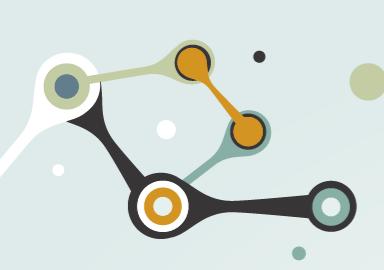


HMM FOR GENOME DECODING

State and base inference in
genomic sequences via Viterbi
and Posterior decoding



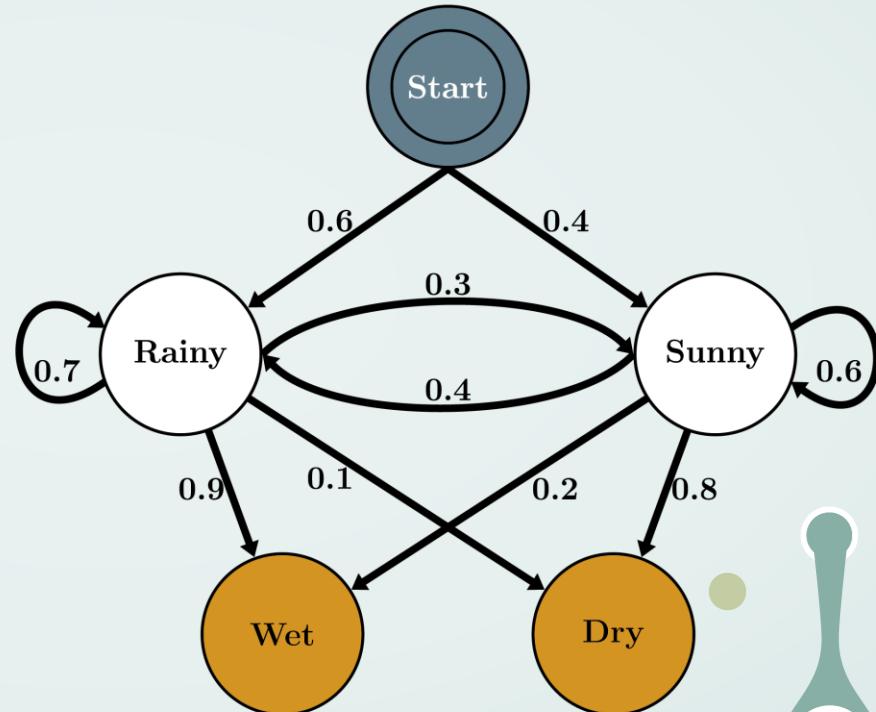
HIDDEN MARKOW MODEL

What is an HMM?

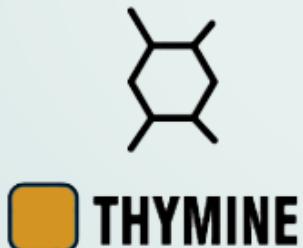
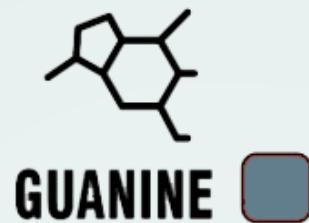
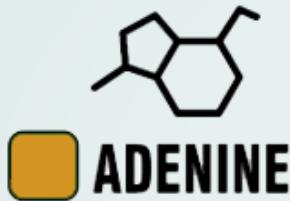
A Hidden Markov Model (HMM) is a statistical framework in which the system evolves through unobserved (hidden) states according to a Markov process, while emitting observable outputs that depend probabilistically on the current hidden state.

Key components:

- Hidden States (S)
- Observations (O)
- Transition Probabilities (A)
- Emission Probabilities (B)
- Initial Probabilities (π)



HIDDEN STATES AND OBSERVATIONS



Hidden States S

$$S = \{GC_Rich, Gc_Poor\}$$

Represent different genomic regions with distinct GC content

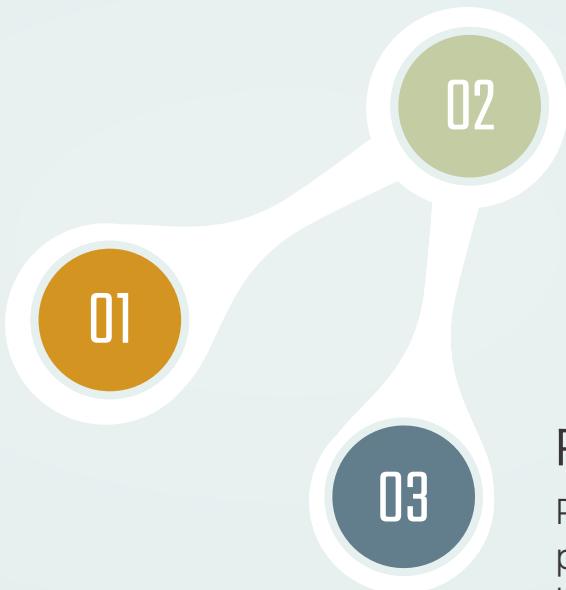
Observations O_t

$$O_t \in \{A, C, G, T\}$$

DNA bases observed at each position t

HMM PARAMETERS

TRANSITION MATRIX A
Probability of switching
between hidden states

$$A_{ij} = P(s_t = j | s_{t-1} = i)$$


EMISSION MATRIX B
Probability of emitting a base given
the previous base and the current
hidden state (order 1)

$$B_j(o_{t-1}, o_t) = P(o_t | o_{t-1}, s_t = j)$$

PROBABILITIES π
Probability that the
process starts in that
hidden state

$$\pi_i = P(s_1 = i)$$

WORKFLOW & OBJECTIVES

TRAINING

- Use Baum-Welch algorithm to estimate HMM parameters (A, B, π)
- Train on **synthetic genomes** (1Mb) and **real genomes**, subdividing real genomes into **coding** and **non-coding** regions

