STUDIES IN THEORETICAL EVOLUTIONARY GENETICS

by

REED A. CARTWRIGHT

(Under the direction of Wyatt W. Anderson and Paul Schliekleman)

ABSTRACT

Although many programs exist for generating simulated alignments of DNA sequences, no program combines the robust GTR substitution model with a model of indel formation. To correct this gap, the application, Dawg, was created. Dawg is the first program to provide the option of producing indels under the power-law model, which is the model most consistent with biological data. Some authors have suggested that the power-law distribution of indel lengths make logarithmic gap costs the preferred method of sequence alignment. Utilizing Dawg, the utility of logarithmic gap costs is investigated in pairwise global alignments. They are shown to be poor performers when compared to the standard affine methods. Furthermore, a model for calculating gap costs is developed which explains why affine costs are a better option than logarithmic costs.

The third study investigates the antagonism between local dispersal and self-incompatibility systems in a continuous plant population. This antagonism is shown to affect the fine-scaled genetic structure of the population and depend on the linkage of markers and self-incompatibility loci. Furthermore, gametophytic and sporophytic self-incompatibility are show to not be much different than obligate outcrossing.

INDEX WORDS: Evolution, Simulation, Alignment, Logarithm, Self-Incompatibility, Local Dispersal, Plant Mating System, Dissertations, Theses (academic)

STUDIES IN THEORETICAL EVOLUTIONARY GENETICS

by

REED A. CARTWRIGHT

B.S., The University of Georgia, 2000

A.B., The University of Georgia, 2000

A Dissertation Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

DOCTOR OF PHILOSOPHY

ATHENS, GEORGIA

2006

© 2006

Reed A. Cartwright

All Rights Reserved

STUDIES IN THEORETICAL EVOLUTIONARY GENETICS

by

REED A. CARTWRIGHT

Approved:

Major Professors:	Wyatt W. Anderson
	Paul Schliekleman
Committee:	James Hamrick
	Jessica Kissinger
	H. Ronald Pulliam
	John Wares

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
August 2006

DEDICATION

To Tiffany

and

For Marjorie

ACKNOWLEDGMENTS

This dissertation was supported by an NSF Predoctoral Fellowship and NIH Grant 5R01 GM48528-06. I would like to thank my advisers—Wyatt Anderson, Marjorie Asmussen, and Paul Schliekelman—my committee—John Avise, Jim Hamrick, Jessica Kissinger, Ron Pulliam, and John Wares—my fellow lab members—Beth Dakin, Nicole Leahy, Renyi Liu, Kyungsun Kim, and Yong-Kyu Kim—my fellow grad students—Gina Baucom, Tina Bell, Scott Cornman, Judith Mank, Sam Odell, Monica Poelchau, Jeff Ross-Ibarra, and Scott Small—other scientists—Shu-Mei Chang, Oliver Hardy, John Nason, Hamish Spenser, Douglas Theobald, Jeff Thorne, Marcy Uyenoyama, and Xavier Vekemans—my parents, and most importantly my wife, Tiffany.

Contents

List of Figures

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Theoretical research is very central to the practice of science. Modern science depends on theoretical research, whether mathematical, statistical, or computational, to develop methodologies and predictions that can be employed in experimental research. Theoretical research has been very important to the study of genetics since the early part of the twentieth century. The theoretical research of Sewall Wright, R.A. Fisher, and J.B.S. Haldane served as the foundation of the modern synthesis of evolution and genetics. Today, out of all the life sciences, evolutionary biology is probably the strongest in generating and utilizing theoretical research.

In the three studies presented in this dissertation, I extend the theoretical evolutionary genetics. In the first study I develop a portable application that can simulate sequence evolution using the robust GTR+Γ+I model of sequence substitution and the power-law model of indel lengths. In the second study, I apply this simulation to the question of logarithmic gaps costs for global pair-wise alignments. I show that, despite some suggestions in the literature, logarithmic gap costs are a horrible way to align sequences. Furthermore, I develop a model for gap costs that can explain why logarithmic gap costs do not derive from indel sizes that obey a power-law. In the third study I change course from studying molecular evolution and look at a topic in population genetics: the antagonism between self-incompatibility systems and local dispersal. I show that gametophytic and sprophytic self-incompatibility have approximately the same effect on inbreeding and gene dispersal.

CHAPTER 2

LITERATURE REVIEW

2.1 SEQUENCE SIMULATION

Many tools and procedures exist that can reconstruct sequence alignments, estimate phylogenetic relationships, or estimate evolutionary parameters from extant data. These tools and procedures are required because true phylogenies and alignments are known in only very rare, experimental instances (e.g. Bull et al., 1993; Hillis et al., 1992, 1994). "Obvious" phylogenies and alignments, although helpful, cannot provide the same level of precision as simulated phylogenies. Because we rely on tools to infer alignments, phylogenies, and evolutionary parameters, the accuracy of these tools is an issue that biologists take seriously. In the absence of known data with true phylogenies, we are left using simulations to test the accuracy of bioinformatic procedures (e.g. Blanchette et al., 2004; Hillis et al., 1994; Kuhner and Felsenstein, 1994). Simulations can produce flawless sequence alignments derived from known (i.e. user specified) phylogenies and evolutionary parameters. Using such simulated sequences, researchers can determine the accuracy of procedures for estimating the alignment, phylogeny, or evolutionary parameters. Because many of these procedures assume certain models of evolution, evolving simulated sequences using such models provides a measure of accuracy in ideal cases. Using models of evolution that deviate from ideal cases can, in turn, demonstrate the robustness of the procedures. Parametric bootstrapping using simulations can also be used to produce confidence intervals for procedures.

Sequences evolve through changing existing residues, adding new residues, or removing existing residues. Therefore, proper simulation of molecular evolution of DNA should involve nucleotide substitution, insertion, and deletion. However, existing tools for simulating sequence evolution (Table 3.1) either do not include indels, like Seq-gen (Rambaut and Grassly, 1997) or evolver

(Yang, 1997), include an underived model of indel formation, like Rose (Stoye et al., 1998), or are designed for a particular purpose and are inflexible like (Hall, 2005). In my first study, I develop a robust simulation to fill in these gaps.

2.2 Logarithmic Alignments

Sequence alignments are essential to the study of molecular biology and systematics. Alignments are necessary because during evolution sequences can gain and lose residues. Alignments reveal such gaps in sequence data. Researchers usually need to align sequences before they can study them. For example, most algorithms that construct phylogenetic trees from sequences require a sequence alignment (Swofford, 2002). Since alignments are intended to tell researchers something about the evolutionary history of sequences, the quality of the inferences we make from alignments depends on the quality of the alignments themselves.

Studies on the distribution of indel lengths have revealed that they obey a power-law (Benner et al., 1993; Chang and Benner, 2004; Gu and Li, 1995; Zhang and Gerstein, 2003). This observation suggests that sequences should be aligned using logarithmic gap costs, i.e. $C_g(x) = a + c\ln(x)$ (Gu and Li, 1995). However, the standard method of sequence alignment uses affine gap costs, i.e. $C_g(x) = a + bx$ (Gotoh, 1982). Affine gap costs are popular because they are efficient (Gotoh, 1982) and model the fact that gaps "cost" more to start then they do to extend. However, researchers cannot adapt Gotoh's algorithm to logarithmic gap costs. Instead researchers must use the more computationally expensive candidate list method of Waterman (1984) as optimized by Miller and Myers (1988). Although, affine gap costs are efficient, my second study seeks to determine whether this efficiency comes with a cost to accuracy.

2.3 Self-Incompatibility and Isolation-by-Distance

Many taxa of angiosperms employ Mendelian mating-types to prevent self-fertilization. These "self-incompatibility" systems come in two forms: gametophytic and sporophyic. In gametophytic self-incompatibility (GSI), the mating-type of the pollen is determined by the pollen haplotype,

and stigmas reject any pollen bearing either one of their alleles. Two gametophytic systems have been studied molecularly: Papaveraceae (e.g. poppy) and Solanaceae (e.g. nightshade, potato, tobacco) (Igic and Kohn, 2001; Nasrallah, 2005; Takayama and Isogai, 2005). In sporophytic self-incompatibility (SSI), the mating-type of the pollen is determined by the pollen-donor's genotype, and stigmas will only accept pollen produced by plants that share none of the stigmas's alleles. One sporophytic system has been studied molecularly: Brassicaceae (e.g. cabbage, broccoli, *Arabidopsis*) (Igic and Kohn, 2001; Nasrallah, 2005; Takayama and Isogai, 2005). These SI systems are controlled by a single highly polymorphic Mendelian locus, the S Locus, which contains several tightly linked genes. This tight linkage results from both close physical association on the chromosome and reduced recombination rate in the area (Kamau and Charlesworth, 2005). Because recombination rates are reduced and S loci experience strong balancing selection, genes near the S loci will exhibit more diversity and longer coalescent times than they would otherwise (Awadalla and Charlesworth, 1999; Charlesworth, 2006; Charlesworth et al., 2006; Hagenblad et al., 2006; Kamau and Charlesworth, 2005; Schierup et al., 2000).

The function of the S Locus is to prevent pollen from fertilizing flowers of genetically similar plants. The primary result is that self-fertilization is impossible, and the secondary result is that matings between relatives are reduced. The S Locus evolves under classical frequency-dependent sexual selection (Wright, 1939), resulting in high fitness for rare and novel alleles. Plants with rare alleles can pollinate more plants than can plants with common alleles.

Although a species may inhabit a geographically contiguous region with no physical barriers to gene flow, the species may not be panmictic. The physical distances separating two individuals may prevent them from mating freely, and instead individuals are more likely to mate with individuals that are near them than individuals that are far away from them. This is known as "isolation-by-distance" (Wright, 1943). Wright (1946) studied isolation-by-distance further and developed the concept of "neighborhood size" to compare how different mating systems affect isolation-by-distance. In a continuous population, isolation-by-distance creates fine scaled genetic structure because individuals near one another are expected to be more closely related than individuals

farther apart. Therefore, alleles can often show a patchy distribution with respect to geography (Epperson, 1990; Rohlf and Schnell, 1971; Sokal and Wartenberg, 1983; Turner et al., 1982).

Self-incompatibility systems promote outbreeding while local dispersal promotes inbreeding. Therefore, an antagonism exists between the two processes. Furthermore, this antagonism will manifest differently in makers linked to the S locus than in markers that are unlinked to the S locus. Several studies have looked at the effect of population subdivision on allelic diversity of S loci (Muirhead, 2001; Neuhauser, 1998; Schierup, 1998; Schierup et al., 2000; Wright, 1939). Furthermore, two studies have looked at the effect of isolation-by-distance on self-incompatibility loci (Brooks et al., 1996; Neuhauser, 1998). However, none of those studies have looked at the antagonism directly, which is the goal of my third study.

2.4 REFERENCES

Awadalla, P. and D. Charlesworth (1999). Recombination and selection at brassica self-incompatibility loci. *Genetics 152*, 413–425.

Benner, S. A., M. A. Cohen, and G. H. Gonnet (1993). Empirical and structural models for insertions and deletions in the divergent evolution of proteins. *J. Mol. Biol. 229*, 1065–1082.

Blanchette, M., E. D. Green, W. Miller, and D. Haussler (2004). Reconstructing large regions of an ancestral mammalian genome in silico. *Genome Res. 14*, 2412–2423.

Brooks, R. J., A. M. Tobias, and M. J. Lawrence (1996). The population genetics of the self-incompatibility polymorphism in papaver rhoeas. xi. the effects of limited pollen and seed dispersal, overlapping generations and variation in plant size on the variance of s-allele frequencies in populations at equilibrium. *Heredity 76*, 367–376.

Bull, J. J., C. W. Cunningham, I. J. Molineux, M. R. Badgett, and D. M. Hillis (1993). Experimental molecular evolution of bacteriophage-t7. *Evolution 47*, 993–1007.

Chang, M. S. S. and S. A. Benner (2004). Empirical analysis of protein insertions and deletions determining parameters for the correct placement of gaps in protein sequence alignments. *J. Mol. Biol. 341*, 617–631.

Charlesworth, D. (2006). Balancing selection and its effects on sequences in nearby genome regions. *PLoS Genet 2*(4), 379–384.

Charlesworth, D., E. Kamau, J. Hagenblad, and C. Tang (2006). Trans-specificity at loci near the self-incompatibility loci in arabidopsis. *Genetics 172*, 2699–2704.

Epperson, B. K. (1990). Spatial autocorrelation of genotypes under directional selection. *Genetics 124*, 757–771.

Gotoh, O. (1982). An improved algorithm for matching biological sequences. *J. Mol. Biol. 162*, 705–708.

Gu, X. and W. H. Li (1995). The size distribution of insertions and deletions in human and rodent pseudogenes suggests the logarithmic gap penalty for sequence alignment. *J. Mol. Evol. 40*, 464–473.

Hagenblad, J., J. Bechsgaard, and D. Charlesworth (2006). Linkage disequilibrium between incompatibility locus region genes in the plant arabidopsis lyrata. *Genetics*, genetics.106.055780. Epub ahead of print.

Hall, B. G. (2005). Comparison of the accuracies of several phylogenetic methods using protein and dna sequences. *Mol. Biol. Evol. 22*, 792–802.

Hillis, D. M., J. J. Bull, M. E. White, M. R. Badgett, and I. J. Molineux (1992). Experimental phylogenetics - generation of a known phylogeny. *Science 255*, 589–592.

Hillis, D. M., J. P. Huelsenbeck, and C. W. Cunningham (1994). Application and accuracy of molecular phylogenies. *Science 264*, 671–677.

Igic, B. and J. R. Kohn (2001). Evolutionary relationships among self-incompatibility rnases. *Proc. Natl. Acad. Sci. U. S. A. 98*, 13167–13171.

Kamau, E. and D. Charlesworth (2005). Balancing selection and low recombination affect diversity near the self-incompatibility loci of the plant arabidopsis lyrata. *Curr Biol 15*, 1773–1778.

Kuhner, M. K. and J. Felsenstein (1994). Simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Mol. Biol. Evol. 11*, 459–468.

Miller, W. and E. W. Myers (1988). Sequence comparison with concave weighting functions. *Bull. Math. Biol. 50*, 97–120.

Muirhead, C. (2001). Consequences of population structure on genes under balancing selection. *Evolution 55*, 1532–1541.

Nasrallah, J. B. (2005). Recognition and rejection of self in plant self-incompatibility: comparisons to animal histocompatibility. *Trends in Immunology 26*(8), 412–418.

Neuhauser, C. (1998). The ancestral graph and gene genealogy under frequency-dependent selection. *Theoretical Population Biology 56*, 203–214.

Rambaut, A. and N. C. Grassly (1997). Seq-gen: An application for the monte carlo simulation of dna sequence evolution along phylogenetic frees. *Comput. Appl. Biosci. 13*, 235–238.

Rohlf, F. J. and G. D. Schnell (1971). An investigation of the isolation-by-distance model. *The American Naturalist 105*, 295–324.

Schierup, M. H. (1998). The number of self-incompatibility alleles in a finite, subdivided population. *Genetics 149*, 1153–1162.

Schierup, M. H., D. Charlesworth, and X. Vekemans (2000). The effect of hitch-hicking on genes linked to a balanced polymorphism in a subdivided population. *Genet. Res. Camb. 76*, 63–73.

Schierup, M. H., X. Vekemans, and D. Charlesworth (2000). The effect of subdivision on variation at multi-allelic loci under balancing selection. *Genet. Res. Camb. 76*, 51–62.

Sokal, R. R. and D. E. Wartenberg (1983). A test of spatial autocorrelation analysis using an isolation-by-distance model. *Genetics 105*, 219–237.

Stoye, J., D. Evers, and F. Meyer (1998). Rose: generating sequence families. *Bioinformatics 14*, 157–163.

Swofford, D. L. (2002). *PAUP*: Phylogenetic Analysis Using Parsimony (and Other Methods) 4.0 Beta*. Sinauer Associates, Inc, Sunderland MA.

Takayama, S. and A. Isogai (2005). Self-incompatibility in plants. *Annu. Rev. Plant Biol. 56*, 467–89.

Turner, M. E., J. C. Stephens, and W. W. Anderson (1982). Homozygosity and patch structure in plant populations as a result of nearest-neighbor pollination. *Proc. Natl. Acad. Sci. USA 79*, 203–207.

Waterman, M. S. (1984). Efficient sequence alignment algorithms. *J. Theor. Biol. 108*, 333–337.

Wright, S. (1939). The distribution of self-sterility alleles in populations. *Genetics 24*, 538–552.

Wright, S. (1943). Isolation by distance. *Genetics 28*, 114–138.

Wright, S. (1946). Isolation by distance under diverse systems of mating. *Genetics 31*, 39–59.

Yang, Z. H. (1997). Paml: a program package for phylogenetic analysis by maximum likelihood. *Comput. Appl. Biosci. 13*, 555–556.

Zhang, Z. and M. Gerstein (2003). Patterns of nucleotide substitution, insertion and deletion in the human genome inferred from pseudogenes. *Nucleic Acids Res. 31*, 5338–5348.

CHAPTER 3

DNA ASSEMBLY WITH GAPS (DAWG): SIMULATING SEQUENCE EVOLUTION[1]

[1]Cartwright, R.A. (2005) *Bioinformatics*. 21 Suppl 3:iii31-iii38.
Reprinted here with permission of publisher.

3.1 INTRODUCTION

Many tools and procedures exist that can reconstruct sequence alignments, estimate phylogenetic relationships, or estimate evolutionary parameters from extant data. These tools and procedures are required because true phylogenies and alignments are known in only very rare, experimental instances (e.g. Bull et al., 1993; Hillis et al., 1992, 1994). "Obvious" phylogenies and alignments, although helpful, cannot provide the same level of precision as simulated phylogenies. Because we rely on tools to infer alignments, phylogenies, and evolutionary parameters, the accuracy of these tools is an issue that biologists take seriously. In the absence of known data with true phylogenies, we are left using simulations to test the accuracy of bioinformatic procedures (e.g. Blanchette et al., 2004; Hillis et al., 1994; Kuhner and Felsenstein, 1994). Simulations can produce flawless sequence alignments derived from known (i.e. user specified) phylogenies and evolutionary parameters. Using such simulated sequences, researchers can determine the accuracy of procedures for estimating the alignment, phylogeny, or evolutionary parameters. Because many of these procedures assume certain models of evolution, evolving simulated sequences using such models provides a measure of accuracy in ideal cases. Using models of evolution that deviate from ideal cases can, in turn, demonstrate the robustness of the procedures. Parametric bootstrapping using simulations can also be used to produce confidence intervals for procedures.

Sequences evolve through changing existing residues, adding new residues, or removing existing residues. Therefore, proper simulation of molecular evolution of DNA should involve nucleotide substitution, insertion, and deletion. However, existing tools for simulating sequence evolution (Table 3.1) either do not include indels, like Seq-gen (Rambaut and Grassly, 1997) or evolver (Yang, 1997), include an underived model of indel formation, like Rose (Stoye et al., 1998), or are designed for a particular purpose and are inflexible like (Hall, 2005). I developed Dawg to fill these gaps.

Dawg is the first sequence simulation program to combine the popular general time reversible model, gamma and invariant rate heterogeneity, and a model of indel formation. Dawg has several different ways to model the distributions of the sizes of insertions and deletions. Most importantly,

Table 3.1: Comparison of simulation programs
Seq-Gen (Rambaut and Grassly, 1997); Evolver (Yang, 1997); Rose (Stoye et al., 1998); EvolveAGene (Hall, 2005)

	Seq-Gen 1.32	Evolver 3.14a	Rose 1.3	EvolveAGene 2.3	Dawg 1.0.0
GTR	Yes	Yes	No	No	Yes
Rate Heterogeneity	Γ+I	Γ	Γ+I	No	Γ+I
Recombination	Yes	No	No	No	Yes
Indels	No	No	Yes	Yes	Yes
Indel Parameter Estimation	N/A	N/A	No	No	Yes
Input Format	Switch	File	File	Menu	File
Unix	Yes	Yes	Yes	No	Yes
Mac OS X	Yes	Yes	Yes	Yes	Yes
Win32	Yes	Yes	No	No	Yes

Dawg is the first program that can explicitly model indel sizes via a power-law distribution, which has been found in both nucleotide and protein sequences (Benner et al., 1993; Chang and Benner, 2004; Gu and Li, 1995; Zhang and Gerstein, 2003). Additionally, Dawg can simulate recombination by using different phylogenies for different sections of the sequence. Dawg restricts recombination and indel formation to blocks of nucleotides of a constant width. Deletions remove whole blocks, while insertions and recombination occur between blocks. In most uses of the program, the blocks are one nucleotide wide. Although the underlying model of evolution is neutral DNA, the evolution of coding sequences can be approximated by setting blocks to be three nucleotides wide and specifying different rates of evolution for the positions in the block.

The utility of any simulation lies in its ability to generate biologically realistic data. To that end, researchers can use standard phylogenetic packages to estimate most of the parameters of the model. A helper script written in Perl, lambda.pl, is provided with Dawg to allow researchers to estimate parameters of indel evolution from biological data, facilitating biologically meaningful simulations of indel formation.

3.2 Systems and Methods

Dawg is a command line program written in standard C++, GNU Bison, and GNU Flex for portability. It is packaged using the GNU autoconf and GNU automake tools and will compile on systems which support them, including most popular derivatives of Unix like Linux, FreeBSD, and Macintosh OS X. It can be compiled in Windows using the minimalist GNU for Windows (MinGW) or using Microsoft Visual Studio .Net 2003 with ports of GNU Bison and Flex. The document INSTALL explains how to compile and install the package for most systems. Development took place on a variety of machines, including a Windows XP workstation, a FreeBSD server, an Irix server, a Linux cluster, and a Macintosh OS X desktop. A Perl script, lambda.pl, is distributed with Dawg and can be used to estimate parameters for the indel model from an alignment and a phylogeny with branch lengths. A few other utility scripts are also included.

Dawg is configured by an input file, an example of which is in Figure 3.1. One of the design goals of Dawg is to offer a robust DNA simulation package, and thus there are a wide range of options. New options and features may be added in future versions of the program. Currently options include controlling the phylogeny, substitution model, indel model, and program output.

3.3 Algorithm

Dawg's algorithm for simulating evolution supports substitution, rate heterogeneity, indel formation, and recombination. There is no limit on the length of sequences or size and structure of phylogenies, except those imposed by hardware and time. The complexity of the algorithm is $O(NLR)$, where N is the number of nodes in the phylogeny, L is the average sequence length, and R is the number of repetitions. Similarly, the memory requirement is $O(NL)$. Simulation of Figure 3.1 with ten thousand repetitions took 10.5 seconds on an Intel Xeon 3.06GHz, Windows XP workstation and consumed less than a megabyte of memory.

```
# example.dawg
Tree = ((AY727331:0.001359,AY727330:0.001359):0.084512,
(AY727327:0.006116,AY727326:0.006116):0.079756);
Model = "GTR"
Params = {1.08031, 2.45581, 0.44452,
          1.09145, 4.06519, 1.00000}
Freqs = {0.353470, 0.143681, 0.178206, 0.324643}
Length = 300
Lambda = 0.143120
GapModel = "NB"
GapParams = {1, 0.753247}
Format = "Clustal"
File = "example.aln"
Seed = 1981
```

Figure 3.1: Example.dawg, a sample input file derived from biological data. The parameters were derived from the sequences of chloroplast trnK introns of four Rosid species. The initial sequence length was shortened and a seed was added for reproducibility. The output, example.aln, can be found in Figure 3.3. See Example Usage (subsection 3.3.11) and Results (section 3.5) for more detail.

3.3.1 SUBSTITUTION

Dawg produces descendent sequences from ancestral sequences via a two-step evolutionary model. The first step simulates substitution via the general time reversible model with gamma and invariant rate heterogeneity (GTR+Γ+I; Felsenstein, 2004; Lanave et al., 1984; Rodríguez et al., 1990; Tavaré, 1986; Waddell and Steel, 1997; Yang, 1994, 1993). The second step simulates indel formation via a continuous-time, length-dependent model derived for Dawg.

GTR is a ten parameter model (eight free, two dependent) which represents nucleotide substitution in continuous time. The parameters are the four stationary nucleotide frequencies, φ_i, and the six symmetric, relative, instantaneous rates of substitution, σ_{ij}. These parameters combine to form the instantaneous substitution rate matrix

$$Q = \begin{bmatrix} q_{AA} & \sigma_{AC}\varphi_C & \sigma_{AG}\varphi_G & \sigma_{AT}\varphi_T \\ \sigma_{AC}\varphi_A & q_{CC} & \sigma_{CG}\varphi_G & \sigma_{CT}\varphi_T \\ \sigma_{AG}\varphi_A & \sigma_{CG}\varphi_C & q_{GG} & \sigma_{GT}\varphi_T \\ \sigma_{AT}\varphi_A & \sigma_{CT}\varphi_C & \sigma_{GT}\varphi_G & q_{TT} \end{bmatrix}$$

where $q_{ii} = -\sum_{j \neq i} \sigma_{ij}\varphi_j$. Many common models (Felsenstein, 1981, 1984; Hasegawa et al., 1985; Jukes and Cantor, 1969; Kimura, 1980, 1981; Tamura and Nei, 1993) can be expressed as specializations of GTR. See Felsenstein (2004) for a recent and detailed description of the GTR model.

3.3.2 Heterogeneous Rates

The Γ+I model of heterogeneous rates of nucleotide evolution allows for the rate of evolution to vary among sites, with a set proportion of sites remaining unchanged (Waddell and Steel, 1997). Under this model, the relative rates of substitution at each position are independent and identically distributed by the hierarchal distribution,

$$f(r|\alpha,\iota) = \begin{cases} 0 & : \quad r < 0 \\ \iota & : \quad r = 0 \\ (1-\iota)\,\frac{(\alpha r)^{\alpha} e^{-\alpha r}}{r\Gamma(\alpha)} & : \quad r > 0 \end{cases}$$

where $0 \leq \iota \leq 1$ is the proportion of invariant sites, $\alpha > 0$ is the shape parameter, and $\Gamma(\alpha)$ is the complete gamma function. The expected value of this distribution is $1-\iota$, and the variance is $(1-\iota)(\gamma+\iota)$, where $\gamma = \alpha^{-1}$ is the coefficient of variance. The coefficient of variance is preferred over the shape parameter for describing the Γ+I distribution. If $\gamma = 0$, the distribution becomes discrete, and a site either evolves in step with the branch length ($r = 1$) or remains unchanged ($r = 0$). The simulation holds constant each site's relative rate of substitution, and daughter nucleotides inherit their parent's rate. Dawg extends the basic Γ+I model by allowing each position in a block to have different γ and ι parameters. Each position also has a relative scaling parameter, s, to allow some positions to evolve relatively faster than others.

3.3.3 RESCALING

Dawg calculates the probability that nucleotide j substitutes for nucleotide i at site n with relative rate r_n and relative scale s_n over time t as the (i, j) entry of matrix

$$P_n(t) = e^{kQs_n r_n t}$$

where k is a correction factor. The expected number of substitutions for block position w is

$$E(Y|t,w) = \sum_{i=\{A,C,G,T\}} -k\varphi_{i,w} q_{ii,w} s_w (1-\iota_w) t$$

where k is the rescaling constant, $\varphi_{i,w}$ is the frequency of nucleotide i for block position w, $q_{ii,w}$ is the (i,i) entry of the GTR matrix for position w, s_w is the relative scalar for position w in a block, and ι_w is the proportion of invariant sites for position w. Therefore, the expected number of substitutions per site given time t is

$$E(Y|t) = \frac{1}{W} \sum_{w=1}^{W} E(Y|t,w)$$

where W is the block width. As Felsenstein (1981) and Yang (1994) suggest, Dawg rescales the substitution matrix such that the branch lengths represent the expected number of substitutions per site. Thus, since $E(Y|t) = t$,

$$k = \frac{W}{\sum_{w=1}^{W} \sum_{i=\{A,C,G,T\}} -\varphi_{i,w} q_{ii,w} s_w (1-\iota_w)}$$

Since the GTR parameters are the same for all block positions, the correction factor simplifies to

$$k = -W \left(\sum_{w=1}^{W} s_w (1-\iota_w) \right)^{-1} \left(\sum_{i=\{A,C,G,T\}} \varphi_i q_{ii} \right)^{-1}$$

This rescaling to substitution time is important for interpretation of the indel model.

