Examinable assignment 4 - part 1

- Use the Urn model starting with 2 red balls of weight 1, and one black (mutation) ball of weight w.
- Draw balls until you have 10 non-black balls.
- If all non-black balls are the same color at the end, what is the posterior distribution of the weight of the black ball?
- If we observe exactly 2 non-black colors at the end, what is the posterior distribution of the weight of the black ball?
- Use a Uniform[0,20] prior for the weight of the black ball

Rejection method

Goal: Generate samples from f(), defined over a domain [a,b].

Suppose: We know $f(x) \leq K$ for all $x \in [a,b]$.

Use the following iterative scheme:

1. Sample $x \sim \text{Unif}[a,b]$
2. Accept x with probability $f(x)/K$.
3. Return to 1.

Results: independent samples from f().

General Rejection method

Goal: Generate samples from f(), defined over a domain [a,b].

Suppose:

a. We have an envelope density h() from which we can simulate.

b. There exists an $K<\infty$ such that $\sup_x f(x)/h(x) \leq K$

Use the following iterative scheme:

So $f(x) \leq Kh(x)$

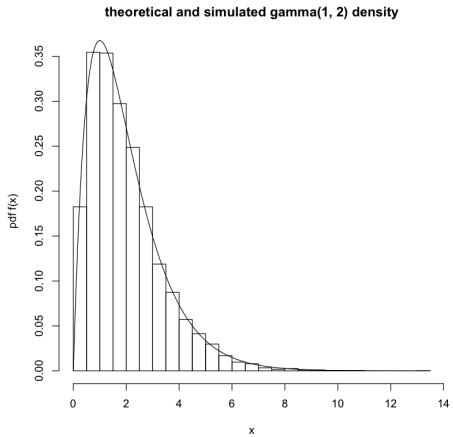
1. Simulate *x* from h().
2. Generate y from Unif[0,Kh(x)]. If y<f(x) then accept x.
3. Return to 1.

Results: independent samples from f().

Previously, our envelope was a rectangle.

Non-examinable assignment

- Write a rejection algorithm that uses the Uniform (rectangle) envelope, with K=1, to generate 10000 samples from a $\Gamma(\lambda,m) = \Gamma(1,2)$ distribution.
  - Plot your results.
  - How many iterations were needed to produce 1000 samples?



- Write a rejection algorithm that uses the exponential envelope $h(x)=\mu e^{-\mu x}$, where $\mu=\lambda/m$ to generate 10000 samples from a $\Gamma(1,2)$ distribution.
  - Plot your results.
  - How many iterations were needed to produce the samples?
- NB. R's built-in gamma density dgamma uses the parameters the other way around. [So you would plot dgamma(.,2,1)]

Pseudocode

```
gamma.sim <- function(lambda, m) {
  # sim a gamma(lambda, m) rv using rejection with an exp envelope
  # assumes m > 1 and lambda > 0
# generate f(x)=lambda^m*x^(m-1)*exp(-lambda*x)/gamma(m)     --- the gamma density at x
# generate h(x)=lambda/m*exp(-lambda/m*x)                    ---  the exponential density at x
# generate  k=m^m*exp(1-m)/gamma(m)                  --- in general. for m=2, λ=1 we have k=4/e
  while (TRUE) {              # keep sampling x's from h(x) and testing them until you accept one
    X <- -log(runif(1))*m/lambda            # generate an x from h, the exponential density
    # Generate y from Unif[0,Kh(x)]
    if (Y < f(X)) return(X)
  }
}

set.seed(1999)
n <- 10000        # number of replicates
g <- rep(0, n)      # create somewhere to keep the answers
for (i in 1:n) g[i] <- gamma.sim(1, 2)    # generate your 10000 gamma r.v.s

hist(g, breaks=20, freq=F, xlab="x", ylab="pdf f(x)",
  main="theoretical and simulated gamma(1, 2) density")
x <- seq(0, max(g), .1)
lines(x, dgamma(x, 2, 1))
```

You need to add the code for the uniform version!

PseudoCodeGamma.R on GitHub

Non-examinable lab assignment

- Code up the Gibbs sampler to simulate samples from:

$$(\theta, \mu) \sim N\left(0, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right)$$

- Compare this to performance of a standard Metropolis-Hastings MCMC algorithm, particularly for high values of ρ. [look at autocorrelation between θ values (or μ values)].
- R package: mvtnorm [dmvnorm(x, mean = rep(0, p), sigma = diag(p), log = FALSE)]
- If you are feeling industrious, implement an adaptive MCMC scheme to see how it compares with the Gibbs sampler in terms of efficiency

Lecture 10
Permutation tests and Importance Sampling

Permutation tests

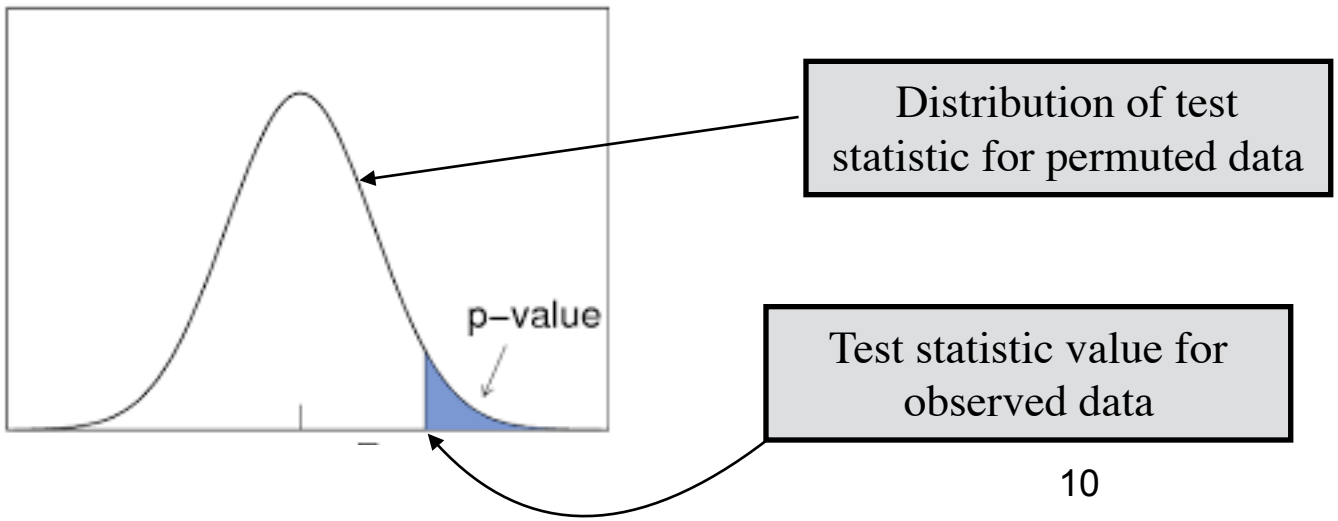
- A method for calculating the null distribution of a test statistic, when this cannot be done analytically.
- ‘No’ distributional assumptions.
- Can take a *very long time* to do!

Example

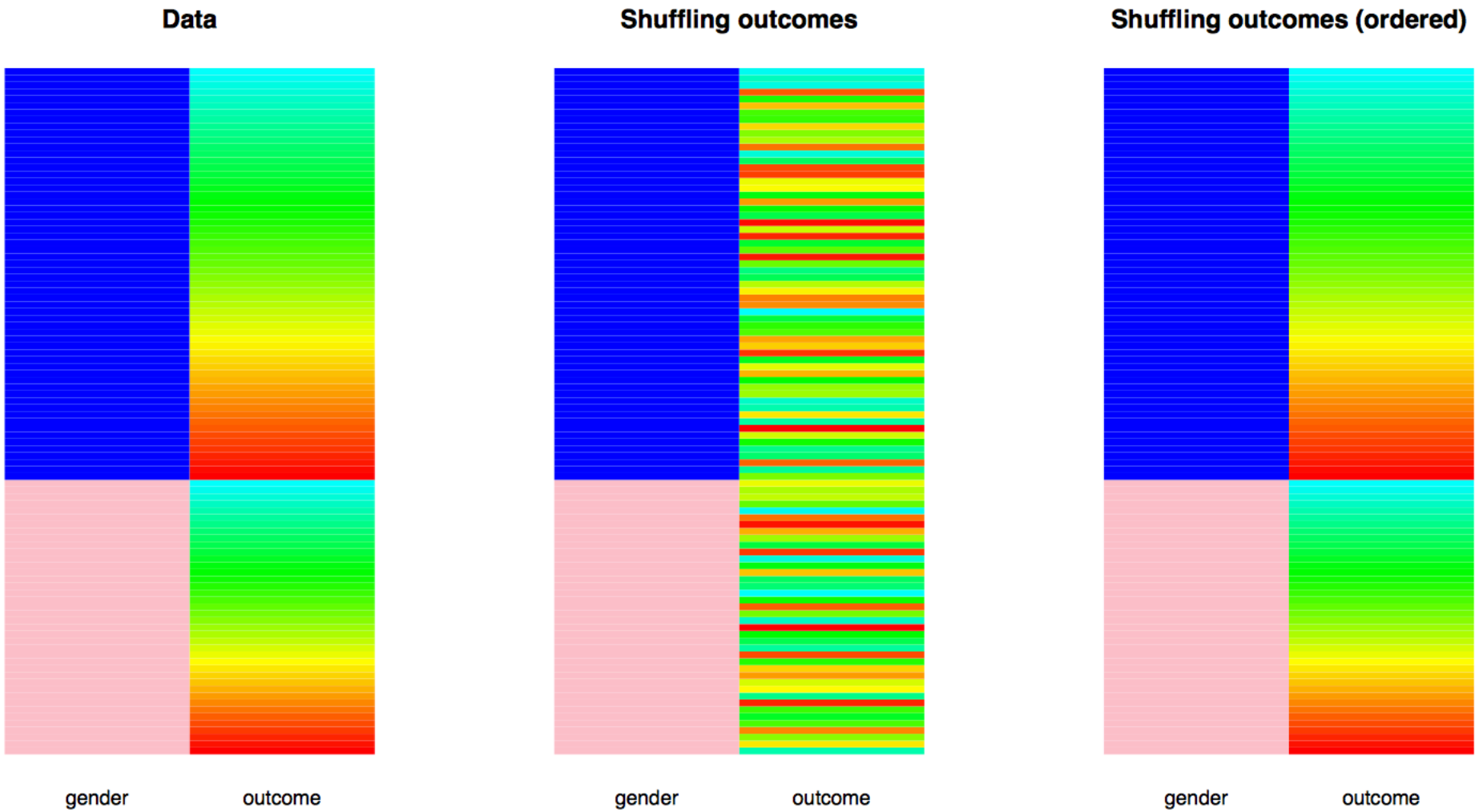
- Imagine we have a variable, X, measured on each individual from two groups ('carriers' and 'non-carriers').
- Suppose we want to test whether the mean value differs significantly between the two groups.
  - Null Hypothesis: no difference in means between two groups;
  - Alternative hypothesis: difference in means between two groups
- What test would you perform?
  - If X is normally distributed *within each group,* then you can perform a t-test (so the null distribution is theoretically derivable)
  - If not, then what?
    - t-test is actually robust to reasonable departures from normality, so in this instance you may not be too worried. But, in general, you have a problem.
    - You could resort to non-parametric tests (no distributional assumptions), but typically at loss of power.
    - Permutation tests often provide an alternative.

