Lecture Notes

Computational genomics

Giulio Caravagna[a,1]

[a]*Department of Mathematics, Informatics and Geosciences, University of Trieste, Italy*

MSc program in Data Science and Artificial Intelligence

Abstract—Notes from the lectures of the Computational Genomics course.

Keywords—*Cancer genomics, Tumour evolution, Bulk Sequencing, Bayesian Inference, Population Genetics*

Contents

List of Figures

1. Mutational Signatures

1.1. Mutational Processes

Mutational signatures are distinct patterns of mutations that result from specific biological processes or exposures. These patterns arise from a variety of causes, including environmental exposures (e.g., UV light, smoking), defective DNA repair mechanisms, or endogenous processes such as spontaneous deamination of cytosine. Each mutational signature reflects the history of *mutational processes* that have been active in the cancer cell, allowing us to infer the underlying mechanisms of tumour development.

Mutational processes can be broadly categorized into endogenous and exogenous sources.

1.1.1. Endogenous Mutational Processes

- Spontaneous deamination of 5-methylcytosine is a common cause of C>T transitions, which are abundant in signature 1, linked to aging.
- DNA replication errors can introduce mutations during cell division, particularly in rapidly dividing tissues.
- Defects in DNA repair mechanisms, such as homologous recombination deficiency (HRD), can give rise to characteristic patterns of mutations, including large insertions or deletions.

1.1.2. Exogenous Mutational Processes

- Ultraviolet (UV) light exposure causes C>T transitions, especially at dipyrimidine sites, a hallmark of signature 7, often found in melanoma.
- Tobacco smoke introduces bulky adducts to DNA, leading to G>T transversions, which are prominent in signature 4, associated with lung cancer.

1.2. Mutational Contexts

Mutational signatures are typically derived from *single nucleotide substitutions* (SBS), and they are classified based on six possible mutation types: C>A, C>G, C>T, T>A, T>C, and T>G. Each mutation type is further subdivided by the context of the flanking nucleotides, resulting in 96 possible trinucleotide combinations. These trinucleotide mutation patterns form the basis of the most commonly studied mutational signatures.

Mutational signatures can also incorporate other types of mutations:

- **Small insertions and deletions (indels)**: Some signatures are enriched for short deletions, particularly those associated with defective DNA mismatch repair.
- **Double base substitutions (DBS)**: Certain signatures exhibit mutations affecting two consecutive bases.
- **Structural variations**: Complex rearrangements and large-scale copy number changes can also contribute to unique mutational signatures, particularly in cancers driven by chromosomal instability.

To get more information you can visit: https://cancer.sanger.ac.uk/signatures/

1.2.1. Mutational Contexts and Probability Model

Mutational signatures are often characterized by `trinucleotide contexts`. For example, a C>T substitution can occur in different contexts depending on the neighboring nucleotides. Therefore, the mutation catalog V is often constructed to reflect mutations in 96 trinucleotide contexts (six possible base substitutions multiplied by four possible nucleotides flanking on each side).

We define a probability vector p_i, where each element represents the probability of observing mutation type i in a given signature. The matrix W contains these probabilities for each mutational signature, ensuring that:

$$\sum_{i=1}^{m} W_{ik} = 1 \quad \forall k$$

This ensures that the columns of W sum to 1, representing valid probability distributions over the mutation types.

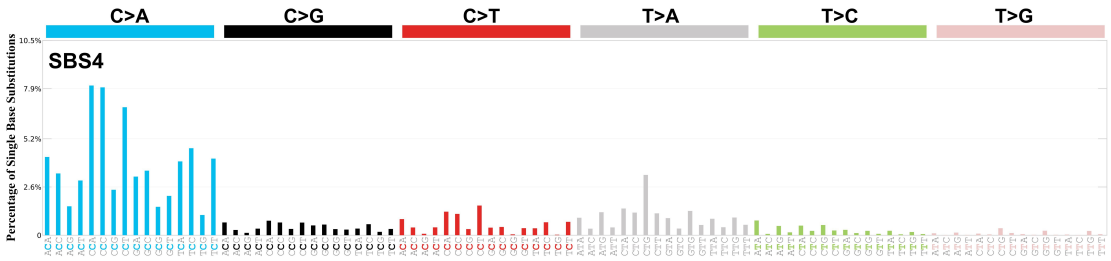


Figure 1. SBS4 mutational signature associated with smoking mutational process

Information

Several mutational signatures have been identified and cataloged in large-scale studies, such as those conducted by the **International Cancer Genome Consortium (ICGC)** and **The Cancer Genome Atlas (TCGA)**. These signatures are available in resources like the **COSMIC mutational signatures database**. Some well-known signatures include:

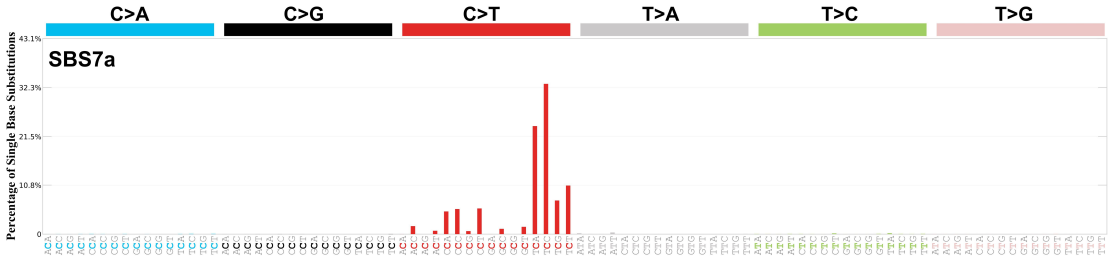


Figure 2. SBS7a mutational signature associated with UV-light mutational process

- Signature 1 (figure 1): Associated with spontaneous deamination of 5-methylcytosine, prevalent in many cancers and strongly linked to aging.
- Signature 4: Caused by tobacco smoke, characterized by G>T transversions.
- Signature 6: Reflecting defective DNA mismatch repair, associated with microsatellite instability and enriched in colorectal cancers.
- Signature 7a (figure 2): Linked to UV light exposure, dominant in skin cancers such as melanoma.