Because the calculation of $P_n(t)$ involves finding the eigenvalues and eigenvectors of kQ, Dawg implements a Jacobi transformation (Press et al., 1992, pages 463-469) optimized for a four-by-four matrix. Although Jacobi transformations are simple, accurate, and numerically stable, they only work on symmetric matrices, and kQ is not symmetric. Dawg utilizes a mathematical trick to find the eigensystem of a symmetric matrix related to kQ and to convert the results to the eigensystem of kQ (Yang, 1995). The matrix $S = \Phi^{1/2} kQ \Phi^{-1/2}$ is symmetric and has the same eigenvalues as kQ, where $\Phi^{1/2} = \text{diag}\left(\varphi_A^{1/2}, \varphi_C^{1/2}, \varphi_G^{1/2}, \varphi_T^{1/2}\right)$. If V_S are right eigenvectors of S, then $V_{kQ} = \Phi^{-1/2} V_S$ are the right eigenvectors of kQ. Once the eigensystem is found, $P_n(t)$ can be calculated and used with the state of the ancestral nucleotide to randomly draw the donor nucleotide for the position.

3.3.4 Indel Formation

Dawg implements a novel model of indel formation. Like substitutions, indels occur in continuous time. The model treats insertions and deletions as different processes, and each one has its own distribution of sizes and instantaneous rate of formation. The model assumes that there is a fixed, instantaneous rate of indels occurring at any site at any time; therefore, indels are more probable in longer sequences and over longer time intervals. To satisfy this assumption, Dawg uses a Poisson process that is linearly dependent on the length of the sequence. Table 3.2 shows some differences between Dawg's model and other indel models.

In this model indel formation is restricted to a certain block width, e.g. 1 for nucleotides and 3 for codons. Indel formation occurs in substitution time, and when the block width is 1, the instantaneous rates of formation approximately represent the ratio of insertions or deletions to substitutions. An indel is identified by two parameters: a location and a length. The location represents the place where nucleotides are inserted or the place at which a deletion begins. The length represents the number of blocks inserted or deleted.

Insertions are rather simple to model. If ℓ is the number of blocks in the subsequence being evolved, then there are $\ell + 1$ possible locations for an insertion to occur, including both ends of

Table 3.2: Comparisons of indel formation models. TKF91 (Thorne et al., 1991); TKF92 (Thorne et al., 1992); Rose (Stoye et al., 1998); McAlign (Keightley and Johnson, 2004); Long Indel (Miklós et al., 2004)

	TKF91	TKF92	Rose	McAlign	Long Indel	Dawg
Poisson Process	Yes	Yes	No	No	Yes	Yes
Length Dependent	Yes	Yes	Yes	Yes	Yes	Yes
Time Dependent	Yes	Yes	Yes	Yes	Yes	Yes
In Substitution Time	No	No	No	Yes	No	Yes
Multiresidue Indels	No	Yes	Yes	Yes	Yes	Yes
Overlapping Ends	N/A	Yes	No	No	Yes	Yes
Time Reversibility Required	Yes	Yes	No	Yes	Yes	No
Immortal Link	Yes	Yes	No	No	Yes	Yes
Insertion-Deletion Differences	Yes	Yes	Yes	No	Yes	Yes
Gaps can overlap	N/A	No	Yes	No	Yes	Yes
Alignment Algorithm	Yes	Yes	No	Yes	Yes	No
Simulation Algorithm	No	No	Yes	No	No	Yes

the sequence. The sequence thus has an "immortal link" (Thorne et al., 1991), ensuring that an insertion can occur if $\ell = 0$. If λ_I is the rate of insertion per location, then the waiting time until an insertion occurs is exponentially distributed with mean $(\lambda_I \ell + \lambda_I)^{-1}$. Inserted nucleotides are randomly drawn from the stationary base frequencies and heterogeneous rate distribution. Insertions are right oriented, which means that, if an insertion occurs at a recombination point, the insertion becomes associated with the rightmost section.

Deletions are more difficult to model because the ends of the subsequence have to be taken into account. A deletion that starts in a region preceding a sequence may still delete part of the sequence. To account for this, I first assume that the subsequence being modeled exists inside a larger sequence of size N blocks, such that $N \gg \ell$. I also assume that the maximum size of a deletion is M blocks, such that $N \gg M$. This allows the ends of the larger sequence to be ignored and the ends of the smaller sequence to be considered. A deletion of size u that occurs in the larger sequence will delete part of the subsequence if it begins at one of the ℓ nucleotides of the subsequence or at one of the $u-1$ nucleotides preceding the subsequence. Therefore, if deletions occur uniformly along

the larger sequence, then the probability that a deletion in the larger sequence of size u removes some part of the smaller sequence is simply $(u-1+\ell)/N$. If $f_D(u)$ is the discrete distribution of the size of deletions, then the total probability that a deletion in the larger sequence removes some part of the smaller sequence is

$$\sum_{u=1}^{M} \frac{u-1+\ell}{N} f_D(u) = \frac{1}{N}\left[\sum_{u=1}^{M} u f_D(u) + (-1+\ell)\sum_{u=1}^{M} f_D(u)\right] = \frac{\bar{u}_D - 1 + \ell}{N} \tag{3.1}$$

If λ_D is the rate of deletion per location, then the total rate of deletion in N is $\lambda_D N$. From this and Equation 3.1, the waiting time until a deletion occurs in the subsequence is exponentially distributed with mean $[\lambda_D(\bar{u}_D - 1) + \lambda_D \ell]^{-1}$. Because N cancels out, we can consider both it and M to be infinite, allowing more flexibility in the choice of $f_D(u)$.

3.3.5 Indel-Size Distributions

In Dawg the length of an indel is represented by the number of blocks that it covers. Dawg has three different ways to model the distribution of indel lengths. The first method allows users to specify the exact discrete distribution of indel lengths. This is referred to as the user model. The second method models indel lengths using a negative binomial distribution. This model takes two parameters, an integer (r) and a proportion (q), and has the probability mass function,

$$f(l) = \binom{r+l-2}{l-1}(1-q)^r q^{l-1}$$

where $l = 1, 2, \ldots$ is the length of an indel. The mean of this distribution is $1 + rq/(1-q)$. If $r = 1$, the distribution is geometric.

The third and most important method models indel lengths via a power-law or Zipf distribution. Indel lengths have been found to approximately obey this distribution (Benner et al., 1993; Chang and Benner, 2004; Gu and Li, 1995; Zhang and Gerstein, 2003), and some theory supports it (Benner et al., 1993). In a Zipf distribution, the probability that an indel has length $l = 1, 2, \ldots$ is

$$f(l) = \frac{l^{-a}}{\zeta(a)}$$

where $a > 1$ is the parameter of the distribution and $\zeta(a)$ is the Riemann Zeta function:

$$\zeta(a) = \sum_{k=1}^{\infty} k^{-a}$$

If $a > 2$, the mean of a Zipf distribution is $\zeta(a-1)/\zeta(a)$, and if $a > 3$, the variance is $\zeta(a-2)/\zeta(a) - (\zeta(a-1)/\zeta(a))^2$. Otherwise, the mean and variance are infinite. Because the tail of a Zipf distribution is often fat, Dawg truncates the distribution to a user specified, maximum indel-size, M.

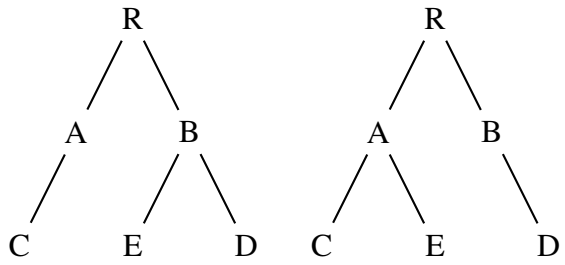
3.3.6 RECOMBINATION

Dawg can produce simulated sequences from phylogenies that contain recombinations. This feature is optional and is enabled when a user specifies a phylogeny of multiple trees in the input file. Other software, e.g. ms (Hudson, 2002), can be used to simulate such tree sets from demographic parameters. Dawg combines trees into a recombinant phylogeny by splitting the sequences at each node in the phylogeny into multiple sections. Each section corresponds to a separate tree and has its own ancestral node. If two or more sections have the same ancestral node, then their distance to that ancestor is specified by the last tree in the group.

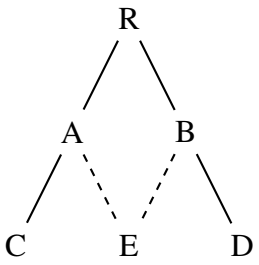
Recombination occurs when different sections in the same node are descended from different ancestors. A recombinant sequence is assembled from donor sequences, and each donor sequence is associated with an ancestral node. If node A is ancestral to section N of the descendant sequence, then section N of the descendant sequence is copied from section N of donor sequence A. Donor sequences are produced by evolving ancestral sequences over the distance separating the ancestor from the descendant. Figure 3.2 describes this process.

```
((C:0.2)A:0.1,(D:0.1,E:0.1)B:0.2)R;
((C:0.2,E:0.2)A:0.1,(D:0.1)B:0.2)R;
```

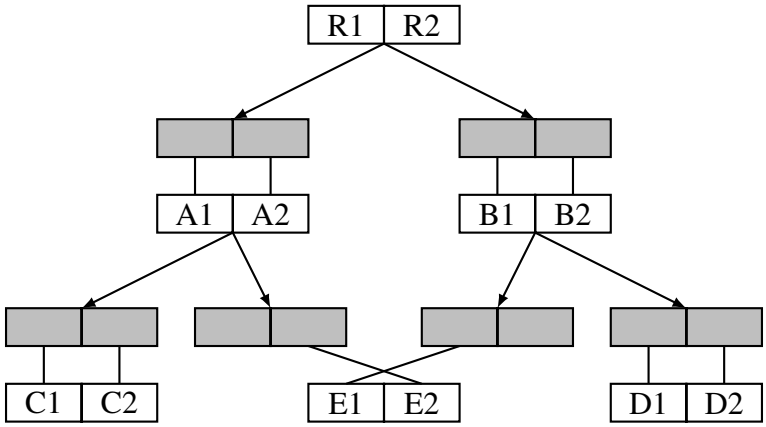
(a) A pair of Newick trees specifying a recombinant phylogeny



(b) A visualization of these trees



(c) A visualization of the recombinant phylogeny, where solid branches are found in both trees and dashed branches are found in only one tree



(d) A graph of how Dawg simulates evolution on the recombinant tree. Boxes, pairs of boxes, and gray boxes represent sections, sequences, and donor respectively. Arrows indicate where evolution occurs, and lines indicate where donor sections are copied to descendant sections

Figure 3.2: Recombination

3.3.7 SIMULATION

Dawg simulates this model of indel formation using a Gillespie algorithm (Gillespie, 1977). For the algorithm, the total event rate is

$$\lambda_T(\ell) = (\lambda_I + \lambda_D)\ell + \lambda_D \bar{u}_D + \lambda_I - \lambda_D$$

and the probability that an event is an insertion is $p(\ell) = (\ell+1)\lambda_I/\lambda_T(\ell)$. Under the algorithm the waiting time until an event occurs is drawn from an exponential distribution with mean $\lambda_T(\ell)^{-1}$. When an indel is formed, it is randomly drawn as an insertion or deletion, with probabilities $p(\ell)$ and $1-p(\ell)$ respectively. The length of the indel is drawn from its indel-size distribution, and its position is drawn uniformly from the pool of all possibilities, given the indel size. The algorithm cycles, updating the length of the sequence each round, until the sum of the waiting times is greater than the branch length.

Using these models of evolution, Dawg can construct the sequence for every node in the phylogeny from the root node. The sequence of the root node can be either specified by the user or randomly constructed from the stationary distribution of nucleotide frequencies and the heterogeneous rate model.

3.3.8 ALIGNMENT

Dawg maintains the indel history of each nucleotide for reconstruction of the true alignment of sequences. The indel history has one of four states: root, insertion, deletion, and deleted insertion. When a descendant sequence is constructed from its ancestors, the indel states of parent nucleotides are copied to daughter nucleotides. In this way sequences at the tips of the phylogeny will contain information about their lineage from the root.

When nucleotides are inserted in the sequence, their indel state is marked as being insertions. When nucleotides are deleted in a sequence, they are not actually removed, but rather are marked

as being deletions. (Dawg skips deletions when evolving sequences.) If an inserted nucleotide is subsequently deleted, it is marked to distinguish it from a deleted root nucleotide.

The alignment algorithm uses the history of sequences to construct their alignment, adding gaps to sequences opposite of insertions and deleted insertions. In the alignment, insertions are once again right-oriented, which means that if an insertion occurs at the same location as a previous deletion, then it will be to the right of that deletion in the alignment.

3.3.9 TRANSLATION

Dawg can also translate aligned nucleotide sequences into amino acid sequences, using a newly developed and extremely efficient algorithm. In the 7-bit portable ASCII code set each letter in the alphabet is represented by a specific number, e.g. “A” is 65. In the table-lookup method (Altschul et al., 1990; Pearson and Lipman, 1988), a 128-entry array is used to convert 7-bit numbers representing nucleotides into 2-bit numbers. Three 2-bit numbers are combined to form a single 6-bit index for the 64-entry array, which serves as a lookup table for the genetic code (amino acids and stop codons). Here I present an improved translation method for unambiguous nucleotides that relies on a novel algorithm and only requires a 64-entry array.

This algorithm relies on a fortuitous property of the standard ASCII numeric representations of nucleotide letters Table 3.3. If we specify four logical groups of nucleotide letters (A/a, C/c, T/t/U/u, and G/g), then the second and third least significant bits are the same in each group and different between groups Table 3.3. Therefore, we are able to use only two binary operations, a mask and a shift, to reduce each nucleotide group to 0, 1, 2, and 3, respectively Table 3.3. This new method is called “mask-shift.”

Table 3.3: Binary Manipulation of Nucleotide Encoding Characters. [a]Masked by 6 (0000 0110) [b]Shifted right by 1

	Decimal	Hexadecimal	Binary	Masked[a]	Shifted[b]	Result
A	65	0x41	0100 0001	0000 0000	0000 0000	0
a	97	0x61	0110 0001			
C	67	0x43	0100 0011	0000 0010	0000 0001	1
c	99	0x63	0110 0011			
T	84	0x54	0101 0100	0000 0100	0000 0010	2
t	116	0x74	0111 0100			
U	85	0x55	0101 0101			
u	117	0x75	0111 0101			
G	71	0x47	0100 0111	0000 0110	0000 0011	3
g	103	0x67	0110 0111			

Table 3.4: Results of Speed Tests for Mask-Shift Algorithm. [a]Arithmetic implementation

System		Compiler		Approximate Speed in Seconds		
Processor	OS	Version	Flags	Mask-Shift	Table-Lookup	Time Saved
AMD Athlon XP 2100+	Windows XP	GCC 3.4.3	-O3 -march=athlon-xp	23.95	33.80	30%
Apple PowerPC G4 867MHZ	Mac OS X	GCC 3.3	-fast -mcpu=7450	19.94	26.53	25%
Apple PowerPC G5 2.0GHZ	Mac OS X	GCC 3.3	-fast	4.75	7.39	36%
IBM Power4+ 1200MHZ	AIX 5.2	XLC 6.0	-O5 -qarch=pwr4 -qlanglvl=stdc99	18.16[a]	25.07[a]	28%
Intel Itanium 2 1300MHZ	Linux 2.4.21	ICC 8.1	-fast -mcpu=itanium2	11.86	22.32	47%
Intel Xeon III 866MHZ	Freebsd 5.3	ICC 8.1	-O3	51.81[a]	58.75	11%
Intel Xeon 3.06GHZ	Windows XP	ICL 8.1	-O3 -QxN -Qipo -Qc99	6.97[a]	8.06[a]	14%
MIPS R12000 300MHZ	Irix 6.5	GCC 3.2	-O3 -mips4	93.51	122.36	24%

Since an index for a 64-entry array can be constructed using masks, shifts, and inclusive ors, these two processes can be efficiently merged, producing the following remarkably compact C algorithm:

```
const char csStd[] = "KNNKTTTTIIIMRSSRQHHQPPPPLLLLRRRR"\
                     "*YY*SSSSLFFL*CCWEDDEAAAAVVVVGGGG";
void Translate(char *csOut, const char *csIn, const char *csCode) {
    for(;*csIn; csIn += 3)
        *csOut++ = csCode[((csIn[0]&6) << 3)
                          |((csIn[1]&6) << 1)
                          |((csIn[2]&6) >> 1)];
    *csOut = '\0';
}
```

This function takes three parameters: the character string for the resulting amino acid translation, the character string containing the nucleotides to be translated, and a character string containing the genetic code. If the first two parameters are identical then the string is translated in-place. A string for the standard genetic code is included in the algorithm above, but nonstandard genetic codes are easy to implement.

This algorithm has five advantages: 1) it is inherently insensitive to case; 2) it naturally treats "T" and "U" equally; 3) it can easily translate using alternate genetic codes; 4) it avoids costly logical branching statements; and 5) it avoids a 128-entry lookup table which costs both memory and speed.

Speed tests show that our algorithm can save as much as 47% of the time taken by the simplified table-lookup translation algorithm (Table 3.4). For some architectures and/or compilers, replacing the binary left-shifts and inclusive ors with equivalent arithmetic operations may result in faster binaries (e.g. "`* 8`" for "`<< 3`," "`* 2`" for "`<< 1`," and "`+`" for "`|`" in the C code above).

3.3.10 Parameter Estimation

I supply a Perl script, lambda.pl, with Dawg. This script contains a simple algorithm to estimate parameters for Dawg's indel model from a nucleotide sequence alignment and a rooted phylogeny with branch-lengths that represent the expected number of substitutions per site. The script does

not estimate parameters for the richest model of indel formation possible with Dawg. Instead it treats insertion and deletion as equal processes and estimates the total rate of indel formation, $\lambda_{ID} = \lambda_I + \lambda_D = 2\lambda_I = 2\lambda_D$. It can be shown that, assuming a block width of 1, the total number of unique indels in a phylogeny, N, is approximately distributed by a Poisson distribution with mean $\lambda_{ID}\bar{L}T$, where $\bar{L}$ is the average length of the sequences and T is the total branch-length of the rooted phylogeny. If $\bar{L}$ and T are known, a maximum likelihood estimate of λ_{ID} is $\hat{\lambda}_{ID} = N/T\bar{L}$. The script calculates T and $\bar{L}$ from the supplied alignment and rooted phylogeny. It then estimates N as the number of unique gaps in the alignment. Although $\hat{\lambda}_{ID}$ is a maximum likelihood estimator for the approximate distribution, its statistical significance to the actual distribution is unknown. I have simulated Dawg and lambda.pl together and verified that $\hat{\lambda}_{ID}$ is consistent with λ_{ID} (results not shown). The largest magnitude of deviation, -3%, occurred when λ_{ID} equaled 1 indel per substitution, which for nonrepetitive DNA is an extremely high and biologically unrealistic value.

The Perl script goes beyond estimating the instantaneous rate of indel formation per site. It also calculates the distribution of indel sizes (user-input model) and fits this distribution to three other models: negative binomial, geometric (NB with $r = 1$), and power-law. The parameters for the geometric and negative-binomial models are estimated via maximum likelihood. The power law model is fitted via linear regression of the first five indel size frequencies on a log-log scale (Jones and Handcock, 2003). This has been shown to be a very good estimator of power law distributions (Goldstein et al., 2004). These models can be distinguished via maximum likelihood for probability, minimum Akaike information (Akaike, 1974) and Bayesian information (Schwarz, 1978) for parsimony, and χ^2 for goodness-of-fit.

It is worth noting that the most popular way to align sequence pairs is with affine gap penalties (Gotoh, 1982). Alternatively, the algorithm of Miller and Myers (1988) can align sequences globally using logarithmic gap penalties.

3.3.11 Example Usage

As an example of how to use Dawg and lambda.pl, I estimated the rate of indel formation of a set of sequences and then parametrically bootstrapped the estimate via Dawg. I used sequences from the intron of the chloroplast trnK gene of four plant species: *Hibiscus mechowii*, *H. cannabinus*, *Prunus nigra*, and *P. virginiana* (Genbank accession numbers AY727326, AY727327, AY727330, and AY727331 respectively; Shaw et al., 2005). The genera are both Rosids, but *Prunus* is a eurosid I and *Hibiscus* is a eurosid II. Insertions and deletions are known to be prevalent in chloroplast sequences (Clegg et al., 1994), and the trnK intron almost certainly evolves neutrally and without recombination.

I first aligned these sequences using ClustalW 1.81 (Thompson et al., 1994) and corrected the alignment where necessary. Next, I used Paup* 4.0 (Swofford, 2002) to estimate the phylogeny and substitution parameters of the sequences for a GTR and molecular clock model. I then used lambda.pl to estimate the indel parameters from the phylogeny and aligned sequences. From the estimates, I constructed a parameter file for Dawg and simulated a thousand sequences evolving under the conditions estimated from the actual data. For each of these simulated sequences sets, I estimated phylogenetic trees and the rate of indel formation using Paup* and lambda.pl.

3.4 Implementation

Dawg is run on the command line. It is controlled through input files and a few command-line switches, which control the processing of the input files. Input files can be processed together or in succession. The structure of an input file is a series of statements in the form of "variable = value." There are several types of values: strings, booleans, numbers, trees, and vectors of values.

The default output is to stdout in Fasta format. Output can also be to a file; Phylip, Nexus, and Clustal formats are also supported. Dawg can return multiple sequence sets, and a Perl script, outsplit.pl, is provided to retrieve single alignments from outputs.

3.5 RESULTS

The rounded, average length of the biological sequences was 741. Their estimated phylogeny was ((AY727331: 0.001359, AY727330: 0.001359): 0.084512, (AY727327: 0.006116, AY727326: 0.006116): 0.079756). The estimated stationary frequency of adenine was 0.353470, cytosine 0.143681, guanine 0.178206, and thymine 0.324643. The symmetric instantaneous rate of substitution for adenine and cytosine was estimated to be 1.08031, adenine and guanine 2.45581, adenine and thymine 0.44452, cytosine and guanine 1.09145, cytosine and thymine 4.06519, and guanine and thymine 1.0. The indel-size distribution was estimated to be geometric with a q of 0.753247.

The estimated rate of indel formation was 0.143120, and bootstrapping via Dawg gave a 95% confidence interval of 0.078530 to 0.213560. In biological terms, this is 8 to 21 indels per 100 substitutions. The phylogenies produced from the simulated data during bootstrapping were consistent with the biological data, having in every case the same topology as the biological phylogeny. Furthermore, the biological phylogeny had a total tree length of 0.179218, and the simulated phylogenies had an average total tree length of 0.180075 and standard deviation of 0.017993.

Figure 3.1 shows an input file, example.dawg, for Dawg, which was derived from the biological data mentioned above. The sequence length was shortened from 741 to 300, and a random number seed was added to make the results suitable for publication. Figure 3.3 shows the resulting output file, example.aln, of Dawg.

3.6 DISCUSSION

Although a geometric model was found to best fit the gap sizes in the trnK intron alignment, this may be due to the small sample size of indels. However, the gap size distribution has little impact on bootstrapping the rate of indel formation.

Table 3.1 compares Dawg to three other published sequence simulation programs. However, it is important to go into some detail about the differences between Dawg and two previ-

```
CLUSTAL multiple sequence alignment (Created by DAWG Version 1.0.0)

AY727326        TTCGAAAATATGTTAGTACTCAATATGAATTCTTTGAGTTAAAAAAGATAAAGCAAA--A
AY727327        TTCGAAAATATGTTAGTACTCAATATGAATTCTTTGAGTTAAGAAAGATAAAGCAAA--A
AY727330        TTCAAAAATATGCTAGGACTGAATATGAATTCTTAAAGTTAAGAAAGATAAAGAAAAACA
AY727331        TTCAAAAATATGCTAGGACTGAATATGAATTCTTAAAGTTAAGAAAGATAAAGAAAAACA

AY727326        ATACATAATGTGATTTCAATATTCCAATTACCTAACAATACGGCTATCAATTAAACGATT
AY727327        ATACATAATGTGATTTCAATATTCCAATTACCTAACAATACGGCTATCAATTAAACGATT
AY727330        GTACATAATGTAAA----TTATTGCAA---------AAAACGGCTAACAATTAGACGATT
AY727331        GTACATAATGTAAA----TTATTGCAA---------AAAACGGCTAACAATTAGACGATT

AY727326        TTAGGATTACACCGACAAATATTAGGCCGATATGAATTTAACATCATGTTGTATTTAGAT
AY727327        TTAGGATTACACCGACAAATATTAGGCCGATATGAATTTACCATCATGTTGTATTTAGAT
AY727330        TTAGGATTACGCTGACAAATATTAGGATGATATTAATTTA------TCTTGTATTTAGAT
AY727331        TTAGGATTACGCTGACAAATATTAGGATGATATTAATTTA------TCTTGTATTTAGAT

AY727326        GCTGTCTTTTATTAACATTCATCATTAAAT-TTGGAACCTTTTGCATTTAAGAAGTACAT
AY727327        GCTGTCTTTTATTAACATTCATCATTAAAT-TTGGAACCTTTTGTATTTAAGAAGTACAT
AY727330        GCTGTCTTTTATCAACATTCATCACTAGATATTGGAACCTATTGCATCTAAGAAGTACAT
AY727331        GCTGTCTTTTATCAACATTCATCACTAGATATTGGAACCTATTGCATCTAAGAAGTACAT

AY727326        GTTTAATAGTGTTTAAAA-TATATATGAAATTGATCATAAGGA---TCTATAAATGCGGT
AY727327        GTTTAATAGTGTTTATAA-TATATATGAAATTGATCGTAAGGA---TCTATAAATGCAGT
AY727330        GTTTAATAGGGTT-AAAACTATATATGAAGTCGATTATAAGGAATTTCTATAAATGTAGC
AY727331        GTTTAATAGGGTT-AAAACTATATATGAAGTCGATTATAAGGAATTTCTATAAATGTAGC

AY727326        TCTTCAATTTCTTG
AY727327        TCTTCAATTTCTTG
AY727330        TCTTCAATTTCCTA
AY727331        TCTTCAATTTCCTA
```

Figure 3.3: Example.aln, an example output file produced from example.dawg (Figure 3.1)

ously published applications for simulating evolution with indels: Rose (Stoye et al., 1998) and EvolveAGene (Hall, 2005).

Stoye and colleagues do not derive Rose's model of indel formation, which differs from the model that I have derived for Dawg. In Rose, a binomial distribution with parameters TL and vp describes the number of insertions and deletions that occur along a branch. The parameter T is the branch length rounded to the nearest integer. Any branch length less than 0.5 will turn off insertions and deletions, which is a problem for estimates from standard nucleotide models which require branch lengths to be in substitution time. The parameter L is the length of the sequence at the bottom of the branch; unlike Dawg, Rose does not update the sequence length as new indels form along the branch. The p is the proportion of nucleotides that have a mutation rate greater than 1; Dawg does not associate the indel model with rate heterogeneity. The parameter v is the insertion or deletion threshold, which is similar to Dawg's parameter λ. Although both models can be made to produce similar distributions, Dawg's model is derived from a simple Poisson process and thus is consistent with the derivation of models for continuous-time substitution.