Permutation test - intuition

- Estimate the sampling distribution of the test statistic under the null hypothesis of no group effect, by constructing many ‘samples’ generated under the null.
- If the null hypothesis is true, changes to the exposure have no effect on the outcome. We can simulate this by randomly shuffling the exposures.
  - If the null hypothesis is true the shuffled data sets should look like the real data.
  - Otherwise they should look different from the real data.
- Calculate the value of your test statistic for the real data and for each of the permuted datasets.
- The ranking of the real test statistic among the permuted-data test statistics gives a p-value. (The p-value is the fraction of permuted datasets that have a more extreme test-statistic value.)

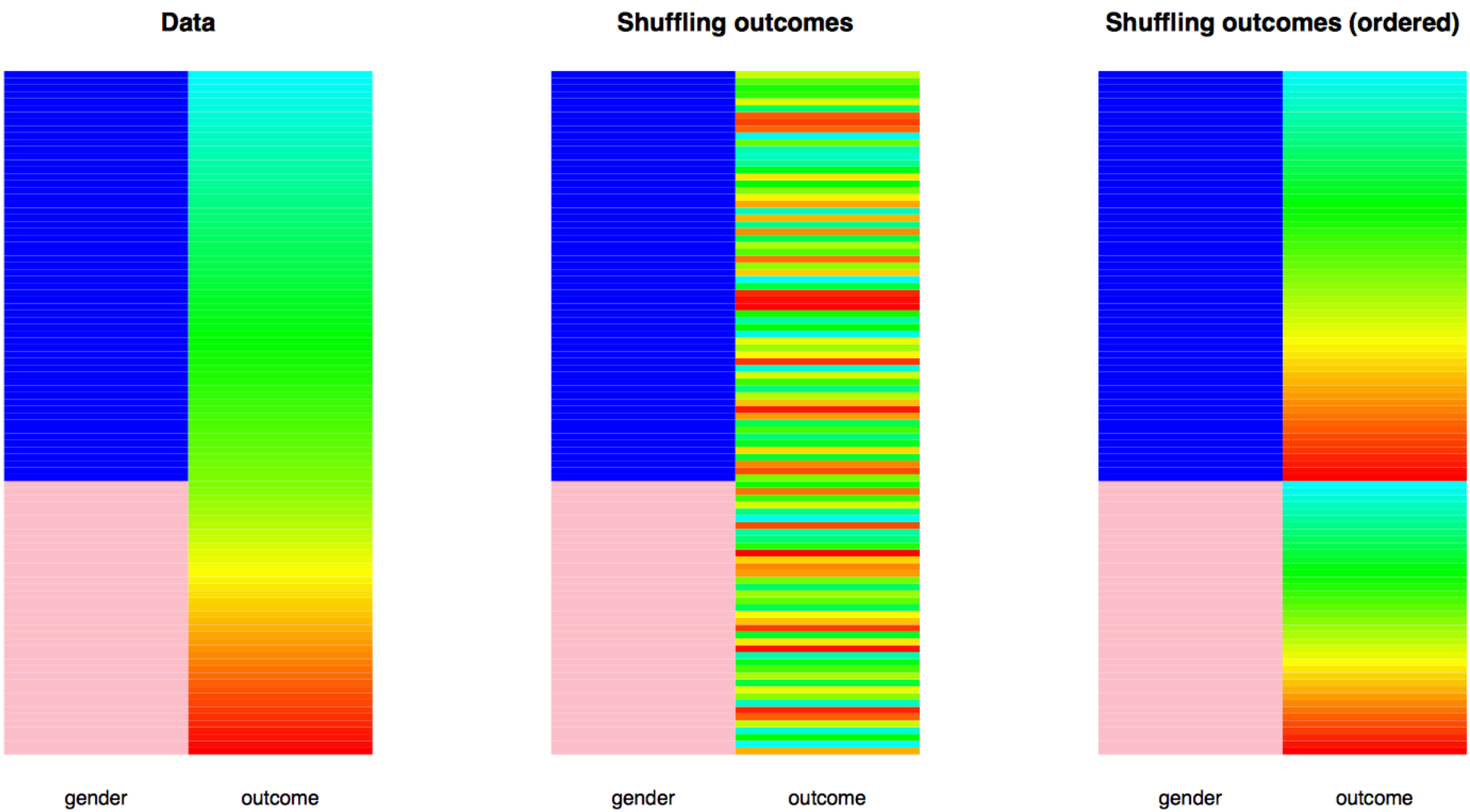


Example: Null is true



http://faculty.washington.edu/kenrice/sisg/SISG-08-06.pdf

Example: Null is false



http://faculty.washington.edu/kenrice/sisg/SISG-08-06.pdf

Permutation test - simple example

- Let's imagine we have 100 carriers and 200 non-carriers.
- We want to test for a difference in mean values between the groups.

- See Github repo Week10-PermatuationTests, file: SimplePerm.Rmd

Importance Sampling

- Richard Feynman: “If you think you understand quantum theory then you don’t understand quantum theory””

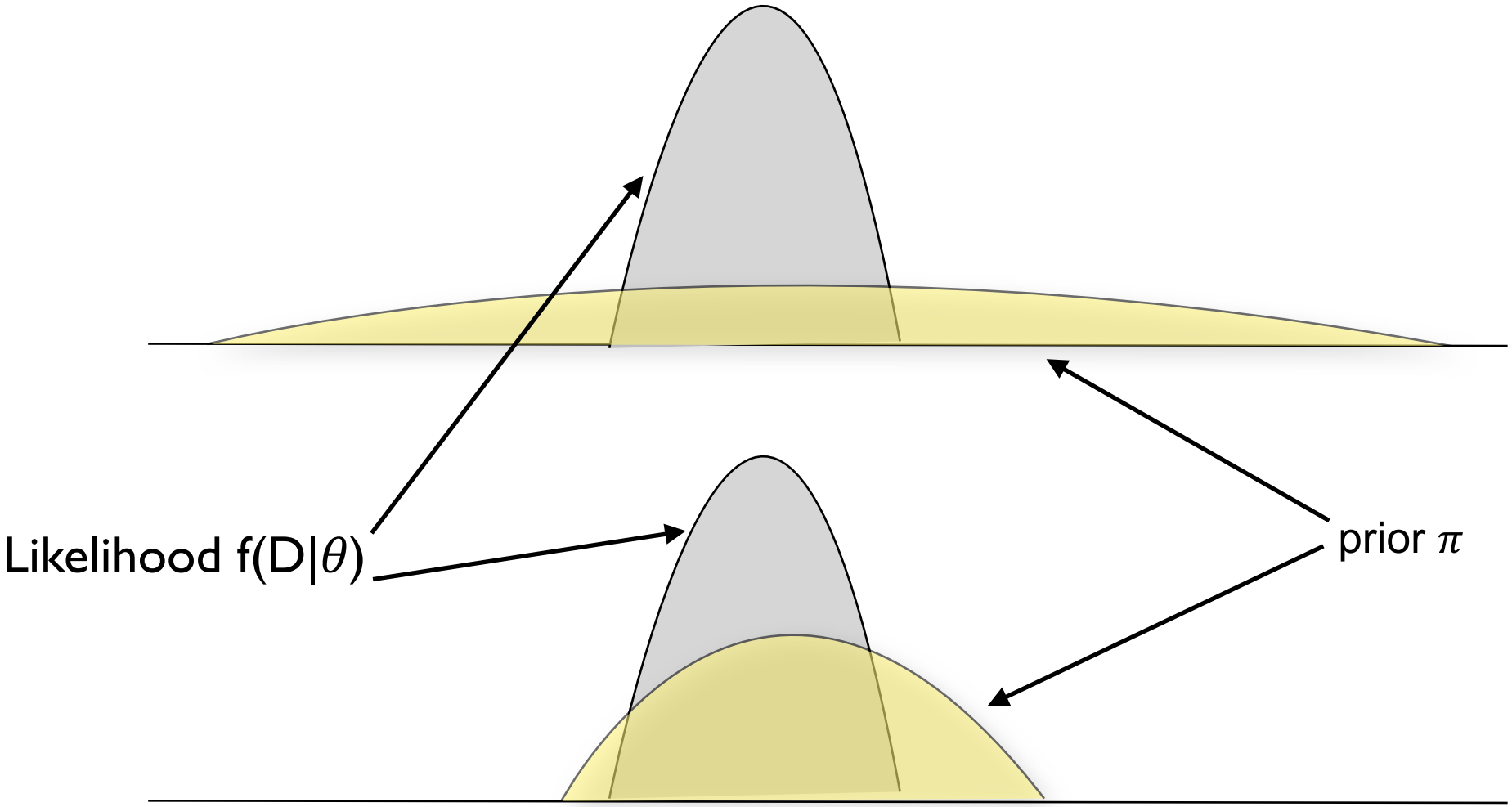
Rejection method - reminder

Suppose we have observed data D, and a model with parameter(s) θ that describes how the data got there. Do the following:

1. Generate parameter(s) θ from prior π.
2. Accept θ with probability P(D|θ)
3. Return to 1.

Result: independent samples from P(θ|D)

Efficiency



Change rejection method prior -> importance sampling

- In a Bayesian context:
  - Rejection method: sample from prior $\pi(\theta)$ and construct posterior;
  - Importance sampling: sample from 'Importance Sampling distribution' [ISD], $\xi(\theta)$, and construct posterior.

- Goal: Choose an ISD that is closer to the posterior (and/or likelihood) than the prior. [By doing so, we improve efficiency.]

Intuition

- Suppose we want to evaluate the expectation E(x) when X is a random variable with density f() defined over space $\mathcal{F}$.
- Suppose we have another density g(), defined over $\mathcal{G}$.

$$
\begin{aligned}
E[(x)] &= \int_{\mathcal{F}} x f(x) dx \\
&= \int_{\mathcal{G}} x \frac{f(x)}{g(x)} g(x) dx
\end{aligned}
$$

- Note that we need $\mathcal{F} \subseteq \mathcal{G}$.
- $f(x)/g(x)$ is sometimes referred to as the “Importance (sampling) weight”.

