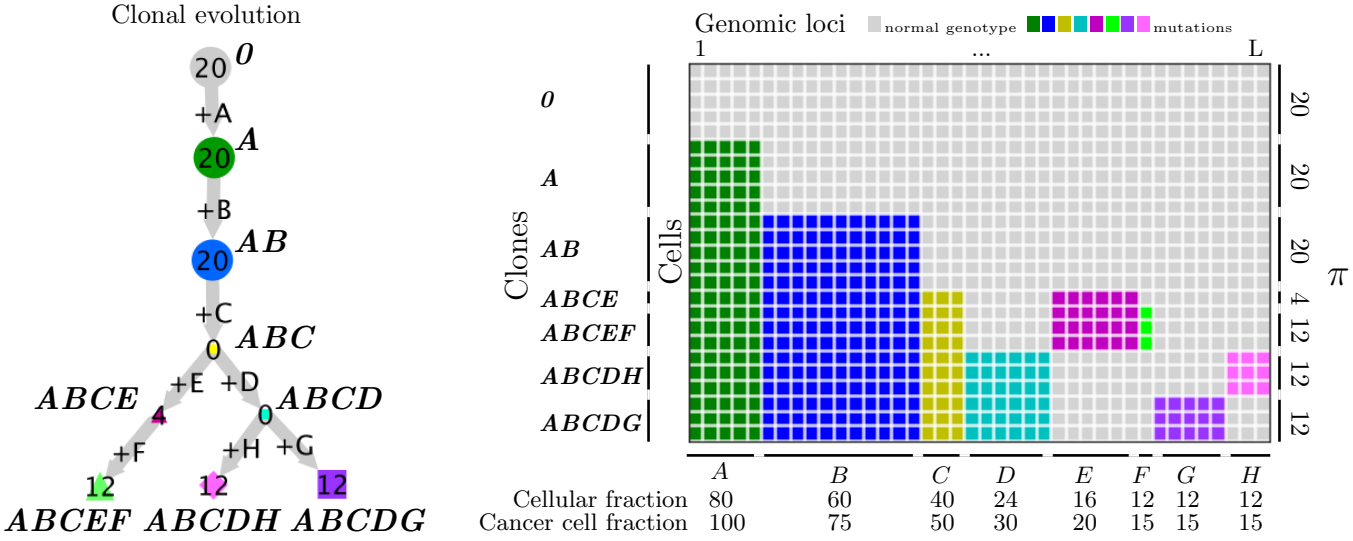
Cancer Evolution 1

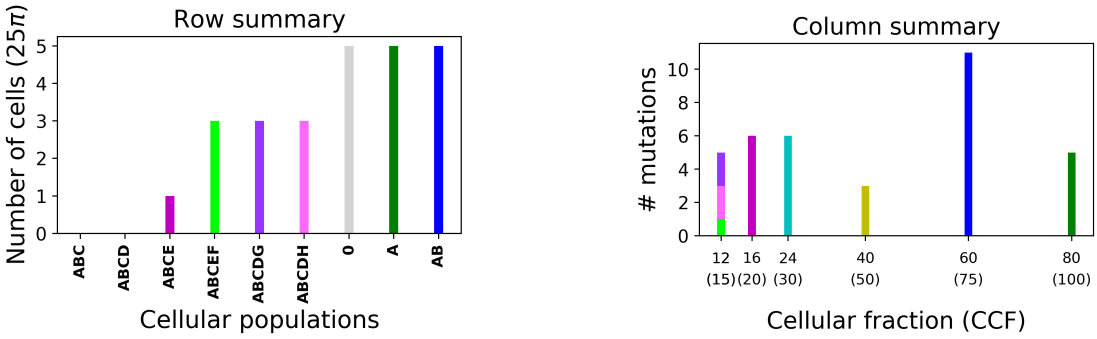
USN: 303039534

1

Figures 1b, 1c and 1d were generated using python and matplotlib. Figure 1a was generated using the python networkx module and visualized using cytoscape. All code is given in the appendix, and additional annotations were added using the LaTeX overpic package.



