B8A.11

Spans in brackets are actual breakpoints based on DNA sequencing. Others are predicted breakpoints based on microarray data. Positions are based to WS219.

Appendix B

Code for JCFreadCGH.m

```
%JCFread_cgh.m
%version 1.1  21 Jan 2010
%James Farslow (jfars@unm.edu)
%This program reads text cgh data and plots the data for the individual chromosomes
%It also plots a subgraph with a sliding window average to reduce the noise
clear;
clc;
%>>>>>>>>>>>>>>>>>specifiy mode of operation: nimblegen or Stephane's files
modeType = input('Input Mode: <N>imblegen or <S>tephane: ','s');
if (modeType == 'N')
   %>>>>>>>>>>>>specify file here
   filenm = input('Input File Name: ','s');
   opennm = 'C:\James\Research\Lab_Work\Celegans Research\MA Microarray Data\';
   %opennm = 'C:\James\Research\Lab_Work\Celegans Research\Recovery Microarray
Data\';
   %opennm = '/Documents and Settings/James/My Documents/Celegans
Research/Kendra
Backup/BergthorssonLabWork/Microarrays/Nimblegen/Lynch_MA_Lines/Corrected(Dy
eSwap)_Nimblegen_Data/';
   %>>>>>>>>>>>>>>>for testing only
```

```
    %opennm = '/Documents and Settings/James/My Documents/Celegans Research/ComputerPrograms/Matlab/testdata/';
    newOpenName = strcat(opennm, filenm,'.txt');
    %first open filehandle for reading only
    fileName = fopen(newOpenName,'r');
    fileStr = regexprep(filenm, '_', ' ');  %filename to be displayed in the titles
    fprintf('Reading data . . .\n');
    %first read headers - use textscan
    headers = textscan(fileName,'%s %s %s %s %s %s %s %s %s %s %s %s %s %s',1);
    %read data file into arrays (cells?)- use textscan - will pick up where it
    %left off - each column of the dataSet array (1 row) is an array of cells
    dataSet = textscan(fileName,'%d %s %s %s %d %f %s %d %f %f %f %f %f %f');
    fclose(fileName);
    fprintf('Assigning data.  Please wait . . .\n');
    %convert cells to individual arrays directly
    A = dataSet{1}; B = dataSet{2}; C = dataSet{3}; D = dataSet{4}; E = dataSet{5};
    F = dataSet{6}; G = dataSet{7}; H = dataSet{8}; I = dataSet{9}; J = dataSet{10};
    K = dataSet{11}; L = dataSet{12}; M = dataSet{13}; N = dataSet{14};
else
    filenm = input('Input File Name: ','s');
    opennm = 'C:\James\Research\Lab_Work\Celegans Research\Micrarray Data Stephane\';
    newOpenName = strcat(opennm, filenm,'.csv');
    fileName = fopen(newOpenName,'r');
    fileStr = regexprep(filenm, '_', ' ');  %filename to be displayed in the titles
    fprintf('Reading data . . .\n');
    %[num, txt] = xlsread(newOpenName);
    %E = num(:,1); N = num(:,3);
    %C = txt{3:100,1};
    %first read headers - use textscan
    headers = textscan(fileName,'%s %s %s %s',2);
    %read data file into arrays (cells?)- use textscan - will pick up where it
    %left off - each column of the dataSet array (1 row) is an array of cells
    dataSet = textscan(fileName,'%s %d %f %f','delimiter',',');
    %fclose(fileName);
    fprintf('Assigning data.  Please wait . . .\n');
    %convert cells to individual arrays directly
    C = dataSet{1}; E = dataSet{2}; M = dataSet{3};  N = dataSet{4};
end
%convert case of C to upper case
C = upper(C);
%break array into blocks based on chromosome (strmatch)
indices1 = strmatch('CHRI ',C); %define indices for chrom 1
%indices1 can now be used to specify the elements of the other arrays
%do the same for the other chromosomes
indices2 = strmatch('CHRII ',C);
```

```
indices3 = strmatch('CHRIII',C);

indices4 = strmatch('CHRIV ',C);

indices5 = strmatch('CHRV ',C);

indicesX = strmatch('CHRX ',C);

%***************create moving average data****************

%make single arrays of indexed data to loop through

posE1 = E(indices1); dataN1 = N(indices1); %positions will match data points

posE2 = E(indices2); dataN2 = N(indices2);

posE3 = E(indices3); dataN3 = N(indices3);

posE4 = E(indices4); dataN4 = N(indices4);

posE5 = E(indices5); dataN5 = N(indices5);

posEX = E(indicesX); dataNX = N(indicesX);

%>>>>>>>>>>>>Set paramters

windowSize = 10;  %set window size for averaging <<<<<<<<<<<<<<<<<<<<<<<<<

%can't loop through all of them at once because array sizes are different

fprintf('Averaging data points, please wait . . .\n');

%******Chromosome I

%get size of loop

loopSize = length(dataN1);

%create array for averages

avN1 = zeros(loopSize-(2*windowSize),1);

%and for positions - corrected for starting at windowSize + 1

posavN1 = posE1(1+windowSize:loopSize-windowSize);  %same range as for loop
```

```
%calculate averages
for ind1 = 1+windowSize:loopSize-windowSize
   avN1(ind1-windowSize) = sum(dataN1(ind1-
windowSize:ind1+windowSize))/((2*windowSize)+1);
end
%get mean and st dev of all data
meanN1 = mean(dataN1);
stdN1 = std(dataN1);
%******Chromosome II
%get size of loop
loopSize = length(dataN2);
%create array for averages
avN2 = zeros(loopSize-(2*windowSize),1);
%and for positions - corrected for starting at windowSize + 1
posavN2 = posE2(1+windowSize:loopSize-windowSize);  %same range as for loop
%calculate averages
for ind1 = 1+windowSize:loopSize-windowSize
   avN2(ind1-windowSize) = sum(dataN2(ind1-
windowSize:ind1+windowSize))/((2*windowSize)+1);
end
%get mean and st dev of all data
meanN2 = mean(dataN2);
stdN2 = std(dataN2);
```

```
%******Chromosome III
%get size of loop
loopSize = length(dataN3);
%create array for averages
avN3 = zeros(loopSize-(2*windowSize),1);
%and for positions - corrected for starting at windowSize + 1
posavN3 = posE3(1+windowSize:loopSize-windowSize);  %same range as for loop
%calculate averages
for ind1 = 1+windowSize:loopSize-windowSize
   avN3(ind1-windowSize) = sum(dataN3(ind1-
windowSize:ind1+windowSize))/((2*windowSize)+1);
end
%get mean and st dev of all data
meanN3 = mean(dataN3);
stdN3 = std(dataN3);
%*******Chromosome IV
%get size of loop
loopSize = length(dataN4);
%create array for averages
avN4 = zeros(loopSize-(2*windowSize),1);
%and for positions - corrected for starting at windowSize + 1
posavN4 = posE4(1+windowSize:loopSize-windowSize);  %same range as for loop
%calculate averages
```

```
for ind1 = 1+windowSize:loopSize-windowSize
   avN4(ind1-windowSize) = sum(dataN4(ind1-
windowSize:ind1+windowSize))/((2*windowSize)+1);
end
%get mean and st dev of all data
meanN4 = mean(dataN4);
stdN4 = std(dataN4);
%*******Chromosome V
%get size of loop
loopSize = length(dataN5);
%create array for averages
avN5 = zeros(loopSize-(2*windowSize),1);
%and for positions - corrected for starting at windowSize + 1
posavN5 = posE5(1+windowSize:loopSize-windowSize);  %same range as for loop
%calculate averages
for ind1 = 1+windowSize:loopSize-windowSize
   avN5(ind1-windowSize) = sum(dataN5(ind1-
windowSize:ind1+windowSize))/((2*windowSize)+1);
end
%get mean and st dev of all data
meanN5 = mean(dataN5);
stdN5 = std(dataN5);
%*******Chromosome X
```

```
%get size of loop
loopSize = length(dataNX);
%create array for averages
avNX = zeros(loopSize-(2*windowSize),1);
%and for positions - corrected for starting at windowSize + 1
posavNX = posEX(1+windowSize:loopSize-windowSize);  %same range as for loop
%calculate averages
for ind1 = 1+windowSize:loopSize-windowSize
   avNX(ind1-windowSize) = sum(dataNX(ind1-
windowSize:ind1+windowSize))/((2*windowSize)+1);
end
%get mean and st dev of all data
meanNX = mean(dataNX);
stdNX = std(dataNX);
fprintf('Averaging complete.  Plotting data.\n');
%***********************PLOTTING*************************************
*****
%plot each subarray - x-axis = position, y-axis = corrected ratio
%set the plots to tile over in a descending pattern
%for each plot, position is defined as E(indicesN)
%corrected ratio is defined as N(indicesN)
figure(1); clf(1);
hold on;
```

```
%plot gridlines in upper subplot
h1 = subplot(2,1,1);
h2 = subplot(2,1,2);
hold (h1);  %hold toggle on
subplot(2,1,1);plot([-100000 max(E(indices1))+100000],[1 1],'-.g');
subplot(2,1,1);plot([-100000 max(E(indices1))+100000],[2 2],'-.g');
subplot(2,1,1);plot([-100000 max(E(indices1))+100000],[-1 -1],'-.g');
subplot(2,1,1);plot([-100000 max(E(indices1))+100000],[-2 -2],'-.g');
%plot data points to upper subplot
subplot(2,1,1);plot(E(indices1),N(indices1),' .k','MarkerSize',3);
set(1,'position',[10 140 1250 580],'Name','JCFread_cgh: Chromosome
I','NumberTitle','off');
title(sprintf('C. elegans Chromosome I CGH Data, File: %s',fileStr),'FontSize',22);
xlabel('Position','FontSize',18);
ylabel('Corrected Ratio','FontSize',16);
axis([-100000 max(E(indices1))+100000 min(N(indices1))-0.5 max(N(indices1))+0.5]);
hold (h1);  %hold toggle off
%draw second subplot of averages - lower subplot
%determine threshold values
upperThr1 = meanN1+(2*stdN1);lowerThr1 = meanN1-(2*stdN1);
annotation('textbox',[0 0.47 0.1 0.1],'string',sprintf('Data Mean: %4.3f\nData STD:
%4.3f\nUpper Threshold: %4.3f\nLower Threshold: %4.3f\n(Threshold = 2
sd)',meanN1,stdN1,upperThr1,lowerThr1));
```

```
hold (h2);  %toggle on
%plot gridlines
subplot(2,1,2);plot([-100000 max(E(indices1))+100000],[1 1],'-.g');
subplot(2,1,2);plot([-100000 max(E(indices1))+100000],[2 2],'-.g');
plot([-100000 max(E(indices1))+100000],[0 0],'-.g');
subplot(2,1,2);plot([-100000 max(E(indices1))+100000],[-1 -1],'-.g');
subplot(2,1,2);plot([-100000 max(E(indices1))+100000],[-2 -2],'-.g');
%add threshold gridlines
subplot(2,1,2);plot([-100000 max(E(indices1))+100000],[upperThr1 upperThr1],'-.r');
subplot(2,1,2);plot([-100000 max(E(indices1))+100000],[lowerThr1 lowerThr1],'-.r');
%plot data to lower subplot
subplot(2,1,2);plot(posavN1,avN1,' .k','MarkerSize',3);
title(sprintf('Average Corrected Ratio Data With a Window of +/-
%d',windowSize),'FontSize',22);
xlabel('Position','FontSize',18);
ylabel(sprintf('Average\nCorrected Ratio'),'FontSize',16);
axis([-100000 max(E(indices1))+100000 min(N(indices1))-0.5 max(N(indices1))+0.5]);
%set to same scale
hold (h2);  %toggle off
hold off;
shg;
figure(2); clf(2);
hold on;
```

```
h1 = subplot(2,1,1);

h2 = subplot(2,1,2);

hold (h1);  %hold toggle on upper

%plot gridlines in upper subplot

subplot(2,1,1);plot([-100000 max(E(indices2))+100000],[1 1],'-.g');

subplot(2,1,1);plot([-100000 max(E(indices2))+100000],[2 2],'-.g');

subplot(2,1,1);plot([-100000 max(E(indices2))+100000],[-1 -1],'-.g');

subplot(2,1,1);plot([-100000 max(E(indices2))+100000],[-2 -2],'-.g');

%plot data points to upper subplot

subplot(2,1,1);plot(E(indices2),N(indices2),' .k','MarkerSize',3);

set(2,'position',[10 120 1250 580],'Name','JCFread_cgh: Chromosome

II','NumberTitle','off');

title(sprintf('C. elegans Chromosome II CGH Data, File: %s',fileStr),'FontSize',22);

xlabel('Position','FontSize',18);

ylabel('Corrected Ratio','FontSize',16);

axis([-100000 max(E(indices2))+100000 min(N(indices2))-0.5 max(N(indices2))+0.5]);

hold (h1);  %hold toggle off upper

%draw second subplot of averages - lower subplot

%determine threshold values

upperThr2 = meanN2+(2*stdN2);lowerThr2 = meanN2-(2*stdN2);

annotation('textbox',[0 0.47 0.1 0.1],'string',sprintf('Data Mean: %4.3f\nData STD:

%4.3f\nUpper Threshold: %4.3f\nLower Threshold: %4.3f\n(Threshold = 2

sd)',meanN2,stdN2,upperThr2,lowerThr2));
```

```
hold (h2);  %toggle on lower
%plot gridlines on lower
subplot(2,1,2);plot([-100000 max(E(indices2))+100000],[1 1],'-.g');
subplot(2,1,2);plot([-100000 max(E(indices2))+100000],[2 2],'-.g');
plot([-100000 max(E(indices2))+100000],[0 0],'-.g');
subplot(2,1,2);plot([-100000 max(E(indices2))+100000],[-1 -1],'-.g');
subplot(2,1,2);plot([-100000 max(E(indices2))+100000],[-2 -2],'-.g');
%add threshold gridlines
subplot(2,1,2);plot([-100000 max(E(indices2))+100000],[upperThr2 upperThr2],'-.r');
subplot(2,1,2);plot([-100000 max(E(indices2))+100000],[lowerThr2 lowerThr2],'-.r');
%plot data to lower subplot
subplot(2,1,2);plot(posavN2,avN2,' .k','MarkerSize',3);
title(sprintf('Average Corrected Ratio Data With a Window of +/-
%d',windowSize),'FontSize',22);
xlabel('Position','FontSize',18);
ylabel(sprintf('Average\nCorrected Ratio'),'FontSize',16);
axis([-100000 max(E(indices2))+100000 min(N(indices2))-0.5 max(N(indices2))+0.5]);
%set to same scale
hold (h2);  %toggle off lower
hold off;
shg;
figure(3); clf(3);
hold on;
```

```
h1 = subplot(2,1,1);
h2 = subplot(2,1,2);
hold (h1);  %hold toggle on upper
%plot gridlines on upper
subplot(2,1,1);plot([-100000 max(E(indices3))+100000],[1 1],'-.g');
subplot(2,1,1);plot([-100000 max(E(indices3))+100000],[2 2],'-.g');
subplot(2,1,1);plot([-100000 max(E(indices3))+100000],[-1 -1],'-.g');
subplot(2,1,1);plot([-100000 max(E(indices3))+100000],[-2 -2],'-.g');
%plot data points on  upper
subplot(2,1,1);plot(E(indices3),N(indices3),' .k','MarkerSize',3);
set(3,'position',[10 100 1250 580],'Name','JCFread_cgh: Chromosome
III','NumberTitle','off');
title(sprintf('C. elegans Chromosome III CGH Data, File: %s',fileStr),'FontSize',22);
xlabel('Position','FontSize',18);
ylabel('Corrected Ratio','FontSize',16);
axis([-100000 max(E(indices3))+100000 min(N(indices3))-0.5 max(N(indices3))+0.5]);
hold (h1);  %hold toggle off upper
%draw second subplot of averages - lower subplot
%determine threshold values
upperThr3 = meanN3+(2*stdN3);lowerThr3 = meanN3-(2*stdN3);
annotation('textbox',[0 0.47 0.1 0.1],'string',sprintf('Data Mean: %4.3f\nData STD:
%4.3f\nUpper Threshold: %4.3f\nLower Threshold: %4.3f\n(Threshold = 2
sd)',meanN3,stdN3,upperThr3,lowerThr3));
```

```
hold (h2);  %toggle on lower
%plot gridlines on lower
subplot(2,1,2);plot([-100000 max(E(indices3))+100000],[1 1],'-.g');
subplot(2,1,2);plot([-100000 max(E(indices3))+100000],[2 2],'-.g');
plot([-100000 max(E(indices3))+100000],[0 0],'-.g');
subplot(2,1,2);plot([-100000 max(E(indices3))+100000],[-1 -1],'-.g');
subplot(2,1,2);plot([-100000 max(E(indices3))+100000],[-2 -2],'-.g');
%add threshold gridlines
subplot(2,1,2);plot([-100000 max(E(indices3))+100000],[upperThr3 upperThr3],'-.r');
subplot(2,1,2);plot([-100000 max(E(indices3))+100000],[lowerThr3 lowerThr3],'-.r');
%plot data to lower subplot
subplot(2,1,2);plot(posavN3,avN3,' .k','MarkerSize',3);
title(sprintf('Average Corrected Ratio Data With a Window of +/-
%d',windowSize),'FontSize',22);
xlabel('Position','FontSize',18);
ylabel(sprintf('Average\nCorrected Ratio'),'FontSize',16);
axis([-100000 max(E(indices3))+100000 min(N(indices3))-0.5 max(N(indices3))+0.5]);
%set to same scale
hold (h2);  %toggle off lower
hold off;
shg;
figure(4); clf(4);
hold on;
```

```
h1 = subplot(2,1,1);
h2 = subplot(2,1,2);
hold (h1);  %hold toggle on upper
%plot gridlines on upper
subplot(2,1,1);plot([-100000 max(E(indices4))+100000],[1 1],'-.g');
subplot(2,1,1);plot([-100000 max(E(indices4))+100000],[2 2],'-.g');
subplot(2,1,1);plot([-100000 max(E(indices4))+100000],[-1 -1],'-.g');
subplot(2,1,1);plot([-100000 max(E(indices4))+100000],[-2 -2],'-.g');
%plot data points on upper
subplot(2,1,1);plot(E(indices4),N(indices4),' .k','MarkerSize',3);
set(4,'position',[10 80 1250 580],'Name','JCFread_cgh: Chromosome
IV','NumberTitle','off');
title(sprintf('C. elegans Chromosome IV CGH Data, File: %s',fileStr),'FontSize',22);
xlabel('Position','FontSize',18);
ylabel('Corrected Ratio','FontSize',16);
axis([-100000 max(E(indices4))+100000 min(N(indices4))-0.5 max(N(indices4))+0.5]);
hold (h1);  %hold toggle off upper
%draw second subplot of averages - lower subplot
%determine threshold values
upperThr4 = meanN4+(2*stdN4);lowerThr4 = meanN4-(2*stdN4);
annotation('textbox',[0 0.47 0.1 0.1],'string',sprintf('Data Mean: %4.3f\nData STD:
%4.3f\nUpper Threshold: %4.3f\nLower Threshold: %4.3f\n(Threshold = 2
sd)',meanN4,stdN4,upperThr4,lowerThr4));
```

```
hold (h2);  %toggle on lower
%plot gridlines on lower
subplot(2,1,2);plot([-100000 max(E(indices4))+100000],[1 1],'-.g');
subplot(2,1,2);plot([-100000 max(E(indices4))+100000],[2 2],'-.g');
plot([-100000 max(E(indices4))+100000],[0 0],'-.g');
subplot(2,1,2);plot([-100000 max(E(indices4))+100000],[-1 -1],'-.g');
subplot(2,1,2);plot([-100000 max(E(indices4))+100000],[-2 -2],'-.g');
%add threshold gridlines
subplot(2,1,2);plot([-100000 max(E(indices4))+100000],[upperThr4 upperThr4],'-.r');
subplot(2,1,2);plot([-100000 max(E(indices4))+100000],[lowerThr4 lowerThr4],'-.r');
%plot data to lower subplot
subplot(2,1,2);plot(posavN4,avN4,' .k','MarkerSize',3);
title(sprintf('Average Corrected Ratio Data With a Window of +/-
%d',windowSize),'FontSize',22);
xlabel('Position','FontSize',18);
ylabel(sprintf('Average\nCorrected Ratio'),'FontSize',16);
axis([-100000 max(E(indices4))+100000 min(N(indices4))-0.5 max(N(indices4))+0.5]);
%set to same scale
hold (h2);  %toggle off lower
hold off;
shg;
figure(5); clf(5);
hold on;
```

```
h1 = subplot(2,1,1);
h2 = subplot(2,1,2);
hold (h1);  %hold toggle on upper
%plot gridlines on upper
subplot(2,1,1);plot([-100000 max(E(indices5))+100000],[1 1],'-.g');
subplot(2,1,1);plot([-100000 max(E(indices5))+100000],[2 2],'-.g');
subplot(2,1,1);plot([-100000 max(E(indices5))+100000],[-1 -1],'-.g');
subplot(2,1,1);plot([-100000 max(E(indices5))+100000],[-2 -2],'-.g');
%plot data points on upper
subplot(2,1,1);plot(E(indices5),N(indices5),' .k','MarkerSize',3);
set(5,'position',[10 60 1250 580],'Name','JCFread_cgh: Chromosome
V','NumberTitle','off');
title(sprintf('C. elegans Chromosome V CGH Data, File: %s',fileStr),'FontSize',22);
xlabel('Position','FontSize',18);
ylabel('Corrected Ratio','FontSize',16);
axis([-100000 max(E(indices5))+100000 min(N(indices5))-0.5 max(N(indices5))+0.5]);
hold (h1);  %hold toggle off upper
%draw second subplot of averages - lower subplot
%determine threshold values
upperThr5 = meanN5+(2*stdN5);lowerThr5 = meanN5-(2*stdN5);
annotation('textbox',[0 0.47 0.1 0.1],'string',sprintf('Data Mean: %4.3f\nData STD:
%4.3f\nUpper Threshold: %4.3f\nLower Threshold: %4.3f\n(Threshold = 2
sd)',meanN5,stdN5,upperThr5,lowerThr5));
```

```
hold (h2);  %toggle on lower

%plot gridlines on lower

subplot(2,1,2);plot([-100000 max(E(indices5))+100000],[1 1],'-.g');

subplot(2,1,2);plot([-100000 max(E(indices5))+100000],[2 2],'-.g');

plot([-100000 max(E(indices5))+100000],[0 0],'-.g');

subplot(2,1,2);plot([-100000 max(E(indices5))+100000],[-1 -1],'-.g');

subplot(2,1,2);plot([-100000 max(E(indices5))+100000],[-2 -2],'-.g');

%add threshold gridlines

subplot(2,1,2);plot([-100000 max(E(indices5))+100000],[upperThr5 upperThr5],'-.r');

subplot(2,1,2);plot([-100000 max(E(indices5))+100000],[lowerThr5 lowerThr5],'-.r');

%plot data to lower subplot

subplot(2,1,2);plot(posavN5,avN5,' .k','MarkerSize',3);

title(sprintf('Average Corrected Ratio Data With a Window of +/-

%d',windowSize),'FontSize',22);

xlabel('Position','FontSize',18);

ylabel(sprintf('Average\nCorrected Ratio'),'FontSize',16);

axis([-100000 max(E(indices5))+100000 min(N(indices5))-0.5 max(N(indices5))+0.5]);

%set to same scale

hold (h2);  %toggle off lower

hold off;

shg;

figure(6); clf(6);

hold on;
```

```
h1 = subplot(2,1,1);
h2 = subplot(2,1,2);
hold (h1);  %hold toggle on upper
%plot gridlines on upper
subplot(2,1,1);plot([-100000 max(E(indicesX))+100000],[1 1],'-.g');
subplot(2,1,1);plot([-100000 max(E(indicesX))+100000],[2 2],'-.g');
subplot(2,1,1);plot([-100000 max(E(indicesX))+100000],[-1 -1],'-.g');
subplot(2,1,1);plot([-100000 max(E(indicesX))+100000],[-2 -2],'-.g');
%plot data points on upper
subplot(2,1,1);plot(E(indicesX),N(indicesX),' .k','MarkerSize',3);
set(6,'position',[10 40 1250 580],'Name','JCFread_cgh: Chromosome
X','NumberTitle','off');
title(sprintf('C. elegans Chromosome X CGH Data, File: %s',fileStr),'FontSize',22);
xlabel('Position','FontSize',18);
ylabel('Corrected Ratio','FontSize',16);
axis([-100000 max(E(indicesX))+100000 min(N(indicesX))-0.5 max(N(indicesX))+0.5]);
hold (h1);  %hold toggle off upper
%draw second subplot of averages - lower subplot
%determine threshold values
upperThrX = meanNX+(2*stdNX);lowerThrX = meanNX-(2*stdNX);
annotation('textbox',[0 0.47 0.1 0.1],'string',sprintf('Data Mean: %4.3f\nData STD:
%4.3f\nUpper Threshold: %4.3f\nLower Threshold: %4.3f\n(Threshold = 2
sd)',meanNX,stdNX,upperThrX,lowerThrX));
```

```
hold (h2);  %toggle on lower

%plot gridlines on lower

subplot(2,1,2);plot([-100000 max(E(indicesX))+100000],[1 1],'-.g');

subplot(2,1,2);plot([-100000 max(E(indicesX))+100000],[2 2],'-.g');

plot([-100000 max(E(indicesX))+100000],[0 0],'-.g');

subplot(2,1,2);plot([-100000 max(E(indicesX))+100000],[-1 -1],'-.g');

subplot(2,1,2);plot([-100000 max(E(indicesX))+100000],[-2 -2],'-.g');

%add threshold gridlines

subplot(2,1,2);plot([-100000 max(E(indicesX))+100000],[upperThrX upperThrX],'-.r');

subplot(2,1,2);plot([-100000 max(E(indicesX))+100000],[lowerThrX lowerThrX],'-.r');

%plot data to lower subplot

subplot(2,1,2);plot(posavNX,avNX,' .k','MarkerSize',3);

title(sprintf('Average Corrected Ratio Data With a Window of +/-

%d',windowSize),'FontSize',22);

xlabel('Position','FontSize',18);

ylabel(sprintf('Average\nCorrected Ratio'),'FontSize',16);

axis([-100000 max(E(indicesX))+100000 min(N(indicesX))-0.5 max(N(indicesX))+0.5]);

%set to same scale

hold (h2);  %toggle off lower

hold off;

shg;
```

Appendix C

Supplemental Material for Chapter 3

Supplemental Data S1

List of ORFs contained in 25 duplications detected by oaCGH in five control and 25 adaptive recovery experimental *C. elegans* lines following 180-212 generations of population expansion under competitive conditions. The duplications are listed in Table 3.1. Duplication breakpoint coordinates and ORFs contained therein are based on Wormbase version WS243.