Example

- Suppose want to estimate the mean of a Normal(0,1) distribution truncated to the interval [0,1].
  - **Rejection method:**
    - Sample x from f(x)~N(0,1).
    - If x $\in$ [0,1] accept x; otherwise reject.
  - Estimate of mean = mean of accepted x's.
  - **Importance Sampler:**
    - Sample x from g(x)~Unif[0,1].
    - Estimate of mean = weighted average of all x values, where weights are given by $\phi(x) / \left[\int_0^1 \phi(z) d(z)\right]$
    - Here, g(x)=1 for all x; f(x)= $\phi(x) / \left[\int_0^1 \phi(z) d(z)\right]$
    - This method uses all datapoints (so is more efficient)

Weighted histograms

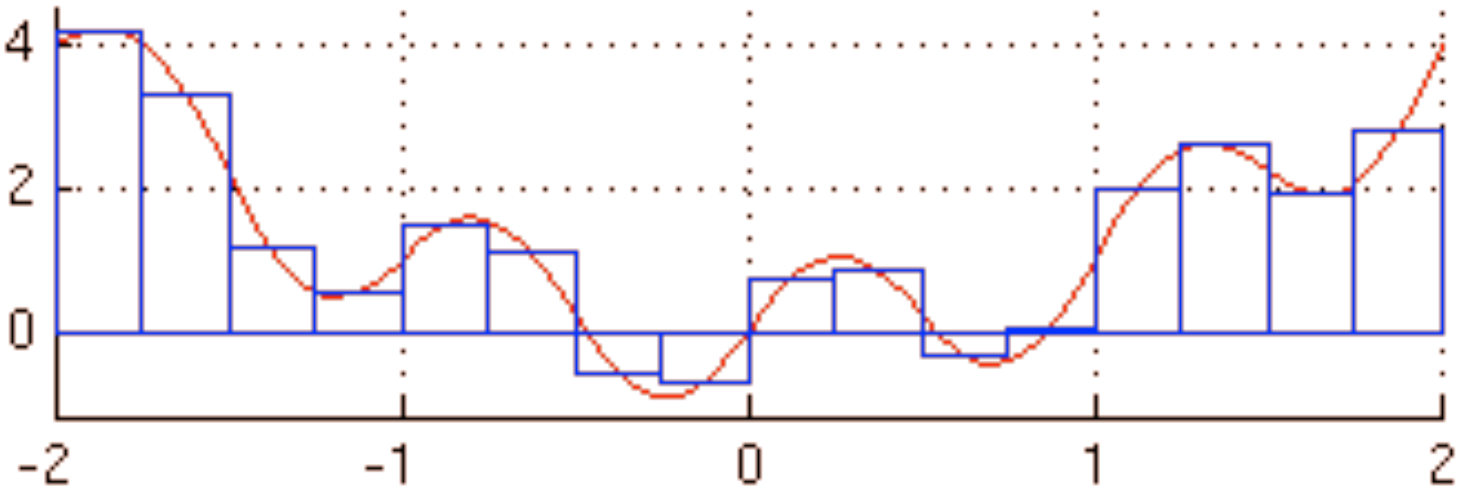
- Need to plot histograms in which the amount that each datapoint contributes is given by its Importance Sampling weight.

- See WeightedHistogramsExample.R in Week10-ImportanceSampling.

Importance Sampling Example

- Task: To evaluate an integral using Monte Carlo methods.
- Goal: To minimize the variance of the estimate
- ‘Importance sampling: an illustrative introduction’ - PH Borcherds, Eur. J. Phys. **21:**405–411, (2000).

Numerical integration



Riemann integration

- At each point, use the average height of the curve
- As number of points/rectangles increases, so does accuracy of approximation
- In the limit as #points -> infinity, the exact integral is obtained.
- Why not make points random?

Numerical integration - the problem

- Suppose we are integrating a function f over some space V.
- Suppose we require 100 points on the axis to accurately approximate the integral
  - If V is 1-dimensional we need 100 points
  - If V is 2-dimensional we need 100^2 points
  - If V is 10 dimensional we need 100^{10} points
- In statistical physics the number of variables is equal to the number of atoms, Monte Carlo methods are typically used.

Monte Carlo integration

- Suppose we are integrating f over V
- Sample n points (x_1, x_2, ... x_n) uniformly from V

$$\int_V f(x) = \mathrm{Vol}(V)\frac{1}{n}\sum_{i=1}^{n} f(x_i)$$

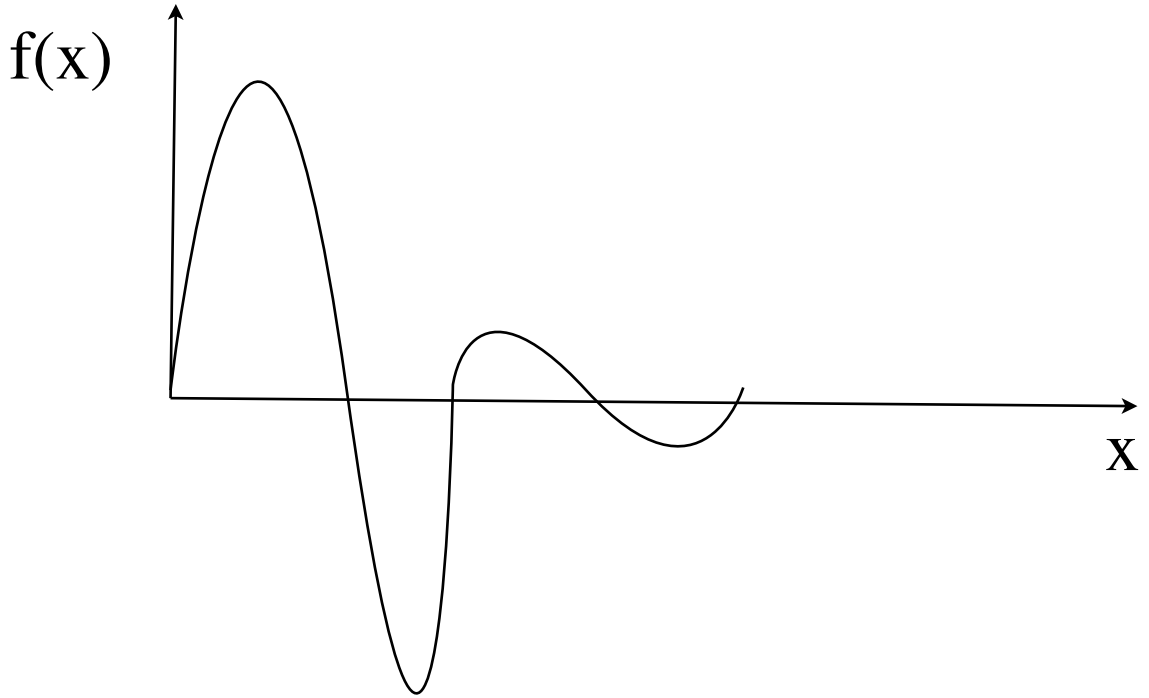
$$= \mathrm{Vol}(V) <f>$$

- where < *f* > is the mean of the sampled $f(x_i)$s

- That method is not particularly efficient - i.e. has relatively high variance (why?)
- So try an importance sampling method

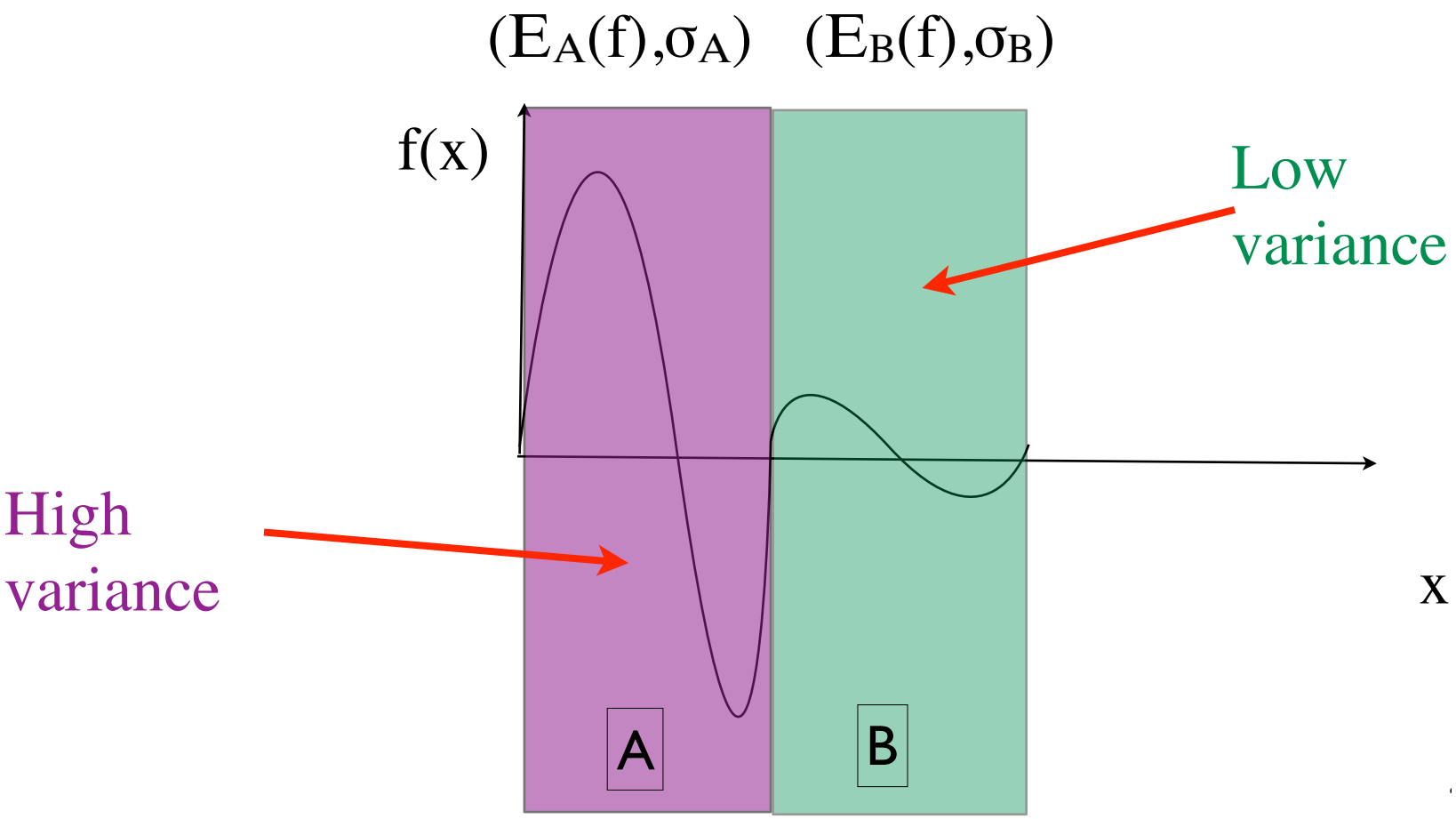
Miser Monte Carlo Integration

- Suppose we divide the domain into two intervals:



Miser Monte Carlo Integration

- Suppose we divide the domain into two intervals:



Suppose we use N_A samples from A, and N_B samples from B.

