ABIN End Sem Rubrics

The rubric is organized by question and sub-question. Each section includes criteria for earning full credit, partial credit, and no credit. Emphasis is placed on logical reasoning, clarity, correctness, use of appropriate biological and computational concepts, and the ability to integrate algorithmic thinking with biological insight.

General Grading Principles Across All Questions:

- **Clarity and Organization:** Responses should be logically structured, clearly stated, and use proper technical terms.
- **Depth of Reasoning:** Answers should demonstrate a conceptual understanding rather than mentions or abstract/vague working.
- **Biological and Computational Integration:** Students should show the ability to blend biological understanding with computational/algorithmic methodology.
- **Use of Examples and Diagrams (where requested):** Appropriate and correctly annotated figures, diagrams, or schematic representations gain additional credit.
- **Mathematical and Algorithmic Constructs:** When points are assigned for including mathematical formalisms, algorithms, or scoring schemes, these should be clearly explained.

Question 1

Part (a): Write the steps of the algorithm behind GSEA. (6 points)

Criteria for Full Credit (5-6 points):

- Clearly lists and describes the main steps of Gene Set Enrichment Analysis (GSEA):
    1. Rank all genes based on their differential expression (e.g., correlation with a phenotype).
    2. Walk down the ranked list, increasing an enrichment score (ES) when encountering genes in the gene set and decreasing it otherwise.

3.Identify the maximum deviation from zero as the ES.

- Demonstrates understanding of the reason behind each step and how the ES is calculated.
- Shows some insight into computational aspects (e.g., handling ties, data normalization).

Partial Credit (2-4 points):

- Some key steps mentioned but lack detail or missing a critical step (e.g., only mentions ranking and computing some score without indicating how the ES is computed).
- Conceptual misinterpretations but still some correct elements.

No Credit (0-1 points):

- Does not describe the algorithmic steps coherently.
- Lacks any understanding of the procedure of GSEA.

Part (b): Clearly describe the approach of testing statistical significance of GSEA scores. (4 points)

Criteria for Full Credit (3-4 points):

- Explains the null hypothesis and permutation approach (either permuting sample labels or gene sets) to estimate the significance of ES.
- Mentions calculation of a nominal p-value and possibly a multiple-testing correction approach (FDR).
- Shows understanding that significance is assessed by comparing observed ES to a distribution of ES from random permutations.

Partial Credit (1-2 points):

- Indicates that permutations are used but does not clearly state what is permuted.

- Vague understanding of statistical testing but lacks detail on how to generate the null distribution.

No Credit (0 points):

- No coherent description of the statistical significance testing process.

Question 2 (Option 1)

Part (a): Approach to finding the most likely sequence of functional regions. (2 points)

Full Credit (2 points):

- Identifies a model-based (e.g., Hidden Markov Model) approach to segment the sequence into exon, intron, splice site states.
- Shows understanding that each state has distinct nucleotide emission probabilities and there are transition probabilities between states.
- Includes a simple diagram or schematic (e.g., states representing exon/intron/splice site and arrows with probabilities).

Partial Credit (1 point):

- Recognizes the need for a probabilistic or pattern-recognition model but not specifically an HMM.
- Attempts to mention emission/transition patterns but is vague or incorrect in linking them to functional regions.

No Credit (0 points):

- No mention of a model-based approach or computational methodology.

Part (b): Propose an algorithm or step-by-step approach. (6 points)

Full Credit (5-6 points):

- Clearly describes using a Hidden Markov Model (HMM) or a related probabilistic algorithm:

1.Define states (exon, intron, splice-site, etc.).

2.Assign emission probabilities for nucleotides in each state.

3.Assign transition probabilities between states.

4.Use the Viterbi algorithm or similar dynamic programming approach to find the most likely path through the states for the given sequence.

- Includes some mathematical formalisms (e.g., probability of a path = product of transition and emission probabilities).
- May provide a short example and/or diagram showing transitions and states.

Partial Credit (2-4 points):

- Mentions using an HMM or related model but does not describe steps in detail.
- Shows some understanding of how to compute likelihoods but omits critical details such as choosing initial probabilities or how to implement the Viterbi algorithm.

No Credit (0-1 points):

- Purely heuristic suggestion with no mention of probabilistic or algorithmic methods.
- Completely off-topic solution.

Part (c): Challenges and how to address them. (2 points)
Full Credit (2 points):

- Identifies common issues: data sparsity, parameter estimation, choosing appropriate training data, complexity of the model.
- Suggests solutions such as smoothing, parameter estimation from large annotated datasets, or using known biological motifs.

