
University of Glasgow
School of Mathematics and Statistics
MSc Dissertation

# Deconvolving mutational signatures from generated cancer genome copy-number mutation data

Lee Sharkey

August 2018

**Abstract**

Cancer mutational signatures are patterns of mutation that correspond to specific mutational processes in cancer cells and thus have clinical relevance. Mutational signatures first must be separated from each other and from noise in sequenced cancer genomes before they can be studied. This project evaluated two algorithms for the deconvolution of cancer mutational signatures in copy-number data: Non-negative Matrix Factorisation (NMF) and Hierarchical Dirichlet Process Latent Dirichlet Allocation (HDP-LDA). To evaluate the algorithms, a generative model was built that resembles - with constraints - real copy-number mutation data. HDP-LDA was found to be unsuitable for the task. NMF was identified as an appropriate deconvolution algorithm and an analysis pipeline was built around the algorithm and generative model.

# Contents

# 1 Introduction

## 1.1 Background on cancer genetics

Healthy cells do not divide uncontrollably. Their replication is regulated by a complex network of proteins - long chains of amino acids that carry out most cell functions. Proteins are made of a selection of twenty kinds of amino acids. Their order and number determine the shape and thence the function of the protein. Genes are sections of a cell's genetic material - deoxyribonucleic acid (DNA) - that encode the order of amino acids in proteins. DNA is composed of two helical chains of nucleotides, the building block molecules of DNA. A chromosome is a single double-helix of DNA and is stored in a cell's nucleus.

Life uses only four different nucleotides to construct DNA, though more are possible. Nucleotides come in pairs, one on each chain: cytosine (C) only pairs with guanine (G); adenine (A) only pairs with thymine (T). Strikingly, genomic DNA is mostly non-coding - it codes for no protein and is considered 'junk' [1].

Cancer is a group of diseases of uncontrolled cell replication. It is caused by mutations (typically many) to genes that regulate cell replication. There are several different types of DNA mutation, which can be divided into two broad classes, **single nucleotide variant** (SNV) mutations or **structural variants** (SV) mutations. SNV mutations are nucleotide 'substitutions' - for example, when a C on one chain is substituted by another letter. There are six basic types of nucleotide (or 'base') substitution.

$$\begin{aligned} \text{C:G} &\rightarrow \text{A:T} \\ \text{C:G} &\rightarrow \text{G:C} \\ \text{C:G} &\rightarrow \text{T:A} \\ \text{T:A} &\rightarrow \text{A:T} \\ \text{T:A} &\rightarrow \text{C:G} \\ \text{T:A} &\rightarrow \text{G:C} \end{aligned}$$

This 'vocabulary' can be expanded by including the flanking nucleotides. For example, when considering trinucleotides in this way, we get a vocabulary of $4^2 \times 6 = 96$ SNVs.

Structural variants are mutations that change the structure of the chromosome; a nucleotide pair might be 'deleted'; a nucleotide pair might be 'inserted'; one or both of the DNA chains might break (single or double strand breaks); whole chromosomes

or segments of DNA might be repeated, sometimes several times over (copy-number mutations); segments of DNA might break off and be 'inverted' i.e. repaired in the wrong direction. A subset of possible copy-number mutations is illustrated in figure 1.

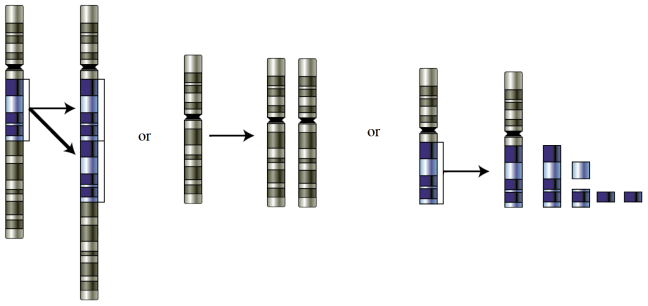


Figure 1: Illustrated examples of chromosomal copy-number variant mutations

## 1.2 Mutational signatures

In recent years, the cost of whole genome sequencing has fallen drastically; this has enabled the sequencing and analysis of whole cancer genomes where hitherto only specific genes could be studied. Therefore we can now cheaply study not only mutations that occur in genes that cause cancer (*driver* mutations) but also mutations that occur in non-coding DNA (*passive* mutations). Cheap whole genome sequencing thus permits the study of patterns of mutation in the whole genome - called **mutational signatures**.

It has long been known that certain mutational processes have a tendency to cause certain types of mutation. For example, in 1964 it was discovered that UV light tended to cause nucleotide C to be substituted by a T [2]. However, prior to cheap whole genome sequencing, deciphering signatures from cancers was difficult since it was not known which mutational processes were active in a cancer; therefore the observed proportions of the various types of mutation in a cancer could be an unknown mixture of signatures, with no way to find out what the mixture components were. Many more genomes can be sequenced with cheap whole genome sequencing, enabling the use of statistical techniques to decipher (or deconvolve) these mixtures of signatures.

Nik-Zainal et al. [3] and Alexandrov et al. [4, 5] characterised the first known mutational signatures. In a sample of cancer genomes they counted the number of each type of substitution and insertions/deletion mutation in each genome in order to construct 'mutational catalogues' of the genomes. Then they used Non-negative

Matrix Factorisation (NMF) [6] to deconvolve the signatures. The details of NMF and other approaches to deconvolve signatures will be discussed in the methods.

In the quest for new and better characterised mutational signatures, ever more genomes from ever more sites of primary cancer have been studied using an expanding repertoire of statistical methods [7, 8, 9, 10]. Catalogues of signatures such as the Mutational Signature Framework 'COSMIC', boast 30 well-defined signatures (though not comprehensive) and can be used to identify the signatures present in individual cancers [11, 5, 12, 13].

The significance of mutational signatures is that they provide insight on the mutational processes operating inside a cancer. By pairing each signature with associated causes of mutation, such as the loss of an intracellular pathway that repairs certain types of DNA damage, we can identify the mutational processes operating inside individual cancers [12]. Associating signatures with mutagenic processes in this way might potentially reveal new targets for anti-cancer therapies. So far, not all known signatures have been associated with a mutagenic process. Signatures can also be used to identify exposures of patients to certain mutagens associated with the signatures [14], which might in future help find the most appropriate treatment for an individual patient [15, 16]. Lastly, we can stratify cancers by the signatures the exhibit, which we can use to infer prognostic associations for a patient's cancer [17].

These signatures largely employ single nucleotide variant mutations and insertion-deletions mutations. But most past studies have done little to catalogue signatures of structural variants, especially copy-number mutations [18], which are the subject of this investigation.

## 1.3 Copy-number mutations

The nature of copy-number mutations presents a unique challenge among mutation types to signature deconvolution. Genomes afflicted with copy-number aberrations can be a complex mixture of fragmented chromosomes. Many of these fragments have varying numbers of copies. Counting the breakpoints (the location of the chromosome where the fragment has broken from) is insufficient to fully characterise the mutations in a way that affords signature deconvolution, since it is also important to account for the different number of copies of each of the fragments. As such, each copy-number variant has a *quantitative* measure (i.e. number of copies) as well as two *qualitative* measures (i.e. whether the breakpoint is present or not and where it is present). Without accounting for all three of these properties, one cannot characterise the copy-number mutation profile of a genome. With single nucleotide

variants and insertion/deletions on the other hand, where a mutation occurs on a genome is almost irrelevant for signature deconvolution; for the most part, only the number and proportions of each type of mutation are important. Thus SNVs and insertion-deletions have only one quantitative property (mutation count) and one only qualitative property (type) that are relevant for signature deconvolution.

Certain cancers such as high grade serous ovarian carcinoma (HGSOC), oesophageal, non-small-cell lung, and triple negative breast cancer exhibit profound copy-number aberrations [19, 20]. This indicates that these genomes are typically exposed to at least one mutational process that other cancers are exposed to more rarely that drives copy-number mutations. The complexity of genomes with profound copy-number aberrations has inhibited the analysis of mutational processes that drive copy-number aberrations, and thus presents a barrier to the study of the prominent mutagenic processes active in these cancers and hence a barrier to targeted therapies.

Mechanisms of chromosomal instability and consequent copy-number mutation include

- **Breakage-fusion-bridge cycle** - where (Step 1) the end of a chromosome breaks off; (Step 2) when the chromosome is replicated, its exposed ends are now able to fuse together; (Step 3) when the cell divides and the fused chromosome twins are pulled apart, they break, not necessarily at the location where they fused. [21]
- **Chromothripsis** - in which a chromosome is broken into hundreds or thousands of fragments. Its causes are not well understood and are probably several [22].
- **Tandem repeats** - where sequences of nucleotides are repeated one or several times directly adjacent to the initial sequence.

In the study of copy-number mutations, we can sequence the fragments present in a cancer genome and identify which regions in the chromosome the fragments have come from by comparing their nucleotide sequences with the genome taken from healthy cells. We can thus represent the chromosomes of fragmented genome graphically as seen on the left hand side of figure 2. Each row is a chromosome, and the number of copies and length of each consecutive fragment being represented by the height and length of the blue line. A perfectly intact chromosome would be represented by a solid blue line at (Absolute copy-number) = 1.

We can describe these data using six features that are hallmarks of above-mentioned mechanisms of chromosomal instability [23, 24, 25, 26]). These features are:

1. **Breakpoint number** - the number of double-strand breaks in a sliding window of 10 million nucleotides (megabases, MB)
2. **Segment copy-number** - the number of copies of each segment
3. **Change point copy-number** - the absolute difference in copy-number between adjacent segments
4. **Normalised distance from the centre of the chromosome** of each double-strand break
5. **Size of segment** - the length of each genome fragment in 10 MB units
6. **Oscillating copy-number length** - the number of adjacent fragments that alternate between two copy-number states

The copy-number mutations of each genome are thus described by six feature distributions. These distributions represent the data that are the subject of this project.

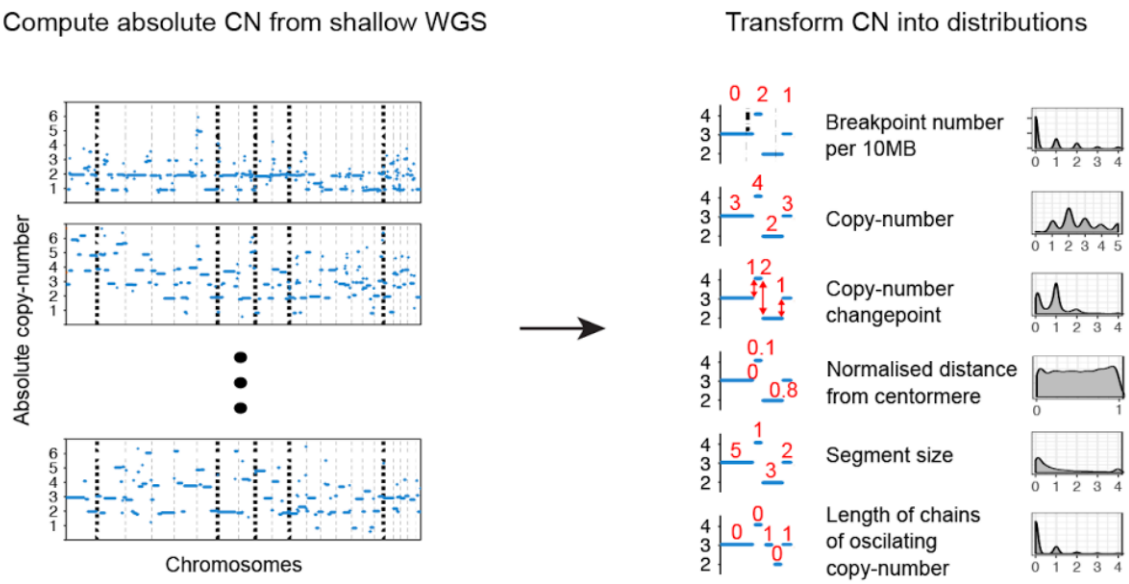


Figure 2: Copy-number mutation data representations. Figure adapted from MacIntyre et al. [27], our collaborating group

## 1.4 Project aims and questions of interest

It is hoped that better understanding of copy-number mutational signatures will yield insights that can be used to improve clinical outcomes. Copy-number data present a challenge to signature deconvolution due to the quantitative and doubly qualitative properties of the mutations. The aims of this project, therefore, are to:

1. Identify or develop suitable methods to deconvolve signatures.
2. Build a generative model and analysis pipeline for copy-number data.
3. Use the generated data to compare the methods of signature deconvolution.

Questions of interest include:

1. Are existing statistical methods for signature deconvolution adequate for copy-number data or do we need new ones?
2. Do our generated data resemble the real copy-number data?
3. Using the analysis pipeline on the generated data, what do the results indicate about its use on real copy-number data?
4. How do methods for signature deconvolution compare? Which are preferred?

# 2 Methods

## 2.1 The generative model

It transpired halfway through the project period that the copy-number data used by MacIntyre et al. [27, 28] were not accessible due to pre-publication embargo by the publisher of their manuscript. Unfortunately, we therefore needed to infer properties of the data from inspection of figures rather than using quantitative approaches to calibrate the generative model. Important properties of their data were visible, however, from the figures in their preprint manuscript. Figure 3 shows their reconstruction of the feature distributions of the eight different preliminary deconvolved signatures. It can be seen that many of the distributions are very similar, even when their signatures were deconvolved using a method that employed summary statistics to select for distributions of different shapes. This suggests that

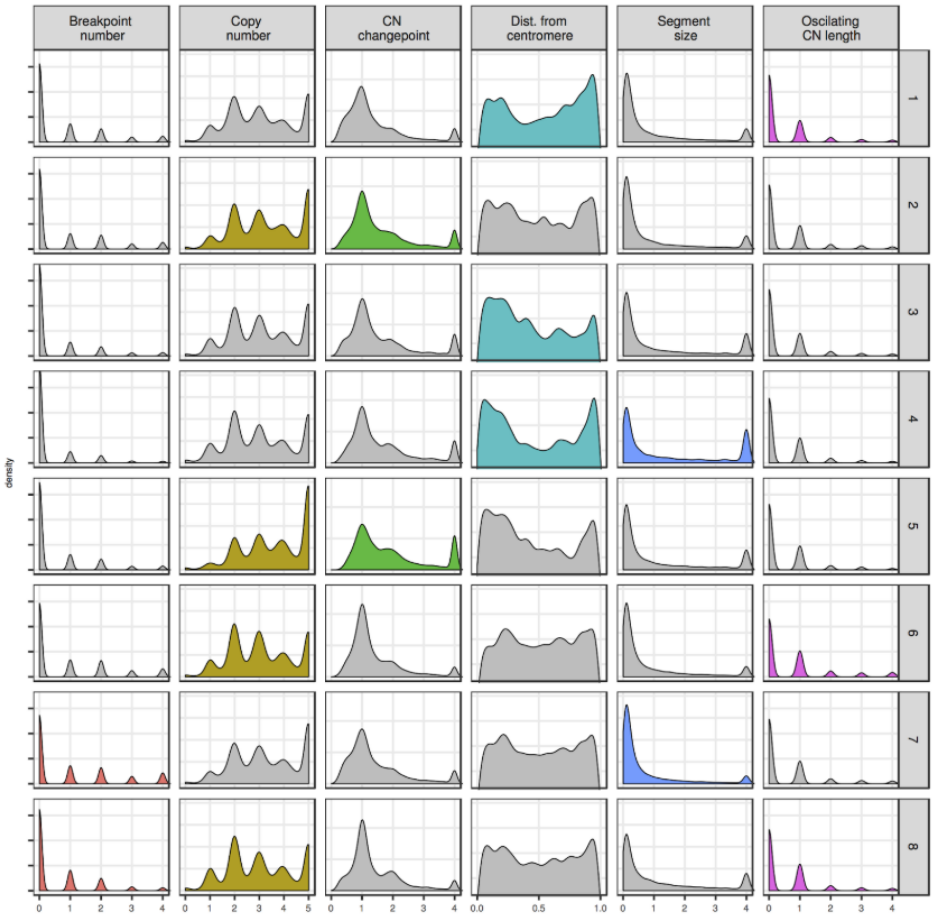


Figure 3: Signatures from copy-number data derived using a previous method. Figure adapted from Supplementary Figure 7 in MacIntyre et al. [27], our collaborating group

the genome data, which will assume are composed of weighted sums of the signatures plus noise, will be similar too.