Each signature provides insight into the mutational processes that have shaped the tumor's genome, aiding in both basic research and clinical applications. The identification of mutational signatures in a tumor can have significant clinical implications:

- **Etiology**: Knowing the mutational processes active in a tumor can help infer the origins of the cancer. For example, the presence of signature 4 strongly suggests exposure to tobacco smoke, which may guide the patient's exposure history.
- **Therapeutic Targeting**: Some mutational signatures are associated with specific therapeutic vulnerabilities. For instance, tumors exhibiting signatures of homologous recombination deficiency (e.g., BRCA1/2 mutations) are more likely to respond to **PARP inhibitors**, a class of drugs that targets defects in DNA repair.
- **Prognosis**: The presence of certain signatures may correlate with better or worse outcomes. For example, the prevalence of mismatch repair deficiency (signature 6) in colorectal cancer is associated with better prognosis but also resistance to certain chemotherapies.

1.3. Signatures as matrix factorisation

The mutation catalog V represents the observed mutation counts in multiple tumor samples across different mutational contexts (e.g., trinucleotide substitutions). Let:

- m be the number of mutation types (e.g., 96 trinucleotide substitution types).
- n be the number of tumor samples.

Then, V is a non-negative matrix $V \in \mathbb{R}^{m \times n}$, where each element V_{ij} represents the count of mutation type i in tumor sample j. The goal of mutational signature deconvolution is to express V as the product of two non-negative matrices (figure 3):

1. $W \in \mathbb{R}^{m \times k}$: A matrix where each column corresponds to a mutational signature, representing the probability distribution of different mutation types for that signature.
2. $H \in \mathbb{R}^{k \times n}$: A matrix representing the exposure of each tumor sample to each mutational signature.

Thus, we aim to factorize V as:

$$V \approx WH$$

where:

- - k is the number of mutational signatures (typically much smaller than m).
- - W_{ik} represents the contribution of mutation type i to mutational signature k.
- - H_{kj} represents the contribution (exposure) of mutational signature k to tumor sample j.

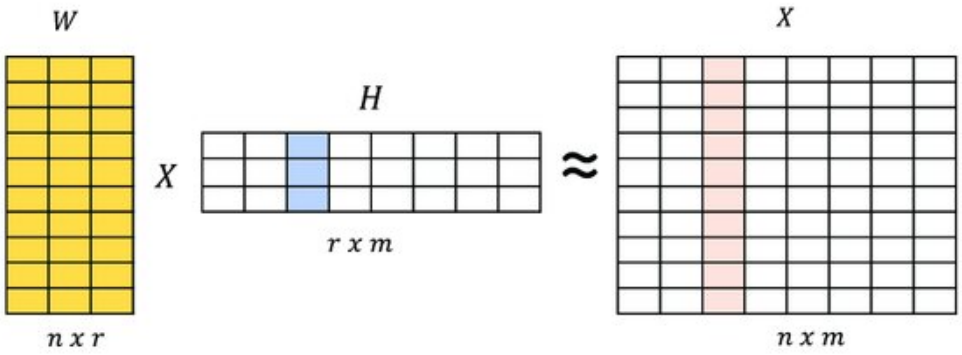


Figure 3. non-negative matrix factorization

Non-negative matrix factorization (NMF) seeks to solve the following optimization problem:

$$\min_{W,H\geq 0} \|V - WH\|_F^2$$

where $\|\cdot\|_F$ is the **Frobenius norm**, defined as:

$$\|A\|_F = \sqrt{\sum_{i,j} A_{ij}^2}$$

The objective is to minimize the reconstruction error between the observed mutation catalog V and the product of W and H, while ensuring that both W and H are non-negative (since mutation counts cannot be negative).

1.4. Solving the model

Multiplicative Update Rules One popular algorithm for solving NMF is through **multiplicative update rules**. The update rules for W and H are given by:

$$H_{kj} \leftarrow H_{kj} \frac{(W^T V)_{kj}}{(W^T WH)_{kj}}$$

$$W_{ik} \leftarrow W_{ik} \frac{(VH^T)_{ik}}{(WHH^T)_{ik}}$$

These update rules are iterated until convergence, ensuring that the matrices remain non-negative at each step.

Bayesian Framework for NMF To introduce uncertainty estimates and regularization into the NMF framework, we can formulate the problem in a **Bayesian** context. In this approach, we model the matrices W and H as random variables and place prior distributions over them.

1.4.1. Generative Model

Let the observed mutation counts V_{ij} be modeled as Poisson-distributed variables:

$$V_{ij} \sim \text{Poisson}\left((WH)_{ij}\right)$$

This Poisson distribution is appropriate because mutation counts are discrete and non-negative.

We place **Dirichlet priors** on both W and H :

$$W_i \sim \text{Dirichlet}(\gamma), \quad H_j \sim \text{Dirichlet}(\omega)$$

where γ, ω are hyperparameters that control the shape Dirichlet distributions.

both signatures and relative exposure matrices include values between 0 and 1, so the dirichelt distrubtuin fits well to describe them.