(a) (b)



(c) (d)

Figure 1: **(a)** Graph illustrating the evolution of the tumor. In constructing the graph, it has been assumed that no set of mutations has developed twice independently. Nodes represent clones and an edge between two nodes indicates that the daughter node has been derived from the parent node. Edges are labelled with the set of mutations giving rise to a particular clone, with labels A-H referring to the corresponding labels in figure (b). Nodes are labelled with their clonal frequency in percent and their genotype, again with reference to (b). **(b)** x-axis corresponds to 40 genetic loci of interest which have been used for the analysis. y-axis corresponds to 25 cells analyzed. Each letter A-H corresponds to a set of mutations at the corresponding loci. Colors refer to a set of cells having a given set of mutations. The genotype of any cell can thus be inferred by considering the set of colored squares for that particular row of the matrix. **(c)** Histogram showing the number of cells with each genotype from (a); the column heights sum to 25. **(d)** Histogram showing the number of loci mutated for a given transition in graph (a). x-axis represents the cellular fraction containing each particular set of mutations. Note that mutations F, G and H all have cellular fractions of 12% and have been overlayed rather than stacked.

2.1

In **figure 1a**, each node represents a single clone. For this purpose, we define a clone as a set of cells with the same mutation pattern at our 40 loci of interests (figure 1b). There is likely to be additional genetic variability at other loci within each clone, but we ignore this variability for the present purpose. The color of a node specifies the mutation pattern of the clone with reference to figure 1b. The colour corresponds to the last set of mutations that was aquired in a given clone; e.g. clone 'AB' has both mutations 'A' and mutations 'B' but is coloured blue as mutations 'B' were aquired on a background of 'A'. The label next to the node specifies the full genotype of the corresponding clone with reference to figure 1b.

In inferring the phylogenetic relationship between clones, we assume that the same set of mutations has not developed multiple times independently and that no mutations have been reversed. This implies that a population of cells {K} with two sets of mutations {X} and {Y} must be descendants of a population of cells {L} that have mutations {X} but not {Y}. This allows us to construct an unambiguous phylogenetic tree from the single cell data in figure 1b.

The number on each node specifies the clonal frequency $\pi_k = \frac{N_k}{\sum_l N_l}$ represented by that particular clone where N_k is the number of cells with genotype k. The clonal frequency is also represented by the size of the node. Note here that π_k is mapped on to the width/height of the node rather than the area of the node in the case of different node shapes. Note also that some nodes have clonal frequencies of 0 and are thus technically not clones as defined in the lectures, but the term is used here to apply to both current and historic clones.

Edges between nodes represent acquisition of a set of mutations generating the child node from the genetic background represented by the parent node. The label on the node (e.g. '+B') indicates which set of mutations has been acquired to generate the child node from the parent node with reference to figure 1b.

In **figure 1b**, each small square represents a given cell at a given locus. The 40 genetic loci [1:L] considered in this assignment are arranged along the x-axis. The 25 cells considered are arranged along the y-axis. Thus if cell i has a coloured square at locus j, this implies that cell i has a mutant phenotype at locus j. The colour of the square indicates which set of mutations in $\{A : G\}$ this locus belongs to. The loci have been grouped such that each block defines a clone in figure 1a.

Along the left y-axis, cells have been grouped according to their genotype with the full genotype of each set of cells (e.g. 'ABC') given in the notation described for figure 1a. The right y-axis indicates the clonal frequencies π_k previously described for figure 1a.

Along the x-axis, mutations have been grouped such that any full set of mutations is shared by all cells with a single mutation from the set. The fraction of cells with a given set of mutations is specified under the labels ('Cellular fraction'). Note that this does not refer to the clones in figure 1a and that the numbers thus do not add to 100. Instead, it reflects the relative height of the corresponding coloured region. The Cancer cell fraction (CCF) τ_j is also given below the cellular fraction, and defined such that $\tau_j = \frac{N_j}{N_{cancer}}$ where N_j is the number of cells with the set of mutations j and N_{cancer} is the total number of cancerous cells. That is, the CCF is the fraction of cancerous cells that carry a given set of mutations.

