FALSACIÓN DE LA TEORÍA EVOLUTIVA CON FLIPP

José del Carmen Rodríguez Santamaría
https://www.ejristos.com
Bogotá, Colombia

14 de diciembre de 2021

Resumen

Una teoría es falsa cuando produce predicciones erróneas. Exhibimos una predicción muy importante y genérica de la Teoría de la Evolución (TE) que es contradicha por la Naturaleza. Encontramos que la Evolución de la Perfección es obligatoria y que la sub-perfección es inmanente a la Evolución cuando se enfrenta a la complejidad, el enemigo natural de la Evolución. Como esto no se observa en la Naturaleza, concluimos: La Evolución Natural no tiene nada que ver con nuestros orígenes. Nuestra falsación es definitiva porque captura la esencia de la Evolución Natural, que se apoya en dos pilares: el Genoma es software, un conjunto de instrucciones codificadas, y la Evolución es un desarrollador de software. Por eso se utilizó como banco de pruebas la Programación Genética (PG), el uso de la Evolución para diseñar software. Nuestra PG utiliza Flipp, un lenguaje de programación especialmente diseñado para imprimir cadenas binarias, para las que complejidad y longitud son sinónimos.

Índice

1. IDEA GENERAL **3**

2. ALGORITMOS GENÉTICOS (AGs) **4**
2.1. La perspectiva histórica 4
2.2. Un sofisma poderoso 5
2.3. Objeciones contra los AGs 6
2.4. Nuestro propósito 6

3. EL GENOMA ES SOFTWARE **7**
3.1. El poder de la palabra 7
3.2. Computadores y software 7
3.3. Informática del genoma 8

4. ALGORÍTMOS POR TODOS LADOS **10**
4.1. Los animales deben resolver tareas algorítmicas 10

5. LA EVOLUCIÓN ES UN DESARROLLADOR DE SOFTWARE **11**
5.1. Interpretación Algoritmica de la diversidad humana 11

6. LA TE VERBAL **11**
6.1. Formulación 12
6.2. ¿Es natural la TE Verbal? 12
6.3. El reto de la Falsificación de la TE 13

7. PONIENDO A PRUEBA LA TE VERBAL **14**
7.1. Una predicción increíble 14
7.2. Alta calidad y extrema complejidad como test 14
7.3. Perfección total y omnipresente como test 15

8. EL ORDENADOR FLIPP **17**

9. EL COSTO DE LA PERFECCIÓN **18**
9.1. ESPEC como una abstracción 18
9.2. El costo de ESPEC 18
9.3. El surgimiento evolutivo de los genes 19

10. EL COSTO DE LA IMPERFECCIÓN **20**
10.1. Nuestras simulaciones 20

11. EL ROMPECABEZAS DE LA TELEONOMÍA **21**

12. BUGS **23**
12.1. Caminatas aleatorias en PG sobre Flipp 24
12.2. La dinámica en PG sobre Flipp 24
12.3. El Tapiz de la Perfección 25

13. CONTEXTUALIZACIÓN **26**
13.1. Relación con el creacionismo 26
13.2. Relación con la Ciencia 27
13.3. Relación con las personas 27
13.4. ¡Jesus viene! 28

14. CONCLUSIÓN **28**

1. IDEA GENERAL

¡Qué bonito es ver a los campeones en sus exhibiciones! Nos muestran el impresionante poder de la especie. En la raíz de estas increíbles realizaciones reside la Evolución: dos cambios aquí, un reordenamiento allá y el resultado puede representar eventualmente una mejora. A veces surge una tendencia, las mejoras se acumulan y aparecen maravillas. Asombroso. Esto afecta a nuestra vida y hábitos personales, a la Cultura que nos rodea y al Genoma. Así pues, la Evolución se encuentra por todas partes.

La acumulación de mejoras podría explicarse por nuestro fuerte instinto de detectar y elegir lo mejor. Estamos tan acostumbrados a ello que se ha necesitado un genio para alinear decenas de observaciones bajo la idea de que en la Naturaleza no es necesario ningún agente de selección porque la elección de lo mejor se produce espontáneamente:

En primer lugar, siempre hay más descendientes de los que pueden sobrevivir, y los que lo hacen de forma recurrente portan rasgos especiales, que pueden variar de un individuo a otro. En segundo lugar, los supervivientes se reproducen, y transmiten sus rasgos ganadores. Tercero: cuando el proceso se desarrolla generación tras generación, surgen la perfección y la complejidad. El proceso global se llama Evolución, mientras que el proceso emergente en el que se favorece la reproducción de los mejores se denomina Selección Natural. Esta fue la piedra fundamental sobre la que Darwin construyó su Teoría Evolutiva (TE) del origen de las especies ([4] 1859).

En cuanto a la corroboración de la TE, la Evolución Biológica Natural forma parte de nuestra realidad cotidiana como lo demuestra la Evolución de los virus. Además, se ha aplicado en muchos y muy diversos campos. En realidad, las simulaciones de la Evolución en forma de Algoritmos Genéticos (AGs) son herramientas ordinarias para encontrar soluciones aproximadas a problemas de optimización. Nuestro interés particular aquí está en un tipo especial de AGs en el que la Evolución se aplica al desarrollo de software, una disciplina floreciente y desafiante que se llama Programación Genética (PG). Recordemos que software es el nombre genérico que reciben los textos que llevan instrucciones que pueden ser decodificadas y ejecutadas por la CPU de un ordenador. Así pues, la esencia de la PG es que estos textos también pueden sufrir una Evolución, igual que los de ADN. Así, la TE predice que la creación de software mediante la Evolución deberá ser fructífera. Esta es en la actualidad una predicción completamente verificada y nuestras propias simulaciones aquí hacen que la verificación sea accesible para computadoras de escritorio y teléfonos celulares.