Partial Credit (1 point):

- Mentions a challenge but no clear solution.
- Only minimal or vague ideas for addressing the challenge.

No Credit (0 points):

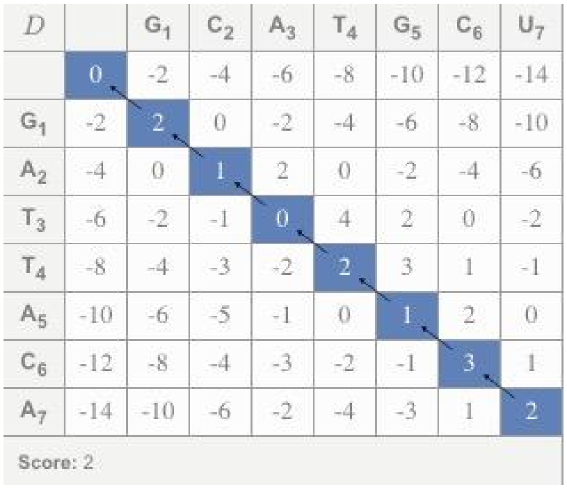
- No mention of challenges or how to solve them.

(Option 2 for Question 2)
If the student chooses the alignment and BWT tasks instead of the HMM-based segmentation:

Part (a): Needleman-Wunsch alignment (5 points)
Full Credit (4-5 points):

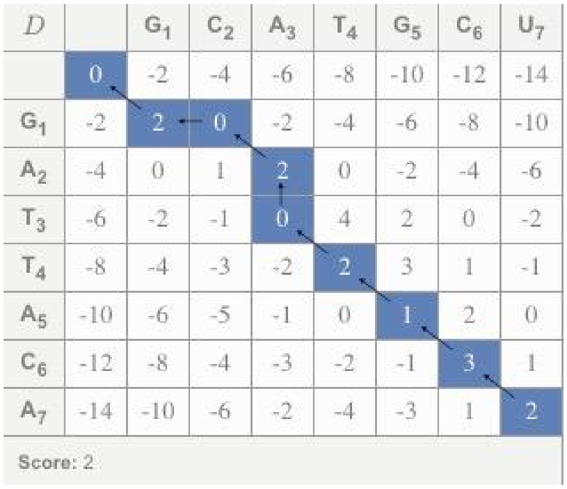
- Correctly sets up the scoring matrix for “GATTACA” vs. “GCATGCU” with match = +2, mismatch = -1, gap = -2.
- Correctly fills out the dynamic programming matrix and identifies the optimal alignment.
- Provides the final alignment and calculates the correct optimal score.

Any possible solution is correct:



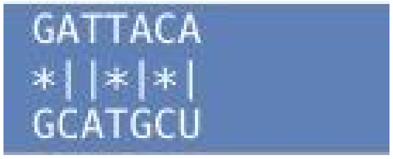
D		G_1	C_2	A_3	T_4	G_5	C_6	U_7
	0	-2	-4	-6	-8	-10	-12	-14
G_1	-2	2	0	-2	-4	-6	-8	-10
A_2	-4	0	1	2	0	-2	-4	-6
T_3	-6	-2	-1	0	4	2	0	-2
T_4	-8	-4	-3	-2	2	3	1	-1
A_5	-10	-6	-5	-1	0	1	2	0
C_6	-12	-8	-4	-3	-2	-1	3	1
A_7	-14	-10	-6	-2	-4	-3	1	2

Score: 2

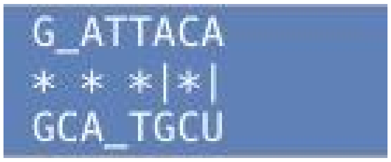


D		G_1	C_2	A_3	T_4	G_5	C_6	U_7
	0	-2	-4	-6	-8	-10	-12	-14
G_1	-2	2	0	-2	-4	-6	-8	-10
A_2	-4	0	1	2	0	-2	-4	-6
T_3	-6	-2	-1	0	4	2	0	-2
T_4	-8	-4	-3	-2	2	3	1	-1
A_5	-10	-6	-5	-1	0	1	2	0
C_6	-12	-8	-4	-3	-2	-1	3	1
A_7	-14	-10	-6	-2	-4	-3	1	2