- Variance of combined estimate $E(f)=0.5E_A(f)+0.5E_B(f)$ is:

$$\mathrm{Var} = (\sigma_A^2/4N_A) + (\sigma_B^2/4N_B)$$

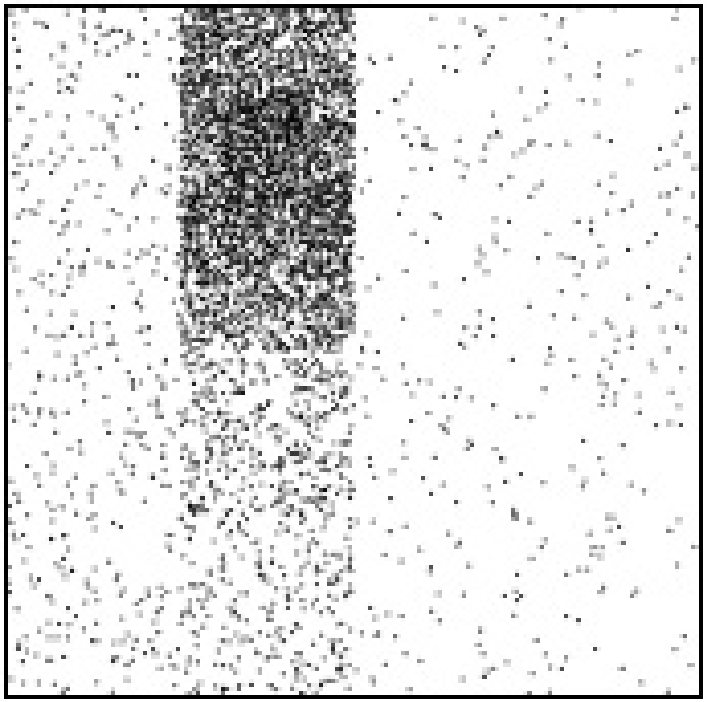
- which is minimized when

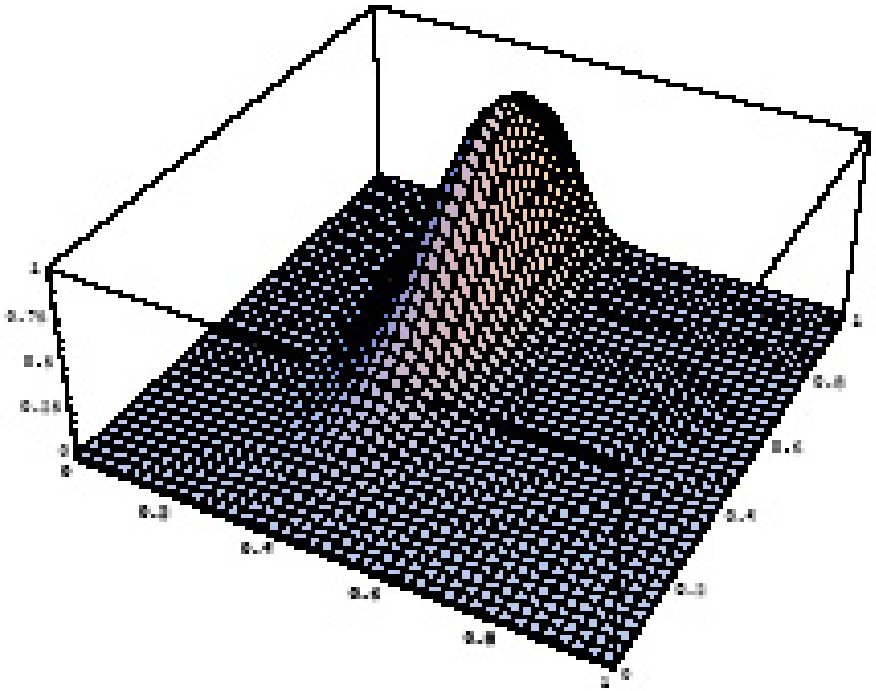
$$N_A/(N_A + N_B) = \sigma_A/(\sigma_A + \sigma_B)$$

So the number of points sampled in each region should be proportional to the variance of f over that region.

Miser MC Algorithm

- Iteratively bisects the domain of the function
- At each iteration: estimate the variance using a subset of samplings, and then, in each subset, assign a number of randomly scattered points proportional to the estimated variance of f in that subset.





Bayesian Importance Sampling

- Suppose we have parameter(s), θ, prior π, importance sampling distribution ξ, and observed data D. Iterate the following:
    1. Sample θ' from ξ.
    2. Accept θ' with probability P(D|θ'). Otherwise reject θ' and return to 1.
    3. Add a mass of $\pi(\theta')/\xi(\theta')$ to the posterior at θ'
    4. Go to 1.

Result: empirical estimate of posterior distribution P(θ|D).
The sampler will be more efficient than naive rejection.
How do we choose ξ?

NB. Rejection methods sample from prior $\pi(\theta)$, rather than ξ, but then construct posterior in the same way.

Bayesian Importance Sampling 2

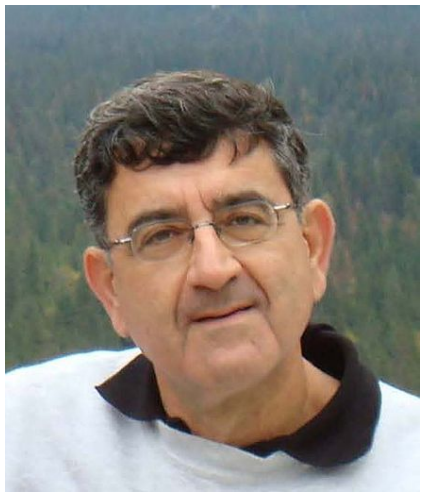
- Suppose we have parameter(s), θ, prior π, importance sampling distribution ξ, and observed data D. Iterate the following:
    1. Sample θ' from ξ, and simulate data D' using θ'.
    2. Accept θ' if D'=D. Otherwise reject θ' and return to 1.
    3. Add a mass of $\pi(\theta')/\xi(\theta')$ to the posterior at θ'
    4. Go to1.

Result: empirical estimate of posterior distribution $P(\theta|D)$.
The sampler will be more efficient than naive rejection.
How do we choose ξ?

Importance sampling application - Genome-wide Association Study p-values

A Fast Method for Computing High-Significance Disease Association in Large Population-Based Studies

Kimmel, G, Shamir, R, *The American Journal of Human Genetics*, 2006 vol. 79 pp. 481-492.



Genome-wide Association Studies

- Collect data on (very) many Single Nucleotide Polymorphisms [SNPs]
- Collect phenotype (Disease status - 0 or 1)
- Test for association between genotype and phenotype.

The growth of data

- Linkage studies (10s of polymorphic loci)
- Candidate genes (100s of loci)
- Expression arrays (1990s)
- SNPchips - Genome-wide Association Studies (2000s)
  - 1-10K of single nucleotide polymorphisms [SNPs]
  - 100Ks - mid-2000s
  - 1Ms - late 2000s
- Next-generation sequencing data - now [3GB of data]

The data

		Phenotypes
0-1-1-0-0-1-0-0-0-1-0-0-1	haplotype 1	1
0-0-0-0-1-0-1-1-0-1-0-1-1	haplotype 2	0
0-0-0-0-1-0-1-1-0-1-0-1-1	haplotype 3	1
1-0-1-1-1-1-0-0-1-0-1-0-0	haplotype 4	0
1-0-0-0-0-0-1-0-0-0-1-0-0	haplotype 5	1
1-0-0-0-0-0-1-0-0-0-1-0-0	haplotype 6	0

The test: Chi-squared

Markers, m		Phenotype, d
0-1-1-0-0-1-0-0-0-1-0-0-1	haplotype 1	1
0-0-0-0-1-0-1-1-0-1-0-1-1	haplotype 2	0
0-0-0-0-1-0-1-1-0-1-0-1-1	haplotype 3	1
1-0-1-1-1-1-0-0-1-0-1-0-0	haplotype 4	0
1-0-0-0-0-0-1-0-0-0-1-0-0	haplotype 5	1
1-0-0-0-0-0-1-0-0-0-1-0-0	haplotype 6	0

	d=0	d=1
m=0	1	2
m=1	2	1

$T_{ij}=\{k|m_k=i,d_k=j\},\ (k=1,...,n)$

$$S(T)=\sum_{i,j}\frac{(T_{i,j}-E(T_{i,j}))^2}{E(T_{i,j})}$$

where $E(T_{i,j})=\sum_a T_{i,a}\sum_a T_{a,j}/\sum_{a,b} T_{a,b}$

- If there are s possible types at a marker, and T is the contingency table for that marker and the (binary) phenotype, then S(T)~ χ^2_{s-1}
- Test every marker in this way
- Look for “small” p-values.
- Large number of markers -> need multiple comparison correction

Bonferroni correction:

- Desired alpha-level α=0.05 (say)
- 1 tests -> use p=0.05
- 2 tests -> use p=0.05/2
- ....
- N (*independent*) tests -> use p=0.05/N

- e.g. 500K tests -> use p=1e-8.
- Extremely conservative - “Number of independent tests”
- A range of variations on this theme exist.

Alternative Monte Carlo approach: permutation testing

- Define CC_{max} as the maximum value of the test statistic over all markers.
- Repeat the following many times:
  1. Permute (i.e., randomly re-order) the phenotype vector
  2. Calculate $CC_{max'}$ for the permuted data
- Define the *genome-wide p-value* for the original data as proportion of times that $CC_{max'} > CC_{max}$

Properties of permutation method

- 'No' distributional assumptions
- Allows for multiple testing
- Allows for non-independence of markers
- Simple, but can take a *very long time* to do!

Properties of permutation method here

- Expected time to search for an event with p-value p is 1/p permutations (definition of p-value!).
- Run time is O(nm/p)
- e.g. 1000cases/1000controls and 10000 loci -> 10^{13} basic computer operations.
- Very time-consuming for large data-sets

SNP interactions

- Many phenotypes are likely to be affected by interactions of multiple SNPs.
- Dependence of tests is even greater in such contexts
- e.g. When also including pairwise interactions, number of tests is $O(m)+O(m^2)$, so grows very quickly
- Problems even if only a few 100s of loci

Kimmel and Shamir

A Fast Method for Computing High-Significance Disease Association in Large Population-Based Studies

Kimmel, G, Shamir, R, *The American Journal of Human Genetics*, 2006 vol. 79 pp. 481-492.

Aim: To empirically estimate the genome-wide p-value

- Standard approach: Monte Carlo simulation, sampling uniformly from all possible permutations of disease status.
- K&S: Sample *only* from permutations that lead to a more extreme value of the test statistic. *So they are using an importance sampling distribution.*

Notation

- n = number of individuals
- m = number of markers
- Data = $(M,\mathbf{d})$ [M an n x m matrix, $\mathbf{d}$ a binary vector]
- Possible alleles: 1,....s.
- $M(i,j)=k$ if i^{th} indiv has type k at the j^{th} marker.