INFERENCE

- Apply Viterbi and Posterior decoding to impute bases and predict hidden states
- State reconstruction performed only on synthetic genomes (where true states are known)

EVALUATION

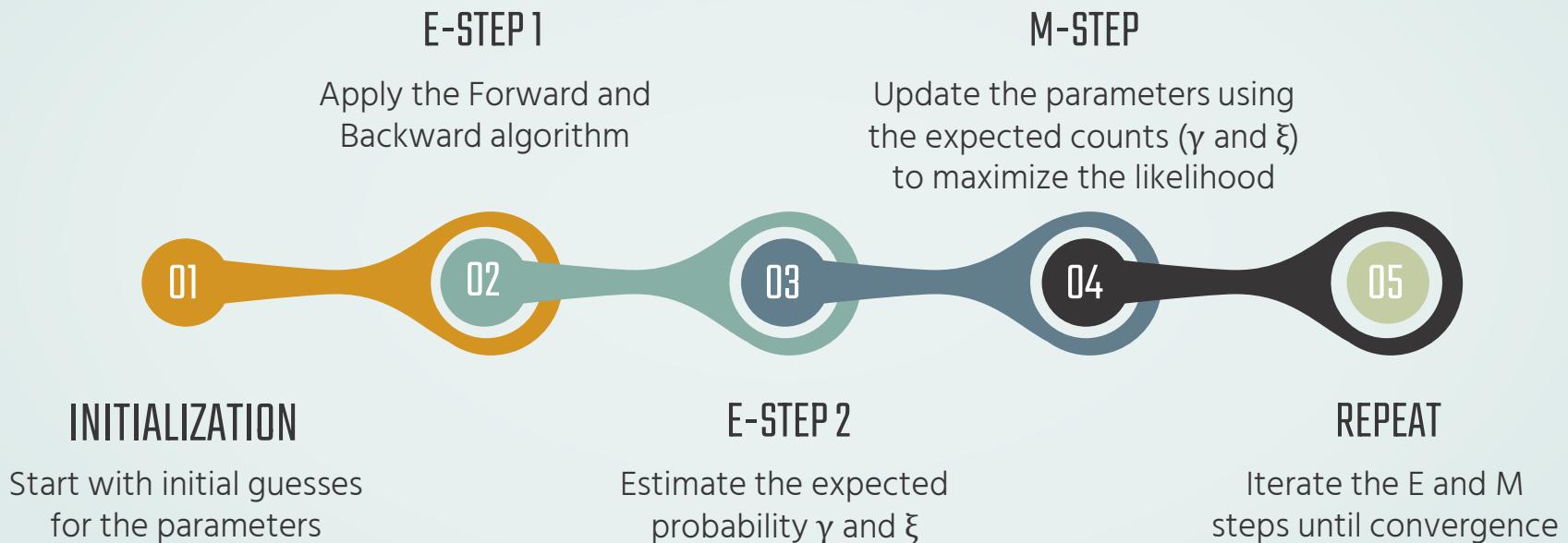
- Measure **accuracy** of base imputation on both synthetic and real data
- Compare results between coding vs non-coding regions in real genomes



BAUM-WELCH ALGORITHM

An Expectation-Maximization (EM) algorithm used to learn the parameters of an HMM from observed data

BAUM-WELCH ALGORITHM



INITIALIZATION

Before training with Baum-Welch, initial parameters are chosen with **biased randomness** to reflect biological intuition

Emission Tensor B

Boosts A/T or G/C random emission probabilities with a random scaling factors

$$\mathbf{u} \in (1, 10)$$

Transition Matrix A

High probability to remain in the same state

$$A = \begin{pmatrix} p & 1-p \\ 1-p & p \end{pmatrix}$$

$$p = u \in (0.65, 0.95)$$

E-STEP 1

FORWARD AND BACKWARD ALGORITHM

FORWARD

Tracks the probability of being in state i at time t , having seen the partial sequence (o_1, o_2, \dots, o_t)

$$\alpha(t) = P(o_1, \dots, o_t, s_t = i | \theta)$$

Initialization

$$\alpha_1(i) = \pi_i \cdot B[i, o_0, o_1]$$

Recursion

$$\alpha_t(i) = \sum_j \alpha_{t-1}(j) \cdot A[i, j] \cdot B[j, o_{t-1}, o_t]$$

BACKWARD

Tracks the probability of observing the rest of the sequence (o_{t+1}, \dots, o_T) given that we're in state i at time t

$$\beta(t) = P(o_{t+1}, \dots, o_T, s_t = i | \theta)$$

Initialization

$$\beta_t(i) = 1$$

Recursion

$$\beta_t(i) = \sum_j \beta_{t+1}(j) \cdot A[i, j] \cdot B[j, o_t, o_{t+1}]$$

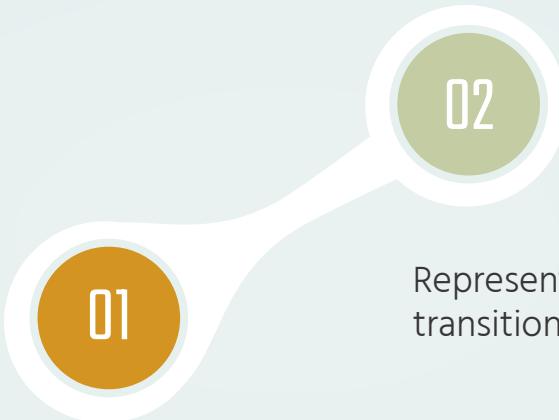
E-STEP 2

EXPECTATION OF HIDDEN VARIABLES

GAMMA (γ)