La relación del sofware con la Biología es que la Evolución Natural es PG Natural. Esta idea proviene de una simple observación: los textos de ADN transmiten instrucciones que deben ser interpretadas y ejecutadas por el ribosoma para sostener y continuar la vida. Nuestro punto es que hay una equivalencia del ribosoma con una CPU y del Genoma con el Software. Por lo tanto, la Evolución del Genoma es de hecho PG Natural. Podemos ahora actualizar la TE en términos de desarrollo de Software. El

resultado se denomina la **TE Verbal**, la cual afirma: *la Evolución es el desarrollador de Software que es el único responsable de la creación del software de todos los genomas de todas las especies en la Tierra.* Para ponerla a prueba, calculamos sus predicciones con simulaciones en JavaScript y Flipp, un lenguaje de programación especialmente adecuado para imprimir cadenas binarias. Nuestro objetivo es ver cómo es la batalla en PG de la Evolución contra la complejidad. Tras una investigación tediosa pero al fin sencilla, encontramos que la subperfección debe ser ubicua y que la Evolución de la Perfección es obligatoria en la PG como observable permanente. Esto equivale a una predicción de la TE Verbal: nuestros cuerpos deberán estar llenos de imperfecciones y nuestra fisiología de disfunciones. Como esto no se encuentra en nuestros cuerpos, concluimos que la Evolución no puede ser la explicación de nuestra existencia. En la filosofía de la ciencia, a esto se le llama una falsación (McLeod [23] 2020): la TE ha sido falsada por la PG sobre Flipp.

Desarrollemos ahora nuestra visión.

2. ALGORITMOS GENÉTICOS (AGs)

La pregunta que nos orienta es si la complejidad y la perfección de los seres vivos pueden ser explicadas por la Naturaleza. Muchos pensadores nos han ayudado con discusiones increíbles. En nuestra opinión, la cima de la claridad la captan los AGs.

2.1. La perspectiva histórica

Los seres humanos son bautizados con dos nombres, uno de la familia más otro personal. Linneo propuso en 1753 bautizar a las especies de la misma manera: un género, el grupo general en el que se encuadran -como `pino` o `Pinus`-, y un nombre personal, como `lambertiana` (Pearson [24] 2003). Detrás de esta idea de contabilidad antropológica, hay una poderosa conceptualización: las criaturas pueden organizarse según características comunes en una jerarquía anidada (BerkeleyUE [3] 2010?). Pasar de ella a árboles era para Lamarck una especie de deber con el plus de interpretar el árbol en términos de descendencia, de Evolución biológica. Propuso a principios del siglo XIX una Teoría de la Evolución (TE) en la que ésta era incapaz de superar la complejidad por sí sola y, por tanto, necesitaba ser guiada por fuerzas alquímicas. De hecho, Lamarck era adepto a la alquimia que prefería a la química cuantitativa de Lavoiser (Wikipedia [36] 2015). Sin embargo, cuando se busca el mecanismo como las fuerzas alquímicas interactúan con la materia viva para guiar su Evolución, el único indicio posible está en los flujos nerviosos, que él enfatiza demasiado como la clave de su teoría (Lamarck [19] 1809).

A continuación, Darwin ([4] l.c., 1859) propuso la Teoría de la Evolución (TE) en la que se postulaba la Evolución como un anulador de la complejidad gracias a la Selección Natural, la reproducción diferencial de los individuos. Carecía de un mecanismo de herencia. Posteriormente, Mendel fue un idealista que hacia 1865 forzó experimentos de campo con avichuelas para sostener su pensamiento de que la herencia está ligada a la transmisión de elementos materiales (Science Learning Hub – Pokapū Akoranga Pūtaiao [31] 2011). Su idea resultó ser correcta y derivó en la moderna Biología Molecular con el ADN, los genes y demás.

Las ideas de Darwin pueden mezclarse de varias maneras con la genética de Mendel. En primer lugar, las esperanzas de Darwin pueden realizarse en la Genética si se asume que la mutación puede detenerse en aquellos sitios que ya obtuvieron un valor correcto: tenemos aquí una mutación guiada, es decir, lamarckismo. La Teoría Evolutiva resultante es milagrosa, tal y como esperaba Darwin (Rodríguez, [29] 2021). En retrospectiva, podemos decir *Darwin + Mendel = Lamarck*. Sin embargo, la idea de una mutación guiada fue seriamente combatida por de Vries en 1901, quien trabajó principalmente con plantas. Afirmó que el medio ambiente no puede guiar la dirección de la mutación, que se produce a saltos. De este modo, liberó la Evolución de las ideas de Lamarck y se opuso al gradualismo de Darwin (Hall [9] 1935). Otra forma de mezclar a Darwin con Mendel se propuso en la Teoría Sintética, formulada hacia 1930, y que para todos los efectos equivale actualmente a la actual Genética de Poblaciones (Jorde [14] 2016). En ella, la mutación no es guiada, sino que es al azar. Esta nueva reformulación de la TE es aceptada por la Ciencia Moderna y se proclama como la explicación de nuestra existencia.

En la segunda mitad del siglo XX, los Científicos de la Computación y los Ingenieros elaboraron el Algoritmo Genético (AG) que es una simulación de la TE en los computadores (Rodríguez [28] 2019). Según mi experiencia, nadie puede estar seguro de la correcta comprensión de la TE si no ejecuta con papel y lápiz un AG. Es muy difícil que los estudiantes simulen la mutación al azar porque tienden inadvertidamente a guiarla. Como piensan que la mutación debe ser guiada, todos ellos aceptan que estamos aquí gracias a la Evolución. En cambio, los AGs utilizan generadores de números aleatorios que son suficientemente buenos y con ellos se llega fácilmente a varias conclusiones:

- La Evolución en los AGs se abstrae en términos de mutación, recombinación, supervivencia diferencial, reproducción y muerte. La idea fundamental de los AGs es que la Evolución puede resolver problemas de optimización si los individuos representan soluciones aproximadas y si éstos pueden reproducirse en proporción a la calidad de la aproximación.
- Los AGs han demostrado ser un poderoso método para encontrar soluciones aproximadas a problemas de optimización de cualquier naturaleza. Esto ha creado un sofisma, véase más adelante.
- La Evolución puede encontrarse en todas partes porque el mecanismo representado por los AGs es simple y omnipresente.
- La diferencia entre la mutación controlable y la aleatoria es que en el primer caso la Evolución va hacia arriba siempre de forma milagrosa. En el segundo, la mutación deshace por la noche lo que hace por el día. El resultado es que la Evolución va hacia arriba gracias a la Selección Natural, pero tan lentamente que ya no puede ser un anulador de la complejidad. Más bien:
- La complejidad es una carga para la Evolución. Por lo tanto, si se mejoran las soluciones, esto podría ocurrir muy, muy lentamente.