1.4.2. Posterior Inference

The goal is to compute the **posterior distribution** of W and H, given the observed data V:

$$P(W, H|V) \propto P(V|W, H)P(W)P(H)$$

where:

- $P(V|W, H)$ is the likelihood, modeled by the Poisson distribution.
- $P(W)$ and $P(H)$ are the Dirichlet priors.

$$p(V, W, H) = \prod_{\forall n} \prod_{\forall m} p(x_{nm} \mid \lambda) \prod_{\forall n} p(H_n) \prod_{\forall k} p(W_k)$$

You can find the probabilicts graphical model of the bayesian non negative matrix factorization in figure 4. here exposure and signatures matricers are represented by α and β respectively. λ is the possion distribution parameter and x is data. we have N samples and F features (e.g., for SBS case F is 96)

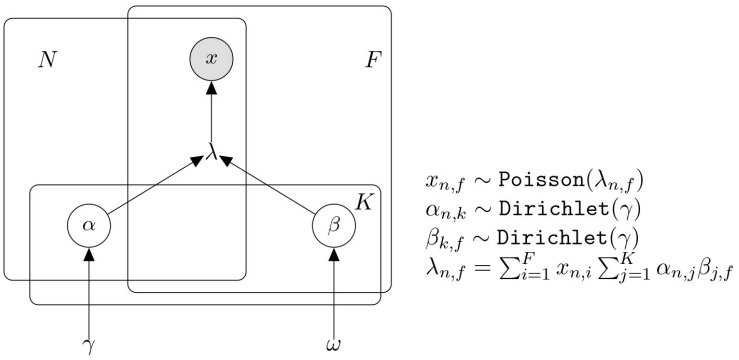


Figure 4. Bayesian non-negative matrix factorization probabilistic graphical model

1.4.3. kullback-liebler divergence

In mathematical statistics, the Kullback–Leibler (KL) divergence (also called relative entropy and I-divergence), denoted $\mathcal{KL}(P \parallel Q)$, is a type of statistical distance: a measure of how one reference probability distribution P is different from a second probability distribution Q. Mathematically, it is defined as:

$$\mathcal{KL}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right)$$

1.4.4. Variational Inference

Variational inference is an umbrella term for algorithms which computes the posterior $p(\mathbf{z} \mid \mathbf{x})$ by casting the intractable integration into an optimization problem (figure 5). Assume $p(\mathbf{z} \mid \mathbf{x})$ as the true posterior distribution, and $q(\mathbf{z} \mid \phi)$ being the approximate distribution, called variational distribution. we approximate that complicated posterior $p(\mathbf{z} \mid \mathbf{x})$ with a simpler distribution $q(\mathbf{z} \mid \phi)$ by minimizing the **kullback-liebler** divergence between $q(\mathbf{z} \mid \phi)$ and $p(\mathbf{z} \mid \mathbf{x})$ by changing the values of ϕ called variational parameters.

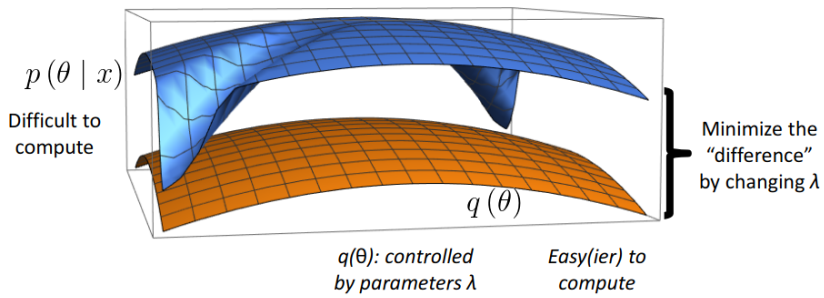


Figure 5. variational inference schematic

If we are trying to find the best approximated distribution $q^*(z)$ using variational Bayes, we define the following objective function :

$$q^*(z) = \arg\min_q \mathcal{KL}\big[q(z) \parallel p(z \mid x)\big]$$

By definition, $\mathcal{KL}$ divergence between two continuous distributions $q(z)$ and $p(z \mid x)$ is defined as :

$$\mathcal{KL}\big[q(z) \parallel p(z \mid x)\big] = \int_{-\infty}^{\infty} q(z) \log \frac{q(z)}{p(z \mid x)} dx = \mathbb{E}_q\big[\log q(z)\big] - \mathbb{E}_q\big[\log p(z \mid x)\big]$$

So for the KL divergence between variational distribution and posterior distribution we have:

$\mathcal{KL}[q(z) \,||\, p(z \mid x)] = \mathbb{E}_q[\log \mathbf{q}(\mathbf{z})] - \mathbb{E}_q[\log \mathbf{p}(\mathbf{z} \mid \mathbf{x})]$
$\mathcal{KL}[q(z) \,||\, p(z \mid x)] = \mathbb{E}_q[\log \mathbf{q}(\mathbf{z})] - \mathbb{E}_q[\mathbf{p}(\mathbf{z}, \mathbf{x})] + \log \mathbf{p}(\mathbf{x})$
$\mathcal{KL}[q(z) \,||\, p(z \mid x)] = -\big(\mathbb{E}_q[\log \mathbf{q}(\mathbf{z})] - \mathbb{E}_q[\mathbf{p}(\mathbf{z}, \mathbf{x})]\big) + \log \mathbf{p}(\mathbf{x})$
$\mathcal{KL}[q(z) \,||\, p(z \mid x)] = -\text{ELBO} + \log \mathbf{p}(\mathbf{x})$

