Latent Variable Models for Genomic Data

Summarizing the EVE method of Frazer et al. 2021

Nick Landolfi and Dan O'Neill
Stanford University

Revision: 24 Mar 2022

Agenda

- Background, goal, overview
- Latent variable models
- Method used in the paper
- Results
- Next steps

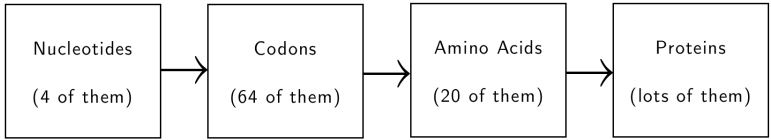
Background

- about me: fourth year Ph.D. candidate in Computer Science at Stanford (computational side)
- about this talk: *unsupervised prediction of protein variant pathogenicity*
  - *Disease variant prediction with deep generative models of evolutionary data* [Frazer et al. 2021]
    - a nature paper from last year
  - involved/sophisticated research effort
    - computational biology team (5 people at Harvard) and machine learning team (3 people at Oxford)
    - most recent contribution in a decade-long research program
    - they build probabilistic models for 3,000+ proteins, each protein takes 80 hr. CPU time

Pathogenicity via probability

- goal: *quantify pathogenicity* of protein variants in disease-related genes
- problem: *infeasible* to label all variants (even with high-throughput experiments); see paper
  - 6.5 million missense variants in the gnomAD dataset of 141,000 human genomes
  - 36 million missense variants associated with 3,219 disease-related genes in ClinVar
- approach: (roughly speaking) protein variants which appear in nature have been *selected for fitness*
  - given a dataset of naturally occurring variants, one could build an *unsupervised* probabilistic model

Proteins as strings



- recall that a protein is a molecule which we can represent as an *amino acid string*
- nucleotides are read into aminos in groups of three called *codons*
  - call the amino *string* corresponding to nucleotide *string* the *sense* of the nucleotide string
  - different nucleotide strings can have same or different sense (depends on codons)
- paper's focus: when a sense is different by one amino in one spot, called a *missense variant*
  - in other words, the paper restricts interest to mutations that swap a single amino acid
  - they look at missense variants of protein-coding genes which are associated with disease

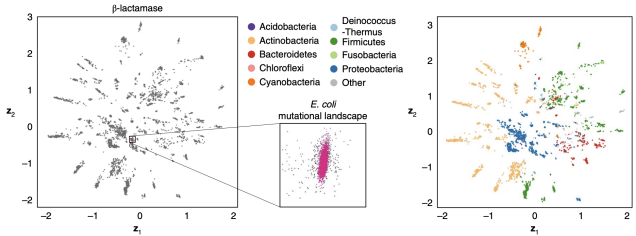
Distribution on amino strings

- paper asserts that *uncommon variants are pathogenic*
- so the goal is to build a protein-specific distribution over naturally occurring amino strings
  - given a length-k *wild-type* (or canonical) protein $x_{\text{wt}} \in \mathcal{X} = \{\text{A}, \text{R}, \cdots, \text{V}\}^k$ for a gene
  - want the distribution $p : \mathcal{X} \to [0, 1]$ of *naturally occurring variants* of this protein
  - use p to *score variants*: $p(x) > p(y)$ means variant x is *more common* than y
- if we had p, we could define the *evolutionary index* E_{v} of variant x_{v} by

$$E_{\text{v}} = -\log \frac{p(x_{\text{v}})}{p(x_{\text{wt}})}$$

  - if a variant has relatively low probability, then it is a candidate for being pathogenic
  - obtaining this index is the point of the paper; hence *evolutionary model of variant effect* (EVE)