Represents the probability of being in state i at time t , given the full observation sequence

$$\gamma_t(i) = \frac{\alpha_t(i) \cdot \beta_t(i)}{\sum_k \alpha_t(k) \cdot \beta_t(k)}$$



$\xi_t(\xi)$

Represents the expected probability of transitioning from state i to state j at time t

$$\xi_t(i) = \frac{\alpha_t(i) \cdot \beta_{t+1}(j) \cdot A_{ij} \cdot B_j(o_{t+1})}{\sum_{ab} \alpha_t(a) \cdot \beta_{t+1}(b) \cdot A_{ab} \cdot B_b(o_{t+1})}$$

$$\xi_t(i,j) \propto P(\text{percorso: } s_t = i \rightarrow s_{t+1} = j)$$

M-STEP

PARAMETER UPDATE

01

PROBABILITIES π

$$\pi_i = \gamma_1(i)$$

Probability of starting in state i

02

TRANSITION MATRIX

$$A_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

Expected number of transitions from state i to j, normalized over all transitions from state i

03

EMISSION MATRIX

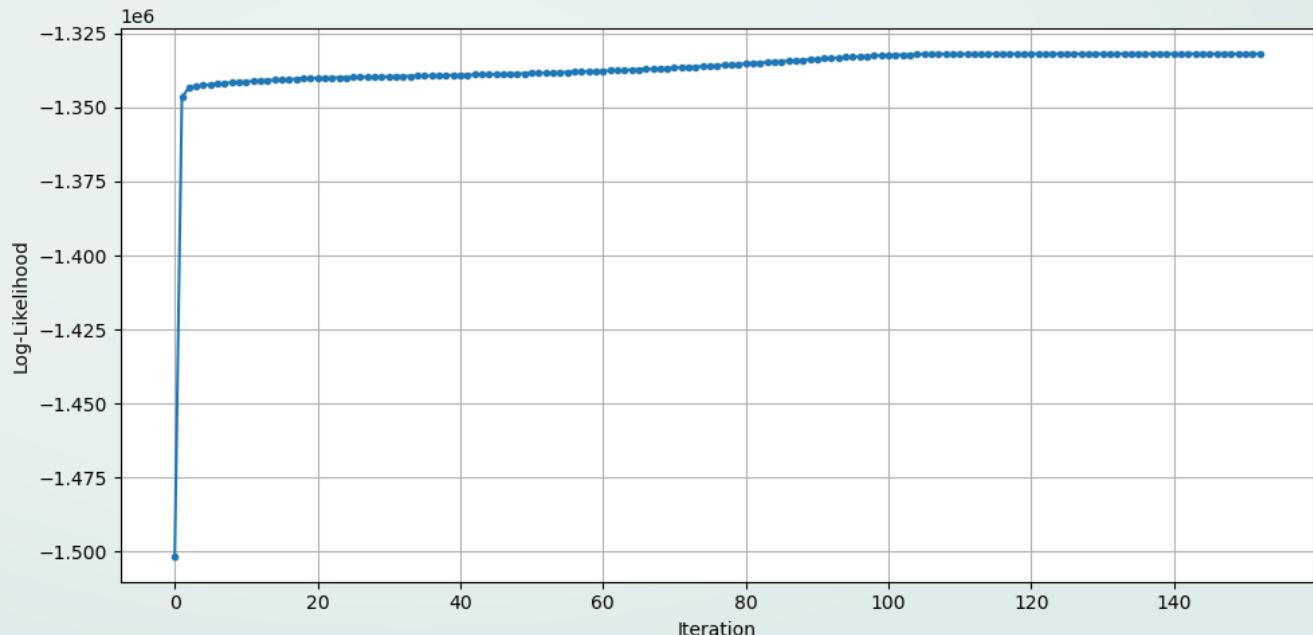
$$A_{ij} = \frac{\sum_{t=2}^T \mathbf{1}_{(o_{t-1}=a, o_t=b)} \cdot \gamma_t(j)}{\sum_{t=2}^T \mathbf{1}_{(o_{t-1}=a)} \cdot \gamma_t(j)}$$

Probability of emitting base b given the previous base a in state j, weighted by $\gamma_t(j)$