The data

		Phenotypes
0-1-1-0-0-1-0-0-0-1-0-0-1	haplotype 1	1
0-0-0-0-1-0-1-1-0-1-0-1-1	haplotype 2	0
0-0-0-0-1-0-1-1-0-1-0-1-1	haplotype 3	1
1-0-1-1-1-1-0-0-1-0-1-0-0	haplotype 4	0
1-0-0-0-0-0-1-0-0-0-1-0-0	haplotype 5	1
1-0-0-0-0-0-1-0-0-0-1-0-0	haplotype 6	0
M		***d***

- $S_j(\mathbf{d})$ is the Pearson chi-squared score for marker j and the disease vector $\mathbf{d}$ (calculated from the contingency table T).
- $S(\mathbf{d}) = \max_j S_j(\mathbf{d})$ [i.e. the highest score genome-wide]
- $\pi(\mathbf{d})=\mathbf{d}_i$: permutation of $\mathbf{d}$.
- Want: the p-value of $S(\mathbf{d})$. *i.e.* $P(S(\pi(\mathbf{d}))>S(\mathbf{d})|$ all permutations equally likely$)$

- Let ξ be the number of diseased indivs.
- There are $\binom{n}{\xi}$ possible distinct permutations.
- $\mathcal{F}$ is the space of possible permutations.
- *P(perm π)= 1/* $|\mathcal{F}|$ *= 1/*$\binom{n}{\xi}$
- $\mathcal{H}$ is the subset of $\mathcal{F}$ defined as

$$\mathcal{H} = \{\mathbf{d}_i \mid \mathbf{d}_i \in \mathcal{F},\ S(\mathbf{d}_i) \geq S(\mathbf{d})\}$$

(the set of perms that lead to a more extreme test-statistic than seen in the original data)

- So, we aim to estimate the value p-value: p = $|\mathcal{H}| / |\mathcal{F}|$

Outline

A standard Monte Carlo permutation/simulation approach would do the following:

1. Randomly permute the phenotypes
2. Calculate the p-value for each locus and record the smallest p-value you see.
3. Estimate the genome-wide p-value as the proportion of permuted data-sets that have a smaller p-value that that observed in the real data. (The 'empirical estimate' of the p-value)

The importance sampler - set-up

- The problem: with 1000 cases/controls, $|\mathcal{F}| \sim 10^{600}$
- Want to sample from $\mathcal{H}$ (permutations that result in a higher test stat value)
- Define $\mathcal{G} \subset \mathcal{H}$ where $\mathcal{G}$ has a different prob. measure $\Pr_{\mathcal{G}}$, (denoted by g()), such that:
  - Can easily sample from $\mathcal{G}$
  - $\Pr_G$ can be easily calculated
  - For each $\mathbf{d}_i \in \mathcal{H}$, $\Pr_{\mathcal{G}}(\mathbf{d}_i)>0$ (so we can still sample all $\mathbf{d}_i \in \mathcal{H}$)
  - If we draw N_R samples from $\mathcal{G}$ then:

$$p = \lim_{N_R \to \infty} \frac{1}{N_R} \sum_{i=1}^{N_g} \frac{f(\mathbf{d}_i)}{g(\mathbf{d}_i)}.$$

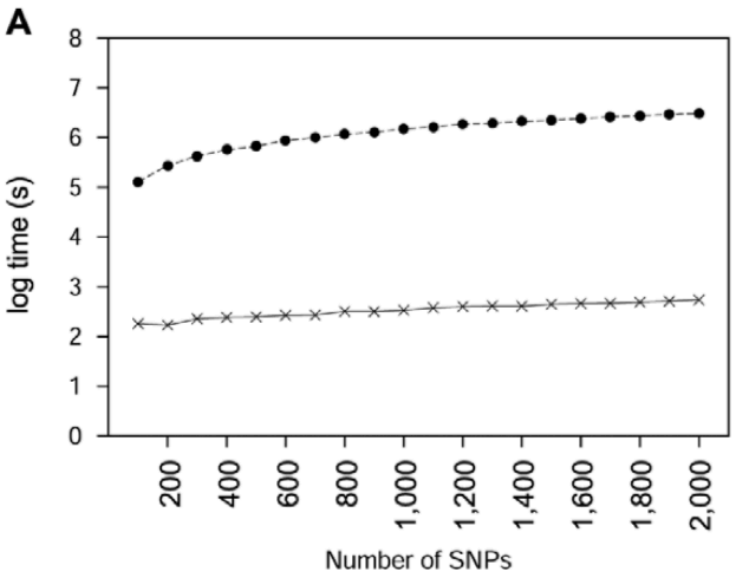
Kimmel and Shamir scheme

1. Sample a column j according to how often it will lead to more extreme values of the test statistic
2. Sample a contingency table T that can result from column j (after permutation of **d**) and that gives a value of the test statistic that is at least as large as was observed for the data
3. Randomly sample a permutation of the disease vector that results in getting T for that column.

Results - simulation study

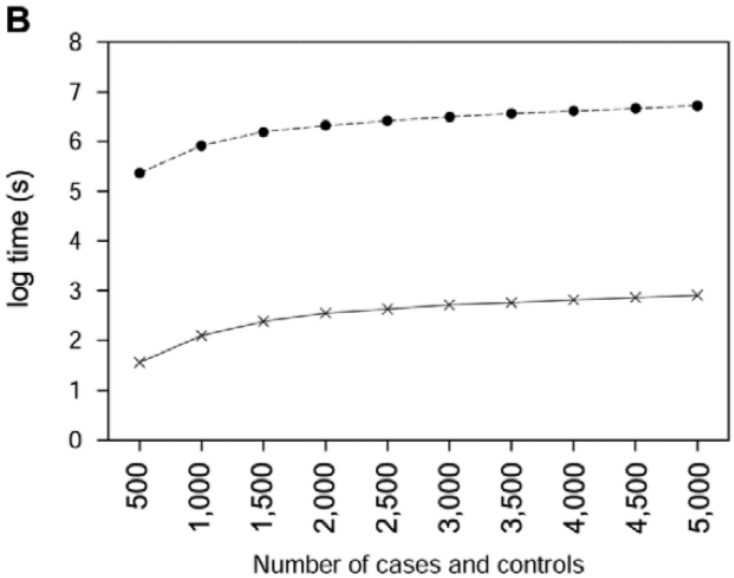
- Simulate data using the coalescent (Hudson's ms)
- Mutation rate = 2.5e-8; recombination rate = 1e-8
- Population size=10000, Minor allele freq. > 5%
- Disease SNP near middle, with freq. ~15%
- Penetrances of genotypes aa, aA, and AA are λ, $\lambda\gamma$, $\lambda\gamma^2$. Use γ=4, λ=0.024 (Zhang 2004)
- $\Rightarrow$ disease prevalence of 0.05 and disease-allele freq. of 0.15.
- Sample N cases and N controls.

Run speed

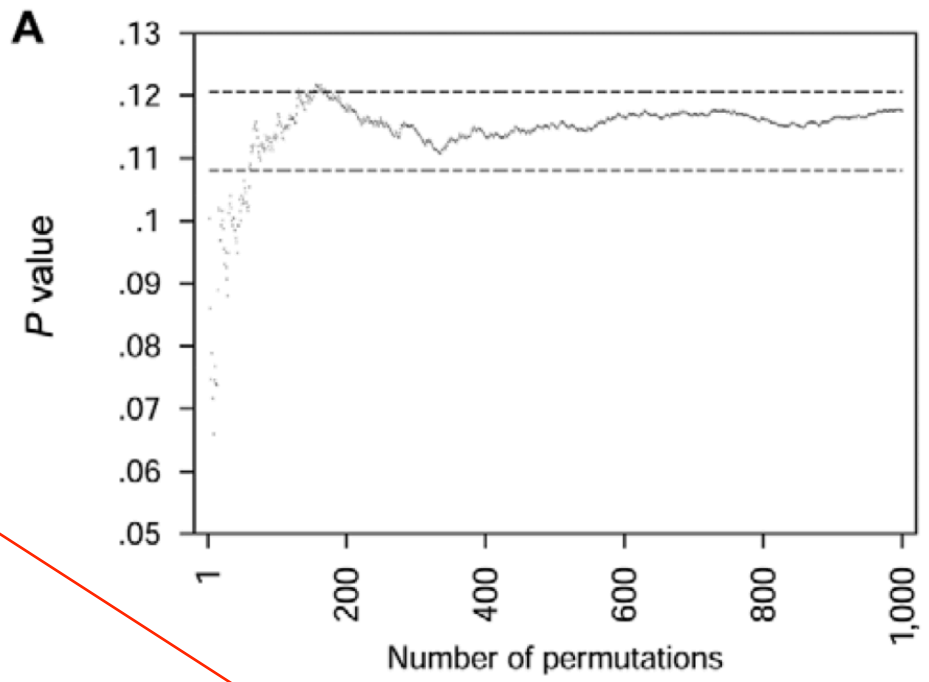


Standard permutation test

Their method



Convergence: 100 cases/controls (~1Mb region [3000 SNPs]) - 5 data sets

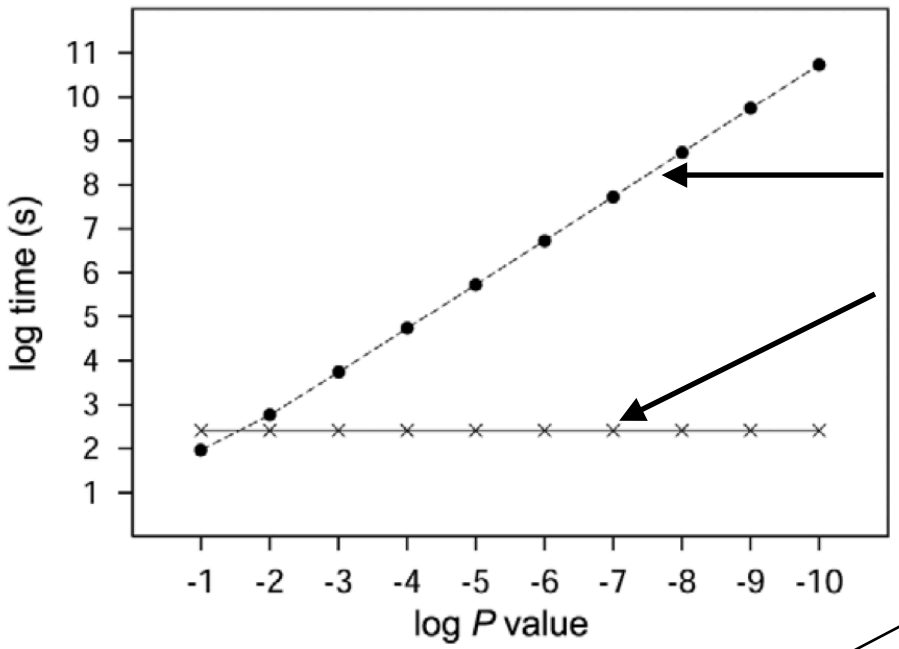


Their method

Standard permutation test

Figure 2. Convergence of RAT to the "true" P value. Each of the five figures represents a different experiment with 100 controls and 100 cases of simulated SNPs in a 1-Mb region (~3,000 SNPs), under the coalescent model. SPT P value was evaluated by applying 10,000 (A, D, and E) or 100,000 (B and C) permutations. The horizontal dashed lines correspond to the 95% CI of SPT P value. Each graph corresponds to the RAT P value.

