

Genome Sequencing, Rare-Variant Methods, and **20** Simulation-Based Approaches in Statistical Genetics

20.1: Learning Objectives

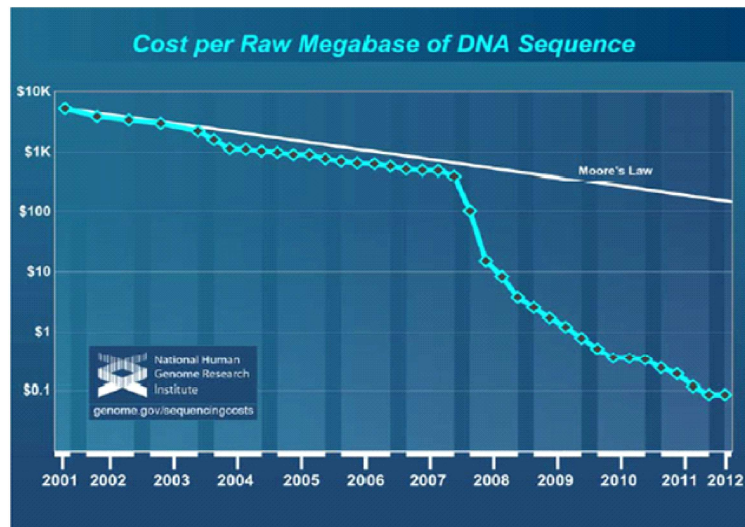
By the end of this chapter, the reader should be able to describe the basic principles of high-throughput (next-generation) sequencing and how it differs from array genotyping, explain the “missing heritability” problem and the common-disease/rare-variant hypothesis, describe burden and variance-component tests for rare-variant association, explain the coalescent and forward-in-time simulation paradigms and why simulation is indispensable in modern statistical genetics, and implement a coalescent simulation and a simulation-based power calculation in R and Python.

20.2: From Genotyping Arrays to High-Throughput Sequencing

All of the association methods developed so far in this book (Chapters 11, 12, 13, 18, and 19) work with genotypes at a pre-specified set of markers, obtained either from a **genotyping array** (which measures only a fixed panel of previously known, common SNPs) or from imputation against a reference panel. **High-throughput sequencing** (also called next-generation sequencing, NGS), by contrast, reads the DNA sequence directly, base by base,

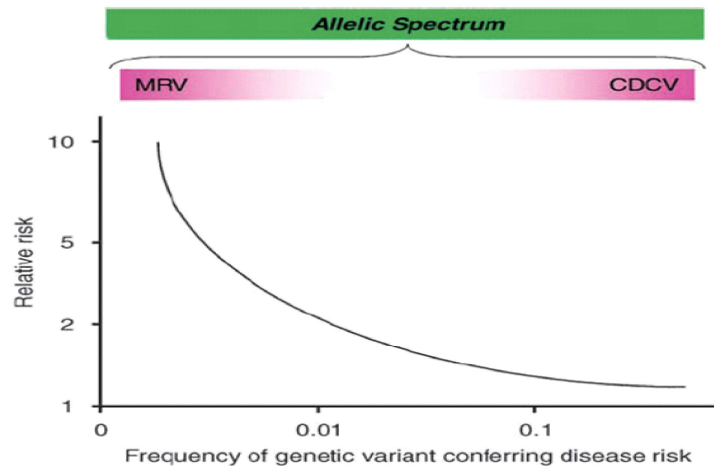
across some or all of an individual's genome, and can therefore detect **novel and rare variants** that were never included on any genotyping array and cannot be reliably imputed, because reliable imputation of a variant absent from every reference-panel haplotype is, by construction, impossible (Chapter 19).

Two sequencing strategies are in common use: **whole-genome sequencing (WGS)**, which reads the entire ~3.2 billion base-pair human genome (at some chosen average **read depth**, or coverage — the number of independent sequencing reads overlapping each position, typically 15–30× for population studies, higher for clinical diagnostic use), and **whole-exome sequencing (WES)**, which sequences only the protein-coding exons (roughly 1–2% of the genome) at much higher depth for a similar total sequencing cost, on the reasoning that a large share of clearly interpretable, high-penetrance disease variants lie in coding sequence. From raw sequencing reads, a standard bioinformatics pipeline aligns reads to a reference genome, identifies positions differing from the reference (**variant calling**), and outputs the results in the standard **Variant Call Format (VCF)**, the file format underlying essentially all modern sequencing-based genetic analysis.