Duplication in 7B:

Chr IV:6,837,045..6,879,487

Size = 42,443 bp

5 protein-coding genes:

lip-1 (C05B10.1), R13H7.2, *srx-20* (R13H7.1), *srx-19* (T05A12.1), *tre-2* (T05A12.2; partial duplication)

1 pseudogene:

R13H7.3

Duplication in 7B:

Chr V:19,505,848..20,101,145

Size = 595,298 bp

94 protein-coding genes:

Y43F8B.3 (partial duplication), Y43F8B.19, phy-4 (Y43F8B.4), Y43F8B.3, Y43F8B.2, Y43F8B.1, B0399.2, B0399.1, *nlp-25* (Y43F8C.1), Y43F8C.20, *oac-1* (B0399.2), *kcn1-1* (B0399.1), *nlp-25* (Y43F8C.1), *grsp-1* (Y43F8C.20), *nlp-26*

(Y43F8C.2),Y43F8C.3, dyf-19 (Y43F8C.4), Y43F8C.5, Y43F8C.6, Y43F8C.7, *mrps-28* (Y43F8C.8), Y43F8C.9, *dmd-3* (Y43F8C.10), Y43F8C.11, *mrp-7* (Y43F8C.12), Y43F8C.13, *ani-3* (Y43F8C.14), Y43F8C.18, *srv-3* (Y43F8C.19), Y43F8C.15, Y43F8C.23, Y43F8C.16, Y43F8C.17, Y116F11A.6, Y116F11A.3, Y116F11A.1, W04E12.7,*fbxa-131* (W04E12.1), W04E12.2, W04E12.3, W04E12.4, W04E12.5, *clec-49* (W04E12.6), *clec-50* (W04E12.8), W04E12.9, M162.5, M162.15 *fbxa-118* (M162.8),*fbxa-194* (M162.11), *srt-45* (M162.3), *clec-258* (M162.2), M162.7, Y116F11B.2, *daf-28* (Y116F11B.1), Y116F11B.17,*pcp-4* (Y116F11B.3), *srw-38* (Y116F11B.5), Y116F11B.6, Y116F11B.7, Y116F11B.8, Y116F11B.1, *gly-4* (Y116F11B.12), Y116F11B.13,*fars-2* (Y60A3A.13), Y116F11B.14, *chk-2* (Y60A3A.12), Y60A3A.19, Y60A3A.16, *skr-4* (Y60A3A.18), Y60A3A.14, *dhs-24* (Y60A3A.10), Y60A3A.9, Y60A3A.8, Y60A3A.7, *srh-172* (Y60A3A.6), *srh-171* (Y60A3A.5), *srh-173* (Y60A3A.4), *srh-183* (Y60A3A.3), Y60A3A.24, *clec-260* (Y60A3A.2), *sri-67* (Y60A3A.22), Y60A3A.25, *unc-51* (Y60A3A.1), Y60A3A.23, Y60A3A.21, *lgc-55* (Y113G7A.5), Y113G7A.16, *spe-19* (Y113G7A.10), *srh-233* (Y113G7A.1), *ttx-1* (Y113G7A.6),*fre-1* (Y113G7A.8), *dcs-1* (Y113G7A.9), *sec-23* (Y113G7A.3), Y113G7A.15 (partial duplication)

43 pseudogenes:

B0399.t16, B0399.t15, B0399.t14, B0399.t1, B0399.t13, B0399.t12, B0399.t2, B0399.t4, B0399.t3, B0399.t5, B0399.t11, B0399.t10, B0399.t9, B0399.t8, B0399.t7, B0399.t6, Y43F8C.t1, Y43F8C.t9, Y43F8C.t2, Y43F8C.t8, Y43F8C.t3, Y43F8C.t4, Y43F8C.t7, Y43F8C.t6, Y43F8C.t5, Y43F8C.24, Y116F11A.4, W04E12.10, M162.12, M162.13, M162.9, M162.4, M162.14, M162.6, Y116F11B.4, Y116F11B.10, Y60A3A.17, Y60A3A.15, Y60A3A.10, Y60A3A.28, Y60A3A.t1, Y60A3A.t2, Y113G7A.2

Duplication in 7D:

Chr IV:505,050..701,113

Size = 196,064 bp

38 protein-coding genes + 3 tRNA genes:

W03G1.5, *pig-1* (W03G1.6), *asm-3* (W03G1.7), W03G1.2, W03G1.8, *glt-7* (W03G1.1), F09C11.1, F56A11.4, *efn-4* (F56A11.3), F56A11.7, F56A11.5, *gex-2* (F56A11.1), F56A11.6, C18H7.12, C18H7.5, C18H7.6, C18H7.4, C18H7.7, C18H7.11, *srt-59* (C18H7.8), *prmt-4* (C18H7.9), *col-102* (C18H7.3), *inx-18* (C18H7.2), C18H7.1, *nhr-76* (C05G6.2), K11H12.9, K11H12.1, *rpl-15* (K11H12.2), K11H12.8, K11H12.7, K11H12.6, K11H12.11, K11H12.3, K11H12.4, K11H12.10, K11H12.5, *cut1-28* (F41A4.1), *cut1-26* (Y55F3C.7) (partial duplication)

1 pseudogene:

Y55F3C.17

Duplication in 16B*:

Chr V: 19,295,123..19,839,705

Size = 544,583 bp

110 protein-coding genes:

F55C9.6 (partial duplication), *fbxb-60* (F55C9.7), F55C9.14, *fbxb-62* (F55C9.8), *fbxb-63* (F55C9.13), *fbxb-61* (F55C9.10), F55C9.15, F55C9.11, C43D7.8, *fbxb-64* (C43D7.9), *srh-208* (C43D7.6), C43D7.7, *sdz-6* (C43D7.5), C43D7.4, *fbxb-65* (C43D7.2), C14B4.2, Y43F8A.1, Y43F8A.2, Y43F8A.3, *srw-84* (Y43F8A.4), Y43F8A.5, C25F9.8, C25F9.13, *srw-86* (C25F9.7), C25F9.12, C25F9.6, C25F9.10, C25F9.5, C25F9.4, C25F9.9, C25F9.15, C25F9.2, *srw-85* (C25F9.1), C25F9.11, C25F9.16, C25F9.14, M04C3.1, M04C3.2, M04C3.5, Y43F8B.14, Y43F8B.13, Y43F8B.24, Y43F8B.15, Y43F8B.25, Y43F8B.23, Y43F8B.12, Y43F8B.11, Y43F8B.10, Y43F8B.9, Y43F8B.22, Y43F8B.17,Y43F8B.28, Y43F8B.18, Y43F8B.7, Y43F8B.29, *scl-21* (Y43F8B.5), Y43F8B.3, Y43F8B.19, *phy-4* (Y43F8B.4), Y43F8B.2, Y43F8B.1, Y43F8B.20, *oac-1* (B0399.2), *kcnl-1* (B0399.1), *nlp-25* (Y43F8C.1), *grsp-1* (Y43F8C.20), *nlp-26* (Y43F8C.2), Y43F8C.3, *dyf-19* (Y43F8C.4), Y43F8C.5, Y43F8C.6, Y43F8C.7, mrps-28 (Y43F8C.8), Y43F8C.9, *dmd-3* (Y43F8C.10), Y43F8C.11, *mrp-7* (Y43F8C.12), Y43F8C.13, *ani-3* (Y43F8C.14), Y43F8C.18, *srv-3* (Y43F8C.19), Y43F8C.15, Y43F8C.23, Y43F8C.16, Y43F8C.17, Y116F11A.6, Y116F11A.3, Y116F11A.1, W04E12.7, *fbxa-131* (W04E12.1), W04E12.2, W04E12.3, W04E12.4, W04E12.5, *clec-49* (W04E12.6), *clec-50* (W04E12.8), W04E12.9, M162.5, M162.15, *fbxa-118* (M162.8), *fbxa-194* (M162.11), *srt-45* (M162.3), *clec-258* (M162.2), *clec-259* (M162.1), M162.7, Y116F11B.2, *daf-28* (Y116F11B.1), Y116F11B.17, *pcp-4* (Y116F11B.3), *srw-38* (Y116F11B.5)

52 pseudogenes:

C43D7.10, C43D7.11, C43D7.12, C43D7.3, C43D7.1, C14B4.t1, Y43F8A.t1, C25F9.t3, C25F9.t2, C25F9.t1, C25F9.t4, C25F9.t5, Y43F8B.8, Y43F8B.21, Y43F8B.6, B0399.t16, B0399.t15, B0399.t14, B0399.t1, B0399.t13, B0399.t12, B0399.t2, B0399.t3, B0399.t4, B0399.t5, B0399.t11, B0399.t10, B0399.t9, B0399.t8, B0399.t7, B0399.t6, Y43F8C.t1, Y43F8C.t9, Y43F8C.t2, Y43F8C.t8, Y43F8C.26, Y43F8C.t3, Y43F8C.t4, Y43F8C.t7, Y43F8C.t6, Y43F8C.t5, Y43F8C.24, Y116F11A.4, W04E12.10, M162.12, M162.13, M162.9, M162.4, M162.14, M162.6, Y116F11B.4, *srz-35* (Y116F11B.4) (partial duplication)

Duplication in 16C:

Chr IV:9,054,304..9,457,751

Size = 403,448 bp

89 protein-coding genes:

nhr-11 (ZC410.1), *mppb-1* (ZC410.2), *mans-4* (ZC410.3), *twk-8* (ZC410.4), ZC410.5, *1pl-1* (ZC410.7), *icln-1* (C01F6.8), C01F6.9, *cpna-3* (C01F6.1), C01F6.2, C01F6.14, *fem-3* (C01F6.4), *aly-1* (C01F6.5), *nrfl-1* (C01F6.6), *delm-1* (F23B2.3), *daf-10* (F23B2.4), *flp-1* (F23B2.5), *rpb-12* (F23B2.13), *aly-2* (F23B2.6), F23B2.7, F23B2.10, *pcp-3* (F23B2.11), *pcp-2* (F23B2.12), C07C7.3, C07C7.1, C46C2.5, C46C2.7, *wnk-1* (C46C2.1), C46C2.6, C46C2.2, C46C2.3, Y11D7A.3, *rab-28* (Y11D7A.4), Y11D7A.5, Y11D7A.7, Y11D7A.8, Y11D7A.9, Y11D7A.10, *col-120* (Y11D7A.11), *flh-1* (Y11D7A.12), Y11D7A.19, *flh-13* (Y11D7A.13), *hum-9* (Y11D7A.14), *nhr-267* (H22D14.1), *nhr-264* (F14A5.1), F49C12.1, F49C12.2, F49C12.3, F49C12.4, F49C12.5, F49C12.6, F49C12.7, F49C12.9, *rpn-7* (F49C12.8), F49C12.10, F49C12.11, F49C12.12, *vha-17* (F49C12.13), F49C12.14, F49C12.15, *CLEC-183* (T20D3.1), T20D3.2, T20D3.3, T20D3.5, T20D3.6, *vps-26* (T20D3.7), T20D3.8, T20D3.11, C10C5.1, C10C5.2, C10C5.3, C10C5.4, C10C5.5, C10C5.7, *daf-15* (C10C5.6), *col-121* (F56D5.1), F56D5.2, F56D5.3, F56D5.6, F56D5.5, F56D5.9, *srxa-2* (F56D5.10), F59B8.1, *idh-1* (F59B8.2), F38E11.9, *hsp-12.3* (F38E11.1), *hsp-12.6* (F38E11.2), *cut1-17* (F38E11.4), *cpin-1* (F38E11.3)

8 pseudogenes:

F23B2.9, F23B2.8, C46C2.4, Y11D7A.1, Y11D7A.16, *srg-52* (Y11D7A.18), F56D5.4, F56D5.8

Duplication in 16C:

Chr V:800,408..1,103,333

Size = 302,926 bp

57 protein-coding genes:

nhr-270 (R13D11.8), R13D11.11, R13D11.4, R13D11.10, R13D11.3, R13D11.1, *srx-32* (R13D11.9), *srx-31* (F41H8.4), F41H8.2, F41H8.1, K09C6.7, K09C6.10, K09C6.8, K09C6.6, *srbc-13* (K09C6.5), *srbc-12* (K09C6.4), K09C6.3, K09C6.9, K09C6.2, K09C6.1, T02B11.3, T02B11.4, T02B11.9, T02B11.8, *srg-53* (T02B11.1), *srj-38* (T02B11.5), T02B11.6, T02B11.10, *nas-32* (T02B11.7), *fmo-5* (H24K24.5), H24K24.4, H24K24.3, H24K24.2, Y50D4C.2, Y50D4C.3, Y50D4C.6, *sqv-6* (Y50D4C.4), *unc-34* (Y50D4C.1), Y50D4C.5, *ergo-1* (R09A1.1), R09A1.2, R09A1.3, *flp-34* (R09A1.5), *nra-4* (C02E11.1), K10C9.4, K10C9.9, *str-224* (K10C9.8), K10C9.3, *str-67* (K10C9.6), K10C9.7, K10C9.1, Y50D4B.7, Y50D4B.6, *clec-203* (Y50D4B.5), Y50D4B.4, Y50D4B.3, Y50D4B.2 (partial duplication)

2 pseudogenes:

srx-30 (F41H8.3), *str-53* (T02B11.2),

Duplication in 16D:

Chr II: 6,248,049..6,406,772

Size = 158,724 bp

48 protein-coding genes:

T24H7.3 (partial duplication), T24H7.2, *phb-2* (T24H7.1), F13H8.5, F13H8.11, *nmgp-1* (F13H8.4), F13H8.12, F13H8.3, F13H8.8, F13H8.2, *bpl-1* (F13H8.10), F13H8.9, F13H8.1, F13H8.6, F13H8.7, C29F5.3, *mps-1* (C29F5.4), C29F5.5, *sdz-3* (C29F5.2), C29F5.1, C29F5.8, *glb-10* (C29F5.7), C32D5.3, C32D5.4, *sma-6* (C32D5.2), *set-4* (C32D5.5), C32D5.6, C32D5.14, C32D5.7, C32D5.8, C32D5.1, *lgg-1* (C32D5.9), C32D5.10, C32D5.11, C32D5.12, K10B2.4, *ani-2* (K10B2.5), *clec-88* (K10B2.3), K10B2.2, *lin-23* (K10B2.1), F58F12.1, F58F12.4, F58F12.2, F58F12.3, *zig-10* (T25D10.2), *btb-2* (T25D10.5), T25D10.1, *spp-11* (T25D10.3) (partial duplication)

1 pseudogene:

K10B2.t1

Duplication in 16D:

Chr V: 19,746,828..19,885,746

Size = 138,919 bp

26 protein-coding genes:

W04E12.4 (partial duplication), W04E12.5, *clec-49* (W04E12.6), *clec-50* (W04E12.8), W04E12.9, M162.5, M162.15, *fbxa-118* (M162.8), *fbxa-194* (M162.11), *srt-45* (M162.3), *clec-258* (M162.2), *clec-259* (M162.1), M162.7, Y116F11B.2, *daf-28* (Y116F11B.1), Y116F11B.17, *pcp-4* (Y116F11B.3), *srw-38* (Y116F11B.5), Y116F11B.6, Y116F11B.7, Y116F11B.9, Y116F11B.8, Y116F11B.9a, Y116F11B.11, *gly-4* (Y116F11B.12), Y116F11B.13 (partial duplication)

8 pseudogenes:

M162.12, M162.13, M162.9, *srt-46* (M162.4), M162.14, M162.6, *srz-35*

(Y116F11B.4), Y116F11B.10

Duplication in 16E*:

Chr V:19,295,580..19,840,162

Size = 544,583 bp

110 protein-coding genes:

F55C9.6 (partial duplication), *fbxb-60* (F55C9.7), F55C9.14, *fbxb-62* (F55C9.8), *fbxb-63* (F55C9.13), *fbxb-61* (F55C9.10), F55C9.15, F55C9.11, C43D7.8, *fbxb-64* (C43D7.9), *srh-208* (C43D7.6), C43D7.7, *sdz-6* (C43D7.5), C43D7.4, *fbxb-65* (C43D7.2), C14B4.2, Y43F8A.1, Y43F8A.2, Y43F8A.3, *srw-84* (Y43F8A.4), Y43F8A.5, C25F9.8, C25F9.13, *srw-86* (C25F9.7), C25F9.12, C25F9.6, C25F9.10, C25F9.5, C25F9.4, C25F9.9, C25F9.15, C25F9.2, *srw-85* (C25F9.1), C25F9.11, C25F9.16, C25F9.14, M04C3.1, M04C3.2, M04C3.5, Y43F8B.14, Y43F8B.13, Y43F8B.24, Y43F8B.15, Y43F8B.25, Y43F8B.23, Y43F8B.12, Y43F8B.11, Y43F8B.10, Y43F8B.9, Y43F8B.22, Y43F8B.17,Y43F8B.28, Y43F8B.18, Y43F8B.7, Y43F8B.29, *scl-21* (Y43F8B.5), Y43F8B.3, Y43F8B.19, *phy-4* (Y43F8B.4), Y43F8B.2, Y43F8B.1, Y43F8B.20, *oac-1* (B0399.2),kcnl-1 (B0399.1), *nlp-25* (Y43F8C.1), *grsp-1* (Y43F8C.20), *nlp-26* (Y43F8C.2), Y43F8C.3, *dyf-19* (Y43F8C.4), Y43F8C.5, Y43F8C.6, Y43F8C.7, *mrps-28* (Y43F8C.8), Y43F8C.9, *dmd-3* (Y43F8C.10), Y43F8C.11, *mrp-7* (Y43F8C.12), Y43F8C.13, *ani-3* (Y43F8C.14), Y43F8C.18, *srv-3* (Y43F8C.19), Y43F8C.15, Y43F8C.23, Y43F8C.16, Y43F8C.17, Y116F11A.6, Y116F11A.3, Y116F11A.1, W04E12.7, *fbxa-131* (W04E12.1), W04E12.2, W04E12.3, W04E12.4, W04E12.5, *clec-49* (W04E12.6), *clec-50* (W04E12.8), W04E12.9, M162.5, M162.15, *fbxa-118*

(M162.8), *fbxa-194* (M162.11), *srt-45* (M162.3), *clec-258* (M162.2), *clec-259* (M162.1), M162.7, Y116F11B.2, *daf-28* (Y116F11B.1), Y116F11B.17, *pcp-4* (Y116F11B.3), *srw-38* (Y116F11B.5)

52 pseudogenes:

C43D7.10, C43D7.11, C43D7.12, C43D7.3, C43D7.1, C14B4.t1, Y43F8A.t1, C25F9.t3, C25F9.t2, C25F9.t1, C25F9.t4, C25F9.t5, Y43F8B.8, Y43F8B.21, Y43F8B.6, B0399.t16, B0399.t15, B0399.t14, B0399.t1, B0399.t13, B0399.t12, B0399.t2, B0399.t3, B0399.t4, B0399.t5, B0399.t11, B0399.t10, B0399.t9, B0399.t8, B0399.t7, B0399.t6, Y43F8C.t1, Y43F8C.t9, Y43F8C.t2, Y43F8C.t8, Y43F8C.26, Y43F8C.t3, Y43F8C.t4, Y43F8C.t7, Y43F8C.t6, Y43F8C.t5, Y43F8C.24, Y116F11A.4, W04E12.10, M162.12, M162.13, M162.9, M162.4, M162.14, M162.6, Y116F11B.4, *srz-35* (Y116F11B.4) (partial duplication)

Duplication in 19C:

Chr V:7,637,941..7,641,911

Size = 3,971 bp

3 protein-coding genes:

clec-46 (F07C4.9) (partial duplication), *clec-45* (F07C4.2), F07C4.10

0 pseudogenes:

Duplication in 19C:

Chr II:14,037,517.. 14,039,164

Size = 7,572 bp

1 protein-coding genes:

daf-45 (W01G7.1) (partial duplication)

0 pseudogenes:

Duplication in 19E:

Chr X:813,802.. 821,373

Size = 7,572 bp

2 protein-coding genes:

ifd-2 (F25E2.4), *daf-3* (F25E2.5) (partial duplication)

0 pseudogenes:

Duplication in 19E:

Chr X:829,580.. 835,392

Size = 5,813 bp

2 protein-coding genes:

F39H12.2, F39H12.1 (partial duplication)

0 pseudogenes:

Duplication in 50A:

Chr V:19,780,484.. 19,972,052

Size = 191,569 bp

30 protein-coding genes:

srt-45 (M162.3), *clec-258* (M162.2), *clec-259* (M162.1), M162.7, Y116F11B.2, *daf-28* (Y116F11B.1), Y116F11B.17, *pcp-4* (Y116F11B.3), *srw-38* (Y116F11B.5), Y116F11B.6, Y116F11B.7, Y116F11B.9, Y116F11B.8, Y116F11B.9a, Y116F11B.11, *gly-4* (Y116F11B.12), Y116F11B.13, *fars-2* (Y60A3A.13), Y116F11B.14, *chk-2* (Y60A3A.12), Y60A3A.19, Y60A3A.16, *skr-4* (Y60A3A.18), Y60A3A.14, *dhs-24* (Y60A3A.10), Y60A3A.9, Y60A3A.8, Y60A3A.7, *srh-172*

(Y60A3A.6), *srh-171* (Y60A3A.5) (partial duplication)

8 pseudogenes:

srt-46 (M162.4), M162.14, M162.6, *srz-35* (Y116F11B.4), Y116F11B.10, Y60A3A.17, *nhr-240* (Y60A3A.15), Y60A3A.11

Duplication in 50A:

Chr X:8,624,771..9,024,484

Size = 399,714 bp

64 protein-coding genes:

K01A12.3 (partial duplication), *stg-2* (F12D9.1), F12D9.2, *rig-1* (K09E2.4), K09E2.2, K09E2.3, K09E2.1, *frpr-8* (K09E2.5), *jbts-14* (F53A9.4), F53A9.3, F53A9.2, F53A9.1, F53A9.6, F53A9.7, F53A9.8, F53A9.9, *tnt-2* (F53A9.10), EGAP4.1, M02D8.6, M02D8.3, M02D8.2, M02D8.7, *asns-2* (M02D8.4), M02D8.5, M02D8.1, ZK271.4, ZK271.3, *unc-27* (ZK271.2), *chup-1* (ZK271.1), R04E5.7, R04E5.8, R04E5.9, R04E5.2, *ifd-1* (R04E5.10), C28G1.5. C28G1.6, *sec-15* (C28G1.3), C28G1.2, *ubc-23* (C28G1.1), C28G1.10, C28G1.4, C06E2.5, C06E2.9, *ins-9* (C06E2.8), *ubc-22* (C06E2.7), *ubc-21* (C06E2.3), C06E2.1, C06E2.2, C13E3.1, D1009.3, *cyn-8* (D1009.2), *nlp-14* (D1009.4), *acs-2* (D1009.1), *dylt-2* (D1009.5), D1073.1, *aexr-3* (C48C5.3), *nmur-1* (C48C5.1), *twk-18* (C24A3.6), C24A3.4, C24A3.2, C24A3.1, C24A3.9, T25B6.4, T25B6.5

0 pseudogenes:

Duplication in 50B:

Chr V:19,781,064.. 19,972,507

Size = 191,444 bp

30 protein-coding genes:

srt-45 (M162.3), *clec-258* (M162.2), *clec-259* (M162.1), M162.7, Y116F11B.2, *daf-28* (Y116F11B.1), Y116F11B.17, *pcp-4* (Y116F11B.3), *srw-38* (Y116F11B.5), Y116F11B.6, Y116F11B.7, Y116F11B.9, Y116F11B.8, Y116F11B.9a, Y116F11B.11, *gly-4* (Y116F11B.12), Y116F11B.13, *fars-2* (Y60A3A.13), Y116F11B.14, *chk-2* (Y60A3A.12), Y60A3A.19, Y60A3A.16, *skr-4* (Y60A3A.18), Y60A3A.14, *dhs-24* (Y60A3A.10), Y60A3A.9, Y60A3A.8, Y60A3A.7, *srh-172* (Y60A3A.6), *srh-171* (Y60A3A.5)

8 pseudogenes:

srt-46 (M162.4), M162.14, M162.6, *srz-35* (Y116F11B.4), Y116F11B.10, Y60A3A.17, *nhr-240* (Y60A3A.15), Y60A3A.11

Duplication in 50C:

Chr V:19,659,829.. 19,976,506

Size = 316,680 bp

58 protein-coding genes:

Y43F8C.11, *mrp-7* (Y43F8C.11), Y43F8C.13, *ani-3* (Y43F8C.14), Y43F8C.18, *srv-3* (Y43F8C.19), Y43F8C.15, Y43F8C.23, Y43F8C.16, Y43F8C.17, Y116F11A.6, Y116F11A.3, Y116F11A.1, W04E12.7, *fbxa-131* (W04E12.1), W04E12.2, W04E12.3, W04E12.4, W04E12.5a, W04E12.5b, *clec-49* (W04E12.6), *clec-50* (W04E12.8), W04E12.9, M162.5, M162.15, *fbxa-118* (M162.8), *fbxa-194* (M162.11), *srt-45* (M162.3), *clec-258* (M162.2), *clec-259* (M162.1), M162.7, Y116F11B.2, *daf-28* (Y116F11B.1), Y116F11B.17, *pcp-4* (Y116F11B.13), *srw-38* (Y116F11B.5), Y116F11B.6, Y116F11B.7, Y116F11B.8, Y116F11B.9a, Y116F11B.11, *gly-4* (Y116F11B.12), Y116F11B.13, *fars-2* (Y60A3A.13), Y116F11B.14, *cchk-2* (Y60A3A.12), Y60A3A.19, Y60A3A.16, *skr-4* (Y60A3A.18), Y60A3A.14, *dhs-24* (Y60A3A.10), Y60A3A.9, Y60A3A.8, Y60A3A.7, *srh-172* (Y60A3A.6), *srh-171* (Y60A3A.5), *srh-173* (Y60A3A.4), *srh-183* (Y60A3A.3) (partial duplication)