In this work, the number of generated genomes in a corpus is $J = 117$ (unless otherwise specified), since collaborators MacIntyre et al. used 117 sequenced genomes. For each genome $j$ we have a set of observations from six distributions $\boldsymbol{x}_j = (x_{j1}, x_{j2}, x_{j3}, x_{j4}, x_{j5}, x_{j6})$ where each $x_{jf}$ is a vector of $n_{jf}$ observations for that feature. We use a a Gaussian mixture model to model each feature of each genome. The mixture component means for the six features $\mu_{1,1}, \cdots, \mu_{n_1,1}, \cdots, \mu_{1,6}, \cdots, \mu_{n_6,6}$ were fixed at the appropriate values judged by inspection of figure 3, except for feature 4, for which the number of components $n_4$ was chosen and the component means $\mu_{1,4}, \cdots, \mu_{n_4,4}$ were set at equal intervals on $[0, 1]$. For all six features' components' standard deviations $\sigma_{1,1}, \cdots, \sigma_{n_1,1}, \cdots, \sigma_{1,6}, \cdots, \sigma_{n_6,6}$, ranges for uniform sampling were chosen such that component widths resembled the figures. Genomes

and signatures all share the same component means and standard deviations; they differ by their component weights,

$$\lambda_{1,1,p}, \cdots, \lambda_{n_1,1,p}, \cdots, \lambda_{1,6,p}, \cdots, \lambda_{n_6,6,p}.$$

Priors on the feature component weights were judged by inspection of figure 3. The priors were used to parameterise a Dirichlet distribution from which the component weights were sampled. The concentration parameter for each distribution was adjusted to induce an appropriate degree of stochasticity in the signatures' feature mixture component weights, again judged by inspection of figure 3. Together, the feature component means, standard deviations, and component weights parameterise the **base-level mixtures**.

Genomes' component weights are constructed from linear combinations of the signatures' component weights. The first step in generating a corpus of genomes is to sample $K$ independent sets of component weights described above, one set for each signature. Then randomly sample the total number of signatures to be present in a genome. The probability of a certain number of signatures being present in each genome was determined by the corresponding densities of a Poisson distribution with an expected value of 4. Because draws of 0 are excluded, there are on average 4.08 signatures in each genome.

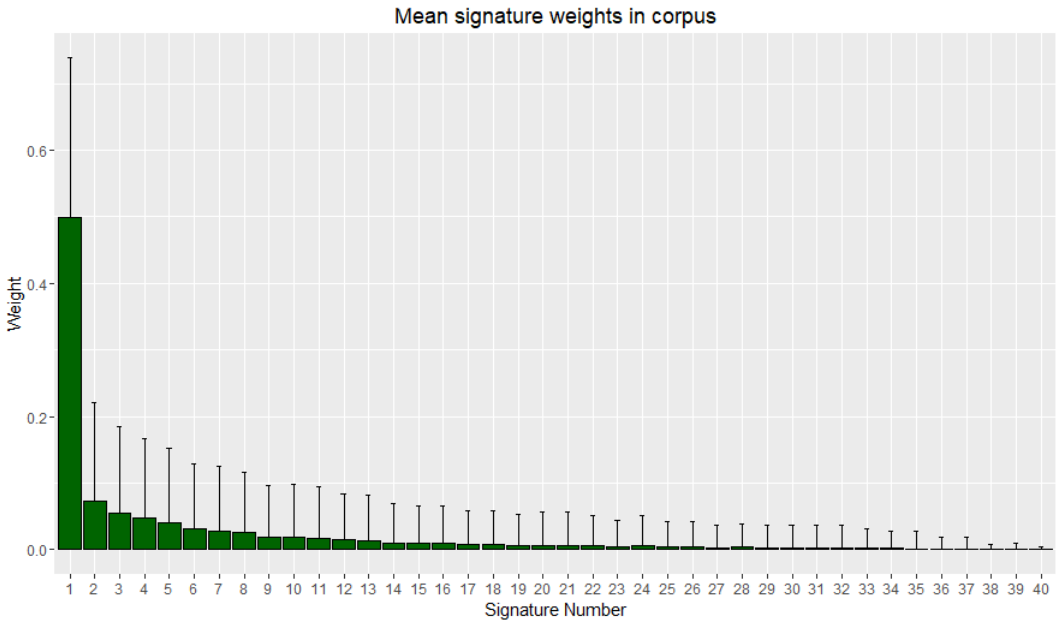


Figure 4: Mean weight of each signature in any given generated corpus. Error bars are standard deviations. In a particular genome, the weights of the present signatures always sum to 1.

Then, select a random subset of the signatures, with the size of the subset deter-

mined by the above Poisson-sampling. The probability of signatures being present in a given genome varies according to a stick-breaking distribution (defined in section 2.3). The weight of each selected signature follows another stick breaking distribution. The Beta distributions in these stick breaking distributions can be parameterised such that the sticks are broken in a more or less even way, as desired. The combination of both stick-breaking distributions and the method of sampling the number of signatures present yields a distribution of signature weights that weights some signatures more and more often while nevertheless generating a heterogenous corpus genome. This reflects the finding in [4, 5] that some signatures are much more common and weighted more heavily than others. The average signature weights in a generated corpus can be seen in figure 4.

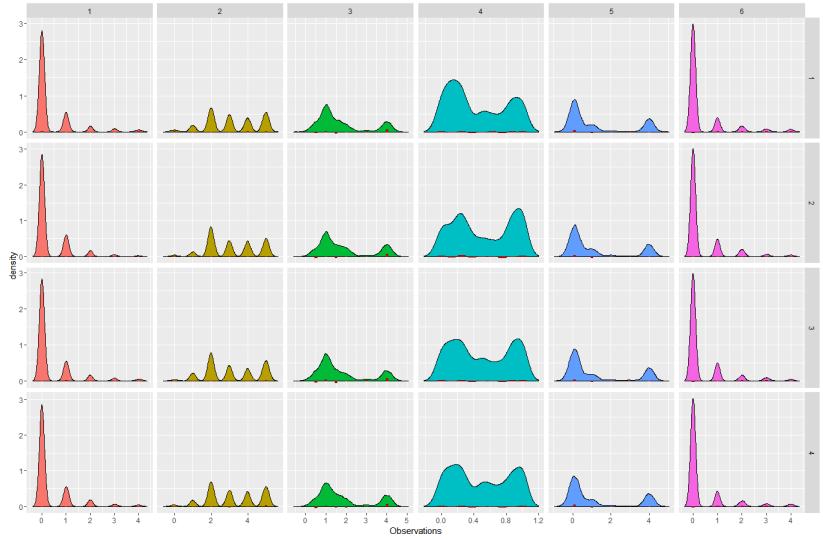


Figure 5: Example generated genomes with biased errors in inferred component weights

With this random selection of signatures with appropriate weighting, the component weights of the signatures are weighted by their exposure and summed to yield the component weights of a genome. Signature component weights thus serve as basis vectors that are linearly combined to make genome component weights. Observations are then generated from the mixture models parameterised by these summed components, illustrated in figure 5.

Originally, it was intended to infer the genome component weights from the observations, then to deconvolve the signature component weights from the genome components using statistical methods. It was possible to use the base-level means and standard deviations to infer approximate genome component weights by calculating the density of each of the components at each observation, then summing the densities for each component across all observations to get the component weights

for the genome.

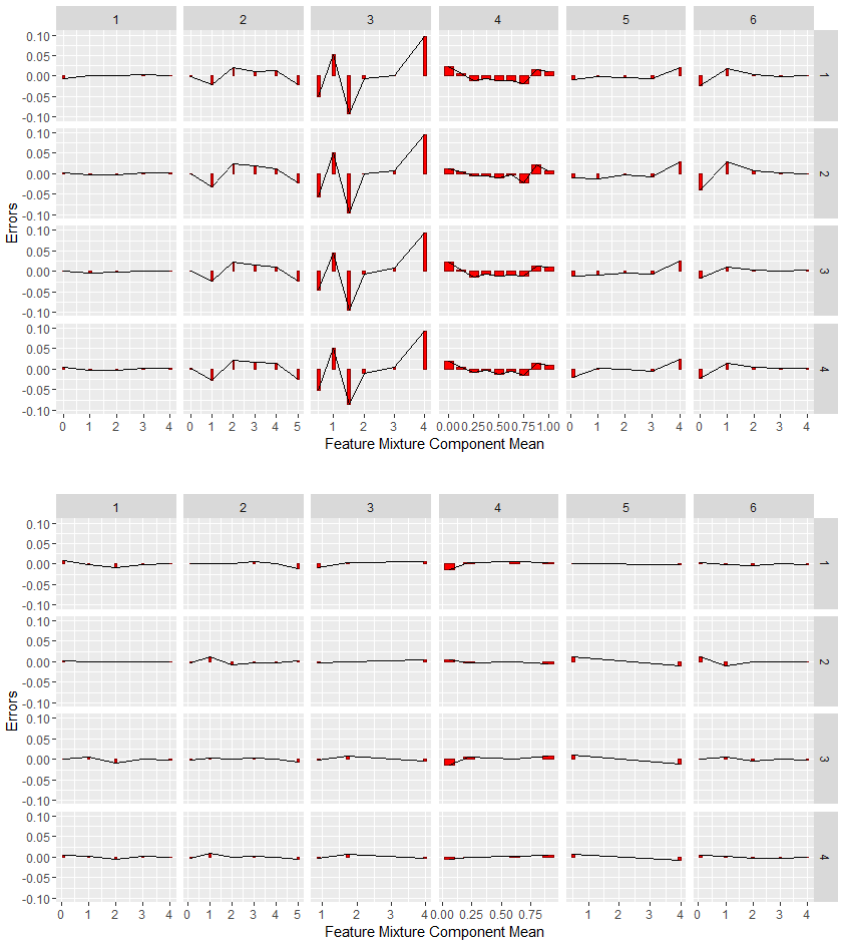


Figure 6: **Top**: Biased errors between four example genome component weights and corresponding weighted summed signature weights. **Bottom**: Unbiased errors between four example genome component weights and corresponding weighted summed signature weights.

However, this approach proved problematic, since there were biased errors in the inferred components where component-densities overlapped. On generated data, it would be trivial to correct for this bias to yield the true genome component weights since we have access to the true component weights. However, this would be problematic when applied to real data, since the ground truth is unobserved. This issue hinders inference of the true signature component weights from these erroneous base-level genome component weights.

To correct this, another level of component weights was added. Now, the same base-level distributions are used to generate observations for signatures and genomes (not only genomes as above), and expectation maximisation (EM) is used to fit a

new **meta-level mixtures** to the observations for signatures and genomes. Thus new meta-level component means and standard deviations as well as new meta-level component weights are fitted to the signatures and genomes. Now, both the genomes and the signatures incorporate the bias. This approach resolved the biased errors between signature and genome component weights (figure 6), reducing the absolute values of the errors from 0.01498 using the base-level representation to 0.003534, reducing the error by a factor greater than 4.

NMF and Hierarchical Dirichlet Process Latent Dirichlet Allocation (HDP-LDA) are used on the genomes' meta-level component weights to deconvolve the signatures' meta-level component weights. NMF was chosen because it is the gold-standard approach for deconvolving signatures from SNV-insertion-deletion mutations. It serves here as a baseline for copy-number data. HDP-LDA was chosen because LDA has been used in SNV-insertion-deletion data [18] and HDP-LDA provides a principled mechanism to automatically infer the number of signatures. Since explanations of HDP-LDA especially are quite involved, in the interest of structure, fuller, gentler explanations of Latent Dirichlet Allocation, Dirichlet Processes, and HDP-LDA are provided in the appendices.

## 2.2 Applying NMF to generated copy-number data

Each genome $j$ is described by its vector of component weights

$$\lambda_{1,1,j}, \cdots, \lambda_{n_1,1,j}, \cdots, \lambda_{1,6,j}, \cdots, \lambda_{n_6,6,j}.$$

To simplify notation, give a unique index $w = 1, \cdots, W$ to each component weight $\lambda_{1,j}, \cdots, \lambda_{W,j}$. Suppose that genome component weight vectors are made of linear combinations of $K$ signature component weight vectors, $s_{1,k}, \cdots, s_{W,k}$. Then define the following matrices:

$$C = \begin{bmatrix} \lambda_{1,1} & \cdots & \lambda_{1,J} \\ \vdots & \ddots & \vdots \\ \lambda_{W,1} & \cdots & \lambda_{W,J} \end{bmatrix}_{W \times J}$$

$$S = \begin{bmatrix} s_{1,1} & \cdots & s_{1,K} \\ \vdots & \ddots & \vdots \\ s_{W,1} & \cdots & s_{W,K} \end{bmatrix}_{W \times K} \qquad E = \begin{bmatrix} e_{11} & \cdots & e_{1,J} \\ \vdots & \ddots & \vdots \\ e_{K,1} & \cdots & e_{K,J} \end{bmatrix}_{K \times J}$$

where $C$ is the matrix of genome component weights, S is the matrix of signature component weights, and E is the matrix of exposures of each genome to each signa-

ture. Given $C$, we want to find non-negative matrices $S$ and $E$ such that

$$C \approx S \times E.$$

This task is NP-hard [29] and ill-posed (i.e. there exist multiple satisfactory solutions for matrices $S$ and $E$). Furthermore, the rank $K$ must be selected in advance. In spite of these difficulties, there exist useful algorithms for the task. After random initialisation of matrices $S$ and $E$, the following updates are applied iteratively [6]

$$e_{k,j} \leftarrow e_{k,j} \frac{[S^T C]_{k,j}}{[S^T S E]_{k,j}}$$
$$s_{w,k} \leftarrow s_{w,k} \frac{[C E^T]_{k,j}}{[S E E^T]_{k,j}},$$

Other algorithms with modified updates exist, such as [30], used in the current implementation.

Applying NMF to matrix $C$ multiple time yields different results due to different random initialisations and NMF's being ill-posed. In order to arrive at consistent estimates, instead of using NMF by itself, an algorithm that employs it, algorithm 1 is used.

Algorithm 1 uses the Frobenius norm, which is the quantity the minimised by NMF, where $A$ is an $m \times n$ matrix,

$$||A||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}.$$

The Frobenius norm is also used to calculate $\epsilon$, the difference between the deconvolved signature matrices at iteration $i$ and $h = \frac{i}{2}$. The reproducibility of the cluster is taken as its average silhouette width, which is a measure of the dispersion of the cluster, and is here defined as the average cosine similarity between the cluster centroid and the cluster members. Cosine similarity is the cosine of the angle $\theta$ between arbitrary vectors $v$ and $u$, defined by

$$cos(\theta) = \frac{v \cdot u}{||v||\ ||u||}.$$

It may be used as a metric of similarity between the direction of two vectors, ignoring their magnitudes.

## 2.3 Applying HDP-LDA to generated copy-number data

*Introductions to Latent Dirichlet Allocation, Dirichlet Processes, and Hierarchical Dirichlet Process-LDA may be found in the appendix, section 5.*

**Algorithm 1:** Deconvolute Signatures using NMF. Adapted from [4]

**Data:** $C$ *(observed corpus component weights)*

Init $\delta$ *(tolerance)*

Init $\epsilon > \delta$

Init $R$ *(max. iterations)*

Init Init $\boldsymbol{v}_c$ (random matrix of cluster centres)

Init $i$ *(index)*

Init $h$ *(increments half as quickly as $i$)*

**while** $\epsilon > \delta$ *OR* $i < R$ **do**

- Perform NMF on matrix $C$ with random initialisation to get matrices $S$ and $E$
- Row-bind $S$ and $E$ to the store-matrices $\widehat{S}$ and $\widehat{E}$ respectively
- Perform k-means clustering on $\widehat{S}$, initialised by $\boldsymbol{v}_c$, to get iteration-averaged matrix of deconvolved signatures $\bar{S}$
- $\boldsymbol{v}_c \leftarrow \bar{S}$
- Append $\bar{S}$ to list $\bar{\boldsymbol{S}}$
- **if** $i$ *is even* **then**
  - Increment $h$
- Increment $i$
- $\epsilon \leftarrow ||\bar{\boldsymbol{S}}_h - \bar{\boldsymbol{S}}_i||_F$ *(Frobenius norm)*

Reorder $\bar{S}$ by cluster reproducibility (where reproducibility is the average silhouette width: the average cosine similarity of the cluster centre to each member of the cluster)

HDP-LDA is commonly applied to topic modelling of documents. In this application of HDP-LDA to SNV-insertion-deletion data, the observed objects are documents (genomes) composed of exchangeable mixtures of discrete words (mutations) from a vocabulary (types of mutation). Each word (mutation) is assigned a latent topic (signature) and the probability that a given topic will yield each type of word is inferred. However, the feature component weights, are continuous on (0,1). Therefore, to make LDA applicable to the present setting, the component weights of each genome are first discretised and converted to strings - a 'component weight-increment'. A signature is therefore defined by the the probability that it will yield some component weight-increment of one of the $V$ feature mixtures components. Thus the discrete objects in our HDP-LDA model are component weight-increments. A signature may therefore contribute a certain shape of distribution defined by the feature component weights to the feature-distributions of a genome. This project sought to deconvolve the shape of the contribution of each present signature to the components.