The dramatic decline in the cost of DNA sequencing per raw megabase since 2001, which made population-scale sequencing studies feasible (NHGRI).

20.3: Missing Heritability and the Common-Disease/Rare-Variant Hypothesis



The allelic spectrum: relative risk versus allele frequency, contrasting the common-disease/common-variant (CDCV) and common-disease/rare-variant (CDRV) hypotheses.

Chapter 18 (Section 18.4) noted that genome-wide-significant common variants identified by GWAS typically explain only a modest fraction of a complex trait's total heritability (Chapter 9) — a gap often called the **missing heritability** problem. The prevailing hypothesis underlying most GWAS design, the **common-disease/common-variant (CDCV)** hypothesis, holds that common disease risk is driven substantially by variants that are themselves common in the population (and therefore well captured by genotyping arrays and imputation). An alternative, complementary hypothesis, **common-disease/rare-variant (CDRV)**, holds that a meaningful share of disease risk and missing heritability is attributable to variants that are individually rare — conventionally, a **rare variant** is defined as one with minor allele frequency below some threshold, commonly 1% or 5%, a cutoff that is somewhat arbitrary but reflects the practical point at which genotyping arrays and standard imputation lose reliable power (Chapter 19, Section 19.4). Rare variants of large effect are difficult to detect using the single-marker association tests of Chapter 11, both because each individual rare variant is carried by very few people in any realistic sample (giving very low power for a single-marker test) and because, being rare, such variants are poorly tagged by nearby common markers and hence poorly imputed — motivating direct sequencing as the appropriate detection tool.

20.4: Rare-Variant Association Tests: Burden and Variance-Component Approaches

Because no single rare variant typically has enough carriers for an adequately powered single-marker test, rare-variant association methods instead **aggregate** information across all rare variants observed within a defined genomic unit (usually a gene). Two broad classes of aggregation test are in standard use:

- **Burden tests** collapse all qualifying rare variants within a gene into a single summary score per individual — most

simply, the total number of rare, qualifying (e.g., predicted-damaging) variants carried by that individual across the gene — and test this single burden score for association with the trait using an ordinary regression model (Chapter 11, Section 11.7). Burden tests are most powerful when most of the rare variants in the gene genuinely increase (or, symmetrically, all decrease) risk in the same direction.

- **Variance-component tests**, of which the **Sequence Kernel Association Test (SKAT)** is the most widely used, instead test whether the *variance* of a set of variant effects within the gene departs from zero, without requiring the variants to act in a consistent direction; this makes SKAT-type tests considerably more robust than a pure burden test when a gene harbours a mixture of risk-increasing and risk-decreasing (or some null) rare variants, at some cost in power when all truly causal variants genuinely act in the same direction. Combined tests (such as SKAT-O) interpolate between the pure burden and pure variance-component approaches, choosing the weighting that best fits the observed data, and are now the default choice in most modern rare-variant sequencing studies.

Theorem 20.1 (Burden Test Power Collapse Under Mixed-Direction Effects)

Consider a gene harbouring K rare variants with true effect sizes $\beta_1, \dots, \beta_K$ on a quantitative trait. The non-centrality parameter (and hence the power) of the burden test, which sums carrier status across all K variants into a single score, is proportional to $(\sum_{k=1}^K \beta_k)^2$, whereas the non-centrality parameter of the SKAT variance-component test is proportional to $\sum_{k=1}^K \beta_k^2$. Consequently, if the β_k are of mixed sign with $\sum_k \beta_k = 0$, the burden test's non-centrality parameter is exactly zero (no power, regardless of the individual effect magnitudes), while SKAT's non-centrality parameter remains strictly positive whenever any $\beta_k \neq 0$.