In the row summary in **figure 1c**, each bar corresponds to a single clone from figure 1a. The x-label and bar color indicate the genotype of the clone as previously described. The y axis represents the number of cells with a given genotype in units of '25π'. That is, the y-axis represents the clonal frequencies π_k scaled such that it takes integer steps of individual cells and sums to the total cell count. The clones are ordered by number of cells from lowest to highest.

In the column summary in **figure 1d**, each bar represents a single set of mutations rather than a single clone, and these are colour-coded as in figure 1b. The x-axis represents the cellular fraction (cancer cell fraction) described for figure 1b. The y-axis indicates how many individual loci make up a particular set - i.e. the width of the corresponding coloured area in figure 1b.

2.2

Finally, we can consider the type of information that would be available from single cell and bulk sequencing data for this particular cell population. If we were to have high-quality single cell sequencing data allowing us to unambiguously assign the genotype at all 40 loci of interest, this would allow us to reconstruct figure 1b unambiguously. From this data, we would in turn be able to reconstruct both row and column summaries, as well as the phylogenetic relationship

between clones and cells giving rise to figure 1a. It would thus give us well-defined clones and the hierarchical relationship between them.

On the contrary, if we were to conduct bulk sequencing of the 25 cells, or a larger number of cells where these 25 are a representative sample, then the information available to us would instead be cellular frequencies.

$$\tau_j^{cell} = \sum_{k=0}^{K} \pi_k G_{Kj} \tag{1}$$

G_{kj} is 1 if site j is mutated in clone k and 0 otherwise. π specifies clonal frequency as above, and $\tau_j^{cell} = (1-\pi_0)\tau_j$

In other words, given bulk sequencing data we would have access to mean mutational frequencies across all cells, corresponding to collapsing the columns of figure 1b and giving us the column summary, but without colors (i.e. the three bars at a cellular fraction of 0.12 would be indistinguishable and instead give rise to a single bar with height 1+3+5=9). That is, we would get a histogram of the mutation counts at different cell fractions. We also would not have access to the fraction of cells that are healthy and therefore only be able to infer the cellular fraction, not the cancer cell fraction, of the mutations. This is exemplified in figure 2a for the data given in the assignment.

Assuming very low noise, we could thus group mutations A, B, C, D, E. However, we would not be able to tell if mutations F, G, H gave rise to a single clone or multiple clones as in the present case. From this data, we would be able to infer some phylogenetic relationships given the assumption that a given mutation does not arise spontaneously twice. We thus know that a mutation j present in clone k must also be present in all descendants of k, i.e.

$$\tau_j^{cell} = \sum_{I \in \mathrm{de}(k)}^{K} \pi_I = (\bar{\Phi} \cdot \pi)_j \tag{2}$$

Here, $\bar{\Phi}$ is the transitive closure of the adjacency matrix Φ; i.e. $\bar{\Phi}_{kl}$ is 1 if k is an ancestor of l and 0 otherwise.

Unfortunately, this problem is underconstrained and we cannot uniquely determine π and $\bar{\Phi}$ from τ^{cell}. However, we can constrain the problem further by noting that the sum of the cellullar frequencies of daugther clones can never exceed the cellular frequency of a parent clone, i.e. $\forall j \;\; \tau_j^{cell} \geq \sum_{i \in ch(j)} \tau_i^{cell}$.

Even given this constraint we are unable to determine unambiguous phylogenetic relationships based on the data in figure 2a. One possibility is the 'correct' graph given in figure 2b. However, simpler graphs exist that are also consistent with the data as illustrated in figure 2c and 2d. Here, the numbers on the nodes represent cellular frequencies rather than clonal frequencies.