For indel size distributions, Rose only provides a user-based model. Dawg provides two models in addition to a user-based model: negative-binomial and power-law. Using power-law distributions for indel sizes is advantageous for researchers because they are biologically realistic. The basic substitution model in Rose is PAM, which was developed for protein sequences. For substitutions, Dawg uses GTR+Γ+I, which was developed for nucleotide sequences. Perhaps the most important difference between these two models is the meaning of the branch lengths. In PAM a branch length of 1 means that sequences have 1% divergence, whereas in GTR it means that each site is expected to have had one substitution.

Hall (2005) developed EvolveAGene using a methodology significantly different than the one employed here to develop Dawg. Whereas Dawg models the process of substitution, EvolveAGene models the separate processes of mutation and acceptance. EvolveAGene simulates the mutation of coding sequences based on the spontaneous mutational spectrum of *Escherichia coli*. EvolveAGene would need to be modified if another mutational spectrum is desired. Relying on

mutation spectra can be restrictive for researchers studying organisms for which the mutation spectra are unknown. Furthermore, EvolveAGene does not allow users to specify their own phylogenies and restricts tree topology to balanced, bifurcating trees. EvolveAGene is rather inflexible when compared to the many options available for Dawg.

Dawg also comes with a way to estimate parameters of indel formation setting it apart from Rose and EvolveAGene. This ability may prove useful for researchers interested in studying indel formation or using gaps to aid in estimating phylogenies. The parameter estimator is not perfect. It can be improved by assigning alignment gaps to individual branches. Furthermore, it estimates a net indel rate, instead of separating insertion and deletion into separate processes. Researchers interested in additional biological realism should consider separating the indel rate into insertion and deletion rates, favoring the deletion rate. For example, Zhang and Gerstein (2003) found that deletions occurred roughly three times more often than insertions in neutral DNA. This result suggests that $\lambda_I = (1/4)\,\lambda_{ID}$ and $\lambda_D = (3/4)\,\lambda_{ID}$ would be biologically realistic parameters.

Although the model of indel formation implemented in Dawg is an improvement over previous models, it does not take into consideration several biological features of indel formation. For instance, indel formation in Dawg is content independent, whereas natural indel formation is heavily influenced by repetitive sequences. Since repetitive sequences create indel hotspots, they also violate the assumption of uniform insertion and deletion rates. Modeling indel formation with extreme biological realism is hard at this time because many of the factors influencing hotspots remain unknown. However, despite these reservations Dawg's model of indel formation offers fruitful avenues for researchers who need to model sequence evolution.

Some researchers are reconstructing extinct genomes and are using simulations to test their methodology (Blanchette et al., 2004; Pennisi, 2005). Dawg is not designed explicitly to simulate genomes but can be utilized in that fashion. It contains many of the features described in genome simulations used by Blanchette and colleagues. Additionally, it can simulate recombination, which may prove useful in some contexts of studying genome reconstruction. However, it does not have

the ability to distinguish transposons from other indels or treat CpG regions differently than other sections of DNA. It also lacks a model of chromosomal rearrangement.

There are many possible features that can be added to Dawg. To improve realism, repetitive DNA and hotspots should be eventually included. Another important feature would be to allow separate GTR models for each block position just as separate Γ+I models are currently allowed. Another possibility is to allow each section of a sequence to evolve with a different evolutionary model. Furthermore, because Zipf distributions often have infinite means, incorporating a Lavalette distribution (Lavalette, 1996; Popescu, 2003; Popescu et al., 1997), which is a non-linear extension to a Zipf distribution, is probably more appropriate for indel lengths. I suspect that a Lavalette distribution may fit empirical indel size distributions better than a Zipf distribution. Incorporating inversions into Dawg's model of molecular evolution may be useful to some researchers. Other researchers might find the addition of protein models of evolution into the program to be quite useful. Other possible places for improvement are the estimation of parameters for indel formation and developing the option for Dawg to have time reversible models of indel formation.

Dawg is a portable, flexible, and robust program for simulating DNA sequence evolution with indels. It supports recombination, the general time reversible model, gamma rate heterogeneity, invariant sites, and indel formation. It is an improvement over existing programs by supporting a statistically derived model of indel formation.

3.7 REFERENCES

Akaike, H. (1974). New look at statistical-model identification. *IEEE Trans. Autom. Control AC19*, 716–723.

Altschul, S. F., W. Gish, W. Miller, E. W. Myers, and D. J. Lipman (1990). Basic local alignment search tool. *J. Mol. Biol. 215*, 403–410.

Benner, S. A., M. A. Cohen, and G. H. Gonnet (1993). Empirical and structural models for insertions and deletions in the divergent evolution of proteins. *J. Mol. Biol. 229*, 1065–1082.

Blanchette, M., E. D. Green, W. Miller, and D. Haussler (2004). Reconstructing large regions of an ancestral mammalian genome in silico. *Genome Res. 14*, 2412–2423.

Bull, J. J., C. W. Cunningham, I. J. Molineux, M. R. Badgett, and D. M. Hillis (1993). Experimental molecular evolution of bacteriophage-t7. *Evolution 47*, 993–1007.

Chang, M. S. S. and S. A. Benner (2004). Empirical analysis of protein insertions and deletions determining parameters for the correct placement of gaps in protein sequence alignments. *J. Mol. Biol. 341*, 617–631.

Clegg, M. T., B. S. Gaut, G. H. Learn, and B. R. Morton (1994). Rates and patterns of chloroplast DNA evolution. *Proc. Natl. Acad. Sci. U. S. A. 91*, 6795–6801.

Felsenstein, J. (1981). Evolutionary trees from DNA-sequences - a maximum-likelihood approach. *J. Mol. Evol. 17*, 368–376.

Felsenstein, J. (1984). Distance methods for inferring phylogenies - a justification. *Evolution 38*, 16–24.

Felsenstein, J. (2004). *Inferring Phylogenies*. Sinauer Associates, Inc. Sunderland, MA.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical-reactions. *J. Phys. Chem. 81*, 2340–2361.

Goldstein, M. L., S. A. Morris, and G. G. Yen (2004). Problems with fitting to the power-law distribution. *Eur. Phys. J. B 41*, 255–258.

Gotoh, O. (1982). An improved algorithm for matching biological sequences. *J. Mol. Biol. 162*, 705–708.

Gu, X. and W. H. Li (1995). The size distribution of insertions and deletions in human and rodent pseudogenes suggests the logarithmic gap penalty for sequence alignment. *J. Mol. Evol. 40*, 464–473.

Hall, B. G. (2005). Comparison of the accuracies of several phylogenetic methods using protein and dna sequences. *Mol. Biol. Evol. 22*, 792–802.

Hasegawa, M., H. Kishino, and T. A. Yano (1985). Dating of the human ape splitting by a molecular clock of mitochondrial-DNA. *J. Mol. Evol. 22*, 160–174.

Hillis, D. M., J. J. Bull, M. E. White, M. R. Badgett, and I. J. Molineux (1992). Experimental phylogenetics - generation of a known phylogeny. *Science 255*, 589–592.

Hillis, D. M., J. P. Huelsenbeck, and C. W. Cunningham (1994). Application and accuracy of molecular phylogenies. *Science 264*, 671–677.

Hudson, R. R. (2002). Generating samples under a wright-fisher neutral model of genetic variation. *Bioinformatics 18*, 337–338.

Jones, J. H. and M. S. Handcock (2003). An assessment of preferential attachment as a mechanism for human sexual network formation. *Proc. R. Soc. Lond. Ser. B-Biol. Sci. 270*, 1123–1128.

Jukes, T. H. and C. R. Cantor (1969). Evolution of protein molecules. In H. N. Munro (Ed.), *Mammalian Protein Metabolism*, Volume 3, pp. 21–132. Academic Press, New York.

Keightley, P. D. and T. Johnson (2004). Mcalign: Stochastic alignment of noncoding dna sequences based on an evolutionary model of sequence evolution. *Genome Res. 14*, 442–450.

Kimura, M. (1980). A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide-sequences. *J. Mol. Evol. 16*, 111–120.

Kimura, M. (1981). Estimation of evolutionary distances between homologous nucleotide-sequences. *Proc Nat Acad Sci Us-Biol Sci 78*, 454–458.

Kuhner, M. K. and J. Felsenstein (1994). Simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Mol. Biol. Evol. 11*, 459–468.

Lanave, C., G. Preparata, C. Saccone, and G. Serio (1984). A new method for calculating evolutionary substitution rates. *J. Mol. Evol. 20*, 86–93.

Lavalette, D. (1996). Facteur d'impact: impartialite ou impuissance? Internal report, Inserm U350, Institut Curie-Recherche, Centre Universitaire, Orsay, France.

Miklós, I., G. A. Lunter, and I. Holmes (2004). A "long indel" model for evolutionary sequence alignment. *Mol. Biol. Evol. 21*, 529–540.

Miller, W. and E. W. Myers (1988). Sequence comparison with concave weighting functions. *Bull. Math. Biol. 50*, 97–120.

Pearson, W. R. and D. J. Lipman (1988). Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA 85*, 2444–2448.

Pennisi, E. (2005). Extinct genome under construction. *Science 308*, 1401–1402.

Popescu, I. I. (2003). On a zipf's law extension to impact factors. *Glottometrics 6*, 83–93.

Popescu, I. I., M. Ganciu, M. C. Penache, and D. Penache (1997). On the lavalette ranking law. *Rom. Rep. Phys. 49*, 3–27.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery (1992). *Numerical Recipes in C: The Art of Scientific Computing* (Second ed.). Cambridge University Press, New York.

Rambaut, A. and N. C. Grassly (1997). Seq-gen: An application for the monte carlo simulation of dna sequence evolution along phylogenetic frees. *Comput. Appl. Biosci. 13*, 235–238.

Rodríguez, F., J. L. Oliver, A. Marín, and J. R. Medina (1990). The general stochastic-model of nucleotide substitution. *J. Theor. Biol. 142*, 485–501.

Schwarz, G. (1978). Estimating dimension of a model. *Ann. Stat. 6*, 461–464.

Shaw, J., E. B. Lickey, J. T. Beck, S. B. Farmer, W. S. Liu, J. Miller, K. C. Siripun, C. T. Winder, E. E. Schilling, and R. L. Small (2005). The tortoise and the hare ii: Relative utility of 21 noncoding chloroplast dna sequences for phylogenetic analysis. *Am. J. Bot. 92*, 142–166.

Stoye, J., D. Evers, and F. Meyer (1998). Rose: generating sequence families. *Bioinformatics 14*, 157–163.

Swofford, D. L. (2002). *PAUP*: Phylogenetic Analysis Using Parsimony (and Other Methods) 4.0 Beta*. Sinauer Associates, Inc, Sunderland MA.

Tamura, K. and M. Nei (1993). Estimation of the number of nucleotide substitutions in the control region of mitochondrial-DNA in humans and chimpanzees. *Mol. Biol. Evol. 10*, 512–526.

Tavaré, S. (1986). Some probabilistic and statistical problems on the analysis of dna sequences. *Lectures in mathematics in the life sciences 17*, 57–86.

Thompson, J. D., D. G. Higgins, and T. J. Gibson (1994). Clustal-w - improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res. 22*, 4673–4680.

Thorne, J. L., H. Kishino, and J. Felsenstein (1991). An evolutionary model for maximum-likelihood alignment of DNA-sequences. *J. Mol. Evol. 33*, 114–124.

Thorne, J. L., H. Kishino, and J. Felsenstein (1992). Inching toward reality - an improved likelihood model of sequence evolution. *J. Mol. Evol. 34*, 3–16.

Waddell, P. J. and M. A. Steel (1997). General time-reversible distances with unequal rates across sites: Mixing gamma and inverse gaussian distributions with invariant sites. *Mol. Phylogenet. Evol. 8*, 398–414.

Yang, Z. B. (1994). Estimating the pattern of nucleotide substitution. *J. Mol. Evol. 39*, 105–111.

Yang, Z. H. (1993). Maximum-likelihood-estimation of phylogeny from DNA-sequences when substitution rates differ over sites. *Mol. Biol. Evol. 10*, 1396–1401.

Yang, Z. H. (1995). On the general reversible markov process model of nucleotide substitution - reply. *J. Mol. Evol. 41*, 254–255.

Yang, Z. H. (1997). Paml: a program package for phylogenetic analysis by maximum likelihood. *Comput. Appl. Biosci. 13*, 555–556.

Zhang, Z. L. and M. Gerstein (2003). Patterns of nucleotide substitution, insertion and deletion in the human genome inferred from pseudogenes. *Nucleic Acids Res. 31*, 5338–5348.

CHAPTER 4

LOGARITHMIC GAP COSTS DECREASE ALIGNMENT ACCURACY[1]

[1]Cartwright, R.A. (submitted) *PLoS Computational Biology*.

4.1 INTRODUCTION

Sequence alignments are essential to the study of molecular biology and systematics because they purport to reveal regions in sequences that are homologous. Because sequences gain and lose residues as they evolve, alignments are necessary for revealing such gaps in sequence data. Therefore, researchers usually need to align sequences before they can be studied. For example, most algorithms that construct phylogenetic trees from sequences require a sequence alignment (Swofford, 2002). Since alignments are an integral part of many research programs, the quality of the inferences we make from alignments depends on the quality of the alignments themselves (e.g. Odgen and Rosenberg, 2006).

There are two main types of alignments: local and global. A local alignment (e.g. BLAST; ref Altschul et al., 1990) attempts to align only parts of sequences often avoiding gaps, whereas a global alignment (e.g. CLUSTAL W; ref Thompson et al., 1994) attempts to align entire sequences, creating gaps. This study will focus on global alignments. The most common way to globally align pairs of sequences is through dynamic programming (Gotoh, 1982; Miller and Myers, 1988; Needleman and Wunsch, 1970; Waterman, 1984; Waterman et al., 1976). Using dynamic programming one can find the alignments that have the lowest cost based on costs for matches, mismatches, and gaps. However, alignment accuracy depends on the assumptions used in picking these costs. For example, gap costs are typically based on the affine model, where the cost of a gap of length k is $G(k) = a + bk$ (Gotoh, 1982). This gap cost is easy to implement, fast, and efficient. Furthermore, since nucleotides are deleted or inserted in groups, it is biologically plausible that gaps should cost more to create than they do to extend, and affine gap costs can model this. However, some researchers have raised questions about the biological justification for the affine gap model.

Studies on the distribution of indel lengths have revealed that the size of an indel is linearly related to its frequency on a log-log scale, and therefore gap-sizes obey a power law (Benner et al., 1993; Chang and Benner, 2004; Gu and Li, 1995; Zhang and Gerstein, 2003). Under a Zipfian power-law distribution, the probability that an indel has length k is $P(k|z) = k^{-z}/\zeta(z)$, where

$\zeta(z) = \sum_{n=1}^{\infty} n^{-z}$ is Reimann's Zeta function and $z > 1$. If $1 < z \leq 2$, the mean of this distribution is infinite, and if $1 < z \leq 3$, the variance is infinite. The observation that indel lengths obey a power law suggests that sequences should be aligned using logarithmic gap costs, i.e. $G(k) = a + c\ln(k)$ (Gu and Li, 1995). However, as mentioned above, the standard method of sequence alignment uses affine gap costs, i.e. $G(k) = a + bk$ (Gotoh, 1982). However, researchers cannot adapt Gotoh's algorithm to logarithmic gap costs. Instead researchers must use the more computationally expensive candidate list method of Waterman (Waterman, 1984) as optimized by Miller and Myers (Miller and Myers, 1988). Although, affine gap costs are efficient, this study seeks to determine whether this efficiency comes with a cost to accuracy.

An alignment is essentially a hypothesis about the evolutionary history of the sequences, specifying formally which residues are homologous to one another. We can define a measurement of alignment accuracy by comparing the hypothesized alignment to the "true" alignment of the sequence pair. An alignment consists of a set of columns which provide per residue homology statements, e.g. residue 100 of sequence A aligns with residue 90 of sequence B. When comparing two alignment, columns fall into three different categories: 1) columns only appearing in the first alignment, and 2) columns only appearing in the second alignment, and 3) columns appearing in both alignments. By counting the number of columns belonging to each category, it is possible to measure how identical two alignments are to one another:

$$I = \frac{2 \times K_3}{2 \times K_3 + K_1 + K_2} \tag{4.1}$$

where K_c is the number of columns in category c. See Figure 4.1 for an example of this measurement. This alignment identity can be used to measure the accuracy of a hypothesized alignment.

Not all sequence pairs are equally easy to align, and the accuracy of a hypothesized alignment is expected to decrease as sequence pairs become more distantly related due to substitution saturation and indel accumulation. Therefore, an appropriate measure of alignment accuracy for a gap cost needs to average across multiple branch lengths and multiple sequence pairs. Branch lengths are often measured in "substitution time", where a unit branch length is equal to 1 substitution,

	1	2	3	4	5	6	7	8
A:	A	C	G	T	A	C	G	T
B:	A	C	G	T	–	–	G	T
	1	2	3	4	–	–	5	6

	1	2	3	4	5	6	7	8
A:	A	C	G	T	A	C	G	T
B:	A	C	G	T	G	T	–	–
	1	2	3	4	5	6	–	–

Figure 4.1: Example alignment pair. Numbers identify the residues in the sequences. K_3 columns—A1B1, A2B2, A3B3, and A4B4—are found in both alignments. K_1 columns—A5B–, A6B–, A7B5, and A8B6—are found in only the left alignment. K_2 columns—A7B–, A8B–, A5B5, and A6B6—are found in only the right alignment. Alignment identity is $I = (2K_3) / (2K_3 + K_1 + K_2) = (2 \times 4) / (2 \times 4 + 4 + 4) = 1/2$.

on average, per nucleotide. According to coalescent theory and neutrality, the number of generations separating any pair of sequences in the same population depends on the effective population size, N_e, and has an exponential distribution with mean $4N_e$ (Hein et al., 2005, p24). If μ is the instantaneous rate of substitution, then the substitution time separating any two sequences has an exponential distribution with mean $\theta = 4N_e\mu$. As branch lengths get longer and sequences become more distant, data is lost from the sequences, and thus no alignment algorithm may be able to recover the true alignment. This limitation can be corrected on a per sequence pair basis by using relative alignment identities: absolute alignment identities divided by the maximum alignment identity found for that sequence pair.

4.2 Results

For the set of sequence pairs, the minimum branch length for any pair was 1.83×10^{-05} mean substitutions per nucleotide, and the maximum branch length was 1.76. Furthermore, the distribution of observed gap sizes, plotted on a log-log scale, is shown in Figure 4.2. This distribution clearly obeys a power-law. The maximum likelihood estimation of the power-law parameter of this distribution is $z = 1.53$.

Alignments were classified via their parameter values into three different schemes. All parameter sets belonged to the log-affine scheme. The affine and logarithmic schemes were subsets of the

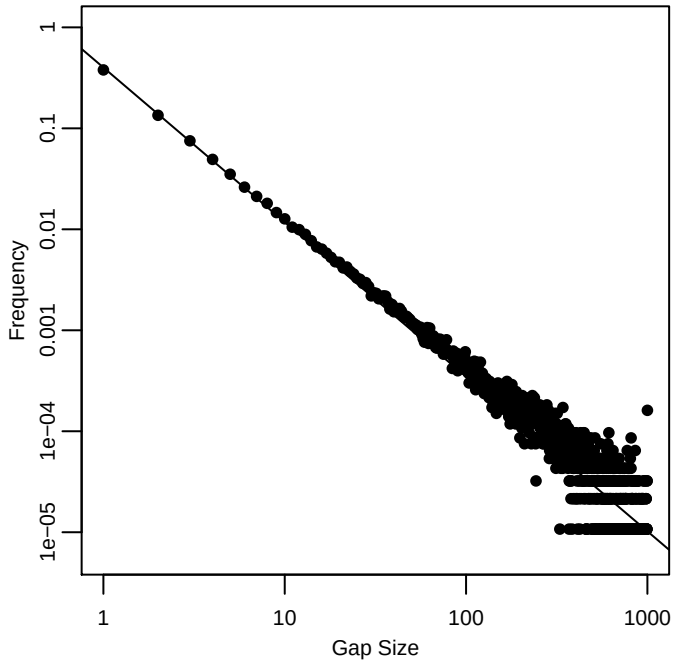
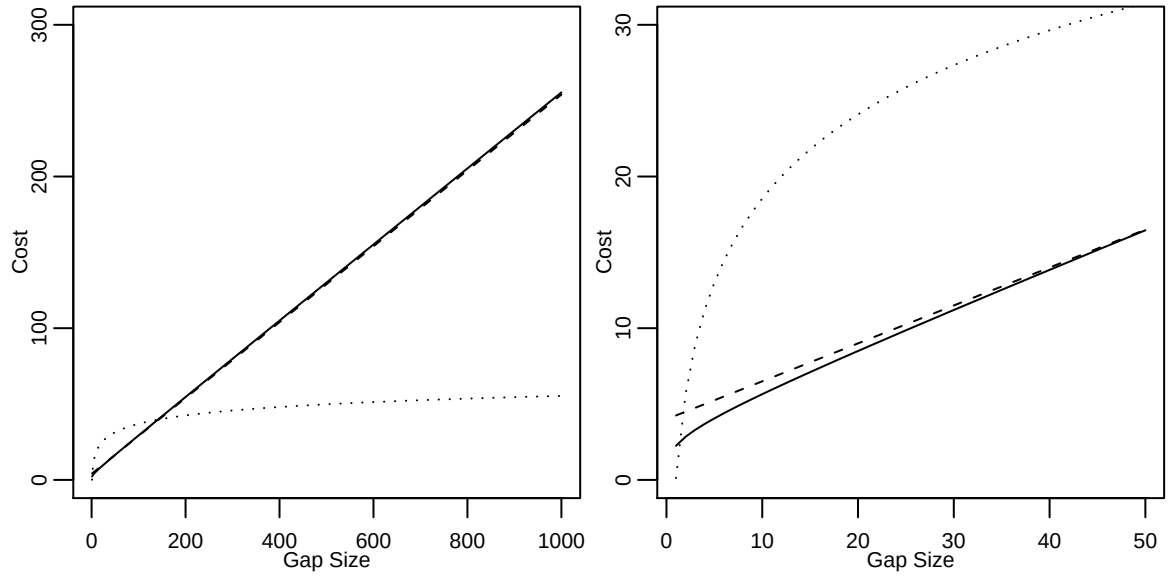


Figure 4.2: Gap sizes obey a power law. Log-Log plot of the distribution of gap sizes measured from the 5000 true alignments. The line is the maximum likelihood fit of a power-law distribution: $\ln f(k) = 0.915 - 1.53 \ln k$

log-affine scheme and consisted of the parameter sets where $c = 0$ and $b = 0$, respectively. Analysis of alignment accuracy was divided into two broad and different questions. How do the best gap costs for each scheme compare to one another? And how do the maximum alignment accuracy for each scheme compare to one another for each sequence pair? The first question investigates what happens if researchers use a single gap cost across many alignments, and the second investigates happens if researchers optimize gap costs to each alignment.

The best gap costs were identified by having the highest average alignment accuracy, i.e. they produced alignments that had the highest average identity to the "true" alignments. The best costs for aligning sequences under the log-affine, affine, and logarithmic schemes were identified respectively as $G(k) = 2 + 1/4k + 1/2 \ln k$ (average identity of 0.941), $G_A(k) = 4 + 1/4k$ (average identity of 0.925), and $G_L(k) = 1/8 + 8 \ln k$ (average identity of 0.687). Figure 4.3 shows the graphs



(a) Full Range (b) Lower Range

Figure 4.3: The curves of the best gap costs. A) The entire range of the curves and B) a magnification of the beginning of the curves. The best gap costs were decided for each scheme based on highest average alignment identity. Log-affine $G(k) = 2 + 1/4k + 1/2 \ln k$ (solid), affine $G_A(x) = 4 + 1/4k$ (dashed), and logarithmic $G_L(k) = 1/8 + 8 \ln k$ (dotted).

of these gap costs, and Figure 4.4 shows the densities of their identities. Log-affine and affine both peak a little below 100% identity, whereas the logarithmic density is nearly flat for most of the parameter space before barely peaking below 100% identity. Table 4.1 and Table 4.2 present some statistical properties of these gap penalties. The best log-affine cost produced alignments that were only slightly better than the ones produced by the best affine cost. Both log-affine and affine costs produced alignments that were considerably better than the ones produced by the best logarithmic cost. In fact, the best log-affine gap cost produced the best alignments for over half the sequence pairs.

Figure 4.5 looks at the distribution of identities produced by each best cost. Figure 4.5a–c plots the identities with respect to their branch lengths, transformed to a uniform scale. Figure 4.5d–f are box-whisker plots of identities grouped into 20 classes based on branch length. The best

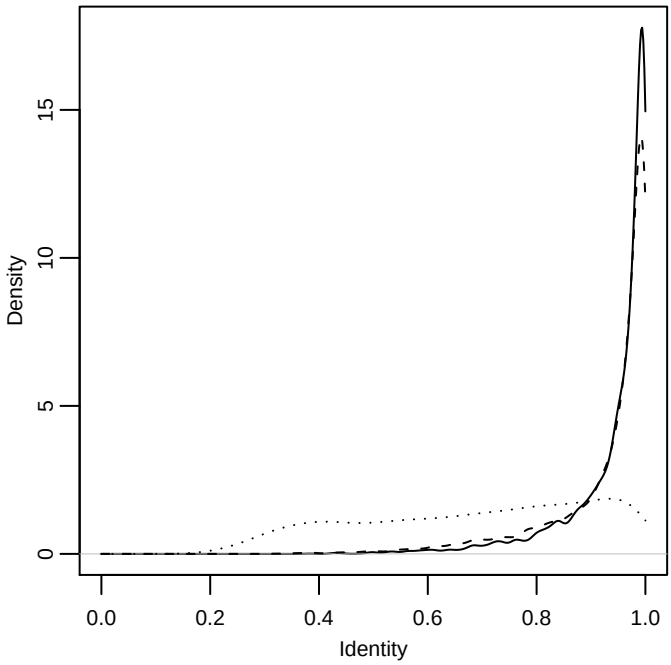


Figure 4.4: Accuracy distribution of best gap costs. Best log-affine (solid), best affine (dashed), and best logarithmic (dotted). Accuracy is measured via alignment identity. See Figure 4.3 for details on the exact gap costs.

Table 4.1: Absolute accuracy properties of the best gap costs. Accuracy is measured via alignment identity. Log-Affine: $G(k) = 2 + 1/4k + 1/2 \ln k$. Affine: $G_A(k) = 4 + 1/4k$. Logarithmic: $G_L(k) = 1/8 + 8 \ln k$.