Proof sketch. Under a standard linear aggregation model, the burden test statistic is (proportional to) a test of $H_0: \sum_k \beta_k = 0$ using the pooled burden score $B_i = \sum_k G_{ik}$ (total rare-allele count across the K variants for individual i); its power depends on the squared true value of the aggregated effect, $(\sum_k \beta_k)^2$, since the test is fundamentally testing this single linear combination. If the true effects are exactly offsetting (e.g., half the variants increase risk, half decrease it by the same average magnitude, so $\sum_k \beta_k = 0$), the quantity being tested is truly zero even though individual $\beta_k \neq 0$, so the burden test has no systematic signal to detect — it is, in effect, testing the wrong (accidentally null) linear combination. SKAT, by contrast, is constructed as a quadratic form test of $H_0: \beta_1 = \dots = \beta_K = 0$ jointly, with a non-centrality parameter that accumulates $\sum_k \beta_k^2$ — a sum of squares, which remains strictly positive as long as at least one $\beta_k \neq 0$, regardless of the signs or any cancellation among the individual effects. ■

Worked Example 20.2 (Choosing burden vs. SKAT for a haemoglobinopathy modifier-gene panel in Nigeria)

A Nigerian sickle-cell research consortium sequences the *BCL11A* gene region (a well-established modifier of fetal haemoglobin, and hence of sickle-cell disease severity) in 2,000 patients, identifying 9 rare variants in the gene. Prior functional evidence suggests 6 of these variants increase fetal haemoglobin (protective, reducing disease severity) while 3 appear to decrease it (deleterious), with roughly comparable average effect magnitudes in each direction. By Theorem 20.1, a simple burden test collapsing all 9 variants into one score would have severely compromised power, since the protective and deleterious effects substantially cancel in the aggregated burden score ($\sum_k \beta_k$ close to zero even though several β_k are individually large) — the consortium should

A Nigerian sickle-cell research consortium sequences the *BCL11A* gene region (a well-established modifier of fetal haemoglobin, and hence of sickle-cell disease severity) in 2,000 patients, identifying 9 rare variants in the gene. Prior functional evidence suggests 6 of these variants increase fetal haemoglobin (protective, reducing disease severity) while 3 appear to decrease it (deleterious), with roughly comparable average effect magnitudes in each direction. By Theorem 20.1, a simple burden test collapsing all 9 variants into one score would have severely compromised power, since the protective and deleterious effects substantially cancel in the aggregated burden score ($\sum_k \beta_k$ close to zero even though several β_k are individually large) — the consortium should

20.5: Simulation as a Tool for Solving Genetics Problems

Throughout this book, closed-form analytic results have been available for many core quantities — Hardy–Weinberg proportions (Chapter 3), the recombination-fraction MLE (Chapter 8), the TDT statistic’s asymptotic distribution (Chapter 13). For many realistic, complicated genetic-epidemiology questions, however, no closed-form solution exists, or the small-sample behaviour of a statistic departs materially from its asymptotic theory — and **simulation** becomes the indispensable computational tool for answering the question anyway. Genetic simulation serves at least three distinct purposes, each illustrated computationally in Section 20.7:

1. **Generating realistic genetic data for method development and teaching**, when real sequence data are unavailable, too small, or too sensitive to share, by simulating genealogies and mutations under a specified population-genetic model (the **coalescent**, first developed in Chapter 16, or its **forward-in-time** alternative, described below).

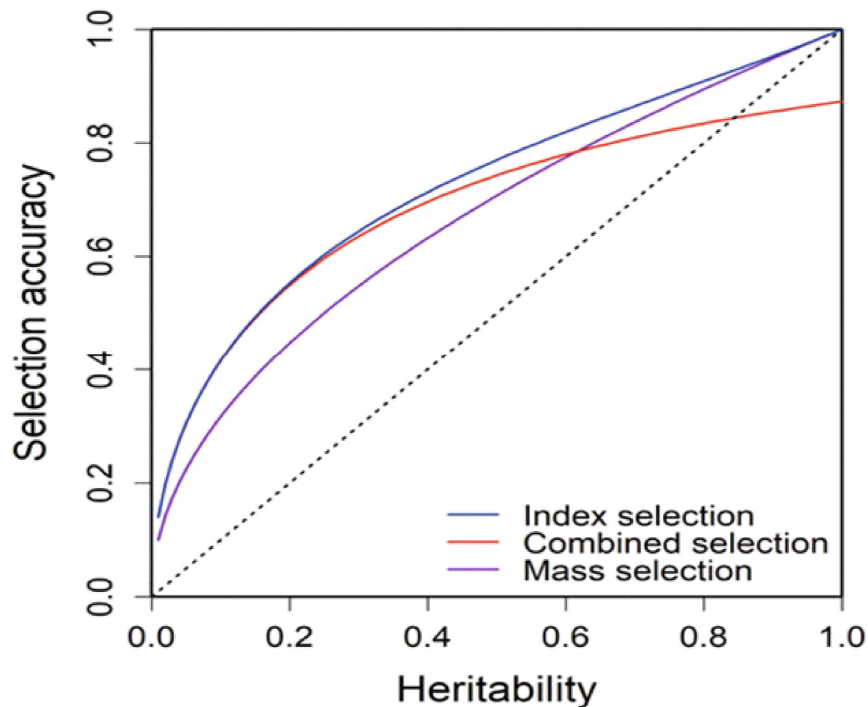
2. **Power and sample-size calculation:** simulating many replicate datasets under a specified true effect size and study design, and measuring the proportion of replicates in which the test of interest achieves statistical significance, gives an empirical power estimate for study designs (e.g., a rare-variant burden test, or a TDT study) too complex for a simple analytic power formula.
3. **Assessing a statistic's null distribution and calibration,** by simulating data under the null hypothesis many times and checking whether a test's p-values are uniformly distributed (i.e., that the test's nominal type-I error rate is actually achieved) — precisely the exercise suggested in Chapter 13's practice questions for the TDT, and a standard due-diligence step before trusting any new or complex statistical test.