REPEAT

Iteratively improve the
model parameters until the
log-likelihood stabilizes

$$\log P(\mathcal{O}|\theta) < \varepsilon$$



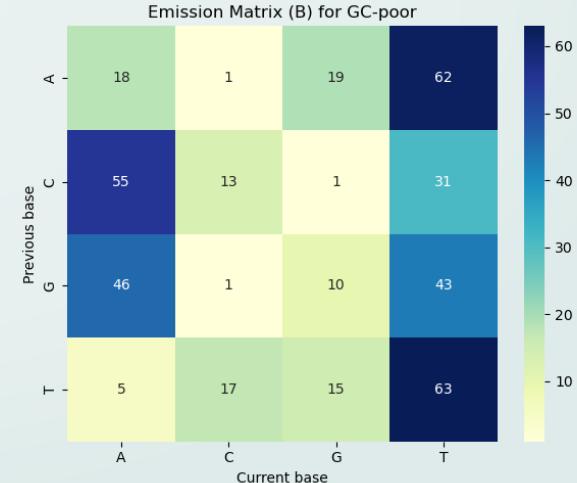
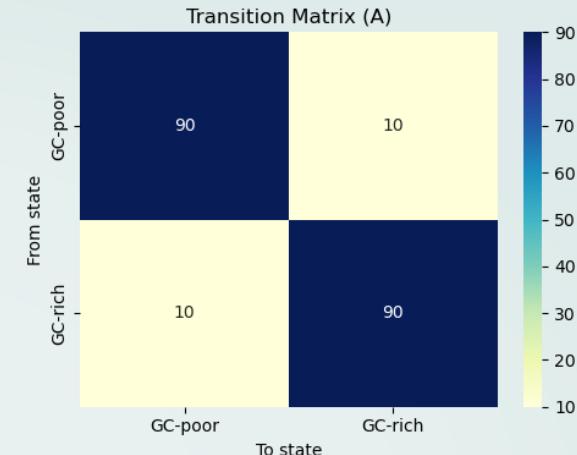
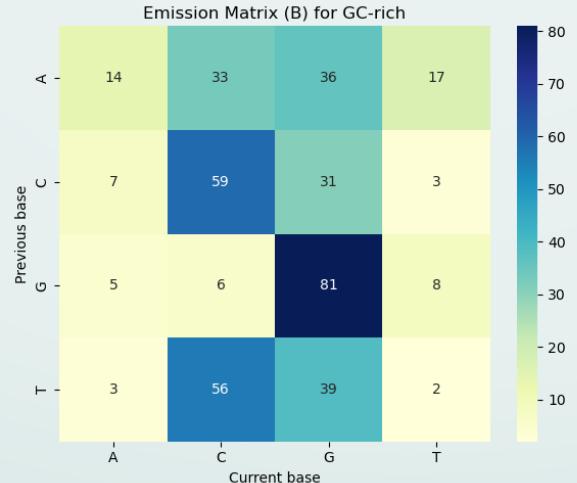
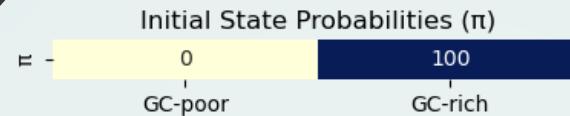
BAUM-WELCH RESULTS ON SYNTHETIC GENOME

AVERAGE ESTIMATION ERROR
PER PARAMETER

Initial State π : 73%

Transition Matrix A: 0.13%

Emission Tensor B: 1.3%



VITERBI ALGORITHM

Find the most probable sequence
of hidden states that explains the
observed sequence



VITERBI STATE ALGORITHM

INITIALITATION

$$\delta_j(1) = \pi_j \cdot B_j(o_1, o_2)$$

TERMINATION

$$s_T = \arg \max_j \delta_i(T)$$



INITIALIZATION

$\delta_j(t)$: highest probability of any path ending in state j at time t
 $\psi_j(t)$: previous state that leads to j with the highest probability

RECURSION

$$\delta_j(t) = \max_i [\delta_i(t-1) \cdot A_{ij}] \cdot B_j(o_t, o_{t+1})$$

$$\psi_j(t) = \arg \max_i [\delta_i(t-1) \cdot A_{ij}]$$

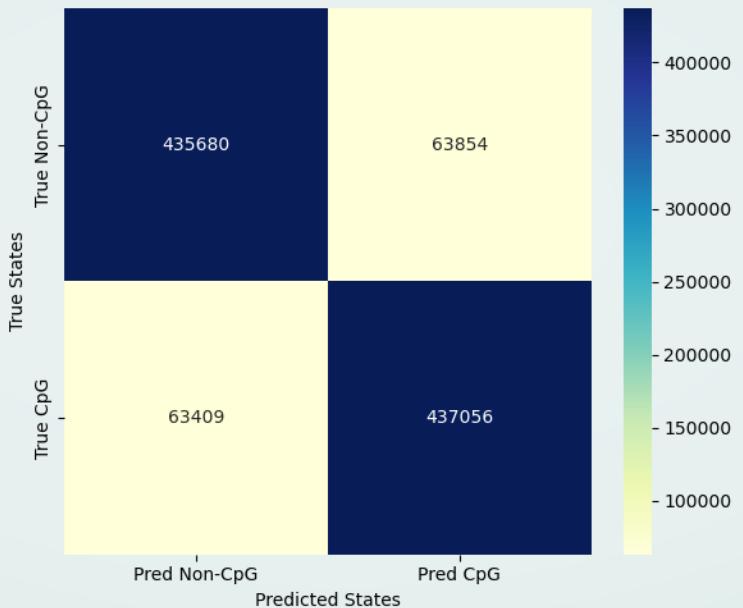
for $t = 2, \dots, T; j = 1, \dots, N$

BACKTRACKING

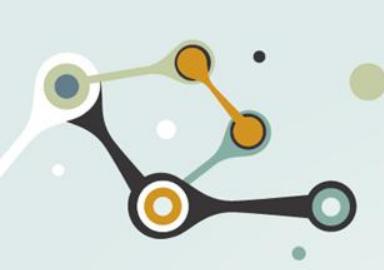
$$s_t = \psi_{s_{t+1}}(t+1)$$

for $t = T-1, \dots, 1$

VITERBI STATE RESULTS ON SYNTHETIC GENOME



ACCURACY = 87,3%



VITERBI BASE ALGORITHM

01

OBJECTIVE

Recover missing nucleotide bases (A, C, G, T)

02

INSERT GAPS

Gaps (_) are inserted randomly to simulate missing data

03

HANDLING GAPS

Emission probabilities are approximated in the presence of gaps
 $B_j(_) = \max_o B_j(o)$

04

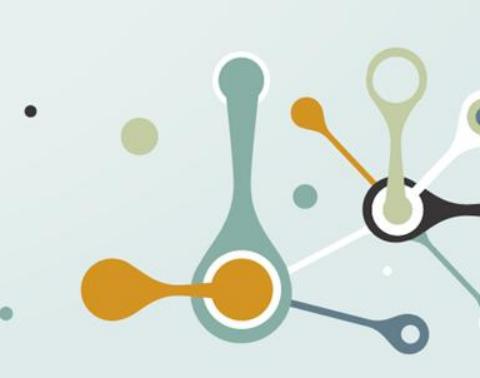
IMPUTATION

Use the **inferred state** and the **previous base** to select the base with the highest emission probability