2.2. Un sofisma poderoso

Los AGs se utilizan todos los días en todo el mundo para atacar problemas de Ciencia e Ingeniería. Por eso se cree y se enseña universalmente que los AGs son una

verificación de la TE y que por lo tanto la TE es verdadera. Sin embargo, esto es un sofisma. En cambio, la verdad es esta:

- La TE sería falsada inmediatamente si los AGs no funcionaran porque los AGs implementan la idea principal de la TE.
- Los AGs funcionan y por lo tanto, esto es realmente una verificación de la TE.
- Si se verifica una teoría, ésta puede ser de todas formas falsa. En el caso que nos ocupa, la verificación es muy potente y, por tanto, se predice que muchos rasgos de la vida pueden ser correctamente explicados y reproducidos por la Evolución. Es tarea de la Ciencia dilucidar mediante el Método Científico qué rasgos encajan en esta clase.
- A la luz de los AGs, la TE tiene una predicción obligatoria que es falsa. La predicción obligatoria es que cuando la Evolución puede resolver un problema complejo, la Evolución de la Perfección debe existir necesariamente: las soluciones sintetizadas al azar son aproximaciones bastante burdas que son convertidas por la Evolución en soluciones mediocres que luego evolucionan a soluciones buenas que luego luchan por la perfección. Esto lleva tiempo. Esto debe tomar mucho tiempo, tanto que se predice por la TE que la imperfección debe abundar en todas partes, tanto en el registro fósil como en nuestro propio cuerpo, anatomía y funciones. Como esto no se ve en la inmensa mayoría de los bebés de todas las especies, la TE es falsa (Rodríguez [29] 2021).

Aprendí de las virtudes y defectos de la Evolución gracias a los AGs. Mientras que la mayoría de la gente los presenta como una poderosa prueba de la TE, nosotros exigimos que se aclare que los AGs demuestran que la Evolución es, efectivamente, útil, pero no la explicación de nuestra existencia, porque la Evolución está lejos de ser verde: el Universo carece de recursos suficientes para hacer funcionar los AGs para explicar nuestra complejidad. Ahora nos preguntamos: ¿Qué tan fuerte es la falsación de la TE por medio de los AGs?

2.3. Objeciones contra los AGs

Uno imagina espontáneamente que los AGs tienen algo que ver con la Naturaleza. Tal creencia se oscurece al observar que los AGs reflejan imperfectamente la Evolución Natural. La visión correcta, justa y transparente de la Evolución Natural tiene en cuenta dos factores adicionales. En primer lugar, la Evolución Natural trata de instrucciones codificadas para continuar la vida. Por lo tanto, es verbal, algorítmica por su propia naturaleza. Segundo: los AGs tienen como propósito la optimización de la función que codifica el problema a resolver, pero la Naturaleza no tiene propósito alguno según la filosofía materialista que sigue la Ciencia. Así y estrictamente, la Ciencia no puede utilizar los AGs ni para apoyar la TE ni para rechazarla. Por eso necesitamos un nuevo marco, el correcto.

2.4. Nuestro propósito

El ADN contiene instrucciones codificadas para la continuación de la vida existente. Así pues, el Genoma es un software y la Evolución de estos textos es de hecho la Evolución del Software. Ahora estamos en posición de reformular la TE en términos

conceptuales y esenciales. El resultado es la **TE Verbal** que elaboramos a continuación. Es muy humana y refleja de forma transparente la Evolución como fenómeno informático. Además, si la claridad se elabora con cuidado, la falsación de la TE debe surgir espontáneamente en la mente del Lector. Nuestra responsabilidad aquí es demostrar que esto es reproducible para cualquier persona con un computador de mesa y según los estándares del Método Científico. Nuestros programas se ejecutan en los navegadores web incluso en los teléfonos móviles, si sólo se selecciona un nivel de complejidad bajo.

Entramos ahora a desarrollar la parte algorítmica de la **TE Verbal** y permitir la formulación de tests de acuerdo al Método Científico.

3. EL GENOMA ES SOFTWARE

El ADN contiene las instrucciones verbales necesarias para sintetizar proteínas, i.e., el ADN es software.

3.1. El poder de la palabra

Podemos conversar. Aunque parezca sencillo, es un poderoso milagro. Veamos.

El profesor le dice a su alumno: *levanta la mano derecha*. El niño levanta la mano derecha. El profesor dice: *canta tu canción preferida*. El niño canta una canción de cuna. El profesor dice: *come y bebe*. El niño se come una manzana. El maestro dice: *suma 2 más 2*. El niño responde: *cuatro*. El maestro dice: *hemos terminado, vete a casa*. El pequeño termina todas las tareas, prepara su maletín, y se va a casa.

El increíble hallazgo es que la comunicación verbal puede codificarse y automatizarse. Basta con observar su mecánica: una instrucción se codifica en una cadena de sonidos verbales o fonemas, se interpreta según una tabla o código determinado, y se ejecuta la instrucción. Esta tripleta es el sello de la comunicación verbal, véase la figura 1.

3.2. Computadores y software

El término **verbal** se refiere a una cadena de sonidos que transmite información. Muchas culturas descubrieron que la comunicación verbal puede codificarse en textos que consisten en cadenas de caracteres o símbolos. La relación de los textos con los enunciados verbales está mediada por un código de interpretación que es caprichoso. Sin embargo, el conjunto está optimizado de forma diversa, por ejemplo, el ruido externo debe causar pocas interferencias. Los textos pueden contener instrucciones que conforman un procedimiento, un programa que al ejecutarse produce acciones.