Latent structure in genetic data

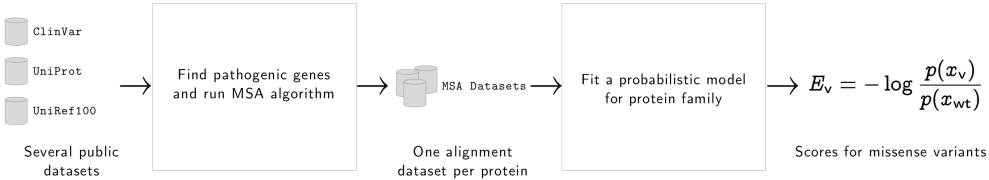


- amino acid space still too big, at least 20^{100} variants...can't write down p
- perhaps (nonlinear) *latent structure*, can use it to *approximate* p
  - conservation across certain subindices of protein
- figure from Riesselman 2018 (prior work by some of the EVE authors)
  - trained a VAE latent variable model (will discuss later) with 2-dimensional latent space

Latent variable models

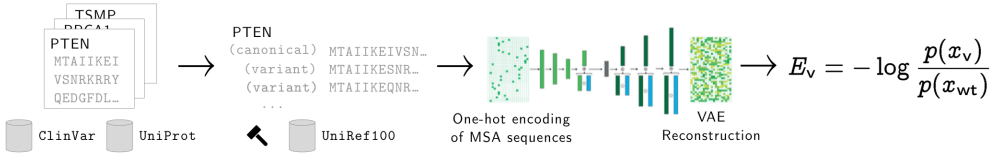
- observe $x \in \mathcal{X}$, postulate $p_x(x) = \int_{\mathcal{Z}} p_{zx}(\cdot, x)$ and $p_{zx} = p_z p_{x|z}$
  - $z \in \mathcal{Z}$ are hidden and *not observed*
- roughly speaking, most of the structure in x comes from structure in z
- ubiquitous example: any signal + noise model
  - other examples include *gaussian mixture models*, *hidden markov models* (HMMs) etc.
    - e.g., `jackhmmr` multiple sequence alignment (MSA) tool used in paper is based on an HMM
  - variational autoencoders (VAEs) are one such latent variable model, which we will discuss later
    - their conditional distribution $p_{x|z}$ is parameterized using a neural network

Block diagram of method used in paper



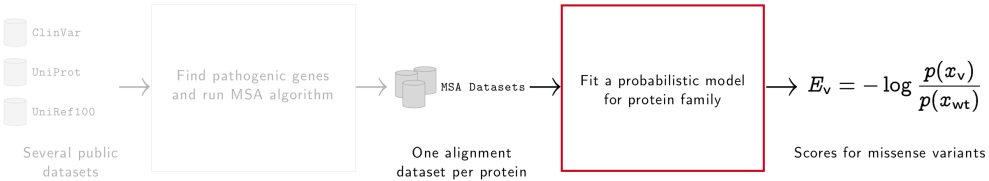
- first, they find pathogenic genes and construct a MSA dataset for each one
- second, they fit a probabilistic latent variable for each protein and score variants

Method used in paper

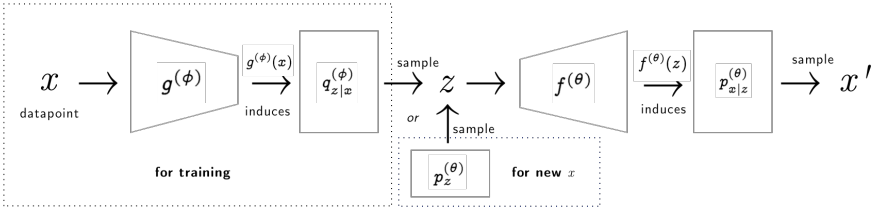


- dataset construction
  1. associate a gene with a canonical wild-type protein; using `ClinVar`, `UniProt`
  2. associate and align many similar proteins found in nature with that canonical one
     - specifically, get a multiple sequence alignment (MSA) using `jackhmmr` against `UniRef100`
- probabilistic model and scoring
  3. fit a VAE to a dataset (a subset of subsequences)
  4. likelihood score all proteins which are one-amino substitutions of the canonical protein

Probabilistic model piece



Variational autoencoder (VAE)