	Absolute Identities		
	Log-Affine	Affine	Logarithmic
Minimum	0.383	0.324	0.183
1st Quartile	0.926	0.904	0.512
Mean	0.941	0.925	0.687
Median	0.976	0.970	0.717
3rd Quartile	0.994	0.992	0.874
Maximum	1.0	1.0	1.0

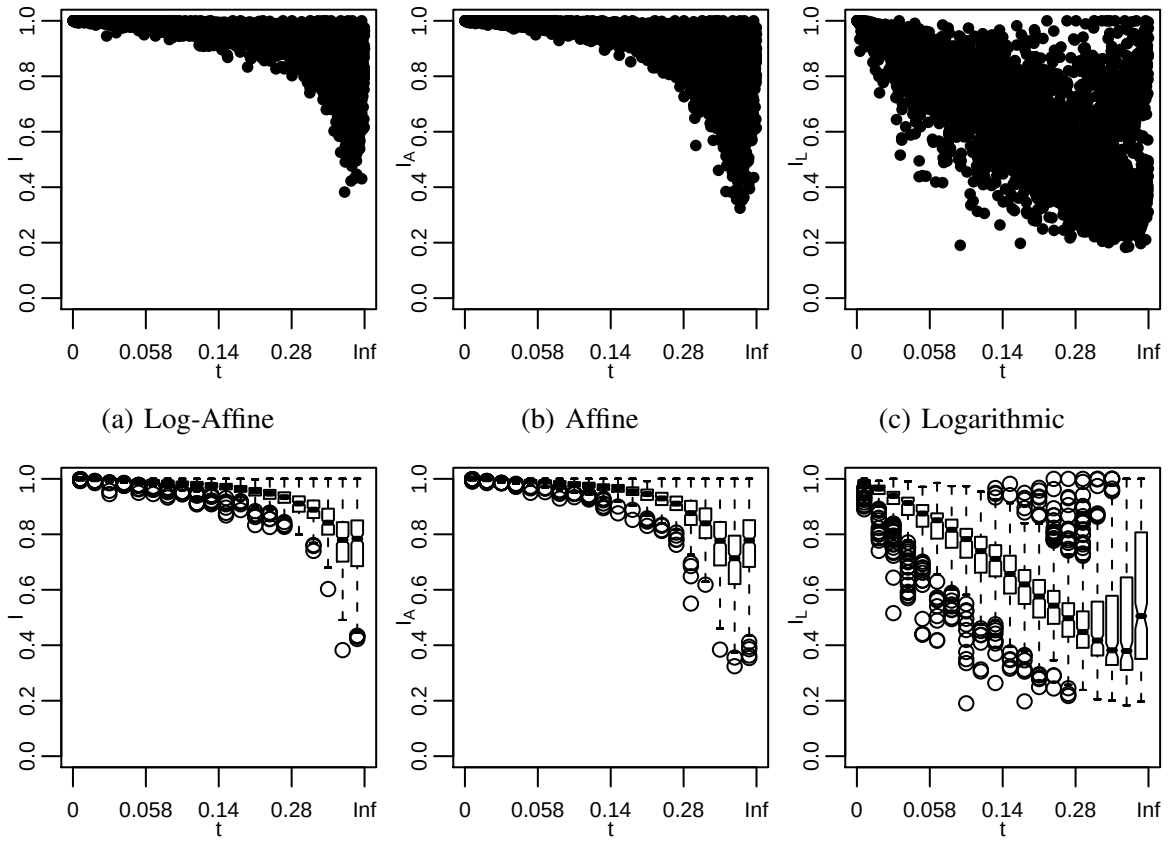
Table 4.2: Relative accuracy properties of the best gap costs. Relative accuracy was calculated as the alignment identity produced by a gap cost for each sequence pair divided by the largest alignment identity produced by any gap cost for the same sequence pair. See notes of Table 4.1.

	Relative Identities		
	Log-Affine	Affine	Logarithmic
Minimum	0.710	0.501	0.193
1st Quartile	0.993	0.971	0.549
Mean	0.992	0.973	0.717
Median	1.0	0.993	0.745
3rd Quartile	1.0	1.0	0.892
Maximum	1.0	1.0	1.0

logarithmic gap cost produces alignments with much lower identities than the best log-affine and affine costs. As expected, identities decrease as branch length increases; however, unexpectedly, the largest branch lengths show increasing alignment identity.

To compare the best gap costs on a per sequence pair basis, Figure 4.6 shows the ratio of affine and logarithmic alignment identities to log-affine alignment identities, plotted via branch length for each sequence pair. The identities produced by the best log-affine gap cost tend to be higher than or equal to the identities produced by the pest affine and logarithmic gap costs. However, there are some sequences for which the best log-affine gap cost produces an alignment that is worse than the alignment produced by the best affine or logarithmic cost. Nevertheless, the best affine costs compare rather well to the best log-affine costs, especially at lower branch lengths. However, the best logarithmic costs do a poor job compared to the best log-affine costs and the best affine costs. Clearly alignments derived from logarithmic costs are of poor quality, and highly sensitive to the divergence between sequences.

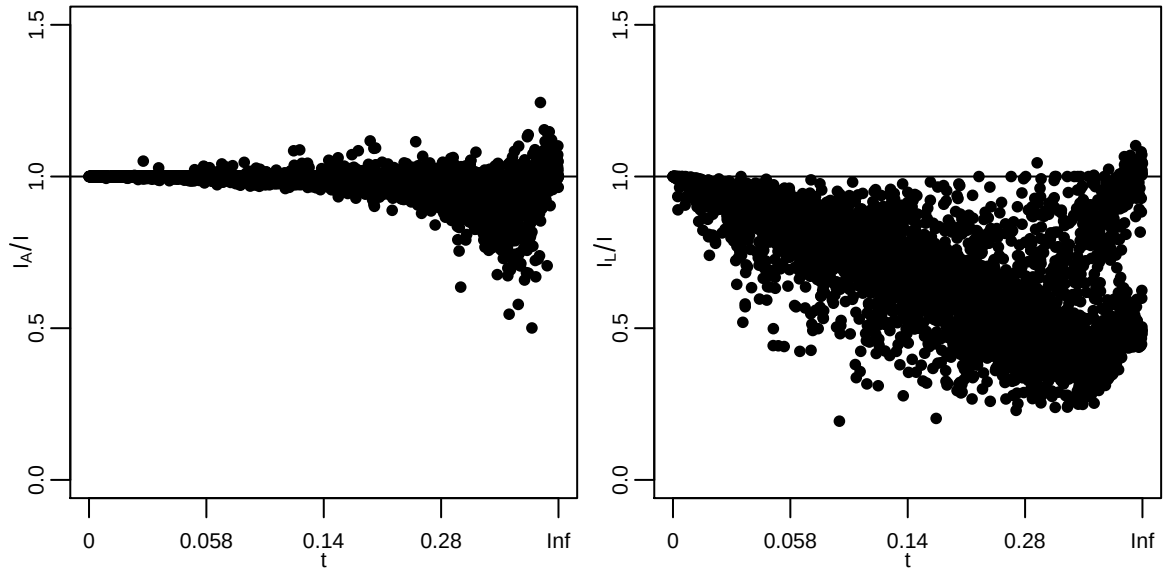
Instead of trying to find gap costs that have the highest average accuracy, we can find the gap costs that have the highest accuracy for each sequence. Therefore, an alternative approach to comparing schemes is to look at the maximum identity produce by each scheme. Similar to Figure 4.5, Figure 4.7 shows the maximum identities of each scheme plotted by transformed branch



(a) Log-Affine (b) Affine (c) Logarithmic

(d) Log-Affine (e) Affine (f) Logarithmic

Figure 4.5: Accuracies of best costs plotted by divergence. I, I_A, and I_L are the alignment identities produced by the best log-affine, affine, and logarithmic gap penalties, respectively. See Figure 3 for more information. a-c) Alignment identities plotted by the branch length of the alignments. Divergence time is plotted on a uniform scale, $u = 1 - \exp(-t/\bar{t})$. d-f) Box-whisker plots of identities grouped into 20 bins of 250 values. Solid bars are medians. Notches are significant range of medians. Bars are the mid-range. Whiskers are the range. Circles are outliers.



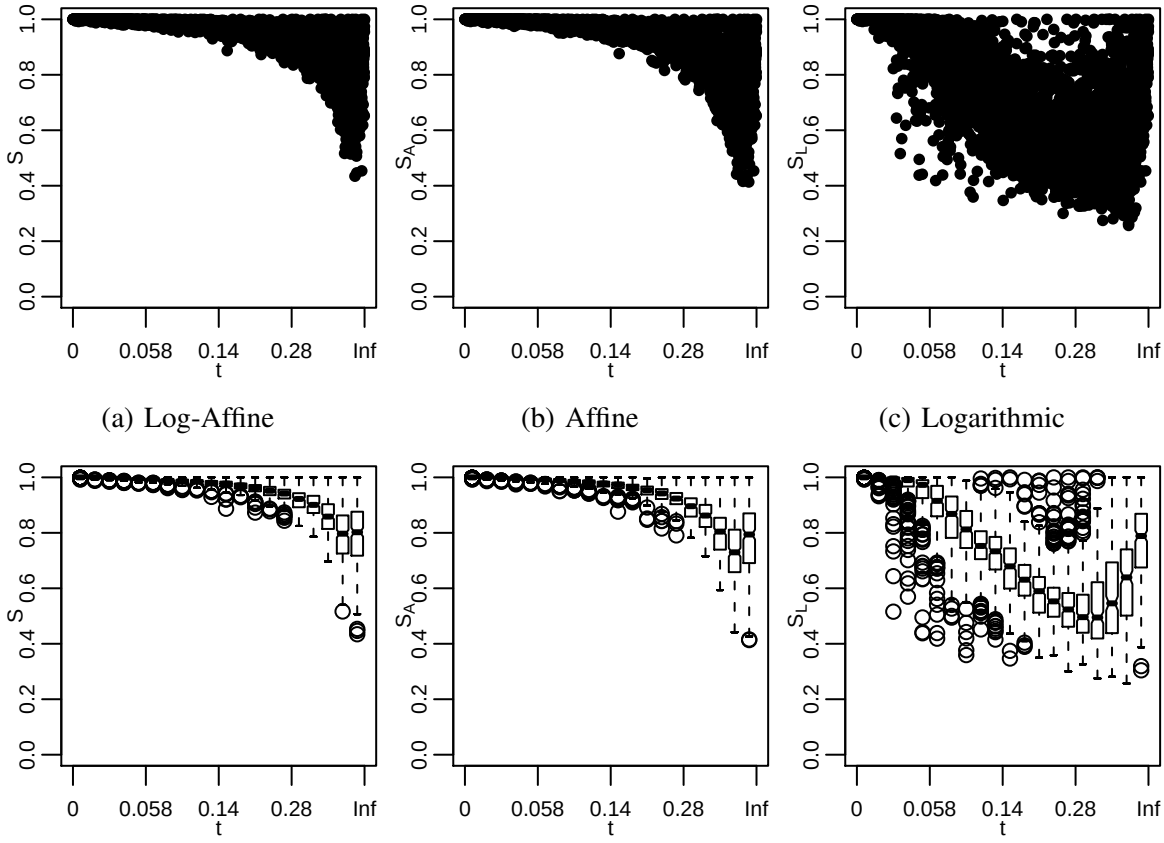
(a) Affine (b) Logarithmic

Figure 4.6: Accuracies of best costs compared per sequence. Ratio of identities produced by a) best affine gap cost and b) best logarithmic gap cost to the identities produced by the log-affine gap cost plotted for each sequence pair by divergence time. See Figure 4.5 for more information.

length, and box-whisker plots of the data. As we saw in the best costs analysis, the maximum affine identities are similar to maximum log-affine identities, and both are distinct from the maximum logarithmic identities. Identities decrease with increasing branch, only to increase with the largest branch lengths. Furthermore, logarithmic densities are once again very sensitive to increasing branch lengths. Similar to Figure 4.6, Figure 4.8 shows the ratio of maximum identities of affine and logarithmic to the log-affine schemes. Once again, the affine scheme has identities similar to the log-affine scheme and the logarithmic scheme does not.

4.3 Discussion

The first issue to consider is whether the parameter space was properly sampled. For log-affine and affine schemes, the best values were found inside the sampled parameter space, representing local maxima and perhaps global maxima. However, for logarithmic gap penalties, the best penalty was

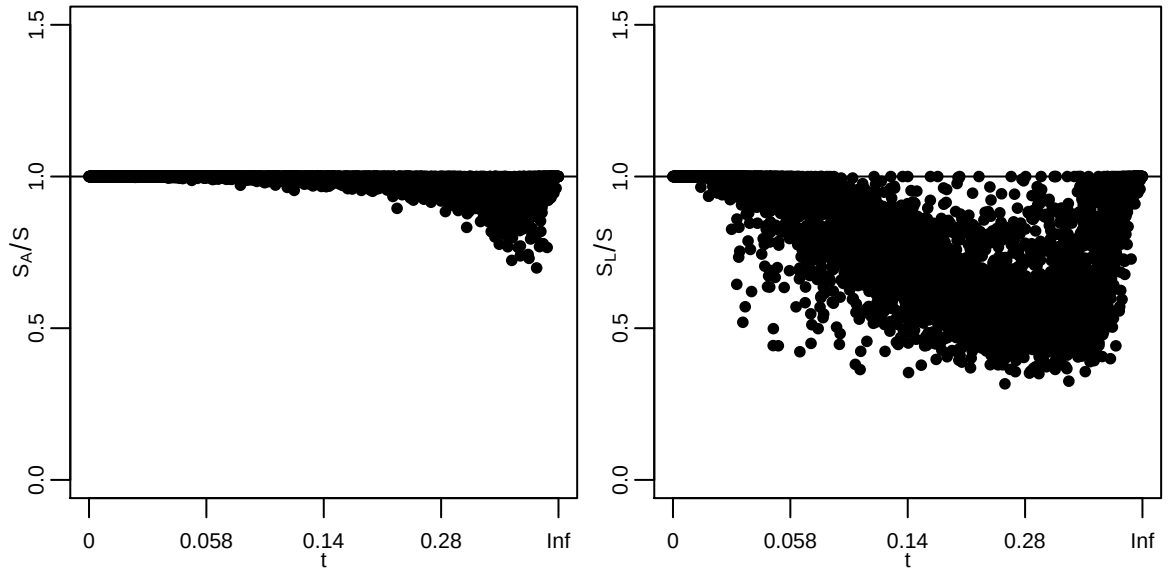


(a) Log-Affine (b) Affine (c) Logarithmic

(d) Log-Affine (e) Affine (f) Logarithmic

Accuracies of best costs plotted by divergence

Figure 4.7: Maximum accuracies plotted by divergence. S, S_A, and S_L are the maximum alignment identity produced for each sequence pair by log-affine, affine, and logarithmic gap costs respectively. The subfigures are the same as in Figure 4.5.



(a) Affine (b) Logarithmic

Figure 4.8: Maximum accuracies compared per sequence. Ratio of maximum identities produced by a) affine gap costs and b) logarithmic gap costs to the maximum identities produced by log-affine gap costs plotted for each sequence pair by divergence time. See Figure 4.5-Figure 4.7 for more information.

found on the edge of the parameter space. Subsequent expansion of the parameter space confirmed that $G(k) = 1/8 + 8\ln k$ represents a local maximum for logarithmic gap costs.

In the simulations, branch lengths were randomly drawn based on $\theta = 4N_e\mu = 0.2$. If the per-nucleotide mutation rate is $\mu = 10^{-9}$, then the effective population size would be 50 million. This is high for most populations, but it does produce many branch lengths that can represent species-species divergence times. When calculating the best gap costs, it is possible to weight the identities in a way that reflects another distribution of branch lengths. Similar results were obtained when weighting to produce a $\theta = 0.002$ distribution.

An interesting feature of the data is that alignment identity improves at the longest branch lengths. This can be attributed to the fact that sequences at long branch lengths, although related, are saturated with indels and thus have very few nucleotides homologous to one another anymore.

Therefore, hypothesized alignments that are also dominated by gaps show high identity to the true alignment.

Clearly from the results, logarithmic gap costs are a poor choice for aligning sequences even though biological results would seem to suggest them. Logarithmic gap costs perform poorly because they increase slowly Figure 4.3. This causes logarithmic costs to "cheat" during pair-wise alignments because two huge gaps, covering the entirety of the sequences may be less costly than three or more moderate gaps. In fact, many logarithmic costs have bimodal distributions; they either work or cheat. However, this may not be a problem because it is easy to tell when logarithmic costs cheat. Log-affine gap costs are noticeably better than simple affine gap costs, even though the difference may not be enough to justify wide spread usage given the slower speed of the candidate list method. According to the above results, affine gap costs only diverge from log-affine gap penalties at large branch lengths.

It is definitely surprising that logarithmic gap costs do so poorly compared to affine and log-affine gap costs, given that initially there seems to be little biological justification for having a linear component in the gap cost. However, as I show in Appendices 4.A and 4.B, converting a maximum likelihood search into a minimum cost search introduces a linear component into the gap cost which can dominate the logarithmic component. In other words, the power law does not imply that gap costs should be logarithmic, instead it implies that gap costs should be log-affine.

From Appendix 4.A, the log-likelihood of a pairwise, global alignment given sequences A and B is

$$\ln L\left(Aln\,|A,B\right) = M\ln\left(1+3e^{-4\theta/3}\right) + R\ln\left(1-e^{-4\theta/3}\right) + \sum_{g=1}^{G}\left[\ln\left(e^{\lambda\theta}-1\right) - \ln\zeta\left(z\right) - z\ln k_g\right] \quad (4.2)$$

and from Appendix 4.B the gap cost derived from Equation 4.2 is

$$G(k) = \frac{\ln \zeta(z) - \ln\left(e^{\lambda\theta} - 1\right) + \ln\left(1 + 3e^{-4\theta/3}\right) k/2 + z \ln k}{\ln\left(1 + 3e^{-4\theta/3}\right) - \ln\left(1 - e^{-4\theta/3}\right)} \tag{4.3}$$

Since $\theta = 0.2$, $\lambda = 0.15$, and $z = 1.5$, Equation 4.2 reduces to

$$\ln L(Aln|A,B) = 1.19M - 1.45R - \sum_{g=1}^{G} [4.45 + 1.5 \ln k_g] \tag{4.4}$$

and Equation 4.3 reduces to

$$G(k) = 1.69 + 0.23k + 0.56 \ln k \tag{4.5}$$

This gap cost is very close to the top gap cost found in the simulations, $G(k) = 2 + 0.25k + 0.5 \ln k$. Furthermore, based on unweighted least squares, the following affine cost bests fits Equation 4.5, $G(k) = 4.17 + 0.23k$ (unweighted mean squared error of 0.0722). This cost is very close to the best affine cost found in the simulations, $w'_k = 4 + 0.25k$. Furthermore, because the linear component Equation 4.5 dominates the logarithmic component, logarithmic costs costs will clearly provide worse fits than affine gap costs. Therefore, one can surmise that the linear component to the gap cost function derives from the conversion of a maximum likelihood search into a minimum cost search. Furthermore, this linear component dominates the gap cost allowing the log component to be removed.

From these results I propose that, if a researcher knows θ, λ, and z for a group of sequences that he wants to align using a match cost of 0 and a mismatch cost of 1, he can estimate a log-affine gap cost via Equation 4.3. Furthermore, an affine gap cost can be estimated by fitting $G(k) = a + bk$ to Equation 4.3 via the method of least squares. However, researchers will find more utility if the procedure outlined in this paper was extended to the models of sequence evolution beyond Jukes-Cantor. In subsequent research, I hope to apply this procedure to more complex models as well as to unrooted trees.

This research has demonstrated that logarithmic gap costs, although suggested by biological data on the surface, are not a good solution for aligning pairs of sequences through dynamic pro-

gramming. In fact, despite previous suggestions, e.g. Gu and Li (1995), the power law does not imply that gap costs should be logarithmic, instead it implies that gap costs should be log-affine. Furthermore, the results find that affine gap costs can serve as a good approximation to log-affine gap costs. Because affine gap costs are quick, efficient, and currently nearly ubiquitous, this research strengthens the rational for existing practices in molecular biology.

4.4 Materials and Methods

Five thousand sequence pairs were generated on unroot trees using the sequence simulation program, Dawg (Cartwright, 2005). Dawg is a sequence simulation program that combines the general time reversible substitution model with a continuous time indel formation model. It is the only sequence simulation program capable of using the power-law model for indel lengths. Each simulation done by Dawg started with a random sequence of 1000 nucleotides. For each ancestral sequence, a single descendant sequence was evolved by Dawg based on the branch length separating the ancestor from the descendant. The branch lengths were drawn from an exponential distribution with a mean of $\theta = 0.2$. Because sequences were to be aligned using equal costs for each mismatch type, the sequences were evolved under the Jukes-Cantor substitution model (Jukes and Cantor, 1969). Indels were created at a rate of 15 per 100 substitutions (Cartwright, 2005), and their lengths were distributed via a truncated power-law with parameter of 1.5 (Zhang and Gerstein, 2003) and a cut-off of 1000 nucleotides. The observed distribution of gaps was checked to see if it obeyed a power-law, and the power-law parameter was estimated using maximum likelihood (Goldstein et al., 2004). Dawg recorded the actual alignment of each sequence pair making it possible to measure the accuracy of alignments generated through dynamic programming.

Pairwise, global alignments were done with Ngila (Cartwright, 2006), an implementation of the candidate-list dynamic programming algorithm of Miller and Myers (Miller and Myers, 1988) for logarithmic gap penalties. The cost of a match was 0 and a mismatch 1. Each sequence pair was aligned using 512 different parameter sets, which specified the coefficients of the gap cost function, $G(k) = a + bk + c\ln k$. Each coefficient was one of eight values: 0, $1/8$, $1/4$, $1/2$, 1, 2, 4,

or 8. The alignment identity (Equation 4.1) of each of these 2.56 million hypothesized alignments was calculated with respect to the appropriate true alignment produced by Dawg. Expansion of the parameter space to verify the local maximum for logarithmic gap penalties used $a = 16$.

The statistical software, R (R Development Core Team, 2006), was used to analyze the alignment identities and produce most figures. Fitting affine gap costs to the optimal gap costs was done via the method of least squares for gap sizes 1 to 1000. The squared error was minimized using the optimization procedure in PopTools 2.7.1 (Hood, 2006).

4.5 References

Altschul, S. F., W. Gish, W. Miller, E. W. Myers, and D. J. Lipman (1990). Basic local alignment search tool. *J. Mol. Biol. 215*, 403–410.

Benner, S. A., M. A. Cohen, and G. H. Gonnet (1993). Empirical and structural models for insertions and deletions in the divergent evolution of proteins. *J. Mol. Biol. 229*, 1065–1082.

Cartwright, R. A. (2005). Dna assembly with gaps (dawg): Simulating sequence evolution. *Bioinformatics 22*(Suppl. 3), iii31–iii38.

Cartwright, R. A. (2006). Ngila: Logarithmic sequence alignments. Software available from author.

Chang, M. S. S. and S. A. Benner (2004). Empirical analysis of protein insertions and deletions determining parameters for the correct placement of gaps in protein sequence alignments. *J. Mol. Biol. 341*, 617–631.

Goldstein, M. L., S. A. Morris, and G. G. Yen (2004). Problems with fitting to the power-law distribution. *Eur. Phys. J. B. 41*, 255–258.

Gotoh, O. (1982). An improved algorithm for matching biological sequences. *J. Mol. Biol. 162*, 705–708.

Gu, X. and W. H. Li (1995). The size distribution of insertions and deletions in human and rodent pseudogenes suggests the logarithmic gap penalty for sequence alignment. *J. Mol. Evol. 40*, 464–473.

Hein, J., M. Schierup, and C. Wiuf (2005). *Gene Genealogies, Variation and Evolution: A Primer in Coalescent Theory*. Oxford University Press, New York.

Hood, G. (2006). Poptools. Accessible on the Internet: `http://www.cse.csiro.au/poptools/`.

Jukes, T. H. and C. R. Cantor (1969). Evolution of protein molecules. In H. N. Munro (Ed.), *Mammalian Protein Metabolism*, Volume 3, pp. 21–132. Academic Press, New York.

Miller, W. and E. W. Myers (1988). Sequence comparison with concave weighting functions. *Bull. Math. Biol. 50*, 97–120.

Needleman, S. B. and C. D. Wunsch (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol. 48*, 443–453.

Odgen, T. and M. Rosenberg (2006). Multiple sequence alignment accuracy and phylogenetic inference. *Systematic Biology 55*, 314–328.

R Development Core Team (2006). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Smith, T. F., M. S. Waterman, and W. M. Fitch (1981). Comparative biosequence metrics. *J. Mol. Evol. 18*, 38–46.

Swofford, D. L. (2002). *PAUP*: Phylogenetic Analysis Using Parsimony (and Other Methods) 4.0 Beta*. Sinauer Associates, Inc, Sunderland MA.

Thompson, J. D., D. G. Higgins, and T. J. Gibson (1994). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res. 22*, 4673–4680.

Waterman, M. S. (1984). Efficient sequence alignment algorithms. *J. Theor. Biol. 108*, 333–337.

Waterman, M. S., T. F. Smith, and W. A. Beyer (1976). Some biological sequence metrics. *Advances in Mathematics 20*, 367–387.

Zhang, Z. and M. Gerstein (2003). Patterns of nucleotide substitution, insertion and deletion in the human genome inferred from pseudogenes. *Nucleic Acids Res. 31*, 5338–5348.