This is the negative ELBO plus the log marginal probability of x

According to Jensen inequality as applied to probability distributions. When f is concave (log is a concave function)

$$f(\mathbb{E}[X]) \geq \mathbb{E}[f(X)]$$

We use Jensen's inequality on the log probability of the observations,

$\log p(x) = \log \int_z p(x, z)$ integrating over z gives the marginal distribution of x
$\log p(x) = \log \int_z p(x, z) \frac{q(z)}{q(z)}$ multiply and divide by $q(z)$
$\log p(x) = \log \int_z q(z) \frac{p(x,z)}{q(z)}$ reordering the terms
$\log p(x) = \log (\mathbb{E}_{q(z)}[\frac{p(x,z)}{q(z)}]) \geq \mathbb{E}_{q(z)}[\log \frac{p(x,z)}{q(z)}] = \text{ELBO(q)}$

Notice that $\log p(x)$ does not depend on q. So, as a function of the variational distribution, minimizing the KL divergence is the same as maximizing the ELBO. And, the difference between the ELBO and the KL divergence is the log normalizer - which is what the ELBO bounds.

1.4.5. Pyro Implementation

NOTE: this code snippet just gives a high level intuition about NMF implemetaion in pyro, and It's not a proper working code.

```
# x : data
import pyro
from pyro.infer import SVI, Trace_ELBO

def model():

   with pyro.plate("n", n_samples):
      alpha = pyro.sample( "latent_exposure", dist.Dirichlet(alpha_conc) )

   with pyro.plate("context", n_signatures):
      beta = pyro.sample( "latent_signatures", dist.Dirichlet(beta_conc) )

   a = torch.matmul(torch.matmul(torch.diag(torch.sum(self.x, axis=1)), alpha), beta)

   with pyro.plate("contexts", n_contexts):
      with pyro.plate("n2", n_samples):
         pyro.sample("obs", dist.Poisson(a), obs=self.x)

def guide():

   with pyro.plate("n", n_samples):
      alpha = pyro.param("alpha", alpha0)
      pyro.sample("latent_exposure", dist.Delta(alpha))

   with pyro.plate("n", n_signatures):
      beta = pyro.param("beta_denovo", beta0)
      pyro.sample("latent_signatures", dist.Delta(beta))


optimizer = Adam(adam_params)
svi = SVI(model, guide, optimizer, loss=Trace_ELBO())

n_steps = 5000
for step in range(n_steps):
   svi.step(data)

```

- observation : `pyto.sample` with the `obs`
- latent random variables : `pyto.sample`
- parameters : `pyto.param`

- define the likelihhod (observed data) and prior distrbution (latent variables) in `model()` function.
- define the variational distribtion and parameters in `guide()` function.

then we set the optimizer (e.g., `Adam`) and create the variational inference model using `SVI` function. To run the model we perform the several iterations using `for` loop where `svi.step` is one step of the gradient descent algorithm.

1.4.6. Adding Regularization: Sparsity Constraints

In many biological cases, not all mutational signatures are expected to contribute to every tumor sample. Therefore, it is often desirable to impose **sparsity constraints** on the exposure matrix H. One way to achieve this is by adding an L_1-regularization term to the objective function:

$$\min_{W,H\geq 0} \|V - WH\|_F^2 + \lambda \sum_{k,j} |H_{kj}|$$

where λ is a regularization parameter that controls the sparsity of the exposures. This encourages many of the entries in H to be zero, reflecting the assumption that only a few mutational signatures are active in each tumor.

Information

Consider a tumor sample with a high mutation burden. Using Bayesian NMF, we may find that this sample has a high exposure to **signature 4** (tobacco smoking-related mutations) and **signature 1** (age-related mutations). The matrix H might look like below wher ethe rows are samples and columns are signatures (SBS1, SBS3, SBS4 respectively):

$$H = \begin{pmatrix} 0.2 & 0.1 & 0.7 \\ 0.8 & 0.05 & 0.15 \\ 0.1 & 0.0 & 0.9 \end{pmatrix}$$

This indicates that the first sample has 70% exposure to signature 4, 20% exposure to signature 1, and 10% to another signature, while the second sample has a high exposure to signature 1 and very little to others.

Information

The multiplicative update rules for Non-Negative Matrix Factorization (NMF) are designed to iteratively minimize the reconstruction error between the data matrix V and the product WH, subject to the constraint that both matrices W and H remain non-negative. The objective function is typically the **Frobenius norm** of the difference between V and WH, which is minimized using an alternating update approach.

The objective is to minimize:

$$\min_{W,H\geq 0} \|V - WH\|_F^2 = \min_{W,H\geq 0} \sum_{i,j} (V_{ij} - (WH)_{ij})^2$$

where $V \in \mathbb{R}^{m\times n}$, $W \in \mathbb{R}^{m\times k}$, and $H \in \mathbb{R}^{k\times n}$. We aim to prove the correctness of the following multiplicative update rules for W and H:

1. **Update rule for** H:

$$H_{kj} \leftarrow H_{kj} \frac{(W^T V)_{kj}}{(W^T WH)_{kj}}$$

2. **Update rule for** W:

$$W_{ik} \leftarrow W_{ik} \frac{(VH^T)_{ik}}{(WHH^T)_{ik}}$$

We will prove the correctness of the multiplicative update rules using an approach based on the **Karush-Kuhn-Tucker (KKT) conditions** for constrained optimization. The objective function for minimizing the reconstruction error is the Frobenius norm:

$$f(W,H) = \frac{1}{2} \sum_{i,j} (V_{ij} - (WH)_{ij})^2$$

The partial derivative of $f(W,H)$ with respect to an individual element H_{kj} is:

$$\frac{\partial f}{\partial H_{kj}} = \sum_i W_{ik} \left((WH)_{ij} - V_{ij} \right)$$

Similarly, the partial derivative of $f(W,H)$ with respect to an individual element W_{ik} is:

$$\frac{\partial f}{\partial W_{ik}} = \sum_j H_{kj} \left((WH)_{ij} - V_{ij} \right)$$

To minimize the objective function, we want to set these gradients to zero, subject to the non-negativity constraints $W \geq 0$ and $H \geq 0$.