- a *variational autoencoder* from latents Z to observations X is a pair $(p^{(\theta)}_z, p^{(\theta)}_{x|z}), q^{(\phi)}_{z|x}$ where
  - $(p^{(\theta)}_z, p^{(\theta)}_{x|z})$ is a deep latent-variable model with parameters θ, called *generative model*
    - $p^{(\theta)}_z : Z \to \mathbf{R}$ is a distribution with parameters from θ, called *latent prior distribution*
    - $p^{(\theta)}_{x|z} : X \times Z \to \mathbf{R}$ is a deep conditional with params from θ, called *decoder distribution*
      - has associated *decoder neural network* $f^{(\theta)}$ with domain Z
  - $q^{(\phi)}_{z|x} : Z \times X \to \mathbf{R}$ is deep conditional with params ϕ, called *encoder distribution*
    - has associated *encoder neural network* $g^{(\phi)}$ with domain X

Evidence lower bound for log likelihood

- *log likelihood* of i.i.d. observed dataset $x^1, \ldots, x^n$ in X under VAE model is $\sum_{i=1}^n \log p_x^{(\theta)}(x^i)$
  - where the *model evidence* $p_x^{(\theta)}(x^i) = \int_Z p_{zx}^{(\theta)}(\zeta, x)$ is assumed (usually is) intractable
- but since $\int_Z q_{z|x}^{(\phi)}(\cdot, x^i) = 1$ and $p_x^{(\theta)}(x^i) = p_{zx}^{(\theta)}(\zeta, x^i)/p_{z|x}^{(\theta)}(\zeta, x^i)$ for all $\zeta \in Z$, can express

$$\log p_x^{(\theta)}(x^i) = \underbrace{\int_Z q_{z|x}^{(\phi)}(\zeta, x) \log \frac{p_{zx}^{(\theta)}(\zeta, x^i)}{q_{z|x}^{(\phi)}(\zeta, x^i)} d\zeta}_{\text{ELBO}(\theta,\phi,x^i)} + \underbrace{\int_Z q_{z|x}^{(\phi)}(\zeta, x^i) \log \frac{q_{z|x}^{(\phi)}(\zeta, x^i)}{p_{z|x}^{(\theta)}(\zeta, x^i)} d\zeta}_{d_{kl}(q_{z|x}^{(\phi)}(\cdot,x^i), p_{z|x}^{(\theta)}(\cdot,x^i)) \geq 0}$$

$$\geq \text{ELBO}(\theta, \phi, x^i)$$

  - d_{kl} is the Kullback-Leibler divergence between two distributions (densities)
    - it is a *nonnegative* similarity measure (but not a metric); $d_{kl}(q, p) = 0$ when $q = p$
    - ϕ are sometimes called *variational* parameters, since one wants $q_{z|x}^{(\phi)} \approx p_{z|x}^{(\theta)}$
- can maximize the *evidence lower bound* (ELBO) as a proxy for the likelihood

ELBO as reconstruction loss and regularization

- recall from the previous slide that $\log p_x^{(\theta)}(x^i) \geq \text{ELBO}(\theta, \phi, x^i)$
- again, since $p_{zx}^{(\theta)}(\zeta, x^i) = p_z^{(\theta)}(\zeta) p_{x|z}^{(\theta)}(x^i, \zeta)$ for all $\zeta \in Z$, express

$$\begin{aligned}\text{ELBO}(\theta, \phi, x^i) &= \underbrace{\int_Z q_{z|x}^{(\phi)}(\zeta, x^i) \log p_{x|z}^{(\theta)}(x^i, \zeta) d\zeta}_{\ell(\theta, \phi, x^i)} + \underbrace{d_{kl}(q_{z|x}^{(\phi)}(\cdot, x^i), p_z^{(\theta)})}_{r(\theta, \phi, x^i)} \\ &= \ell(\theta, \phi, x^i) + r(\theta, \phi, x^i)\end{aligned}$$

  - can interpret ℓ a *reconstruction loss* and r as a *regularization*
    - ℓ is an integral (expectation) and may be estimated via monte carlo
    - r is often analytical since it is a divergence of two distributions
  - if these are differentiable in parameters, can apply usual stochastic gradient methods (next slide)