The stick-breaking construction of the HDP [31] is as follows. Define the stick-breaking distribution thus

$$\begin{aligned} V_k &\sim Beta(1, \alpha_0) && k = 1, \cdots, \infty \\ \pi_k &= V_k \prod_{l=1}^{k-1} (1 - V_l) \\ \boldsymbol{\pi} &= (\pi_k)_{k=1}^{\infty}. \end{aligned} \tag{1}$$

For convenience, we say $\boldsymbol{\pi} \sim Stick(\alpha_0)$. Note that $\sum_{k=1}^{\infty} \pi_k = 1$. Figure 7 illustrates the stick-breaking process.

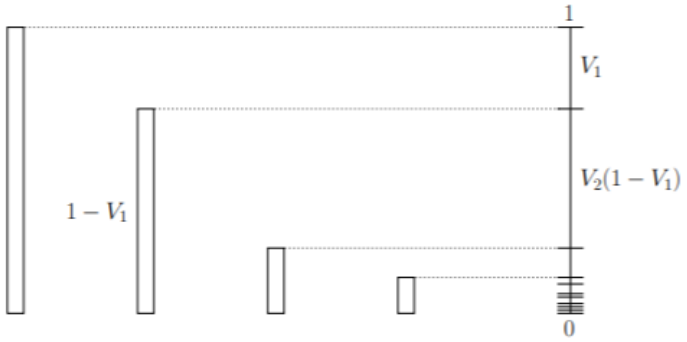


Figure 7: An illustration of the stick-breaking process. Taken from reference [32]

Now letting $z_{ji}$ be an indicator variable such that $\theta_{ji} = \phi_{z_{ji}}$, we have

$$\begin{aligned} \phi_k &\sim H, & k = 1, \cdots, \infty \\ \boldsymbol{\beta} &\sim Stick(\gamma), \\ \boldsymbol{\pi}_j &\sim DP(\alpha_0, \boldsymbol{\beta}), \\ z_{ji} &\sim \boldsymbol{\pi}_j, \\ x_{ji} &\sim F(\phi_{z_{ji}}) \end{aligned} \tag{2}$$

The atoms $\phi_k \sim H$ are independent draws from the baseline probability distribution $H$ and determine the probability of sampling a word from the vocabulary given the topic $k$. Here $H$ is a Dirichlet distribution with dimension $W$. Inference will determine which of the samples from $H$ resemble real signatures, and thus which $\phi_k$ are weighted heavily in genomes.

The weight of each signature $\phi_k$ will vary in each genome. Since a Dirichlet process is a 'probability measure over probability measures, each $\boldsymbol{\pi}_j$ is similar to the distribution $\boldsymbol{\beta}$, and the degree of similarity is defined by concentration parameter $\alpha_0$. $\boldsymbol{\beta}$ is thus the global weight of signatures, and $\boldsymbol{\pi}_j$ the weight of the signatures in genome $j$. Each component weight-increment $x_{ji}$ is associated with a topic number, i.e. a signature number, $z_{ji}$. $z_{ji}$ is an indicator variable that that is sampled from $\boldsymbol{\pi}_j$. Therefore each integer $k$ has probability $\pi_{jk}$ of being sampled from $\boldsymbol{\pi}_j$). Finally, let $F(\phi_{z_{ji}}) = Mul(1, \phi_{z_{ji}})$, allowing us to generate single component weight-increments parameterised by the signatures.

## 2.4 Inference on HDPs

There have been several approaches adopted for HDP-LDA inference. These include standard Gibbs Sampling [31], collapsed variational inference [33], practical collapsed variational bayes inference [34], practical collapsed stochastic variational inference [35] and online variational inference [36] among others. For convenience of implementation, Online variational inference for HDP is used here, described in Wang et al. [36, 37]. As with all variational inference, it involves maximising a lower bound on the difference between the true distribution and an analytic approximation distribution [38]. Scope dictates that further delineation of inference for HDPs be confined to citation of [36].

## 2.5 Technical methodological notes

The R programming language was used for all data generation and analysis, with the exception of HDP-LDA, which took as input data generated in R but used the gensim topic modelling package for Python[39]. NMF was carried out using the R NMF package [40]. Gaussian mixtures were fitted using the Mixtools package in R [41].

# 3 Analysis

## 3.1 Identifying appropriate signature deconvolution algorithms

The first stage of this project was theoretical - to identify whether existing statistical methods for signature deconvolution were adequate for copy-number data or whether new ones were needed. The project as it was originally conceived would model a signature as six dimensional Gaussian for which we would derive an inference algorithm for a Bayesian-non-parametric Gaussian mixture model of the genomes. Here it is shown what this approach faced issues that necessitated the use of other algorithms and signature representations.

Originally it was supposed that simple signatures could be described by a six-dimensional Gaussian on six-dimensional feature-space; a Normal-Wishart (NW) distribution as the conjugate prior to a multivariate Gaussian could be used. The hierarchical structure of the model, given $\mu_o, \rho, \Lambda, \nu, \gamma$ and $\alpha_0$, would be

$$\begin{aligned} G_0 &\sim DP(\gamma, \mathcal{NW}(\mu, S|\mu_0, \rho, \Lambda, \nu)) \\ G_j &\sim DP(\alpha_0, G_0) \\ \theta_{ji} = (\mu_{ji}, S_{ji}) &\sim G_j \\ x_{ji} &\sim \mathcal{N}(\mu_{ji}, S_{ji}^{-1}) \end{aligned}$$

This is equivalent to the following stick-breaking construction:

$$\begin{aligned} \boldsymbol{\beta} = (\beta_k)_{k=1}^{\infty} &\sim Stick(\gamma) \\ \boldsymbol{\pi}_j = (\pi_{jk})_{k=1}^{\infty} &\sim DP(\alpha_0, \beta) \\ \phi_k = (\mu_k, S_k) &\sim \mathcal{NW}(\mu, S|\mu_o, \rho, \Lambda, \nu) \qquad k = 1, ..., \infty \\ z_{ji} &\sim \boldsymbol{\pi}_j \\ x_{ji} &\sim \mathcal{N}(\phi_{z_{ji}}) = \mathcal{N}(\mu_{z_{ji}}, S_{z_{ji}}^{-1}) \end{aligned}$$

and

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\phi_k} \qquad\qquad G_j = \sum_{k=1}^{\infty} \pi_{jk} \delta_{\phi_k}.$$

In the generative model, genome and signature observations would be drawn from the 6-dimensional Gaussian $\mathcal{N}(\mu_{z_{ji}}, S^{-1}_{z_{ji}})$. However, in inference, it is necessary to group the observations of the features into groups of six, one for each feature. This is problematic: usually when conducting inference on six-dimensional data, each observation is already grouped into a six dimensional vector. This is not the case for these copy-number data, where each feature is generated using different processes. One observation of 'breakpoint number' (feature 1) is generated every move of the sliding window; one observation of 'segment copy-number' (feature 2) and 'size of segment' (feature 5) is generated per fragment; one observation of 'change point copy-number' (feature 3) and 'normalised distance from the centre of the chromosome' (feature 4) and 'oscillating copy-number length' is generated per breakpoint. The number of these observations for each is different. This means that there is no six-dimensional vectors to use as observations and no principled way to group them. Using random combinations of observations in each feature suffers from computational intractability since there is a large number of groups of six observations. Faced with these difficulties, other signature representations and algorithms were sought. For reasons described in section 2.1, NMF and HDP-LDA were chosen.

## 3.2 Fidelity of the generative model to real data

The second stage of the project required the construction of a generative model to generate copy-number mutation data. A good generative model should generate data that resemble real data. As discussed in section 2.1, the model uses a base-level set of Gaussian mixture models and a meta-level set. Genome observations are sampled at the base-level mixtures and signatures are deconvolved at the meta-level. An arbitrary number of base-level component Gaussians can therefore be chosen; it is therefore possible to model the real copy-number data arbitrarily accurately. Non-Gaussian base-level mixtures may even be used and may be more appropriate in some features where observations below 0 (all features) and/or above 1 are not possible. One could in this case use mixtures of Beta or Gamma distributions. In this implementation, however, observations produced by the base-level mixtures that are outside of the appropriate ranges are simply ignored and the meta-level mixtures are fitted only to observations only in the appropriate range.

Unexpectedly, the real copy-number data were placed under embargo until (very re-

cent) publication by our collaborators' publisher [28], and hence the appropriateness of the mixture could not be evaluated quantitatively, though some measures of the variability and values of the component weights of genomes and signatures can be observed by inspection of figure 3.

## 3.3 Evaluation and comparing NMF and HDP-LDA

Investigation of the NMF algorithm was carried out in three primary experiments. Real data can be supplied to a modified pipeline. The first experiment looked at a realistic scenario with a noisy genome and unknown number of signatures; the second studies the outcomes of an optimal scenario where the number of signatures is known; and another finds the optimal number of mixture components to fit to the observations, which can then be used to fit and deconvolve signatures.

### 3.3.1 NMF experiment 1 - Realistic scenario

The aim of this experiment was to generate a fixed catalogue of forty signatures from which to construct the corpus of genomes, use methods described in [4] to determine the optimal rank, and hence, the supposed optimal number of signatures.

The success of the algorithm can be measured in a number of ways. The first is to evaluate the reconstruction error - the Frobenius norm between the combined signature and exposure matrices ($S \times E$) and the original observed matrix ($M$). The average reconstruction error is compared to the mean reproducibility of the derived clusters, where reproducibility is the average silhouette width: the average cosine similarity of the cluster centre to each member of the cluster. The optimal rank is that which yields the lowest reconstruction error and highest reproducibility. There were 5 repeats for each rank.

Figure 8 (top right) shows that reproducibility steadily declines as rank increases and in figure 8 (top left) that the reconstruction error is lowest for rank=6. Higher reproducibilities of clusters and the lower reconstruction errors are preferred; this justifies the selection of rank 6. It suggests the algorithm was able to deconvolve at most six signatures from the almost 40 present in the corpus.

The deconvolved signatures were matched to their supposed original signatures by comparing each original signature to each deconvolved signature and assigning each deconvolved to the original signature with which it shares its maximum cosine similarity. Consequently, in some cases different deconvolved signatures may be assigned

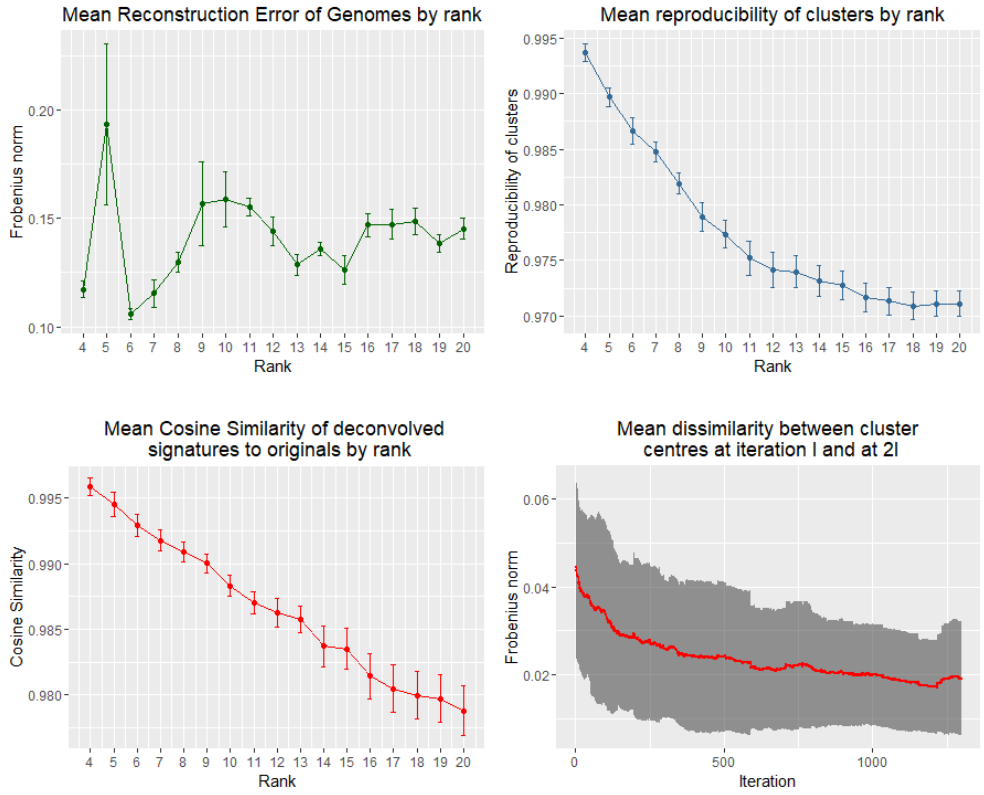


Figure 8: **Top Left**: Mean reconstruction error of deconvolved signatures and exposure matrices. **Top Right**: Mean reproducibility of deconvolved signature clusters. **Bottom Left**: Mean cosine similarity of original signatures to each other. **Bottom Right**: The averaged error (Frobenius norm) between the matrix of deconvolved signatures at iteration I and at 2I. Note the eventual convergence of algorithm. The 2I was capped at 1500 for all repetitions. In all, error bars are standard errors.

to the same original signature. Figure 8 (bottom left) shows that the average cosine similarity between matched signatures decreases approximately linearly as the number of deconvolved signatures increases, while standard error increases. Although the average cosine similarity for all ranks is high, it should be noted from figure 10 (top) that the similarity of original signatures to each other is also very high ($\mu = 0.8900$, $s = 0.09824$). The similarity of signatures to each other places a lower bound on the acceptable cosine similarity for deconvolved signatures, since deconvolved signatures with cosine similarity to original signatures that is below 0.8900 may be approximating any signature. To ensure the deconvolved signatures are deconvolved with sufficient accuracy that they reasonable a specific original signature, a rejection threshold of one standard deviation above the mean is used ($0.8900 + 0.09824 = 0.98824$).

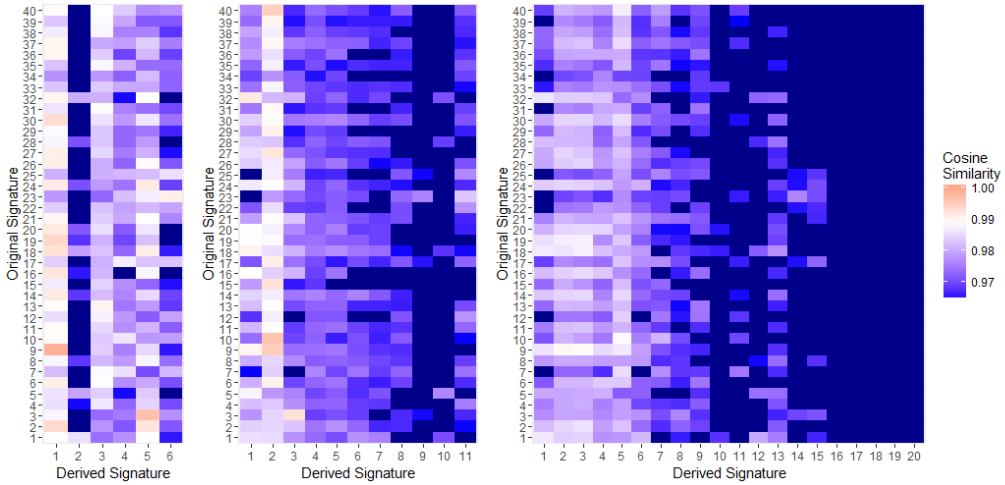


Figure 9: **Left**: Mean cosine similarity of original signatures to deconvolved signatures using rank=6. **Middle**: Mean cosine similarity of original signatures to deconvolved signatures using rank=11. **Right**: Mean cosine similarity of original signatures to deconvolved signatures using rank=20. In all, white corresponds to a cosine similarity of 0.988, and dark blue corresponds to $< 0.965$.