20.6: Two Simulation Paradigms: Coalescent and Forward-in-Time

Chapter 16 introduced the **coalescent**, which simulates genealogies *backward in time* from a present-day sample to their most recent common ancestor(s), and is highly computationally efficient because it only ever generates the genealogical history of the sampled individuals, not the entire population. Modern coalescent simulators (msprime in Python, and the closely related scrm/ms lineage of programs, accessible from R via the coala package) extend the basic model of Chapter 16 to simulate full chromosome-length sequences, complete with recombination, variable population size through time, and multiple populations exchanging migrants — directly generating realistic genotype or haplotype data for exactly the kind of admixture and population-substructure exercises of Chapter 14, or the LD patterns of Chapter 15, under a fully specified and reproducible demographic history.

Forward-in-time simulation (implemented in software such as SLiM), by contrast, simulates the entire population, generation by generation, moving *forward* in time from a specified ancestral population, explicitly modelling selection, changing population size, and complex mating structures acting on every individual in the population at every generation. Forward simulation is far more computationally expensive than the coalescent for neutral (non-selected) scenarios, but is required whenever natural selection, complex non-neutral demographic scenarios, or explicit pedigree structures (extending Chapter 6) must be modelled with full realism.

A worked application: simulation in genomic prediction. Simulation is not only used to generate data for testing statistical methods (Section 20.5) — it is also used routinely to evaluate and compare entire prediction strategies before they are deployed on real data. In **genomic selection** (an application originating in animal and plant breeding but now also used for human polygenic risk scores, Chapter 18), a common question is: how accurately can an individual's genetic value for a quantitative trait (Chapter 9) be predicted from their genome-wide genotypes, and how does this accuracy depend on the trait's heritability and on which prediction strategy is used?



Simulated selection (prediction) accuracy as a function of trait heritability, compared across three genomic prediction strategies — index selection, combined selection, and mass selection — illustrating how simulation is used to evaluate competing statistical strategies before real-world deployment.

Such a plot is generated by simulating, for each heritability value, thousands of individuals with known true genetic values, applying each candidate prediction method to noisy phenotypic and genotypic data generated from a specified genetic architecture, and measuring the correlation between predicted and true genetic value — exactly the simulation-based evaluation logic of Section 20.5, applied here to compare prediction strategies (as opposed to comparing test statistics) before committing to one in an expensive real breeding or clinical programme.

20.7: Computational Corner: Simulating Sequence Data and a Rare-Variant Power Calculation

Python (coalescent simulation with msprime).

pip install msprime —break-system-packages

import msprime

import numpy as np

Simulate a 1 Mb chromosome segment for 200 diploid individuals (400 haplotypes)

under a simple constant-size coalescent model ($N_e = 10,000$), with recombination and mutation

```
ts = msprime.sim_ancestry(
    samples=200, population_size=10_000,
    sequence_length=1_000_000,
    recombination_rate=1e-8, random_seed=20)
```

```
mts = msprime.sim_mutations(ts, rate=1e-8,
    random_seed=21)
```

```
geno = mts.genotype_matrix()          # variants x haplotypes
```

```
freqs = geno.mean(axis=1) / 1 # allele frequency per variant
(haplotype-based)
```

```
n_rare = np.sum((freqs > 0) & (freqs < 0.01))
```

```
print(f'Simulated {mts.num_sites} variants; {n_rare} are rare
(MAF < 1%)')
```

R (equivalent conceptual workflow via the coalescent theory of Chapter 16, using base R for a simplified multi-locus version, since scrm/coala require CRAN installation).