The KKT conditions provide necessary conditions for a solution to be optimal in constrained optimization problems. For non-negative matrix factorization, the KKT conditions tell us that at the optimal solution, we must have:

$$H_{kj} \left(\frac{\partial f}{\partial H_{kj}} \right) = 0, \quad H_{kj} \geq 0, \quad \frac{\partial f}{\partial H_{kj}} \geq 0$$

$$W_{ik} \left(\frac{\partial f}{\partial W_{ik}} \right) = 0, \quad W_{ik} \geq 0, \quad \frac{\partial f}{\partial W_{ik}} \geq 0$$

The multiplicative update rules are constructed to satisfy these KKT conditions. To show this, let us derive the update rule for H and

prove that it leads to a decrease in the objective function.

Let's focus on minimizing f with respect to H, holding W fixed. The gradient with respect to H_{kj} is:

$$\frac{\partial f}{\partial H_{kj}} = \sum_i W_{ik} \left((WH)_{ij} - V_{ij} \right)$$

To satisfy the KKT condition, we want H_{kj} to decrease where the gradient is positive and increase where the gradient is negative, without violating the non-negativity constraint $H_{kj} \geq 0$.

We can multiply H_{kj} by a factor that will drive the gradient toward zero. This leads to the following update rule:

$$H_{kj} \leftarrow H_{kj} \frac{\sum_i W_{ik} V_{ij}}{\sum_i W_{ik} (WH)_{ij}}$$

In matrix notation, this becomes:

$$H_{kj} \leftarrow H_{kj} \frac{(W^T V)_{kj}}{(W^T WH)_{kj}}$$

Next, we minimize f with respect to W, holding H fixed. The gradient with respect to W_{ik} is:

$$\frac{\partial f}{\partial W_{ik}} = \sum_j H_{kj} \left((WH)_{ij} - V_{ij} \right)$$

As before, to satisfy the KKT conditions, we update W_{ik} multiplicatively. This gives the following update rule:

$$W_{ik} \leftarrow W_{ik} \frac{\sum_j H_{kj} V_{ij}}{\sum_j H_{kj} (WH)_{ij}}$$

In matrix notation, this is:

$$W_{ik} \leftarrow W_{ik} \frac{(VH^T)_{ik}}{(WHH^T)_{ik}}$$

The multiplicative update rules are designed to ensure that the matrices W and H remain non-negative at every iteration. Since both update rules involve element-wise multiplication by positive factors, if W and H are initialized with non-negative values, they will remain non-negative throughout the optimization process.

To prove the correctness of the multiplicative update rules, we must show that they lead to a monotonic decrease in the objective function $f(W, H)$. This is guaranteed by the nature of the update rules, which are derived from the KKT conditions. At each step, the objective function either decreases or remains the same, ensuring convergence to a local minimum.

1.5. Dirichlet process and stick-breaking formulation

The Dirichlet Process (DP) is a powerful and flexible concept in probability theory and statistics. It's particularly useful in the realm of Bayesian nonparametric methods, where traditional statistical ones often assume a fixed and predetermined number of parameters.

In practical terms, the DP is often used as a prior distribution in Bayesian statistics. It's especially handy when dealing with problems involving an unknown number of clusters or components. It provides a way to model uncertainty about the number of groups in the data, allowing for a more adaptive and data-driven approach.

A DP is a stochastic process that assigns probabilities to sets of outcomes. It's often denoted as $DP(\alpha, P_0)$, where

- α is a positive prior concentration parameter controlling the degree of shrinkage of P toward P_0.
- P_0 is the base measure, providing an initial guess at P.

For any partition $(A_1, \dots, A_k)$ of the sample space Ω, if P is a random probability measure drawn from $DP(\alpha P_0)$, then the joint distribution of $(P(A_1), \dots, P(A_k))$ follows a Dirichlet distribution

$$P(A_1), \dots, P(A_k) \sim \text{Dirichlet}(\alpha P_0(A_1), \dots, \alpha P_0(A_k)) \tag{1}$$

The definition of the Dirichlet process and properties of the Dirichlet distribution imply,

$$P(A) \sim \text{Beta}(\alpha P_0(A), \alpha(1 - P_0(A)), \tag{2}$$

so that the marginal random probability assigned to any subset A of the support is simply beta distributed.

The stick-breaking representation allows us to induce $P \sim \text{DP}(\alpha P_0)$ by letting

$$\begin{aligned} P(A) &= \sum_{h=1}^{\infty} \pi_h \delta_{\theta_h}(A), \\ \pi_h &= \beta_h \prod_{l<h} (1 - \beta_l), \\ \beta_h &\sim \text{Beta}(1, \alpha), \\ \theta_h &\sim P_0, \end{aligned} \tag{3}$$

for any measurable space A, where δ_θ denotes a degenerate distribution with all its mass at θ, the locations $(\theta_h)_{h=1}^{\infty}$ are generated independently from the base distribution P_0, π_h is the probability mass at location θ_h, and these probability masses are generated from a so-called stick-breaking process that guarantees that the weights sum to 1.