Inmediatamente captamos una conexión con los ordenadores:

Un computador es una máquina que puede ejecutar tareas verbales de la más variada naturaleza, al igual que nosotros. Por ejemplo, se puede ordenar el envío de un

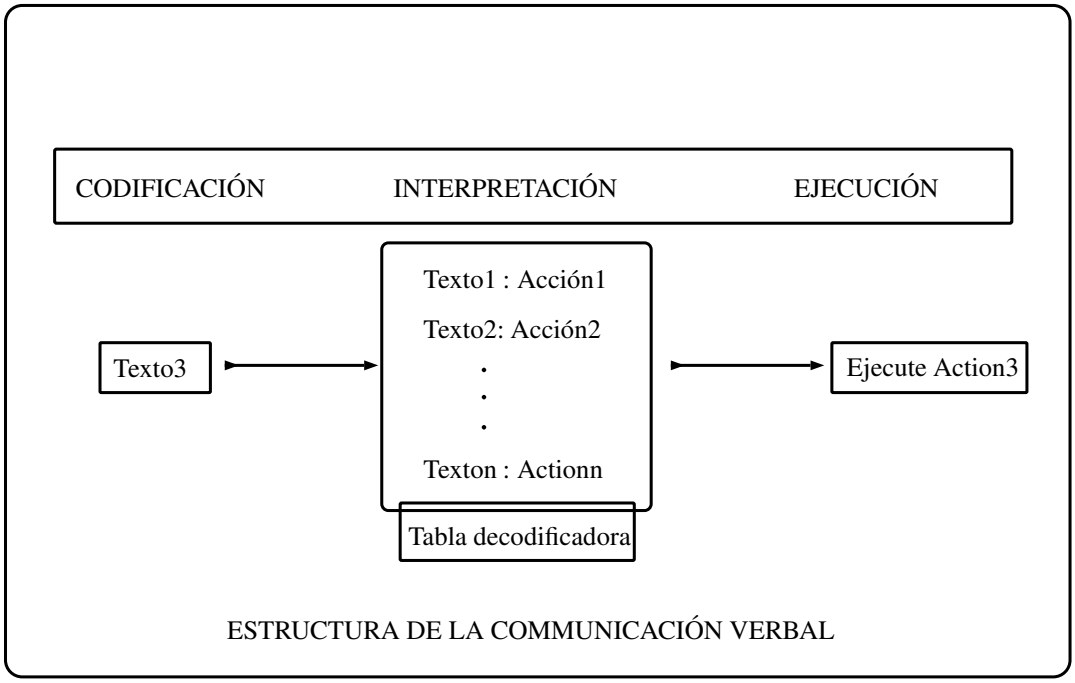


Figura 1: *Una instrucción verbal se codifica en un texto, se interpreta según una tabla o código determinado, y se ejecuta. Esta tripleta es universal y ha permitido el diseño de todo tipo de ordenadores.*

correo electrónico, pasar a reproducir música, continuar con el cálculo del balance financiero de una empresa, se sigue con la simulación de un avión en vuelo, para terminar con el cierre de todas las operaciones.

Las definiciones elementales de la Informática son las siguientes:

- El hardware es el conjunto de componentes físicos de un ordenador.
- Un programa es una lista de instrucciones verbales, funciones, métodos o procedimientos para operar un computador.
- El software se compone de los programas para su funcionamiento.
- Un robot es un ordenador capaz de tomar decisiones y convertirlas en acciones. Puede gozar de un grado extraordinario de autonomía.
- Un libro es un paquete de software para ayudar al ser humano a programar su mente para mejorar. El ser humano es un robot bastante extraño, tanto que le encanta afirmar que es libre y que, por tanto, es responsable de sus actos.

Ahora, una conexión con la Biología.

3.3. Informática del genoma

La célula es el elemento fundamental de la vida. El ADN de la célula funciona con procesamiento verbal. Por ejemplo, el ADN de las células del estómago comienza a

ensamblar una proteína y dice:

Inicialice una cadena polipeptídica con `Ala`. La célula inicializa una cadena polipeptídica con Alalina. El ADN dice: *añada* `Leu`. La célula añade un aminoácido de Leucina a la cadena. El ADN dice: *añada* `Ile`. La célula añade un aminoácido de Isoleucina... La célula dice: *añadir* `Ala` y se añade Alalina a la cadena. La célula dice: *marcar la cadena para que sea exportada fuera de la célula*. La célula marca la cadena con la señalización adecuada. El ADN dice: *finalizar*. La célula termina la cadena y la libera para su posible procesamiento posterior. Y todo esto se habría desencadenado por la necesidad de fabricar más pepsina para digerir los alimentos en el estómago de un cerdo.

La forma en que se codifican las órdenes en el ADN se denomina **código genético** (NIH, [22] 2021).

1 ***Ejercicio.*** *Demuestre que el Código Genético se ajusta a la estructura general de la comunicación verbal. Respuesta*

Así, vemos que una célula es una entidad robótica. Esto significa que la célula tiene un ordenador interno que se basa en el procesamiento verbal. Está programado para permitir una toma de decisiones autónoma pero interactiva, como la auto-alimentación y la reproducción.

La prueba más asombrosa de que la célula depende de procesamiento verbal es que puede seguir sobre la marcha las órdenes más extrañas.

Ejemplo con bacterias. Puede llegar un fago, un virus asesino de bacterias, que invade la célula bacteriana, despliega su propio material genético, la célula lo capta, e inmediatamente la célula comienza a sintetizar aquellas proteínas y material genético que permiten el ensamblaje espontáneo de más virus de la misma clase y que muy pronto destruirán la célula. Esto se utiliza en medicina como terapia de fagos contra el poder de la Evolución que viene aquí en forma de resistencia a los antibióticos (Kurzgesagt [18] 2018). Con respecto a un mamífero específico, decenas de tipos diferentes de virus podrían infectar sus células. Por ejemplo, los murciélagos son extremadamente tolerantes a los virus y unos 50 tipos diferentes los infectan (Racaniello [25] 2013).