Gradient of ELBO

- recall $\text{ELBO}(\theta, \phi, x^i) = \ell(\theta, \phi, x^i) + r(\theta, \phi, x^i)$
  - for first-order (gradient) methods, one wants $\nabla_{(\theta,\phi)}\text{ELBO}$
- loss $\ell(\theta, \phi, x^i)$ is an integral (expectation) of $\log p^{(\theta)}_{x|z}$, use *monte carlo* to approximate

$$\int_Z q^{(\phi)}_{z|x}(\zeta, x^i) \log p^{(\theta)}_{x|z}(x^i, \zeta) d\zeta \approx \sum_{j=1}^{m} \log p^{(\theta)}_{x|z}(x^i, \zeta_i^{(j)})$$

  with m samples $\zeta_i^{(j)} \sim q^{(\phi)}_{z|x}(\cdot, x^i)$ from the encoder model

  - empirical fact $m = 1$ works; so approximate $\ell(\theta, \phi, x^i) \approx \log p^{(\theta)}_{x|z}(x^i, \zeta_i)$ where $\zeta_i \sim q^{(\phi)}_{z|x}(\cdot, x^i)$
  - difficulty: the sampling distribution depends on ϕ; fix: reparameterize ζ_i (called *reparameterization trick*)
    - e.g., suppose ζ_i is gaussian with mean $\mu_i^{(\phi)}$ and covariance $\Sigma_i^{(\phi)}$ (ζ_i params depends on ϕ)
    - $\mu_i^{(\phi)} + (\Sigma_i^{(\phi)})^{1/2}\varepsilon_i$ for ε_i mean-zero identity-covariance gaussian (ε_i params don't depend on ϕ)
- the regularization $r(\theta, \phi, x^i)$, a divergence, is assumed (often) analytically computable
  - e.g., for gaussian latent and gaussian encoder, exists closed form for divergence between two gaussians

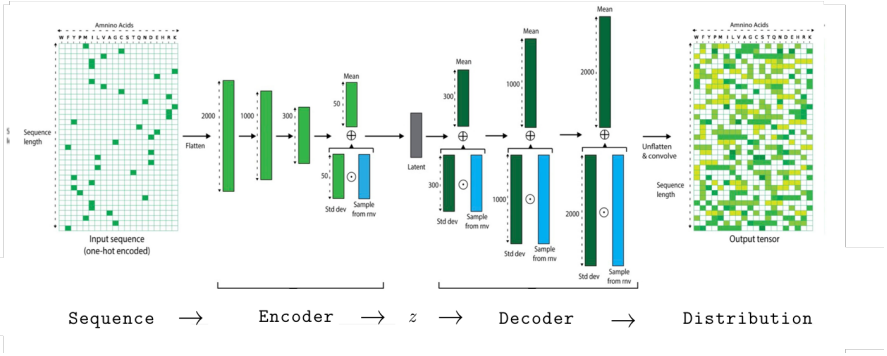
Finding parameters for VAE

- in summary, we have bounded below the log likelihood of the dataset

$$\sum_{i=1}^{n} \log p_x^{(\theta)}(x^i) \geq \sum_{i=1}^{n} \text{ELBO}(\theta, \phi, x^i) = \sum_{i=1}^{n} \ell(\theta, \phi, x^i) + r(\theta, \phi, x^i)$$

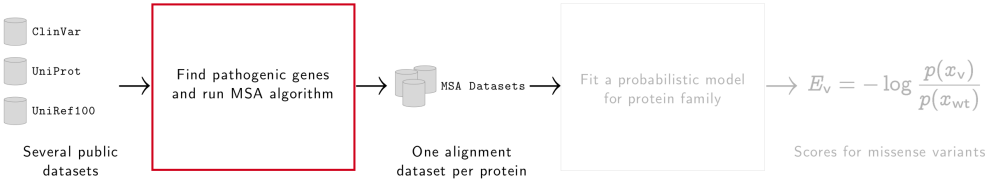
- use *minibatch stochastic gradient ascent* to maximize right hand side
  - i.e., sample k points from dataset, compute gradients using techniques on previous slide
  - algorithm is called *auto-encoding variational bayes* [Kingma & Welling 2014]
  - the gradient estimator is called *stochastic gradient variational bayes estimator* [Rezende et al. 2014]