APPENDIX 4.A ALIGNMENT LOG-LIKELIHOOD

To find the most likely alignment we need a measurement of the likelihood of an alignment given the sequence pair, $L(Aln|A,B)$. This likelihood is proportional to the density of the alignment given the sequence pair:

$$L(Aln|A,B) \propto f(Aln|A,B) = \frac{f(Aln,A,B)}{f(A,B)} \propto f(Aln,A,B) \tag{4.6}$$

To calculate Equation 4.6 completely for two sequences related by a common ancestor, one would have to consider all sequences that could be the most recent common ancestor of A and B and all possible branch lengths between this ancestor and A and B. However, our simulations assumed that the tree relating A and B was unrooted, and thus A was considered to be descended from B, eliminating the need to consider the set of all possible progenitors for both sequences. We calculate Equation 4.6 based on the evolutionary distance or branch length t between sequences A and B:

$$f(Aln,A,B) = \int_t f(Aln,A,B,t)\,dt \tag{4.7}$$

It is possible to derive Equation 4.7 from an evolutionary process. Specifically the probability that B gave rise to A over evolutionary distance t with indels to produce alignment Aln.

$$f(Aln,A,B,t) = f(B \to A,t,Aln) = f(A|Aln,B,t)\,f(Aln|B,t)\,f(t)\,f(B) \tag{4.8}$$

where $f(t) = \exp(-t/\theta)/\theta$ is the density of branch lengths between A and B and $f(B) = 4^{-L_b}$ is the probability for ancestral sequence B of length L_b. If L_a, M, and R are respectively the length of sequence A, the number of matches in the alignment, and the number of replacements, then under the Jukes-Cantor model,

$$f(A|Aln,t,B) = \frac{\left(1+3e^{-4t/3}\right)^M \left(1-e^{-4t/3}\right)^R}{4^{L_a}} \tag{4.9}$$

The probability that an indel occurs at any position is $1-e^{-\lambda t}$, and, if we ignore the issue of overlapping indels,

$$f\left(Aln\,|B,t\right)=\left(e^{-\lambda t}\right)^{L_b-N}\left(1-e^{-\lambda t}\right)^{N}\prod_{g=1}^{N}f\left(k_g\right) \tag{4.10}$$

where N is the number of indels in the alignment, $f\left(k_g\right)=k_g^{-z}/\zeta\left(z\right)$ is the probability that an indel has a length of size k_g, and λ is the instantaneous rate of indel formation per unit branch length. Putting this all together,

$$f\left(Aln,A,B,t\right)=\frac{e^{-\lambda t L_b}}{4^{L_a+L_b}}\left(1+3e^{-4t/3}\right)^{M}\left(1-e^{-4t/3}\right)^{R} \times\left(e^{-\lambda t}\right)^{-N}\left(1-e^{-\lambda t}\right)^{N}\frac{e^{-t/\theta}}{\theta}\prod_{g=1}^{N}f\left(k_g\right) \tag{4.11}$$

For simplicity we will not integrate Equation 4.7 to find $f\left(Aln,A,B\right)$. Instead, we will approximate it based on the mean value of t:

$$f\left(Aln,A,B\right)\approx f\left(Aln,A,B|t=\bar{t}\right)=f\left(Aln,A,B,t=\bar{t}\right)/f\left(t=\bar{t}\right)$$

Upon removing factors that are constant for sequences A and B we get the likelihood for the alignment Aln given sequence pair A and B:

$$L\left(Aln\,|A,B\right)=\left(1+3e^{-4\theta/3}\right)^{M}\left(1-e^{-4\theta/3}\right)^{R}\prod_{g=1}^{N}\frac{e^{\lambda\theta}-1}{\zeta\left(z\right)}k_g^{-z} \tag{4.12}$$

The log-likelihood is therefore

$$\ln L\left(Aln\,|A,B\right)=M\ln\left(1+3e^{-4\theta/3}\right)+R\ln\left(1-e^{-4\theta/3}\right) +\sum_{g=1}^{N}\left[\ln\left(e^{\lambda\theta}-1\right)-\ln\zeta\left(z\right)-z\ln k_g\right] \tag{4.13}$$

APPENDIX 4.B GAP COSTS

As developed in Smith et al. (1981) and extended below, a maximum likelihood search can be converted to a minimum cost search. Based on a statistical model, the scores of "matches" of type i, α_i, and the penalties of gaps of length k, w_k, can be used to calculate the alignment with maximum log-likelihood:

$$l = \max\left\{\sum \alpha_i \eta_i - \sum w_k \Delta_k\right\} \tag{4.14}$$

where η_i is the number of residue matches of type i and Δ_k is the number of gaps of length k. A minimum cost analog of Equation 4.14 is

$$d = \min\left\{\sum \beta_i \eta_i + \sum G(k)\,\Delta_k\right\} \tag{4.15}$$

To begin constructing the minimum cost analog, let $\beta_i = (x - \alpha_i)/y$ be the cost of a match of type i, therefore

$$\begin{aligned} -l = \min\left\{-\sum \alpha_i \eta_i + \sum w_k \Delta_k\right\} &= \min\left\{\sum (y\beta_i - x)\,\eta_i + \sum w_k \Delta_k\right\} \\ &= y \min\left\{\sum \beta_i \eta_i - \frac{x}{y}\sum \eta_i + \sum \frac{w_k}{y}\Delta_k\right\} \end{aligned} \tag{4.16}$$

The lengths of the sequences being aligned, n and m, can be related to the alignment itself via the equation $n + m = 2\sum \eta_i + \sum k\Delta_k$. Using this relationship, Equation 4.16 can be expressed as

$$
\begin{aligned}
-l &= y\min\left\{\sum\beta_i\eta_i - \frac{x}{2y}\left(n+m-\sum k\Delta_k\right)+\sum\frac{w_k}{y}\Delta_k\right\} \\
&= -\frac{x(n+m)}{2}+y\min\left\{\sum\beta_i\eta_i+\frac{x}{2y}\sum k\Delta_k+\sum\frac{w_k}{y}\Delta_k\right\} \\
&= -\frac{x(n+m)}{2}+y\min\left\{\sum\beta_i\eta_i+\sum\left(\frac{xk}{2y}+\frac{w_k}{y}\right)\Delta_k\right\} \\
&= -\frac{x(n+m)}{2}+y\min\left\{\sum\beta_i\eta_i+\sum G(k)\Delta_k\right\} \qquad (4.17)
\end{aligned}
$$

From this it can be clearly seen that $d=\min\left\{\sum\beta_i\eta_i+\sum G(k)\Delta_k\right\}$ maximizes the likelihood of the alignment, where $G(k)=\left(xk/2+w_k\right)/y$ is the cost of a gap of length k. Applying this method to Equation 4.13 such that the cost of a match is 0 and the cost of a mismatch is 1 produces the following equation for a gap cost:

$$
G(k)=\frac{\ln\zeta(z)-\ln\left(e^{\lambda\theta}-1\right)+\ln\left(1+3e^{-4\theta/3}\right)k/2+z\ln k}{\ln\left(1+3e^{-4\theta/3}\right)-\ln\left(1-e^{-4\theta/3}\right)} \qquad (4.18)
$$

CHAPTER 5

ANTAGONISM BETWEEN LOCAL DISPERSAL AND SELF-INCOMPATIBILITY SYSTEMS IN A CONTINUOUS PLANT POPULATION[1]

[1]Cartwright, R.A. (submitted) *The American Naturalist*.

5.1 Introduction

Both local dispersal and self-compatibility systems are common in plant species. Local dispersal of seeds creates population structures in plant populations such that relatives are likely to be found near one another. Furthermore, local dispersal of pollen coupled with local dispersal of seeds creates pollen pools containing relatives. Therefore, local dispersal and can facilitate inbreeding and geographic differentiation within a population. Furthermore, many plant taxa have also evolved self-incompatibility systems that prevent selfing. Because these systems are genetically based, relatives are likely to be incompatible. Therefore, self-incompatibility systems, in addition to requiring outcrossing, also promote outbreeding. Clearly, if a plant population has both local dispersal and self-incompatibility systems then an antagonism can exist between their evolutionary effects. This study seeks to investigate this antagonism on on effective population sizes, conditional inbreeding coefficients, fine-scale genetic structure, and neighborhood sizes via a computational model.

5.1.1 Inbreeding

Inbreeding occurs when related individuals mate more frequently than would be expected if the population mated randomly. The extreme form of inbreeding is selfing, where individuals fertilize themselves. A common mating strategy among plants is mixed mating, where individuals can both self and outcross. Inbreeding does not directly affect allele frequencies, rather it affects genotype frequencies by increasing the number of homozygotes. This exposes the phenotypes of recessive alleles, which, if deleterious, can decrease the fitness of individuals, i.e. inbreeding depression.

Inbreeding in a population is measured from the probability that an individual is "identical-by-descent" at a particular locus, also known as "autozygous." An individual is identical-by-descent at a locus if its two copies are descended from the same ancestral copy without any mutations to change the gene (Malécot, 1975). Clearly inbreeding increases the chance that an individual will inherit two copies of a gene from a single ancestral copy. Besides inbreeding, other evolutionary forces like genetic drift or selection can increase the probability that an individual is autozygous.

5.1.2 Isolation-by-Distance and Neighborhood Sizes

Although a species may inhabit a geographically contiguous region with no physical barriers to gene flow, the species may not be panmictic. Physical distances separating two individuals may prevent them from mating freely, and instead individuals are more likely to mate with nearby individuals than individuals that are far away. Wright (1943) referred to this as "isolation-by-distance", and Wright (1946) developed the concept of "neighborhood size" to compare how different mating systems affect isolation-by-distance. For example, if gametes disperse independently and identically on the north-south and east-west axes with a normal distribution of mean 0 and variance σ^2, then the eucledian parent-offspring distance, r, will have a Rayleigh distribution:

$$f(r) = \frac{r}{\sigma^2} e^{\frac{r^2}{2\sigma^2}}$$

Wright (1946) defined neighborhood size, N_b (N in his paper), as a number of individuals having the same magnitude as the inverse of the probability of self-fertilization. Assuming that parent-offspring distance obeys the above Rayleigh distribution, he derived the neighborhood size of a hermaphroditic population as

$$N_b = 4\pi\sigma^2 d \tag{5.1}$$

where d is the density of individuals per unit area. Although σ^2 is the variance of dispersal along the axes, it is one half the mean squared euclidean distance.

Wright's concept of neighborhood size does not adequately extend to other mating systems and dispersal types that also produce isolation-by-distance because self-fertilization is not consistently linked with gene dispersal. However, instead of defining neighborhood sizes based on self-fertilization, it is possible to use Wright's result, Equation 5.1, to define neighborhood sizes based on dispersal (Fenster et al., 2003; Vekemans and Hardy, 2004). The advantage of this is that σ^2 is an important component of how identity-by-descent is related to isolation-by-distance (Rousset, 2000).

Let Q_r be the probability that genes separated by a geographic distance of r are identical-by-descent, and Q_w be the probability within a single individual. Furthermore, the quantity $a_r \equiv (Q_w - Q_r)/(1 - Q_w)$ will depend in the level of isolation-by-distance. Rousset (2000), drawing on Rousset (1997) and Sawyer (1977), showed that for a wide range of dispersal functions

$$a_r \approx \frac{\ln r}{4\pi\sigma^2 d} + constant \tag{5.2}$$

One should not confuse the $4\pi\sigma^2 d$ in Equation 5.2 with Wright's N_b, which happens to equal $4\pi\sigma^2 d$ under Gaussian dispersal. However, as Fenster et al. (2003) and Vekemans and Hardy (2004) have done, one can redefine $N_b \equiv 4\pi\sigma^2 d$, which can be calculated from a correlegram of kinship coefficients, F_{ij}, using Equation 5.2. (The equation for F_{ij} is defined later.) They have defined the Sp statistic as

$$Sp \equiv \frac{-b_{ln}}{1 - F_1}$$

where F_1 is the average F_{ij} for distance class 1 and b_{ln} is the slope of the linear regression of F_{ij} to $\ln(r)$. Under isolation-by-distance $1/Sp = 4\pi\sigma^2 d = N_b$, allowing one to estimate neighborhood sizes from correlegrams (See Fenster et al., 2003; Vekemans and Hardy, 2004, for more information.).

Because individuals near one another are expected to be more closely related than individuals farther apart, isolation-by-distance generates fine-scaled genetic structure where the closer individuals are geographically, the more likely they have identical alleles. Therefore, alleles can often show a patchy distribution with respect to geography (Epperson, 1990; Rohlf and Schnell, 1971; Sokal and Wartenberg, 1983; Turner et al., 1982).

5.1.3 Self-Incompatibility Systems

Self-fertilization is the most extreme form of inbreeding. Many taxa of angiosperms prevent self-fertilization via self-incompatibility systems (SI) that are controlled by single genes. Self-

incompatibility systems have been studied in Brassicaceae (e.g. cabbage, broccoli, *Arabidopsis*), Papaveraceae (e.g. poppy), Solanaceae (e.g. nightshade, potato, tobacco), Scrophulariaceae (e.g. snapdragon), and Rosaceae (e.g. rose, apple, almond) (Igic and Kohn, 2001; Nasrallah, 2005; Takayama and Isogai, 2005). These SI systems are controlled by a single highly polymorphic Mendelian locus, the S locus, which contains several tightly linked genes. The function of the S locus is to prevent pollen from fertilizing flowers of genetically similar plants. The primary result is that self-fertilization is impossible, and the secondary result is that matings between relatives are reduced. The S locus evolves under classical frequency-dependent sexual selection (Wright, 1939), resulting in high fitness for rare and novel alleles. Plants with rare alleles can pollinate more plants than can plants with common alleles.

There are two main types of SI systems: gametophytic and sporophytic. There is a third type, late-acting, that occurs after seeds are formed, but it has been studied less (Gibbs and Bianchi, 1999; Lipow and Wyatt, 2000; Seavey and Bawa, 1986). In gametophytic self-incompatibility (GSI), the mating-type of the pollen is determined by the pollen haplotype. Styles reject any pollen bearing either one of their alleles. For example, a plant with genotype S_1S_2 produces S_1 and S_2 pollen. S_1 pollen can fertilize S_2S_3 plants but not S_1S_2 or S_1S_3 plants. Likewise, S_2 pollen can fertilize S_1S_3 plants but not S_1S_2 or S_2S_3. Furthermore, if we assume panmictic dispersal, an S_aS_b plant will accept $1-p_a-p_b$ proportion of the pollen pool, where p_a and p_b are the allele frequencies of S_a and S_b respectively.

In sporophytic self-incompatibility (SSI), the mating-type of the pollen is determined by the pollen-donor's genotype. Under it, stigmas will only accept pollen produced by plants that share none of the stigmas's alleles. (Technically, this is codominant SSI.) For example, a plant with genotype S_1S_2 produces $S_{1,2}$ pollen regardless of whether the pollen grain carries the S_1 or S_2 allele. $S_{1,2}$ pollen can fertilize S_3S_4 plants but not S_1S_2, S_1S_3, S_1S_4, S_2S_3, or S_2S_4. Furthermore, if we assume panmictic dispersal and well-mixing of alleles, a plant with genotype S_aS_b will accept $1-2p_a-2p_b+p_ap_b$ proportion of the pollen pool.

The genes of the S locus specify both the male and female mating-type determinants. Because these determinants must work in unison, they are tightly linked at the S locus. This tight linkage results from both close physical association on the chromosome and reduced recombination rate in the area (Kamau and Charlesworth, 2005). Because recombination rates are reduced and S loci experience strong balancing selection, genes near the S loci will exhibit more diversity and longer coalescent times than they would otherwise (Awadalla and Charlesworth, 1999; Charlesworth, 2006; Charlesworth et al., 2006; Hagenblad et al., 2006; Kamau and Charlesworth, 2005; Schierup et al., 2000).

Self-incompatibility has evolved independently multiple times, which reflects a common adaptive advantage to avoid selfing. Three completely different self-incompatibility mechanisms have been characterized so far. One belonging to the Brassicaceae, another belonging to the Papaveraceae, and a third belonging to the Solanaceae, Scrophulariaceae, and Rosaceae (Igic and Kohn, 2001). These are respectively referred to as "brassiciceae-type", "papaveraceae-type" and "solanaceae-type" systems, or "b-type", "p-type", and "s-type" for short. Since the s-type system is found in both rosids and asterids (Igic and Kohn, 2001), the system was most likely in the ancestor of those groups, which lived over 90 million years ago (Nasrallah, 2005). Furthermore, since brassicacids are rosids, they lost the s-type system and evolved the b-type system.

The brassiciceae-type system is the only sporophytic system that has been characterized genetically. In this system the S locus consists of three genes: S-locus cystein-rich protein (SCR or SP11), S-locus receptor kinase (SRK), and S-locus glycoprotein (SLG). The male mating type is determined by SCR, which is secreted by anthers and accumulates in the pollen coat. SRK determines the female mating type and is a membrane-bound protein that is found in stigma papilla cells. SRK binds SCR which sets off a signaling cascade resulting in pollen rejection. The SLG proteins are accessory proteins that localize to the cell walls of stigma papilla cells and function to enhance the SI phenotype of some S locus haplotypes (Igic and Kohn, 2001).

The solanaceae-type system is an ancient, gametophytic system found first in the Solanaceae family. In this system the S locus consists of two genes: S-RNase, and S locus f-box (SLF or SFB).

S-RNase is expressed in the style and determines the female mating-type. It enters pollen tubes and degrades RNA of self-pollen, stopping pollen tube growth. SLF is expressed in pollen tubes and is hypothesized to act as an inhibitor of non-self-S-RNases, although the nature of the interaction is still unclear (Igic and Kohn, 2001).

Compared to the other two systems, the gametophytic, papaveraceae-type system has only been recently studied molecularly. The male determinant is still unknown and the female determinant goes by the generic name "S protein". S protein is secreted by the stigma and appears to function as a ligand for a receptor on the pollen grain (the hypothesized male determinant). Binding of the ligand to the receptor causes an influx of Ca^{2+} ions into the pollen tube. This influx inhibits pollen tube growth and leads to pollen tube death (Igic and Kohn, 2001).

5.1.4 Antagonism

Self-incompatibility systems promote outbreeding while local dispersal can facilitate inbreeding; therefore, an antagonism exists between the two processes. Furthermore, this antagonism will manifest differently in markers linked to the S locus than in markers that are unlinked to the S locus. One important motivation for this study is to investigate how unlinked loci are influenced by S loci. If unlinked loci are strongly influenced by S loci, then studies of plant populations, including computational models, cannot ignore S loci.

Several studies have looked at the effect of population subdivision on allelic diversity of S loci (Muirhead, 2001; Neuhauser, 1998; Schierup, 1998; Schierup et al., 2000; Wright, 1939). Furthermore, two studies have looked at the effect of isolation-by-distance on self-incompatibility systems (Brooks et al., 1996; Neuhauser, 1998). However, none have studied the effect of isolation-by-distance and self-incompatibility systems on linked loci.

5.2 Model

5.2.1 Geography

The population is a rectangular lattice of width X and height Y. Each cell in the lattice is a hermaphroditic, diploid individual.

5.2.2 Genetics

Each individual has $2N = 2$ chromosomes, i.e. diploid organisms with 1 chromosome per haploid set. The chromosomes contain contained multiple markers and an S locus, ordered such that the self-incompatibility locus, S, is to the right of all the markers. One crossover occurs each time a gamete is produced. Each marker locus had a recombination rate with locus S of 2^{-n}, where n is the marker number. The left-most locus, 1, has a recombination rate of 50% with the S locus. The second-to-left-most locus, 2, has a recombination rate of 25% with the S locus, etc. (Following this numbering scheme, the S locus is actually the infinite locus.) On the first generation each locus has $2N$ alleles, and thus every individual is heterozygous at each locus and no two individuals share an allele at any locus. The advantage of this formulation is that all homozygotes that occur during simulations are identical-by-descent. This is atypical but should not bias the results because most of the allelic diversity is lost in the first hundred generations.

5.2.3 Compatibility

The model implements four different types of mating-systems: no self-incompatibility (NSI), physical self-incompatibility (PSI), gametophytic self-incompatibility (GSI), and sporophytic self-incompatibility (SSI). Under NSI all individuals are compatible with themselves and all other individuals. Under PSI individuals are obligate out-crossers but there are no genetic mating-types. Selfing is prevented but fullsib, halfsib, cousin, etc. inbreeding is allowed. Under GSI pollen can only fertilize plants which do not contain its S-locus haplotype. Under SSI plants can only pollinate plants which do not share either of its S-locus genes.

Castric and Vekemans (2004) review some situations that would complicate the above model of compatibility. There could be dominance relationships among the S alleles in sporophytic self-incompatibility (Bateman, 1952; Schierup et al., 1997; Uyenoyama, 2000). S alleles may not be selectively neutral because they could be linked to recessive, deleterious mutations (Uyenoyama, 1997, 2003). S alleles can violate Mendel's laws by segregating unevenly (Bechsgaard et al., 2004).

Let m and n represent the position of the pollen donor on the x- and y-axes, and let k and l similarly represent the seed parent. If we define $C(k,l,m,n)$ as the proportion of pollen from individual mn that is compatible with individual kl, then we can calculate it for each of the four mating-systems. For NSI, $C(k,l,m,n) = 1$ for all kl and mn, and for PSI, $C(k,l,k,l) = 0$ and $C(k,l,m,n) = 1$ for $kl \neq mn$. For GSI, $C(k,l,m,n) = 0$ if kl and mn have the same SI genotype, $C(k,l,m,n) = 1/2$ if kl and mn share one SI allele, and $C(k,l,m,n) = 1$ if kl and mn share no SI alleles. For SSI, $C(k,l,m,n) = 0$ if kl and mn share at least one SI allele and $C(k,l,m,n) = 1$ otherwise.

5.2.4 DISPERSAL

There are two different types of dispersal: male gametes (pollen) and embryos (seeds). In this model, both forms of dispersal have a uniform radius and an exponential distance. Exponential dispersal is leptokurtic, which is common for plant dispersal (Kot et al., 1996). If σ_d^2 is the variance of dispersal distance, then the probability density of dispersal is

$$f_{r\theta}(r,\theta|\sigma_d) = \frac{1}{2\pi}\frac{e^{-r/\sigma_d}}{\sigma_d}$$

where θ is the angle of dispersal from the positive x-axis and r is the radius of dispersal. The model allows for seeds and pollen to have different parameters of dispersal, i.e. $\sigma_d = \sigma_s$ or $\sigma_d = \sigma_p$. In Cartesian coordinates this becomes probability density

$$f_{xy}(x,y|\sigma_d) = \frac{1}{2\pi}\frac{e^{-\sqrt{x^2+y^2}/\sigma_d}}{\sqrt{x^2+y^2}\sigma_d}$$

Since the population is structured on a lattice, dispersal to an individual is calculated by integrating dispersal to its cell. The probability density of dispersal to an individual with offset ij is

$$f_{ij}(i,j|\sigma_d) = \int_{i-1/2}^{i+1/2} \int_{j-1/2}^{j+1/2} f_{xy}(x,y|\sigma_d)\ dy\ dx$$

Seeds and pollen are assumed to be infinite, which allows for dispersal to be simulated backwards. For each cell, a mother is drawn from the previous generation based on seed dispersal centered on the cell and repeated until a valid mother is found. (Invalid mothers are cells that are off the lattice.) Once a mother is found, a pollen grain is drawn from the previous generation based on pollen dispersal centered on the mother and Mendel's laws. If the pollen grain comes from an invalid (off-lattice) father or if it is incompatible with the mother, the parent pair is rejected and the process repeats until a valid, compatible pair is chosen. The chosen pollen grain is combined with a random gamete from the mother to form the daughter cell.

Rejecting a mother if the pollen is invalid or incompatible penalizes females for pollen availability and pollen compatibility, two processes that can happen in nature. The probability that for cell ij individual kl is the mother and individual mn is the father is

$$P_{mp}(k,l,m,n|i,j) = \frac{1}{K(i,j)} C(k,l,m,n)\, f_{ij}(k-i,l-j|\sigma_s)\, f_{ij}(m-k,n-l|\sigma_p) \tag{5.3}$$

where $K(i,j)$ is the correction factor

$$K(i,j) = \sum_k \sum_l \sum_m \sum_n C(k,l,m,n)\, f_{ij}(k-i,l-j|\sigma_s)\, f_{ij}(m-k,n-l|\sigma_p)$$

When self-fertilization is allowed (under NSI), the probability of self-fertilization is

$$f_{ij}(0,0) \approx 1 - e^{-1/\sqrt{\pi}\sigma_p} \tag{5.4}$$

i.e. pollen comes from the circle with an area of 1 centered on the seed parent.

By taking the marginal distributions of Equation 5.3, the maternal and paternal distributions can be calculated for cell ij:

$$P_m(k,l|i,j) = \sum_m \sum_n P_{mp}(k,l,m,n|i,j) \tag{5.5}$$

$$P_p(m,n|i,j) = \sum_k \sum_l P_{mp}(k,l,m,n|i,j) \tag{5.6}$$

And finally, the probability that kl is a mother is

$$P_m(k,l) = \frac{1}{N} \sum_i \sum_j P_m(k,l|i,j)$$

and the probability that mn is a father is

$$P_p(m,n) = \frac{1}{N} \sum_i \sum_j P_p(m,n|i,j)$$

where $N = X \times Y$ is the size of the population.

5.2.5 Simulation

A simulation based on this model was written in C++. Thirty-six hundred simulations were run, one hundred for each of thirty-six different parameter sets. These parameter sets consisted of four different SI systems—NSI, PSI, GSI, or SSI—by three different seed dispersal levels—$\sigma_s = 1$, $\sigma_s = 2$, or panmictic—by three different pollen dispersal levels—$\sigma_p = 2$, $\sigma_p = 4$, or panmictic. The population size was $50 \times 50 = 2500$ individuals. The S locus and 15 marker loci were tracked for 2500 generations. For each locus in each generation, the simulations recorded expected heterozygosity (H_T), the actual heterozygosity (H_I), the conditional inbreeding coefficient (F), and the number of alleles (K). The genetic map of the population was recorded on the last generation and used to measure fine-scale genetic structure.

5.3 METHODS

Effective population size was calculated for each simulation using variance in fecundity, V_K (Wright, 1938):

$$N_e = \frac{4N-2}{V_K+2}$$

Under this definition, the range of N_e is 1 to $2N-1$, where N is the census size. Since fecundity is the number of offspring an individual has, populations with larger variations in fecundity will have larger levels of identity-by-descent and lower effective population sizes. Because variance in fecundity is essentially constant during a simulation (results not shown), N_e was calculated from the V_K of the final generation. Since each parameter set was represented by 100 simulations, the distribution of the N_e of each parameter set could be established. Using R (R Development Core Team, 2006), a t-test with unequal variances was used to compare the N_e distributions of each pair of parameter sets, and the resulting p-values were corrected for multiple tests using Holm correction, which is more powerful than Bonferroni correction (Holm, 1979; Shaffer, 1995; Wright, 1992).

Inbreeding was measured using the conditional inbreeding coefficient (Hardy and Vekemans, 1999; Malécot, 1975; Wright, 1965),

$$F = \frac{f-\bar{\theta}}{1-\bar{\theta}} = \frac{H_T - H_I}{H_T}$$

where f is the probability of identity-by-descent of genes in individuals and $\bar{\theta}$ is the probability of identity-by-descent for two genes chosen at random in the population. Since in this model, every homozygote is autozygous, F can be calculated from the expected heterozygosity in the population, H_T, and the actual heterozygosity in the population, H_I.

To investigate fine scale genetic structure, Nason's kinship coefficient, F_{ij} (Loiselle et al., 1995), was calculated for all 3.1 million pairs of individuals in each simulation at generation 2500 using SPAGeDi (Hardy and Vekemans, 2002). F_{ij} for any pair of individuals i and j is defined as

$$F_{ij} = \frac{\sum_l \left[\sum_a (p_{ila} - p_{la})\left(p_{jla} - p_{la}\right) + \sum_a p_{la}(1 - p_{la}) / (n_l - 1)\right]}{\sum_l \sum_a p_{la}(1 - p_{la})}$$

where p_{ila} is the frequency of allele a at locus l in individual i, p_{la} is the frequency of allele a at locus l in the entire sample, and n_l is the number of distinct copies of locus l in the sample. F_{ij} can be calculated for each locus separately or for the entire genome. In this study, loci were expected to show differences in fine-scale genetic structure, and therefore, F_{ij} was calculated for each locus separately. Using SPAGeDi, pairs of individuals were divided into ten, roughly equal-sized distance classes based on the distance separating the individuals. The average F_{ij} was calculated from the pairs in each distance class.

Neighborhood size, N_b was calculated for each simulation from the F_{ij} correlegrams using the S_p statistic:

$$N_b = \frac{1}{S_p} = \frac{F_1 - 1}{b_{ln}}$$

where F_1 is the average F_{ij} for distance class 1 and b_{ln} is the slope of the linear regression of F_{ij} to $\ln(distance)$.

5.4 Results

5.4.1 Effective Population Sizes

Table 5.1 lists the average N_e for each parameter set in decreasing order along with their 95% confidence intervals. The confidence intervals have been adjusted for multiple tests ($n = 36$) using Bonferroni correction. Table 5.1 also lists the average variance in total, male, and female fecudity as well as the average covariance of male and female fecudity. Figure 5.1 shows whether two parameter sets have significantly different average effective population sizes at the $\alpha = 0.05$ level with Holm correction for multiple tests (Holm, 1979; Shaffer, 1995; Wright, 1992).

Parameter sets can be classified into different groups based on their associated effective population sizes. The two highest groups, A and B, consist of populations that are self-incompatible and

Table 5.1: Average Effective Population Sizes. For σ_s and σ_p, "P" signifies panmictic dispersal. The 95% confidence intervals of average N_e were estimated using a Student's t distribution and have been adjusted for multiple (36) tests using Bonferroni correction.