A comparison of individual deconvolved signatures to the originals is warranted. Recall that the optimal rank for the algorithm was rank 6. In figure 9, observe that rank 6 achieved some deconvolved signatures. However, it appears that they are not uniquely deconvolved, since they are above the threshold for several of the original signatures. Nevertheless, they are matched to the signature with which they share the largest cosine similarity. Other ranks were less successful at deconvolving signatures successfully. The matches can be found in table 3.3.1.

| Deconvolved Signature | Matched Original Signature | Cosine Similarity |
|---|---|---|
| 1 | 9 | 0.999 |
| 2 | 1 | 0.986 |
| 3 | 13 | 0.996 |
| 4 | 17 | 0.990 |
| 5 | 3 | 0.997 |
| 6 | 23 | 0.991 |

Table 1: Matches between deconvolved signatures (using rank 6) and original signatures with assigned cosine similarities.

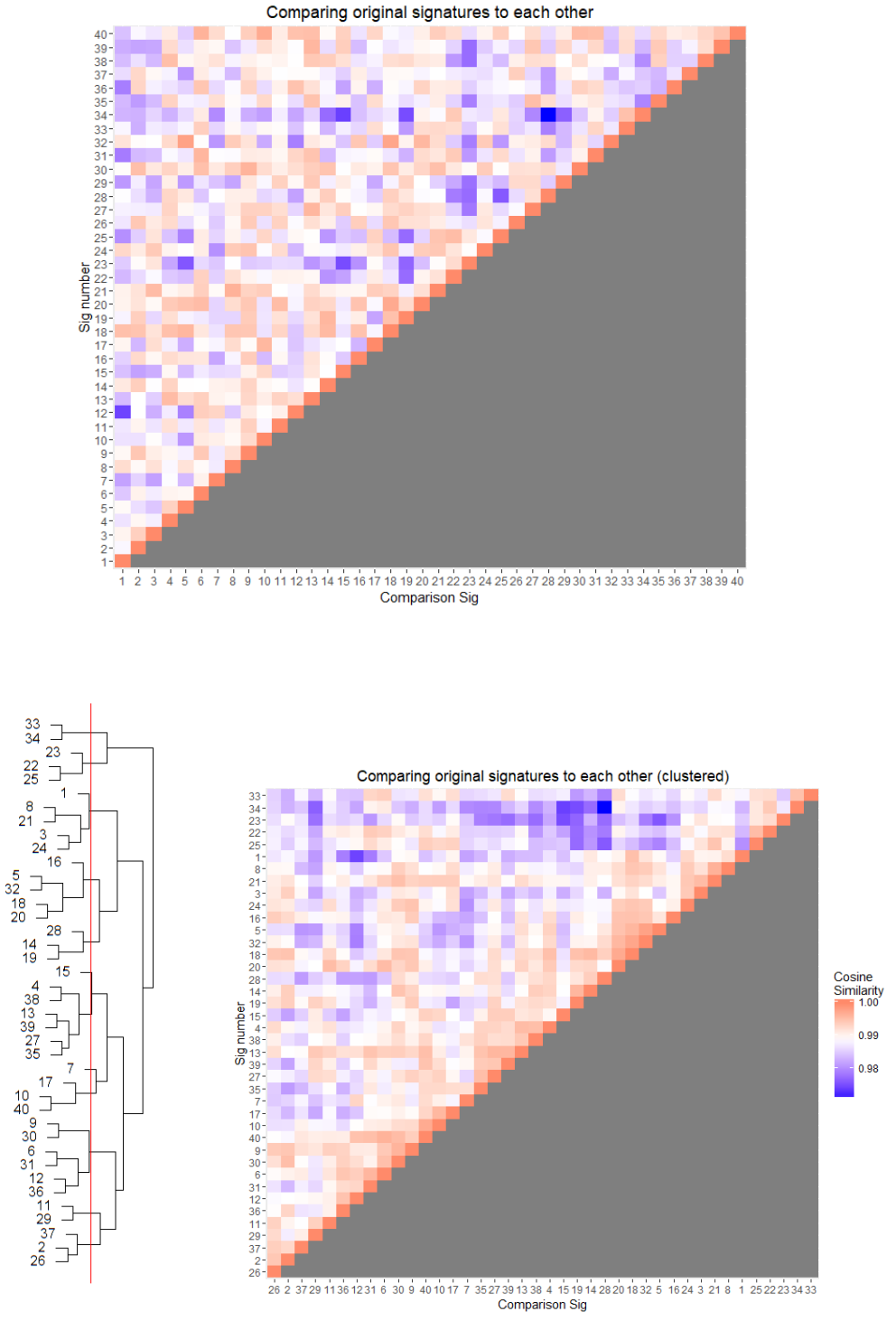


Figure 10: **Top**: Heatmap of cosine similarities between original signatures. **Bottom Left**: Dendrogram of clusters of original signatures cut at the level of 12 signatures. **Bottom Right**: Clustered heatmap of cosine similarities between original signatures.

It is noteworthy that not only does the match with original signature 1 not surpass the acceptability threshold of 0.988, but signatures with much 'lighter' exposure in the corpus, such as 17 or 23, are deconvolved instead. Since many of the original signatures are similar, it is possible that the weights of groups of very similar signatures combine such that the resulting deconvoluted signatures are more similar to the groups than to signature 1 and other heavily weighted signatures specifically. To investigate this possibility, the closest original signatures were clustered together and the deconvolved signatures were compared to the clusters of original signatures. Specifically, hierarchical clustering is used, where every data point is assigned to its own cluster, the two clusters that are closest together using Euclidean distance are identified, their cluster memberships are combined, and this process is repeated until all data are members of the same cluster. This yielded the dendrogram and clusters illustrated in figure 10 (bottom). The number of 12 clusters was chosen to retain a sufficient amount of the structure of the original data while nevertheless greatly reducing the number of different signatures. With more than 12 clusters, the group containing original signature 1 was not matched to one of the deconvolved signatures, motivating the selection of 12 clusters. The new 'cluster signatures' are taken as the centroids of the twelve clusters. The average between-cluster-signature cosine similarity is 0.9364 and the sample standard deviation is 0.06663. To use the same method as before to decide the rejection threshold would yield a threshold of 0.9974, which is extremely high. Being constrained to (0,1), most of the variance in cosine similarity exists below the mean. The rejection threshold is therefore arbitrarily relaxed to 0.99; discussion on how better to choose this threshold is left to the conclusions.

Table 2: Matches between deconvolved signatures and cluster-signatures

| Deconvolved Signature | Matched Cluster-Signature | Cosine Similarity |
|---|---|---|
| 1 | 2 | 0.9987 |
| 2 | 4 | 0.9857 |
| 3 | 3 | 0.9957 |
| 4 | 7 | 0.9895 |
| 5 | 1 | 0.9968 |
| 6 | 11 | 0.9912 |

In addition to the shape of the deconvolved signatures, the performance of the algorithm on the deconvolved exposures is also of interest. This is a harder problem for NMF than for the signatures: Although NMF has the attractive property of yielding a sparse factorisation of the original matrix[42], NMF is ill-posed. There

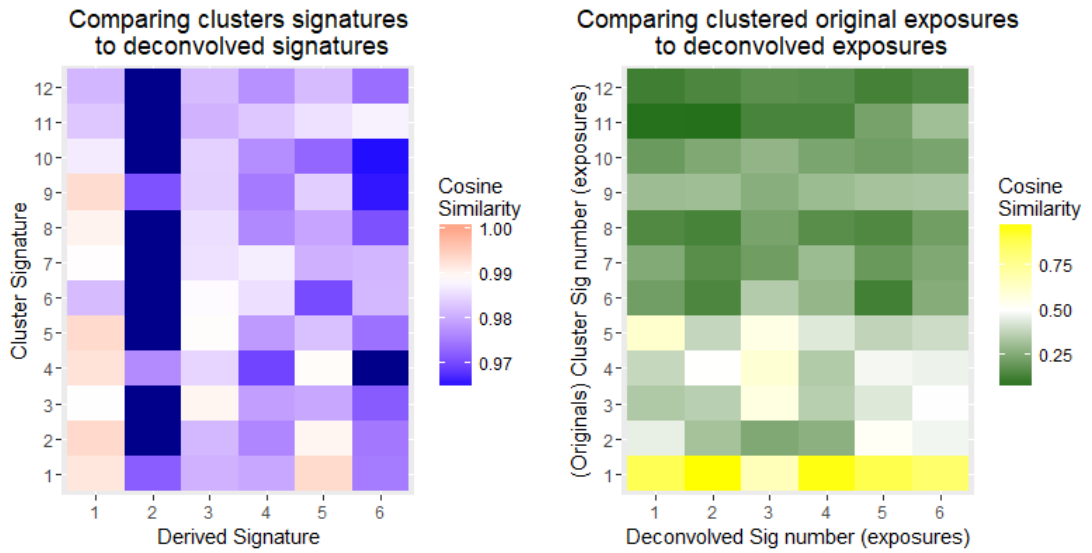


Figure 11: **Left**: Cosine similarity between cluster signatures and deconvolved signatures. **Right**:Cosine similarity between the cluster exposures and deconvolved exposures.

are many satisfactory solutions, and hence when all the solutions are aggregated as in algorithm 1, significant noise is introduced. The $40 \times 117$ signature exposure matrices are generally sparse, with only on average 4.08 non-zero entries per row. The deconvolved exposure matrices on the other hand consist entirely of non-zero elements that are larger than 0.01. Therefore the representation of the exposures matrix might benefit from being less sparse. Fortunately, this can be achieved by clustering.

The sparse original exposures were clustered using the cluster memberships defined by the signatures. The cluster-exposures are the centroids of those clusters. Figure 11 (right) shows the cosine similarity between the exposures of the original signatures and the deconvolved exposures. The only cluster signature with an accurately deconvolved exposure is cluster signature 1, which is composed of original signatures 1, 3, 8, 21, and 24. Although the number of clusters was chosen to be the maximum that permitted original signature 1 to be deconvolved, this cluster now comprises over 85.5% of the overall cluster signature-exposure in the corpus (figure 12).

When one cluster comprises such a large majority of the average exposure in the corpus, it becomes very difficult to deconvolve other, 'lighter' clusters. Nevertheless, deconvolved signature 1, 3, 5 and 6 surpass the cosine similarity threshold for clustered signatures of 0.99 and illustrated in figure 13. For comparison, the matched cluster signatures are plotted below in figure 14. The matches look ac-

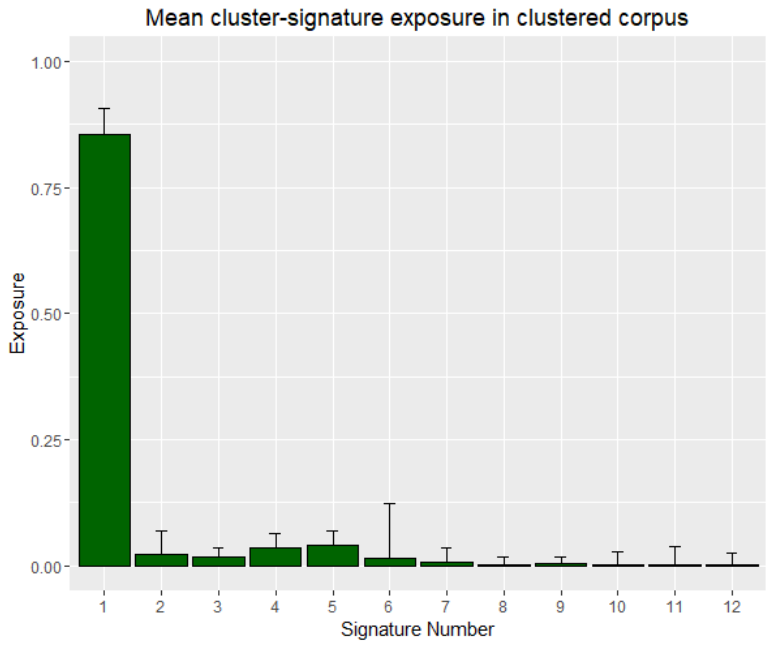


Figure 12: The average exposure of each cluster signature per genome in the corpus. Error bars are standard deviations (n=117).

ceptable, though there are large discrepancies in features 4 and 5. Additionally, note the difference in shape between figure 14 and the example signatures given in 5, which were generated using the base-level mixtures. The difference arises from this figure's observations being plotted using the meta-level mixtures instead of the base-level mixtures; since the deconvolution algorithm is acting on the meta-level, the meta-level was used here to plot observations to allow for direct comparison.

### 3.3.2 Experiment 2 - Investigating the optimal scenario for variable number of signatures

In experiment 2, a variable number of signatures is used in each deconvolution. Instead of using a corpus composed of 40 signatures, for each n in $2, \cdots, N$ a catalogue of signatures with n many signatures is constructed, and then a corpus of genomes for each is made. Then the signatures are deconvolved from the corpus using NMF (rank=$m$) in order to evaluate if all signatures could be derived with increasing genome complexity. There were 5 repeats for each rank.

Figure 15 (left) shows that as $m$ increases, the mean cosine similarity of the deconvolved signatures decreases. The mean falls below 0.988 at rank 9. Therefore, with true signatures are this similar, one should not expect algorithm 1 to be able to

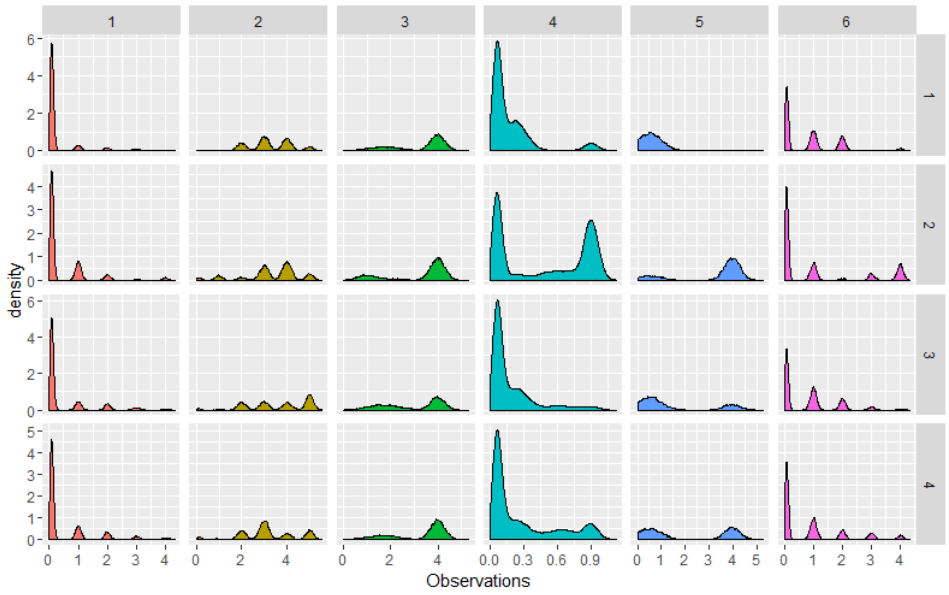


Figure 13: The four successfully deconvolved signatures. In order, deconvolved signature 1, 3, 5 and 6, corresponding to cluster signatures 2, 3, 1, and 11 respectively, plotted below.

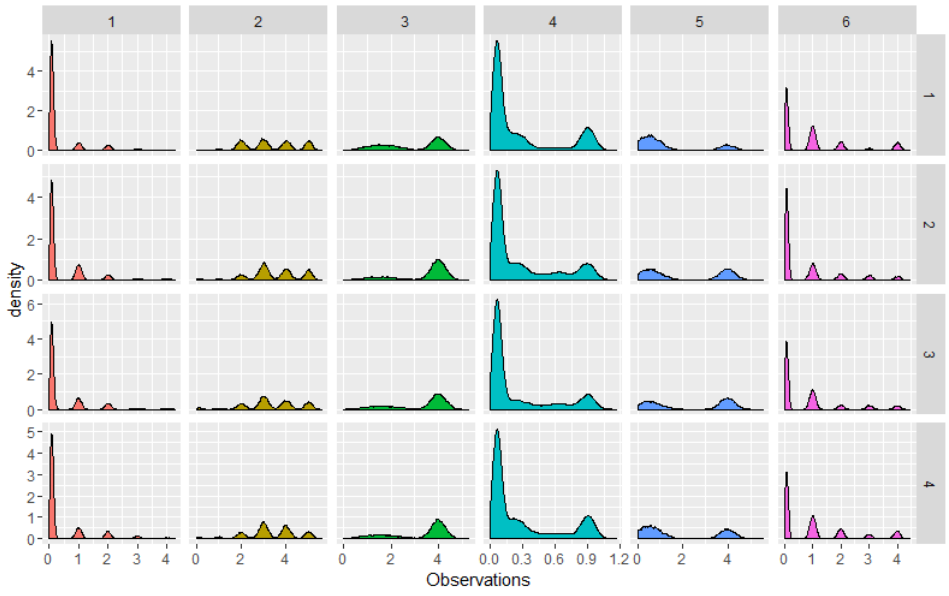


Figure 14: The cluster signatures 2, 3, 1, and 11 that were matched to the deconvolved signatures, plotted above.

reliably deconvolve more than 9 signatures under the best conditions.