20 pseudogenes:

Y43F8C.t7, Y43F8C.t6, Y43F8C.t5, Y43F8C.24, Y116F11A.4, W04E12.10, M162.12, M162.13, M162.9, *srt-46* (M162.4), M162.14, M162.6, *srz-35* (Y116F11B.4), Y116F11B.10, Y60A3A.17, *nhr-240* (Y60A3A.15), Y60A3A.17, *nhr-240* (Y60A3A.15), Y60A3A.11, Y60A3A.28

Duplication in 50D:

Chr IV:560,240.. 1,024,886

Size = 464,647 bp

84 protein-coding genes:

efn-4 (F56A11.3), F56A11.7, F56A11.5, *gex-2* (F56A11.1), F56A11.6, C18H7.12, C18H7.5, C18H7.6, C18H7.4, C18H7.7, C18H7.11, *srt-59* (C18H7.8), *prmt-4* (C18H7.9), *col-102* (C18H7.3), *inx-18* (C18H7.2), C18H7.1, *nhr-76* (C05G6.2), K11H12.9, K11H12.1, *rpl-15* (K11H12.2), K11H12.8, K11H12.7, K11H12.6, K11H12.11, K11H12.3, K11H12.4, K11H12.10, K11H12.5, *cutl-28* (F41A4.1), *cutl-26* (Y55F3C.7), *clec-164* (Y55F3C.5), Y55F3C.10, Y55F3C.9, *srt-24* (Y55F3C.8), *kvs-5* (Y55F3C.3), *srt-23* (Y55F3C.2), *gst-40* (F56B3.10), *col-103* (F56B3.1), F56B3.2, F56B3.3, F56B3.9, *mrpl-2* (F56B3.8), *ugt-52* (F56B3.7), F56B3.4, F56B3.6, *skr-18* (F56B3.12), F56B3.11, *ech-5* (F56B3.5), *mrpl-46*

(Y55F3BL.1), Y55F3BL.4, Y55F3BL.6, Y55F3BL.2, *madf-1* (Y55F3BR.5), Y55F3BR.10, Y55F3BR.6, Y55F3BR.7, *lgc-33* (Y55F3BR.4), *lem-4* (Y55F3BR.8), Y55F3BR.11, Y55F3BR.2, Y55F3BR.1, *mak-2* (C44C8.6), *fbxc-1* (C44C8.4), *fbxc-9* (C44C8.10), *fbxc-2* (C44C8.3), *fbxc-10* (C44C8.9), *fbxc-4* (C44C8.2), *fbxc-11* (C44C8.8), *fbxc-5* (C44C8.1), *fbxc-12* (C44C8.7), *fbxc-3* (F58H7.8), *fbxc-8* (F58H7.7), F58H7.5, *lgc-30* (F58H7.3), F58H7.1, *faah-3* (F58H7.2), *plx-1* (Y55F3AL.1), *egrh-2* (Y55F3AM.7), Y55F3AM.6, Y55F3AM.5, *immp-2* (Y55F3AM.8), Y55F3AM.9, *atg-3* (Y55F3AM.4), Y55F3AM.3

2 pseudogenes:

Y55F3C.17, Y55F3C.13,

Duplication in 50D:

Chr V:18,703,541..18,723,878

Size = 20,338 bp

4 protein-coding genes:

Y69H2.9 (partial duplication), Y17D7C.1, Y17D7C.6, Y17D7C.2

5 pseudogenes:

Y69H2.18, Y69H2.16, Y17D7C.5, Y17D7C.4, Y17D7C.3

Duplication in 50D:

Chr V:19,780,935..19,966,260

Size = 185,326 bp

30 protein-coding genes:

srt-45 (M162.3), *clec-258* (M162.2), *clec-259* (M162.1), M162.7, Y116F11B.2, *daf-28* (Y116F11B.1), Y116F11B.17, *pcp-4* (Y116F11B.3), *srw-38* (Y116F11B.5), Y116F11B.6, Y116F11B.7, Y116F11B.9, Y116F11B.8, Y116F11B.9a, Y116F11B.11, *gly-4* (Y116F11B.12), Y116F11B.13, *fars-2* (Y60A3A.13), Y116F11B.14, *chk-2* (Y60A3A.12), Y60A3A.19, Y60A3A.16, *skr-4* (Y60A3A.18), Y60A3A.14, *dhs-24* (Y60A3A.10), Y60A3A.9, Y60A3A.8, Y60A3A.7, *srh-172* (Y60A3A.6), *srh-171* (Y60A3A.5) (partial duplication)

8 pseudogenes:

srt-46 (M162.4), M162.14, M162.6, *srz-35* (Y116F11B.4), Y116F11B.10, Y60A3A.17, Y60A3A.15, Y60A3A.11

Duplication in 50E:

Chr II:6,312,598..6,444,674

Size = 132,077 bp

32 protein-coding genes:

C32D5.3, C32D5.4, *sma-6* (C32D5.2), *set-4* (C32D5.5), C32D5.6, C32D5.14, C32D5.7, C32D5.8, C32D5.1, *lgg-1* (C32D5.9), C32D5.10, C32D5.11, C32D5.12, K10B2.4, *ani-2* (K10B2.5), *clec-88* (K10B2.3), K10B2.2, *lin-23* (K10B2.1), F58F12.1, F58F12.4, F58F12.2, F58F12.3, *zig-10* (T25D10.2), *btb-2* (T25D10.5), T25D10.1, *spp-11* (T25D10.3), T25D10.4, K03H9.3, *col-75* (K03H9.2), K03H9.1, *cutl-16* (K06A1.3), K06A1.2 (partial duplication)

1 pseudogene:

K10B2.t1

Duplication in 50E:

Chr V:19,780,952..19,966,162

Size = 185,211 bp

30 protein-coding genes:

srt-45 (M162.3), *clec-258* (M162.2), *clec-259* (M162.1), M162.7, Y116F11B.2, *daf-28* (Y116F11B.1), Y116F11B.17, *pcp-4* (Y116F11B.3), *srw-38* (Y116F11B.5), Y116F11B.6, Y116F11B.7, Y116F11B.9, Y116F11B.8, Y116F11B.9a, Y116F11B.11, *gly-4* (Y116F11B.12), Y116F11B.13, *fars-2* (Y60A3A.13), Y116F11B.14, *chk-2* (Y60A3A.12), Y60A3A.19, Y60A3A.16, *skr-4* (Y60A3A.18), Y60A3A.14, *dhs-24* (Y60A3A.10), Y60A3A.9, Y60A3A.8, Y60A3A.7, *srh-172* (Y60A3A.6), *srh-171* (Y60A3A.5) (partial duplication)

8 pseudogenes:

srt-46 (M162.4), M162.14, M162.6, *srz-35* (Y116F11B.4), Y116F11B.10, Y60A3A.17, *nhr-240* (Y60A3A.15), Y60A3A.11

Duplication in 66C:

Chr V:19,393,526..20,054,330

Size = 660,805 bp

121 protein-coding genes:

C25F9.8, C25F9.13, *srw-86* (C25F9.7), C25F9.12, C25F9.6, C25F9.10, C25F9.5, C25F9.4, C25F9.9, C25F9.15, C25F9.2, *srw-85* (C25F9.1), C25F9.11, C25F9.16, C25F9.14, M04C3.1, Y43F8B.14, Y43F8B.13, Y43F8B.24, Y43F8B.15, Y43F8B.25, Y43F8B.23, Y43F8B.12, Y43F8B.11, Y43F8B.10, Y43F8B.9, Y43F8B.22, Y43F8B.17, Y43F8B.28, Y43F8B.18, Y43F8B.7, Y43F8B.29, *sc1-21* (Y43F8B.5), Y43F8B.3, Y43F8B.19, *phy-4* (Y43F8B.4b), Y43F8B.2, Y43F8B.1, Y43F8C.20, *oac-1* (B0399.2), *kcnl-1* (B0399.1), *nlp-25* (Y43F8C.1), *grsp-1* (Y43F8C.2 0), *nlp-26* (Y43F8C.2), Y43F8C.3, *dyf-19* (Y43F8C.4), Y43F8C.5, Y43F8C.6, Y43F8C.7, *mrps-28* (Y43F8C.8), Y43F8C.9, *dmd-3* (Y43F8C.10), Y43F8C.11, *mrp-7* (Y43F8C.12), Y43F8C.13, *ani-3* (Y43F8C.14), Y43F8C.18, *srv-3* (Y43F8C.19), Y43F8C.15, Y43F8C.23, Y43F8C.16, Y43F8C.17, Y116F11A.6, Y116F11A.3, Y116F11A.1, W04E12.7, *fbxa-131* (W04E12.1), W04E12.2, W04E12.3, W04E12.4, W04E12.5, *clec-49* (W04E12.6), *clec-50* (W04E12.8), W04E12.9, M162.5, M162.15, *fbxa-118* (M162.8), *fbxa-194* (M162.11), *srt-45* (M162.3), *clec-258* (M162.2), *clec-259* (M162.1), M162.7, Y116F11B.2, *daf-28* (Y116F11B.1), Y116F11B.17, *pcp-4* (Y116F11B.3), *srw-38* (Y116F11B.5), Y116F11B.6, Y116F11B.7, Y116F11B.8, Y116F11B.9a, Y116F11B.11, *gly-4* (Y116F11B.12), Y116F11B.13, *fars-2* (Y60A3A.13), Y116F11B.14, *chk-2* (Y60A3A.12), Y60A3A.19, Y60A3A.16, *skr-4* (Y60A3A.18), Y60A3A.14, *dhs-24* (Y60A3A.10), Y60A3A.9, Y60A3A.8, Y60A3A.7, *srh-172* (Y60A3A.6), *srh-171* (Y60A3A.5), *srh-173* (Y60A3A.4), *srh-183* (Y60A3A.3), Y60A3A.24, *clec-260* (Y60A3A.2), *sri-67* (Y60A3A.22), Y60A3A.25, *unc-51* (Y60A3A.1), Y60A3A.23, Y60A3A.21, *lgc-55* (Y113G7A.5), Y113G7A.16, *spe-19* (Y113G7A.10), *srh-233* (Y113G7A.1), *ttx-1* (Y113G7A.6) (partial duplication)

52 pseudogenes:

Y43F8A.t1, C25F9.t3, C25F9.t2, C25F9.t1, C25F9.t4, C25F9.t5, Y43F8B.8, Y43F8B.21, Y43F8B.6, B0399.t16, B0399.t15, B0399.t14, B0399.t1, B0399.t13, B0399.t12, B0399.t2, B0399.t4, B0399.t3, B0399.t5, B0399.t11, B0399.t10, B0399.t9, B0399.t8, B0399.t7, B0399.t6, Y43F8C.t1, Y43F8C.t9, Y43F8C.t2, Y43F8C.t8, Y43F8C.t3, Y43F8C.t4, Y43F8C.t7, Y43F8C.t6, Y43F8C.t5, Y43F8C.24, Y116F11A.4, W04E12.10, M162.12, M162.13, M162.9, M162.4, M162.14, M162.6, Y116F11B.4, Y116F11B.10, Y60A3A.17, Y60A3A.15, Y60A3A.11, Y60A3A.28, Y60A3A.t1, Y60A3A.t2, Y113G7A.2

Duplication in 66E:

Chr V:19,295,300..19,839,882

Size = 544,583 bp

111 protein-coding genes:

F55C9.6 (partial duplication), *fbxb-60* (F55C9.7), F55C9.14, *fbxb-62* (F55C9.8), *fbxb-63* (F55C9.13), *fbxb-61* (F55C9.10), F55C9.11, F55C9.15, C43D7.8, *fbxb-64* (C43D7.9), *srh-208* (C43D7.6), C43D7.7, *sdz-6* (C43D7.5), C43D7.4, *fbxb-65* (C43D7.2), C14B4.2, Y43F8A.1, Y43F8A.2, Y43F8A.3, *srw-84* (Y43F8A.4), Y43F8A.5, C25F9.8, C25F9.13, *srw-86* (C25F9.7), C25F9.12, C25F9.6, C25F9.10, C25F9.5, C25F9.4, C25F9.9, C25F9.15, C25F9.2, *srw-85* (C25F9.1), C25F9.11, C25F9.16, C25F9.14, M04C3.1, M04C3.2, M04C3.5, Y43F8B.14, Y43F8B.13, Y43F8B.24, Y43F8B.15, Y43F8B.25, Y43F8B.23, Y43F8B.12, Y43F8B.11, Y43F8B.10, Y43F8B.9, Y43F8B.22, Y43F8B.17, Y43F8B.28, Y43F8B.18, Y43F8B.7, Y43F8B.29, *scl-21* (Y43F8B.5), Y43F8B.3, Y43F8B.19, *phy-4* (Y43F8B.4), Y43F8B.2, Y43F8B.1, Y43F8B.20, *oac-1* (B0399.2), *kcnl-1* (B0399.1), *nlp-25* (Y43F8C.1), *grsp-1* (Y43F8C.20), *nlp-26* (Y43F8C.2), Y43F8C.3, *dyf-19* (Y43F8C.4), Y43F8C.5, Y43F8C.6, Y43F8C.7, *mrps-28* (Y43F8C.8), Y43F8C.9, *dmd-3* (Y43F8C.10), Y43F8C.11, *mrp-7* (Y43F8C.12), Y43F8C.13, *ani-3* (Y43F8C.14), Y43F8C.18, *srv-3* (Y43F8C.19), Y43F8C.15, Y43F8C.23, Y43F8C.16, Y43F8C.17, Y116F11A.6, Y116F11A.3, Y116F11A.1, W04E12.7, *fbxa-131* (W04E12.1), W04E12.2, W04E12.3, W04E12.4, W04E12.5, *clec-49*

(W04E12.6), *clec-50* (W04E12.8), W04E12.9, M162.5, M162.15, *fbxa-118* (M162.8), *fbxa-194* (M162.11), *srt-5* (M162.3), *clec-258* (M162.2), *clec-259* (M162.1), M162.7, Y116F11B.2, *daf-28* (Y116F11B.1), Y116F11B.17, *pcp-4* (Y116F11B.3), *srw-38* (Y116F11B.5)

50 pseudogenes:

C43D7.10, C43D7.11, C43D7.12, C43D7.3, C43D7.1, C14B4.t1, Y43F8A.t1, C25F9.t3, C25F9.t2, C25F9.t1, C25F9.t4, C25F9.t5, Y43F8B.8, Y43F8B.21, Y43F8B.6, B0399.t16, B0399.t15, B0399.t14, B0399.t1, B0399.t13, B0399.t12, B0399.t2, B0399.t4, B0399.t3, B0399.t5, B0399.t11, B0399.t10, B0399.t9, B0399.t8, B0399.t7, B0399.t6, Y43F8C.t1, Y43F8C.t9, Y43F8C.t2, Y43F8C.t8, Y43F8C.t3, Y43F8C.t4, Y43F8C.t7, Y43F8C.t6, Y43F8C.t5, Y43F8C.24, Y116F11A.4, W04E12.10, M162.12, M162.13, M162.9, M162.4, M162.14, M162.6, Y116F11B.4 (partial duplication)

Duplication in C2:

Chr V:19,295,101..19,839,683

Size = 544,583 bp

111 protein-coding genes:

F55C9.6 (partial duplication), *fbxb-60* (F55C9.7), F55C9.14, *fbxb-62* (F55C9.8), *fbxb-63* (F55C9.13), *fbxb-61* (F55C9.10), F55C9.11, F55C9.15, C43D7.8, *fbxb-64* (C43D7.9), *srh-208* (C43D7.6), C43D7.7, *sdz-6* (C43D7.5), C43D7.4, *fbxb-65* (C43D7.2), C14B4.2, Y43F8A.1, Y43F8A.2, Y43F8A.3, *srw-84* (Y43F8A.4), Y43F8A.5, C25F9.8, C25F9.13, *srw-86* (C25F9.7), C25F9.12, C25F9.6, C25F9.10, C25F9.5, C25F9.4, C25F9.9, C25F9.15, C25F9.2, *srw-85* (C25F9.1), C25F9.11, C25F9.16, C25F9.14, M04C3.1, M04C3.2, M04C3.5, Y43F8B.14, Y43F8B.13, Y43F8B.24, Y43F8B.15, Y43F8B.25, Y43F8B.23, Y43F8B.12, Y43F8B.11, Y43F8B.10, Y43F8B.9, Y43F8B.22, Y43F8B.17, Y43F8B.28, Y43F8B.18, Y43F8B.7, Y43F8B.29, *scl-21* (Y43F8B.5), Y43F8B.3, Y43F8B.19, *phy-4* (Y43F8B.4), Y43F8B.2, Y43F8B.1, Y43F8B.20, *oac-1* (B0399.2), *kcnl-1* (B0399.1), *nlp-25* (Y43F8C.1), *grsp-1* (Y43F8C.20), *nlp-26* (Y43F8C.2), Y43F8C.3, *dyf-19* (Y43F8C.4), Y43F8C.5, Y43F8C.6, Y43F8C.7, *mrps-28* (Y43F8C.8), Y43F8C.9, *dmd-3* (Y43F8C.10), Y43F8C.11, *mrp-7* (Y43F8C.12), Y43F8C.13, *ani-3* (Y43F8C.14), Y43F8C.18, *srv-3* (Y43F8C.19), Y43F8C.15, Y43F8C.23, Y43F8C.16, Y43F8C.17, Y116F11A.6, Y116F11A.3, Y116F11A.1, W04E12.7, *fbxa-131* (W04E12.1), W04E12.2, W04E12.3, W04E12.4, W04E12.5, *clec-49* (W04E12.6), *clec-50* (W04E12.8), W04E12.9, M162.5, M162.15, *fbxa-118* (M162.8), *fbxa-194* (M162.11), *srt-5* (M162.3), *clec-258* (M162.2), *clec-259* (M162.1), M162.7, Y116F11B.2, *daf-28* (Y116F11B.1), Y116F11B.17, *pcp-4* (Y116F11B.3), *srw-38* (Y116F11B.5)

50 pseudogenes:

C43D7.10, C43D7.11, C43D7.12, C43D7.3, C43D7.1, C14B4.t1, Y43F8A.t1, C25F9.t3, C25F9.t2, C25F9.t1, C25F9.t4, C25F9.t5, Y43F8B.8, Y43F8B.21, Y43F8B.6, B0399.t16, B0399.t15, B0399.t14, B0399.t1, B0399.t13, B0399.t12, B0399.t2, B0399.t4, B0399.t3, B0399.t5, B0399.t11, B0399.t10, B0399.t9, B0399.t8, B0399.t7, B0399.t6, Y43F8C.t1, Y43F8C.t9, Y43F8C.t2, Y43F8C.t8, Y43F8C.t3, Y43F8C.t4, Y43F8C.t7, Y43F8C.t6, Y43F8C.t5, Y43F8C.24, Y116F11A.4, W04E12.10, M162.12, M162.13, M162.9, M162.4, M162.14, M162.6, Y116F11B.4

(partial duplication)

Supplemental Data S2

List of ORFs contained in 25 deletions detected by oaCGH in five control and 25 adaptive recovery experimental *C. elegans* lines following 180-212 generations of population expansion under competitive conditions. The deletions are listed in Table 3.2. Deletion breakpoint coordinates and ORFs contained therein are based on Wormbase version WS243.

Deletion in 16A*:

Chr X:817,573..830,086

Size = 12,514 bp

1 protein-coding genes:

daf-3 (F25E2.5)

Deletion in 16D*:

Chr V:7,663,133..7,687,447

Size = 24,315 bp

7 protein-coding genes:

C12D5.5, C12D5.4, C12D5.3, *cyp-33A1* (C12D5.7), *sre-11* (C12D5.11), *nhr-94* (C12D5.8), *nhr-152* (C12D5.2) (partial deletion)

Deletion in 19A*:

Chr X:800,773..827,100

Size = 26,328 bp

5 protein-coding genes:

gtr-1 (F25E2.1), F25E2.2, F25E2.3, *ifd-2* (F25E2.4), *daf-3* (F25E2.5)

Deletion in 19C:

Chr V:7,642,395..7,682,740

Size = 40,346 bp

10 protein-coding genes:

F07C4.11, *str-47* (F07C4.1), F07C4.12, *srh-234* (F07C4.14), *srh-200* (F07C4.13), C12D5.5, C12D5.4, C12D5.3, *cyp-33A1* (C12D5.7), *sre-11* (C12D5.11)

Deletion in 19E:

Chr X:821,499..829,454

Size = 7,956 bp

1 protein-coding genes:

daf-3 (F25E2.5) (partial deletion)

Deletion in 50B:

Chr V:7,650,284..7,693,435

Size = 43,152 bp

12 protein-coding genes:

F07C4.12 (partial deletion), *srh-234* (F07C4.14), *srh-200* (F07C4.13), C12D5.5, C12D5.4, C12D5.3, *cyp-33A1* (C12D5.7), *sre-11* (C12D5.11), *nhr-94* (C12D5.8), *nhr-152* (C12D5.2), C12D5.9, C12D5.10

1 Pseudogene:

str-147 (C12D5.1)

Deletion in 50C:

Chr V:

7,647,125..7,696,096

Size = 48,972 bp

14 protein-coding genes:

str-47 (F07C4.1), F07C4.12, *srh-234* (F07C4.14), *srh-200* (F07C4.13), C12D5.5, C12D5.4, C12D5.3, *cyp-33A1* (C12D5.7), *sre-11* (C12D5.11), *nhr-94* (C12D5.8), *nhr-152* (C12D5.2), C12D5.9, C12D5.10, ZK105.3 (partial deletion)

1 Pseudogene:

str-147 (C12D5.1)

Deletion in 50C:

Chr X:

1,029..273,082

Size = 272,054 bp

35 protein-coding genes:

CE7X_3.1, Y73B3A.1, Y73B3A.20, Y73B3A.18, Y73B3A.3, Y73B3A.4, *elk-2* (Y73B3A.5), *fbxa-221* (Y73B3A.15), *fbxa-222* (Y73B3A.22), *fbxa-16* (Y73B3A.14), Y73B3A.13, Y73B3A.7, *cal-6* (Y73B3A.12), Y73B3A.8, Y73B3A.11, Y73B3A.9, Y73B3A.10, T08D2.1, T08D2.4, T08D2.5, T08D2.6, T08D2.7, T08D2.8, Y73B3B.1, Y73B3B.3, *set-28* (Y73B3B.2), AC8.4, AC8.3, AC8.7, AC8.11, AC8.10, AC8.12, *set-33* (Y108F1.3), *math-43* (Y108F1.4), Y108F1.5 (partial deletion)

18 Pseudogenes:

cTel7X.1, CE7X_3.2, CE7X_3.4, Y35H6.3, Y73B3A.21, Y73B3A.2, Y73B3A.17, Y73B3A.16, Y73B3A.t1, T08D2.9, T08D2.2, T08D2.3, Y73B3B.5, AC8.6, AC8.5, AC8.9, pme-6 (AC8.1), AC8.2

Deletion in 50D*:

Chr V:7,653,667..7,680,465

Size = 26,799 bp

6 protein-coding genes:

srh-234 (F07C4.14) (partial deletion), *srh-200* (F07C4.13), C12D5.5, C12D5.4, C12D5.3, *cyp-33A1* (C12D5.7) (partial deletion)

Deletion in 50D:

Chr X:1,029..295,671

Size = 294,643 bp

38 protein-coding genes:

CE7X_3.1, Y73B3A.1, Y73B3A.20, Y73B3A.18, Y73B3A.16, Y73B3A.3, Y73B3A.4, *elk-2* (Y73B3A.5), *fbxa-221* (Y73B3A.15), *fbxa-222* (Y73B3A.22), *fbxa-16* (Y73B3A.14), Y73B3A.13, Y73B3A.7, *cal-6* (Y73B3A.12), Y73B3A.8, Y73B3A.11, Y73B3A.9, Y73B3A.10, T08D2.1, T08D2.4, T08D2.5, T08D2.6, T08D2.7, T08D2.8, Y73B3B.1, Y73B3B.3, *set-28* (Y73B3B.2), AC8.4, AC8.3, AC8.7, AC8.11, AC8.10, AC8.12, *set-33* (Y108F1.3), *math-43* (Y108F1.4), Y108F1.5, Y108F1.1, Y47C4A.1

20 Pseudogenes:

cTel7X.1, CE7X_3.2, CE7X_3.4, Y35H6.3, Y73B3A.21, Y73B3A.2, Y73B3A.17, Y73B3A.16, Y73B3A.t1, T08D2.9, T08D2.2, T08D2.3, Y73B3B.5, AC8.6, AC8.5, AC8.9, *pme-6* (AC8.1), AC8.2, Y47C4A.t1, Y47C4A.t2

Deletion in 50E*:

Chr V:7,652,044..7,682,914

Size = 30,871 bp

8 protein-coding genes:

F07C4.12B (partial deletion), *srh-234* (F07C4.14), *srh-200* (F07C4.13), C12D5.5, C12D5.4, C12D5.3, *cyp-33A1* (C12D5.7), *sre-11* (C12D5.11) (partial deletion)

Deletion in 66B:

Chr V:15,258,727..15,326,180

Size = 67,454 bp

26 protein-coding genes:

srsx-37 (M01B2.7), M01B2.8, M01B2.10, M01B2.12, M01B2.13, T10H4.13, *srw-22* (T10H4.3), T10H4.4, *srw-16* (T10H4.5), *srw-17* (T10H4.6), *srw-19* (T10H4.8), *srx-51* (T10H4.9), *cyp-34A1* (T10H4.10), *cyp-34A2* (T10H4.11), *str-96* (T10H4.2), *cpr-3* (T10H4.12), *srx-48* (T26H8.2), T26H8.5, T26H8.4, *srz-10* (ZK1037.11), *irld-62* (ZK1037.1), *srt-22* (ZK1037.3), *nhr-246* (ZK1037.4), ZK1037.13, *nhr-247* (ZK1037.5), ZK1037.6 (partial deletion)

5 Pseudogenes:

srw-18 (T10H4.7), T10H4.1, *srx-49* (T26H8.3), ZK1037.12, ZK1037.2

Deletion in 66B*:

Chr X:9,983,441..9,999,107

Size = 15,667 bp

2 protein-coding genes:

F19C6.5, *grk-1* (F19C6.1) (partial deletion)

Deletion in 66D:

Chr V:18,665,661..18,670,354

Size = 4,694 bp

1 protein-coding gene:

Y69H2.10 (partial deletion)

Deletion in 66D:

Chr V:18,701,820..18,725,404

Size = 23,585 bp

3 protein-coding genes:

nhr-241 (Y69H2.8) (partial deletion), Y69H2.9, Y17D7C.1, Y17D7C.6, Y17D7C.2

5 Pseudogenes:

Y69H2.18, Y69H2.16, Y17D7C.5, Y17D7C.4, Y17D7C.3

Deletion in 66D:

Chr X:961,361..963,014

Size = 1,654 bp

1 protein-coding gene:

ncs-1 (C44C1.3) (partial deletion)

Deletion in 66D:

Chr X:7,528,608..7,529,729

Size = 1,122 bp

1 protein-coding gene:

ceh-14 (F46C8.5) (partial deletion)

Deletion in 66E:

Chr X:7,528,608..7,529,729

Size = 1,122 bp

1 protein-coding gene:

ceh-14 (F46C8.5) (partial deletion)

Deletion in C1:

Chr I:15,060,622..15,071,438

Size = 10,817 bp

0 protein-coding genes

4 rRNA genes:

F31C3.7, F31C3.11, F31C3.9, F31C3.8

1 Pseudogene:

rrn-3.56 (F31C3.10)

Deletion in C2:

Chr I:15,060,388..15,071,427

Size = 11,040 bp

0 protein-coding genes

4 rRNA genes:

F31C3.7, F31C3.11, F31C3.9, F31C3.8

1 Pseudogene:

rrn-3.56 (F31C3.10)

Deletion in C3:

Chr II:14,034,460..14,039,471

Size = 5,012 bp

1 protein-coding gene:

daf-5 (W01G7.1) (partial deletion)

Deletion in C3*:

Chr X:7,527,813..7,529,236

Size = 1,424 bp

1 protein-coding gene:

ceh-14 (F46C8.5) (partial deletion)

Deletion in C4:

Chr I:15,060,388..15,071,427

Size = 11,040 bp

0 protein-coding genes

4 rRNA genes:

F31C3.7, F31C3.11, F31C3.9, F31C3.8

1 Pseudogene:

rrn-3.56 (F31C3.10)

Deletion in C5:

Chr I:15,061,973..15,071,438

Size = 9,466 bp

0 protein-coding genes

4 rRNA genes:

F31C3.7, F31C3.11, F31C3.9, F31C3.8

Deletion in C5:

Chr X:823,167..827,286

Size = 4,120 bp

1 protein-coding gene:

daf-3 (F25E2.5) (partial deletion)

Supplemental Data S3

List of ORFs contained in eight overlapping duplications and deletions in experimental *C. elegans* lines following 180-212 generations of population expansion under competitive conditions. Duplication/deletion breakpoint coordinates and ORFs contained therein are based on Wormbase version WS243.