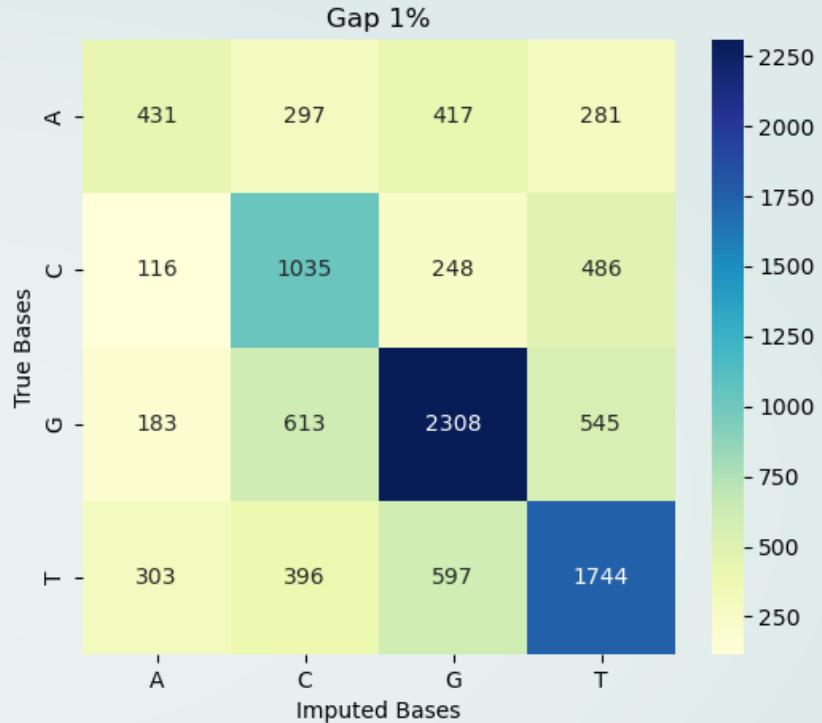
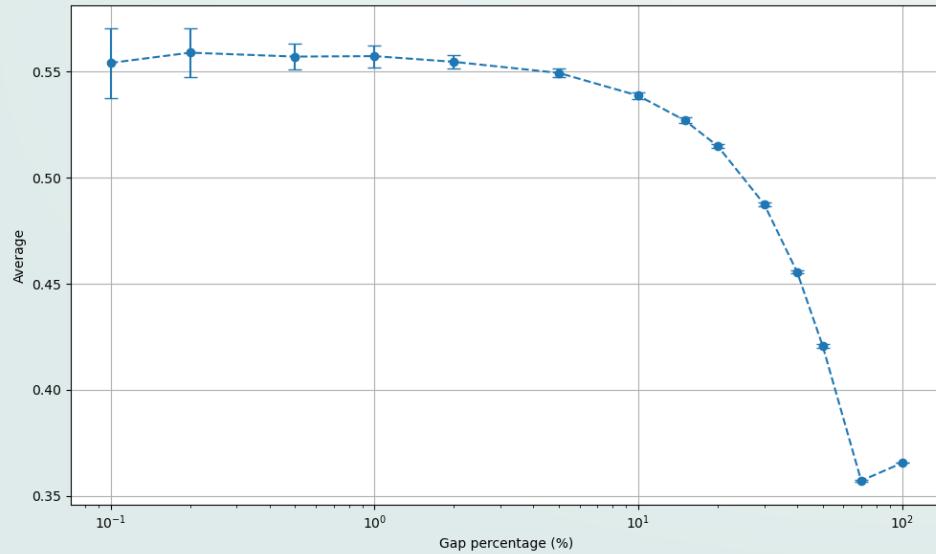
05

EVALUATION

Repeat the experiment at various **gap percentages**. Compute **accuracy** only on imputed positions



VITERBI BASE RESULTS ON SYNTHETIC GENOME

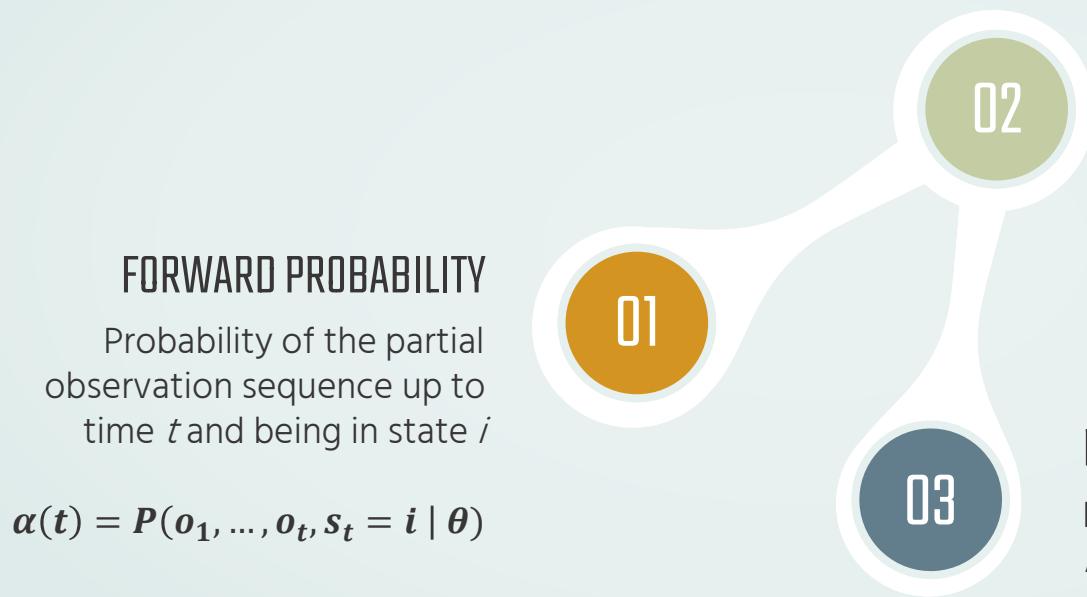


POSTERIOR ALGORITHM

Compute the most probable hidden state at each position, independently, given the observed sequence