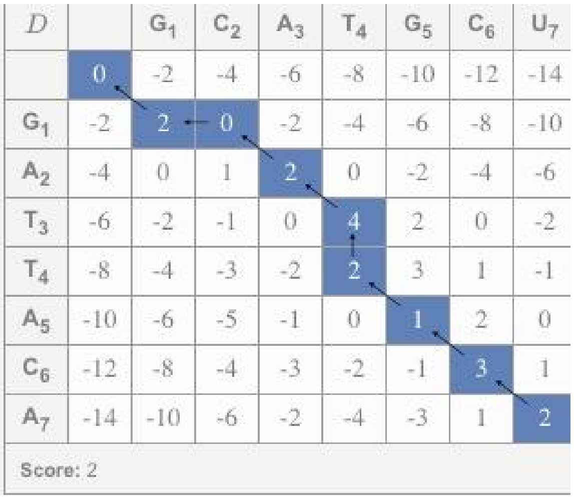
Score: 2



```
GATTACA
*||*|*|
GCATGCU
```

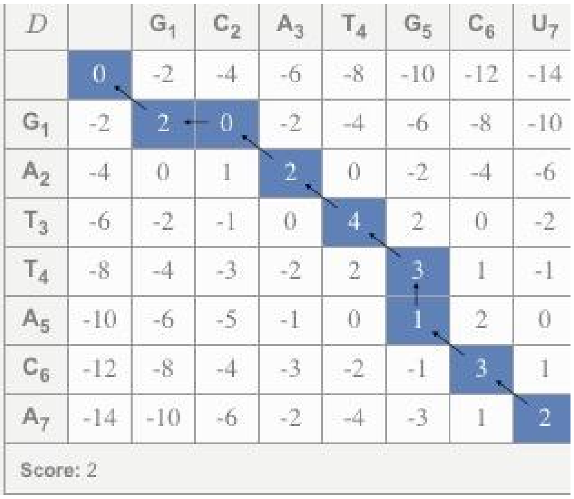


```
G_ATTACA
* * *|*|
GCA_TGCU
```



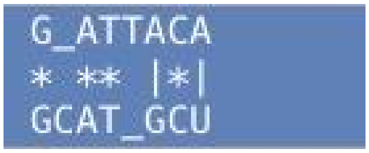
D		G_1	C_2	A_3	T_4	G_5	C_6	U_7
	0	-2	-4	-6	-8	-10	-12	-14
G_1	-2	2	0	-2	-4	-6	-8	-10
A_2	-4	0	1	2	0	-2	-4	-6
T_3	-6	-2	-1	0	4	2	0	-2
T_4	-8	-4	-3	-2	2	3	1	-1
A_5	-10	-6	-5	-1	0	1	2	0
C_6	-12	-8	-4	-3	-2	-1	3	1
A_7	-14	-10	-6	-2	-4	-3	1	2

Score: 2

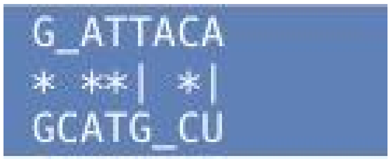


D		G_1	C_2	A_3	T_4	G_5	C_6	U_7
	0	-2	-4	-6	-8	-10	-12	-14
G_1	-2	2	0	-2	-4	-6	-8	-10
A_2	-4	0	1	2	0	-2	-4	-6
T_3	-6	-2	-1	0	4	2	0	-2
T_4	-8	-4	-3	-2	2	3	1	-1
A_5	-10	-6	-5	-1	0	1	2	0
C_6	-12	-8	-4	-3	-2	-1	3	1
A_7	-14	-10	-6	-2	-4	-3	1	2

Score: 2



```
G_ATTACA
* ** |*|
GCAT_GCU
```



```
G_ATTACA
* **| *|
GCATG_CU
```

Partial Credit (2-3 points):

- Attempts the Needleman-Wunsch matrix but makes some arithmetic errors.
- Gets a plausible alignment but incorrect final score or some off-by-one errors.