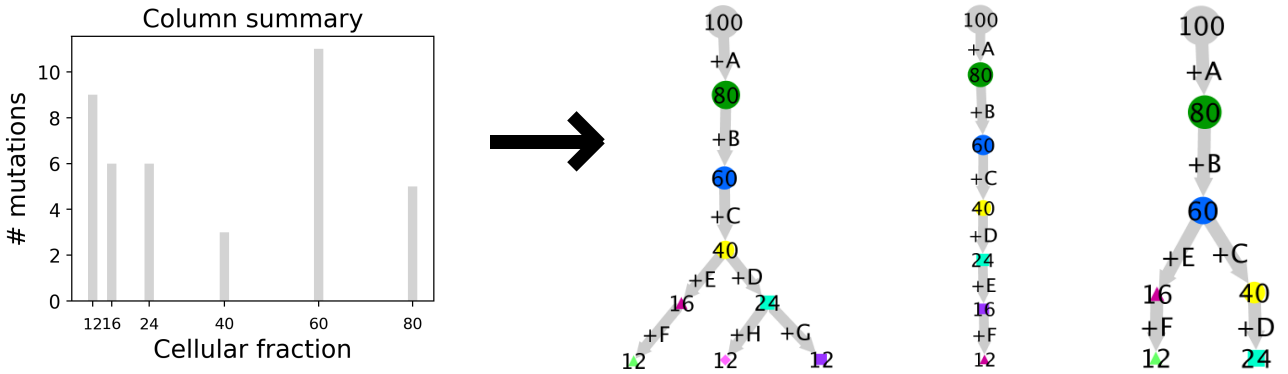


Figure 2: **(a)** example of summary bulk sequencing data given mutation count for different cellular frequencies. In reality, data is noisy and this noise-free diagram must be inferred by clustering. **(b)-(d)** examples of clonal phylogeny consistent with the data in (a). Labels on nodes indicate cellular frequencies for the most recently acquired mutation.

If we also want to infer copy numbers using bulk sequencing data, we can use the full per-locus read count ratios D_j and allelic read counts R_{jB_j} to set up a Hidden Markov Model allowing us to infer maximum likelihood cancer cell fractions τ_j and copy numbers (c_A, c_B) at all our loci of interest along the genome. In the case of single cell data, we can instead infer relative copy number data directly from the sequencing data, and convert this to absolute copy number using paired normal samples.