POSTERIOR STATE ALGORITHM



FORWARD PROBABILITY

Probability of the partial observation sequence up to time t and being in state i

$$\alpha(t) = P(o_1, \dots, o_t, s_t = i | \theta)$$

BACKWARD PROBABILITY

Probability of the remaining observations from time $t+1$ given state i at time t

$$\beta(t) = P(o_{t+1}, \dots, o_T, s_t = i | \theta)$$

POSTERIOR PROBABILITY

Probability of being in state i at time t , given the whole observation sequence

$$s_t = \arg \max_i \gamma_t(i)$$

POSTERIOR STATE RESULTS ON SYNTHETIC GENOME



ACCURACY = 87,1%

POSTERIOR BASE ALGORITHM

HANDLING GAPS

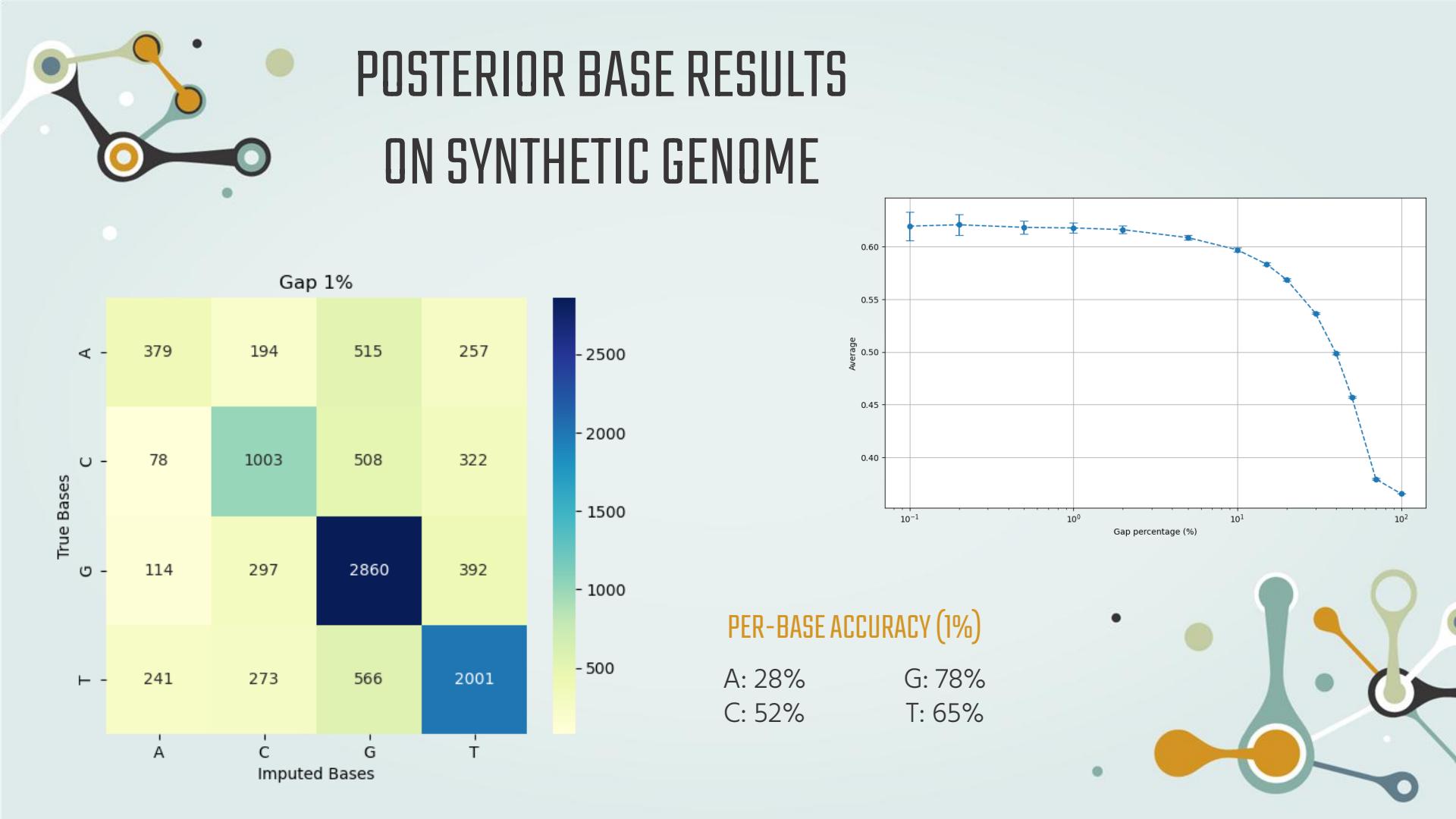
FORWARD AND BACKWARD

$$B_j(_) = \max_o B_j(o)$$

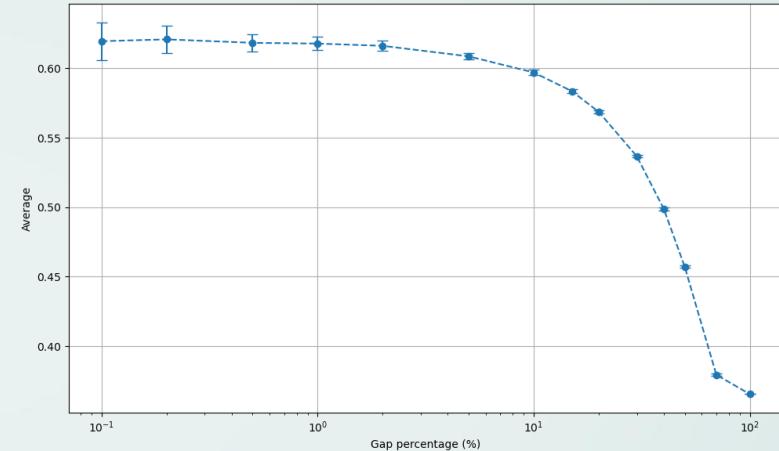
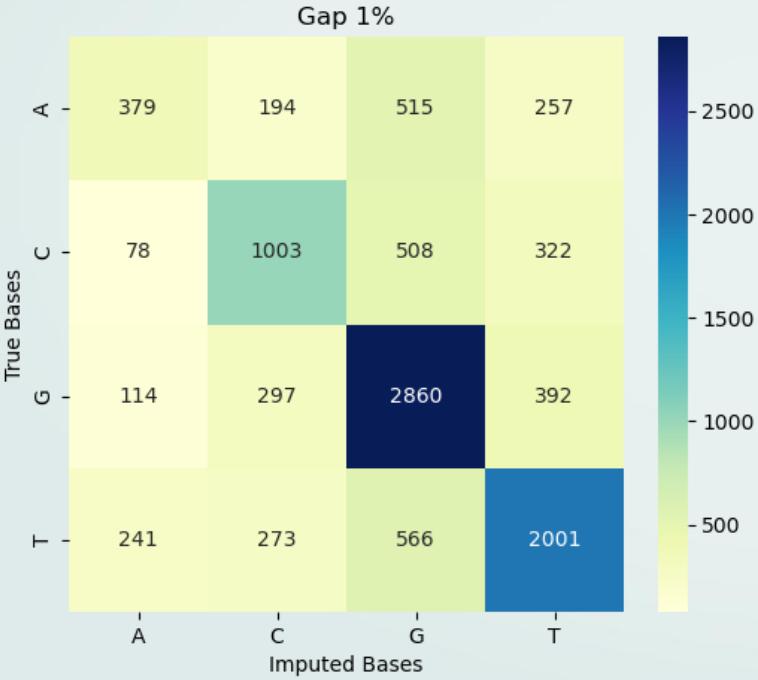
BASE IMPUTATION

When imputing a missing base b_t ,
we use posterior probabilities (γ)
in a **soft decision** strategy

$$\begin{aligned} Score(b) &= \max_j (\gamma_{t-1}(j) \cdot B_j(o_{t-1}, b)) \cdot \max_j (\gamma_t(j) \cdot B_j(o_{t+1}, b)) \\ b_t &= \arg \max_b Score(b) \end{aligned}$$