Group	SI	σ_s	σ_p	Ne	Average Variance of Fecundity			
					Total	Female	Male	2Cov
A	PSI	1	2	2692±10.98	1.71	0.79	0.98	−0.06
	GSI	1	2	2686±12.55	1.72	0.79	0.99	−0.06
	PSI	1	4	2669±11.22	1.75	0.79	1.00	−0.04
	GSI	1	4	2661±12.16	1.76	0.79	1.00	−0.03
	SSI	1	4	2660±10.81	1.76	0.79	1.00	−0.03
	SSI	1	2	2655±11.07	1.77	0.79	0.99	−0.02
	SSI	1	P	2646±11.31	1.78	0.78	0.99	0.00
	PSI	1	P	2644±12.38	1.78	0.78	1.00	−0.00
	NSI	1	P	2641±11.26	1.79	0.78	1.00	0.00
	GSI	1	P	2641±11.65	1.79	0.79	1.00	0.00
B	PSI	2	2	2571±12.79	1.89	0.93	0.99	−0.03
	PSI	2	P	2566±12.15	1.90	0.91	0.99	−0.01
	GSI	2	2	2564±11.24	1.90	0.93	1.00	−0.02
	NSI	2	P	2556±10.55	1.91	0.91	1.00	0.00
	PSI	2	4	2555±12.07	1.91	0.92	1.00	−0.01
	GSI	2	P	2554±12.45	1.91	0.91	1.00	−0.00
	SSI	2	P	2550± 9.93	1.92	0.92	1.00	0.00
	SSI	2	2	2549±10.80	1.92	0.93	1.00	−0.01
	SSI	2	4	2547±12.15	1.93	0.93	1.00	−0.00
	GSI	2	4	2546±10.66	1.93	0.93	1.00	−0.00
C	NSI	1	4	2523±11.73	1.96	0.79	0.98	0.20
D	GSI	P	P	2508±11.34	1.99	1.00	1.00	−0.01
	PSI	P	P	2502±10.31	2.00	1.00	0.99	−0.00
	NSI	P	P	2496±11.70	2.00	1.00	1.00	0.01
	SSI	P	P	2494±10.89	2.01	1.00	1.00	0.01
E	GSI	P	2	2465±12.62	2.06	1.01	1.01	0.03
	PSI	P	2	2455±12.31	2.07	1.02	1.02	0.03
	SSI	P	2	2448±12.69	2.08	1.02	1.02	0.05
	SSI	P	4	2446±11.80	2.09	1.02	1.02	0.04
	GSI	P	4	2443±10.49	2.09	1.02	1.02	0.04
	PSI	P	4	2440±11.80	2.10	1.02	1.03	0.05
F	NSI	1	2	2434±13.57	2.11	0.78	0.96	0.36
	NSI	2	4	2399±11.98	2.17	0.92	1.00	0.25
	NSI	2	2	2302±11.24	2.34	0.92	0.99	0.44
	NSI	P	4	2296±12.39	2.36	1.02	1.01	0.33
	NSI	P	2	2199±12.40	2.55	1.01	1.01	0.53

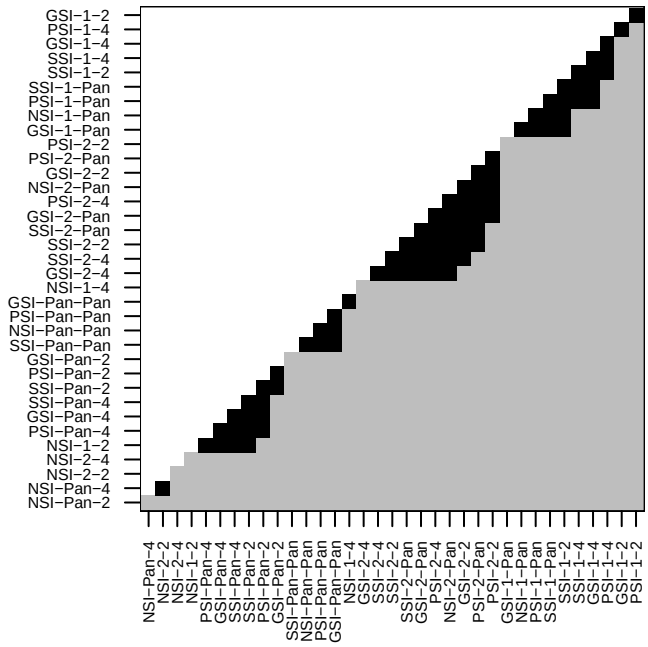


Figure 5.1: Pairwise Comparisons of N_e. Gray squares are runs with significantly different ($\alpha = 0.05$) average N_e.

have local seed dispersal, with $\sigma_s = 1$ for Group A and $\sigma_s = 2$ for Group B. Group C consists of a a single parameter set: NSI, $\sigma_p = 1$, and $\sigma_s = 4$. Three groups, A-C, are characterized by effective population sizes that are above the census size, $N = 2500$. Group D consists of the four parameter sets in which pollen and seed dispersed panmicticly and is the only group that has an effective population size consistent with the census size. Group E is characterized by populations with panmictic seed dispersal, local pollen dispersal, and no selfing. Finally, Group F consists of populations with non-trivial rates of selfing. Groups E and F are characterized by effective population sizes that are lower than the census size.

5.4.2 INBREEDING

Figure 5.2 shows the average conditional inbreeding coefficients for each locus organized by the mating system and dispersal regime. In Figure 5.3(a), it is clear that under NSI six different dis-

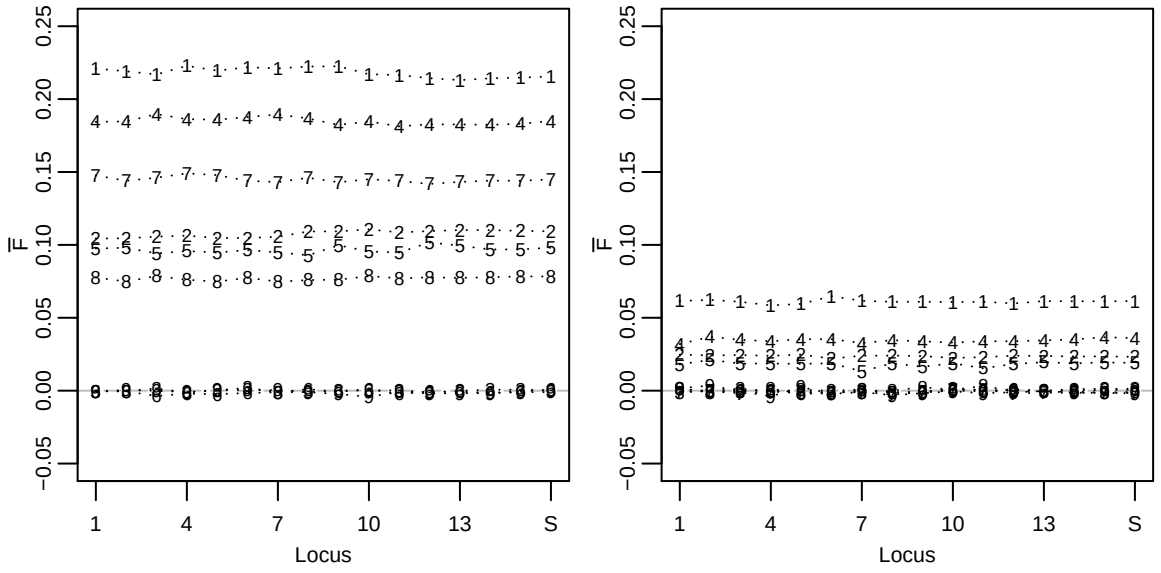
persal regimes show positive conditional inbreeding coefficients. From highest to lowest these are σ_s–σ_p = 1–2 (line 1), 2–2 (line 4), Pan–2 (line 7), 1–4 (line 2), 2–4 (line 5), and Pan–4 (line 8). In Figure 5.3(b), only four different dispersal regimes produce positive inbreeding coefficients under PSI: 1–2, 1–4, 2–2, and 2–4. Furthermore, the inbreeding levels for PSI are much lower that for NSI. GSI (Figure 5.3(c)) and SSI (Figure 5.3(d)) begin similar to PSI, but the inbreeding coefficients decrease as loci get closer to the S locus, switching from inbreeding to outbreeding between markers 5 and 8 or 3 to 0.4 centimorgans from the S locus.

Figure 5.3 looks at conditional inbreeding coefficients differently. In this figure, the levels of inbreeding are compared for each mating system on each dispersal regime. Figure 5.4(a) shows the results for the unlinked locus, and Figure 5.4(b) shows the results for the S locus. The levels of inbreeding for the unlinked locus are the same for PSI, GSI, and SSI. However, for the S locus, PSI is significantly different than GSI and SSI. Furthermore, for both loci, the inbreeding coefficients of GSI and SSI are not significantly different.

5.4.3 Fine-Scale Genetic Structure

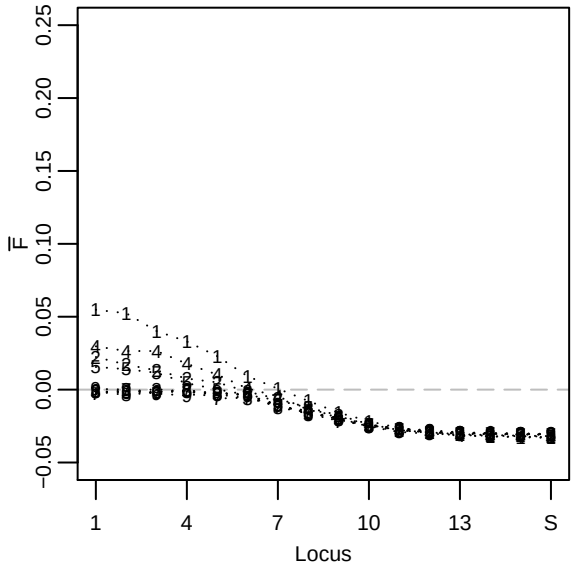
The fine-scale genetics structures of the unlinked and S loci are presented as correlegrams in Figure 5.4. Each locus is split into four subfigures representing the local dispersal regimes and containing correlegrams for four different mating systems. These correlegrams show how kinship (F_{ij}) decreases on average as the distance between individuals increases. Populations with more structure will have higher average kinship for the first distance class and decrease much faster than populations with less structure. Regardless of the locus or dispersal regime, NSI generates more structure than the self-incompatible mating systems. Furthermore, there is no significant difference between GSI and SSI in any of the correlegrams. For the unlinked locus, PSI, GSI, and SSI are not significantly different, whereas for the S locus, PSI is significantly different than GSI and SSI. Structure is greatest for 1–2 dispersal, followed by 2–2, 1–4, and 2–4.

Correlegrams for all loci, dispersal regimes, and mating systems can be found in Online Appendix 5.A. This appendix contains both raw correlegrams and correlegrams corrected for

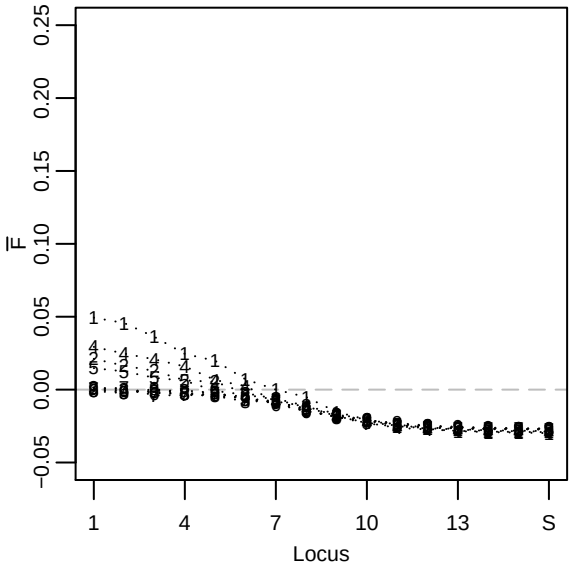


(a) NSI

(b) PSI

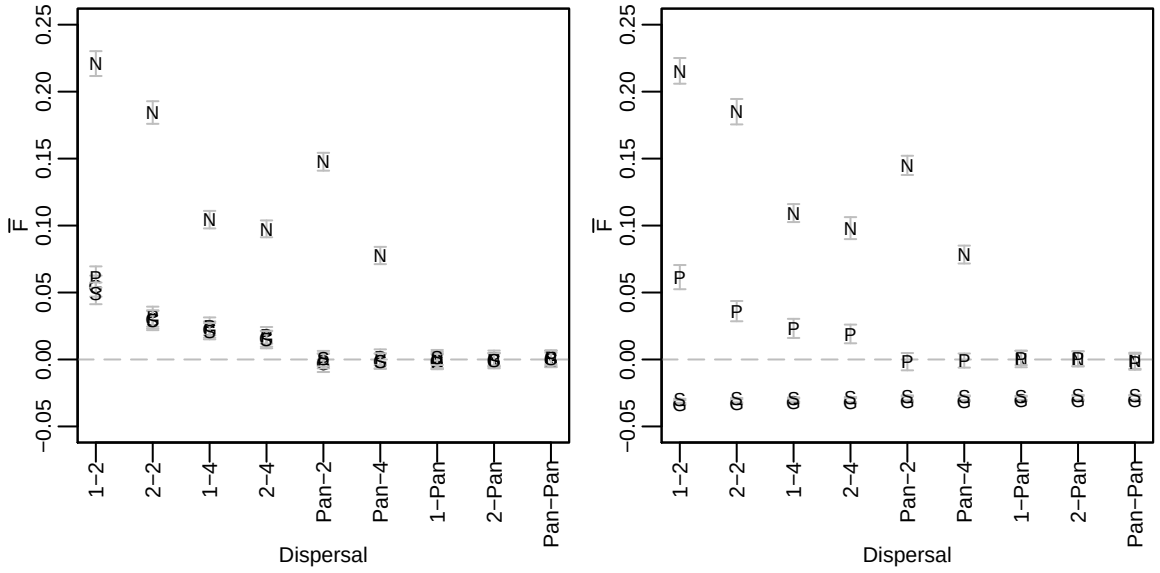


(c) GSI



(d) SSI

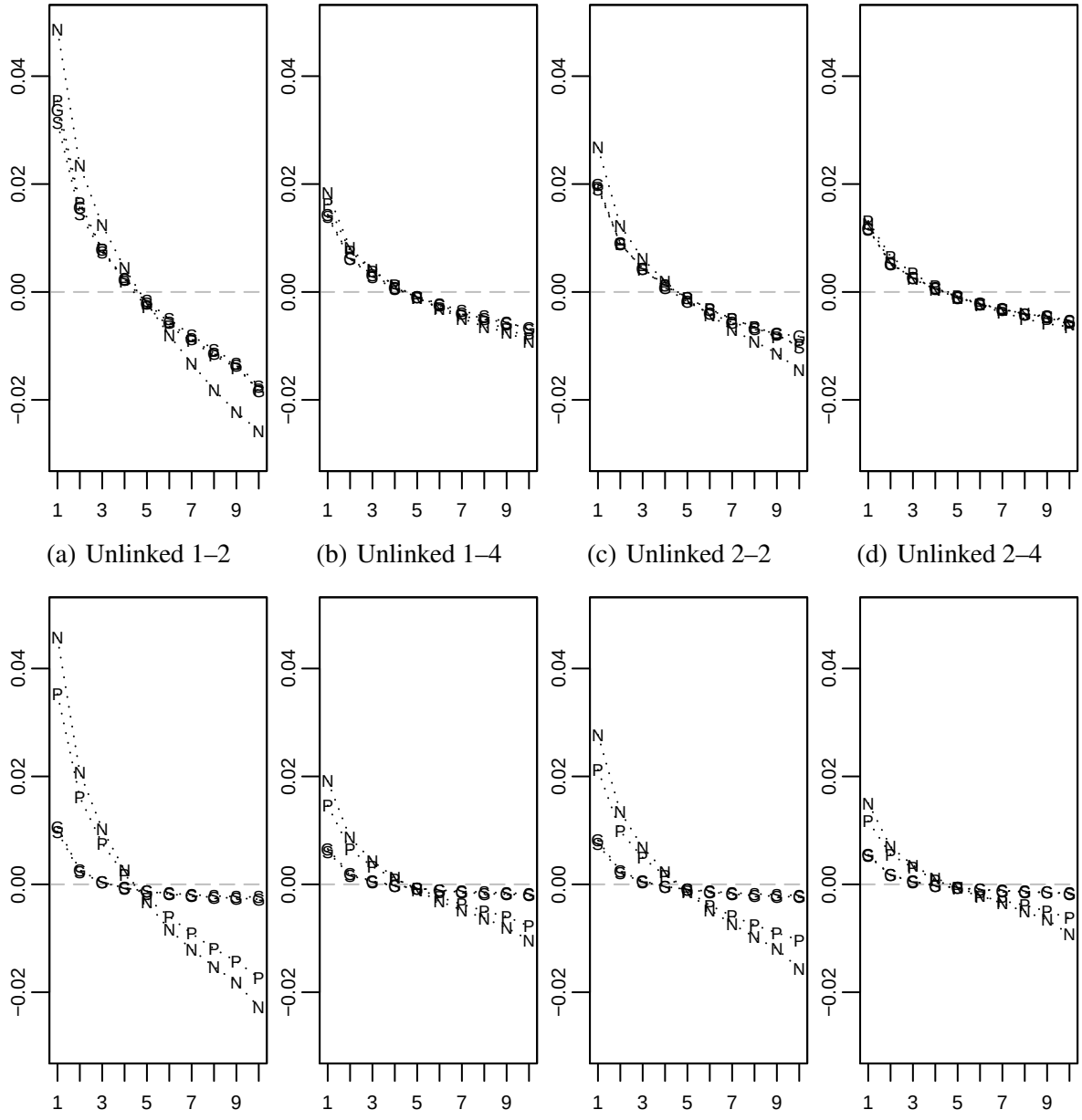
Figure 5.2: Conditional Inbreeding Coefficients by Locus. The lines in the figure correspond to different dispersal regimes. Lines 1–9 respectively correspond to σ_s–σ_d of 1–2, 1–4, 1–Pan, 2–2, 2–4, 2–Pan, Pan–2, Pan–4, and Pan–Pan.



(a) Unlinked Locus (b) S Locus

Figure 5.3: Conditional Inbreeding Coefficients by SI with 95% confidence intervals Bonferroni corrected with $n = 4 \times 9 \times 16 = 576$. Dispersal represents σ_s–σ_p. N, P, G, and S correspond to NSI, PSI, GSI, and SSI respectively.

non-significant values. In these correlegrams, populations with local dispersal of both seed and pollen build up fine-scaled genetic structure. Individuals that are near one another are positively correlated, and individuals that are far apart are negatively correlated. In the absence of self-incompatibility, all markers show approximately equal levels of kinship (Figures 5.6, 5.7, 5.8, and 5.9), but in the presence of self-incompatibility, kinship decreases from the unlinked loci to the S loci (Figures 5.10, 5.11, 5.12, and 5.13). As expected, populations with panmictic dispersal in seeds show no significant level of kinship (Subfigures g–i in Figures 5.7, 5.9, 5.11, and 5.13). Surprisingly, populations with panmictic pollen dispersal, but local seed dispersal show minute but significant levels of kinship at the first distance class and no significant kinship after that (Subfigures c and f in Figures 5.7, 5.9, 5.11, and 5.13).



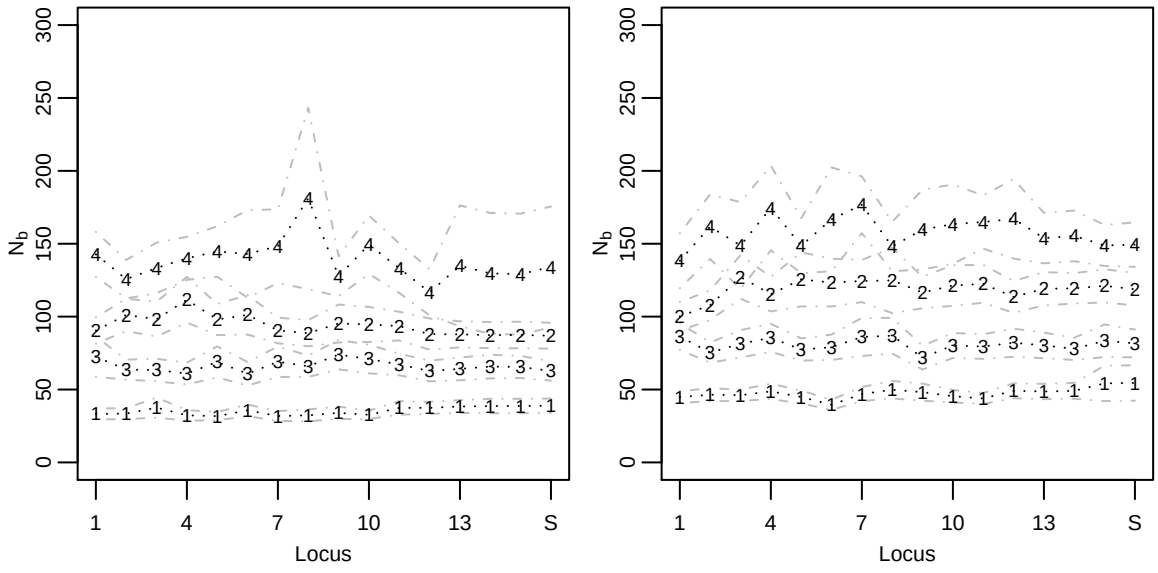
(a) Unlinked 1–2 (b) Unlinked 1–4 (c) Unlinked 2–2 (d) Unlinked 2–4

(e) S Locus 1–2 (f) S Locus 1–4 (g) S Locus 2–2 (h) S Locus 2–4

Figure 5.4: Correlegrams. The x-axis represents distance class, and the y-axis represents average F_{ij}. Each subfigure contains four correlegrams, representing different mating systems for a specific locus and dispersal level. Lines N, P, G, and S correspond to NSI, PSI, GSI, and SSI. Dispersal is represented by σ_s–σ_p.

5.4.4 Neighborhood Sizes

Figure 5.5 shows the neighborhood sizes of populations with local dispersal of seeds and pollen. (These are the populations expected to have reasonable values of N_b.) As expected, NSI and PSI show no effect of linkage to neighborhood size, but GSI and SSI do. Additionally, higher levels of local dispersal produce higher neighborhood sizes. Finally, the neighborhood sizes for the unlinked locus do not show any significant effect of mating system.



(a) NSI (b) PSI

(c) GSI (d) SSI

Figure 5.5: Average Neighborhood Sizes. Only populations with positive, finite neighborhood sizes have been included in this analysis. Lines 1–4 refer to σ_s–σ_p of 1–2, 1–4, 2–2, and 2–4 respectively. 95% confidence intervals have not been corrected for multiple comparisons.

5.5 Discussion

5.5.1 Effective Population Size

Effective population sizes are determined by variances in female and male fecundities as well as the covariance between them. In this study's model, these variances are determined by the dispersal kernel and mating system. Selfing will cause a large covariance in male and female fecundities and thus reduce the effective population size. The rate of selfing is determined by both the mating system and dispersal. Selfing can only happen under NSI, and low values for σ_p have high selfing rates. Alternatively, local dispersal of either pollen or seed decreases the variance in fecundity because it increases isolation-by-distance, which can turn a continuous population into many different effectively isolated subpopulations. However, local dispersal can also increase the variance in reproductive success because individuals on the edge of the population may not have the same opportunities to reproduce as individuals in the middle of the population. They can have both lower pollen pools and less access to suitable habitat for their offspring.

Of the six identified groups, only Group D, which had panmictic pollen and seed dispersal, produced effective populations sizes consistent with the census size. Based on the assumptions for calculating the effective population size, we expect this result for NSI–Pan–Pan; however, we find that the result also holds for other panmictic mating systems. Therefore, under panmixia the effect of self-incompatibility on reproduction was weak, which is not surprising since self-fertilization was rare under panmixia.

Groups A and B produced the highest effective population sizes. These groups were characterized by self-incompatibility and local seed dispersal and had a noticeable decrease in the variance of female fecundity. This decrease in variance can be explained by the decrease of seed shadows and decrease in the overlap of seed shadows. The differences between Groups A and B are consistent with this explanation because Group A has a lower seed dispersal and thus smaller shadows than Group B: $\sigma_s = 1$ versus $\sigma_s = 4$. Furthermore, since these groups are self-incompatible, they

avoid the problem of decreases in local dispersal increasing inbreeding and decreasing the effective population size.

Group E, which consisted of populations with self-incompatibility, panmictic seed dispersal, and local pollen dispersal, produced effective population sizes lower than the census size. These populations have slightly higher variances and covariances of fecundity, which can be attributed to variation in pollen availability due to edge effects.

Groups C and F consist of populations that are self-compatible with local pollen dispersal and thus have non-trivial rates of selfing. However, Group C produced effective population sizes above the census size while Group F produced effective population sizes below the census size. The difference between these two groups highlights the antagonistic interaction of seed and pollen dispersal under NSI. Decreasing seed dispersal increases the effective population sizes because of a decrease in the overlap of seed shadows. However, decreasing pollen dispersal increases the effective population size because it increases selfing. In Group C, which has low seed dispersal and high pollen dispersal, the effects of seed dispersal are greater than the effects of pollen dispersal. However, in Group F, the effects of pollen dispersal are greater than the effects of seed dispersal.

Interestingly, for every dispersal regime PSI, SSI, and GSI are found together in the same group, suggesting that their effects on effective population sizes are similar. The pairwise comparisons in Figure 5.1 found significant differences among the effects of PSI, GSI, and SSI in only there per dispersal comparisons: PSI–2–2 versus SSI–2–2, PSI–1–2 versus SSI–1–2, and GSI–1–2 versus SSI–1–2. At low pollen dispersals, the effects of SSI appear to diverge from PSI and GSI. For these SSI populations, the covariance between male and female fecundity is slightly higher than in their companion PSI and GSI populations. This difference in covariance may be due to SSI producing slightly stronger balancing selection.

5.5.2 Inbreeding

In Figure 5.2 NSI produces higher conditional inbreeding coefficients than PSI, GSI, and SSI. For instance, the largest average conditional inbreeding coefficient of the unlinked locus is greater than

0.20 under NSI but roughly 0.06 in the other three mating systems. From Equation 5.4 the probabilities of selfing expected under NSI are approximately 0.25 when $\sigma_p = 2$ and 0.13 when $\sigma_p = 4$. Furthermore, the probability of selfing contributes half of its value to the conditional inbreeding coefficient because half of selfed progeny are autozygous. This is consistent with the differences observed between the conditional inbreeding coefficients of self-compatible and self-incompatible mating systems, which are roughly half the above selfing probabilities. In fact, populations with panmictic seed dispersal and local pollen dispersal, Pan–2 and Pan–4, have positive conditional inbreeding coefficients only under NSI. The existence and magnitude of these coefficients can be attributed to the presence of selfing in these populations under NSI and its absence under the other mating systems.

Because there is an antagonism between local dispersal and self-incompatibility systems, we expect GSI and SSI to show that inbreeding coefficients depend on linkage to the S locus, which is confirmed by Figures 5.3(c) and 5.3(d). The more closely linked a marker is to the S locus, the lower is its inbreeding coefficient, with the markers near the S locus are outbred while markers farther away are inbred.

From Figure 5.3, we can see that GSI and SSI do not significantly affect conditional inbreeding coefficients for unlinked loci beyond preventing self-fertilizations because the levels of inbreeding for the unlinked locus are the same for PSI, GSI, and SSI. However, because GSI and SSI require the S locus to be outbred, PSI is significantly different than GSI and SSI for the S locus. Surprisingly, no significant difference was detected between GSI and SSI for the inbreeding coefficients.

5.5.3 Fine-Scale Genetic Structure

As seen in Figure 5.4 and section 5.A, NSI generates the greatest genetic structure. Because NSI allows selfing, pollen disperses less, which in turn generates a more structured population. Unlinked loci showed little difference between the correlegrams of PSI, GSI, and SSI, which once again suggests that GSI and SSI only affect unlinked loci by preventing selfing.

The correlegrams found in section 5.A reveal that populations with local seed dispersal and panmictic pollen dispersal have positive levels of kinship in their first distance class and no kinship in the rest of them. Because pollen is panmictic, only seed dispersal generates genetic structure. Local seed dispersal causes halfsibs to be located near one another, generating small levels of kinship in the first distance class. These correlegrams are not affected by the self-incompatibility because self-incompatibility affects pollen dispersal not seed dispersal.