Si la célula es un robot, ¿qué pasa con su computador? Algunos detalles del superordenador celular son los siguientes: las moléculas de ADN sirven como *memoria de acceso aleatorio de solo lectura*, y guardan los mensajes genéticos o genes. Una molécula de ADN es una doble cinta, cada una de las cuales es una cadena de bases químicas y que funciona igual que nuestros textos escritos pero con sólo 4 letras: ATCG. Los textos genéticos pueden estar organizados en libros o rollos que llamamos cromosomas. Estos forman el **Genoma.** Las cadenas genéticas, para ser útiles, necesitan una potente maquinaria que decodifique la información verbal que contienen según un código determinado. Para ello, las moléculas de ADN se transcriben a cadenas de m-ARN, y la maquinaria de decodificación agrupa los residuos de a 3, subcadenas que reciben el nombre de codones, y que constituyen órdenes verbales elementales. La información es leída y ejecutada por los ribosomas, que funcionan como la CPU multinúcleo de un ordenador. Los productos de los ribosomas son cadenas de péptidos, algunos de los cuales sirven como elementos estructurales y otros como enzimas. Los ribosomas utilizan una tabla para la traducción, el **código genético**, que desempeña

el papel de una tabla hash de interpretación. El código está escrito en los t-ARN, que son sintetizados por las aminoacil-t-ARN sintetasas. Así, estas enzimas son por fin los portadores del Código Genético.

La diversidad de los individuos se basa en la naturaleza verbal del genoma: un conjunto de instrucciones produce un individuo, pero un conjunto ligeramente diferente produce otro que posiblemente sea muy distinto. La naturaleza verbal del genoma es también explotada por la ingeniería genética que produce modificaciones localizadas que afectan a mensajes específicos.

Con esto, hemos argumentado que **el Genoma es software y que la célula es una entidad robótica**.

La palabra `software` es básica en Ciencias de la Computación, pero necesitamos reconocer que el software tiene que ver con algorítmos, recetas o procedimientos, y que por tanto, nuestras vidas están llenas de ellos.

4. ALGORÍTMOS POR TODOS LADOS

Los humanos vivimos desarrollando software a toda hora, como por ejemplo, procedimientos o recetas para cocinar. Estamos desarrollando software cada vez que nos preguntamos: ¿Cómo puedo hacer eso? Pero no somos los únicos.

4.1. Los animales deben resolver tareas algorítmicas

Cuando planificamos un viaje para ir de la casa al centro de la ciudad debemos diseñar un algoritmo que consta de una lista de instrucciones, algunas en serie, cuando deben ejecutarse una tras otra, otras en paralelo, cuando hay que considerar caminos o medios de transporte alternativos. Esto demuestra que los algoritmos son constitutivos en nuestra vida. A los animales les ocurre lo mismo. Precisamente, los ratones han demostrado ser excelentes en la tarea de explorar laberintos. ¿Implica esto que el cerebro animal utiliza redes neuronales para desarrollar software? Esta posibilidad es justo una búsqueda permanente en el oficio (Nahla Davies [5] 2020).

2 ***Ejercicio.*** *Es importante darse cuenta de que toda nuestra vida está hecha de software, de instrucciones para hacer cosas, de conocimientos técnicos. Esto también es válido para los animales. Por ejemplo, los rebaños de búfalos son tan buenos en el diseño óptimo de rutas que los primeros raíles que cruzaban los Estados Unidos se trazaron siguiendo sus caminos en su migración. ¿Implica esto que el cerebro animal puede simular algoritmos genéticos? Mientras tanto, diviértase con el juego para computadores desarrollado por* `w3schools` *y que tan amablemente proporcionan en línea (w3schools [35] 2020). El juego consiste en desarrollar el software para diseñar la ruta para conseguir el premio. La complejidad crece suavemente. Preste atención a la siguiente configuración para decidir cuándo se trata de un software: Existe un conjunto de símbolos o de palabras reservadas que son el* `lenguaje de programación`*. Una lista de símbolos o palabras reservadas es un* `programa`*. La tabla que asigna los comandos a los símbolos o palabras es el* `código de interpretación`*. Una interpretación símbolo por símbolo o palabra por palabra se hace por un* `interpretador`*. Cuando se toma todo el programa*

como una unidad, con posibles determinantes sintácticos o anotaciones, se trata de un `compilador`*. La* `ejecución` *la realiza una Unidad Central de Proceso (CPU) o un microprocesador.*

Afirmación. Todos nosotros estamos muy ocupados en la creación de software, recetas o listas de instrucciones, en todo momento y en todas partes. Nuestra vida es tan dura debido a que está llena de tareas de ejecución de algoritmos y/o de su desarrollo. Eso implica que todo ser humano es un experto en el desarrollo de software y, por lo tanto, tiene la autoridad para juzgar lo que decimos (si somos capaces de ser claros).

5. LA EVOLUCIÓN ES UN DESARROLLADOR DE SOFTWARE

Veamos ahora cómo la Biología Molecular también se ocupa del desarrollo de software.

5.1. Interpretación Algoritmica de la diversidad humana

El contenido de nuestro ADN es diferente de una persona a otra. Así, todos tenemos Genomas diferentes, un conjunto distinto de instrucciones codificadas que nos permiten ser lo que somos. Esa diferenciación es la que permite hacer pruebas de paternidad (Adams [1] 08). Incluso los clones tienen un ADN diferente al de sus madres (Jiang, [13] 2011). Por otro lado, las diferencias constituyen una fracción pequeña de todo el genoma y se explican en su mayoría por sustituciones sinónimas (Thaler, [34]).

Fijémonos ahora: la diversidad humana equivale a una Evolución de algoritmos porque nuestras instrucciones genéticas han evolucionado, han demostrado ser viables, y hemos sido capaces de sobrevivir y de llenar la Tierra. Esto demuestra que es un hecho de la Naturaleza que **La Evolución es un desarrollador de software:** a partir del software, la Evolución ha producido más software. Esto nos parece encantador y maravilloso.