Reconstruction error and average reproducibility of the clusters are used not to find the optimal rank, but to cross-evaluate the application of the method to the

case in experiment 1 with forty of signatures and variable rank of deconvolution. This experiment thus reveals artefacts produced by the algorithm. Note that the reconstruction error of approximately 0.4 is markedly higher and more variable than the reconstruction error evoked in experiment 1 with rank=6 with 40 signatures; this better reflects the variability produced in the genome generation process, since a new genome was generated for every repeat in experiment 2 but experiment 1 used only one genome.

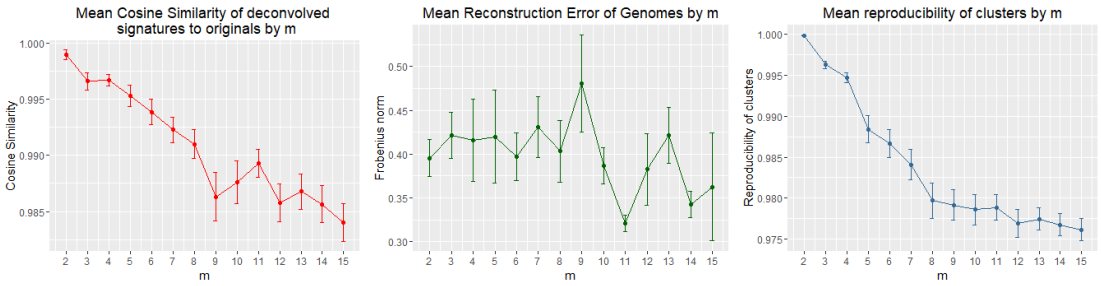


Figure 15: **Left**: Cosine similarity of derived signatures to original signatures when rank of NMF is equal to the number of signatures present in the corpus. **Middle**: Mean reconstruction error by number of signatures present in corpus. **Right**: Reproducibility of clusters by number of signatures present in corpus. In all, error bars are standard errors (n=5).

Observe the very similar pattern between experiments 1 and 2 of decreasing mean reproducibility and increasing standard error (comparing figures 8 and 15). This suggests that no matter how many signatures are present in a genome, the algorithm's accuracy degrades in a similar way and therefore, we might surmise, for the same reason. It appears to plateau at an average cosine similarity of 0.97 in experiment 1 and similarly in experiment 2. This is below the baseline cosine similarity between original signatures (0.988). This suggests that while rank increases, at least with these generated data, the deconvolved signatures begin to resemble a 'generic' original signature - one that might have been sampled from the original generation process - but with noise. Thus suggest that the factor limiting the accuracy of the deconvolution algorithm in both experiments is the limited by the accuracy of NMF. In both experiments, this issue is compounded by the fact that the higher the signature number, the less the corpus as a whole is exposed to that signature, thus making it harder to deconvolve.

### 3.3.3 NMF Experiment 3 - Optimising the number of components in feature mixture models

For variable number of meta-level components, the most appropriate number of components for each feature 3, 4, and 5 is sought. Features 1, 2, and 6 are ignored since the number of components required is the subset of integers they occupy.

Since these generated copy-number data have not been able to be calibrated to the real data, one cannot say that the results here resemble the results we would see on real data. But the example data shown here are illustrative of expected results when applying the analysis pipeline to real copy-number data.

For these generated data, the average cosine similarity of the deconvolved signatures varies in a different way for each figure, as could be expected for features exhibiting different distributions. This experiment provides a straightforward method to select the optimal number of components to use for each feature when studying real data.

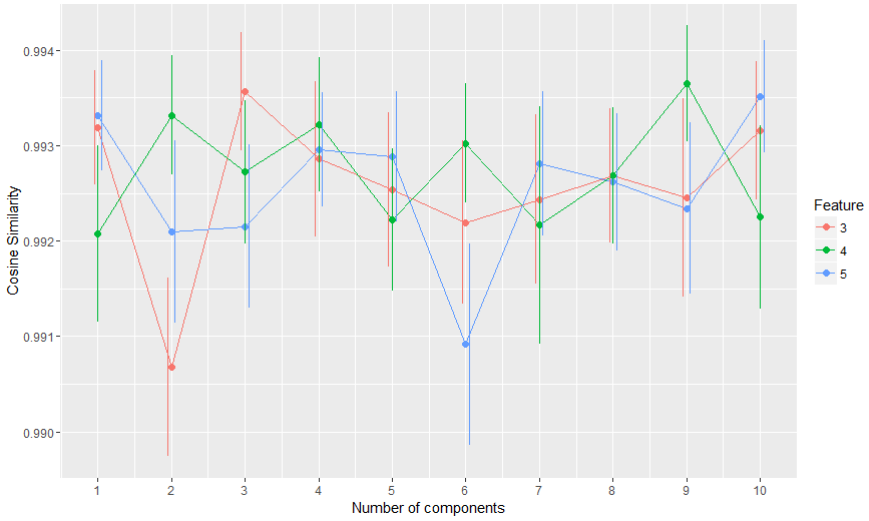


Figure 16: Average cosine similarities of deconvolved signatures with variable number of meta-level components for features 3, 4, and 5. Error bars and standard errors. (n=7)

## 3.4 Evaluating Hierarchical Dirichlet Process algorithm

While NMF is the gold standard algorithm for deconvolving signatures from single nucleotide variant mutations and insertion/deletions, other methods such as LDA

have been applied. Here the non-parametric generalisation of LDA - celled HDP - is applied to generated copy-number data.

The feature component weights are discretised by multiplying each feature component weight by 10'000 and assigning each component weight a unique string, grouped together to make a 'document'. These documents form the corpus that is supplied to the HDP algorithm.

Since an infinite mixture model is computationally intractable, online variational inference infers over a truncated, finite subset of signatures that may be much larger than the true number of signatures. There are two truncation levels for the number of signatures - the corpus-level truncation level, $T$, and the document level truncation level, $K$. Here $T = 40$ and $K = 10$ are used. In this implementation, all signatures are assigned a probability given the data, and the tolerance can be chosen in order to discard unlikely signatures. Surprisingly, the HDP algorithm assigned almost all the probability weight (92.1%) on only one signature which exhibits a high cosine similarity to all original signatures, particularly signature 1, the average cosine similarity being 0.9913 (see figure 17). The next most likely signature was assigned only a small probability (4.3%) and has an average cosine similarity to original signatures of 0.9246 and a maximum of 0.9358. This is below the average similarity of signatures to each other, therefore all signatures except the first can be disregarded.

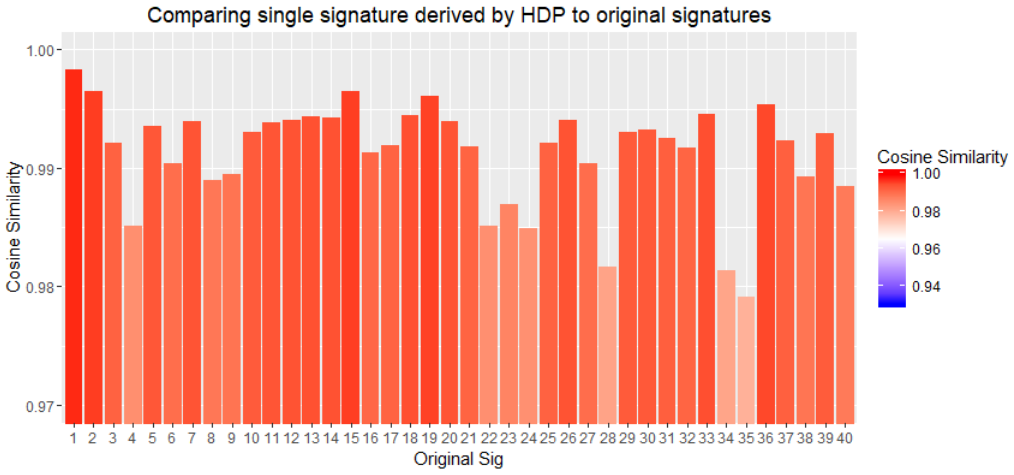


Figure 17: The cosine similarity between the single signature deconvolved by HDP. Note the truncated y-axis.

It is unclear why HDP is unable to distinguish between the signatures sufficiently to deconvolve more than one signature and, then, unable to distinguish a particular original signature. Several arguments can be made to explain the result:

**The small-corpus argument** It might be that the number of documents was too small to deconvolve signatures that are as similar as these. If true, and if real signatures are as similar as these generated ones, this problem would prevent the method's use on real data, since the number of sequenced genomes is typically small. Increasing the number of generated genomes to 2500 yielded similar results, the algorithm again proffering only one meaningful signature, with 99.6% probability weight being assigned to the signature and an average cosine similarity to the original signatures of 0.9916.

**The skewed weights argument** Another hypothesis is that the weight of signature 1 is overwhelming compared with other signatures. But figure 4 illustrates that other signatures are afforded non-trivial exposure weightings, suggesting that they should not be hard to deconvolve, at least not for this reason. Even if this were the case, one can expect to observe weighting of signatures like this in real data; therefore it is an essential requirement that a deconvolution algorithm overcome this difficulty.

**The small-vocabulary argument** It might be hypothesized that the vocabulary size is too small, and therefore too much importance is being loaded on the *proportions* of words to differentiate topics rather than being able to make use of large vocabularies with topics defined sparely over the vocabulary. But there appear to be no obvious theoretical grounds for the small-vocabulary argument.

**The small concentration parameters argument** A different hypothesis is that the concentration parameters of the HDP are too small, such that in the process of inference, the random generation of priors for signatures yields signatures that are too different from each other, when in fact all the signatures are very different. However, increasing the concentration parameters, $\gamma$ and *kappa*, from both being equal to 1 to being equal to 10e+20 (and several values within that range) did not change the number of signatures.

**The signature/genome similarity argument** In the absence of other, stronger hypotheses, the last (and least founded in the absence of further justification) hypothesis is that the signatures are simply not diverse enough to be able to separate the signatures. Another version of this argument is that the genomes, being weighted sums of genomes, are even more similar to each other on average than are signatures. This makes the task of deconvolving similar signatures in a dense (as opposed to

'sparse') vocabulary-space very difficult, and it may be too great a challenge for an approximate inference algorithm such as online variational inference [36].

None of these arguments are convincing however, and it remains unclear why HDP-LDA failed to deconvolve meaningful signatures.

# 4 Conclusions

In this project we evaluated algorithms for signature deconvolution of copy-number data. We built a generative model that resemble real data (qualitatively - due to constraints on data availability) and identified an appropriate deconvolution algorithm, NMF. An analysis pipeline was built around the generative model and the NMF deconvolution algorithm. We evaluated an approach using HDP-LDA and found it to be unsuitable for the task and explored possible reasons why. Therefore NMF is the preferred algorithm.

Nevertheless, the NMF deconvolution algorithm used here left room for improvement. It was incapable of deconvolving more than 5 acceptable signatures; the signatures it deconvolves were only good deconvolutions for a subset of features; and it struggles to produce sparse representations of the exposure matrix. That the algorithm struggled to deconvolve more than five signatures is unsurprising. Alexandrov et al. found that the number of genomes required to deconvolve a given number of signatures increased exponentially with the number of signatures[4]. They were able to deconvolve 20 signatures with 200 generated genomes; they were studying SNV-insertion-deletion mutations and, as a consequence, their genomes' basis vector representations appear to exhibit greater variability than our copy-number data representations. As such, one should expect to be able to deconvolve fewer than 20 signatures. One future direction to improve analysis of copy-number data is to find better representations of the mutations that do not suffer from such low variability. Indeed, in the recently published update of their analysis, MacIntyre et al. [28] have used alternative methods of preprocessing, yielding feature distributions to which they fitted either Gaussian or Poisson distributions. An additional approach might be to deconvolve signatures from each feature distribution individually and then study associations between deconvolved feature-signatures. To tackle the issue of sparsity, one could evaluate the use of augmented NMF methods that selects for even sparser matrices [43, 42].

A methodological weakness of the current approach is the arbitrary acceptability threshold used reject deconvolved signatures. Here we used the sum of the mean and standard deviation of the cosine similarities of signatures to each other; in the

case of cluster signatures, the threshold was chosen almost entirely arbitrarily. To improve this, one might instead fit a beta distribution to the cosine similarities of the signatures to each other, and define a quantile below which to reject deconvolved signatures.

In the absence of calibration to real data, it is difficult to make inferences about the applicability of the generative model and deconvolution algorithm to the underlying biology. However, the updated methods employed in [28] are broadly consistent with the approach taken here, even though they were conceived independently. Adapting the generative model and analysis pipeline to mixtures of Gaussian and Poisson distributions is straightforward and may provide a method to uncover the strengths and limitations of their statistical approach. This will help identify better methods of signature deconvolution, hopefully yielding a better understanding of copy-number mutational signatures and suggest avenues to improve clinical outcomes.

# References


[1] D. Graur, "An Upper Limit on the Functional Fraction of the Human Genome," *Genome Biology and Evolution*, vol. 9, pp. 1880–1885, jul 2017.

[2] B. D. Howard and I. Tessman, "Identification of the altered bases in mutated single-stranded DNA: III. Mutagenesis by ultraviolet light," *Journal of Molecular Biology*, vol. 9, pp. 372–375, aug 1964.

[3] S. Nik-Zainal, L. B. Alexandrov, D. C. Wedge, P. Van Loo, C. D. Greenman, K. Raine, D. Jones, J. Hinton, J. Marshall, L. A. Stebbings, A. Menzies, S. Martin, K. Leung, L. Chen, C. Leroy, M. Ramakrishna, R. Rance, K. W. Lau, L. J. Mudie, I. Varela, D. J. McBride, G. R. Bignell, S. L. Cooke, A. Shlien, J. Gamble, I. Whitmore, M. Maddison, P. S. Tarpey, H. R. Davies, E. Papaemmanuil, P. J. Stephens, S. McLaren, A. P. Butler, J. W. Teague, G. Jönsson, J. E. Garber, D. Silver, P. Miron, A. Fatima, S. Boyault, A. Langerød, A. Tutt, J. W. M. Martens, S. A. J. R. Aparicio, Å. Borg, A. V. Salomon, G. Thomas, A.-L. Børresen-Dale, A. L. Richardson, M. S. Neuberger, P. A. Futreal, P. J. Campbell, M. R. Stratton, and t. B. C. W. G. o. t. I. C. G. Breast Cancer Working Group of the International Cancer Genome Consortium, "Mutational processes molding the genomes of 21 breast cancers.," *Cell*, vol. 149, pp. 979–93, may 2012.

[4] L. B. Alexandrov, S. Nik-Zainal, D. C. Wedge, P. J. Campbell, and M. R. Stratton, "Deciphering signatures of mutational processes operative in human cancer.," *Cell reports*, vol. 3, pp. 246–59, jan 2013.

[5] L. B. Alexandrov, S. Nik-Zainal, D. C. Wedge, S. A. Aparicio, S. Behjati, A. V. Biankin, G. R. Bignell, N. Bolli, A. Borg, A. L. Børresen-Dale, S. Boyault, B. Burkhardt, A. P. Butler, C. Caldas, H. R. Davies, C. Desmedt, R. Eils, J. E. Eyfjörd, J. A. Foekens, M. Greaves, F. Hosoda, B. Hutter, T. Ilicic, S. Imbeaud, M. Imielinsk, N. Jäger, D. T. Jones, D. Jonas, S. Knappskog, M. Koo, S. R. Lakhani, C. López-Otín, S. Martin, N. C. Munshi, H. Nakamura, P. A. Northcott, M. Pajic, E. Papaemmanuil, A. Paradiso, J. V. Pearson, X. S. Puente, K. Raine, M. Ramakrishna, A. L. Richardson, J. Richter, P. Rosenstiel, M. Schlesner, T. N. Schumacher, P. N. Span, J. W. Teague, Y. Totoki, A. N. Tutt, R. Valdés-Mas, M. M. Van Buuren, L. Van 'T Veer, A. Vincent-Salomon, N. Waddell, L. R. Yates, J. Zucman-Rossi, P. Andrew Futreal, U. McDermott, P. Lichter, M. Meyerson, S. M. Grimmond, R. Siebert, E. Campo, T. Shibata, S. M. Pfister, P. J. Campbell, and M. R. Stratton, "Signatures of mutational processes in human cancer," *Nature*, vol. 500, pp. 415–421, aug 2013.