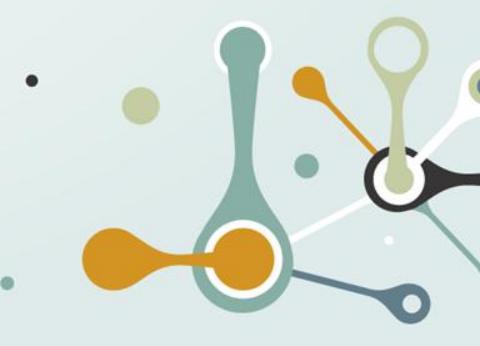


POSTERIOR BASE RESULTS ON SYNTHETIC GENOME



PER-BASE ACCURACY (1%)

A: 28% G: 78%
C: 52% T: 65%





04

RESULTS ON REAL GENOMES

Assessment of the results obtained on real genomic data and the associated issues

REAL GENOMES

ESCHERICHIA COLI

Bacteria

Coding: 4.0 Mbp

Non-coding: 0.64 Mbp



SACCHAROMYCES CEREVIAE

Fungus

Coding: 9.1 Mbp

Non-coding: 2.9 Mbp



HOMO SAPIENS CHR. 22

Animal

Coding: 1 Mbp (extracted: [5, 6] Mbp)

Non-coding: 1 Mbp (extracted: [4, 5] Mbp)



BACILLUS SUBTILIS

Bacteria

Coding: 3.72 Mbp

Non-coding: 0.49 Mbp



PNEUMOCYSTIS JIROVECII

Fungus

Coding: 5.4 Mbp

Non-coding: 3.1 Mbp



PER-BASE ACCURACY (1%)

V: Viterbi
P: Posterior
C: Coding
N: Non-coding

	ESCHERICHIA COLI	BACILLUS SUBTILIS	SACCHAROMYCES CEREVICIAE	PNEUMOCYSTIS JIROVECII	HOMO SAPIENS CHR 22
V C	29,6%	30,0%	34,9%	38,3%	35,8%
P C	31,5%	31,1%	34,8%	38,8%	33,3%
V N	31,2%	32,6%	37,8%	42,3%	35,3%
P N	29,8%	34,3%	37,9%	42,8%	37,1%