Simplified multi-site coalescent-style simulation of a gene region with recombination

(illustrative implementation of the same idea as msprime, for teaching purposes)

```
set.seed(20)
```

```
n_haps <-400; n_sites <-50; Ne <-10000; mu <-1e-8;
site_len_bp <-20000
```

```
simulate_site <-function(n_haps, Ne, mu, site_len_bp) {
# crude infinite-sites approximation: expected pairwise
diversity theta = 4*Ne*mu*site_len_bp
  theta <-4* Ne * mu * site_len_bp
  n_mut <-rpois(1, theta *sum(1/ (1:(n_haps -1))))
  freq <-rbeta(max(n_mut, 1), 1, n_haps -1) # allele-
frequency-spectrum-like draw
  freq[1:max(n_mut, 0)]
}
all_freqs <-unlist(lapply(1:n_sites, simulate_site, n_haps =
n_haps, Ne = Ne, mu = mu, site_len_bp = site_len_bp))
n_rare <-sum(all_freqs <0.01)
cat("Simulated", length(all_freqs), "variants;", n_rare, "are
rare (MAF < 1%)\n")
```

Python (simulation-based power calculation for a gene-based burden test).

```
import numpy as np
from scipy import stats

rng = np.random.default_rng(99)

def simulate_burden_power(n_subjects=1000,
n_rare_variants=20, maf=0.005,
                        odds_ratio=2.0, n_reps=500, alpha=0.05):
    rejections =0
    for _ in range(n_reps):
        carrier_status = rng.binomial(1, 1- (1-
maf)**n_rare_variants, n_subjects) # >=1 rare variant in
gene
        baseline_p =0.10
```

```

    p_disease = np.where(carrier_status == 1,
                          (baseline_p * odds_ratio) / (1 - baseline_p +
                          baseline_p * odds_ratio),
                          baseline_p)
    disease = rng.binomial(1, p_disease)
    table = np.array([[np.sum((carrier_status == 1) &
    (disease == 1)),
    np.sum((carrier_status == 1) & (disease == 0))],
    [np.sum((carrier_status == 0) & (disease == 1)),
    np.sum((carrier_status == 0) & (disease == 0))]])
    _, p_value, _, _ = stats.chi2_contingency(table,
    correction=False)
    if p_value < alpha:
        rejections += 1
    return rejections / n_reps

power = simulate_burden_power()
print(f"Estimated power: {power:.2f}")

```

Running the `msprime` script generates realistic coalescent-simulated haplotype data for a 1 Mb region and directly counts how many simulated variants fall below the conventional 1% rare-variant threshold of Section 20.3, illustrating in miniature why sequencing (rather than array genotyping) is required to observe the great majority of variants segregating in any real human population sample — most polymorphic sites in a sequenced sample are, in fact, rare. The burden-test power simulation demonstrates the second major use of simulation described in Section 20.5: by repeatedly simulating an entire gene-based rare-variant study under an assumed true odds ratio and re-running the same statistical test used in a real analysis, the proportion of simulated replicates reaching significance directly estimates the study design’s statistical power — a calculation with no simple closed-form formula once realistic features (aggregated rare

variants, unequal carrier frequencies) are included, and precisely the kind of problem that simulation, and simulation alone, is equipped to solve.

Theorem 20.2 (Burden Tests Lose Power Under Bidirectional Rare-Variant Effects)

Consider a gene region with K independent rare variants, each with a true individual effect β_k on a quantitative trait, and let the burden test statistic be based on the simple sum of an individual's minor allele counts across all K variants (Section 20.4). If the true effects β_k are not all of the same sign, the non-centrality parameter of the burden test can be made arbitrarily small (even when individual variants have large true effects), whereas the non-centrality parameter of a variance-component test remains strictly positive.