Overlapping Duplications:

1. Chromosome II:

16D: 6,248,049..6,406,772

50E: 6,312,598..6,444,674

Overlapping region: 6,312,598-6,406,772 = 94,175 bp

26 protein-coding ORFs

C32D5.3

Biological process: apoptotic process; embryo development ending in birth or egg hatching; receptor mediated reproduction

C32D5.4

Unclassified

sma-6 (C32D5.2)

Biological process: BMP signaling pathway; body morphogenesis; dauer larval development; defense response to fungus; innate immune response; maintenance of protein location in nucleus; positive regulation of multicellular organism growth; positive regulation of protein catabolic process; positive regulation of transcription from RNA polymerase II promoter; protein phosphorylation; regulation of cell adhesion; regulation of cell morphogenesis; reproduction; tail tip morphogenesis

Cellular component: membrane; plasma membrane

Molecular functions: ATP binding; BMP binding; protein kinase activity; transforming growth factor beta-activated receptor activity; transmembrane receptor protein serine/threonine kinase activity

set-4 (C32D5.5)

Biological process: determination of adult lifespan; embryo development ending in birth or egg-hatching

Molecular functions: protein binding

C32D5.6

Biological process: cellular response to DNA damage stimulus

Molecular functions: protein binding

C32D5.14

Unclassified

C32D5.7

Unclassified

C32D5.8

Biological process: embryo development ending in birth or egg hatching

C32D5.1

Unclassified

lgg-1 (C32D5.9)

Biological process: autophagy; dauer larval development; determination of adult lifespan; embryo development; embryo development ending in birth or egg-hatching; growth; necrotic cell death; positive regulation of necrotic cell death; programmed cell death

Cellular component: autophagic vacuole; autophagic vacuole membrane; cytoplasm; nucleus

C32D5.10

Biological process: nematode larval development; reproduction

Molecular function: metal ion binding; protein binding; zinc ion binding

C32D5.11

Biological process: apoptotic process; lipid storage

Molecular function: protein binding; zinc ion binding

C32D5.12

Biological process: body morphogenesis; embryo development ending in birth or egg-hatching; locomotion; nematode larval development; oxidation-reduction process; steroid biosynthetic process

Molecular function: 3-beta-hydroxy-delta5-steroid dehydrogenase activity; oxidoreductase activity, acting on the CH-OH group of donors, NAD or NADP as receptor

K10B2.4

Unclassified

ani-2 (K10B2.5)

Biological process: gonad development; embryo development ending in birth or egg hatching; multicellular organism reproduction; oogenesis; reproduction; body morphogenesis; apoptotic process

Cellular component: cytoplasm

clec-88 (K10B2.3)

Molecular function: carbohydrate binding

K10B2.2

Biological process: proteolysis

Molecular function: serine-type carboxypeptidase activity

lin-23 (K10B2.1)

Biological process: body morphogenesis; determination of adult lifespan; embryo development ending in birth or egg hatching; hermaphrodite genitalia development; locomotion; negative regulation of cell proliferation; nematode larval development; neuron projection morphogenesis; receptor-mediated endocytosis

Cellular component: cytoplasm; nucleus

Molecular function: protein binding; protein dimerization activity

F58F12.1

Biological process: ATP synthesis coupled proton transport; embryo development ending in birth or egg hatching; nematode larval development; reproduction

Cellular component: mitochondrion; proton-trasport ATP synthase complex, catalytic core F(1)

Molecular function: proton-transporting ATP synthase activity, rotational mechanism

F58F12.4

Unclassified

F58F12.2

Unclassified

F58F12.3

Unclassified

zig-10 (T25D10.2)

Unclassified

btb-2 (T25D10.5)

Molecular function: protein binding

T25D10.1

Unclassified

spp-11 (T25D10.3 – partial duplication)

Unclassified

2. Chromosome IV:

7D: 505,050..701,113

50D: 560,240..1,024,886

Overlapping region: 560,240-701,113 = 140,874 bp

30 protein-coding ORFs

efn-4 (F56A11.3)

Biological process: cell migration involved in gastrulation; embryo development ending in birth or egg hatching; morphogenesis of embryonic epithelium; regulation of cell adhesion; reproduction; tail tip morphogenesis

Cellular component: axon, membrane, neuronal cell body

F56A11.7

Unclassified

F56A11.5

Molecular function: catalytic activity; molybdenum ion binding; pyridoxal phosphate binding

gex-2 (F56A11.1)

Biological process: axon guidance; body morphogenesis; dendrite development; embryo development; embryo development ending in birth or egg hatching; hermaphrodite genitalia development; locomotion; nematode larval development; oviposition

Cellular component: cell junction; cytoplasm

F56A11.6

Biological process: embryo development ending in birth or egg hatching

C18H7.12

Unclassified

C18H7.5

Unclassified

C18H7.6

Unclassified

C18H7.4

Biological process: protein phosphorylation

Molecular function: protein binding; protein kinase activity; protein tyrosine kinase activity

C18H7.7

Unclassified

C18H7.11

Unclassified

srt-59 (C18H7.8)

Unclassified

prmt-4 (C18H7.9)

Unclassified

col-102 (C18H7.3)

Molecular function: structural constituent of cuticle

inx-18 (C18H7.2)

Cellular component: gap junction

C18H7.1

Unclassified

nhr-76 (C05G6.2)

Biological process: regulation of transcription DNA-templated; steroid hormone mediated signaling pathway

Cellular component: nucleus

Molecular function: sequence-specific DNA binding; sequencespecific DNA binding transcription factor activity; steroid hormone receptor activity; zinc ion binding

K11H12.9

Biological process: protein phosphorylation

Molecular function: ATP binding; protein kinase activity

K11H12.1

Unclassified

rpl-15 (K11H12.2)

Biological process: apoptotic process; embryo development ending in birth or egg hatching; molting cycle, collagen and cuticulin-based cuticle; nematode larval development; positive regulation of multicellular organism growth; reproduction; translation

Cellular component: ribosome

Molecular function: structural constituent of ribosome

K11H12.8

Unclassified

K11H12.7

Unclassified

K11H12.6

Unclassified

K11H12.11

Unclassified

K11H12.3

Biological process: reproduction

K11H12.4

Unclassified

K11H12.10

Unclassified

K11H12.5

Unclassified

cutl-28 (F41A4.1)

Biological process: blood coagulation; determination of adult lifespan; proteolysis

Cellular component: extracellular region

Molecular function: protein binding

cutl-26 (Y55F3C.7 - partial duplication)

Unclassified

3. Chromosome V:

7B: 19,505,848..20,101,145

16B*: 19,295,123..19,839,705

16D: 19,746,828..19,885,746

16E*: 19,295,580..19,840,162

50A*: 19,780,484..19,972,052

50B: 19,781,064..19,972,507

50C: 19,659,829..19,976,506

50D: 19,780,935 ..19,966,260

50E: 19,780,952..19,966,162

66C: 19,393,526..20,054,330

66E*: 19,295,300..19,839,882

C2*: 19,295,101..19,839,683

Overlapping region: 19,781,064-19,839,683 = 58,620 bp

11 protein-coding ORFs

fbxa-118 (M162.8 – partial duplication)

Molecular function: protein-binding

fbxa-194 (M162.11)

Molecular function: protein-binding

srt-45 (M162.3)

Unclassified

clec-258 (M162.2)

Molecular function: carbohydrate-binding

clec-259

Molecular function: carbohydrate-binding

M162.7

Unclassified

Y116F11B.2

Unclassified

daf-28 (Y116F11B.1)

Biological processes: dauer larval development; determination of adult lifespan; regulation of insulin receptor signaling pathway; regulation of transcription factor import into nucleus

Cellular components: extracellular regions; extracellular space

Molecular functions: hormone activity; insulin receptor binding

Y116F11B.17

Unclassified

pcp-4 (Y116F11B.3)

Biological processes: proteolysis

Cellular components: membrane raft

Molecular functions: serine-type peptidase activity

srw-38 (Y116F11B.5)

Cellular components: integral component of membranes

Overlapping Deletions:

4. Chromosome X:

16A: 817,573..830,086

19A: 800,773..827,100

19E: 821,499..829,454

C5: 823,167..827,286

Overlapping region: 823,167-827,100= 3,934 bp

1 protein-coding ORFs

daf-3 (F25E2.5 - partial deletion)

Biological process: dauer larval development; negative regulation of transcription from RNA polymerase II promoter; regulation of pharyngeal pumping; regulation of transcription, DNA-templated; transforming growth factor beta receptor signaling pathway

Cellular component: condensed chromosome; cytoplasm; intracellular; nucleus; transcription factor complex

Molecular function: enhancer sequence-specific DNA binding; sequence-specific DNA binding transcription factor activity

5. Chromosome V:

16D: 7,663,133..7,687,447

19C: 7,642,395..7,682,740

50B: 7,650,284..7,693,435

50C: 7,647,125..7,696,096

50D*: 7,653,667..7,680,465

50E*: 7,652,044..7,682,914

Overlapping region: 7,663,133-7,680,465= 17,333 bp

4 protein-coding ORFs

C12D5.5

Unclassified

C12D5.4

Unclassified

C12D5.3

Unclassified

Cyp-33A1 (C12D5.7 - partial deletion)

Biological process: oxiation-reduction process

Molecular function: heme binding; iron ion binding; oxidoreductase activity, acting on donors, with incorporation or reduction of molecular oxygen

6. Chromosome X:

66D: 7,528,608..7,529,729

66E: 7,528,608..7,529,729

C3*: 7,527,813..7,529,236

Overlapping region: 7,528,608-7,529,236= 629 bp

1 protein-coding ORFs

ceh-14 (F46C8.5 - partial deletion)

Biological process: regulation of transcription, DNA-tempated; thermosensory behaviour

Cellular component: nucleus

Molecular function: DNA binding; protein binding; sequencespecific DNA binding; sequence-specific DNA binding transcription factor activity; zinc ion binding

7. Chromosome I:

C1: 15,060,622..15,071,438

C2: 15,060,388..15,071,427

C4: 15,060,388..15,071,427

C5: 15,061,973..15,071,438

Overlapping region: 15,061,973..15,071,427= 9,455 bp

4 rRNA genes

rrn-1.1 (F31C3.7)

rrn-2.1 (F31C3.11)

rrn-3.1 (F31C3.9)

rrn-1.2 (F31C3.8)

8. Chromosome X:

50C: 1,029..273,082

50D: 1,029..295,671

Overlapping region: 1,029..273,082 = 272,054 bp

CE7X_3.1

Unclassified

Y73B3A.1

Unclassified

Y73B3A.20

Unclassified

Y73B3A.18

Biological process: embryo development ending in birth or egg hatching, hermaphrodite genitalia development, reproduction

Y73B3A.3

Biological process: embryo development ending in birth or egg hatching

Y73B3A.4

Unclassified

elk-2 (Y73B3A.5)

Biological process: embryo development ending in birth or egg hatching, hermaphrodite genitalia development, negative regulation of vulval development

fbxa-221 (Y73B3A.15)

Molecular function: protein binding

fbxa-222 (Y73B3A.22)

Molecular function: protein binding

fbxa-16 (Y73B3A.14)

Unclassified

Y73B3A.13

Unclassified

Y73B3A.7

Unclassified

cal-6 (Y73B3A.12)

Biological process: embryo development ending in birth or egg hatching, receptor-mediated endocytosis, reproduction

Molecular function: calcium-ion binding

Y73B3A.8

Unclassified

Y73B3A.11

Unclassified

Y73B3A.9

Unclassified

Y73B3A.10

Biological process: cellular protein metabolic process, reproduction

Molecular function: ATP binding

T08D2.1

Biological process: locomotion, transport

Cellular component: integral component of membrane

T08D2.4

Molecular function: protein binding, zinc ion binding

T08D2.5

Unclassified

T08D2.6

Unclassified

T08D2.7

Biological process: protein phosphorylation

Molecular function: ATP binding, protein binding, protein kinase activity, transferase activity, transferring phosphorus-containing groups

T08D2.8

Molecular function: binding

Y73B3B.1

Molecular function: protein binding

Y73B3B.3

Unclassified

set-28 (Y73B3B.2)

Molecular function: protein binding

AC8.4

Unclassified

AC8.3

Unclassified

AC8.7

Unclassified

AC8.11

Unclassified

AC8.10

Unclassified

AC8.12

Unclassified

set-33 (Y108F1.3)

Biological process: embryo development ending in birth or egg hatching, nematode larval development, RNA interference

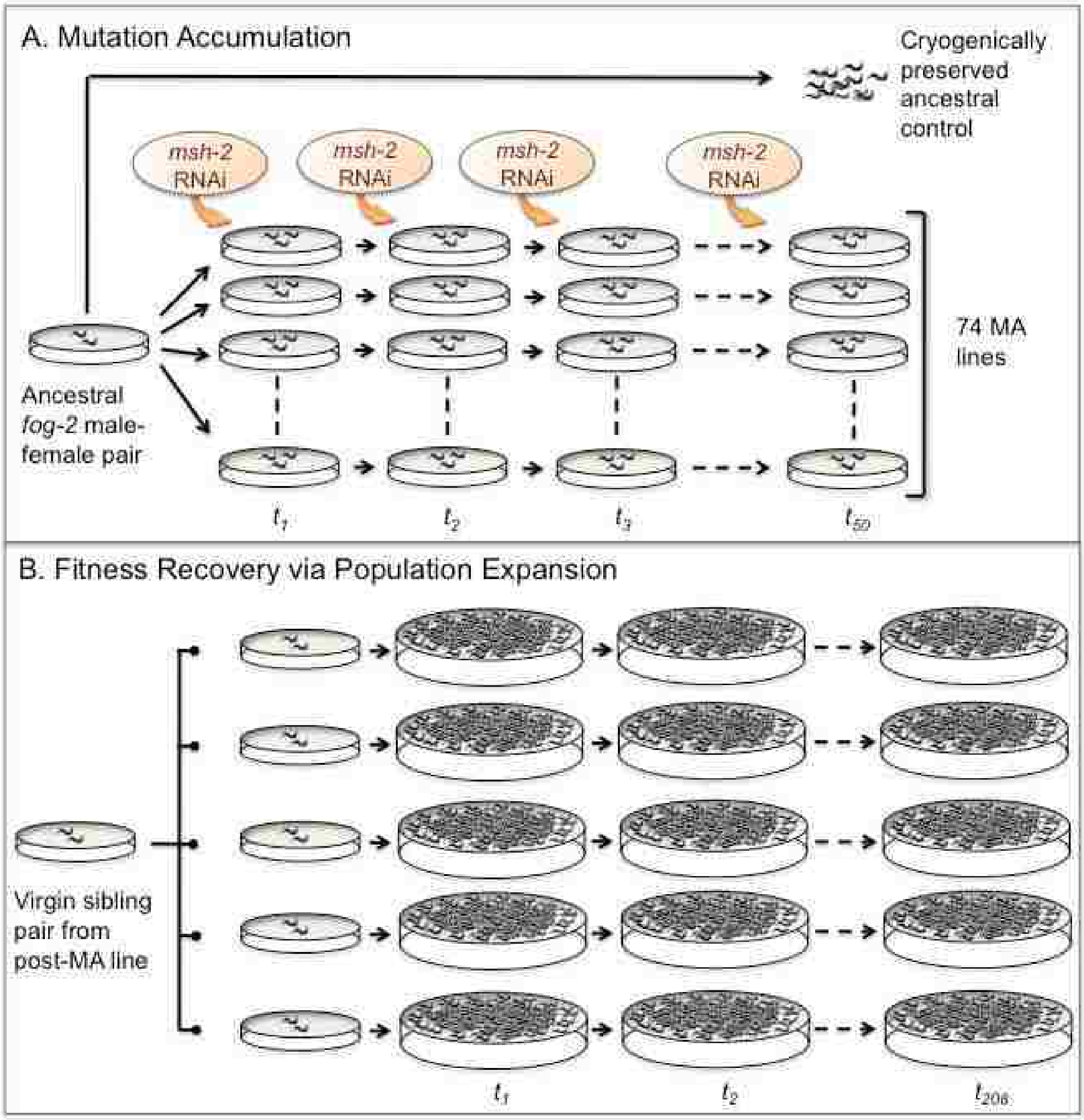
Molecular function: protein binding

math-43 (Y108F1.4)

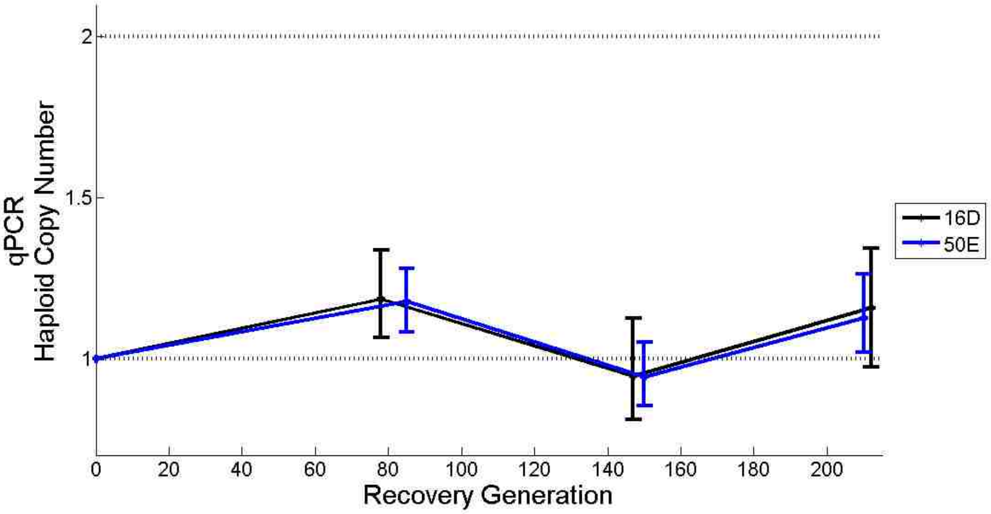
Unclassified

Y108F1.5 (partial deletion)

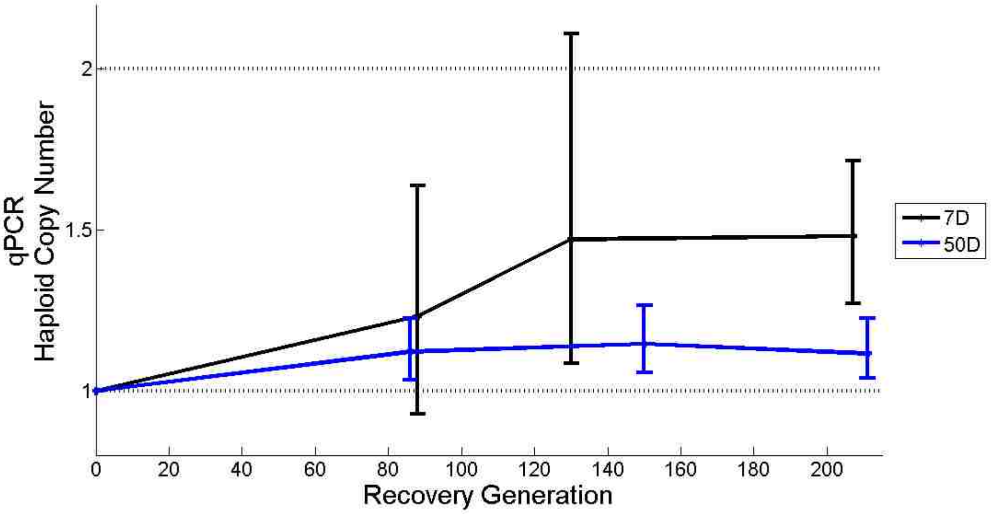
Molecular function: helicase activity, protein binding


A. Mutation Accumulation
Cryogenically preserved ancestral control
msh-2 RNAi
msh-2 RNAi
msh-2 RNAi
msh-2 RNAi
74 MA lines
Ancestral fog-2 male-female pair
t_1
t_2
t_3
t_{50}
B. Fitness Recovery via Population Expansion
Virgin sibling pair from post-MA line
t_1
t_2
t_{208}

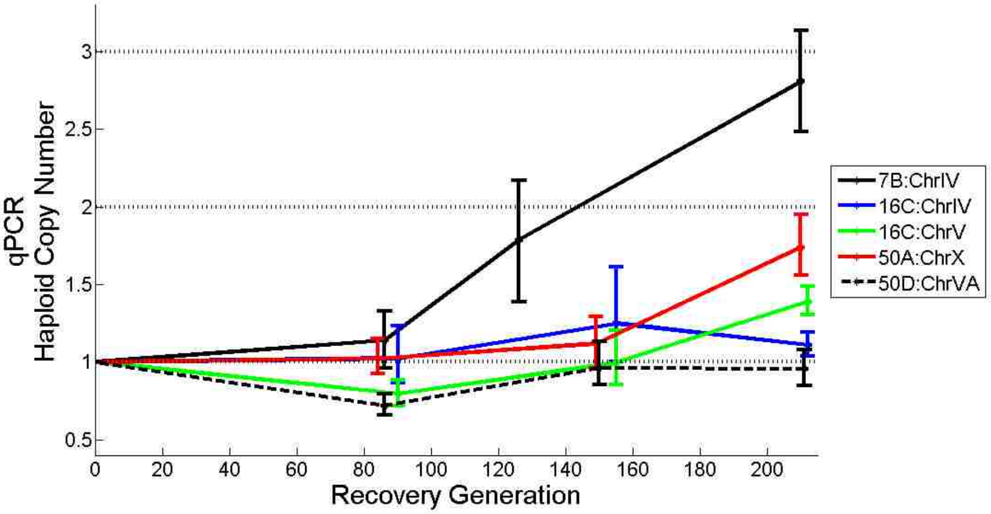
Supplemental Figure C.1. Illustration of Caenorhabditis elegans experimental evolution study with mutation accumulation (MA) and adaptive recovery phases. A. The MA experiment was initiated by establishing 74 lines descended from a single, mated fog-2 female whose additional descendants were expanded for several generations and frozen as ancestral, pre-MA controls. Each generation, the MA regime comprised (i) population bottlenecks of one random female worm and two male siblings (Ne = ~2.67) per generation, and (ii) RNAi-mediated knockdown of the mismatch repair gene msh-2. The MA experiment with msh-2 RNAi was terminated at 50 generations and extant MA lines were subjected to 15 additional generations of full-sib mating without msh-2 RNAi to maximize homozygosity. **B.** To enable fitness/adaptive recovery of mutationally degraded lines, five MA lines (MA7, 16, 19, 50 and 66) exhibiting the greatest decline in fitness following the MA regime were expanded into five sublines (A-E) and independently maintained at large population sizes in the absence of msh-2 RNAi. New generations were established every four days by agar chunk transfers that enabled maintenance of large population sizes across generations. For simplicity, the fitness recovery phase displayed in the figure only depicts population expansion for one MA line and its five descendant sublines, A-E.