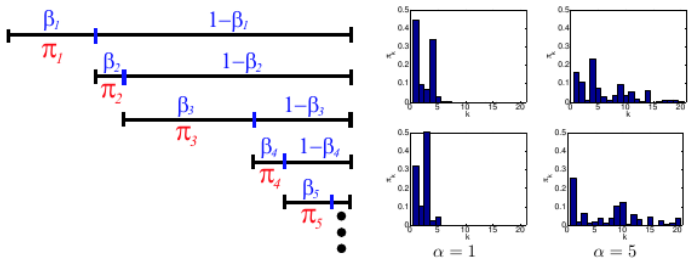


Figure 6. Intuition behind the stick breaking formulation. Wiecki, Thomas. (2013). Sequential sampling models in computational psychiatry: Bayesian parameter estimation, model selection and classification.

To describe the stick-breaking process, we start with a stick of unit length representing the total probability to be allocated to all the atoms. We initially break off a random piece of length β_1, with the length generated from a $\text{Beta}(1, \alpha)$ distribution, and allocate this $\pi_1 = \beta_1$ probability weight to the randomly generated first atom $\theta_1 \sim P_0$ (Figure 6).

There is then $1 - \beta_1$ of the stick remaining to be allocated to the other locations. We break off a proportion $\beta_2 \sim \text{Beta}(1, \alpha)$ of the $1 - \beta_1$ stick and allocate the probability $\pi_2 = \beta_2(1 - \beta_1)$ to the second atom $\theta_2 \sim P_0$.

As we proceed, the stick gets shorter and shorter so that the lengths allocated to the higher indexed atoms decrease stochastically, with the rate of decrease depending on the DP precision parameter α (Figure 7).

1.6. Non-parametric clustering using multiple types of mutational signatures

Patterns of similar exposures across patients can unravel distinct molecular subtypes, suggesting that patients could be clustered by exposures. The main difference from signature detection, however, is that subtypes should be detected by considering distinct types of signatures (i.e., SBS, DBS, etc.). This section reports a Bayesian non-parametric model to cluster patients into groups with similar exposure patterns (Figure 8). The model uses a tensor-based clustering algorithm that leverages a Dirichlet process (Blei and Jordan, 2006) to combine V signature types and automatically determine the number of clusters G.

Mixture models are a class of probabilistic models that are employed in statistics and machine learning to represent the underlying probability distribution of a random variable as a combination of multiple probability distributions. Each component distribution in the mixture model is associated with a certain weight, indicating the likelihood of selecting that particular component.

These models are particularly useful when dealing with populations that exhibit heterogeneity, meaning they can be sub-divided into distinct subgroups or clusters.

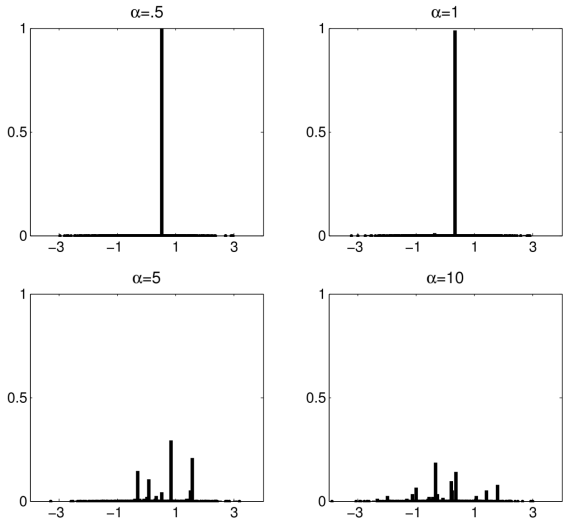


Figure 7. Samples from the stick-breaking representation of the Dirichlet process with different settings of the precision parameter α. Image taken from Gelman, A. et. al. (2013). "Bayesian Data Analysis (3rd ed.)."

In parametric mixture models the number of clusters G is given as input of the model, and the optimal number of G is chosen through model selection, i.e., optimising quantities such as Bayesian Information Criterion (BIC), Akaike Information Criterion (AIC) or Integrated Completed Likelihood (ICL). These values are computed by adding penalties provided by the sample size, model complexity, clusters entropy to the likelihood.

We can use a DP as the prior for the mixing proportions to implement a non-parametric mixture model and thus avoid the model selection phase. The stick-breaking formulation generates an infinite number of breaks. In practice, a maximum value for G is provided and only the clusters with non-zero weight will be retained.

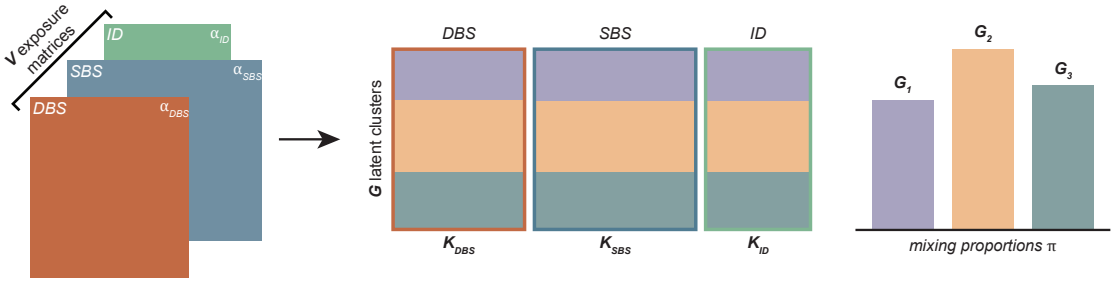


Figure 8. Non-parametric tensor clustering graphical representation. The input of the method is a tensor composed of V exposure matrices of distinct signature types (SBS, DBS, ID), each with distinct number of signatures (K_{SBS}, K_{DBS} and K_{ID}, respectively). A cohort of N patients is then clustered in G latent groups based on the distinct input signals.