Run-time vs. p-value



Standard permutation test

Their method

Standard permutation test

Figure 4. Running times of RAT and SPT at different *P* values. The data sets are simulated data under the coalescent model with recombination of a 1-Mb region (~3,300 SNPs) of 5,000 cases and 5,000 controls. To obtain different *P* values, the simulations were performed with different phenocopy rates (λ parameter) of the multiplicative disease model. × = RAT; circles = SPT. The *Y*-axis shows the logarithm (base 10) of the running time in seconds, and the *X*-axis shows the logarithm (base 10) of the *P* value.

Conclusions

- Method is 3-5 orders of magnitude faster than standard permutation testing, and works **VERY** well for small p-values.
- e.g. 1000 cases/controls, 10K SNPs, required accuracy 1e-6:
  - 30 days using standard methods
  - 2 minutes using RAT.
- e.g 5,000 cases/controls, 30K SNPs:
  - 4.6 years for standard methods
  - 24 minutes using RAT.
- Importance sampling is hard to write down

Bayesian Importance Sampling

- Suppose we have parameter(s), θ, prior π, and importance sampling distribution ξ, and observed data D. Iterate the following:
    1. Sample θ' from ξ.
    2. Accept θ' with probability P(D|θ'). Otherwise reject θ' and return to 1.
    3. Add a mass of $\pi(\theta)/\xi(\theta)$ to the posterior at θ'
    4. Rejection method: sample from prior $\pi(\theta)$ and construct posterior;

Result: empirical estimate of posterior distribution P(θ|D).
The sampler will be more efficient than naive rejection.
How do we choose ξ?

-

Bayesian Importance Sampling 2

- Suppose we have parameter(s), θ, prior π, and importance sampling distribution ξ, and observed data D. Iterate the following:
  1. Sample θ' from ξ, and simulate data D' using θ'.
  2. Accept θ' if D'=D. Otherwise reject θ' and return to 1.
  3. Add a mass of $\pi(\theta)/\xi(\theta)$ to the posterior at θ'
  4. Rejection method: sample from prior $\pi(\theta)$ and construct posterior;

Result: empirical estimate of posterior distribution P(θ|D).
The sampler will be more efficient than naive rejection.
How do we choose ξ?

Examinable assignment 4 - part 2

- Repeat the Urn model assignment to find the posterior distribution of the weight of the black ball given that we observe just one color in an urn containing **10 non-black balls.**

Weight of black ball (i.e., mutation rate)

- This time, sample θ' from ξ ~ exponential(λ), rather than a Uniform distribution.
- If you accept that value of θ' (because the simulated urn has just one color in it), then add a mass of $\pi(\theta')/\xi(\theta')$ to the posterior distribution of θ (whereas we used to add a mass of 1).
- If we truncate the exponential at 20 (say), this means:
  - $\pi(\theta') = 1/20$
  - $\xi(\theta') = \lambda e^{-\lambda\theta'}/(1-e^{-20\lambda})$
- Compare efficiency to a rejection method that just samples directly from a Uniform[0,20] distribution. (i.e. how many simulations do you need to generate 10000 acceptances; and how smooth does the histogram look?)

Examinable Assignment - Part 3

- Many tissues contain 2 (or more) cell-types.
- Investigator wants a way of testing whether each of 2 cell types in a given tissue is homogeneously distributed (in space).
- Your job, is to come up with such a test, **using Monte Carlo methods**.

PLOS ONE

OPEN ACCESS

Citation: Courcoubetis G, Ali S, Nuzhdin SV, Marjoram P, Haas S (2019) Threshold response to stochasticity in morphogenesis. PLoS ONE 14(1):

RESEARCH ARTICLE

Threshold response to stochasticity in morphogenesis

George Courcoubetis[1], Sammi Ali[2], Sergey V. Nuzhdin[2]*, Paul Marjoram[3], Stephan Haas[1]*

1 Department of Physics and Astronomy, University of Southern California, Los Angeles, California, United States of America, **2** Department of Molecular and Computational Biology, University of Southern California, Los Angeles, California, United States of America, **3** Department of Preventative Medicine, Keck School of Medicine of USC, Los Angeles, California, United States of America

* snuzhdin@usc.edu (SN); shaas@dornsife.usc.edu (SH)

Abstract

During development of biological organisms, multiple complex structures are formed. In many instances, these structures need to exhibit a high degree of order to be functional, although many of their constituents are intrinsically stochastic. Hence, it has been suggested that biological robustness ultimately must rely on complex gene regulatory networks and clean-up mechanisms. Here we explore developmental processes that have evolved inherent robustness against stochasticity. In the context of the Drosophila eye disc, multiple

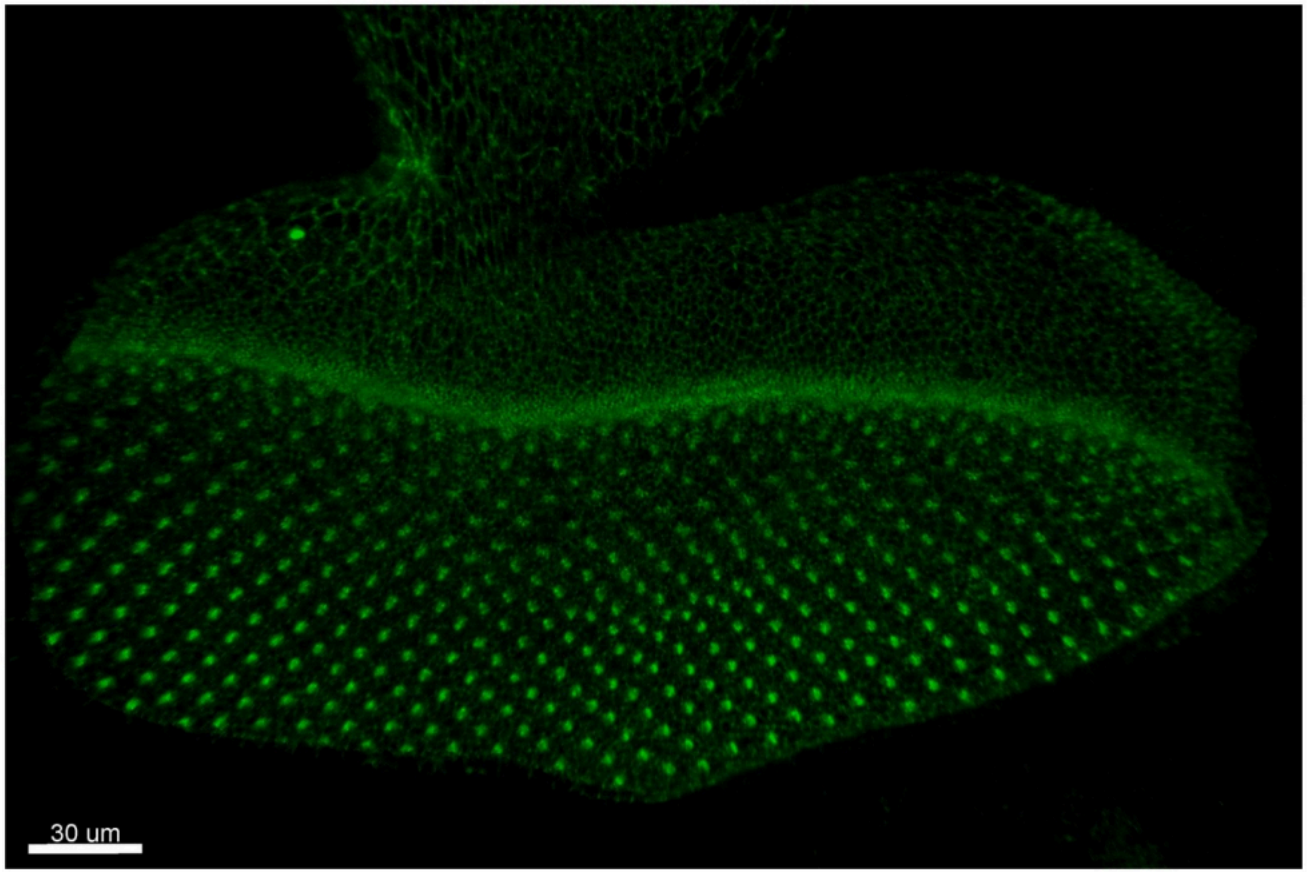
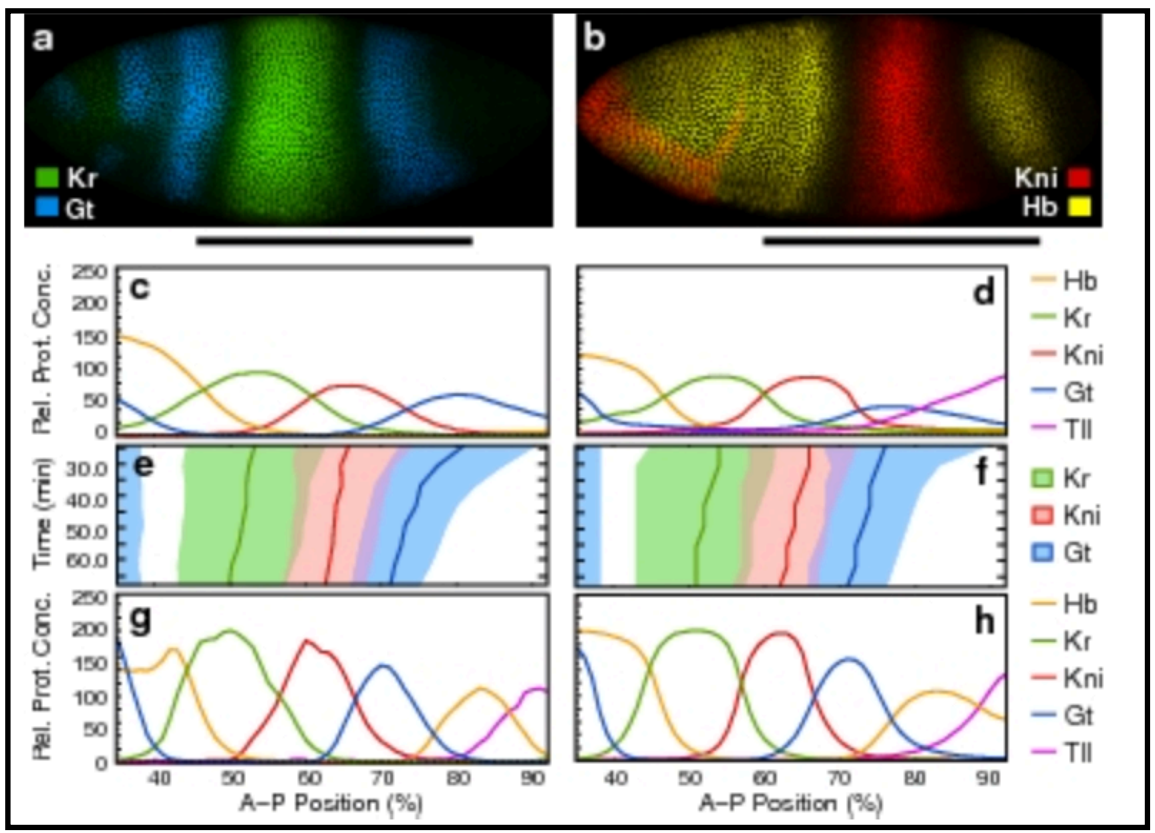