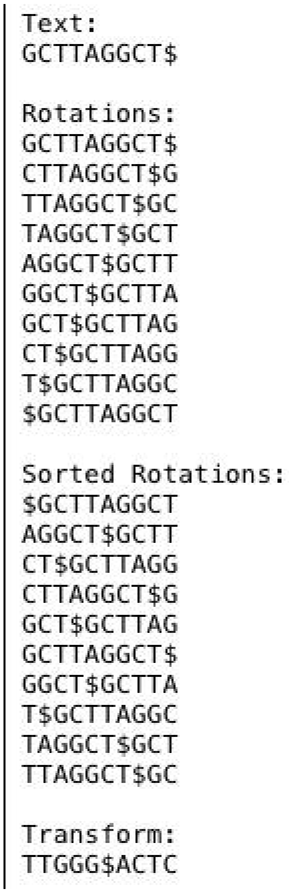
No Credit (0-1 points):

- No coherent attempt at the dynamic programming table.
- Completely incorrect method.

Part (b): Perform the BWT (5 points)
Full Credit (4-5 points):

- Shows the rotations of “GCTTAGGCT$”.
- Correctly sorts the rotations and identifies the last column to produce the BWT.
- Final BWT is correct.

Solution :



```
Text:
GCTTAGGCT$

Rotations:
GCTTAGGCT$
CTTAGGCT$G
TTAGGCT$GC
TAGGCT$GCT
AGGCT$GCTT
GGCT$GCTTA
GCT$GCTTAG
CT$GCTTAGG
T$GCTTAGGC
$GCTTAGGCT

Sorted Rotations:
$GCTTAGGCT
AGGCT$GCTT
CT$GCTTAGG
CTTAGGCT$G
GCT$GCTTAG
GCTTAGGCT$
GGCT$GCTTA
T$GCTTAGGC
TAGGCT$GCT
TTAGGCT$GC

Transform:
TTGGG$ACTC
```

Partial Credit (2-3 points):

- Understands the concept of BWT but makes errors in either listing rotations or sorting them.
- Produces a partially correct transformation but the final result is off.

No Credit (0-1 points):

- No understanding of the BWT process.

Question 3

Part (a): Normalization, HVG finding, clustering, DE analysis in scRNA-seq. (4 points)
Full Credit (4 points):

- Explains biological rationale: normalization corrects for sequencing depth/technical variation, identifying highly variable genes (HVG) isolates biologically interesting variation, clustering groups cells with similar expression patterns, differential expression reveals genes that vary significantly between clusters or conditions.
- Provides biologically relevant justifications for each step, not just technical definitions.

Partial Credit (2-3 points):

- Describes the steps but lacks a biological perspective (e.g., only a technical explanation without biological context).
- Misses one of the steps or provides a weaker rationale.

No Credit (0-1 points):

- No accurate explanation of these steps or their biological purpose.

Part (b): Importance of mate-pair sequencing in genome assembly. (2 points)
Full Credit (2 points):

- Clearly explains that mate-pair reads span large genomic distances, helping to resolve repeats and scaffolding contigs into larger assemblies.
- Uses a diagram to show how long-range information can link distant contigs.

Partial Credit (1 point):

- Recognizes that mate-pair reads help in scaffolding but provides no visual or minimal explanation.

No Credit (0 points):

- No mention of how mate-pairs improve assembly or just incorrect reasoning.

Part (c): Why shortest superstring fails for human genomes. (4 points)
Full Credit (3-4 points):

- Explains that the shortest superstring approach is too simplistic given repetitive elements in the human genome. It may collapse repeats incorrectly, failing to represent the true genome structure.
- Provides a concrete example, e.g., highly repetitive sequences that appear identical and cause the shortest superstring approach to merge distinct genomic regions incorrectly.

Partial Credit (1-2 points):

- Recognizes shortest superstring is not suitable due to complexity and repeats but does not provide a concrete example.
- Missing a clear explanation of how repeats break the model.

No Credit (0 points):

- No accurate explanation or reference to genomic complexity and repeats.

Question 4

Part (a): Prioritizing driver mutations (2 points)
Full Credit (2 points):

- Proposes a computational framework that considers both mutation frequency differences (cancer vs. normal) and network connectivity.
- Suggests ranking sites by a combined score that weighs mutation frequency difference and network centrality or influence.

Partial Credit (1 point):

- Mentions using mutation frequency and/or network data but no clear integrated scoring or ranking strategy.

No Credit (0 points):

- No coherent approach to prioritize mutations.

Part (b): Accounting for neighboring sites' influence (2 points)
Full Credit (2 points):

- Suggests a model (e.g., graph-based approach or Markov Random Field) where the importance of a site depends on its neighbors' frequencies and their known regulatory interactions.
- Could mention diffusion scores or network propagation methods.

Partial Credit (1 point):

- Mentions neighboring effects but only vaguely (e.g., "consider neighbors" without specifying how).