[6] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, oct 1999.

[7] L. B. Alexandrov and M. R. Stratton, "Mutational signatures: the patterns of somatic mutations hidden in cancer genomes," *Current Opinion in Genetics & Development*, vol. 24, pp. 52–60, feb 2014.

[8] T. Helleday, S. Eshtad, and S. Nik-Zainal, "Mechanisms underlying mutational signatures in human cancers," *Nature Reviews Genetics*, vol. 15, pp. 585–598, sep 2014.

[9] A. Baez-Ortega and K. Gori, "Computational approaches for discovery of mutational signatures in cancer," *Briefings in Bioinformatics*, jul 2017.

[10] L. Alexandrov, J. Kim, N. J. Haradhvala, M. N. Huang, A. W. T. Ng, A. Boot, K. R. Covington, D. A. Gordenin, E. Bergstrom, N. Lopez-Bigas, L. J. Klimczak, J. R. McPherson, S. Morganella, R. Sabarinathan, D. A. Wheeler, V. Mustonen, G. Getz, S. G. Rozen, M. R. Stratton, P. M. S. W. Group, and I. P.-C. A. o. W. G. Net, "The Repertoire of Mutational Signatures in Human Cancer," *bioRxiv*, p. 322859, may 2018.

[11] Wellcome Sanger Institute, "Signatures of mutational processes in human cancer," 2013.

[12] R. Rosenthal, N. McGranahan, J. Herrero, B. S. Taylor, and C. Swanton, "deconstructSigs: Delineating mutational processes in single tumors distinguishes DNA repair deficiencies and patterns of carcinoma evolution," *Genome Biology*, vol. 17, no. 1, 2016.

[13] P.-J. Huang, L.-Y. Chiu, C.-C. Lee, Y.-M. Yeh, K.-Y. Huang, C.-H. Chiu, and P. Tang, "mSignatureDB: a database for deciphering mutational signatures in human cancers," *Nucleic Acids Research*, vol. 46, 2018.

[14] S. L. Poon, S.-T. Pang, J. R. McPherson, W. Yu, K. K. Huang, P. Guan, W.-H. Weng, E. Y. Siew, Y. Liu, H. L. Heng, S. C. Chong, A. Gan, S. T. Tay, W. K. Lim, I. Cutcutache, D. Huang, L. D. Ler, M.-L. Nairismagi, M. H. Lee, Y.-H. Chang, K.-J. Yu, W. Chan-on, B.-K. Li, Y.-F. Yuan, C.-N. Qian, K.-F. Ng, C.-F. Wu, C.-L. Hsu, R. M. Bunte, M. R. Stratton, P. A. Futreal, W.-K. Sung, C.-K. Chuang, C. K. Ong, S. G. Rozen, P. Tan, and B. T. Teh, "Genome-Wide Mutational Signatures of Aristolochic Acid and Its Application as a Screening Tool," *Science Translational Medicine*, vol. 5, pp. 197ra101–197ra101, aug 2013.

[15] H. Davies, D. Glodzik, S. Morganella, L. R. Yates, J. Staaf, X. Zou, M. Ramakrishna, S. Martin, S. Boyault, A. M. Sieuwerts, P. T. Simpson, T. A. King, K. Raine, J. E. Eyfjord, G. Kong, Å. Borg, E. Birney, H. G. Stunnenberg, M. J. van de Vijver, A.-L. Børresen-Dale, J. W. M. Martens, P. N. Span, S. R. Lakhani, A. Vincent-Salomon, C. Sotiriou, A. Tutt, A. M. Thompson, S. Van

Laere, A. L. Richardson, A. Viari, P. J. Campbell, M. R. Stratton, and S. Nik-Zainal, "HRDetect is a predictor of BRCA1 and BRCA2 deficiency based on mutational signatures," *Nature Medicine*, vol. 23, pp. 517–525, apr 2017.

[16] E. M. Swisher, K. K. Lin, A. M. Oza, C. L. Scott, H. Giordano, J. Sun, G. E. Konecny, R. L. Coleman, A. V. Tinker, D. M. O'Malley, R. S. Kristeleit, L. Ma, K. M. Bell-McGuinn, J. D. Brenton, J. M. Cragun, A. Oaknin, I. Ray-Coquard, M. I. Harrell, E. Mann, S. H. Kaufmann, A. Floquet, A. Leary, T. C. Harding, S. Goble, L. Maloney, J. Isaacson, A. R. Allen, L. Rolfe, R. Yelensky, M. Raponi, and I. A. McNeish, "Rucaparib in relapsed, platinum-sensitive high-grade ovarian carcinoma (ARIEL2 Part 1): an international, multicentre, open-label, phase 2 trial," *The Lancet Oncology*, vol. 18, pp. 75–87, jan 2017.

[17] Y. K. Wang, A. Bashashati, M. S. Anglesio, D. R. Cochrane, D. S. Grewal, G. Ha, A. McPherson, H. M. Horlings, J. Senz, L. M. Prentice, A. N. Karnezis, D. Lai, M. R. Aniba, A. W. Zhang, K. Shumansky, C. Siu, A. Wan, M. K. McConechy, H. Li-Chang, A. Tone, D. Provencher, M. de Ladurantaye, H. Fleury, A. Okamoto, S. Yanagida, N. Yanaihara, M. Saito, A. J. Mungall, R. Moore, M. A. Marra, C. B. Gilks, A.-M. Mes-Masson, J. N. McAlpine, S. Aparicio, D. G. Huntsman, and S. P. Shah, "Genomic consequences of aberrant DNA repair mechanisms stratify ovarian cancer histotypes," *Nature Genetics*, vol. 49, pp. 856–865, apr 2017.

[18] T. Funnell, A. Zhang, Y.-J. Shiah, D. Grewal, R. Lesurf, S. McKinney, A. Bashashati, Y. K. Wang, P. Boutros, and S. Shah, "Integrated single-nucleotide and structural variation signatures of DNA-repair deficient human cancers," *bioRxiv*, p. 267500, feb 2018.

[19] G. Ciriello, M. L. Miller, B. A. Aksoy, Y. Senbabaoglu, N. Schultz, and C. Sander, "Emerging landscape of oncogenic signatures across human cancers," *Nature Genetics*, vol. 45, pp. 1127–1133, oct 2013.

[20] K. A. Hoadley, C. Yau, D. M. Wolf, A. D. Cherniack, D. Tamborero, S. Ng, M. D. M. Leiserson, B. Niu, M. D. McLellan, V. Uzunangelov, J. Zhang, C. Kandoth, R. Akbani, H. Shen, L. Omberg, A. Chu, A. A. Margolin, L. J. Van't Veer, N. Lopez-Bigas, P. W. Laird, B. J. Raphael, L. Ding, A. G. Robertson, L. A. Byers, G. B. Mills, J. N. Weinstein, C. Van Waes, Z. Chen, E. A. Collisson, C. C. Cancer Genome Atlas Research Network, C. C. Benz, C. M. Perou, and J. M. Stuart, "Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin.," *Cell*, vol. 158, pp. 929–944, aug 2014.

[21] B. McClintock, "The Stability of Broken Ends of Chromosomes in Zea Mays.," *Genetics*, vol. 26, pp. 234–82, mar 1941.

[22] A. Rode, K. K. Maass, K. V. Willmund, P. Lichter, and A. Ernst, "Chromothripsis in cancer cells: An update," *International Journal of Cancer*, vol. 138, pp. 2322–2333, may 2016.

[23] J. P. Murnane, "Telomere dysfunction and chromosome instability," *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, vol. 730, pp. 28–36, feb 2012.

[24] J. O. Korbel and P. J. Campbell, "Criteria for Inference of Chromothripsis in Cancer Genomes," *Cell*, vol. 152, pp. 1226–1236, mar 2013.

[25] F. Menghi, K. Inaki, X. Woo, P. A. Kumar, K. R. Grzeda, A. Malhotra, V. Yadav, H. Kim, E. J. Marquez, D. Ucar, P. T. Shreckengast, J. P. Wagner, G. MacIntyre, K. R. Murthy Karuturi, R. Scully, J. Keck, J. H. Chuang, and E. T. Liu, "The tandem duplicator phenotype as a distinct genomic configuration in cancer," *Proceedings of the National Academy of Sciences*, vol. 113, pp. E2373–E2382, apr 2016.

[26] C. K. Ng, S. L. Cooke, K. Howe, S. Newman, J. Xian, J. Temple, E. M. Batty, J. C. Pole, S. P. Langdon, P. A. Edwards, and J. D. Brenton, "The role of tandem duplicator phenotype in tumour evolution in high-grade serous ovarian cancer," *The Journal of Pathology*, vol. 226, pp. 703–712, apr 2012.

[27] G. Macintyre, T. E. Goranova, D. De Silva, D. Ennis, A. M. Piskorz, M. Eldridge, D. Sie, A. Lewsley, A. Hanif, C. Wilson, S. Dowson, R. M. Glasspool, M. Lockley, E. Brockbank, A. Montes, A. Walther, S. Sundar, R. Edmondson, G. D. Hall, A. Clamp, C. Gourley, M. Hall, C. Fotopoulou, H. Gabra, J. Paul, A. Supernat, D. Millan, A. Hoyle, G. Bryson, C. Nourse, L. Mincarelli, L. Navarro Sanchez, B. Ylstra, M. Jimenez, L. Moore, O. Hofmann, F. Markowetz, I. A. McNeish, J. D. Brenton, F. Markowetz FlorianMarkowetz, and I. McNeish, "Copynumber signatures and mutational processes in ovarian carcinoma," *bioRxiv*, 2018.

[28] G. Macintyre, T. E. Goranova, D. De Silva, D. Ennis, A. M. Piskorz, M. Eldridge, D. Sie, L.-A. Lewsley, A. Hanif, C. Wilson, S. Dowson, R. M. Glasspool, M. Lockley, E. Brockbank, A. Montes, A. Walther, S. Sundar, R. Edmondson, G. D. Hall, A. Clamp, C. Gourley, M. Hall, C. Fotopoulou, H. Gabra, J. Paul, A. Supernat, D. Millan, A. Hoyle, G. Bryson, C. Nourse, L. Mincarelli, L. N. Sanchez, B. Ylstra, M. Jimenez-Linan, L. Moore, O. Hofmann, F. Markowetz, I. A. McNeish, and J. D. Brenton, "Copy number signatures and mutational processes in ovarian carcinoma," *Nature Genetics*, p. 1, aug 2018.

[29] S. A. Vavasis, "On the Complexity of Nonnegative Matrix Factorization," *SIAM Journal on Optimization*, vol. 20, pp. 1364–1377, aug 2010.

[30] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, pp. 4164–9, mar 2004.

[31] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet processes," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2006.

[32] T. Broderick, M. I. Jordan, and J. Pitman, "Beta processes, stick-breaking, and power laws," *Bayesian Analysis*, vol. 7, pp. 439–476, 2012.

[33] Y. W. Teh, D. Newman, M. Welling, and D. Neaman, "A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation," *Advances in Neural Information Processing Systems 19 (NIPS 2006)*, pp. 1353–1360, 2007.

[34] I. Sato, K. Kurihara, and H. Nakagawa, "Practical collapsed variational bayes inference for hierarchical dirichlet process," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*, p. 105, 2012.

[35] A. Bleier, "Practical Collapsed Stochastic Variational Inference for the HDP," *Proceedings of the NIPS workshop on topic models*, p. arXiv:1312.0412 [cs.LG], 2013.

[36] C. Wang, J. Paisley, and D. M. Blei, "Online Variational Inference for the Hierarchical Dirichlet Process," *Icais*, vol. 15, pp. 752–760, 2011.

[37] blei lab, "blei-lab/online-hdp," 2011.

[38] M. I. Jordan and L. K. Saul, "An Introduction to Variational Methods for Graphical Models," *Machine Learning*, vol. 37, pp. 183–233, 1999.

[39] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, (Valletta, Malta), pp. 45–50, ELRA, May 2010. http://is.muni.cz/publication/884893/en.

[40] R. Gaujoux and C. Seoighe, "A flexible r package for nonnegative matrix factorization," *BMC Bioinformatics*, vol. 11, no. 1, p. 367, 2010.

[41] T. Benaglia, D. Chauveau, D. R. Hunter, and D. Young, "mixtools: An R package for analyzing finite mixture models," *Journal of Statistical Software*, vol. 32, no. 6, pp. 1–29, 2009.

[42] N. Gillis and N. G. Be, "Sparse and Unique Nonnegative Matrix Factorization Through Data Preprocessing," tech. rep., 2012.

[43] P. O. Hoyer, "Non-negative Matrix Factorization with Sparseness Constraints," tech. rep., 2004.

[44] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet allocation," *Journal of machine Learning Research*, vol. 3, no. Figure 1, pp. 993–1022, 2003.

[45] J. Foulds, L. Boyles, C. DuBois, P. Smyth, and M. Welling, "Stochastic Collapsed Variational Bayesian Inference for Latent Dirichlet Allocation," *Kdd*, pp. 446–454, 2013.

[46] J. Sethuraman, "A Constructive Definition of Dirichlet Priors," *Statistica Sinica*, vol. 4, pp. 639–650, 1994.

[47] D. Blackwell and J. B. MacQueen, "Ferguson Distributions Via Polya Urn Schemes," *The Annals of Statistics*, vol. 1, pp. 353–355, mar 1973.

[48] D. J. Aldous, "Exchangeability and related topics," pp. 1–198, Springer, Berlin, Heidelberg, 1985.

# 5 Appendix

## 5.1 An introduction to Latent Dirichlet Allocation

LDA is a three-level hierarchical Bayesian model with applications in sematic topic modelling [44]. Formally, with definitions of the objects in LDA given below:

- Word - an item from a vocabulary indexed by $1, \cdots, W$. They are represented by $W$-dimensional unit basis vectors (i.e. one-hot encoded). For example, the third word in the vocabulary is denoted $(0, 0, 1, 0, \cdots, 0)$. The $i^{\text{th}}$ word in the $j^{\text{th}}$ document is denoted $x_{ij}$. In the context of this project, a word is a component weight-increment.
- Document - A collection of words i.e. a genome $x_{\cdot j} = (x_1, \cdots, x_{n_{j\cdot\cdot}})$. The number of words in each document, $n_{j\cdot\cdot}$, may vary; in this project, the number of mutations happen to be equivalent in each genome.
- Corpus - A collection of $D$ documents, $x_{\cdot\cdot} = \boldsymbol{x}$.

- Topic and topic-assignment: Each word $x_{ij}$ is associated with a (latent) topic-assignment $z_{ij} \in 1, \cdots, K$. $z_{ij}$ is merely an indicator variable that selects the index of the topic $\phi_k$, which is a probability vector of length $W$ that defines the probability that a topic will yield each word in the vocabulary. Thus, given a topic $k$, the word $x_{ij}$ takes on value $w \in W$ with probability $\phi_{kw}$.

Also let $n_{jkw} = |\{i; x_{ij} = w, z_{ij} = k\}|$, with dot notation indicating summation: $n_{\cdot kw} = \sum_j n_{jkw}$. It follows that $n_{j\cdot\cdot}$ is the total number of words in document $j$.

Using these definitions, the generative LDA model is as follows.

1. Assume $K$ latent topics, each a Multinomial distribution over a vocabulary of size $W$. Let the probability of choosing a certain word $w$ given a certain topic $k$ have a Dirichlet prior thus

$$\phi_k = (\phi_{k1}, \cdots, \phi_{kW}) \sim Dir(\boldsymbol{\beta}).$$

2. For each document $j$, draw a mixing proportion

$$\theta_j = (\theta_{j1}, \cdots, \theta_{jK}) \sim Dir(\boldsymbol{\alpha})$$

   over $K$ topics. This is the mixing proportion of topics in this document.

3. To get one word,

   (a) First draw a topic

$$z_{ij} \sim Mul(1, \theta_j)$$

   with topic $k$ chosen with probability $\theta_{jk}$

   (b) Then, using the chosen topic $z_{ij}$, the $i$th word in the document is drawn

$$x_{ij} \sim Mul(1, \phi_{z_{ij}}).$$

   So $x_{ij}$ takes on value $w \in W$ with probability $\phi_{z_{ij}w}$.