Fig 1. Experimental image of the positions of the R8 cells during morphogenesis. In this fluorescent microscopy image, one can see DE-Cadherin linked with GFP. This labels the top part of each cell shown as a ring. When multiple cells are in close proximity, they appear as one bright point, which in this case point the positions of the R8 cells.

https://doi.org/10.1371/journal.pone.0210088.g001

Examinable Assignment - Part 3



Drosophila gap genes

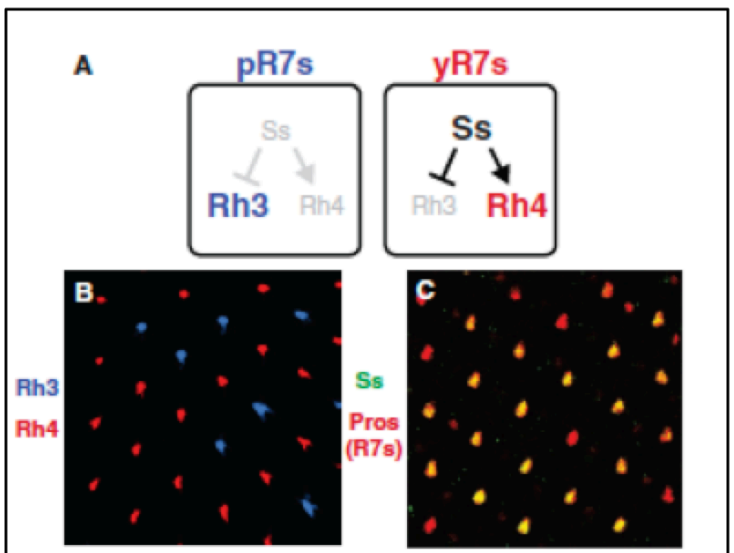


Figure 1: Reproduced from [10] (A) *ss* is absent from pR7s, allowing for Rh3 expression. *ss* is expressed in yR7s activating Rh4. (B) Stochastic distribution of ommatidial subtypes (C) *ss* is expressed in a random subset of R7s, *Pros* marks all R7s.

Drosophila eye cells

Color corresponds to gene expression

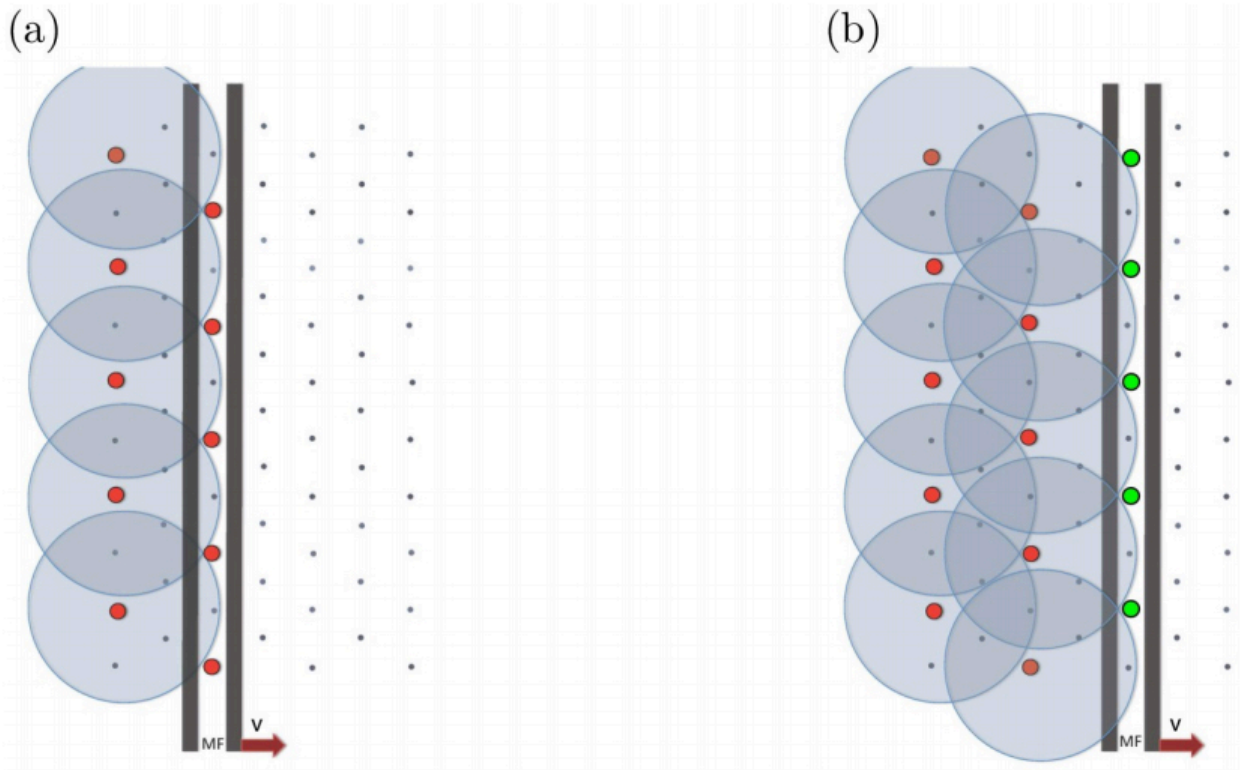


Fig 2. Visualization of the foundations of the R8 cell specification mechanism in the Drosophila eye disc. Simplified illustration of pattern formation mechanism in the Drosophila eye disc as a result of the competition between short-range inhibitor and long-range activator morphogens. Posterior region of the eye disc with an initial row of differentiated precursor cells, denoted by red dots. The morphogenetic furrow (MF), modeled by a plane wave front, moves to the right towards the anterior region. The gray circles represent the boundaries of regions affected by the short-range inhibitor, where the morphogenetic furrow (MF) will not initiate production of *atonal*. Therefore differentiation can only occur in those locations which are not affected by the short-range inhibitor, leading to the hexagonal supper-lattice of differentiated R8 cells, which is experimentally observed. This diagram does not incorporate cluster formation.

https://doi.org/10.1371/journal.pone.0210088.g002

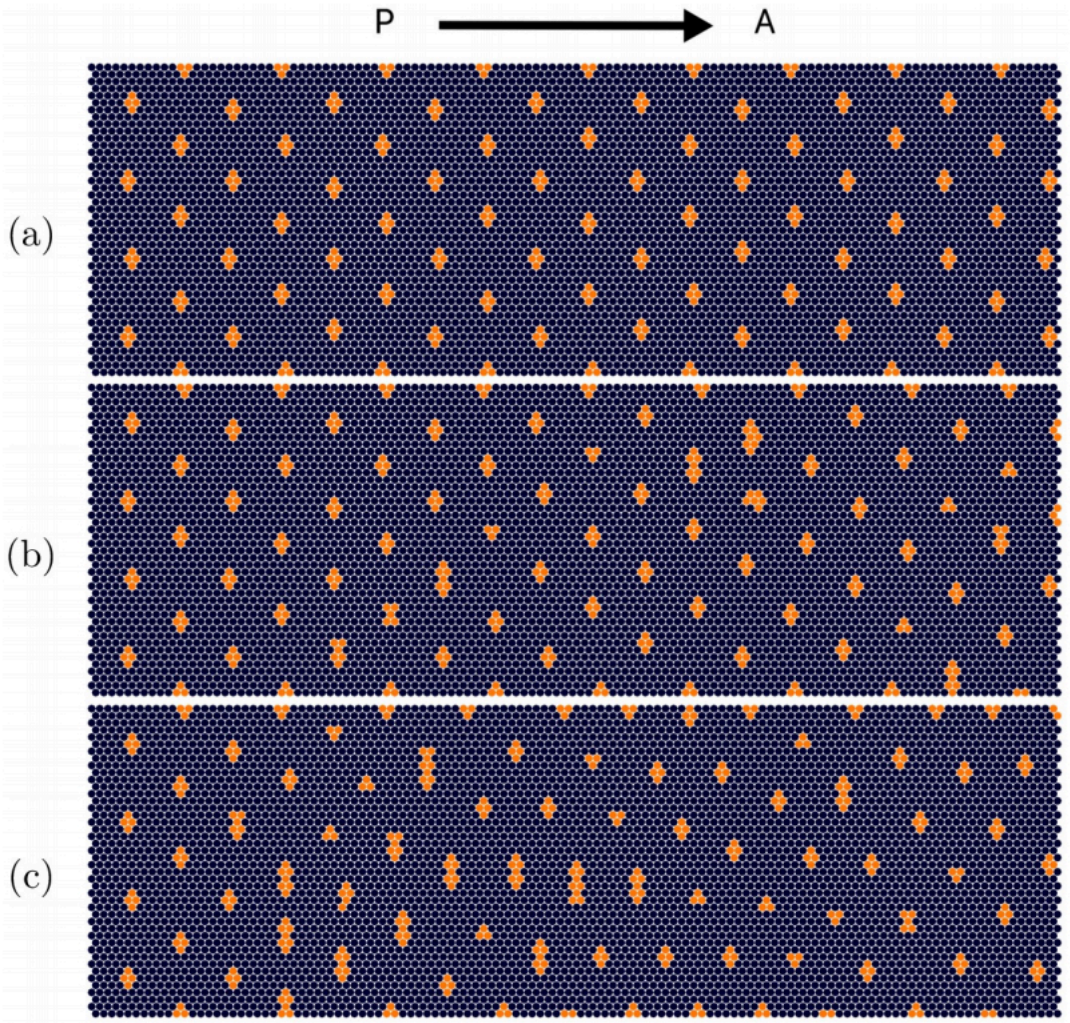
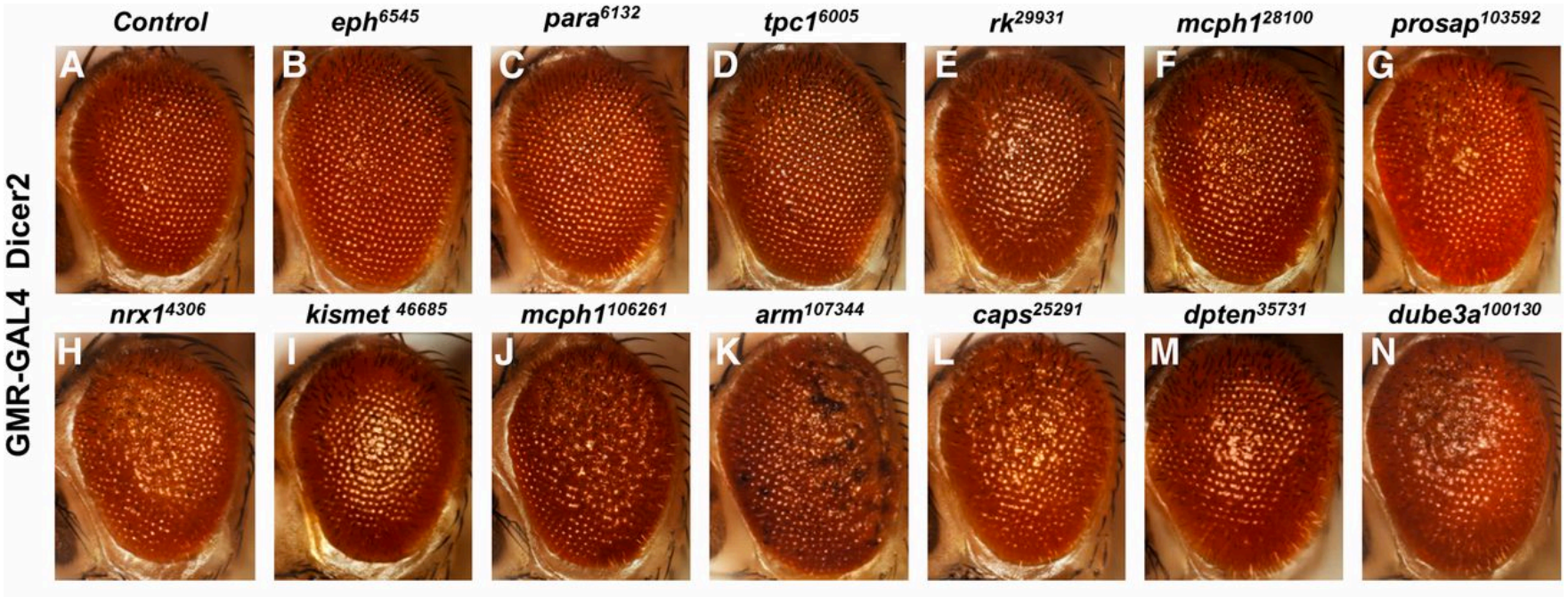