6. LA TE VERBAL

Si consideramos la increíble semejanza de las entrañas de una gallina con las de un cerdo, sentiríamos que tenemos el sagrado deber de reinventar la Teoría Evolutiva (TE) que debe afirmar: la gallina y el cerdo tienen un ancestro común. Esa teoría se abstrajo a los AGs que representaban una formulación descriptiva de la TE. Pasar de ella a la quintaesencia conceptual de esa teoría es inmediato. Como tanto el Genoma como la Evolución son algorítmicos, la formulación correcta de la TE debe ser también verbal, algorítmica.

Presentamos esa nueva formulación en 2010 (Rodríguez, [26]), a la que adjuntamos nuestras razones, la mayoría de sentido común, para pensar que esta nueva postura proporciona a los investigadores una falsación inmediata de la TE. Aquí repetimos la formulación de la teoría, la TE Verbal, más un laboratorio en PG para ponerla a prueba. Todas los tests se hicieron en JS y Flipp, un lenguaje de programación para imprimir cadenas binarias. Todos los programas se ejecutaron sobre los navegadores web Firefox o Chrome bajo Ubuntu 20.04, Windows 10 y Android 10 y 11. Se han ejecutado sobre ordenadores de sobremesa con 8 o 16 GB de ram y teléfonos móviles con 4 MB. Y con

8 o 12 núcleos a casi 2GH, pero esto es menos importante. No se han utilizado GPUs. El tiempo de duración de cada ejecución fue de segundos a horas y días. Se pueden explorar diversos niveles de complejidad para ver cuál se adapta mejor a las propias posibilidades. Aumentando un poco la complejidad, un programa se ejecutaría durante semanas en lugar de horas.

6.1. Formulación

La **TE Verbal** es la completación directa de una descripción trasparente tanto del Genoma como de la Evolución Natural. Veamos como emerge de una revisión de nuestro material:

- Los seres vivos son objetos que existen gracias a la ejecución de un programa informático que está escrito en el ADN de una célula germinal que en el caso de los humanos resulta de la fusión de un espermatozoide con un óvulo. El programa completo se llama **Genoma**. Estamos diciendo que **el Genoma es software**. Pero un lenguaje tiene mucho más que procedimientos para hacer cosas, tiene poesía así que, el ADN también tiene mucho más que genes para hacer enzimas de forma regulada, interactiva. Por ejemplo, el ADN debe ser compatible con las operaciones de mantenimiento y logística, una tarea que plantea demasiadas exigencias a las hebras de ADN. Con esta precaución, decimos que el Genoma es software.

- A nivel molecular, **la Evolución consiste en la mutación y la recombinación** de los textos del ADN. La mutación cambia los textos y la recombinación compone nuevos textos concatenando partes de otros. Una parte de los textos evolucionados es aceptada por la maquinaria celular ejecutante y, por tanto, se clasifican como nuevas instrucciones de programación, como nuevos algoritmos.

- Dado que **el Genoma es software,** y puesto que la Evolución puede hacer cambios en los algoritmos genéticos para producir nuevas instrucciones que eventualmente pueden ser beneficiosas, debemos concluir que **la Evolución es un desarrollador de software**. Esto no es una teoría. Es un hecho de la Naturaleza gracias al cual existimos y somos únicos.

- Sobre esta base, **la Teoría de la Evolución Verbal (TE Verbal) dice: La Evolución es el desarrollador de software que es el responsable del origen de los genomas de todas las especies.** La TE Verbal arrastra consigo todos los estudios y creencias de la TE ordinaria. Es una actualización que aquí es más técnica que filosófica. En particular, el proceso evolutivo es natural, automático y materialista.

6.2. ¿Es natural la TE Verbal?

Debemos preguntarnos ahora en voz alta:

¿Es nuestra formulación algorítmica sólo un interesante punto de vista académico o es natural, con implicaciones industriales y de supervivencia?

Nuestra respuesta es:

En primer lugar, el punto de vista algorítmico sobre la Evolución es la descripción correcta del mundo Genético: el Genoma es un montón de instrucciones verbales para continuar la vida. Ser verbal también significa que el Código Genético parece una asociación arbitraria. Sin embargo, se ha afirmado que el Código optimiza propiedades muy importantes (Freeland, [7] 1998). De todos modos, sería muy difícil demostrar que algunas permutaciones dentro del Código Genético no serían tan buenas como la existente.

En segundo lugar, la visión algorítmica ofrece una explicación sencilla del hecho de que la industria farmacéutica esté siempre al borde de ser derrotada por la Evolución de las bacterias y los virus. ¿Por qué? Concretamente y con respecto a las bacterias, tenemos:

Posiblemente con una sola mutación, una bacteria podría dejar de ser el objetivo de un antibiótico. Ella adquiere así resistencia a los antibioticos. Ahora bien, ¿por qué la empresa farmacéutica no hace un pequeño cambio en el antibiótico para remediar la situación? Porque para la empresa no hay ningún cambio minúsculo: los antibióticos son el fruto de una complicada cadena de reacciones o, en nuestras palabras, cualquier producto de la empresa es el resultado de un algoritmo. Por lo tanto, para hacer un cambio minúsculo la empresa debe hacer grandes cambios correlativos en todo su algoritmo. Y esto lleva tiempo, mucho tiempo y gastos muy elevados. Desarrollar un nuevo antibiótico cuesta miles de millones.

En tercer lugar, el razonamiento anterior se aplica a toda la vida en lo que se refiere a la adaptación y muy especialmente a nuestra época en la que la actividad humana provoca calentamiento y un clima errático: algunos cambios en el hábitat pueden ser contrarrestados por algunos cambios minúsculos en algunas pocas enzimas. No hay problema. Pero esta capacidad de adaptación se rompe más bien pronto que tarde porque cambios correlativos se hacen necesarios para sobrevivir. Esto requiere un tiempo que nadie tiene. El resultado es una pérdida permanente de especies.