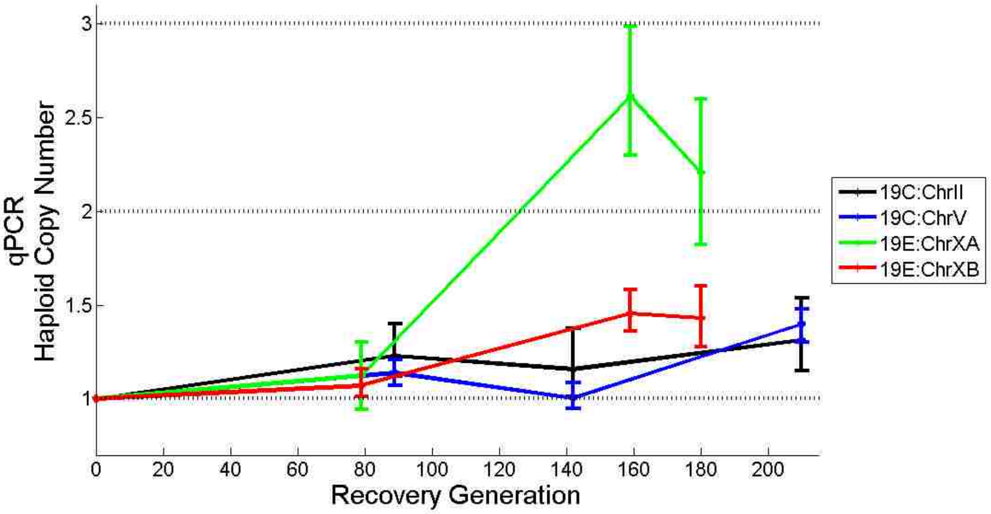
Supplemental Figure C.2. Increase in the frequency of parallel duplication events in two populations containing an overlapping region on Chromosome II. The average copynumber per haploid genome was calculated from qPCR results and is indicated on the vertical axis. The number of recovery generations is indicated on the horizontal axis.



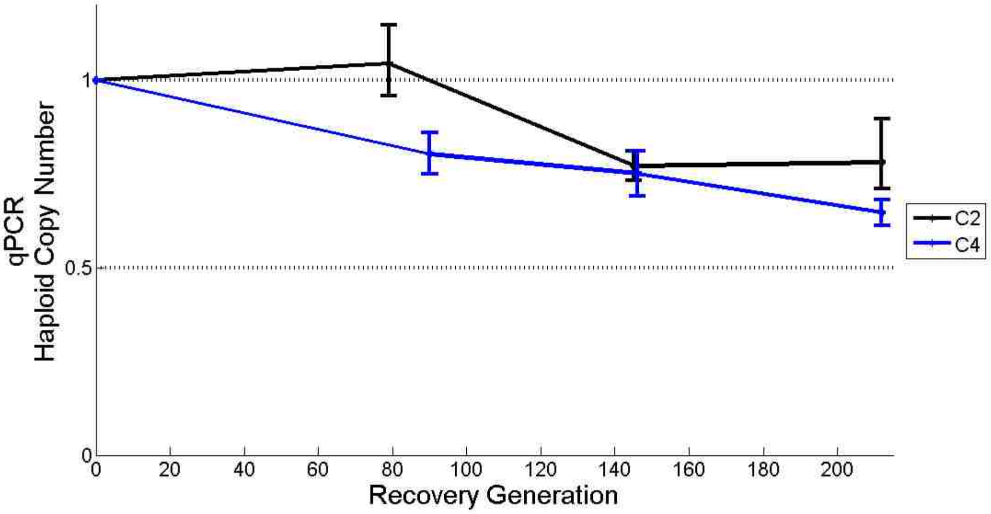
Supplemental Figure C.3. Increase in the frequency of parallel duplication events in two populations containing an overlapping region on Chromosome IV. The average copynumber per haploid genome was calculated from qPCR results and is indicated on the vertical axis. The number of recovery generations is indicated on the horizontal axis.



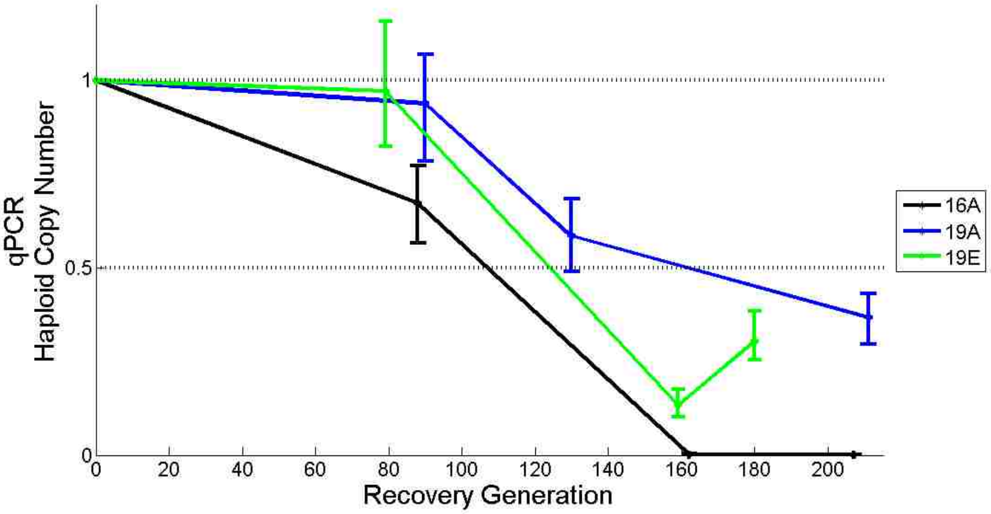
Supplemental Figure C.4. Increase in the frequencies of five unique duplications that lack overlap in their duplication spans. Frequencies of five unique duplications in adaptive recovery populations 7B, 16C, 50A, and 50D. The average copy-number per haploid genome was calculated from qPCR results and is indicated on the vertical axis. The number of recovery generations is indicated on the horizontal axis.



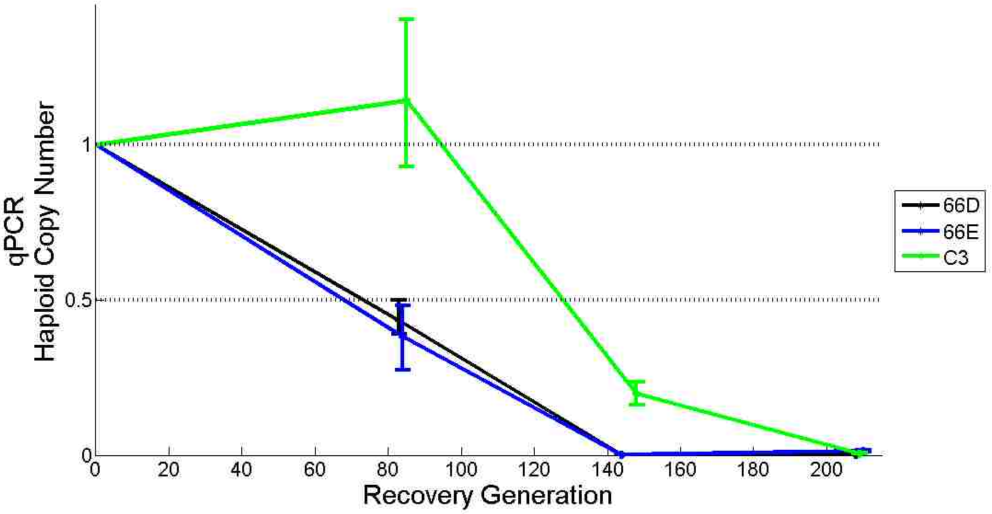
Supplemental Figure C.5. Increase in the frequencies of four unique duplications that lack overlap in their duplication spans. Frequencies of four unique duplications in adaptive recovery populations 19C, and 19E. The average copy-number per haploid genome was calculated from qPCR results and is indicated on the vertical axis. The number of recovery generations is indicated on the horizontal axis.



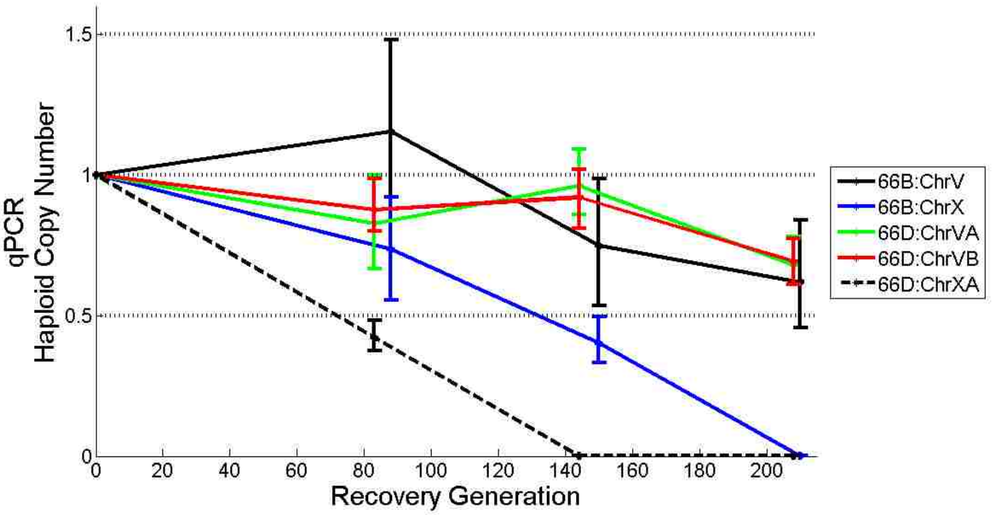
Supplemental Figure C.6. Increase in the frequencies of parallel deletion events in two control populations, C2 and C4, containing an overlapping region on Chromosome I. The average copy-number per haploid genome was calculated from qPCR results and is indicated on the vertical axis. The number of recovery generations is indicated on the horizontal axis. The results show a strong decline in average copy-number of these two independent deletions that were initially detected by oaCGH. The deletions have reached fixation when the average copy-number has reached 0.



Supplemental Figure C.7. Increase in the frequencies of parallel deletion events in three adaptive recovery populations (16A, 19A, and 19E), containing an overlapping region on Chromosome X. The average copy-number per haploid genome was calculated from qPCR results and is indicated on the vertical axis. The number of recovery generations is indicated on the horizontal axis. The results show a strong decline in average copynumber of these three independent deletions that were initially detected by oaCGH. The deletions have reached fixation when the average copy-number has reached 0.



Supplemental Figure C.8. Increase in the frequencies of parallel deletion events in two adaptive recovery populations (66D, and 66E) and one control population (C3) containing another overlapping region on Chromosome X. The average copy-number per haploid genome was calculated from qPCR results and is indicated on the vertical axis. The number of recovery generations is indicated on the horizontal axis. The results show a strong decline in average copy-number of these three independent deletions that were initially detected by oaCGH. The deletions have reached fixation when the average copynumber has reached 0.



Supplemental Figure C.9. Copy-number decreases for five unique deletion events in two adaptive recovery populations (66B, and 66D) that lack overlap in their deletion

spans. The average copy-number per haploid genome was calculated from qPCR results and is indicated on the vertical axis. The generation from which the copy-number was estimated is indicated on the horizontal axis. The deletions have reached fixation when the average copy-number has reached 0.