1.6.1. Generative Model

Assume to have V exposure matrices $\alpha^{(1)}, \cdots, \alpha^{(V)}$ of distinct signature types. Each matrix has N rows (i.e., samples) and $K^{(1)}, \cdots, K^{(V)}$ columns (i.e., signatures). To cluster the N samples based on multiple signature types, we embed the exposure matrices in a tensor $\mathbf{A}$ of dimension $V \times N \times K'$. We define $K' = \max_{v=1,\cdots,V} K^{(v)}$ as the maximum number of signatures among the distinct signature types and perform zero-padding on the other exposure matrices, to standardise the dimensions.

The likelihood of the model is

$$p(\boldsymbol{A}|\boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\theta}) = \prod_{v=1}^{V} \prod_{n=1}^{N} \sum_{g=1}^{G} \pi_g \mathrm{p}(\boldsymbol{A}_{\boldsymbol{v},\boldsymbol{n}}|\boldsymbol{\theta}_{\boldsymbol{v},\boldsymbol{g}}). \tag{4}$$

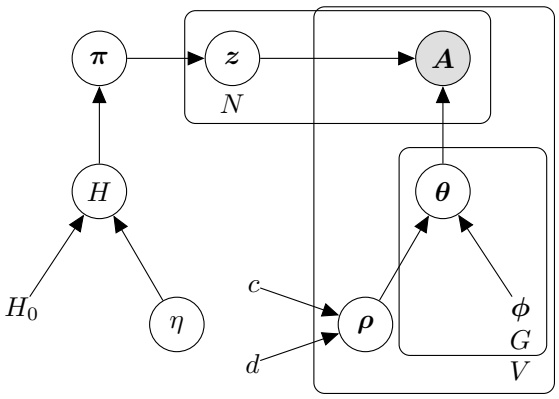
Each mixture component is distributed as a Dirichlet with concentration parameter $\theta_{v,g}$

$$\mathrm{p}(\boldsymbol{A}_{\boldsymbol{v},\boldsymbol{n}}|z_n = g, \boldsymbol{\theta}_{\boldsymbol{v},\boldsymbol{g}}) = \mathrm{Dirichlet}(\boldsymbol{\theta}_{\boldsymbol{v},\boldsymbol{g}}).$$

In the model, π is a G-dimensional vector of mixing proportions, distributed as a Dirichlet Process and such that $\sum_{g=1}^{G} \pi_g = 1$. The model formulates the marginal distribution over the latent clustering assignments $p(Z)$ in terms of π as $p(z_n = g) = \pi_g$, where π_g represents the probability for a data point to be assigned to cluster g.

Figure 9 reports the PGM and generative model for the stick-breaking Dirichlet mixture used to cluster patients.

- The tensor of exposure matrices $\mathbf{A}$, inferred via Bayesian NMF, is the observed variable of the model. The distribution of each exposure



(a) Probabilistic graphical model of the stick-breaking mixture model.

$$
\begin{aligned}
H &\sim DP(\eta, H_0) \\
\boldsymbol{\pi} &\sim H \\
z_n &\sim \text{Categorical}(\boldsymbol{\pi}) \\
\boldsymbol{\rho}_v &\sim \text{Gamma}(c, d) \\
\boldsymbol{\theta}_{v,g} &\sim \text{Dirichlet}(\phi_{v,g} \cdot \rho_v) \\
\boldsymbol{A}_{v,n} &\sim \text{Dirichlet}(\boldsymbol{\theta}_{v,z_n})
\end{aligned}
$$

(b) Prior distributions of the stick-breaking mixture model.

Figure 9. Bayesian nonparametric mixture model.

matrix is assumed to be a Dirichlet, since each patient's exposure is a probability vector summing to 1. The concentration parameter of the Dirichlet is $\theta_{v,g}$.

- θ is a $V \times G \times K'$ matrix. Each value $\theta_{v,g}$ represents the K'-dimensional centroid exposures for samples assigned to group g. Similarly to $\boldsymbol{A}_{v,n}$, the centroid is distributed as a Dirichlet with concentration $\phi_{v,g} \cdot \rho_v$.
- $\boldsymbol{\rho}$ is a V-dimensional scale factor introduced to reduce the variance of the Dirichlet distribution and is sampled from a Gamma prior.
- Z corresponds to a N dimensional vector of latent clustering assignments. For each patient, a value for z_n is sampled from a Categorical with probability π.
- π is the vector of mixing proportions, where each component corresponds to the weight of each cluster. It is sampled from a DP H with parameters ν and H_0.

After the inference of the latent parameters, each patient is assigned to the cluster with highest posterior probability $p(Z|\boldsymbol{A}, \boldsymbol{\gamma})$, with $\boldsymbol{\gamma}$ defined as the set of infered parameters.

