**📊 1️⃣ Bioinformatics Code Generation Prompt 1 (VCF → allele frequencies)**

| **Aspect** | **GPT-4** | **Claude** | **Gemini** |
| --- | --- | --- | --- |
| Correct basic functionality | ✅ | ✅ | ✅ |
| Filtering for bi-allelic SNPs | Partial (ALT simplified) | ✅ (via filter function) | ✅ (explicit SNP mask) |
| Modularity | ❌ (monolithic script) | ✅ (modular functions + utilities) | ✅ (modular, with VCF generation demo) |
| Extra features | No | Summary stats + plotting + filtering | Sample VCF creation + full comments |
| Code clarity | Good | Excellent | Excellent |
| Usability for real pipelines | Basic | High | High |

**Summary**:

* **Claude** and **Gemini** produced richer, more reusable, and better-commented code than GPT-4 here.
* **Gemini** added a very nice educational component by creating a demo VCF file.
* **Claude** added reusable plotting and SNP filtering helpers.

**📊 2️⃣ Bioinformatics Code Generation Prompt 2 (PCA of gene expression)**

| **Aspect** | **GPT-4** | **Claude** | **Gemini** |
| --- | --- | --- | --- |
| Correct basic PCA pipeline | ✅ | ✅ | ✅ |
| Data scaling | ✅ | ✅ | ✅ |
| Plotting | Basic scatterplot | Professional plot (with optional coloring, cumulative explained variance) | Basic but labeled scatterplot |
| Extra features | None | Cumulative variance, top gene contributors, variance explained plot | Creates sample CSV with biologically meaningful pattern |
| Code modularity | Basic | Excellent (full pipeline) | Good (includes example data generation) |
| Educational clarity | Good | Excellent | Excellent |

**Summary**:

* **Claude** clearly wins here — most comprehensive and reusable pipeline, includes advanced analysis.
* **Gemini** did a nice job with sample data generation to ensure the plot would show meaningful separation (great for demos).
* **GPT-4** was correct but quite minimal.

**📊 3️⃣ Pipeline Explanation Prompt 1 (RNA-seq DESeq2)**

| **Aspect** | **GPT-4** | **Claude** | **Gemini** |
| --- | --- | --- | --- |
| Step coverage | ✅ | ✅ | ✅ |
| Depth of explanation | Medium | High | High |
| Plain language clarity | Good | Excellent (uses analogies effectively) | Excellent (analogy + real-world example) |
| Visual aids suggested | Yes (workflow) | Yes (plots + PCA emphasis) | Yes (plots + diagrams) |
| Added real-world example | No | Yes (COVID-19 study) | No |

**Summary**:

* **Claude** provided the **most accessible** and **student-friendly** explanation, with **analogies** and a **real research example**.
* **Gemini** was also very clear, with a strong analogy-driven explanation.
* **GPT-4** was correct and well-structured but drier.

**📊 4️⃣ Pipeline Explanation Prompt 2 (GATK Best Practices)**

| **Aspect** | **GPT-4** | **Claude** | **Gemini** |
| --- | --- | --- | --- |
| Step completeness | ✅ | ✅ | ✅ |
| Detail level | High | Very high | Very high |
| Example commands included | Some | Comprehensive | Comprehensive |
| Data format explanations | Yes | Excellent | Excellent |
| Variants of pipeline (exome, somatic) | No | Yes | No |
| Educational clarity | Good | Excellent | Excellent |

**Summary**:

* **Claude** provided the **most comprehensive and teachable guide**, suitable for real-world use or a tutorial.
* **Gemini** also did very well, with a great breakdown of data formats and key concepts.
* **GPT-4** covered the pipeline correctly but with less pedagogical depth.

**📊 5️⃣ Health Data Debugging Prompt 1 (Kaplan-Meier in R)**

| **Aspect** | **GPT-4** | **Claude** | **Gemini** |
| --- | --- | --- | --- |
| Correct code | ✅ | ✅ | ✅ |
| Professional plotting | ✅ (ggsurvplot) | Extensive (ggsurvplot + rich stats) | Basic but correct |
| Added statistical tests | Yes (p-value) | Yes (log-rank test, median survival, extended analysis) | Yes |
| Educational clarity | Good | Excellent | Good |
| Reusability | High | Very high | High |

**Summary**:

* **Claude** delivered an **extremely professional** full clinical-grade K-M analysis pipeline.
* **GPT-4** was solid and correct but more minimal.
* **Gemini** was also correct but simpler.

**📊 6️⃣ Health Data Debugging Prompt 2 (SQL join for latest patient visit)**

| Aspect | GPT-4 | ✅ | ✅ | ✅ |  
| Provided ROW\_NUMBER() window function | ✅ | ✅ | ✅ |  
| Provided GROUP BY alternative | ✅ | ✅ | ✅ |  
| Provided correlated subquery | ❌ | ✅ | ✅ |  
| Discussion of LEFT JOIN / no-visits case | ❌ | ✅ | ✅ |  
| Handling of tie-breaks | ❌ | ✅ | ✅ |  
| Code clarity | Good | Excellent | Excellent |

**Summary**:

* **Claude** and **Gemini** gave **exceptionally good SQL debugging guides**, covering multiple edge cases and solutions.
* **GPT-4** provided a solid basic answer but did not cover as many edge cases or variations.

**Overall Performance Ranking (based on your test set):**

| **Rank** | **Model** | **Strengths** |
| --- | --- | --- |
| 🥇 1 | **Claude** | Best explanations, most comprehensive code, great for educational + production code |
| 🥈 2 | **Gemini** | Creative outputs, excellent data generation for examples, very clear |
| 🥉 3 | **GPT-4** | Solid and correct, but more minimal and less pedagogically polished in these cases |

**Conclusion:**

✅ **Claude** is the best choice when you want:

* Explanatory, teaching-friendly outputs
* Professional-grade analysis pipelines
* Coverage of edge cases

✅ **Gemini** is a great choice when you want:

* Reproducible example code
* Creative demos (e.g. sample VCF, sample PCA CSV)
* Clear explanations

✅ **GPT-4** is:

* Consistently correct
* Reliable for quick code generation
* Less comprehensive unless specifically prompted to go deeper