Appendix

```
"""
code for generating figure 1 in Cancer Evolution assignment 1.
Additional annotations were added in LaTeX.
The graph (figure 1a) was visualized in cytoscape.
"""

import matplotlib.pyplot as plt
import numpy as np
from matplotlib.patches import Rectangle
import networkx as nx
import copy

G = nx.Graph() #initialize graph

#prepare figure 1b
ax = plt.subplots()[1]
plt.xlim(0,40)
plt.ylim(0, 25)

#initialize vectors for storing data for summary plots
fracs = []
xs = []
nmuts = []
muts = []
names = []
ncells= []
cols = ['#d3d3d3', 'g', 'b', 'y', 'c', 'm', '#00FF00', '#9933FF', '#FF66FF' ]

def rects(lims, col, name, parent, size):
    '''given x, y values for a clonal block, plots it and stores statistics'''
    rect = Rectangle((lims[0], lims[2]), lims[1]-lims[0],\
                     lims[3]-lims[2], facecolor = col)
    ax.add_patch(rect)

    #store some information for row/column summaries
    fracs.append((lims[3]-lims[2])/25)
    xs.append([lims[0], lims[1]])
    nmuts.append(lims[1]-lims[0])
    names.append(name)
    muts.append(name[-1])
    ncells.append(size)

    #add node to graph
    G.add_node(name, lab = str(int(size/25*100)), size = int(size/25*100))
    if not parent == 'None':
        #add edge to graph
        G.add_edge(parent, name, label = '+'+name[-1])

#add segments corresponding to each clone
rects([0,40,0,25], '#d3d3d3', '0', 'None', 5)
rects([0,5,0,20], 'g', 'A', '0', 5)
rects([5,5+11,0,15], 'b', 'AB', 'A', 5)
rects([16,16+3,0,10], 'y', 'ABC', 'AB', 0)
rects([19,19+6,0,6], 'c', 'ABCD', 'ABC', 0)
rects([25,25+6,6,6+4], 'm', 'ABCE', 'ABC', 1)
rects([31,31+1,6,6+3], '#00FF00', 'ABCEF', 'ABCE', 3)
```

```
rects([32,32+5,0,3], '#9933FF', 'ABCDG', 'ABCD', 3)
rects([37,37+3,3,3+3], '#FF66FF', 'ABCDH', 'ABCD', 3)

#add gridlines
for x in range(40):
    plt.axvline(x, color='w')
for y in range(25):
    plt.axhline(y, color='w')

#save figure
dirn = '/Users/kris/Documents/cambridge/mphil/assignments/ce/ce1/'
plt.xticks([])
plt.yticks([])
plt.savefig(dirn+'squares.png', bbox_inches = 'tight', pad_inches = 0.8)
plt.show()

#store graph
nx.write_graphml(G,
    '/Users/kris/Documents/cambridge/mphil/assignments/ce/ce1/graph.graphml')
nx.draw(G, with_labels = True)
plt.show()

fs = 16 #fontsize in plots
s = 0.8 #scaling of plot size

#create row summary
inds = np.argsort(ncells)
plt.figure(figsize = np.array([6,3])*s)
plt.bar(range(len(names)), np.array(ncells)[inds],\
        color = np.array(cols)[inds], width = 0.2)
plt.xticks(range(len(names)), np.array(names)[inds], rotation=90, weight='bold')
plt.yticks(FontSize = fs-4)
plt.title('Row summary', FontSize=fs)
plt.xlabel('Cellular populations', FontSize=fs)
plt.ylabel('Number of cells (25$\pi$)', FontSize=fs)
plt.savefig(dirn+'rows.png', bbox_inches = 'tight', dpi = 360)
plt.show()

#create column summary
inds = np.argsort(-np.array(nmuts[1:]))
newfracs = copy.deepcopy(fracs); newfracs[5]=0.18; newfracs[4] = 0.255
ticks, va = [], []
for f in np.array(fracs[1:])[inds]: #need ticks for cellular fraction and CCF
    ticks.append(str(int(100*f)))
    va.append(-0.03)
    ticks.append('('+str(int(100*f*25/20))+')')
    va.append(-0.14)
    ticks.append(' ')
    va.append(-0.23)
fig = plt.figure(figsize = np.array([6,3])*s )
ax = plt.gca()
plt.bar(np.array([int(f*100) for f in newfracs[1:]])[inds],
        np.array(nmuts[1:])[inds],
        color = np.array(cols[1:])[inds], width = 1.5)
ax.set_xticks( np.repeat(np.array([int(f*100) for f in newfracs[1:]])[inds], 3))
ax.set_xticklabels( ticks )
for t, y in zip( ax.get_xticklabels( ), va ):
```

```
    t.set_y( y )
plt.yticks(FontSize = fs-4)
plt.title('Column summary', FontSize=fs)
plt.xlabel('Cellular fraction (CCF)', FontSize=fs)
plt.ylabel('# mutations', FontSize=fs)
plt.savefig(dirn+'cols.png', bbox_inches = 'tight', dpi=360)
plt.show()

#create 'bulk sequencing' graph
nmuts = [9,6,6,3,11,5]
fracs= [0.12,0.16,0.24,0.40,0.60,0.80]
ticks, va = [], []
for f in np.array(fracs):
    ticks.append(str(int(100*f)))
    va.append(-0.02)
    ticks.append(' ')
    va.append(-0.02)
    ticks.append(' ')
    va.append(-0.0)
fig = plt.figure(figsize = np.array([4,3])*s*1.3 )
ax = plt.gca()
plt.bar(np.array([int(f*100) for f in fracs]),
        np.array(nmuts),
        color = '#d3d3d3', width = 2)
ax.set_xticks( np.repeat(np.array([int(f*100) for f in fracs]), 3))
ax.set_xticklabels( ticks )
for t, y in zip( ax.get_xticklabels( ), va ):
    t.set_y( y )
plt.yticks(FontSize = fs-4)
plt.title('Column summary', FontSize=fs)
plt.xlabel('Cellular fraction', FontSize=fs)
plt.ylabel('# mutations', FontSize=fs)
plt.savefig(dirn+'bulk.png', bbox_inches = 'tight', dpi=360)
plt.show()
```