1.6.2. Implementation and inference in Pyro

Stochastic Variational Inference (SVI) In Bayesian inference we consider our latent parameters distributed according to their priors. Considering a latent variable Z and a set of hyperparameters ψ, the posterior distribution of Z given the observed data X and ψ, can be computed according to the Bayes Theorem as

$$p(Z|X, \psi) = \frac{p(X|Z, \psi)P(Z|\psi)}{p(X|\psi)}. \tag{5}$$

The aim of the inference is thus to find the set of latent parameters governing Z such as to maximise the posterior, i.e., the maximum a posteriori (MAP) estimates. In the present model, the latent variable corresponds to the assignments vector Z, and the MAP inference can be formulated as

$$\hat{Z_{MAP}}(X) = \underset{Z}{\text{argmax}}\, p(X|Z, \psi)p(Z|\psi) \tag{6}$$

The denominator of the Bayes Theorem, also called evidence, computes the marginal density of the observations and is usually analytically intractable, thus approximation methods are often used for the inference. A common technique to approximate the posterior is variational inference (VI) (Blei, Kucukelbir, and McAuliffe 2017). The goal of VI, given a family of candidate distributions Q, is to find the distribution $q*(Z) \in Q$, named the variational distribution, better approximating the true posterior. The task can be formulated as an optimization problem

$$q*(Z) = \underset{q(Z) \in Q}{\text{argmin}} KL[q(Z)||p(Z|X)] \tag{7}$$

where $KL[q(Z)||p(Z|X)]$ is the Kullback-Liebler divergence between the variational and posterior distributions. The KL divergence is an entropy-based measure computing the dissimilarity of the two distributions and can be formulated as

$$
\begin{aligned}
KL[q(Z)||p(Z|X)] &= E[\log q(Z)] - E[\log p(Z|X)] = \\
&= E[\log q(Z)] - E[\log p(Z, X)] + \log p(X)
\end{aligned} \tag{8}
$$

with explicit dependance on the evidence $p(X)$. The KL is minimised when the two distributions coincide.

Since the KL cannot be directly computed, we consider an additional objective function called evidence lower bound (ELBO). The ELBO represents a lower bound of the evidence and can be formulated as equal to the KL divergence up to a constant

$$
\begin{aligned}
ELBO(q) &= E[\log p(Z, X)] - E[\log q(Z)] = \\
&= E[\log p(X|Z)] + E[\log p(Z)] - E[\log q(Z)] = \\
&= E[\log p(X|Z)] - KL[q(Z)||p(Z)].
\end{aligned} \tag{9}
$$

Minimising the KL divergence equals maximising the ELBO, or minimising the negative ELBO (Blei, Kucukelbir, and McAuliffe 2017).

A technique to solve this optimization problem is given by gradient-based optimization as in stochastic variational inference (SVI). In SVI, given the set of variational parameters , the ELBO maximisation task is carried out by taking its gradient with respect to the variational parameters

$$\nabla_{\phi} ELBO = \nabla_{\phi} E_{q(Z)}[\log p(X,Z) - \log q_{\phi}(Z)]. \tag{10}$$

Model implementation in Pyro In the following block is reported a possible model implementation in Pyro.

```

  import torch
  import pyro
  import pyro.distributions as dist
  import pyro.distributions.constraints as constraints
  import torch.nn.functional as F
  from pyro.ops.indexing import Vindex
  from pyro.infer import SVI, Trace_ELBO

  hyperparameters = {"pi_conc0":0.6,  # DP concentration
    "z_tau":0.1,  # relaxed one-hot temperature
    "scale_factor_centroid":1,  # rho
    "alpha_thr":0.01}

  def _mix_weights(beta):
    '''
    Function used for the stick-breaking process.
    '''
    beta1m_cumprod = (1 - beta).cumprod(-1)
    return F.pad(beta, (0, 1), value=1) * F.pad(beta1m_cumprod, (1, 0), value=1)

  def model_mixture(cluster, alpha):
    n_samples = alpha.shape[0]
    n_variants = alpha.shape[1]
    n_sigs = alpha.shape[2]

    pi_conc0 = hyperparameters["pi_conc0"]
    z_tau = torch.tensor(hyperparameters["z_tau"]).double()

    # stick-breaking process
    with pyro.plate("beta_plate", cluster-1):
      pi_beta = pyro.sample("beta_pi", dist.Beta(1, pi_conc0))
    pi = _mix_weights(pi_beta)

  # alpha_prior = V x G x K
  sf_conc = hyperparameters["scale_factor_centroid"]
  alpha_centr = torch.ones((n_variants, cluster, n_sigs))
  with pyro.plate("g1", cluster, dim=-1):
    with pyro.plate("n_vars1", n_variants, dim=-2):
      sf_centroid = pyro.sample("scale_factor_centroid", dist.Gamma(sf_conc*2, 2))
      alpha_prior = pyro.sample("alpha_prior", dist.Dirichlet(alpha_centr * sf_centroid.unsqueeze(-1)
    ))

  tau = pyro.sample("z_tau", dist.Normal(z_tau, 10.))
  # alpha = N x V x K
  with pyro.plate("n", n_samples):
    z_onehot = pyro.sample("latent_class", dist.RelaxedOneHotCategorical(temperature=tau, logits=
    torch.log(pi)))
    z = z_onehot.argmax(dim=-1).long()
    for v in pyro.plate("n_vars2", n_variants):
      # alpha -> V x N x K
      # alpha_prior -> V x G x K
      alpha_c = Vindex(alpha_prior)[v,z,:]
      pyro.sample(f"obs_{v}", dist.Dirichlet(alpha_c), obs=alpha[:,v,:])
```