Appendix D

Code for programs for Chapter 4

Code for JCFqPCR_bs_sim_5_Ct_sets.m

```
%JCFqPCR_bs_sim_5_Ct_sets.m

%James Farslow (jfars@unm.edu)

%30 Jan 2015

%Bootstrap simulation - 5 sets of Ct values

%This program will simulate 4 methods of bootstrapping CIs and also

%calculate CIs based on the square root of the sum of the squares of the group SEmeans.

%The four bootstrap methods are:

%   1 - group means method - should be preferred if n > 10?

%       bootstraps each group to create a new mean for each group

%   2 - Single Paired (dCt) resampling method (mine) - not REST (Pfaffl 2002)

%       Pair the values within test and ref DNA Cts. Bootstrap: randomly select one

%       pair for each, then determine ddCt from them

%   3 - Single Random resampling (mine)

%       Bootstrap: Randomly select one Ct from each group, use those to
%       calculate ddCt
%   4 - Paired Means Method
%       Pair the values within test and ref Cts, bootstrap paired values
%       and calculate new dCt means to determine ddCt
%This simulation is not analyzing biological replicates, only a single
%run of technical replicates.
%This simulation includes only ddCt values, not copy numbers which are
%derived from them.
%This simulation also includes varying ddCt values from -4 to +4
%Note: ddCt = -4 indicates an N-fold increase of 16x
%       ddCt = +4 indicates an N-fold decrease of 1/16 x
clc;
clear;
rng('shuffle');
tic;
%set parameters
cycDiff = [-4 0 4]; %cycle difference range for ddCt
```

```
lencycDiff = length(cycDiff);

reps = input('Number of trials: ');

num = input('Number of Cts: ');

alph = .05;  %set alpha (which is a reserved word), error level

df = (4*num)-4; %degrees of freedom

Tvalue = input(sprintf('Tcrit value from table (df = %d,a = %4.3f): ',df,alph/2));  %with 4 sets of Ct values, df = 4n-4

mn1 = 19;  %fix mu parameter mean for R/R'

mn2 = 19;  %fix mu parameter mean for T/R'

mn3 = 20;  %fix mu parameter mean for R/T'

bootreps = 10000; %number of bootstrap repetitions

%sig = [0.05:0.05:0.5];  %sigma parameter values

sig = [.05:.1:.45];

lenSig = length(sig);

groupCorFac = 1;  %correction factors must be set based on N(Ct)

pairCorFac = 1;  %correction factor of 1 means no correction

count1 = zeros(lenSig,lencycDiff);  %initialize counts for proportion mu capture

count2 = zeros(lenSig,lencycDiff);
```

```
count3 = zeros(lenSig,lencycDiff);

count4 = zeros(lenSig,lencycDiff);

countC = zeros(lenSig,lencycDiff);  %calculated method

countR = zeros(lenSig,lencycDiff);  %is mu within data range?

trData = zeros(reps,15,lenSig,lencycDiff);  %set array for trials data

mnTr1 = zeros(lenSig,lencycDiff);  %mean trials data

mnTr2 = zeros(lenSig,lencycDiff);

mnTr3 = zeros(lenSig,lencycDiff);

mnTr4 = zeros(lenSig,lencycDiff);

mnTrC = zeros(lenSig,lencycDiff);

mnLo1 = zeros(lenSig,lencycDiff);  %mean lower CI

mnLo2 = zeros(lenSig,lencycDiff);

mnLo3 = zeros(lenSig,lencycDiff);

mnLo4 = zeros(lenSig,lencycDiff);

mnLoC = zeros(lenSig,lencycDiff);

mnUp1 = zeros(lenSig,lencycDiff);  %mean upper CI

mnUp2 = zeros(lenSig,lencycDiff);
```

```
mnUp3 = zeros(lenSig,lencycDiff);

mnUp4 = zeros(lenSig,lencycDiff);

mnUpC = zeros(lenSig,lencycDiff);

sdTr1 = zeros(lenSig,lencycDiff);  %trials standard deviation

sdTr2 = zeros(lenSig,lencycDiff);

sdTr3 = zeros(lenSig,lencycDiff);

sdTr4 = zeros(lenSig,lencycDiff);

sdTrC = zeros(lenSig,lencycDiff);

csdTr = zeros(lenSig,lencycDiff);

proCount1 = zeros(lenSig,lencycDiff); %proportion with mu in range

proCount2 = zeros(lenSig,lencycDiff);

proCount3 = zeros(lenSig,lencycDiff);

proCount4 = zeros(lenSig,lencycDiff);

proCountC = zeros(lenSig,lencycDiff);

proCountR = zeros(lenSig,lencycDiff);

mnCount1 = zeros(lencycDiff);  %mean of proportion mu in range

mnCount2 = zeros(lencycDiff);
```

```
mnCount3 = zeros(lencycDiff);

mnCount4 = zeros(lencycDiff);

mnCountR = zeros(lencycDiff);

mnCountC = zeros(lencycDiff);

trueDdct = zeros(lencycDiff,1);

%set warning boxes

figure(100);

clf(100);

text(0,.5,sprintf('Simulation In Progress\n Please Do Not Touch'),'FontSize',40);

set(100,'Position',[250 400 800
200],'Name','Warning','NumberTitle','off','MenuBar','none');

set(gca,'Visible','off');

%figure(100)

figure(101);

clf(101);

text(0,.5,sprintf('Simulation In Progress\n Please Do Not Touch'),'FontSize',40);

set(101,'Position',[-1000 300 800 200],'Name','Extended
Warning','NumberTitle','off','MenuBar','none');
```

```
set(gca,'Visible','off');

%figure(101)

%wait bar diff cycle

wtbr3 = waitbar(0,'cycDiff Loops Complete');

set(wtbr3, 'Position',[15 500 300 50],'Name','cycDiff');

%set cycle diff loop

for indc = 1:lencycDiff

  %wait bar sigma

  wtbr1 = waitbar(0,'Sigma Loops Complete');

  set(wtbr1, 'Position',[332 500 300 50],'Name','Sigma');

  %set mu parameter for T/T' - changes

  mn4 = 20 + cycDiff(indc);

  %set sigma loop

  for inds = 1:lenSig

    %wait bar trials

    wtbr2 = waitbar(0,'Trials Loops Complete');

    set(wtbr2, 'Position',[650 500 300 50],'Name','Trials');
    %set trials loop

    for ind1 = 1:reps

      clc;      %display to command window which sigma and trial iteration

      fprintf('Cycle Difference: %d\n',cycDiff(indc));

      fprintf('Sigma: %4.2f\n',sig(inds));

      fprintf('Trial: %d\n',ind1);

      fprintf('Simulation elapsed time: %d hr %d min %d sec',floor(toc/3600),floor((toc/60)-floor(toc/3600)*60),floor(toc-floor(toc/60)*60));

      %timer box

      figure(102);

      clf(102);

      text(-.1,.5,sprintf('Elapsed Time\n%d hr %d min %d sec\nN(Ct) = %d',floor(toc/3600),floor((toc/60)-floor(toc/3600)*60),floor(toc-floor(toc/60)*60),num),'FontSize',36);

      set(102,'Position',[20 50 450 275],'Name','Timer','NumberTitle','off','MenuBar','none');

      set(gca,'Visible','off');

      %figure(102)
```

```
%get a set of Ct values for this trial
data(1:num,1) = normrnd(mn1,sig(inds),num,1);  %one set of Ct values
data(1:num,2) = normrnd(mn2,sig(inds),num,1);  %one set of Ct values
data(1:num,3) = normrnd(mn3,sig(inds),num,1);  %one set of Ct values
data(1:num,4) = normrnd(mn4,sig(inds),num,1);  %one set of Ct values
%is the mean within the extreme values range?  see method 3 below
%in other words, within the range of the data
%data is arranged as:
   %col 1 - R/R'
   %col 2 - T/R'
   %col 3 - R/T'
   %col 4 - T/T'
%Formula: ddCt = (T/T'-T/R')-(R/T'-R/R')
%                (4 - 2) - (3 - 1)
%bootstrap 1 - group means method
%create bootstrap data array
bsData = zeros(bootreps,1);
```

```
    for ind2 = 1:bootreps
        %randomly select from each Ct group, make new groups
        tempData = zeros(num,4);
        x(1:num,1:4) = randi(num,num,4); %resample selection - uniform distribution from 1 to num
        tempData(1:num,1) = data(x(1:num,1),1);  %use random numbers to resample each group of Ct values
        tempData(1:num,2) = data(x(1:num,2),2);
        tempData(1:num,3) = data(x(1:num,3),3);
        tempData(1:num,4) = data(x(1:num,4),4);
        %calculate means of each group
        mnBsData1 = mean(tempData(1:num,1));
        mnBsData2 = mean(tempData(1:num,2));
        mnBsData3 = mean(tempData(1:num,3));
        mnBsData4 = mean(tempData(1:num,4));
        %calculate ddCt for that bootstrap iteration, store in bootstrap array
        %Formula: ddCt = (T/T'-T/R')-(R/T'-R/R')
        bsData(ind2) = (mnBsData4-mnBsData2)-(mnBsData3-mnBsData1);
```

```
end;
%sort bootstrap array
sortData = sort(bsData);
%get median, upper and lower CIs from bootstrap, store in trial array
%columns 1, 2, and 3
trData(ind1,1,inds,indc) = sortData(floor(bootreps/2)); %median
trData(ind1,2,inds,indc) = sortData(ceil((1-(alph/2))*bootreps)); %upper
trData(ind1,3,inds,indc) = sortData(floor(bootreps*alph/2)); %lower
%test for mu capture, add to count1
%use correction factor
upper = trData(ind1,1,inds,indc)+groupCorFac*(trData(ind1,2,inds,indc)-trData(ind1,1,inds,indc));%mean + corrected (upper - mean)
lower = trData(ind1,1,inds,indc)-groupCorFac*(trData(ind1,1,inds,indc)-trData(ind1,3,inds,indc));%mean - corrected (mean - lower)
%determine true ddCt value
trueDdct(indc) = (mn4 - mn2)-(mn3 - mn1);
if (lower < trueDdct(indc) && upper > trueDdct(indc))  %no change from true ddCt
```

```
   count1(inds,indc) = count1(inds,indc)+1;
end
%boostrap 2 - Single Pairwise Reallocation Method - not REST, see
%Pfaffl (2002), REST program.
%Assumes R/R' and R/T' as well as T/T' and T/R' are paired
%Change data sets to dCt values, then resample pairs randomly to
%obtain ddCt values
%May need to look at this one seperately.  How does the
%distribution change if we change the pairings?
%reset bootstrap array
bsData = zeros(bootreps,1);
for ind2 = 1:bootreps
    %create paired arrays
    dCt1 = data(:,4)-data(:,2);  %T/T'-T/R'
    dCt2 = data(:,3)-data(:,1);  %R/T'-R/R'
    %randomly select one dCt from each group
    x = randi(num,2,1); %resample selection - uniform distribution from 1 to num
```

```
        y1 = dCt1(x(1));
        y2 = dCt2(x(2));
        %calculate ddCt and store in bootstrap array
        bsData(ind2) = y1-y2;
    end
    %sort bootstrap data
    sortData = sort(bsData);
    %get median, upper and lower CIs from bootstrap, store in trials array
    %columns 4, 5, and 6
    trData(ind1,4,inds,indc) = sortData(floor(bootreps/2)); %median
    trData(ind1,5,inds,indc) = sortData(ceil((1-(alph/2))*bootreps)); %upper
    trData(ind1,6,inds,indc) = sortData(floor(bootreps*alph/2)); %lower
    %test for mu capture, add to count2
    %use correction factor
    upper = trData(ind1,4,inds,indc)+pairCorFac*(trData(ind1,5,inds,indc)-trData(ind1,4,inds,indc));%mean + corrected (upper - mean)
    lower = trData(ind1,4,inds,indc)-pairCorFac*(trData(ind1,4,inds,indc)-trData(ind1,6,inds,indc));%mean - corrected (mean - lower)
```

```
if (lower < trueDdct(indc) && upper > trueDdct(indc))  %no change from ddCt

  count2(inds,indc) = count2(inds,indc)+1;

end

%bootstrap 3 - Single Random Reallocation Method

%select one Ct value from each group and calculate ddCt from that

%reset bootstrap array

bsData = zeros(bootreps,1);

for ind2 = 1:bootreps

   %randomly select one Ct from each group

   x = randi(num,4,1); %resample selection - uniform distribution from 1 to num

   y1 = data(x(1),1);  % R/R'

   y2 = data(x(2),2);  % T/R'

   y3 = data(x(3),3);  % R/T'

   y4 = data(x(4),4);  % T/T'

   %calculate ddCt and store in bootstrap array

   %Formula: ddCt = (T/T'-T/R')-(R/T'-R/R')

   bsData(ind2) = (y4-y2)-(y3-y1);
```

```
end

%sort bootstrap data

sortData = sort(bsData);

%get min and max of the sorted data to maximum range of ddCts

%Is mu within this range?

minData = min(sortData);

maxData = max(sortData);

%test for mu capture, add to countR - within range?

if (minData < trueDdct(indc) && maxData > trueDdct(indc))

   countR(inds,indc) = countR(inds,indc)+1;

end

%get median, upper and lower CIs from boostrap, store in trials array

%columns 7, 8, and 9

trData(ind1,7,inds,indc) = sortData(floor(bootreps/2)); %median

trData(ind1,8,inds,indc) = sortData(ceil((1-(alph/2))*bootreps)); %upper

trData(ind1,9,inds,indc) = sortData(floor(bootreps*alph/2)); %lower

%test for mu capture within CI
```

```
if (trData(ind1,9,inds,indc) < trueDdct(indc) && trData(ind1,8,inds,indc) > trueDdct(indc))
  count3(inds,indc) = count3(inds,indc)+1;
end
%bootstrap 4 - Paired Means Method
%Pair the Ct values within test and reference DNA, resample the
%dCt values and calculate the means, then determine ddCt from
%those means
%reset bootstrap array
bsData = zeros(bootreps,1);
for ind2 = 1:bootreps
   %create paired arrays
   dCt1 = data(:,4)-data(:,2);  %T/T'-T/R'
   dCt2 = data(:,3)-data(:,1);  %R/T'-R/R'
   %randomly resample from these arrays
   x = randi(num,num,2); %resample selection - uniform distribution from 1 to num
   rdCt1 = dCt1(x(:,1)); %resampled dCt1
```

```
    rdCt2 = dCt2(x(:,2)); %resampled dCt2
    %get means
    mndCt1 = mean(rdCt1);
    mndCt2 = mean(rdCt2);
    %calculate ddCt and store in bootstrap array
    %Formula: ddCt = (T/T'-T/R')-(R/T'-R/R')
    bsData(ind2) = (mndCt1)-(mndCt2);
end
%sort bootstrap data
sortData = sort(bsData);
%get median, upper and lower CIs from boostrap, store in trials array
%columns 10, 11, and 12
trData(ind1,10,inds,indc) = sortData(floor(bootreps/2)); %median
trData(ind1,11,inds,indc) = sortData(ceil((1-(alph/2))*bootreps)); %upper
trData(ind1,12,inds,indc) = sortData(floor(bootreps*alph/2)); %lower
%test for mu capture, add to countR - within range?
if (trData(ind1,12,inds,indc) < trueDdct(indc) && trData(ind1,11,inds,indc) > trueDdct(indc))
```

```
  count4(inds,indc) = count4(inds,indc)+1;

end

%Calculated method -technical reps

%calculate mean, standard deviation, SEmean, and CIs, store in trials array

%columns 13 (mean), 14 (upper CI), and 15 (lower CI)

%calculate means of each group

mnData1 = mean(data(1:num,1));

mnData2 = mean(data(1:num,2));

mnData3 = mean(data(1:num,3));

mnData4 = mean(data(1:num,4));

%Formula: ddCt = (T/T'-T/R')-(R/T'-R/R')

trData(ind1,13,inds,indc) = (mnData4-mnData2)-(mnData3-mnData1);

%get standard deviation, std for each group, then calculate

%combined std and SEmean

[means,stands] = getCtStats(data);

SET = sqrt(((stands(4)^2)+(stands(2)^2))/num);

SER = sqrt(((stands(3)^2)+(stands(1)^2))/num);
```

```
      SEmean = sqrt(SET^2+SER^2);

      calcCI = SEmean*Tvalue;

      trData(ind1,14,inds,indc) = trData(ind1,10,inds,indc)+calcCI; %upper Ct CI

      trData(ind1,15,inds,indc) = trData(ind1,10,inds,indc)-calcCI; %lower Ct CI

      %test for mu capture, add to countC

      if (trData(ind1,15,inds,indc) < trueDdct(indc) && trData(ind1,14,inds,indc) >
trueDdct(indc))

        countC(inds,indc) = countC(inds,indc)+1;

      end

      %extend waitbar trials

      waitbar(ind1/reps,wtbr2);

    %end trials loop

    end;

    delete(wtbr2);

    %determine count proportion for each sigma per cyc diff

    proCount1(inds,indc) = count1(inds,indc)/reps;

    proCount2(inds,indc) = count2(inds,indc)/reps;

    proCount3(inds,indc) = count3(inds,indc)/reps;
```

```
proCount4(inds,indc) = count4(inds,indc)/reps;

proCountC(inds,indc) = countC(inds,indc)/reps;

proCountR(inds,indc) = countR(inds,indc)/reps;



%get means of trials data

mnTr1(inds,indc) = mean(trData(:,1,inds,indc));

mnTr2(inds,indc) = mean(trData(:,4,inds,indc));

mnTr3(inds,indc) = mean(trData(:,7,inds,indc));

mnTr4(inds,indc) = mean(trData(:,10,inds,indc));

mnTrC(inds,indc) = mean(trData(:,13,inds,indc));

mnUp1(inds,indc) = mean(trData(:,2,inds,indc));

mnUp2(inds,indc) = mean(trData(:,5,inds,indc));

mnUp3(inds,indc) = mean(trData(:,8,inds,indc));

mnUp4(inds,indc) = mean(trData(:,11,inds,indc));

mnUpC(inds,indc) = mean(trData(:,14,inds,indc));

mnLo1(inds,indc) = mean(trData(:,3,inds,indc));

mnLo2(inds,indc) = mean(trData(:,6,inds,indc));
```

```
    mnLo3(inds,indc) = mean(trData(:,9,inds,indc));

    mnLo4(inds,indc) = mean(trData(:,12,inds,indc));

    mnLoC(inds,indc) = mean(trData(:,15,inds,indc));

    %get standard deviation of the mean distributions

    sdTr1(inds,indc) = std(trData(:,1,inds,indc));

    sdTr2(inds,indc) = std(trData(:,4,inds,indc));

    sdTr3(inds,indc) = std(trData(:,7,inds,indc));

    sdTr4(inds,indc) = std(trData(:,10,inds,indc));

    sdTrC(inds,indc) = std(trData(:,13,inds,indc));

    %calculate the standard deviation based on the Ct distribution sigma

    csdTr(inds,indc) = sqrt(2*sig(inds)^2);   %

    %extend waitbar sigma

    waitbar(inds/lenSig,wtbr1);

%end sigma loop

end;

delete(wtbr1);

%extend waitbar cycDiff
```

```
    waitbar(indc/lencycDiff,wtbr3);

    %get means of counts over sigma

    mnCount1(indc) = mean(proCount1(:,indc));

    mnCount2(indc) = mean(proCount2(:,indc));

    mnCount3(indc) = mean(proCount3(:,indc));

    mnCount4(indc) = mean(proCount4(:,indc));

    mnCountC(indc) = mean(proCountC(:,indc));

    mnCountR(indc) = mean(proCountR(:,indc));

%end cycle diff loop

end

runtime = toc;

%fprintf simulation time, number of trials, number of Cts values, means of

%proportion mu capture - can't do the last after adding cycle differences

clc;

fprintf('Simulation elapsed time: %d hr %d min %d
sec\n',floor(runtime/3600),floor((runtime/60)-floor(runtime/3600)*60),floor(runtime-
floor(runtime/60)*60));

fprintf('Total Trials: %d\t\tNumber of Cts: %d\n',reps*lenSig*lencycDiff,num);
```

```
% fprintf('Means of mu capture percent:\n');
% fprintf('\tGroup means method: %5.2f\n',mnCount1*100);
% fprintf('\tPairwise Reallocation Method: %5.2f\n',mnCount2*100);
% fprintf('\tRandom Pairing Method: %5.2f\n',mnCount3*100);
% fprintf('\tCalculated SE Method: %5.2f\n',mnCountC*100);
commandwindow;
%strike the gong to signal finished
load gong.mat;
sound(y, Fs);
close(100);
close(101);
close(102);
delete(wtbr3);
%first get date and convert to string for file names
labelDate = datestr(now,'yyyymmddHH');
%determine fixed sigma and cycDiff points to use - midrange
fixSigma = ceil(lenSig/2);
```

```
fixcycDiff = ceil(lencycDiff/2);

%figure 1 - boxplot of trial means for each sigma, bootstrap 1 (column 1),

%midrange of cycDiff

figure(1);clf(1);

boxData(:,:) = trData(:,1,:,fixcycDiff);

boxplot(boxData);

set(1,'Position',[10 420 625 300],'Name','Group Means Method, Trial Medians');

set(gca,'XTick',[1:lenSig],'XTickLabel',sig,'YGrid','on');

title(sprintf('Group Means Method\nBootstrapped Median ddCt Distributions vs.
Sigma\nNumber of Cts = %d, Cyc Diff = %d',num,cycDiff(fixcycDiff)),'FontSize',16);

xlabel('Sigma','FontSize',14);

ylabel('Bootstrapped ddCt Medians','FontSize',12);

%writeName = strcat('C:\James\Research\Lab_Work\Bootstrapping Protocol\Data\5 Set
Figures and Data\',labelDate,'_5_set_sim_',num2str(num),'_Cts_Figure_1');

%saveas(h1,writeName,'fig');

%figure 2 - boxplot of trial means for each sigma, bootstrap 2 (column 4),

%midrange of cycDiff

figure(2);clf(2);
```

```
boxData(:,:) = trData(:,4,:,fixcycDiff);

boxplot(boxData);

set(2,'Position',[10 420 625 300],'Name','Single Paired Resampling Method, Trial
Medians');

set(gca,'XTick',[1:lenSig],'XTickLabel',sig,'YGrid','on');

title(sprintf('Single Paired Resampling Method\nBootstrapped Median ddCt Distributions
vs. Sigma\nNumber of Cts = %d, Cyc Diff =
%d',num,cycDiff(fixcycDiff)),'FontSize',16);

xlabel('Sigma','FontSize',14);

ylabel('Bootstrapped ddCt Medians','FontSize',12);

%figure 3 - boxplot of trial means for each sigma, bootstrap 3 (column 7),

%midrange of cycDiff

figure(3);clf(3);

boxData(:,:) = trData(:,7,:,fixcycDiff);

boxplot(boxData);

set(3,'Position',[10 420 625 300],'Name','Single Random Resampling Method, Trial
Medians');

set(gca,'XTick',[1:lenSig],'XTickLabel',sig,'YGrid','on');
```

```
title(sprintf('Single Random Resampling Method\nBootstrapped Median ddCt
Distributions vs. Sigma\nNumber of Cts = %d, Cyc Diff =
%d',num,cycDiff(fixcycDiff)),'FontSize',16);

xlabel('Sigma','FontSize',14);

ylabel('Bootstrapped ddCt Medians','FontSize',12);

%figure 4 - boxplot of trial means for each sigma, bootstrap 4 (column 10),

%midrange of cycDiff

figure(4);clf(4);

boxData(:,:) = trData(:,10,:,fixcycDiff);

boxplot(boxData);

set(4,'Position',[10 420 625 300],'Name','Paired Means Method, Trial Medians');

set(gca,'XTick',[1:lenSig],'XTickLabel',sig,'YGrid','on');

title(sprintf('Paired Means Method\nBootstrapped Median ddCt Distributions vs.
Sigma\nNumber of Cts = %d, Cyc Diff = %d',num,cycDiff(fixcycDiff)),'FontSize',16);

xlabel('Sigma','FontSize',14);

ylabel('Bootstrapped ddCt Medians','FontSize',12);

%figure 5 - boxplot of trial means for each sigma, calculated (column 13),

%midrange of cycDiff
```

```
figure(5);clf(5);
boxData(:,:) = trData(:,13,:,fixcycDiff);
boxplot(boxData);
set(5,'Position',[10 420 625 300],'Name','Calculated SE Method, Trial Means');
set(gca,'XTick',[1:lenSig],'XTickLabel',sig,'YGrid','on');
title(sprintf('Calculated SE Method\nCalculated ddCt Distributions vs. Sigma\nNumber of Cts = %d, Cyc Diff = %d',num,cycDiff(fixcycDiff)),'FontSize',16);
xlabel('Sigma','FontSize',14);
ylabel('Calculated ddCts','FontSize',12);
%figure 6 - boxplot of trial means for each cycDiff, bootstrap 1 (column 1),
%midrange of sigma
figure(6);clf(6);
boxData2(:,:) = trData(:,1,fixSigma,:);
boxplot(boxData2);
set(6,'Position',[10 35 625 300],'Name','Group Means Method, Trial Medians');
set(gca,'XTick',[1:lencycDiff],'XTickLabel',cycDiff,'YGrid','on');
title(sprintf('Group Means Method\nBootstrapped Median ddCt Distributions vs. Cycle Difference\nNumber of Cts = %d, Sigma = %3.2f',num,sig(fixSigma)),'FontSize',16);
```

```
xlabel('Cycle Difference','FontSize',14);

ylabel('Bootstrapped ddCt Medians','FontSize',12);

%figure 7 - boxplot of trial means for each cycDiff, bootstrap 2 (column 4),

%midrange of sigma

figure(7);clf(7);

boxData2(:,:) = trData(:,4,fixSigma,:);

boxplot(boxData2);

set(7,'Position',[10 35 625 300],'Name','Single Paired Resampling Method, Trial
Medians');

set(gca,'XTick',[1:lencycDiff],'XTickLabel',cycDiff,'YGrid','on');

title(sprintf('Single Paired Resampling Method\nBootstrapped Median ddCt Distributions
vs. Cycle Difference\nNumber of Cts = %d, Sigma =
%3.2f',num,sig(fixSigma)),'FontSize',16);

xlabel('Cycle Difference','FontSize',14);

ylabel('Bootstrapped ddCt Medians','FontSize',12);

%figure 8 - boxplot of trial means for each cycDiff, bootstrap 3 (column 7),

%midrange of sigma

figure(8);clf(8);
```

```
boxData2(:,:) = trData(:,7,fixSigma,:);

boxplot(boxData2);

set(8,'Position',[10 35 625 300],'Name','Single Random Resampling Method, Trial Medians');

set(gca,'XTick',[1:lencycDiff],'XTickLabel',cycDiff,'YGrid','on');

title(sprintf('Single Random Resampling Method\nBootstrapped Median ddCt Distributions vs. Cycle Difference\nNumber of Cts = %d, Sigma = %3.2f',num,sig(fixSigma)),'FontSize',16);

xlabel('Cycle Difference','FontSize',14);

ylabel('Bootstrapped ddCt Medians','FontSize',12);

%figure 9 - boxplot of trial means for each cycDiff, bootstrap 4 (column 10),

%midrange of sigma

figure(9);clf(9);

boxData2(:,:) = trData(:,10,fixSigma,:);

boxplot(boxData2);

set(9,'Position',[10 35 625 300],'Name','Paired Means Method, Trial Medians');

set(gca,'XTick',[1:lencycDiff],'XTickLabel',cycDiff,'YGrid','on');
```

```
title(sprintf('Paired Means Method\nBootstrapped Median ddCt Distributions vs. Cycle
Difference\nNumber of Cts = %d, Sigma = %3.2f',num,sig(fixSigma)),'FontSize',16);
xlabel('Cycle Difference','FontSize',14);
ylabel('Bootstrapped ddCt Medians','FontSize',12);
%figure 10 - boxplot of trial means for each cycDiff, calculated (column 13),
%midrange of sigma
figure(10);clf(10);
boxData2(:,:) = trData(:,13,fixSigma,:);
boxplot(boxData2);
set(10,'Position',[10 35 625 300],'Name','Calculated SE Method, Trial Means');
set(gca,'XTick',[1:lencycDiff],'XTickLabel',cycDiff,'YGrid','on');
title(sprintf('Calculated SE Method\nCalculated ddCt Distributions vs. Cycle
Difference\nNumber of Cts = %d, Sigma = %3.2f',num,sig(fixSigma)),'FontSize',16);
xlabel('Cycle Difference','FontSize',14);
ylabel('Calculated ddCts','FontSize',12);
%figure 11 - boxplot of upper CI values for each sigma, bootstrap 1 (column
%2), midrange of cycDiff
figure(11);clf(11);
```

```
upperData(:,:) = trData(:,2,:,fixcycDiff);
boxplot(upperData);
set(11,'Position',[650 420 625 300],'Name','Group Means Method, Trial Upper CIs');
set(gca,'XTick',[1:lenSig],'XTickLabel',sig,'YGrid','on');
title(sprintf('Group Means Method\nUpper CI Distribution vs. Sigma\nNumber of Cts =
%d, Cyc Diff = %d',num,cycDiff(fixcycDiff)),'FontSize',16);
xlabel('Sigma','FontSize',14);
ylabel('Bootstrapped CI Upper Bound','FontSize',12);
%figure 12 - boxplot of upper CI values for each sigma, bootstrap 2 (column
%5), midrange of cycDiff
figure(12);clf(12);
upperData(:,:) = trData(:,5,:,fixcycDiff);
boxplot(upperData);
set(12,'Position',[650 420 625 300],'Name','Single Paired Resampling Method, Trial
Upper CIs');
set(gca,'XTick',[1:lenSig],'XTickLabel',sig,'YGrid','on');
title(sprintf('Single Paired Resampling Method\nUpper CI Distribution vs.
Sigma\nNumber of Cts = %d, Cyc Diff = %d',num,cycDiff(fixcycDiff)),'FontSize',16);
```

```
xlabel('Sigma','FontSize',14);

ylabel('Bootstrapped CI Upper Bound','FontSize',12);

%figure 13 - boxplot of upper CI values for each sigma, bootstrap 3 (column

%8), midrange of cycDiff

figure(13);clf(13);

upperData(:,:) = trData(:,8,:,fixcycDiff);

boxplot(upperData);

set(13,'Position',[650 420 625 300],'Name','Single Random Resampling Method, Trial
Upper CIs');

set(gca,'XTick',[1:lenSig],'XTickLabel',sig,'YGrid','on');

title(sprintf('Single Random Resampling Method\nUpper CI Distribution vs.
Sigma\nNumber of Cts = %d, Cyc Diff = %d',num,cycDiff(fixcycDiff)),'FontSize',16);

xlabel('Sigma','FontSize',14);

ylabel('Bootstrapped CI Upper Bound','FontSize',12);

%figure 14 - boxplot of upper CI values for each sigma, bootstrap 4 (column

%11), midrange of cycDiff

figure(14);clf(14);

upperData(:,:) = trData(:,11,:,fixcycDiff);
```

```
boxplot(upperData);
set(14,'Position',[650 420 625 300],'Name','Paired Means Method, Trial Upper CIs');
set(gca,'XTick',[1:lenSig],'XTickLabel',sig,'YGrid','on');
title(sprintf('Paired Means Method\nUpper CI Distribution vs. Sigma\nNumber of Cts =
%d, Cyc Diff = %d',num,cycDiff(fixcycDiff)),'FontSize',16);
xlabel('Sigma','FontSize',14);
ylabel('Bootstrapped CI Upper Bound','FontSize',12);
%figure 15 - boxplot of upper CI values for each sigma, calculated (column
%14), midrange of cycDiff
figure(15);clf(15);
upperData(:,:) = trData(:,14,:,fixcycDiff);
boxplot(upperData);
set(15,'Position',[650 420 625 300],'Name','Calculated SE Method, Trial Upper CIs');
set(gca,'XTick',[1:lenSig],'XTickLabel',sig,'YGrid','on');
title(sprintf('Calculated SE Method\nUpper CI Distribution vs. Sigma\nNumber of Cts =
%d, Cyc Diff = %d',num,cycDiff(fixcycDiff)),'FontSize',16);
xlabel('Sigma','FontSize',14);
ylabel('Calculated CI Upper Bound','FontSize',12);
```

```
%figure 16 - boxplot of upper CI values for each cycDiff, bootstrap 1 (column
%2), midrange of sig
figure(16);clf(16);
upperData2(:,:) = trData(:,2,fixSigma,:);
boxplot(upperData2);
set(16,'Position',[650 35 625 300],'Name','Group Means Method, Trial Upper CIs');
set(gca,'XTick',[1:lencycDiff],'XTickLabel',cycDiff,'YGrid','on');
title(sprintf('Group Means Method\nUpper CI Distribution vs. Cycle
Difference\nNumber of Cts = %d, Sigma = %3.2f',num,sig(fixSigma)),'FontSize',16);
xlabel('Sigma','FontSize',14);
ylabel('Bootstrapped CI Upper Bound','FontSize',12);
%figure 17 - boxplot of upper CI values for each cycDiff, bootstrap 2 (column
%5), midrange of sig
figure(17);clf(17);
upperData2(:,:) = trData(:,5,fixSigma,:);
boxplot(upperData2);
set(17,'Position',[650 35 625 300],'Name','Single Paired Resampling Method, Trial Upper
CIs');
```

```
set(gca,'XTick',[1:lencycDiff],'XTickLabel',cycDiff,'YGrid','on');

title(sprintf('Single Paired Resampling Method\nUpper CI Distribution vs. Cycle
Difference\nNumber of Cts = %d, Sigma = %3.2f',num,sig(fixSigma)),'FontSize',16);

xlabel('Sigma','FontSize',14);

ylabel('Bootstrapped CI Upper Bound','FontSize',12);

%figure 18 - boxplot of upper CI values for each cycDiff, bootstrap 3 (column

%8), midrange of sig

figure(18);clf(18);

upperData2(:,:) = trData(:,8,fixSigma,:);

boxplot(upperData2);

set(18,'Position',[650 35 625 300],'Name','Single Random Resampling Method, Trial
Upper CIs');

set(gca,'XTick',[1:lencycDiff],'XTickLabel',cycDiff,'YGrid','on');

title(sprintf('Single Random Resampling Method\nUpper CI Distribution vs. Cycle
Difference\nNumber of Cts = %d, Sigma = %3.2f',num,sig(fixSigma)),'FontSize',16);

xlabel('Sigma','FontSize',14);

ylabel('Bootstrapped CI Upper Bound','FontSize',12);

%figure 19 - boxplot of upper CI values for each cycDiff, bootstrap 4 (column
%11), midrange of sig

figure(19);clf(19);

upperData2(:,:) = trData(:,11,fixSigma,:);

boxplot(upperData2);

set(19,'Position',[650 35 625 300],'Name','Paired Means Method, Trial Upper CIs');

set(gca,'XTick',[1:lencycDiff],'XTickLabel',cycDiff,'YGrid','on');

title(sprintf('Paired Means Method\nUpper CI Distribution vs. Cycle
Difference\nNumber of Cts = %d, Sigma = %3.2f',num,sig(fixSigma)),'FontSize',16);

xlabel('Sigma','FontSize',14);

ylabel('Bootstrapped CI Upper Bound','FontSize',12);

%figure 20 - boxplot of upper CI values for each cycDiff, calculated (column

%14), midrange of sig

figure(20);clf(20);

upperData2(:,:) = trData(:,14,fixSigma,:);

boxplot(upperData2);

set(20,'Position',[650 35 625 300],'Name','Calculated SE Method, Trial Upper CIs');

set(gca,'XTick',[1:lencycDiff],'XTickLabel',cycDiff,'YGrid','on');
```

```
title(sprintf('Calculated SE Method\nUpper CI Distribution vs. Cycle
Difference\nNumber of Cts = %d, Sigma = %3.2f',num,sig(fixSigma)),'FontSize',16);

xlabel('Sigma','FontSize',14);

ylabel('Calculated CI Upper Bound','FontSize',12);

%figure 21 - 3D bar plot of proportion of trials with mu capture, x = per

%sigma, y = per cycDiff, z = proportion capture, bootstrap 1

figure(21);clf(21);

bar3(proCount1);

set(21,'Position',[125 100 1000 600],'Name','Group Means Method, Capture Proportion');

axis([.5 lencycDiff+.5 .5 lenSig+.5 0 1.1]);

set(gca,'XTick',[1:lencycDiff],'XTickLabel',cycDiff);

set(gca,'YTick',[1:lenSig],'YTickLabel',sig);

title(sprintf('Group Means Method\nMu Capture Proportion vs. Cycle Difference and
Sigma \nNumber of Cts = %d',num),'FontSize',16);

xlabel('Cycle Difference','FontSize',14,'Rotation',25);

ylabel('Sigma','FontSize',14,'Rotation',-35);

zlabel('Capture Proportion','FontSize',14);

%figure 22 - 3D bar plot of proportion of trials with mu capture, x = per
```

```
%sigma, y = per cycDiff, z = proportion capture, bootstrap 2
figure(22);clf(22);
bar3(proCount2);
set(22,'Position',[125 100 1000 600],'Name','Single Paired Resampling Method, Capture Proportion');
axis([.5 lencycDiff+.5 .5 lenSig+.5 0 1.1]);
set(gca,'XTick',[1:lencycDiff],'XTickLabel',cycDiff);
set(gca,'YTick',[1:lenSig],'YTickLabel',sig);
title(sprintf('Single Paired Resampling Method\nMu Capture Proportion vs. Cycle Difference and Sigma \nNumber of Cts = %d',num),'FontSize',16);
xlabel('Cycle Difference','FontSize',14,'Rotation',25);
ylabel('Sigma','FontSize',14,'Rotation',-35);
zlabel('Capture Proportion','FontSize',14);
%figure 23 - 3D bar plot of proportion of trials with mu capture, x = per
%sigma, y = per cycDiff, z = proportion capture, bootstrap 3
figure(23);clf(23);
bar3(proCount3);
```

```
set(23,'Position',[125 100 1000 600],'Name','Single Random Resampling Method,
Capture Proportion');
axis([.5 lencycDiff+.5 .5 lenSig+.5 0 1.1]);
set(gca,'XTick',[1:lencycDiff],'XTickLabel',cycDiff);
set(gca,'YTick',[1:lenSig],'YTickLabel',sig);
title(sprintf('Single Random Resampling Method\nMu Capture Proportion vs. Cycle
Difference and Sigma \nNumber of Cts = %d',num),'FontSize',16);
xlabel('Cycle Difference','FontSize',14,'Rotation',25);
ylabel('Sigma','FontSize',14,'Rotation',-35);
zlabel('Capture Proportion','FontSize',14);
%figure 24 - 3D bar plot of proportion of trials with mu capture, x = per
%sigma, y = per cycDiff, z = proportion capture, bootstrap 4
figure(24);clf(24);
bar3(proCount4);
set(24,'Position',[125 100 1000 600],'Name','Paired Means Method, Capture Proportion');
axis([.5 lencycDiff+.5 .5 lenSig+.5 0 1.1]);
set(gca,'XTick',[1:lencycDiff],'XTickLabel',cycDiff);
set(gca,'YTick',[1:lenSig],'YTickLabel',sig);
```

```
title(sprintf('Paired Means Method\nMu Capture Proportion vs. Cycle Difference and
Sigma \nNumber of Cts = %d',num),'FontSize',16);

xlabel('Cycle Difference','FontSize',14,'Rotation',25);

ylabel('Sigma','FontSize',14,'Rotation',-35);

zlabel('Capture Proportion','FontSize',14);

%figure 25 - 3D bar plot of proportion of trials with mu capture, x = per

%sigma, y = per cycDiff, z = proportion capture, calculated

figure(25);clf(25);

bar3(proCountC);

set(25,'Position',[125 100 1000 600],'Name','Calculated Method, Capture Proportion');

axis([.5 lencycDiff+.5 .5 lenSig+.5 0 1.1]);

set(gca,'XTick',[1:lencycDiff],'XTickLabel',cycDiff);

set(gca,'YTick',[1:lenSig],'YTickLabel',sig);

title(sprintf('Calculated Method\nMu Capture Proportion vs. Cycle Difference and Sigma
\nNumber of Cts = %d',num),'FontSize',16);

xlabel('Cycle Difference','FontSize',14,'Rotation',25);

ylabel('Sigma','FontSize',14,'Rotation',-35);

zlabel('Capture Proportion','FontSize',14);
```

```
%figure 26 - 3D bar plot of proportion of trials with mu in range, x = per
%sigma, y = per cycDiff, z = proportion in range,
figure(26);clf(26);
bar3(proCountR);
set(26,'Position',[125 100 1000 600],'Name','Proportion Within Range');
axis([.5 lencycDiff+.5 .5 lenSig+.5 0 1.1]);
set(gca,'XTick',[1:lencycDiff],'XTickLabel',cycDiff);
set(gca,'YTick',[1:lenSig],'YTickLabel',sig);
title(sprintf('Proportion Mu Within Range vs. Cycle Difference and Sigma \nNumber of Cts = %d',num),'FontSize',16);
xlabel('Cycle Difference','FontSize',14,'Rotation',25);
ylabel('Sigma','FontSize',14,'Rotation',-35);
zlabel('Capture Proportion','FontSize',14);
%calculate correlation of trueDdct capture proportion and sigma
%Do it with capture proportion and cycDiff
%bootstrap 1
corStat1A = zeros(lencycDiff,2);  %gets Pearson correlation and p-value
X1 = zeros(lenSig,2);           %relative to sigma with cycDiff fixed
```

```
X1(:,2) = sig';
for ind1 = 1:lencycDiff
    X1(:,1) = proCount1(:,ind1);
    [RHO1 PVAL1] = corr(X1);
    corStat1A(ind1,:) = [RHO1(2,1) PVAL1(2,1)];
end
corStat1B = zeros(lenSig,2);  %gives Pearson correlation and p-value
X1 = zeros(lencycDiff,2);           %relative to cycDiff with sigma fixed
X1(:,2) = cycDiff';
for ind1 = 1:lenSig
    X1(:,1) = proCount1(ind1,:);
    [RHO1 PVAL1] = corr(X1);
    corStat1B(ind1,:) = [RHO1(2,1) PVAL1(2,1)];
end
%bootstrap 2
corStat2A = zeros(lencycDiff,2);  %gets Pearson correlation and p-value
X1 = zeros(lenSig,2);           %relative to sigma with cycDiff fixed
```

```
X1(:,2) = sig';

for ind1 = 1:lencycDiff

    X1(:,1) = proCount2(:,ind1);

    [RHO1 PVAL1] = corr(X1);

    corStat2A(ind1,:) = [RHO1(2,1) PVAL1(2,1)];

end

corStat2B = zeros(lenSig,2);  %gives Pearson correlation and p-value

X1 = zeros(lencycDiff,2);           %relative to cycDiff with sigma fixed

X1(:,2) = cycDiff';

for ind1 = 1:lenSig

    X1(:,1) = proCount2(ind1,:);

    [RHO1 PVAL1] = corr(X1);

    corStat2B(ind1,:) = [RHO1(2,1) PVAL1(2,1)];

end

%bootstrap 3

corStat3A = zeros(lencycDiff,2);  %gets Pearson correlation and p-value

X1 = zeros(lenSig,2);           %relative to sigma with cycDiff fixed
```

```
X1(:,2) = sig';

for ind1 = 1:lencycDiff

   X1(:,1) = proCount3(:,ind1);

   [RHO1 PVAL1] = corr(X1);

   corStat3A(ind1,:) = [RHO1(2,1) PVAL1(2,1)];

end

corStat3B = zeros(lenSig,2);  %gives Pearson correlation and p-value

X1 = zeros(lencycDiff,2);           %relative to cycDiff with sigma fixed

X1(:,2) = cycDiff';

for ind1 = 1:lenSig

   X1(:,1) = proCount3(ind1,:);

   [RHO1 PVAL1] = corr(X1);

   corStat3B(ind1,:) = [RHO1(2,1) PVAL1(2,1)];

end

%bootstrap 4

corStat4A = zeros(lencycDiff,2);  %gets Pearson correlation and p-value

X1 = zeros(lenSig,2);           %relative to sigma with cycDiff fixed
```

```
X1(:,2) = sig';
for ind1 = 1:lencycDiff
   X1(:,1) = proCount4(:,ind1);
   [RHO1 PVAL1] = corr(X1);
   corStat4A(ind1,:) = [RHO1(2,1) PVAL1(2,1)];
end
corStat4B = zeros(lenSig,2);  %gives Pearson correlation and p-value
X1 = zeros(lencycDiff,2);          %relative to cycDiff with sigma fixed
X1(:,2) = cycDiff';
for ind1 = 1:lenSig
   X1(:,1) = proCount4(ind1,:);
   [RHO1 PVAL1] = corr(X1);
   corStat4B(ind1,:) = [RHO1(2,1) PVAL1(2,1)];
end
%calculated
corStatCA = zeros(lencycDiff,2);  %gets Pearson correlation and p-value
X1 = zeros(lenSig,2);          %relative to sigma with cycDiff fixed
```

```
X1(:,2) = sig';

for ind1 = 1:lencycDiff

   X1(:,1) = proCountC(:,ind1);

   [RHO1 PVAL1] = corr(X1);

   corStatCA(ind1,:) = [RHO1(2,1) PVAL1(2,1)];

end

corStatCB = zeros(lenSig,2);  %gives Pearson correlation and p-value

X1 = zeros(lencycDiff,2);           %relative to cycDiff with sigma fixed

X1(:,2) = cycDiff';

for ind1 = 1:lenSig

   X1(:,1) = proCountC(ind1,:);

   [RHO1 PVAL1] = corr(X1);

   corStatCB(ind1,:) = [RHO1(2,1) PVAL1(2,1)];

end

%store data in excel file, one for each bootstrap style and one for

%calculated, one worksheet for each ddCt (cycDiff)

%Group means method data - bootstrap 1
```

```
for ind1 = 1:lencycDiff

   %write to xls file

   writeData1 =
{'ddCt',cycDiff(ind1);'Trials',reps;'Bootstraps',bootreps;'Cts',num;'Alpha',alph;

      'True ddCt',trueDdct(ind1);'Mean Proportion in CI',mnCount1(ind1);

      'Mean Proportion in Range',mnCountR(ind1);

      'Pearson Correlation of Proportion in CI (capture) and Sigma at Fixed ddCts:','ddCt';

      '','Correlation';'','P-Value'};

   writeData2 = [cycDiff;corStat1A(:,1)';corStat1A(:,2)'];

   writeData3 = {'Pearson Correlation of Proportion in CI and ddCt at Fixed
Sigmas:','Sigma';

      '','Correlation';'','P-Value'};

   writeData4 = [sig;corStat1B(:,1)';corStat1B(:,2)'];

   writeData5 = {'Ct Distribution Sigma';'Calculated ddCt Std Dev from Ct Distribution
Std Devs';'Actual ddCt Std Dev from Trial ddCts';'Proportion in CI';'Proportion in
Range';'Trial Means';

      'Trial Lower CIs';'Trial Upper CIs'};
```

```
    writeData6 =
[sig;csdTr(:,ind1)';sdTr1(:,ind1)';proCount1(:,ind1)';proCountR(:,ind1)';mnTr1(:,ind1)';m
nLo1(:,ind1)';mnUp1(:,ind1)'];
    writeName = strcat(labelDate,'_5_set_sim_group_means_',num2str(num),'_Cts','.xls');
    xlswrite(writeName,writeData1,ind1,'A1');
    xlswrite(writeName,writeData2,ind1,'C9');
    xlswrite(writeName,writeData3,ind1,'A12');
    xlswrite(writeName,writeData4,ind1,'C12');
    xlswrite(writeName,writeData5,ind1,'A16');
    xlswrite(writeName,writeData6,ind1,'B16');
end
%Single paired resampling method data - bootstrap 2
for ind1 = 1:lencycDiff
    %write to xls file
    writeData1 =
{'ddCt',cycDiff(ind1);'Trials',reps;'Bootstraps',bootreps;'Cts',num;'Alpha',alph;
        'True ddCt',trueDdct(ind1);'Mean Proportion in CI',mnCount2(ind1);
        'Mean Proportion in Range',mnCountR(ind1);
```

```
      'Pearson Correlation of Proportion in CI (capture) and Sigma at Fixed ddCts:','ddCt';
      ",'Correlation';",'P-Value'};
   writeData2 = [cycDiff;corStat2A(:,1)';corStat2A(:,2)'];
   writeData3 = {'Pearson Correlation of Proportion in CI and ddCt at Fixed
Sigmas:','Sigma';
      ",'Correlation';",'P-Value'};
   writeData4 = [sig;corStat2B(:,1)';corStat2B(:,2)'];
   writeData5 = {'Ct Distribution Sigma';'Calculated ddCt Std Dev from Ct Distribution
Std Devs';'Actual ddCt Std Dev from Trial ddCts';'Proportion in CI';'Proportion in
Range';'Trial Means';
      'Trial Lower CIs';'Trial Upper CIs'};
   writeData6 =
[sig;csdTr(:,ind1)';sdTr2(:,ind1)';proCount2(:,ind1)';proCountR(:,ind1)';mnTr2(:,ind1)';m
nLo2(:,ind1)';mnUp2(:,ind1)'];
   writeName =
strcat(labelDate,'_5_set_sim_single_paired_resampling_',num2str(num),'_Cts','.xls');
   xlswrite(writeName,writeData1,ind1,'A1');
   xlswrite(writeName,writeData2,ind1,'C9');
   xlswrite(writeName,writeData3,ind1,'A12');
```

```
    xlswrite(writeName,writeData4,ind1,'C12');

    xlswrite(writeName,writeData5,ind1,'A16');

    xlswrite(writeName,writeData6,ind1,'B16');

end

%Single random resampling method data - bootstrap 3

for ind1 = 1:lencycDiff

    %write to xls file

    writeData1 =

{'ddCt',cycDiff(ind1);'Trials',reps;'Bootstraps',bootreps;'Cts',num;'Alpha',alph;

        'True ddCt',trueDdct(ind1);'Mean Proportion in CI',mnCount3(ind1);

        'Mean Proportion in Range',mnCountR(ind1);

        'Pearson Correlation of Proportion in CI (capture) and Sigma at Fixed ddCts:','ddCt';

        ",'Correlation';",'P-Value'};

    writeData2 = [cycDiff;corStat3A(:,1)';corStat3A(:,2)'];

    writeData3 = {'Pearson Correlation of Proportion in CI and ddCt at Fixed

Sigmas:','Sigma';

        ",'Correlation';",'P-Value'};

    writeData4 = [sig;corStat3B(:,1)';corStat3B(:,2)'];
```

```
   writeData5 = {'Ct Distribution Sigma';'Calculated ddCt Std Dev from Ct Distribution
Std Devs';'Actual ddCt Std Dev from Trial ddCts';'Proportion in CI';'Proportion in
Range';'Trial Means';
        'Trial Lower CIs';'Trial Upper CIs'};
   writeData6 =
[sig;csdTr(:,ind1)';sdTr3(:,ind1)';proCount3(:,ind1)';proCountR(:,ind1)';mnTr3(:,ind1)';m
nLo3(:,ind1)';mnUp3(:,ind1)'];
   writeName =
strcat(labelDate,'_5_set_sim_single_random_resampling_',num2str(num),'_Cts','.xls');
   xlswrite(writeName,writeData1,ind1,'A1');
   xlswrite(writeName,writeData2,ind1,'C9');
   xlswrite(writeName,writeData3,ind1,'A12');
   xlswrite(writeName,writeData4,ind1,'C12');
   xlswrite(writeName,writeData5,ind1,'A16');
   xlswrite(writeName,writeData6,ind1,'B16');
end
%Paired means method data - bootstrap 4
for ind1 = 1:lencycDiff
   %write to xls file
```

```
  writeData1 =
{'ddCt',cycDiff(ind1);'Trials',reps;'Bootstraps',bootreps;'Cts',num;'Alpha',alph;
      'True ddCt',trueDdct(ind1);'Mean Proportion in CI',mnCount4(ind1);
      'Mean Proportion in Range',mnCountR(ind1);
      'Pearson Correlation of Proportion in CI (capture) and Sigma at Fixed ddCts:','ddCt';
      '','Correlation';'','P-Value'};
  writeData2 = [cycDiff;corStat4A(:,1)';corStat4A(:,2)'];
  writeData3 = {'Pearson Correlation of Proportion in CI and ddCt at Fixed
Sigmas:','Sigma';
      '','Correlation';'','P-Value'};
  writeData4 = [sig;corStat4B(:,1)';corStat4B(:,2)'];
  writeData5 = {'Ct Distribution Sigma';'Calculated ddCt Std Dev from Ct Distribution
Std Devs';'Actual ddCt Std Dev from Trial ddCts';'Proportion in CI';'Proportion in
Range';'Trial Means';
      'Trial Lower CIs';'Trial Upper CIs'};
  writeData6 =
[sig;csdTr(:,ind1)';sdTr4(:,ind1)';proCount4(:,ind1)';proCountR(:,ind1)';mnTr4(:,ind1)';m
nLo4(:,ind1)';mnUp4(:,ind1)'];
  writeName = strcat(labelDate,'_5_set_sim_paired_means_',num2str(num),'_Cts','.xls');
```

```
    xlswrite(writeName,writeData1,ind1,'A1');

    xlswrite(writeName,writeData2,ind1,'C9');

    xlswrite(writeName,writeData3,ind1,'A12');

    xlswrite(writeName,writeData4,ind1,'C12');

    xlswrite(writeName,writeData5,ind1,'A16');

    xlswrite(writeName,writeData6,ind1,'B16');

end

%Calculated method data

for ind1 = 1:lencycDiff

    %write to xls file

    writeData1 =

{'ddCt',cycDiff(ind1);'Trials',reps;'Bootstraps',bootreps;'Cts',num;'Alpha',alph;

        'True ddCt',trueDdct(ind1);'Mean Proportion in CI',mnCountC(ind1);

        'Mean Proportion in Range',mnCountR(ind1);

        'Pearson Correlation of Proportion in CI (capture) and Sigma at Fixed ddCts:','ddCt';

        ",'Correlation';",'P-Value'};

    writeData2 = [cycDiff;corStatCA(:,1)';corStatCA(:,2)'];
```

```
   writeData3 = {'Pearson Correlation of Proportion in CI and ddCt at Fixed
Sigmas:','Sigma';
      '','Correlation';'','P-Value'};
   writeData4 = [sig;corStatCB(:,1)';corStatCB(:,2)'];
   writeData5 = {'Ct Distribution Sigma';'Calculated ddCt Std Dev from Ct Distribution
Std Devs';'Actual ddCt Std Dev from Trial ddCts';'Proportion in CI';'Proportion in
Range';'Trial Means';
      'Trial Lower CIs';'Trial Upper CIs'};
   writeData6 =
[sig;csdTr(:,ind1)';sdTrC(:,ind1)';proCountC(:,ind1)';proCountR(:,ind1)';mnTrC(:,ind1)';
mnLoC(:,ind1)';mnUpC(:,ind1)'];
   writeName = strcat(labelDate,'_5_set_sim_calculated_',num2str(num),'_Cts','.xls');
   xlswrite(writeName,writeData1,ind1,'A1');
   xlswrite(writeName,writeData2,ind1,'C9');
   xlswrite(writeName,writeData3,ind1,'A12');
   xlswrite(writeName,writeData4,ind1,'C12');
   xlswrite(writeName,writeData5,ind1,'A16');
   xlswrite(writeName,writeData6,ind1,'B16');
end
```

```
fprintf('\n\nComplete . . .\n');
```