The magnitude of fine-scale genetic structure, although significant, is much lower in these simulations than what is often found in nature. There are several reasons why the simulations may be prevented from generating typical magnitudes of genetic structure. The population size, 2500, may be too small to build up typical levels of fine-scale genetic structure. The habitat size may also be small relative to dispersal. Furthermore, the model does not include mutation, which prevents novel mutations from arising and distinguishing areas of the population. These effects can be tested in subsequent studies.

5.5.4 Neighborhood Sizes

As expected, NSI and PSI showed no effect of linkage to neighborhood size, but GSI and SSI do. Under GSI and SSI balancing selection cause S loci to disperse further than unlinked loci. The neighborhood sizes for the unlinked locus do not show any significant effect of mating system. Furthermore, little to no significant difference between GSI and SSI is apparent from the neighborhood sizes. Additionally, higher levels of local dispersal produce higher neighborhood sizes.

Wright (1946) derived neighborhood size as $N_b = 4\pi\sigma^2 d$, where d is the density of the population. This was derived from the assumption that gametes disperse on the x-axis and y-axis independently and identically with normal distribution of mean 0 and variance σ^2. Because distance is measured in individuals, the density of this study's model was 1. In Wright's model and the model in this study, σ^2 refers to the same thing: one half the mean squared distance of dispersal. Because plants disperse pollen and seed instead of male and female gametes, σ^2 needs to be derived from pollen and seed dispersal: $\sigma^2 = \sigma_s^2 + \sigma_p^2/2$ (Crawford, 1984). For NSI we can apply

this formula directly, resulting in the following neighborhood sizes for the local dispersal regimes: 37.7 for 1–2, 113.1 for 1–4, 75.4 for 2–2, and 150.8 for 2–4. For PSI σ_p^2 needs to be corrected to take into account the lack of selfing. Therefore the corrected value is $\sigma_p^{2\prime} = \sigma_p^2 + \sigma_p/\sqrt{\pi} + 1/2\pi$; note that under Wright's model $\sigma_p^{2\prime} = \sigma_p^2 + 1/2\pi$ .Using this correction, the neighborhood sizes for PSI are as follows: 45.8 for 1–2, 128.3 for 1–4, 83.5 for 2–2, and 166.0 for 2–4.

It is not possible to calculate σ^2 for GSI and SSI without knowing the genetic structure of the population. However, based on the common observation from the simulations that the effects of PSI, SSI, and GSI are similar for unlinked loci, it is possible to use the estimated σ^2 of PSI for estimations for GSI and SSI. Using the PSI values for GSI and SSI, these values are very good fits for the neighborhood sizes of unlinked loci estimated from the correlegrams and suggests that measures for other loci are appropriate. The significance of these fits can be calculated by using a t-test on the logarithm of the neighborhood sizes and adjusting the p-values using Holm correction. Out of these significance tests, only NSI 1–4, PSI 1–4, GSI 1–4, and SSI 1–2 have averages significantly different ($\alpha = 0.05$) than the expectations calculated above. These dispersal regime appear to violate slightly but significantly one of the assumptions used to calculate neighborhood sizes.

5.5.5 Conclusion

This study has confirmed that antagonism exists between local dispersal and self-incompatibility systems. However, loci unlinked to the S locus do not appear to be significantly affected by self-incompatibility systems beyond the prevention of selfing. This suggests that studies of plant populations can ignore the genetics of self-incompatibility when studying other aspects of plant biology. Specifically, computational models can assume obligate outcrossing (physical self-incompatibility) without the need to model its genetics. Surprisingly, differences between gametophytic and sporophytic self-incompatibility systems were small to non-significant, suggesting that their impact on plant populations are equivalent. These results suggest that the evolutionary advantage of self-

incompatibility systems in large populations is the prevention of selfing and does not extend to the prevention of inbreeding.

5.6 References

Awadalla, P. and D. Charlesworth (1999). Recombination and selection at brassica self-incompatibility loci. *Genetics 152*, 413–425.

Bateman, A. J. (1952). Self-incompatibility systems in angiosperms i. theory. *Heredity 6*, 285–310.

Bechsgaard, J., T. Bataillon, and M. H. Schierup (2004). Uneven segregation of sporophytic self-incompatibility alleles in *arabidopsis lyrata*. *Journal of Evolutionary Biology 17*, 554–561.

Brooks, R. J., A. M. Tobias, and M. J. Lawrence (1996). The population genetics of the self-incompatibility polymorphism in papaver rhoeas. xi. the effects of limited pollen and seed dispersal, overlapping generations and variation in plant size on the variance of s-allele frequencies in populations at equilibrium. *Heredity 76*, 367–376.

Castric, V. and X. Vekemans (2004). Plant self-incompatibility in natural populations: a critical assessment of recent theoretical and empirical adavances. *Molecular Ecology 13*, 2873–2889.

Charlesworth, D. (2006). Balancing selection and its effects on sequences in nearby genome regions. *PLoS Genetics 2*(4), 379–384.

Charlesworth, D., E. Kamau, J. Hagenblad, and C. Tang (2006). Trans-specificity at loci near the self-incompatibility loci in arabidopsis. *Genetics 172*, 2699–2704.

Crawford, T. J. (1984). The estimation on neighbourhood parameters for plant populations. *Heredity 52*, 273–283.

Epperson, B. K. (1990). Spatial autocorrelation of genotypes under directional selection. *Genetics 124*, 757–771.

Fenster, C. B., X. Vekemans, and O. J. Hardy (2003). Quantifying gene flow from spatial genetic structure data in a metapopulation of *chamaecrista fasciculata* (leguminosae. *Evolution 57*, 995–1007.

Gibbs, P. E. and M. B. Bianchi (1999). Does late-acting self-incompatibility (lsi) show family clustering? two more species of bignoniaceae with lsi:dolichandra cynanchoides and tabebuia nodosa. *Annals of Botany 84*(4), 449–457.

Hagenblad, J., J. Bechsgaard, and D. Charlesworth (2006). Linkage disequilibrium between incompatibility locus region genes in the plant arabidopsis lyrata. *Genetics*, genetics.106.055780. Epub ahead of print.

Hardy, O. J. and X. Vekemans (1999). Isolation by distance in a continuous popualtion: reconciliation between spatial autocorrelation analysis and population genetics models. *Heredity 83*, 145–154.

Hardy, O. J. and X. Vekemans (2002). Spagedi : a versatile computer program to analyse spatial genetic structure at the individual or population levels. *Molecular Ecology Notes 2*, 618–620.

Holm, D. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics 6*, 65–70.

Igic, B. and J. R. Kohn (2001). Evolutionary relationships among self-incompatibility rnases. *Proc. Natl. Acad. Sci. U. S. A. 98*, 13167–13171.

Kamau, E. and D. Charlesworth (2005). Balancing selection and low recombination affect diversity near the self-incompatibility loci of the plant arabidopsis lyrata. *Current Biology 15*, 1773–1778.

Kot, M., M. A. Lewis, and P. van den Driessche (1996). Dispersal data and the spread of invading organisms. *Ecology 77*, 2027–2042.

Lipow, S. R. and R. Wyatt (2000). Single gene control of postzygotic self-incompatibility in poke milkweed, asclepias exaltata l. *Genetics 154*(2), 893–907.

Loiselle, B. A., V. L. Sork, J. Nason, and C. Graham (1995). Spatial genetic structure of a tropical understory shrub, *Psychotria officinalis* (Rubiaceae). *American Journal of Botany 82*, 1420–1425.

Malécot, G. (1975). Heterozygosity and relationship in regularly subdivided populations. *Theoretical Population Biology 8*, 212–241.

Muirhead, C. (2001). Consequences of population structure on genes under balancing selection. *Evolution 55*, 1532–1541.

Nasrallah, J. B. (2005). Recognition and rejection of self in plant self-incompatibility: comparisons to animal histocompatibility. *Trends in Immunology 26*(8), 412–418.

Neuhauser, C. (1998). The ancestral graph and gene genealogy under frequency-dependent selection. *Theoretical Population Biology 56*, 203–214.

R Development Core Team (2006). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Rohlf, F. J. and G. D. Schnell (1971). An investigation of the isolation-by-distance model. *The American Naturalist 105*, 295–324.

Rousset, F. (1997). Genetic differentiation and estimation of gene flow from f-statistics under isolation by distance. *Genetics 145*, 1219–1228.

Rousset, F. (2000). Genetic differentiation between individuals. *J. Evol. Biol. 13*, 58–62.

Sawyer, S. (1977). Asymptotic properties of the equilibrium probability of identity in geographically structured populations. *Advances in Applied Probability 9*, 268–282.

Schierup, M. H. (1998). The number of self-incompatibility alleles in a finite, subdivided population. *Genetics 149*, 1153–1162.

Schierup, M. H., D. Charlesworth, and X. Vekemans (2000). The effect of hitch-hicking on genes linked to a balanced polymorphism in a subdivided population. *Genetical Research Cambridge 76*, 63–73.

Schierup, M. H., X. Vekemans, and D. Charlesworth (2000). The effect of subdivision on variation at multi-allelic loci under balancing selection. *Genetical Research Cambridge 76*, 51–62.

Schierup, M. H., X. Vekemans, and F. B. Christiansen (1997). Evolutionary dynamics of sporophytic self-incompatibility alleles in plants. *Genetics 147*, 835–846.

Seavey, S. R. and K. S. Bawa (1986). Late-acting self-incompatibility in angiosperms. *Botanical Review 52*(2), 195–219.

Shaffer, J. P. (1995). Multiple hypothesis testing. *Annual Review of Psychology 46*(1), 561–584.

Sokal, R. R. and D. E. Wartenberg (1983). A test of spatial autocorrelation analysis using an isolation-by-distance model. *Genetics 105*, 219–237.

Takayama, S. and A. Isogai (2005). Self-incompatibility in plants. *Annual Review of Plant Biology 56*, 467–489.

Turner, M. E., J. C. Stephens, and W. W. Anderson (1982). Homozygosity and patch structure in plant populations as a result of nearest-neighbor pollination. *Proc. Natl. Acad. Sci. USA 79*, 203–207.

Uyenoyama, M. K. (1997). Genealogical structure among alleles regulating self-incompatibility in natural populations of flowering plants. *Genetics 147*, 1389–1400.

Uyenoyama, M. K. (2000). Evolutionary dynamics of self-incompatibility alleles in *brassica*. *Genetics 156*, 351–359.

Uyenoyama, M. K. (2003). Genealogy-dependent variation in viability among self-incompatibility genotypes. *Theoretical Population Biology 63*, 281–293.

Vekemans, X. and O. J. Hardy (2004). New insights from fine-scale spatial genetics structure analyses in plant populations. *Molecular Ecology 13*, 921–935.

Wright, S. (1938). Size of population and breeding structure in relation to evolution. *Science 87*, 430–431.

Wright, S. (1939). The distribution of self-sterility alleles in populations. *Genetics 24*, 538–552.

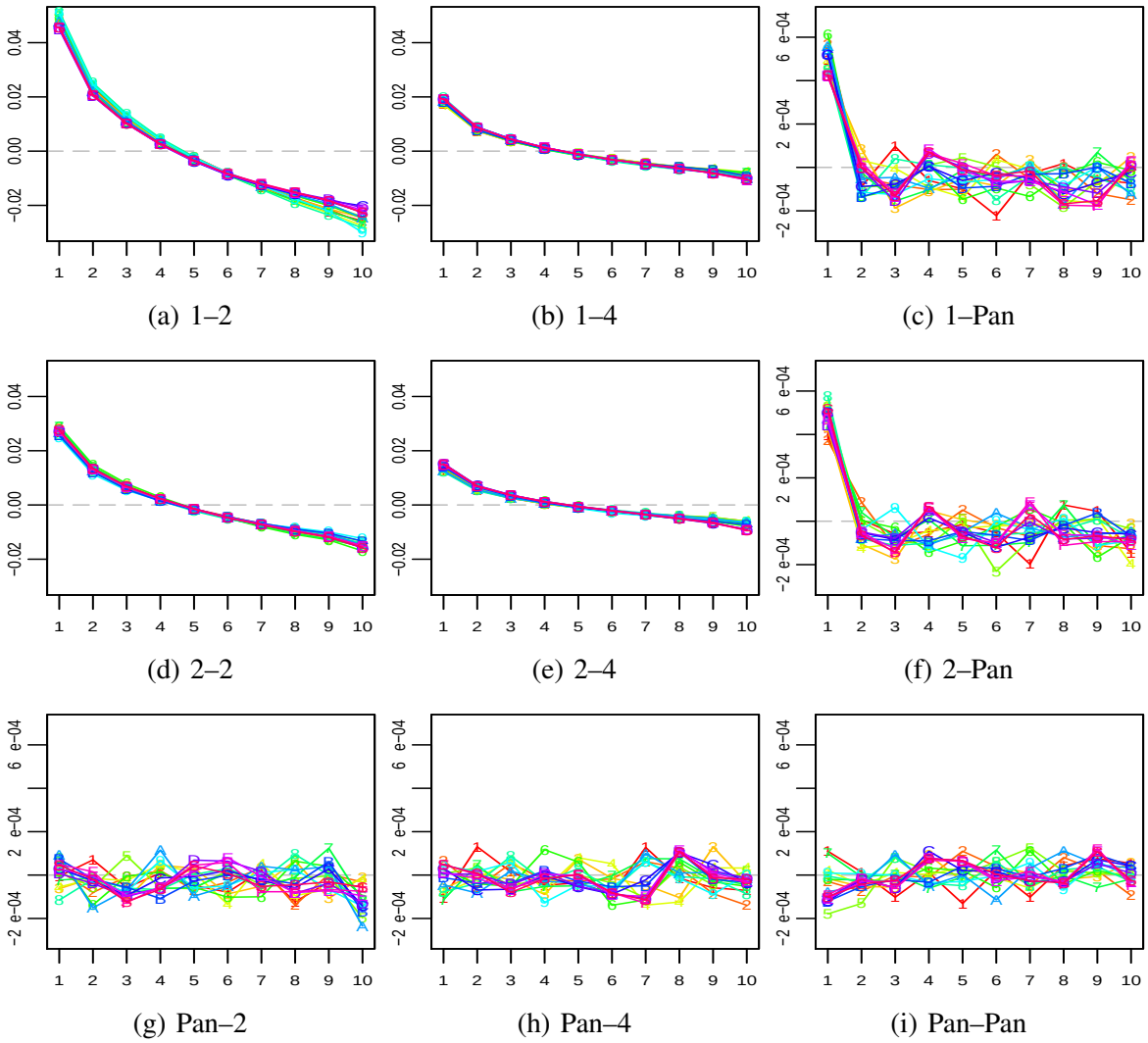
Wright, S. (1943). Isolation by distance. *Genetics 28*, 114–138.

Wright, S. (1946). Isolation by distance under diverse systems of mating. *Genetics 31*, 39–59.

Wright, S. (1965). The interpretation of population structure by f-statistics with special regard to systems of mating. *Evolution 19*, 395–420.

Wright, S. P. (1992). Adjusted p-values for simultaneous inference. *Biometrics 48*(4), 1005–1013.

APPENDIX 5.A CORRELEGRAMS

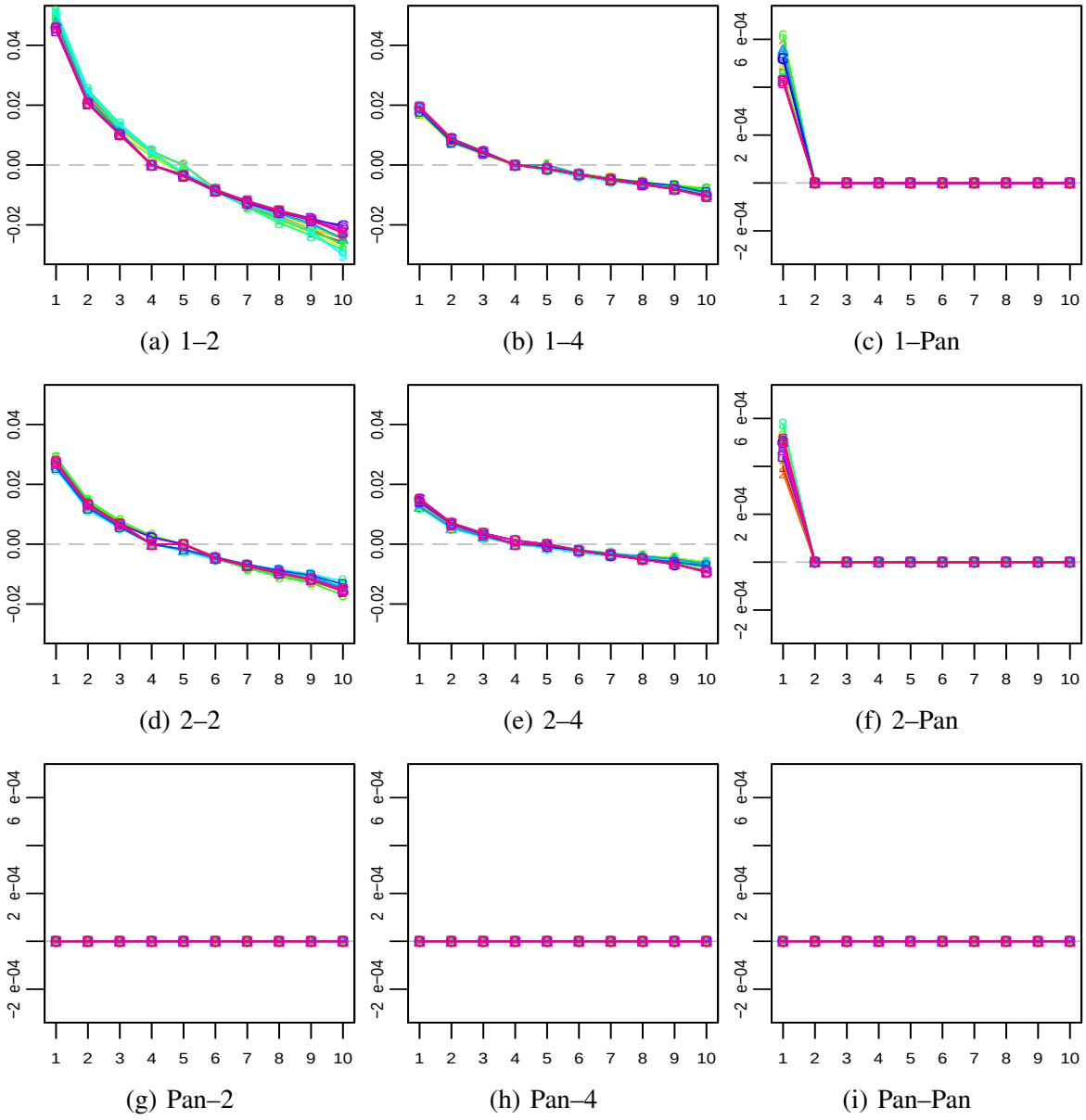


(a) 1–2 (b) 1–4 (c) 1–Pan

(d) 2–2 (e) 2–4 (f) 2–Pan

(g) Pan–2 (h) Pan–4 (i) Pan–Pan

Figure 5.6: NSI Correlegrams. Lines 1-F represent the 15 marker loci with 1 being the unlinked locus and f being locus 15. Line S represents the S locus. Subfigure labels refer to dispersal, i.e. σ_s–σ_p. The x-axis is distance class, and the y-axis is $\bar{F}_{ij}$. The scale of the y-axis is different for panmictic dispersals.

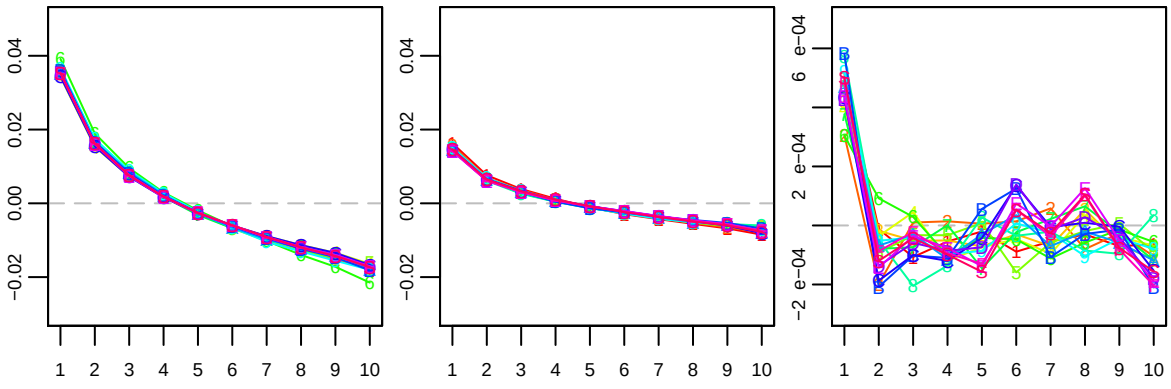


(a) 1–2 (b) 1–4 (c) 1–Pan

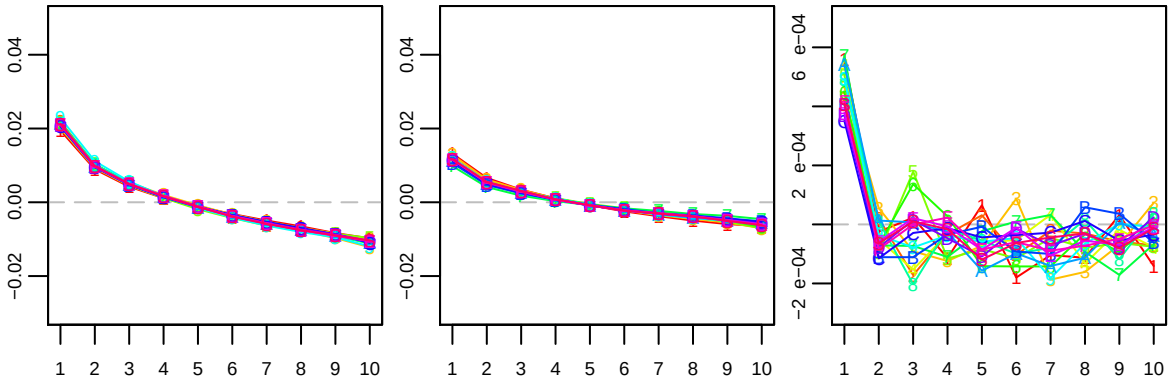
(d) 2–2 (e) 2–4 (f) 2–Pan

(g) Pan–2 (h) Pan–4 (i) Pan–Pan

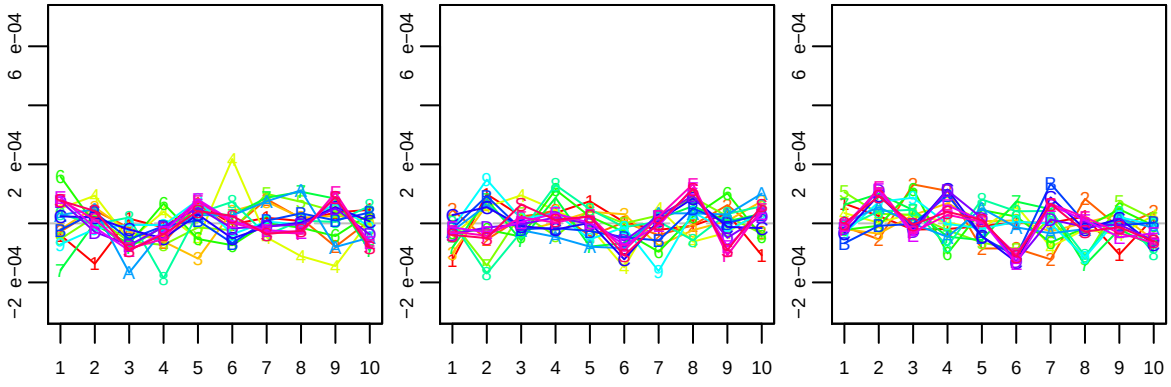
Figure 5.7: Corrected NSI Correlegrams. Figure 5.6 has been modified such that any $\bar{F}_{ij}$ not significantly different than zero is set as zero. Significance is determined by 95% confidence intervals Bonferroni corrected for $10 \times 9 \times 16 = 5760$ trials.



(a) 1–2 (b) 1–4 (c) 1–Pan

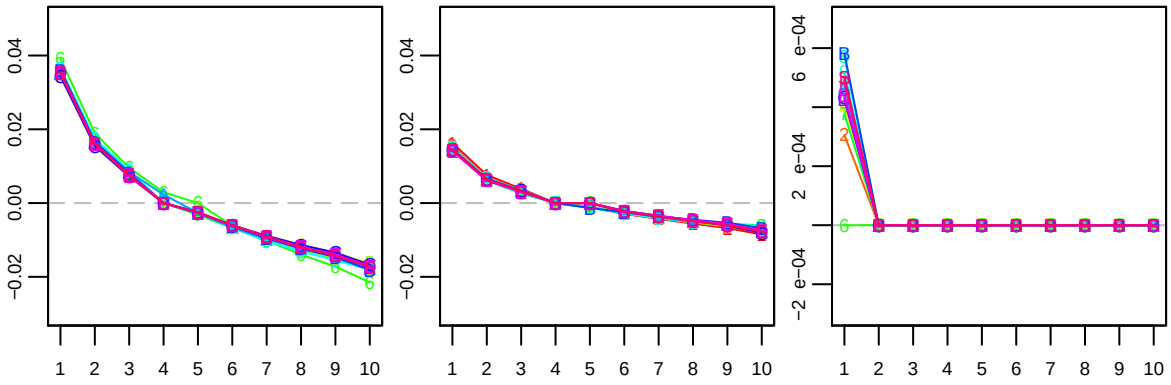


(d) 2–2 (e) 2–4 (f) 2–Pan



(g) Pan–2 (h) Pan–4 (i) Pan–Pan

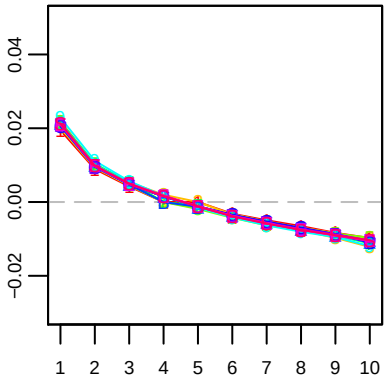
Figure 5.8: PSI Correlegrams. See Figure 5.6 for a description of the subfigures.