KEY CHALLENGES

ORDER

Real genome may not
be order 1

STATES

GC-rich/poor division
not always valid

LOCAL MAXIMA

Baum-Welch may converge
to suboptimal solutions

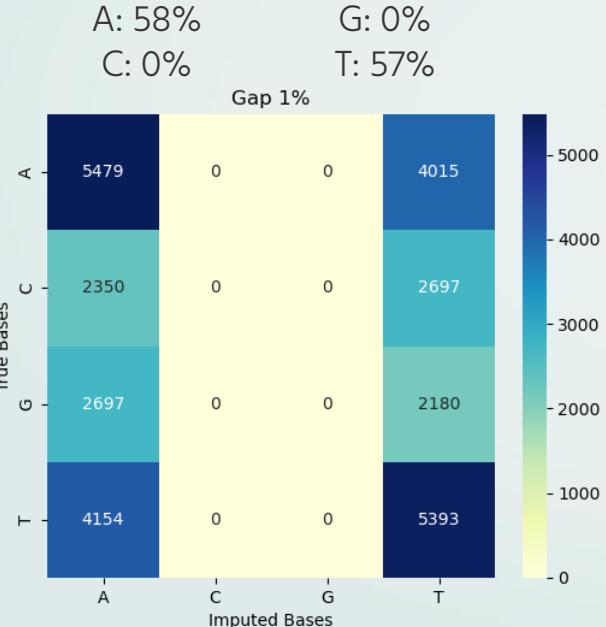
COMPLEXITY

Complex dependencies beyond
simple Markov assumptions

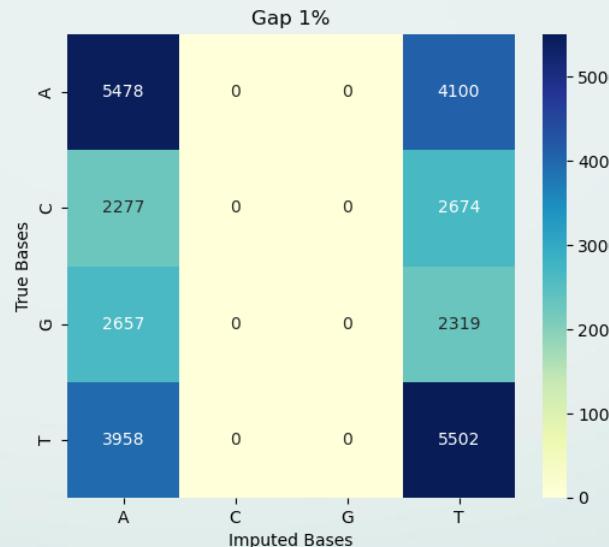


CHALLENGES IN BAUM-WELCH

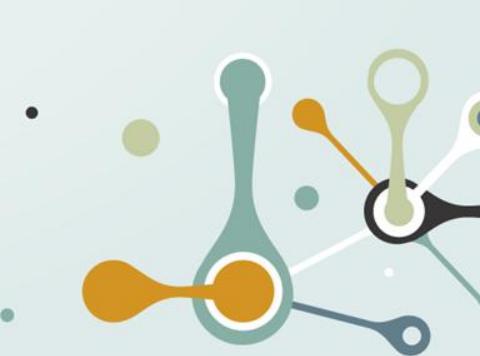
VITERBI SACCAROMYCES NON-CODING PER-BASE ACCURACY (1%)



POSTERIOR SACCAROMYCES NON-CODING PER-BASE ACCURACY (1%)



A first challenge is the genomic structure, which often differs from the expected model and increases the difficulty of parameter estimation in the Baum-Welch algorithm





05

CONCLUSION

Discussion of the obtained data
and possible future work

DISCUSSION

The algorithms perform well on **synthetic genomes**, built to match the expected structure.

On **real genomes**, results are less accurate due to hidden complexity and mismatch with model assumptions.

The **posterior algorithm** generally performs better than **Viterbi** thanks to its *soft* assignment strategy.

Non-coding regions are often easier to impute, likely due to simpler base patterns.



POSSIBLE FUTURE WORKS

MORE GENOMES

Analyze additional genomes to identify patterns across species

AMBIGUOUS BASES

Use IUPAC codes (e.g., N, S...) instead of _ to reduce ambiguity

STATE ADAPTATION

Adjust hidden states and training to better fit genomic features

01

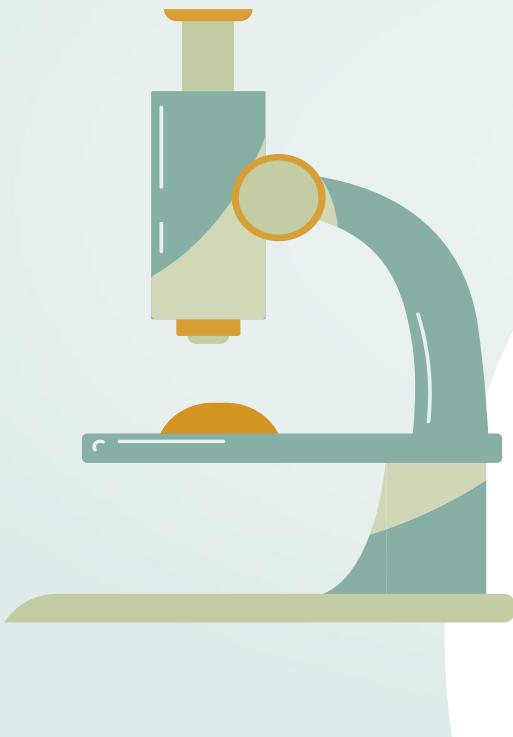
02

03





BIBLIOGRAPHY



Books:

- Statistical Methods in Bioinformatics
- Ewans
- Programmazione Scientifica - Barone

Sites:

- [Sgd-archive.org](#)
- [Ncbi.gov](#)
- [Ensemble.org](#)
- [Ebi.ac.uk](#)
- [Zenodo.org](#)

THANKS