Code for function getCtStats.m

```
function [dataN,stdev] = getCtStats (data)

%This function accepts a 2D array of Ct data from a qPCR and returns

%normalized data (to the mean) and the standard deviations of each reaction type

[m,n] = size(data);

dataN = zeros(m,n);

for ind1 = 1:n %normalizes to the mean by column

   dataN(:,ind1) = data(:,ind1)-mean(data(:,ind1));

   stdev(ind1) = std(data(:,ind1));

end
```

Code for JCFqPCR_bs_sim_calculated_methods2.m

```
%JCFqPCR_bs_sim_calculated_methods2.m

%James Farslow (jfars@unm.edu)

%4 Aug 2015 (version 1 written 11 Jul 2015)
```

```
%Calculated method simulation
%This program will simulate the calculated method of combined data
%This simulation is analysing biological replicates with technical
%replicates.
%This simulation includes only ddCt values, not copy numbers which are
%derived from them.
%This simulation also includes varying ddCt values from -4 to +4
%Note: ddCt = -4 indicates an N-fold increase of 16x
%      ddCt = +4 indicates an N-fold decrease of 1/16 x
clc;
clear;
rng('shuffle');
tic;
%set parameters
cycDiff = [-4 0 4]; %cycle difference range for ddCt
lencycDiff = length(cycDiff);
reps = input('Number of trials: ');
```

```
num = input('Number of Cts: ');

alph = .05;  %set alpha (which is a reserved word), error level

Zdf = num-1; %bio rep degrees of freedom

ZTvalue = input(sprintf('Tcrit value from table method 2(df = %d,a = %4.3f):
',Zdf,alph/2));  %with 4 sets of Ct values, df = 4n-4

mn1 = 19;  %fix mu parameter mean for R/R'

mn2 = 19;  %fix mu parameter mean for T/R'

mn3 = 20;  %fix mu parameter mean for R/T'

bootreps = 10000; %number of bootstrap repetitions

sig = [.05:.1:.45]; %Ct distribution sigma parameter values

lenSig = length(sig);

biosig = 1; %additional sigma among bio reps

countC2 = zeros(lenSig,lencycDiff); %initialize counts for proportion mu capture

countZC1 = zeros(lenSig,lencycDiff);

proCountC2 = zeros(lenSig,lencycDiff); %proportion with mu in range

proCountZC1 = zeros(lenSig,lencycDiff);

mnCountC2 = zeros(lencycDiff);  %mean of proportion mu in range

trueDdct = zeros(lencycDiff,1);
```

```
biorepDdct = zeros(lenSig,lencycDiff,reps);
%set warning boxes
figure(100);
clf(100);
text(0,.5,sprintf('Simulation In Progress\n Please Do Not Touch'),'FontSize',40);
set(100,'Position',[250 400 800
200],'Name','Warning','NumberTitle','off','MenuBar','none');
set(gca,'Visible','off');
%figure(100)
figure(101);
clf(101);
text(0,.5,sprintf('Simulation In Progress\n Please Do Not Touch'),'FontSize',40);
set(101,'Position',[-1000 300 800 200],'Name','Extended
Warning','NumberTitle','off','MenuBar','none');
set(gca,'Visible','off');
%wait bar diff cycle
wtbr3 = waitbar(0,'cycDiff Loops Complete');
set(wtbr3, 'Position',[15 500 300 50],'Name','cycDiff');
```

```
%figure(101)

%set cycle diff loop

for indc = 1:lencycDiff

   %wait bar sigma

   wtbr1 = waitbar(0,'Sigma Loops Complete');

   set(wtbr1, 'Position',[332 500 300 50],'Name','Sigma');

   %set mu parameter for T/T' - changes

   mn4 = 20 + cycDiff(indc);

   %get trueddCt from means of distributions

   trueDdct(indc) = (mn4 - mn2) - (mn3 - mn1);

   %set sigma loop

   for inds = 1:lenSig

      %wait bar trials

      wtbr2 = waitbar(0,'Trials Loops Complete');

      set(wtbr2, 'Position',[650 500 300 50],'Name','Trials');

      %set trials loop

      for ind1 = 1:reps
```

```
clc;     %display to command window which sigma and trial iteration
fprintf('Cycle Difference: %d\n',cycDiff(indc));
fprintf('Sigma: %4.2f\n',sig(inds));
fprintf('Trial: %d\n',ind1);
fprintf('Simulation elapsed time: %d hr %d min %d sec',floor(toc/3600),floor((toc/60)-floor(toc/3600)*60),floor(toc-floor(toc/60)*60));
%timer box
figure(102);
clf(102);
text(-.1,.5,sprintf('Elapsed Time\n%d hr %d min %d sec\nN(Ct) = %d',floor(toc/3600),floor((toc/60)-floor(toc/3600)*60),floor(toc-floor(toc/60)*60),num),'FontSize',36);
set(102,'Position',[20 50 450 275],'Name','Timer','NumberTitle','off','MenuBar','none');
set(gca,'Visible','off');
%figure(102)
%get deviation shifts for bio reps
shift1 = normrnd(0,biosig,1,1);
```

```
shift2 = normrnd(0,biosig,1,1);
shift3 = normrnd(0,biosig,1,1);
%get sets of Ct values for each  biorep in this trial
data1(1:num,1) = normrnd(mn1,sig(inds),num,1);  %one set of Ct values
data1(1:num,2) = normrnd(mn2,sig(inds),num,1);  %one set of Ct values
data1(1:num,3) = normrnd(mn3,sig(inds),num,1);  %one set of Ct values
data1(1:num,4) = normrnd(mn4,sig(inds),num,1);  %one set of Ct values
data2(1:num,1) = normrnd(mn1,sig(inds),num,1);  %one set of Ct values
data2(1:num,2) = normrnd(mn2,sig(inds),num,1);  %one set of Ct values
data2(1:num,3) = normrnd(mn3,sig(inds),num,1);  %one set of Ct values
data2(1:num,4) = normrnd(mn4,sig(inds),num,1);  %one set of Ct values
data3(1:num,1) = normrnd(mn1,sig(inds),num,1);  %one set of Ct values
data3(1:num,2) = normrnd(mn2,sig(inds),num,1);  %one set of Ct values
data3(1:num,3) = normrnd(mn3,sig(inds),num,1);  %one set of Ct values
data3(1:num,4) = normrnd(mn4,sig(inds),num,1);  %one set of Ct values
%data is arranged as:
   %col 1 - R/R'
   %col 2 - T/R'

   %col 3 - R/T'

   %col 4 - T/T'

%Formula: ddCt = (T/T'-T/R')-(R/T'-R/R')

%                 (4 - 2) - (3 - 1)

%calculate mean ddCt and variance,

%tech rep 1 calculate means of each group

mnData11 = mean(data1(1:num,1));

mnData12 = mean(data1(1:num,2));

mnData13 = mean(data1(1:num,3));

mnData14 = mean(data1(1:num,4));

%Formula: ddCt = (T/T'-T/R')-(R/T'-R/R')

mnTech1 = shift1+(mnData14-mnData12)-(mnData13-mnData11);

biorepDdct(inds,indc,ind1) = mnTech1;

%tech rep 2 calculate means of each group

mnData21 = mean(data2(1:num,1));

mnData22 = mean(data2(1:num,2));
```

```
mnData23 = mean(data2(1:num,3));
mnData24 = mean(data2(1:num,4));
mnTech2 = shift2+(mnData24-mnData22)-(mnData23-mnData21);
%tech rep 3 calculate means of each group
mnData31 = mean(data3(1:num,1));
mnData32 = mean(data3(1:num,2));
mnData33 = mean(data3(1:num,3));
mnData34 = mean(data3(1:num,4));
mnTech3 = shift3+(mnData34-mnData32)-(mnData33-mnData31);
%added - calculate data set 1 as if it were bioreps
%Answer the question what does happen if you treat the data as
%if it were paired?
%pair the data and look at capture rate
%No figures, just print the results
Zdct1 = data1(:,4)-data1(:,2);
Zdct2 = data1(:,3)-data1(:,1);
Zddct = Zdct1 - Zdct2;
```

```
%get sd

Zsdddct = std(Zddct);

Zseddct = Zsdddct/sqrt(num); %se mean

ZCI = Zseddct*ZTvalue;

upper = mean(Zddct) + ZCI;

lower = mean(Zddct) - ZCI;

%test for mu capture

if (lower < trueDdct(indc) && upper > trueDdct(indc))

   countZC1(inds,indc) = countZC1(inds,indc)+1;

end

%calculate standard deviation for 3 bioreps

sd2Biorep = std([mnTech1 mnTech2 mnTech3]);

SE2Biorep = sd2Biorep/sqrt(3); %N=3, df = 2

CI2 = SE2Biorep * 4.303; %different 2.776

mnDdct2 = (mnTech1+mnTech2+mnTech3)/3;

upper2 = mnDdct2 + CI2; %actually lower N-fold

lower2 = mnDdct2 - CI2; %actually upper N-fold
```

```
        %test for mu capture, add to countC2
        if (lower2 < trueDdct(indc) && upper2 > trueDdct(indc))
          countC2(inds,indc) = countC2(inds,indc)+1;
        end
        %extend waitbar trials
        waitbar(ind1/reps,wtbr2);
    %end trials loop
    end;
     delete(wtbr2);
     %determine count proportion for each sigma per cyc diff
     proCountC2(inds,indc) = countC2(inds,indc)/reps;
    proCountZC1(inds,indc) = countZC1(inds,indc)/reps;
     %extend waitbar sigma
    waitbar(inds/lenSig,wtbr1);
%end sigma loop
end;
 delete(wtbr1);
```

```
    %extend waitbar cycDiff

   waitbar(indc/lencycDiff,wtbr3);

    %get means of counts over sigma

   mnCountC2(indc) = mean(proCountC2(:,indc));

%end cycle diff loop

end

%overall mean capture rate

mnData = mean(mean(proCountC2));

Zcapture = mean(mean(proCountZC1));

runtime = toc;

%fprintf simulation time, number of trials, number of Cts values, means of

%proportion mu capture

clc;

fprintf('Simulation elapsed time: %d hr %d min %d
sec\n',floor(runtime/3600),floor((runtime/60)-floor(runtime/3600)*60),floor(runtime-
floor(runtime/60)*60));

fprintf('Total Trials: %d\t\tNumber of Cts: %d\n',reps*lenSig*lencycDiff,num);

fprintf('Mean of Biorep mu capture percent: %5.2f\n',mnData*100);
fprintf('Mean of paired test mu capture percent: %5.2f\n',Zcapture*100);
commandwindow;
%strike the gong to signal finished
load gong.mat;
sound(y, Fs);
close(100);
close(101);
close(102);
delete(wtbr3);
%do an anova on biorep method - proCountC2
%rows - sigma, columns = cycDiff
%run 2 way anova
%figure 1 for table
[p,table,stats] = anova2(proCountC2,1);
fprintf('ANOVA: Rows = Sigma, Columns = ddCt\n');
set(1,'Position',[10 570 450 150]);
%figure 2 - 3D scatter plot of proportion of trials with mu capture, x = per
```

```
%sigma, y = per cycDiff, z = proportion capture, calculated bioreps

figure(2);clf(2);

bar3(proCountC2);

set(2,'Position',[125 100 1000 600],'Name','Calculated Biorep Method for Combined
Data, Capture Proportion');

axis([.5 lencycDiff+.5 .5 lenSig+.5 0 1.1]);

set(gca,'XTick',[1:lencycDiff],'XTickLabel',cycDiff);

set(gca,'YTick',[1:lenSig],'YTickLabel',sig);

title(sprintf('Calculated Biorep Method, Combined Data\nMu Capture Proportion vs.
Cycle Difference and Sigma \nNumber of Cts = %d',num),'FontSize',16);

xlabel('Cycle Difference','FontSize',14,'Rotation',25);

ylabel('Sigma','FontSize',14,'Rotation',-35);

zlabel('Capture Proportion','FontSize',14);

%

fprintf('\n\nComplete . . .\n');
```

References

Adler, M., Anjum, M., Berg, O., Andersson, D.I., and Sandegren, L. (2014). High fitness costs and instability of gene duplications reduce rates of evolution of new genes by duplication-divergence mechanisms. *Mol Biol Evol.* **31**, 1526-1535.

Agrawal, A.F., and Wang, A.D. (2008). Increased transmission of mutations by low-condition females: evidence for condition-dependent DNA repair. *PLoS Biol.* **6**, e30.

Almagro-Garcia, J., Manske, M., Carret, C., Campino, S., Auburn, S., MacInnis, B.L., Maslen, G., Pain, A., Newbold, C.I., Kwiatkowski, D.P., and Clark, T.G. (2009). SnoopCGH: software for visualizing comparative genomic hybridization data. *Bioinformatics.* **25**(20), 2732-2733.

Altschul, S., Gish, W., Miller, W., Myers, E., and Lipman, D. (1990). Basic local alignment search tool. *J Mol Biol.* **215**(3), 403-410.

Andersson, D.I., and Hughes, D. (2009). Gene amplification and adaptive evolution in bacteria. *Annu Rev Genet.* **43**, 167-195.