Succinctly:

**Algorithm 2:** Generative model of Latent Dirichlet Allocation

Initialise vectors $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ where $\dim(\boldsymbol{\alpha}) = \dim(\boldsymbol{\beta}) = K$

**for** *each* $k$ *in* $(1, \cdots, K)$ **do**

  sample $\phi_k \sim \text{Dir}(\boldsymbol{\beta})$

**for** *each document* $j$ *in* $(1, \cdots, D)$ **do**

  sample or set $n_{j\cdot}$

  sample $\theta_j \sim Dir(\boldsymbol{\alpha})$

  **for** *each word* $i$ *in* $(1, \cdots, n_{j\cdot})$ **do**

    sample a topic-assignment $z_{ij} \sim Mu(1, \theta_j)$

    sample a word $x_{ij} \sim Mu(1, \phi_{z_{ij}})$

The LDA model employs the 'bag of words' assumption, where the order of the words in the documents does not matter. To this extent, words in a document are said to be 'exchangeable' random variables, and a key property of LDA is that the model is exchangeable at the levels of both words within a document and documents within a corpus. This allows inferences concerning topics to be made from a whole corpus, since documents too can share topics.

In general, LDA requires that the number of topics $K$ is set in advance, an important limitation for this project, since a model is sought that infers the correct number of signatures from the data. This motivates the non-parametric generalisation of LDA, HDP-LDA, introduced later.

The conditional joint probability density function for the LDA model, conditional on symmetric priors $\boldsymbol{\alpha}, \boldsymbol{\beta}$, is:

$$\begin{aligned} p(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{x} | \boldsymbol{\alpha}, \boldsymbol{\beta}) =& p(\boldsymbol{\phi}|\boldsymbol{\beta}) p(\boldsymbol{\theta}|\boldsymbol{\alpha}) p(\boldsymbol{z}|\boldsymbol{\phi}) p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\phi}) \\ =& \prod_{k=1}^{K} Dir(\phi_k; \boldsymbol{\beta}) \prod_{j=1}^{D} (Dir(\theta_j; \boldsymbol{\alpha}) \prod_{i=1}^{n_{j\cdot}} Mul(z_{ij}; 1, \theta_j) Mul(x_{ij}; 1, \phi_{z_{ij}})) \\ =& \prod_{j=1}^{D} (\frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K} \prod_{k=1}^{K} \theta_{jk}^{\alpha - 1 + n_{jk\cdot}}) \prod_{k=1}^{K} (\frac{\Gamma(W\beta)}{\Gamma(\beta)^W} \prod_{w=1}^{W} \phi_{kw}^{\beta - 1 + n_{\cdot kw}}) \end{aligned} \tag{3}$$

The full derivation may be found in Appendix section 5.5.

From the full joint probability density function, $\boldsymbol{\theta}$ may be marginalised out by integration and $\boldsymbol{z}$ marginalised out by summation to get the conditional probability distribution over the corpus, given $\boldsymbol{\alpha}, \boldsymbol{\beta}$. Thus, the posterior distribution is

$$p(\boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\phi} | \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{p(\boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{x} | \boldsymbol{\alpha}, \boldsymbol{\beta})}{p(\boldsymbol{x} | \boldsymbol{\alpha}, \boldsymbol{\beta})},$$

which is intractable. Yet a method of inference to infer the appropriate values of $\boldsymbol{z}, \boldsymbol{\theta}$, and $\boldsymbol{\phi}$, given $\boldsymbol{x}, \boldsymbol{\alpha}$, and $\boldsymbol{\beta}$, is still required. Analytic inference may be side-stepped using variational inference or collapsed Gibbs sampling. In variational inference, we maximise the variational lower bound on the intractable posterior distribution. Blei et al. [44] gave the original delineation of standard variational inference for LDA, though other more efficient methods have since been derived [33] [45]. Continuing with the development of the model, we now turn to an approach that will allow permit inference of the correct number of topics (signatures) using Bayesian non-parametrics, Dirichlet Process Mixture Models.

## 5.2 An introduction to Dirichlet processes and Dirichlet process mixture models

A *Dirichlet process*, denoted $DP(\alpha_0, G_0)$, is a "probability measure on a probability measure". Its two parameters are the scaling parameter $\alpha_0 > 0$ and the *base probability measure*, $G_0$. $G_0$ is a probability measure on the measurable space $(\Theta, \mathcal{B})$, where $\Theta$ is the sample space, and $\mathcal{B}$ is the set of all subsets of $\Theta$. Put simply, $G_0$ is the original distribution over the sample space. If the aim is to characterise an unknown probability distribution $G$, $G_0$ would be an initial guess at what it is, and we'd say that the true, unknown $G$ is a sample from $G \sim DP(\alpha_0, G_0)$. Alternatively, one might want a way to make a lot ($J$ many, say) of distributions that are like $G_0$ but not identical, so one would draw $G_j \sim DP(\alpha_0, G_0)$. Crucially, the similarity of the $G_j$s to $G_0$ can be adjusted by changing the concentration parameter $\alpha_0$; the larger the $\alpha_0$, the more similar the distributions $G_j$s are to distribution $G_0$.

Formally, then, where $G_0$ is the base probability measure:

**Definition: Dirichlet process** A Dirichlet Process $DP(\alpha_0, G_0)$ is the distribution of another random probability distribution $G$ over $(\Theta, \mathcal{B})$ such that for any finite, measurable partition of $\Theta$, $(A_1, \cdots, A_r)$,

$$(G(A_1), \cdots, G(A_r)) \sim Dir(\alpha_0 G_0(A_1), \cdots, \alpha_0 G_0(A_r)$$

Equivalently,

$$G \sim DP(\alpha_0, G_0).$$

Note now and later that $r$ is arbitrary and merely used to index the members of the partition.

It is known that draws from $DP(\alpha_0, G_0)$ are discrete (since they are point masses sampled from the base probability distribution $G_0$; this will become clearer later in the explanation), which makes it good for placing priors on mixture components in mixture modelling. DPs are good for separating observations into groups while still letting the groups share statistical strength.

Readers seeking a grounded definition of 'probability measure', of which extensive use has been made in the definition of the Dirichlet Process, are invited to read the definitions of 'Measurable space', '$\sigma$-algebra', 'Measure space' in the Auxiliary Definitions, section 5.4.

There are three major representations of DP mixture models (DPMM), each lending themselves to different interpretations: the stick-breaking construction, the Chinese Restaurant Franchise/Polya urn scheme, and the infinite limit of finite mixture models. Only the first two will be discussed here.

### 5.2.1 The Stick-Breaking Construction of Dirichlet process mixture models

The stick-breaking comes from Sethuraman (1994) [46]. First make an infinite set of samples $(V_k)_{k=1}^{\infty}$ where $V_k \sim Beta(1, \alpha_0)$. Therefore $\forall k(V_k \in (0, 1))$. Also draw an infinite set of 'atoms' $(\phi_k)_{k=1}^{\infty}$ where $\phi_k \sim G_0$; each $\phi_k$ is therefore a realisation from the sample space, $\Theta$. For illustration, say that $G_0 = Poi(\lambda_0)$. When one samples $\infty$ atoms from $G_0$ and plot a finite subset of the $\infty$ atoms (samples) on a histogram, it would look like the standard Poisson density plot. It is worth noting that $\phi_k$ may be multidimensional depending on the dimension of distribution $G_0$. Indeed, later a Dirichlet distribution will be used as the base distribution for copy-number signatures.

Because what is being constructed is a probability distribution, the total weight that can be applied to all atoms sums to 1. To 'break the sticks', therefore one divides a stick of unit length infinitely many times, and the weights correspond to the length of the broken segment of stick. Thus, the stick is broken:

$$\pi_k = V_k \prod_{l=1}^{k-1}(1 - V_l), \qquad \boldsymbol{\pi} = (\pi_k)_{k=1}^{\infty}.$$

For convenience, we say $\boldsymbol{\pi} \sim Stick(\alpha_0)$. Note that $\sum_{k=1}^{\infty} \pi_k = 1$

A formal characterisation of the stick-breaking construction of the DP relies on two further definitions: the *support* of a random variable, which is used in turn to

define a *degenerate distribution*. Introductions to these terms may be referred to as necessary in the Auxiliary Definitions, section 5.4.

Now the prerequisites have been discussed, we define the stick-breaking construction of the Dirichlet Process $G \sim DP(\alpha_0, G_0)$. The contribution of Sethuraman [46] was to show that

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}$$

where $\delta_{\phi_k}$ is the degenerate probability distribution (infinitely) concentrated at $\phi_k$.

Just as a specific realisation of random variable $X$ is often denoted $x$, we will soon assign $\theta_i$ to a random variable that may take specific values $\phi_k$.

It is now straightforward to see that the probability $G(\theta_i = \phi_k) = \pi_k$, which is a property central of the stick-breaking construction. Thus, in the example where infinitely many samples were taken from $\phi_k \sim G_0 = Poi(\lambda)$ to get $(\phi_k)_{k=1}^{\infty}$, one arrives at a $G$ that is defined by the above sum, where, for an example Poisson-sampled atom $\phi_k$, the probability measure $\delta_{\phi_k}$ is weighted by each $\pi_k$ and samples the value $\phi_k$ with probability 1. Now that $G$ has been defined, it can be seen that the following definitions are equivalent:

$$\begin{aligned} G \sim DP(\alpha_0, G_0) \quad &\leftrightarrow \quad (G(A_1), \cdots, G(A_r)) \sim Dir(\alpha_0 G_0(A_1), \cdots, \alpha_0 G_0(A_r) \\ &\leftrightarrow \quad G(\theta_i = \phi_k) = \pi_k \end{aligned} \tag{4}$$

A DPMM may be decribed more generally:

$$\begin{aligned} G &\sim DP(\alpha_0, G_0) \\ \theta_i &\sim G \\ x_i &\sim F(\theta_i) \end{aligned} \tag{5}$$

where $F(\theta_i)$ is the distribution of $x_i$ given $\theta_i$. If $G_0$ is a Dirichlet distribution and $F$ is multinomial, this has straightforward applications to LDA topic (signature) modelling. Due to G's stick-breaking construction, the factors $\theta_i$ take on values $\phi_k$ with probability $\pi_k$ where $\boldsymbol{\pi} \sim Stick(\alpha_0)$ as before. So the DPMM can be written equivalently as

$$\begin{aligned} \boldsymbol{\pi} &\sim Stick(\alpha_0), \\ \phi_k &\sim G_0, \qquad\qquad\qquad k = 1, \cdots, \infty \\ z_i &\sim \boldsymbol{\pi}, \\ x_i &\sim F(\phi_{z_i}) = F(\theta_i). \end{aligned} \tag{6}$$

### 5.2.2 The Chinese Restaurant Process/Polya Urn scheme for Dirichlet process mixture models

Another way to use the DP is the Chinese Restaurant Process/Polya urn scheme. This refers to draws from G, but doesn't refer to G directly. Blackwell and MacQueen [47]) showed that, $G$ is integrated out, we can arrive at a conditional distribution, called the Polya urn predictive rule

$$\theta_i|\theta_1, \cdots, \theta_{i-1}, \alpha_0, G_0 \sim \frac{\alpha_0}{i-1+\alpha_0} G_0 + \sum_{h=1}^{i-1} \frac{1}{i-1+\alpha_0} \delta_{\theta_h}$$

These conditional distributions can be interpreted by analogy to an urn. In the Polya urn scheme, each ball in the urn has a distinct colour and is associated with each atom. When a ball is drawn from the urn, it is placed back in the urn along with another ball of the same colour. Additional, with a probability *proportional* to $\alpha_0$, a new atom is created by drawing the new atom from $G_0$/adding a ball of a new color to the urn. This interpretation makes explicit both the discrete and clustering property of DPs.

This analogy is isomorphic to the Chinese Restaurant interpretation [48]: a new customer, $\theta_i$, arriving at a Chinese restaurant is assigned randomly to existing tables (atoms, $\phi_1, \phi_2, \cdots$) in proportion to the number of people already sitting at the table, and to a new table with probability proportional to $\alpha_0$.

The clustering property can be made more explicit: let $\phi_1, \cdots, \phi_K$ be the distinct values (tables) taken on by $\theta_1, \cdots, \theta_{i-1}$ (customers). Each $\phi_1, \cdots, \phi_K$ is unique, and it is worth being explicit that $(i-1)$ does not necessarily equal $K$. For each $k = 1, \cdots, K$, define $m_k = \sum_{h=1}^{k-1} \mathbb{1}_{\theta_{h'}=\phi_k}$, where $\theta_{h'}$ is the realisation of $\theta_h$. $m_k$ can therefore be viewed as the number of customers already sitting at table $k$. Thus

$$\theta_i|\theta_1, \cdots, \theta_{i-1}, \alpha_0, G_0 \sim \frac{\alpha_0}{i-1+\alpha_0} G_0 + \sum_{h=1}^{i-1} \frac{m_k}{i-1+\alpha_0} \delta_{\phi_h}$$

So for each $i$, as customer $i$ arrives, either

1. assign $\theta_i = \phi_k$ (with probability proportional to $m_k$, the number of customers already sitting at the table) OR
2. increment $K$, draw $\phi_K \sim G_0$ and set $\theta_i = \phi_K$.

A single level of hierarchy in a DPMM is insufficient to model $J$ documents that share a finite number ($K$) of signatures. To achieve this, an additional level of hierarchy must be introduced, discussed in appendix section 5.3.

## 5.3 An introduction to hierarchical Dirichlet processes

In order to introduce dependencies between different groups, one cannot use atoms from a parameterised base distribution (e.g. a continuous $G_0(\tau)$) as one would in a standard hierarchical Bayesian model, since draws from $DP(\alpha_0, G_0)$ are independent and share no atoms. Alternatively, one could require that $G_0$ be from a discrete parametric family, but this is overly restrictive. A HDP overcomes these issues. Let

$$\begin{aligned} G_0 &\sim DP(\gamma, H) \\ G_j &\sim DP(\alpha_0, G_0) \end{aligned}$$

H may be discrete or continuous, thus returning the desired flexibility. Also define a new set of stick-breaking weights, $\pi_{jk} \sim Stick(\alpha_0)$ for each document $j$.

### 5.3.1 Hierarchical Dirichlet process mixture models

Observations (words or component-weight increments) are organised into groups (documents or genomes) and are assumed exchangeable within a group. Also assume that groups are exchangeable. Observations are indexed within a group using $i$ and groups are indexed using $j$, as in $\boldsymbol{x}_j = (x_{j1}, x_{j2}, \cdots)$. Each observation is distributed as $x_{ji} \sim F(\theta_{ji})$. The factors for each group (and hence each observation) are defined as $\boldsymbol{\theta}_j = (\theta_{j1}, \theta_{j2}, \cdots)$ and are conditionally independent given $G_j$, the probability measure for each group, which itself is sampled from a DP parameterised by the global probability measure, $G_0$, which is in turn distributed by a DP parameterised by $\gamma$ and the baseline probability measure $H$. Therefore the complete formulation of the HDP is

$$\begin{aligned} G_0 &\sim DP(\gamma, H) \\ G_j &\sim DP(\alpha_0, G_0) \\ \theta_{ji} &\sim G_j && \text{for } j = 1, 2, \cdots \\ x_{ji} &\sim F(\theta_{ji}) && \text{for } i = 1, 2, \cdots \end{aligned} \tag{7}$$

The baseline probability measure $H$ provides the prior distribution for the factors $\theta_{ji}$, but $G_j$ ends up being the actual distribution over the factors in the $j^{th}$ group. If it is desired that the variability within different groups be different, then one can

specify a group concentration parameter, $\alpha_j$. Teh et al. [31] use a vague gamma prior for both $\gamma$ and $\alpha_0$.

As applied to the vanilla DP, one can now discuss the stick-breaking construction and Chinese Restaurant/Polya urn interpretations of the HDP.