Fig 4. Pattern formation simulation results in the Drosophila eye disc, the morphogenetic furrow moves from left to right. Cluster positions and sizes of *ato* expressing are shown for increasing noise σ in the diffusivity of the morphogens D_u and D_s. (a)-(c) are the final patterns for $\sigma/\mu = 0$, $\sigma/\mu = 30\%$, and $\sigma/\mu = 40\%$ respectively. Here, μ is the mean of the corresponding normal distribution and the value that is used to generate the perfect pattern.

https://doi.org/10.1371/journal.pone.0210088.g004

Eye mutants



Quantitative Assessment of Eye Phenotypes for Functional Genetic Studies Using *Drosophila melanogaster*

Iyer et al.G3: GENES, GENOMES, GENETICS May 1, 2016 vol. 6 no. 5 1427-1437

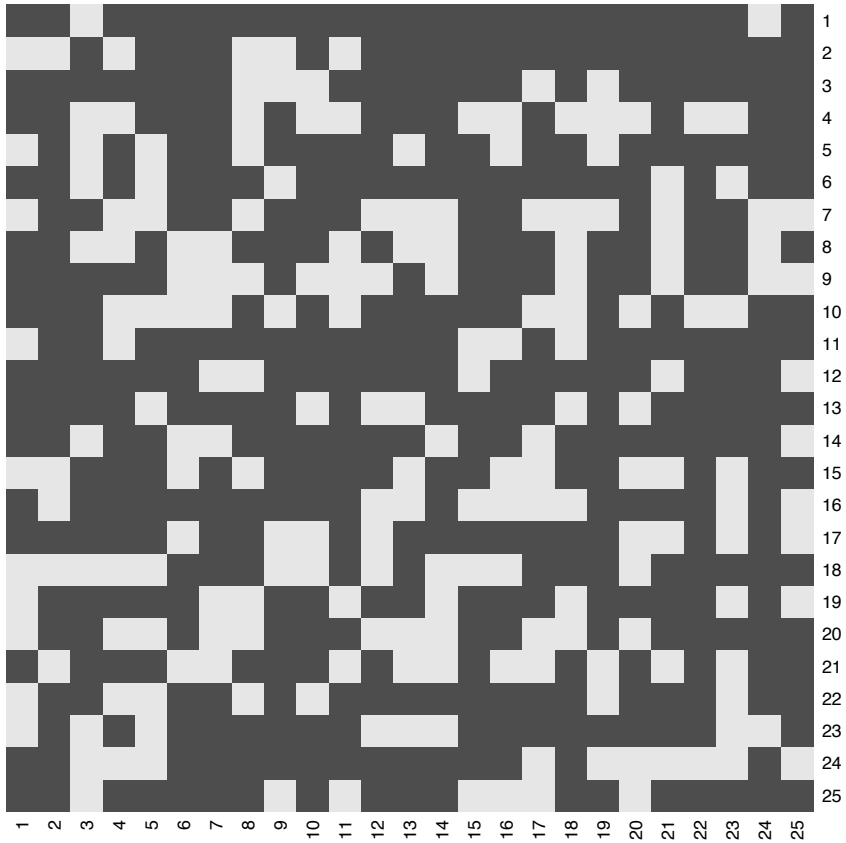
Examinable Assignment - Part 3

Schematically:

Assume cells fall on a 25x25 2D grid

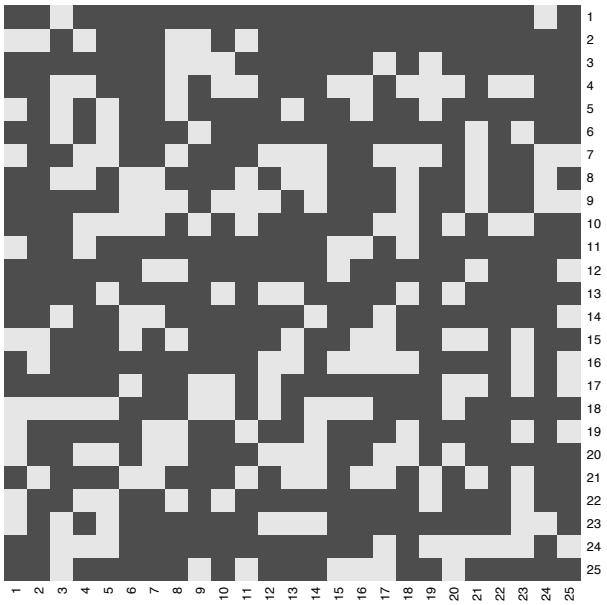
White = cell type 1

Black = cell type 2

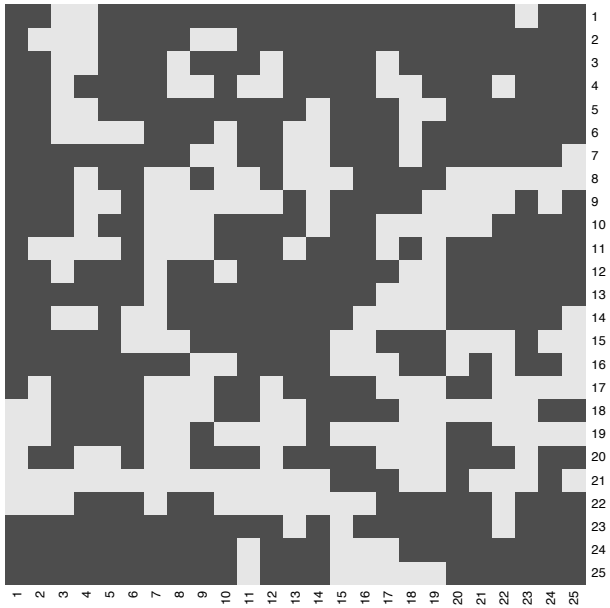


"Grid1.txt" on Blackboard

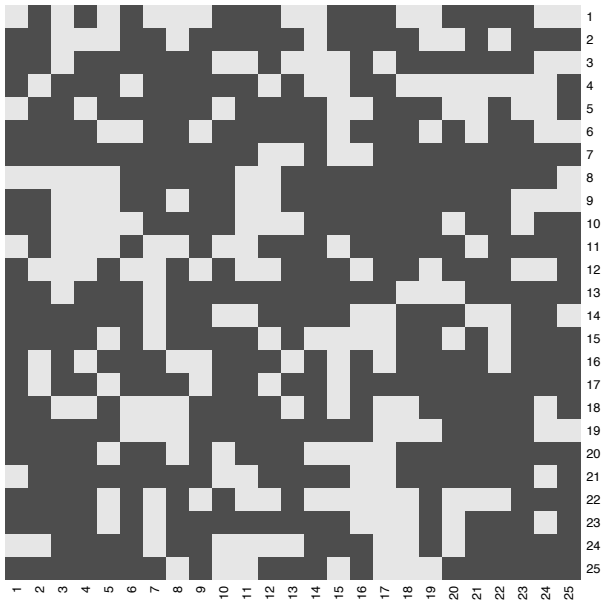
Examinable Assignment - Part 3



"Grid1.txt"



"Grid2.txt"



"Grid3.txt"

Examinable Assignment - Part 3

- Many tissues contain 2 (or more) cell-types.
- Investigator wants a way of testing whether each of 2 cell types in a given tissue is homogeneously distributed (in space). So:
  - H_0: cell types are homogeneously (randomly) arranged
  - H_1: cell types are not homogeneously arranged (i.e. they are clustered in some fashion).
- Your job is to come up with such a test, using Monte Carlo methods.
  1. Formulate your test.
  2. Apply it to each of Grid1, Grid2, Grid3 and determine a p-value for rejecting the H_0 in favor of H_1.
  3. Write-up your test (i.e. Methods) and your Results as an Markdown file.

Assignment 4 is due on April 27th (3 weeks from today)

END