Anderson, R.P., and Roth, J.R. (1977). Tandem genetic duplications in phage and bacteria. *Annu Rev Microbiol.* **31**, 473–505.

Anderson, R.P., and Roth, J.R. (1981). Spontaneous tandem genetic duplications in *Salmonella typhimurium* by unequal recombination between rRNA (rrn) cistrons. *Proc Natl Acad Sci USA.* **78**, 3113–3117.

Beckmann, J.S., Estivill, X., and Antonarakis, S.E. (2007). Copy number variants and genetic traits: closer to the resolution of phenotypic and genotypic variability. *Nat Rev Genet.* **8**, 639-646.

Bender, A., and Pringle, J.R. (1989). Multicopy suppression of the cdc24 budding defect in
yeast by *CDC42* and three newly identified including the *ras*-related gene *RSR1*.
Proc Natl Acad Sci USA. **86**, 9976-9980.

Berg, C.M., Wang, M.D., Vartak, N.B., and Liu, L. (1988). Acquisition of new metabolic capabilities: multicopy suppression by cloned transaminase genes in *Escherichia coli* K-12. *Gene.* **65**, 195-202.

Bergthorsson, U., Andersson, D.I., and Roth, J.R. (2007). Ohno's dilemma: evolution of new genes under continuous selection. *Proc Natl Acad Sci USA.* **104**, 17004–17009.

Bergthorsson, U., and Ochman, H. (1999). Chromosomal changes during experimental evolution in laboratory population of *Escherichia coli. J Bacteriol.* **181**, 1360-1363.

Bu, L. (2015). *Comparative study of genomic features of evolutionarily young gene duplicates.* (Doctoral dissertation) Retrieved from LoboVault (Publication identifier http://hdl.handle.net/1928/30374).

Bull, J.J., Badgett, M.R., Wichman, H.A., Huelsenbeck, J.P., Hillis, D.M., Gulati, A., *et al.* (1997). Exceptional convergent evolution in a virus. *Genetics.* **147**, 1497-1507.

Bustin, S.A., Benes, V., Garson, J.A., Hellemans, J., Huggett, J., Kubista, M., Mueller, R., Nolan, T., Pfaffl, M.W., Shipley, G.L., Vandesompele, J., and Wittwer, C.T. (2009). The MIQE guidelines: minimum information for publication of quantitative real-time PCR experiments. *Clin Chem.* **55**, 611-622.

Cassata, G., Kagoshima, H., Andachi, Y., Kohara, Y., Dürrenberger, M.B., Hall, D.H., *et al.* (2000). The LIM homeobox gene *ceh-14* confers thermosensory function to the AFD neurons in *Caenorhabditis elegans*. *Neuron.* **25**, 587-597.

Chan, Y.F., Marks, M.E., Jones, F.C., Villareal Jr., G., Shapiro, M.D., Brady, S.D., *et al.* (2010). Adaptive evolution of pelvic reduction in sticklebacks by recurrent deletion of a *pitx1* enhancer. *Science.* **327**, 302-305.

Conrad, D.F., Pinto, D., Redon, R., Feuk, L., Gokcumen, O., Zhang, Y.J., *et al.* (2010). Origins and functional impact of copy number variation in the human genome. *Nature.* **464**, 704-712.

Cusack, B.P., and Wolfe, K.H. (2007). Not born equal: increased rate asymmetry in relocated and retrotransposed rodent gene duplicates. *Mol Biol Evol.* **24**, 679–686.

Cutter, A.D. (2006). Nucleotide polymorphism and linkage disequilibrium in wild populations of the partial selfer, *Caenorhabditis elegans*. *Genetics.* **172**, 171–184.

Cutter, A.D., and Charlesworth, B. (2006). Selection intensity on preferred codons correlates with overall codon usage bias in *Caenorhabditis remanei*. *Curr Biol.* **16**, 2053–2057.

Cutter, A.D., Day, A., and Murray, R.L. (2009). Evolution of the *Caenorhabditis elegans* genome. *Mol Biol Evol.* **26**, 1199–1234.

Degtyareva, N.P., Greenwell, P., Hofmann, E.R., Hengartner, M.O., Zhang, L., Culotti, J.G., *et al.* (2002). *Caenorhabditis elegans* DNA mismatch repair gene *msh-2* is required for microsatellite stability and maintenance of genomic integrity. *Proc Natl Acad Sci USA.* **99**, 2158-2163.

Denver, D.R., Dolan, P.C., Wilhelm, L.J., Sung, W., Lucas-Lledo, J.I., Howe, D.K., Lewis, S.C., Okamoto, K., Thomas, W.K., Lynch, M., *et al.* (2009). A genome-wide view of *Caenorhabditis elegans* base-substitution mutation processes. *Proc Natl Acad Sci USA.* **106**, 16310–16314.

Denver, D.R., Howe, D.K., Wilhelm, L.J., Palmer, C.A., Anderson, J.L., Stein, K.C., *et al.* (2010). Selective sweeps and parallel mutation in the adaptive recovery from deleterious mutation in *Caenorhabditis elegans*. *Genome Res.* **20**, 1663-1671.

Dobzhansky, T. (1970). *Genetics of the Evolutionary Process.* New York: Columbia University Press.

Edgar, R., Domrachev, M., and Lash, A.E. (2002). Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucl Acids Res.* **30**, 207-10.

Emerson, J.J., Cardoso-Moreira, M., Borevitz, J.O., and Long, M. (2008). Natural selection shapes genome-wide patterns of copy-number polymorphism in *Drosophila melanogaster*. *Science.* **320**, 1629–1631.

Falconer, D.S. (1989). *Introduction to quantitative genetics. 3rd ed.* New York, NY: John Wiley & Sons, Inc..

Ferreira, I.D., do Rosario, V.E., and Cravo, P.V.L. (2006). Real-time quantitative PCR with SYBR Green I detection for estimating copy numbers of nine drug resistance candidate genes in *Plasmodium falciparum*. *Malar J.* **5**, 1.

Fire, A., Xu, S.Q., Montgomery, M.K., Kostas, S.A., Driver, S.E., and Mello, C.C. (1998). Potent and specific genetic interference by double-stranded RNA in *Caenorhabditis elegans*. *Nature.* **391**, 806-811.

Fisher, R.A. (1930). *The Genetical Theory of Natural Selection.* Oxford, England: Clarendon Press.

Force, A., Lynch, M., Pickett, F.B., Amores, A., Yan, Y.L., and Postlethwait, J. (1999). Preservation of duplicate genes by complementary, degenerative mutations. *Genetics.* **151**, 1531–1545.

Futuyma, D.J. (1998). *Evolutionary Biology, 3rd ed.* Sunderland, MA: Sinauer Associates.

Gilsbach, R., Kouta, M., Bönisch, H., and Brüss, M. (2006). A comparison of in vitro and in vivo reference genes for internal standardization of quantitative real-time PCR data. *Biotechniques.* **40**, 173.

Gomez, M., De Castro, E., Guarin, E., Sasakura, H., Kuhara, A., Mori, I., *et al.* (2001). Ca2$^+$ signaling via the neuronal calcium sensor-1 regulates associated learning and memory in *C. elegans*. *Neuron.* **30**, 241-248.

Gonzalez, E., Kulkarni, H., Bolivar, H., Mangano, A., Sanchez, R., Catano, G., *et al.* (2005). The influence of CCL3L1 gene-containing segmental duplications on HIV-1/AIDS susceptibility. *Science.* **307**, 1434-1440.

Haldane, J.B.S. (1932). *The Causes of Evolution.* London: Longmans, Green & Co..

Han, M.V., Demuth, J.P., McGrath, C.L., Casola, C., and Hahn, M.W. (2009). Adaptive evolution of young gene duplicates in mammals. *Genome Res.* **19**, 859–867.

Headrick, T.C. (2010). *Statistical Simulation: Power Method Polynomials and Other Transformations.* Boca Raton, FL: Chapman & Hall/CRC.

Hellemans, J., Mortier, G., De Paepe, A., Speleman, F., and Vandesomple, J. (2007). qBase relative quantification framework and software for management and automated analysis of real-time quantitative PCR data. *Genome Biol.* **8**, R19.

Hughes, T.R., Roberts, C.J., Dai, H., Jones, A.R., Meyer, M.R., Slade, D., *et al.* (2000). Widespread aneuploidy revealed by DNA microarray expression profiling. *Nat Genet.* **25**, 333-337.

Iafrate, A.J., Feuk, L., Rivera, M.N., Listewnik, M.L., Donahoe, P.K., Qi, Y., Scherer, S.W., and Lee, C. (2004). Detection of large-scale variation in the human genome. *Nat Genet.* **36**, 949–951.

Innan, H., and Kondrashov, F. (2010). The evolution of gene duplications: classifying and distinguishing between models. *Nat Rev Genet.* **11**, 97-108.

Jin, N., He, K., and Liu, L. (2006). qPCR-DAMS: a database tool to analyze, manage, and store both relative and absolute quantitative real-time PCR data. *Physiol Genom.* **25**, 525-527.

Jones, M.R., Rose, A.M., and Baillie, D.L. (2012). Oligoarray comparative genomic hybridization-mediated mapping of suppressor mutations generated in a deletion-biased mutagenesis screen. *G3- Genes Genomes Genet.* **2**, 657-663.

Kamath, R.S., Martinez-Campos, M., Zipperlen, P., Fraser, A.G., and Ahringer, J. (2001). Effectiveness of specific RNA-mediated interference through ingested double-stranded RNA in *Caenorhabditis elegans*. *Genome Biol.* **2**, research0002.0001-10.

Katju, V. (2012). In with the old, in with the new: the promiscuity of the duplication process engenders diverse pathways for novel gene creation. *Int J Evol Biol.* ID 341932.

Katju, V., and Bergthorsson, U. (2013). Copy-number changes in evolution: rates, fitness effects and adaptive significance. *Front Genet.* **4**, 273.

Katju, V., Farslow, J.C., and Bergothorsson, U. (2009). Variation in gene duplicates with low synonymous divergence in *Saccharomyces cerevisiae* relative to *Caenorhabditis elegans*. *Genome Biol.* **10**, R75.

Katju, V., and Lynch, M. (2003). The structure and early evolution of recently arisen gene duplicates in the *Caenorhabditis elegans* genome. *Genetics.* **165**, 1793–1803.

Katju, V., and Lynch, M. (2006). On the formation of novel genes by duplication in the *Caenorhabditis elegans* genome. *Mol Biol Evol.* **23**, 1056–1067.

Katju, V., Packard, L.B., Bu, L., Keightley, P.D., and Bergthorsson, U. (2015). Fitness decline in spontaneous mutation accumulation lines of *Caenorhabditis elegans* with varying effective population sizes. *Evolution.* **69**, 104-116.

Kimura, M., and Ohta, T. (1969). The average number of generations until fixation of a mutant gene in a finite population. *Genetics.* **61**, 763-771.

Kitchen, R.R., Kubista, M., and Tichopad, A. (2010). Statistical aspects of quantitative real-time PCR experiment design. *Methods.* **50**, 231-236.

Kondrashov, F.A. (2012). Gene duplication as a mechanism of genomic adaptation to a changing environment. *Proc R Soc B.* **279**, 5048-5057.

Koskiniemi, S., Sun, S., Berg, O.G., and Andersson, D.I. (2012). Selection-driven gene loss in bacteria. *PLoS Genet.* **8**, e1002787.

Lam, K.W.G., and Jeffreys, A.J. (2007). Processes of *de novo* duplication of human α-globin genes. *Proc Natl Acad Sci USA.* **104**, 10950–10955.

Lee, M-C., and Marx, C.J. (2012). Repeated, selection-driven reduction of accessory genes in experimental populations. *PLoS Genet.* **8**, e1002651.

Lewis, J.A., and Fleming, J.T. (1995). Basic cultural methods. In: Epstein, H.P., and Shakes, D.C., editors. *Methods in cell biology: Caenorhabditis elegans: Modern Biological Analysis of an Organism.* London: Academic Press, pp. 4-29.

Li, W., Kennedy, S.G., and Ruvkun, G. (2003). *daf-28* encodes a *C. elegans* insulin superfamily member that is regulated by environmental cues and acts in the DAF-2 signaling pathway. *Genes Dev.* **17**, 844-858.

Lipinski, K.J., Farslow, J.C., Fitzpatrick, K.A., Lynch, M., Katju, V., and Bergthorsson, U. (2011). High spontaneous rate of gene duplication in *Caenorhabditis elegans*. *Curr Biol.* **21**, 306-310.

Liu, P., Carvalho, C.M.B., Hastings, P.J. and Lupski, J.R. (2012). Mechanisms for recurrent and complex human genomic rearrangements. *Curr Opin Genet and Dev.* 22, 211-220.

Livak, K.J., and Schmittgen, T.D. (2001). Analysis of relative gene expression data using real-time quantitative PCR and the $2^{-\Delta\Delta Ct}$ method. *Methods.* **25**, 402-408.

Long, M., Betran, E., Thornton, K., and Wang, W. (2003). The origin of new genes: glimpses from the young and old. *Nat Rev Genet.* **4**, 865-875.

Lynch, M., and Conery, J.S. (2000). The evolutionary fate and consequences of duplicate genes. *Science.* **290,** 1151–1154.

Lynch, M., and Conery, J.S. (2003). The evolutionary demography of duplicate genes. *J Struct Funct Genomics.* **3**, 35–44.

Lynch, M., and Conery, J.S. (2003). The origins of genome complexity. *Science.* **302**, 1401–1404.

Lynch, M., and Katju, V. (2004). The altered evolutionary trajectories of gene duplicates. *Trends Genet.* **20**, 544–549.

Lynch, M., O'Hely, M., Walsh, B., and Force, A. (2001). The probability of fixation of a newly arisen gene duplicate. *Genetics.* **159**, 1789–1804.

Lynch, M., Sung, W., Morris, K., Coffey, N., Landry, C.R., Dopman, E.B., Dickinson, W.J., Okamoto, K., Kulkarni, S., Hartl, D.L., *et al*. (2008). A genome-wide view of the spectrum of spontaneous mutations in yeast. *Proc Natl Acad Sci USA.* **105**, 9272–9277.

Maroni, G., Wise, J., Young, J.E., and Otto, E. (1987). Metallothionein gene duplications and metal tolerance in natural populations of *Drosophila melanogaster*. *Genetics.* **117**, 739-744.

Maruyama, I.N., Miller, D.M., and Brenner, S. (1989). Myosin heavy chain gene amplification as a suppressor mutation in *Caenorhabditis elegans*. *Mol Gen Genet.* **219**, 113-118.

Maydan, J.S., Lorch, A., Edgley, M.L., Flibotte, S., and Moerman, D.G. (2010). Copy number variation in the genomes of twelve natural isolates of *Caenorhabditis elegans*. *BMC Genomics.* **11**, 62.

Maydan, J.S., Flibotte, S., Edgley, M.L., Lau, J., Selzer, R.R., Richmond, T.A., Pofahl, N.J., Thomas, J.H., and Moerman, D.G. (2007). Efficient high-resolution deletion discovery in *Caenorhabditis elegans* by array comparative genome hybridization. *Genome Res.* **17**, 337–347.

Maynard Smith, J., and Haigh, J. (1974). The hitch-hiking effect of a favourable gene. *Genet Res.* **23**, 23-35.

Mayr, E. (1963). *Animal Species and Evolution.* Cambridge, MA: The Belknap Press of Harvard University Press.

Menez, J., Remy, E., and Buckingham, R.H. (2001). Suppression of thermosensitive peptidyl-tRNA hydrolase mutation in *Escherichia coli* by gene duplication. *Microbiol.* **147**, 1581-1589.

Miller, B.G., and Raines, R.T. (2004). Identifying latent enzyme activities: substrate ambiguity within modern bacterial sugar kinases. *Biochem.* **43**, 6387-6392.

Mills, R.E., Walter, K., Stewart, C., Handsaker, R.E., Chen, K., Alkan, C., *et al.* (2011). Mapping copy number variation by population-scale genome sequencing. *Nature.* **470**, 59-65.

Moore, J.K., and Haber, J.E. (1996). Cell cycle and genetic requirements of two pathways of nonhomologous end-joining repair of double-strand breaks in *Saccharomyces cerevisiae. Mol Cell Biol.* **16**(5), 2164-2173.

Muller, P.Y., Janovjak, H., Miserez, A.R., and Dobbie, Z. (2002). Processing of gene expression data generated by quantitative real-time RT-PCR. *Biotechniques.* **32**, 1372-1378.

Nair, S., Miller, B., Barends, M., Jaidee, A., Patel, J., Mayxay, M., *et al.* (2008). Adaptive copy number evolution in malaria parasites. *PLoS Genet.* **4**, e1000243.

Nair, S., Nash, D., Sudimack, D., Jaidee, A., Barends, M., Uhlemann, A.C., Krishna, S., Nosten, F., and Anderson, T.J.C. (2007). Recurrent gene amplification and soft selective sweeps during evolution of multidrug resistance in malaria parasites. *Mol Biol Evol.* **24**, 562–573.

Newcomb, R.D., Gleeson, D.M., Yong, C.G., Russell, R.J., and Oakeshott, J.G. (2005). Multiple mutations and gene duplications conferring organophosphorus insecticide resistance have been selected at the *Rop-1* locus of the sheep blowfly, *Lucilia cuprina*. *J Mol Evol.* **60**, 207-220.

Ohno, S. (1970). *Evolution by Gene Duplication.* Heidelberg, Germany: Springer-Verlag.

Ohta, T. (1988). Time for acquiring a new gene by duplication. *Proc Natl Acad Sci USA.* **85**, 3509–3512.

Ott, R.L. (1993). *An Introduction to Statistical Methods and Data Analysis, 4th ed..*, Belmont, CA: Wadsworth Publishing Company.

Otto, S.P., and Yong. P. (2002). The evolution of gene duplicates. In: Dunlap, J., and Wu, C-T., editors. *Advances in Genetics, Vol. 46*. San Diego, CA: Academic Press, pp. 451–483.

Pan, D., and Zhang, L. (2007). Quantifying the major mechanisms of recent gene duplications in the human and mouse genomes: a novel strategy to estimate gene duplication rates. *Genome Biol.* **8**, R158.

Patrick, W.M., Quandt, E.M., Swartzlander, D.B., and Matsumara, I. (2007). Multicopy suppression underpins metabolic evolvability. *Mol Biol Evol.* **24**, 2716-2722.

Peirson, S.N., Butler, J.N., and Foster, R.G. (2003). Experimental validation of novel and conventional approaches to quantitative real-time PCR data analysis. *Nucleic Acids Res.* **31**, e73.

Perry, G.H., Dominy, N.J., Claw, K.G., Lee, A.S., Fiegler, H., Redon, R., Werner, J., Villanea, F.A., Mountain, J.L., Misra, R., *et al*. (2007). Diet and the evolution of human amylase gene copy number variation. *Nat Genet.* **39**, 1256–1260.

Pettersson, M.E., Sun, S., Andersson, D.I., and Berg, O.G. (2009). Evolution of new gene functions: simulation and analysis of the amplification model. *Genetica.* **135**, 309–324.

Pfaffl, M.W. (2001). A new mathematical model for relative quantification in real-time RT-PCR. *Nucleic Acids Res.* **29**, e45.

Pfaffl, M.W., Horgan, G.W., and Dempfle, L. (2002). Relative expression software tool (REST©) for group-wise comparison and statistical analysis of relative expression results in real-time PCR. *Nucleic Acids Res.* **30**(9), e36.

Pfaffl, M.W., Vandesompele, J., and Kubista, M. (2009). Data Analysis Software. In: Logan, J., Edwards, K., and Saunders, N., editors. *Real_Time PCR: Current Technology and Applications.* Poole, UK: Caister Academic Press, pp. 65-83.

Reams, A.B., and Neidle, E.L. (2003). Genome plasticity in *Acinetobacter*: new degradative capabilities acquired by the spontaneous amplification of large chromosomal segments. *Mol Microbiol.* **47**, 1291-1304.

Riddle, D.L., and Brenner, S. (1978). Indirect suppression in *Caenorhabditis elegans*. *Genetics.* **89**, 299-314.

Riehle, M.M., Bennett, A.F., and Long, A.D. (2001). Genetic architecture of thermal adaptation in *Escherichia coli*. *Proc Natl Acad Sci USA.* **98**, 525-530.

Rokyta, D.R., Joyce, P., Caudle, S.B., and Wichman, H.A. (2005). An empirical test of the mutational landscape model of adaptation using a single-stranded DNA virus. *Nat Genet.* **37**, 441–444.

Rozen, S. and Skaletsky, H. J. (1998) Primer3. Code available at http://www-genome.wi.mit.edu/genome_software/other/primer3.html.

Samuels, M., Witmer, J., and Schaffner, A. (2012). *Statistics for the Life Sciences, 4th ed.*. Boston, MA: Pearson Education, Inc..

Sarin, S., Bertrand, V., Bigelow, H., Boyanov, A., Doitsidou, M., Poole, R.J., Narula, S., and Hobert, O. (2010). Analysis of multiple ethyl methanesulfonate-mutagenized *Caenorhabditis elegans* strains by whole-genome sequencing. *Genetics.* **185**, 417–430.

Schedl, T., and Kimble, J. (1988). *fog-2*, a germ-line specific sex determination gene required for hermaphrodite spermatogenesis in *Caenorhabditis elegans*. *Genetics.* **119**, 43-61.

Schmittgen, T.D., and Livak, K.J. (2008). Analyzing real-time PCR data by the comparative C_T method. *Nat Protoc.* **3**, 1101-1108.

Schrider, D.R., Houle, D., Lynch, M., and Hahn, M.W. (2013). Rates and genomic consequences of spontaneous mutational events in *Drosophila melanogaster*. *Genetics.* **194**, 937–954.

Sebat, J., Lakshmi, B., Troge, J., Alexander, J., Young, J., Lundin, P., Maner, S., Massa, H., Walker, M., Chi, M.Y., *et al.* (2004). Large-scale copy polymorphism in the human genome. *Science.* **305**, 525–528.

Serebrijski, I., Wojcik, F., Reyes, O., and Leblon, G. (1995). Multicopy suppression by *asd* gene and osmotic stress-dependent complementation by heterologous *proA* in *proA* mutants. *J Bacteriol.* **177**, 7255-7260.

Shapira, S.K., and Finnerty, V.G. (1986). The use of genetic complementation in the study of eukaryotic macromolecular evolution: rate of spontaneous gene duplication at two loci of *Drosophila melanogaster*. *J Mol Evol.* **23**, 159–167.

Sonti, R.V., and Roth, J.R. (1989). Role of gene duplications in the adaptation of *Salmonella typhimurium* to growth on limiting carbon sources. *Genetics.* **123**, 19-28.

Sulston, J., and Hodgkin, J. (1988). Methods. In: W.B. Wood, ed., *The Nematode Caenorhabditis elegans*, Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press, pp. 587-606.

Tijsterman, M., Pothof, J., and Plasterk, R.H. (2002). Frequent germline mutations and somatic repeat instability in DNA mismatch-repair-deficient *Caenorhabditis elegans*. *Genetics.* **161**, 651-660.

Timms, A.R., and Bridges, B.A. (1998). Reversion of the tyrosine ochre strain *Escherichia coli* WU3610 under starvation conditions depends on a new gene *tas*. *Genetics.* **148**, 1627-1635.

Tlsty, T.D., Albertini, A.M., and Miller, J.H. (1984). Gene amplification in the *lac* region of *E. coli*. *Cell.* **37**, 217-24.

Trempy, J.E., and Gottesman, S. (1989). Alp, a suppressor of lon protease mutants in *Escherichia coli*. *J Bacteriol.* **171**, 3348-3353.

Tsai, I.J., Bensasson, D., Burt, A., and Koufopanou, V. (2008). Population genomics of the wild yeast *Saccharomyces paradoxus*: quantifying the life cycle. *Proc Natl Acad Sci USA.* **105**, 4957–4962.

Turner, D.J., Miretti, M., Rajan, D., Fiegler, H., Carter, N.P., Blayney, M., Beck, S., and Hurles, M.E. (2008). Germline rates of de novo meiotic deletions and duplications causing genomic disorders. *Nat Genet.* **40**, 90–95.

Ueguchi, C., and Ito, K. (1992). Multicopy suppression: an approach to understanding intracellular functioning of the protein export system. *J Bacteriol.* **174**, 1454-1461.

Van Ommen, G.J.B. (2005). Frequency of new copy number variation in humans. *Nat Genet.* **37**, 333–334.

Vassilieva, L.L., and Lynch, M. (1999). The rate of spontaneous mutation for life-history traits in *Caenorhabditis elegans*. *Genetics.* **151**, 119–129.

Vassilieva, L.L., Hook, A.M., and Lynch, M. (2000). The fitness effects of spontaneous mutations in *Caenorhabditis elegans*. *Evolution.* **54**, 1234-1246.

Veitia, R.A. (2004). Gene dosage balance in cellular pathways: implications for dominance and gene duplicability. *Genetics.* **168**, 569-74.

Watanabe, Y., Takahashi, A., Itoh, M., and Takano-Shimizu, T. (2009). Molecular spectrum of spontaneous de novo mutations in male and female germline cells of *Drosophila melanogaster*. *Genetics.* **181**, 1035–1043.

Williams, B.D., Schrank, B., Huynh, C., Shownkeen, R., Waterston, R.H. (1992). A geneticmapping system in *Caenorhabditis elegans* based on polymorphic sequence-tagged sites. *Genetics.* **131**, 609-624.

Wood, T.E., Burke, J.M., and Rieseberg, L.H. (2005). Parallel genotypic adaptation: when evolution repeats itself. *Genetica.* **123**, 157-170.

Yamanaka, K., Ogura, T., Koonin, E.V., Niki, H., and Hiraga, S. (1994). Multicopy suppressors, *mssA* and *mssB*, of an *smbA* mutation of *Escherichia coli*. *Mol Gen Genet.* **243**, 9-16.

Yampolsky, L.Y., and Stoltzfus, A. (2001). Bias in the introduction of variation as an orienting factor in evolution. *Evol Dev.* **3**, 73–83.

Yuan, J.S., Reed, A., Chen, F., and Stewart, C.N. Jr. (2006). Statistical analysis of real-time PCR data. *BMC Bioinformatics.* **7**, 85.

Zhang, J. (2003). Evolution by gene duplication: an update. *Trends in Ecol Evol.* **18**, 292–298.