6.3. El reto de la Falsificación de la TE

Argumentemos ahora que la TE Verbal es obviamente falsa y que cada persona es un juez autorizado para dar ese veredicto. La prueba es sencilla: hacemos muchos intentos fallidos antes de tener éxito, si es que lo tenemos. ¿Dónde están los intentos fallidos de la Evolución antes de hacer un gato? Más extendido:

Las operaciones verbales son extremadamente complejas y siempre conllevan imperfecciones. Esto significa que todo el mundo puede expresarse, pero hacerlo de forma correcta y atractiva es algo reservado a muy pocas personas. E incluso éstas necesitan trabajar duro y largo para conseguir algo interesante. Así, la perfección se alcanza como la cumbre de un proceso de Evolución de la Perfección. Del mismo modo, los seres vivos son el epítome de la complejidad y la perfección, por lo que si aparecieron por Evolución, surge una predicción obligatoria: nuestros cuerpos, anatomía y fisiología, así como el registro fósil, deben tener un abundante testimonio de la batalla de la Evolución contra la complejidad en forma de todo tipo de malformaciones y disfunciones. Pero dado que la mayoría de los bebés de todas las especies son absolutamente perfectos, la TE Verbal es obviamente falsa.

Nuestras apreciaciones subjetivas producen **el reto de Falsar la TE**: *Encontrar el marco teórico más completo, más justo, más claro y más discriminate que haga que la Teoría de la Evolución (TE) sea obviamente falsa.*

Hasta aquí, hemos propuesto una solución a este reto: es la TE Verbal. ¿Qué tan buena es esa teoría como marco para falsificar la TE? Necesitamos saberlo.

7. PONIENDO A PRUEBA LA TE VERBAL

En el Método Científico, las teorías se hacen para ser puestas a prueba. El procedimiento consiste simplemente en formular una predicción de la teoría y compararla con las observaciones. Si no hay grandes diferencias, se inventa otra prueba, y así sucesivamente hasta que la teoría se rompe y surge la necesidad de inventar una nueva. Al final, una teoría resiste varias pruebas importantes y se adopta como dogma o, en todo caso, como una teoría de gran éxito y con gran poder predictivo. Todo esto es puro sentido común, nada que ver con naves espaciales, salvo porque las pruebas pueden ser muy ingenuas y no captan la falsedad fragante.

Apliquemos esto a la TE Verbal.

7.1. Una predicción increíble

Es difícil formular predicciones importantes. Aquí tenemos una.

Predicción increíble e inmediata de la TE Verbal:
Si la Evolución es responsable del algoritmo que describe la construcción de nuestro cuerpo, entonces la Evolución debe ser útil para la síntesis de software para uso humano tanto en la Academia como en la Industria. Todo lo que se necesita es un AG cuyo procedimiento de selección esté dirigido a filtrar los programas informáticos con las características requeridas. Por lo tanto, esto es realizable y es crucial. En términos oficiales, la TE Verbal es falsable, y por lo tanto científica.

Esta predicción es increíble porque el desarrollo de software es una de las tareas más difíciles que ha inventado la humanidad en su conjunto. Y es lo que hacemos a toda hora. Es por eso que nuestras vidas son tan terriblemente pesadas. ¿Es correcta la predicción?

Sí, lo es.

La Ciencia de la Computación ha trabajado a fondo este test y el resultado es una disciplina desafiante que se llama **Programación Genética (PG)**, la síntesis de software mediante la Evolución (John Koza et al. [17] 2005, cuyos primeros trabajos se publicaron en 1988). De ella surgen, año tras año, códigos impresionantes que compiten con éxito con los programas creados por el ser humano (Programación Genética [8] 2021). Es una disciplina extremadamente difícil y costosa, pero existe una introducción comprensible para los lenguajes Java y Clojure (Rodríguez [27] 2017).

7.2. Alta calidad y extrema complejidad como test

Los seres biológicos, desde los virus hasta los humanos, son todos asombrosamente complejos e increíblemente perfectos. Esto se convierte naturalmente en un test.

La predicción de la Gloriosa PG:
Si la Evolución es responsable del algoritmo que describe la construcción de nuestros cuerpos, que son tan complejos y tan funcionales, entonces la Evolución

debe ser útil para la síntesis de un software muy complejo y de alta calidad para el uso humano.

El test pregunta por el valor de los programas realizados por la PG. En este sentido, un éxito de la PG que admiro es la programación de un ordenador cuántico para una determinada tarea (Spector et al.[32] 1998). Por eso declaramos con alegría:

Veredicto. La TE resiste la prueba de la Gloriosa PG: la gloria de la Evolución Aplicada en general es la de su imaginación.

La reacción natural ante este impresionante éxito es concluir incondicionalmente que la TE es la explicación correcta de nuestra existencia. Sin embargo, un mentiroso siempre dice muchas verdades. Por eso, pasar un agudo test no significa que una teoría sea correcta. Se necesitan más pruebas y más conocimientos, y se sabe cuándo parar: nunca. Por eso seguimos en la lucha.

7.3. Perfección total y omnipresente como test

Me invitaron a visitar a una madre y a su hija recién nacida. La madre me permitió sostener a la bebé y, con alegría, admiré sus diminutos dedos y uñas y me quedé tan sorprendido con su perfección que exclamé: ¡Es perfecta! Cuando la bebé me escuchó, abrió sus profundos ojos azules, me miró durante un segundo y los volvió a cerrar. A continuación, todos juntos dimos gracias al Señor por aquel milagro.

Dondequiera que dirijo mis ojos en el reino viviente, veo la perfección. Y si miro los fósiles, por ejemplo, de varios tipos de dinosaurios, también parecen perfectos.