No Credit (0 points):

- No reference to incorporating neighbor information.

Part (c): Iterative refinement of predictions (6 points)

Full Credit (5-6 points):

- Suggests iterative updating of site importance scores based on refined parameters, e.g., re-estimating mutation significance after removing known passenger mutations or re-weighting network edges.
- Proposes a feedback loop: initial ranking → filter based on biological validation → re-run model with updated parameters.
- Could mention semi-supervised learning or iterative EM-like approaches.

Partial Credit (2-4 points):

- Recognizes the need to refine results but does not detail a clear iterative methodology.
- Only a partial iterative approach is described.

No Credit (0-1 points):

- Does not describe any iterative or refinement process.

Question 5 (10 points)

Steps of a fast rare cell detection algorithm

Case 1: Outlines FiRE algorithm

Full Credit (9-10 points)

The response must include:

1. **Preprocessing and Dimensionality Reduction (Optional):**
    - Description of preprocessing steps like normalization and filtering of data.
    - Mention of dimensionality reduction techniques such as selecting highly variable genes or PCA for noise reduction.
2. **Sketching and Hashing-Based Density Estimation:**

- Explanation of **Sketching**, where high-dimensional data is mapped to binary hash codes (bit vectors).
- Description of how cells are assigned to “buckets” based on hash codes, approximating their densities in high-dimensional space.
- Mention that density is computed across multiple passes (estimators) for variance reduction.

3. **FiRE Score Calculation:**

- Formula for computing the FiRE score:

$$\text{FiRE score}_i = -2\sum_{l=1}^{L} \log_e(p_{il})$$

  - Explanation that higher scores indicate rare cells based on lower density in their buckets.

4. **Thresholding and Rare Cell Identification:**

- Use of IQR-based statistical thresholds to classify rare cells:

$$\text{Rare if FiRE score} \geq Q_3 + 1.5 \times \text{IQR}$$

- Mention of downstream analysis or validation methods like biological marker checks or clustering for rare subtypes.

5. **Advantages of FiRE Algorithm:**

- Clearly states that FiRE is computationally efficient (O(N)), avoids clustering, and provides a continuous rarity score for fine-grained analysis.

Partial Credit (4-8 points)

- **Dimensionality Reduction:** Mentioned but lacks detail or omits why it is necessary. (1 point)
- **Sketching and Hashing:** Partial understanding, e.g., mentions buckets and hashing but no explanation of density estimation or multiple estimators. (1-3 points)

- **FiRE Score Calculation:** Formula mentioned but not adequately explained. (1 point)
- **Thresholding and Rare Cell Identification:** Vague or incomplete explanation of thresholding or rare cell selection. (1 point)
- **Advantages:** Recognizes that FiRE avoids clustering or mentions scalability but does not explain its significance. (1 point)

No Credit (0-3 points)

- Does not describe preprocessing, dimensionality reduction, or hashing techniques.
- Fails to mention the FiRE score calculation or its purpose.
- Lacks a coherent explanation of how rare cells are identified.
- No mention of computational efficiency or advantages of the FiRE algorithm.

Case 2: Outlines any other algorithm
Criteria for Full Credit (9-10 points):

- Outlines a clear, step-by-step algorithm:
  1. Preprocessing of single-cell data (e.g., normalization).
  2. Dimension reduction (e.g., PCA) to capture variation.
  3. Density-based or graph-based clustering to identify clusters. How this algorithm would be faster than normal clustering algorithms which take n^2 time.
  4. Statistical tests or anomaly detection metrics (e.g., Mahalanobis distance, DBSCAN outliers) to detect rare groups of cells.
  5. Validation steps and thresholding criteria to find rare cells.
- Integrates mathematical constructs (e.g., defining a probability distribution, using a likelihood ratio test for rarity).
- Includes diagrams of clusters and indicates rare clusters visually.

- Shows understanding of complexity and possible optimizations.

Partial Credit (4-8 points):

- Proposes a logical method but lacks detail in one or more steps.
- Mentions some mathematical methods or equations but not well integrated.
- Provides a plausible approach but missing a key step (e.g., no clear thresholding or no mention of computational efficiency).

No Credit (0-3 points):

- Vague approach with no computational strategy.
- Lacks any mathematical or algorithmic detail.

Overall, good grading is awarded for:

- Integrate biological reasoning and computational methods.
- Provide step-by-step logic where algorithms are asked for.
- Use diagrams & mathematical depictions where requested..