Proof. The burden test's power is governed by its non-centrality parameter, which is proportional to $(\sum_{k=1}^K \beta_k)^2$ under standard regularity assumptions (the burden statistic estimates a single aggregated effect equal to the sum of the true per-variant effects weighted by allele frequency, and its test statistic's expectation under the alternative scales with this sum). If the β_k take both positive and negative values, the terms in the sum $\sum_k \beta_k$ can partially or fully cancel; in particular, for any large individual effects $|\beta_k|$, one can choose signs such that $\sum_k \beta_k = 0$ exactly (e.g., pairing equal-magnitude effects of opposite sign), driving the burden test's non-centrality parameter to zero and its power to the nominal significance level, regardless of how large the individual $|\beta_k|$ are. In contrast, a variance-component test (such as SKAT) is constructed from a quadratic form in the β_k (proportional to $\sum_k \beta_k^2$ in expectation), which is strictly positive whenever at least one β_k is non-zero, irrespective of the signs of the individual effects. This proves

that no configuration of non-zero, opposite-signed effects can drive the variance-component test's non-centrality parameter to zero, formally establishing the burden test's vulnerability to effect-direction heterogeneity that the chapter's simulation-based discussion (Section 20.4) illustrates numerically.

20.8: Chapter Summary

- High-throughput sequencing reads DNA directly and can detect rare and novel variants invisible to genotyping arrays and beyond the reach of standard imputation; whole-genome and whole-exome sequencing are the two dominant strategies, with results reported in Variant Call Format (VCF).
- The missing-heritability problem motivates the common-disease/rare-variant (CDRV) hypothesis, complementary to the common-disease/common-variant hypothesis underlying standard GWAS.
- Because individual rare variants have very low power for single-marker tests, rare-variant association methods aggregate variants within a gene using burden tests (assuming a consistent direction of effect) or variance-component tests such as SKAT (robust to mixed-direction effects).
- Simulation — via the backward-in-time coalescent (Chapter 16) or forward-in-time, individual-based models — is indispensable wherever no closed-form analytic result exists: for generating realistic genetic data, for empirical power and sample-size calculation, and for checking a statistical test's null calibration.
- msprime (Python) is the modern standard tool for coalescent simulation of full sequence data with recombination and demography; forward-in-time simulators such as SLiM are required when selection or complex non-neutral scenarios must be modelled explicitly.

Mini-Project

Mini-Project 20: Simulating Rare-Variant Data and Comparing Burden vs. Variance-Component Tests

Simulate rare-variant sequence data under two different genetic architectures and compare the power of burden and variance-component tests.

- Step 1:** Simulate a gene region with 20 rare variants in 1,000 cases and 1,000 controls, first under a scenario where all causal variants increase risk in the same direction.
- Step 2:** Apply a burden test (Section 20.4) and a variance-component test to this scenario and compare power.
- Step 3:** Repeat the simulation under a scenario where causal variants have effects in both directions (some protective, some risk-increasing).
- Step 4:** Apply both tests again and compare how their relative power changes between the two scenarios.
- Step 5:** Write a short paragraph explaining, in terms of the tests' underlying assumptions, why their relative performance reverses between the two scenarios.

Deliverable: *Simulation code, a power comparison table across both scenarios, and a short written explanation of the underlying reason for the reversal.*

20.9: Practice Questions

- Q1.** Explain, in two or three sentences, why a genotyping array cannot detect a truly novel variant that has never before been observed in any human, while whole-genome sequencing can.
- Q2.** A gene harbours 15 rare variants; laboratory evidence suggests 10 increase disease risk and 5 decrease it. Would

a simple burden test or a SKAT-type variance-component test be expected to have greater power in this scenario? Justify your answer with reference to Section 20.4.

- Q3.** Using the `msprime` code of Section 20.7, increase the simulated sample size from 200 to 2,000 diploid individuals and re-run the simulation. Does the number of variants classified as rare ($\text{MAF} < 1\%$) increase, decrease, or stay roughly the same proportion of the total, and why?
- Q4.** Using the Python burden-test power simulation of Section 20.7, vary `odds_ratio` from 2.0 down to 1.2 and report how the estimated power changes. Explain the pattern you observe in terms of effect size and statistical power generally.
- Q5.** Explain briefly why the coalescent (backward-in-time) approach is generally far more computationally efficient than forward-in-time simulation for simulating neutral genetic variation in a sample from a single large population, and describe one genetic scenario for which forward-in-time simulation would nonetheless be necessary.
- Q6.** In the genomic-prediction simulation of Section 20.6, explain why prediction accuracy for all three selection strategies increases with the trait's heritability, using the variance-partitioning framework of Chapter 9.

Appendices