Details of paper's VAE

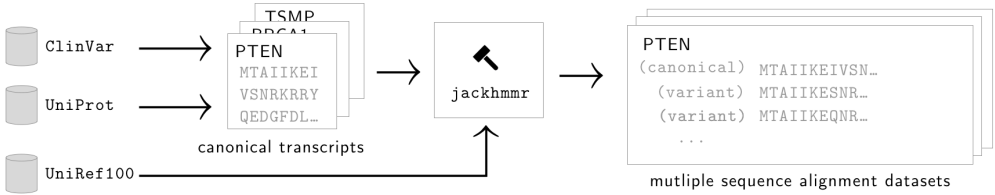


- details of neural network architecture in paper, lots of exploration to land on this model
- slight wrinkle: paper uses a Bayesian VAE (i.e. learns a distribution on decoder weights θ)
  - same ELBO machinery we discussed works, gives one additional term in loss
  - paper claims that using a Bayesian neural network further improved results

Dataset construction piece

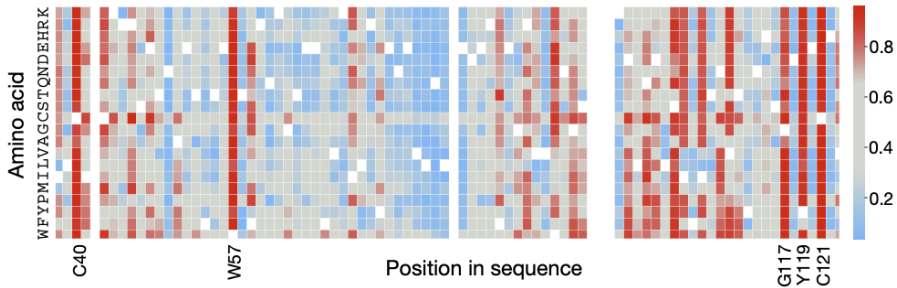


Multiple sequence alignment datasets



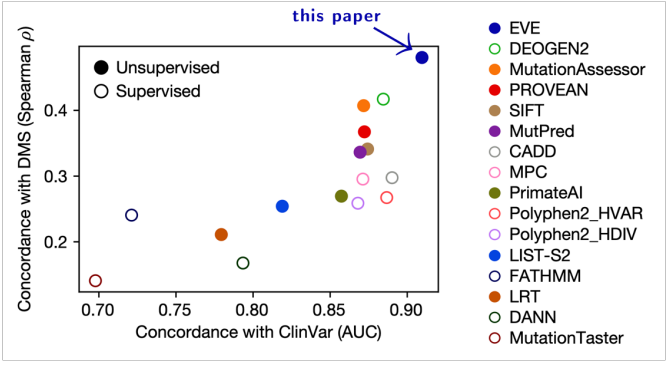
- goal: turn public sources into a dataset of aligned protein families
  - `ClinVar`: variant labels; `UniProt`: canonical protein transcript; `UniRef100`: naturally occurring proteins
- use `ClinVar` database to identify genes associated with disease; for each such gene:
  1. use `UniProt` database to find the *canonical protein* associated with the gene
  2. use `jackhmmer` against `UniRef100` to find and align *homologous proteins* to that protein
     - heuristic 1: pick subset of those sequences that "well-match" the canonical one
     - heuristic 2: pick subset of focus indices "well-conserved" across this subset of sequences
- methodology originally proposed in Hopf et al. 2017 (same lab at Harvard)

Results: example plot for SCN1B



- heat map of EVE pathogenicity scores in SCN1B, hotter (red) is more pathogenic
- paper's bottom line: get one of these for each gene (canonical protein) of interest
  - suggest using these scores to filter pathogenic candidates for further investigation

Comparison to other methods



Next steps

- what is the meaning of the learned latent variables?
- how sensitive is the model to the dataset generation choices?
- does one need to use VAEs? there are other generative models, are there simpler choices?
- beyond missense variants?

Conclusion

- *latent variable models* are viable for genomic data
- sophisticated approaches obtain state of the art results
- several directions for future work on simplifications, extensions, interpretations