### 5.3.2 Stick-breaking construction for the Hierarchical Dirichlet process

The atoms are distributed $\phi_k \sim H$. Thus $\boldsymbol{\phi} = (\phi_k)_{k=1}^{\infty}$. The top-level (global) DP weights are obtained by breaking sticks as before $\boldsymbol{\beta} = (\beta_k)_{k=1}^{\infty} \sim Stick(\gamma)$. Now since $G_0$ has support (valid possible samples) at $\boldsymbol{\phi} = (\phi_k)_{k=1}^{\infty}$ each $G_j$ necessarily has the same support; this is how atoms (signatures) are shared across documents (genomes). Thus

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\phi_k} \qquad\qquad G_j = \sum_{k=1}^{\infty} \pi_{jk} \delta_{\phi_k}$$

Teh et al. [31] showed that $G_j \sim DP(\alpha_0, G_0) \Rightarrow \boldsymbol{\pi}_j \sim DP(\alpha_0, \boldsymbol{\beta})$. This makes intuitive sense, since they share the same atoms, and only the weights can change when making $G_j$, which is a slightly different distribution from $G_0$.

Like in the non-hierarchical case, this leads to an equivalent representation of the HDPMM using the stick-breaking construction. Again, letting $z_{ji}$ be an indicator variable such that $\theta_{ji} = \phi_{z_{ji}}$, we have

$$\begin{aligned}
\phi_k &\sim H, && k = 1, \cdots, \infty \\
\boldsymbol{\beta} &\sim Stick(\gamma), \\
\boldsymbol{\pi}_j &\sim DP(\alpha_0, \boldsymbol{\beta}), \\
z_{ji} &\sim \boldsymbol{\pi}_j, \\
x_{ji} &\sim F(\phi_{z_{ji}})
\end{aligned} \tag{8}$$

which permits a plain interpretation of the model used in this project. As before, the Chinese Restaurant interpretation will afford us a way to see how the number of signatures might be inferred. Instead of a single Chinese restuarant, however, Teh et al. [31] generalised the scheme to multiple restaurants that shared a menu - a franchise.

### 5.3.3 The Chinese Restaurant Franchise

Multiple restaurants/groups/documents/genomes share the same possible set of menu of dishes/atoms/topics/signatures. The $i^{\text{th}}$ customer arrives at restaurant

$j$ (the customer is therefore $\theta_{ji}$) and selects the dish $\phi_k$ for the table from the menu $\phi_1, \cdots, \phi_K$. The actual dish served at table $t$ in restaurant $j$ is $\psi_{jt}$. Thus each customer $\theta_{ji}$ is associated with one dish-on-table $\psi_{jt}$ is associated with one dish-on-menu $\phi_k$. In particular, let

- $t_{ji}$ be the index of dish-on-table $\psi_{jt}$ associated with customer $\theta_{ji}$, i.e. $\theta_{ji} = \psi_{jt_{ji}}$
- $k_{jt}$ be the index of $\phi_k$ associated with $\psi_{jt}$, i.e. $\psi_{jt} = \phi_{k_{jt}}$.
- $z_{ji} = k_{jt_{ji}}$ be the mixture component associated with observation $x_{ji}$

Note that customer $i$ in restaurant $j$ sits at table $t_{ji}$, while table $t$ in restaurant $j$ serves $k_{jt}$.

Also,

- $n_{jtk}$ represents the number of customers in restaurant $j$ at table $t$ eating dish $k$
- $n_{jt\cdot}$ represents the number of customers in restaurant $j$ at table $t$.
- $n_{j\cdot k}$ represents the number of customers in restaurant j eating dish k
- $m_{jk}$ is the number of tables in restaurant $j$ serving dish $k$
- $m_{j\cdot}$ is the number of tables in restaurant $j$
- $m_{\cdot k}$ is the number of tables serving dish $k$.
- $m_{\cdot\cdot}$ is the number of tables occupied across all restaurants.

From the expression derived by Blackwell and MacQueen [47] for a single restaurant, if $G_j$ is integrated out, it yields a conditional distribution for $\theta_{ji}$. Originally, for the single Chinese restaurant, we had:

$$\theta_i|\theta_1, \cdots, \theta_{i-1}, \alpha_0, G_0 \sim \frac{\alpha_0}{i-1+\alpha_0} G_0 + \sum_{h=1}^{i-1} \frac{m_k}{i-1+\alpha_0} \delta_{\phi_h}$$

But for the Chinese restaurant franchise we now have

$$\theta_{ji}|\theta_{j1}, \cdots, \theta_{j,i-1}, \alpha_0, G_0 \sim \frac{\alpha_0}{i-1+\alpha_0} G_0 + \sum_{t=1}^{m_{j\cdot}} \frac{n_{jt\cdot}}{i-1+\alpha_0} \delta_{\psi_{jt}} \tag{9}$$

Note that $\phi_{k_{jt}} = \psi_{jt}$. Now for each customer $i$ arriving at the restaurant $j$ (i.e. for each $\theta_{ji}$) either:

1. Assign them to table $\theta_{ji} = \psi_{jt}$ with probability proportional $n_{jt\cdot}$, the number of people already sitting at that table (therefore also letting $t_{ji} = t$ for the assigned $t$ value.

2. Create a new table by incrementing $m_{j\cdot}$ (the number of tables in restaurant $j$), then draw $\psi_{j,m_{j\cdot}} \sim G_0$, set $\theta_{ji} = \psi_{j,m_{j\cdot}}$, and therefore also let $t_{ji} = m_{j\cdot}$

Next, apply the same principles and integrate out $G_0$, the next step above $G_j$ in the model hierarchy. Since $G_0$ is distributed according to a DP, apply the result from Blackwell and MacQueen again to get the conditional distribution of $\psi_{jt}$:

$$\psi_{jt}|\psi_{11}, \psi_{12}, \cdots, \psi_{21}, \cdots, \psi_{j,t-1}\gamma, H \sim \frac{\gamma}{m_{\cdot\cdot} + \gamma} H + \sum_{k=1}^{K} \frac{m_{\cdot k}}{m_{\cdot\cdot} + \gamma} \delta_{\phi_k} \tag{10}$$

This is where the number of signatures is incremented up in the model. Instead of assigning customers to tables, one is now assigning dishes-from-menu ($\phi_k$s) to dishes-on-tables ($\psi_{jt}$). Remember that a dish-on-table as already been assigned to each customer in the steps where we set $\theta_{ji} = \psi_{j,m_{j\cdot}}$ or $\theta_{ji} = \psi_{jt}$. So one needs to assign each $\psi_{jt}$ to a $\phi_k$. Therefore each restaurant $j$ and each table $t$, either

1. Assign $\psi_{jt} = \phi_k$ with probability proportional $m_{\cdot k}$, the number of times dish $k$ has already been assigned (and also letting $k_{jt} = k$ for the for the assigned $k$ value). Or

2. Create a new dish by incrementing $K$ (the number of different dishes actually being served on tables by the franchise), then draw $\phi_k \sim H$, set $\psi_{j,t} = \phi_k$, and therefore also let $k_{jt} = K$.

Thus, each $j, t, i, k$ is assigned a corresponding description: $\theta_{ji} = \psi_{jt} = \phi_k$. When using these equations to obtain samples of $\theta_{ji}$s, one uses Equation 9. While using Equation 9, if one needs to sample a new $\psi_{jt}$ from $G_0$ to assign to a $\theta_{ji}$, one uses Equation 10. In this way, we may infer the number of topics, or in this case mutational signatures, automatically.

## 5.4 Auxiliary definitions

**Definition: Exchangeability** The random vector $(x_1, \cdots, x_N)$ is exchangeable iff $P(x_1, \cdots, x_N) = P(x_{\pi(1)}, \cdots, x_{\pi(N)})$ where $\pi$ is some permutation of $1, \cdots, N$.

**Definition: Measurable space** A *measurable space*, e.g. $(\Theta, \mathcal{B})$, is a tuple consisting of a non-empty set $\Theta$ (e.g. the whole numbers, $\mathbb{N}$, but any non-empty set will do), and a corresponding $\sigma$-algebra of subsets of $\Theta$ e.g $\mathcal{B}$.

**Definition: $\sigma$-algebra** A $\sigma$-algebra defines the subsets of $\Theta$ that will be 'measured' - i.e. 'ordered' or 'compared'. A $\sigma$-algebra is required because if one want to measure/order subsets of a space, there may be ways to define subsets of that space that are, for most purposes, meaningless. The $\sigma$-algebra defines the subsets of interest. An example would be a $\sigma$-algebra $\mathcal{B}$ that subsets the whole numbers, $\mathbb{N}$ into $\{\{0\}, \{1\}, \{2\}, \{3\}, \{4\}, \cdots\}$ such as the number of goals scored by a football team, and not into

$$\{\{4, 205\}, \{72, 0, 2, 10\}, \{92'549'210\}, \cdots\}.$$

This permits the definition of a measurable space $(\mathbb{N}, \mathcal{B})$. Note that a measur*able* space should not be confused with *measure space*, defined next.

**Definition: Measure space** A triple $(\Theta, \mathcal{B}, \mu)$ consisting of a set and a corresponding $\sigma$-algebra of subsets as with a measurable space, but also with a *measure* $\mu$, which are only defined here informally: A *measure* is a way to assign a number to each subset of a measurable space corresponding to the 'size' of the subspace. For example, one could use a *probability measure* $P$ to map the measurable space $(\mathbb{N}, \mathcal{B})$ onto the interval $[0, 1]$, corresponding to the probability that the team will score a certain number of goals that season; being a probability measure, the numbers $P$ assigns to subsets of the event space needs to sum to exactly 1. Thus the measure space in this example would be $(\mathbb{N}, \mathcal{B}, P)$. Note that this probability measure might indeed be a probability distribution, such as a Poisson distribution.

But a measure does not need to be a probability measure; continuing the example, a football club may have set up an incentive scheme/a measure $\mu$, such that the manager is paid £1000 for every goal scored by the team that season. Thus the measure $\mu$ maps goals scored onto multiples of 1000, which is clearly not a probability measure as it neither sums to 1 nor is its co-domain the interval $[0, 1]$. Probability mass functions and probability density functions are both probability measures.

**Definition: Support** The support of a random variable is the set of possible sample values for which the distribution of the variable does **not** have a probability of zero. For example, a binomial distribution $Bin(\theta, n)$ has a support of $0, 1, \cdots, n$. As another example, the support of the Gaussian is the entire real line. So the support is *the set of all possible valid samples from the distribution.*

**Definition: Degenerate distribution** A degenerate distribution is an n-dimensional distribution in which at least one of the variables is a deterministic function of the others, i.e. *its support has dimension* $\boldsymbol{m < n}$.

Let $\delta_{\phi_k}(\cdot)$ be the probability measure *degenerate* at $\phi_k$. One may also find this called "the probability measure *concentrated* at exactly $\phi_k$". Specifically, it is *infinitely* concentrated at $\phi_k$ - sampling yields exactly $\phi_k$ every time.

In the **univariate case**, a degenerate distribution is therefore a deterministic distribution, where one may only sample one outcome with probability 1, and any other outcome (like tails on a double-headed coin) is has probability 0. Therefore the statement *"$\delta_{\phi_k}(\cdot)$ is the probability measure degenerate at $\phi_k$"* means that

$$\delta_{\phi_k}(\Phi) = \begin{cases} 1, & \text{where } \Phi = \phi_k \\ 0, & \text{otherwise} \end{cases}$$

In the **multivariate case**, for example, the random vector $(X, Y, Z)$ being characterised by $Y = 2X$, one may determine the probability $P((X, Y, Z) = (2, 1, 1))$ as essentially being defined by $P(X, Z)$ (or equivalently, by $P(Y, Z)$). This example is a case that is not used in the present explication of Dirichlet Processes but will be described for completeness. In this multivariate case, there is still scope for randomness in the random vector, whereas the Dirichlet Processes require that the random vector be degenerate in the same way as in the univariate case. This is allowed, since in the univariate case, $m = 0 < n = 1$ necessarily, but one may still define $m = 0$ even when $n > 1$. Alternatively, suppose the random vector $(X_1, X_2, X_3)$ can take values on the unit (3-1)-simplex, $\Delta_2$, i.e. $X_j > 0$ and $\sum X_j = 1$. Recall that $\phi_k \sim G_0$. Thus let $G_0$ be a 3-dimensional Dirichlet distribution, and also let $\phi_k = ((x_1, x_2, x_3)_k$ be a particular realisation of $(X_1, X_2, X_3)$. One can then say that

$$\delta_{(x_1,x_2,x_3)_k}(X_1, X_2, X_3) = \begin{cases} 1, & \text{where } (X_1, X_2, X_3) = (x_1, x_2, x_3)_k \\ 0, & \text{otherwise} \end{cases}.$$

This permits sampling of $\phi_k$'s that exhibit the properties desired for copy-number signatures - $\phi_k$'s (topics) that define exactly the probabilities of yielding a sample of a component weight-increments (words) from a vocabulary of $W$ component weight-increments.

## 5.5 Proofs

### 5.5.1 Proof of the conditional JPDF of LDA

$$
\begin{aligned}
& p(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{x} | \boldsymbol{\alpha}, \boldsymbol{\beta}) \\
=& p(\boldsymbol{\phi}|\boldsymbol{\beta}) p(\boldsymbol{\theta}|\boldsymbol{\alpha}) p(\boldsymbol{z}|\boldsymbol{\phi}) p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\phi}) \\
=& \prod_{k=1}^{K} Dir(\phi_k; \boldsymbol{\beta}) \prod_{j=1}^{D} (Dir(\theta_j; \boldsymbol{\alpha}) \prod_{i=1}^{n_{j\cdot\cdot}} Mul(z_{ij}; 1, \theta_j) Mul(x_{ij}; 1, \phi_{z_{ij}})) \\
=& \prod_{k=1}^{K} (\frac{\Gamma(W\beta)}{\Gamma(\beta)^W} \prod_{w=1}^{W} \phi_{kw}^{\beta-1}) \prod_{j=1}^{D} ((\frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K} \prod_{k=1}^{K} \theta_{jk}^{\alpha-1}) \\
& \qquad \prod_{i=1}^{n_{j\cdot\cdot}} ((\frac{\Gamma(1+1)}{\Gamma(z_{ij1}+1)\cdots\Gamma(z_{ijK}+1)} \prod_{k=1}^{K} \theta_{jk}^{z_{ijk}}) \\
& \qquad\qquad (\frac{\Gamma(1+1)}{\Gamma(x_{ij1}+1)\cdots\Gamma(x_{ijW}+1)} \prod_{w=1}^{W} \phi_{jw}^{x_{ijw}}))) \\
=& \prod_{k=1}^{K} (\frac{\Gamma(W\beta)}{\Gamma(\beta)^W} \prod_{w=1}^{W} \phi_{kw}^{\beta-1}) \prod_{j=1}^{D} ((\frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K} \prod_{k=1}^{K} \theta_{jk}^{\alpha-1}) \prod_{i=1}^{n_{j\cdot\cdot}} ((\prod_{k=1}^{K} \theta_{jk}^{z_{ijk}})(\prod_{w=1}^{W} \phi_{jw}^{x_{ijw}}))) \qquad (11) \\
& \qquad\qquad \text{because both are one-hot encoded} \\
=& \prod_{k=1}^{K} (\frac{\Gamma(W\beta)}{\Gamma(\beta)^W} \prod_{w=1}^{W} \phi_{kw}^{\beta-1}) \prod_{j=1}^{D} ((\frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K} \prod_{k=1}^{K} \theta_{jk}^{\alpha-1}) \prod_{k=1}^{K} \theta_{jk}^{\sum_i z_{ijk}} \prod_{w=1}^{W} \phi_{jw}^{\sum_i x_{ijw}}) \\
=& \prod_{k=1}^{K} (\frac{\Gamma(W\beta)}{\Gamma(\beta)^W} \prod_{w=1}^{W} \phi_{kw}^{\beta-1}) \prod_{j=1}^{D} ((\frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K} \prod_{k=1}^{K} \theta_{jk}^{\alpha-1}) \prod_{k=1}^{K} \theta_{jk}^{\sum_w n_{jkw}} \prod_{w=1}^{W} \phi_{jw}^{\sum_j n_{jkw}}) \\
=& \prod_{j=1}^{D} (\frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K} \prod_{k=1}^{K} \theta_{jk}^{\alpha-1+\sum_w n_{jkw}}) \prod_{k=1}^{K} (\frac{\Gamma(W\beta)}{\Gamma(\beta)^W} \prod_{w=1}^{W} \phi_{kw}^{\beta-1+\sum_j n_{jkw}}) \\
=& \prod_{j=1}^{D} (\frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K} \prod_{k=1}^{K} \theta_{jk}^{\alpha-1+n_{jk\cdot}}) \prod_{k=1}^{K} (\frac{\Gamma(W\beta)}{\Gamma(\beta)^W} \prod_{w=1}^{W} \phi_{kw}^{\beta-1+n_{\cdot kw}})
\end{aligned}
$$