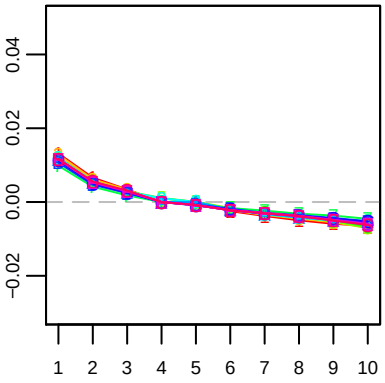
(a) 1–2

(b) 1–4

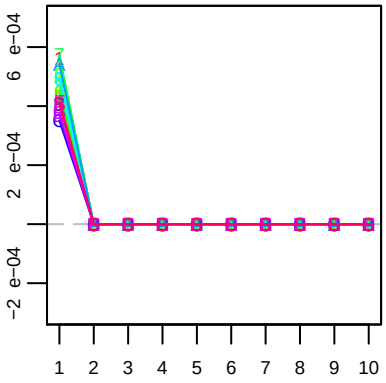
(c) 1–Pan



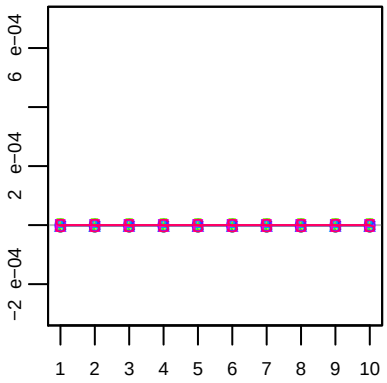
(d) 2–2



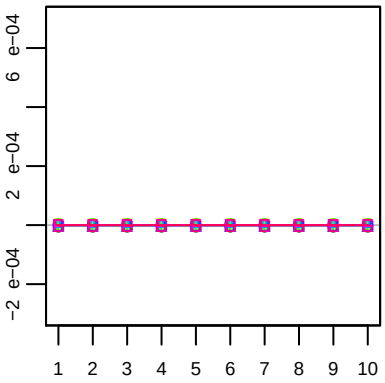
(e) 2–4



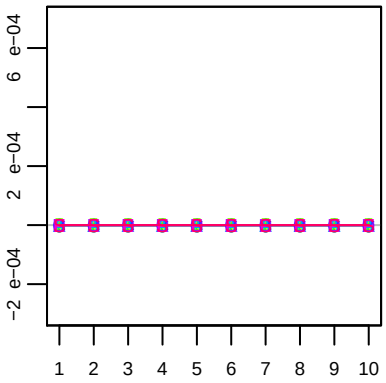
(f) 2–Pan



(g) Pan–2

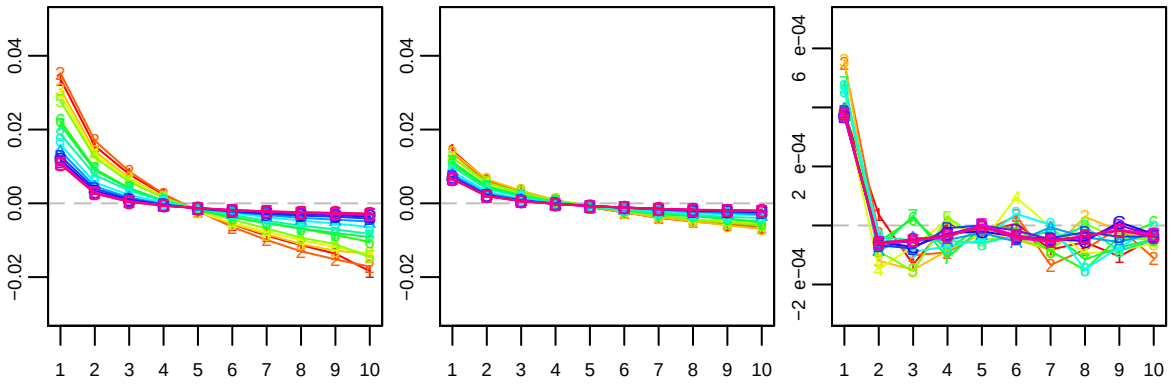


(h) Pan–4

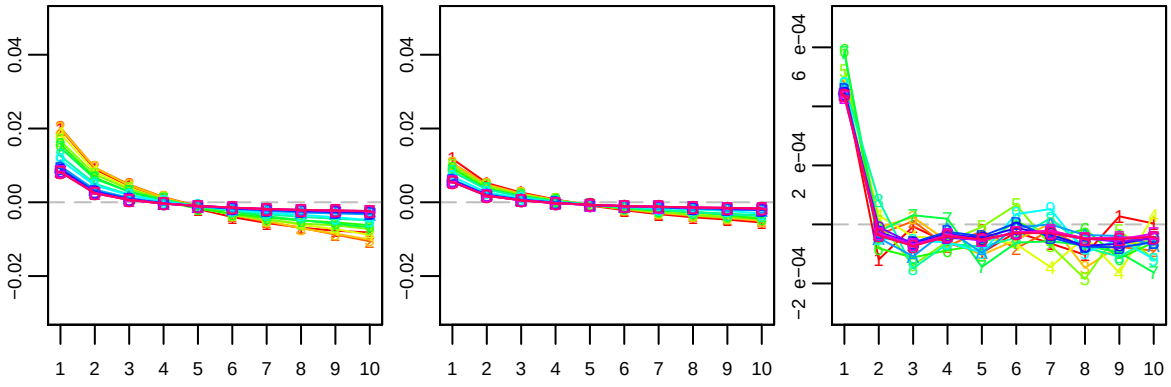


(i) Pan–Pan

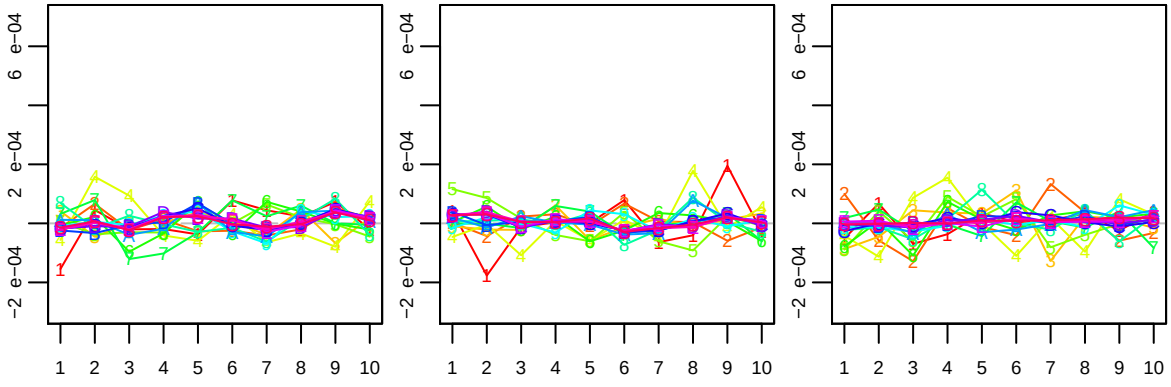
Figure 5.9: Corrected PSI Correlegrams. See Figure 5.6 and Figure 5.7 for descriptions of the subfigures.



(a) 1–2 (b) 1–4 (c) 1–Pan

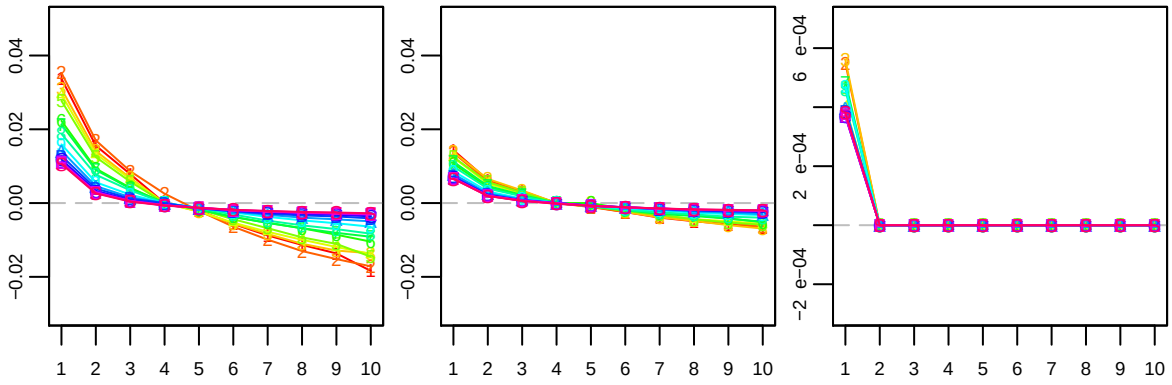


(d) 2–2 (e) 2–4 (f) 2–Pan

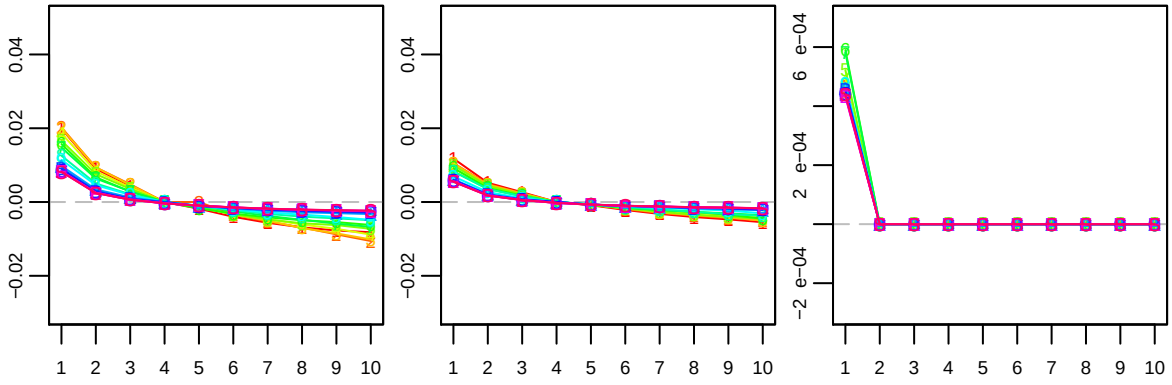


(g) Pan–2 (h) Pan–4 (i) Pan–Pan

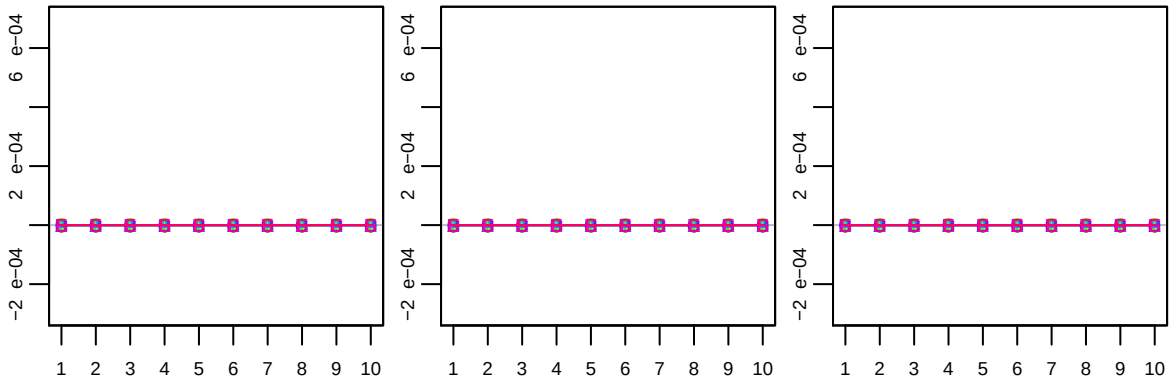
Figure 5.10: GSI Correlegrams. See Figure 5.6 for a description of the subfigures.



(a) 1–2 (b) 1–4 (c) 1–Pan

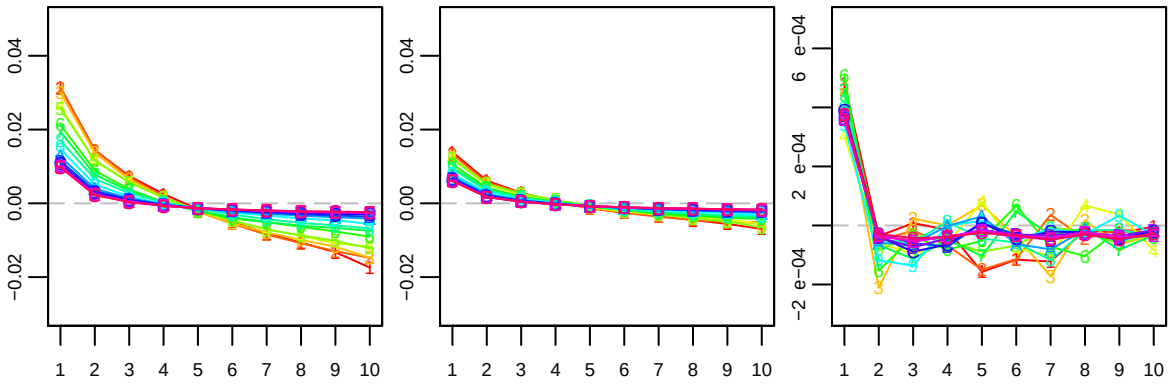


(d) 2–2 (e) 2–4 (f) 2–Pan

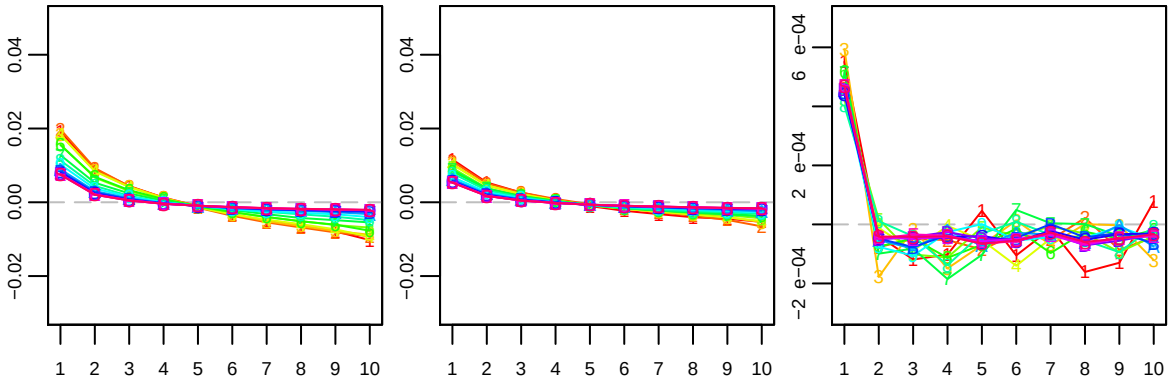


(g) Pan–2 (h) Pan–4 (i) Pan–Pan

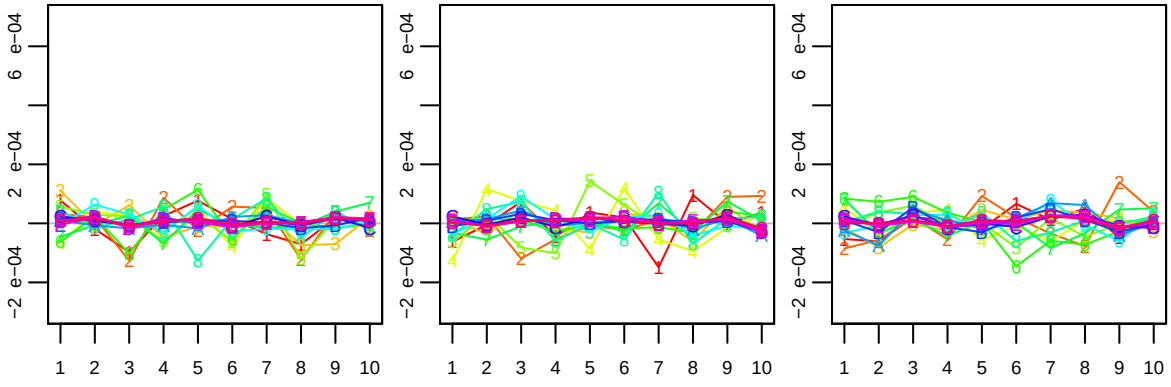
Figure 5.11: Corrected GSI Correlegrams. See Figure 5.6 and Figure 5.7 for descriptions of the subfigures.



(a) 1–2 (b) 1–4 (c) 1–Pan

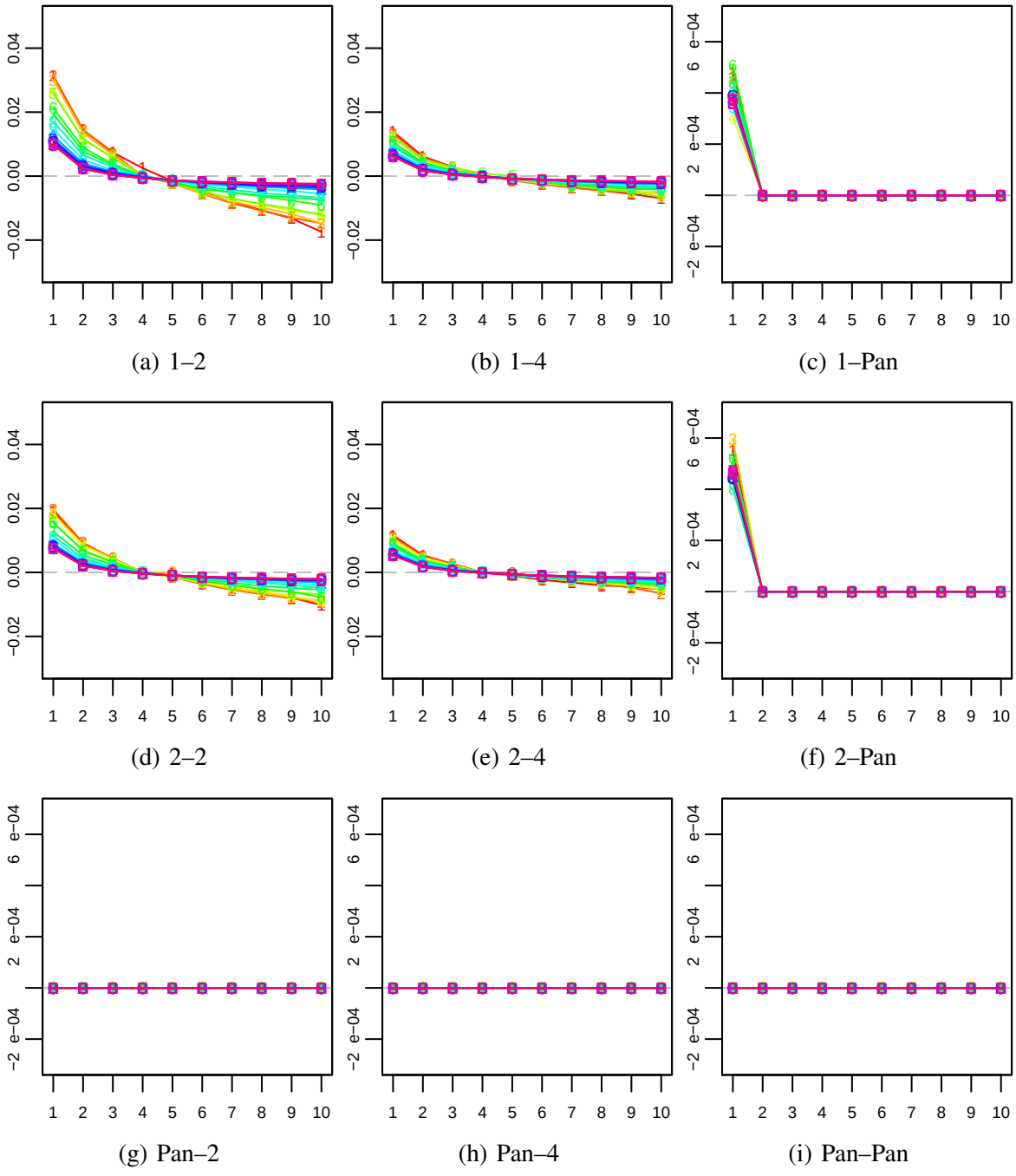


(d) 2–2 (e) 2–4 (f) 2–Pan



(g) Pan–2 (h) Pan–4 (i) Pan–Pan

Figure 5.12: SSI Correlegrams. See Figure 5.6 for a description of the subfigures.



(a) 1–2 (b) 1–4 (c) 1–Pan

(d) 2–2 (e) 2–4 (f) 2–Pan

(g) Pan–2 (h) Pan–4 (i) Pan–Pan

Figure 5.13: Corrected SSI Correlegrams. See Figure 5.6 and Figure 5.7 for descriptions of the subfigures.

CHAPTER 6

CONCLUSION

In the first study, I developed, Dawg, a portable application capable of simulating sequence evolution using the robust GTR+Γ+I model of sequence substitution and the power-law model of indel lengths. This program offers many advantages over existing sequence simulation programs.

In the second study, I applied Dawg to investigate whether logarithmic gap costs could improve alignment accuracy as has been suggested in the literature. I found that logarithmic gap costs actually decrease alignment accuracy. Furthermore, this decrease is considerable compared the affine gap costs. This unexpected result can be explained by the conversion of a maximum likelihood search into a minimum cost search which introduces a linear component into the cost. This linear component dominates the logarithmic component that derives from the power-law distribution of gap lengths. Using the mean distance between sequences, the rate of indel formation, and the distribution of gap lengths, I derived a model for calculating the gap costs for biological data. This model could prove useful for tuning alignment parameters to biological data.

In the final study, I investigated the antagonism between self-incompatibility systems and local dispersal, focusing on the interaction of isolation-by-distance and identity-by-descent. As expected, populations with local dispersal built up fine scaled genetic structure and self-incompatibility systems modified the level of structure. Also as expected, self-incompatibility loci had a stronger effect on closely linked markers than on unlinked loci. However, unexpectedly, gametophytic and sporophytic self-incompatibility showed no significant differences in their effect on effective population sizes, neighborhood sizes, conditional inbreeding coefficients, or fine scale genetic structure.

APPENDIX A

NGILA: GLOBAL PAIRWISE ALIGNMENTS WITH LOGARITHMIC AND AFFINE GAP COSTS

Over the last two decades, several molecular studies have demonstrated that the lengths of indels obey a power-law. Under a power law, a log-log transformation of frequencies is linear. A Zipfian distribution produces a power-law in discrete data. Under a Zipfian distribution, the probability that an observation is $x = \{1,2,\ldots\}$ is $F(x|z) = x^{-z}/\zeta(z)$, where $z > 1$ is a parameter and $\zeta(z) = \sum_{n=1}^{\infty} n^{-z}$ is Reimann's Zeta Function. The mean and variance of this distribution are undefined (i.e. infinite) for $z \leq 2$ and $z \leq 3$, respectively.

Based on this power law some researchers have argued that logarithmic gap penalties may be appropriate for sequences alignments (Gonnet et al., 1992; Gu and Li, 1995; Waterman, 1984). However, there are no programs available that can align sequences using logarithmic gap penalties, even though established algorithms like Miller and Myers (1988) show how to do it. Here I present Ngila, a portable implementation of their algorithm that can globally align pairs of sequences using logarithmic and affine gap penalties.

Needleman and Wunsch (1970) provided a method of sequence alignment where the maximum similarity of a pair of sequences was calculated using dynamic programming. Sellers (1974) provided an alternative approach where the minimum distance of the sequence pair was calculated instead. These algorithms took $O(MN)$ time and $O(MN)$ space where M and N are the lengths of the two sequences being aligned. Waterman et al. (1976) generalized Sellers's metric to arbitrary gap weights, producing an $O(MN(M+N))$ algorithm, referred to as WSB. Gotoh (1982) demonstrated that with affine gap penalties the WSB algorithm can be run in $O(MN)$ time. Waterman (1984) subsequently modified the WSB algorithm into the candidate list method for concave gap weights, which could simplify the search process. Waterman's candidate list method was further refined by Miller and Myers (1988) into an $O(MN)$ algorithm. In addition, Miller and Myers (1988) pointed out how their algorithm could be modified using Hirschberg (1975)'s divide-and-conquer method to produce alignments in $O(MNlog(M+N))$ time but with $O(M)$ space. Ngila implements this algorithm in order find a least costly global alignment of two sequences given a gap cost of $w_k = a + bk + c\ln k$.

Ngila is controlled through a command-line interface. Switches "-a", "-b", and "-c" control the coefficients of gap cost. Switch "-m" sets the cost of a match, and switch "-r" sets the cost of a mismatch, whereas switch "-x" specifies a substitution cost matrix. Switch "-f" tells Ngila to align using cost-free end gaps. Source code for Ngila can be downloaded from its development website: `http://scit.us/projects/ngila/`. The source code can be compiled on systems that support the GNU Autoconfig and Automake tools. This includes most flavors of Unix like Linux or FreeBSD, the MinGW environment for Windows, and Macintosh OS X. Project files for Microsoft's Visual Studio .Net 2003 are also included with the source.

The algorithm in Ngila finds the minimum cost of aligning two sequences (Sellers, 1974) instead of the maximum similarity of aligning two sequences (Needleman and Wunsch, 1970). An extension of the method of Smith et al. (1981) can be used to convert a maximum search to a minimum search. This is important when looking to use Ngila for maximum likelihood alignments. Based on a statistical model, the scores of "matches" of type i, α_i, and the penalties of gaps of length k, w_k, can be used to calculate the alignment with maximum log-likelihood:

$$l = \max\left\{\sum \alpha_i \eta_i - \sum w_k \Delta_k\right\} \tag{A.1}$$

where η_i is the number of residue matches of type i and Δ_k is the number of gaps of length k. A minimum cost analog of Equation A.1 is

$$d = \min\left\{\sum \beta_i \eta_i + \sum G(k)\,\Delta_k\right\} \tag{A.2}$$

To begin constructing the minimum cost analog, let $\beta_i = (x - \alpha_i)/y$ be the cost of a match of type i, therefore

$$\begin{aligned} -l = \min\left\{-\sum \alpha_i \eta_i + \sum w_k \Delta_k\right\} &= \min\left\{\sum (y\beta_i - x)\,\eta_i + \sum w_k \Delta_k\right\} \\ &= y \min\left\{\sum \beta_i \eta_i - \frac{x}{y}\sum \eta_i + \sum \frac{w_k}{y}\Delta_k\right\} \end{aligned} \tag{A.3}$$

The lengths of the sequences being aligned, n and m, can be related to the alignment itself via the equation $n+m=2\sum\eta_i+\sum k\Delta_k$. Using this relationship, Equation A.3 can be expressed as

$$\begin{aligned}
-l &= y\min\left\{\sum\beta_i\eta_i-\frac{x}{2y}\left(n+m-\sum k\Delta_k\right)+\sum\frac{w_k}{y}\Delta_k\right\} \\
&= -\frac{x(n+m)}{2y}+y\min\left\{\sum\beta_i\eta_i+\frac{x}{2y}\sum k\Delta_k+\sum\frac{w_k}{y}\Delta_k\right\} \\
&= -\frac{x(n+m)}{2y}+y\min\left\{\sum\beta_i\eta_i+\sum\left(\frac{xk}{2y}+\frac{w_k}{y}\right)\Delta_k\right\} \\
&= -\frac{x(n+m)}{2y}+y\min\left\{\sum\beta_i\eta_i+\sum G(k)\Delta_k\right\} \qquad \text{(A.4)}
\end{aligned}$$

From this it can be clearly seen that $d=\min\left\{\sum\beta_i\eta_i+\sum G(k)\Delta_k\right\}$ maximizes the likelihood of the alignment, where $G(k)=(xk/2+w_k)/y$ is the cost of a gap of length k. This result allows minimum cost algorithms like Ngila to be used to calculate maximum likelihood alignments.

A.1 REFERENCES

Gonnet, G. H., M. A. Cohen, and S. A. Benner (1992). Exhaustive matching of the entire protein sequence database. *Science 256*, 1443–1445.

Gotoh, O. (1982). An improved algorithm for matching biological sequences. *J. Mol. Biol. 162*, 705–708.

Gu, X. and W. H. Li (1995). The size distribution of insertions and deletions in human and rodent pseudogenes suggests the logarithmic gap penalty for sequence alignment. *J. Mol. Evol. 40*, 464–473.

Hirschberg, D. S. (1975). A linear space algorithm for computing maximal common subsequences. *Comm. ACM. 18*, 341–343.

Miller, W. and E. W. Myers (1988). Sequence comparison with concave weighting functions. *Bull. Math. Biol. 50*, 97–120.

Needleman, S. B. and C. D. Wunsch (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol. 48*, 443–453.

Sellers, P. H. (1974). On the theory and computation of evolutionary distances. *SIAM J. Appl. Math. 26*, 787–793.

Smith, T. F., M. S. Waterman, and W. M. Fitch (1981). Comparative biosequence metrics. *J. Mol. Evol. 18*, 38–46.

Waterman, M. S. (1984). Efficient sequence alignment algorithms. *J. Theor. Biol. 108*, 333–337.

Waterman, M. S., T. F. Smith, and W. A. Beyer (1976). Some biological sequence metrics. *Advances in Mathematics 20*, 367–387.