Por muy subjetivas que sean mis apreciaciones, han programado mi mente para rechazar la TE de todo corazón. Y la gran gloria de la Ciencia no ha hecho tambalear mis convicciones. ¿Por qué? Porque la Ciencia se equivoca: la perfección que yo veo no viene por mejoras de mejoras, y la complejidad de la vida exige de la TE predecir la Evolución de la Perfección. Pero esto no se encuentra. Para ver por qué, basta con mirar nuestra propia faz:

El diseño de nuestra cara es muy atractivo: 2 ojos, 2 cejas, 2 fosas nasales, una boca. Para explicar esto, la TE verbal argumenta que fue originado por la Evolución: el patrón de nuestra cara es el mismo que en todos los mamíferos, aves, y dinosaurios. Por lo tanto, su origen y fijación debió ocurrir hace mucho tiempo. Pero, ¿dónde están las huellas de esa fijación? Los tiburones tienen tejidos en la parte lateral de su cuerpo que pueden degustar posibles presas: ¿dónde están los ancestros de los mamíferos con tejidos de degustación en los brazos? Muchas arañas tienen 4 pares de ojos (diferente a tener 8 ojos): ¿dónde están nuestros antepasados con 2 pares de ojos, uno en la parte delantera y otro en la nuca? ¿Considera Usted que nuestras preguntas son tontas? En ese caso, piense en la disposición de los distintos elementos de la cara. ¿Apareció por azar y nunca evolucionó? ¿No? ¿Fue fijada por la Evolución que premia la reproducción del más apto? ¿Sí? ¿Dónde están los ancestros con las otras variaciones, por ejemplo, con los ojos en la frente y las fosas nasales en la barbilla? ¿Dónde hay algo en la Naturaleza que pueda convencerme de que el patrón de mi cara fue producto de la Evolución?

En ninguna parte.

Ahora, sólo hay una respuesta posible:

El árbol evolutivo conecta la archea con el ser humano. Así, la Perfección también evolucionó porque la perfección de la archea no es la misma que la de los organismos complejos. Ahora bien, en lo que respecta a una especie determinada, la Evolución de la Perfección efectivamente ocurrió pero no existe ningún registro fósil debido a tres razones: la mayoría de los cambios no dejan registro fósil, efectos de muestreo, y la Evolución silenciosa en el ADN basura o en el material duplicado.

Nuestra decisión es exigir que se desconfíe de ésta y de toda justificación de la ausencia de la Evolución de la Perfección: El número de especies es superior a 7 y no se ha encontrado ningún registro fósil de la Evolución de su Perfección. ¿Es fiable decir que aparecieron por Evolución?

Ahora bien, nos negamos a alimentar nuestra sospecha con más retórica. Ya tenemos suficiente. En su lugar, debemos aportar una investigación experimental. Así pues, recurramos a la PG para ver cómo es la Evolución de la Perfección.

Objeción. Todo el mundo está harto del término Perfección que puede significar cualquier cosa. ¿Puede alguien dar una definición operacional?

3 ***Ejercicio.*** *Resolver la objeción.* *Respuesta*

Pasemos ahora a la PG experimental. El problema es que ha sido un misterio impenetrable para casi todo el mundo.

Ya no más.

En lo que sigue, propondremos un juego para niños que nos permitirá hacer PG para ver la batalla de la Evolución contra la complejidad en lo que respecta al desarrollo de software. Pero, en medio de tantos detalles, es importante tener en cuenta lo que estamos haciendo: estamos incorporando estudios cuantitativos para descartar algunas mentiras. Esto lo hacemos por rutina. Ahora bien, como la información de nuestro ADN es similar a la riqueza, el siguiente ejemplo popular nos ayudará a fijar ideas.

Ejemplo: José solía ganarse la vida normalmente, pero en los últimos días aparecía con coches caros y gente que le servía, algunos vestidos de negro fino. Decía que todo esto era fruto de su trabajo y sabiduría. Pero como de repente despilfarrase mucho dinero, las Autoridades lo acusaron de lavar dinero. Él niega todas las acusaciones y no acepta las pruebas. En la discusión, hacen una predicción: si está limpio, debe tener suficientes documentos escritos así como personas dispuestas a testificar. No se encontraron documentos sobre altas sumas de dinero y por eso llamaron a Pedro y a Santiago. Pedro dijo: He transportado mercancías para él desde la fábrica hasta sus almacenes durante 20 años. Santiago también dijo lo mismo. Después de algunos cálculos, las autoridades respondieron: ¿Dónde están los otros 1.200 empleados que trabajaron durante 20 años y que eran necesarios para asegurar unas ganancias netas tan grandes como las que usted reporta? José fue incapaz de decir algo coherente y por eso fue condenado. Ya lleva 9 años de cárcel y pronto terminará su servicio.

Para empezar con nuestra tarea de contabilidad, presentaremos un computador muy sencillo junto con un lenguaje para manejarlo. Sus hijos lo encontrarán muy divertido. Y, será suficiente para nuestras pruebas de PG en las que enfrentamos la Evolución contra la complejidad.

8. EL ORDENADOR FLIPP

El programa `FlippComputer.html` nos muestra un `Computador Flipp` con varios periféricos:

- Un `computador Flipp` tiene una celda de memoria que puede guardar un bit, un 0 o un 1. Para empezar o después de limpiar, la celda contiene el valor 0.
- `Flipp` es un lenguaje de programación para un computador Flip. Flipp es una abreviatura de *flip* (dar el bote, conmutar) y *print* (imprimir).
- Un ordenador Flip es una máquina Flip que está programada con Flipp.
- El valor de la celda de memoria se modifica por los comandos de Flipp, que son sólo dos. Utilice el `punto .` para imprimir el contenido de la celda al área de texto OUTPUT. Utilice el `signo de exclamación !` para conmutar e imprimir: Al conmutar se convierte el 0 en 1 y el 1 en 0.
- Los comandos escritos forman un programa que aparece en el área de texto PROGRAM.
- Use este ordenador para diseñar programas para imprimir textos binarios específicos. Esta es la tarea de todos los programas en todos los computadores.
- Entrénese viendo lo que hacen los comandos. Cuando esté preparado, haga una prueba: Si marca la casilla CHALLENGE, aparecerá una cadena binaria que lo retará: Su misión, si la acepta, es desarrollar el programa Flipp que reproduzca la cadena dada. Para terminar su programación, haga clic en el botón EXECUTE PROGRAM. Debe superar 7 niveles.
- ¿Considera que programar en Flipp es fácil? ¿Recomendaría una métrica para apoyar su respuesta?.

Ejemplo. Algunas contraseñas y sus programas generadores en Flipp:

```
Contraseña = 0
Prog Flipp = .

Contraseña = 1
Prog Flipp = !

Contraseña = 0000
Prog Flipp = ....

Contraseña = 11111
Prog Flipp = !....

Contraseña = 000110001001
Prog Flipp = ...!.!.!!!

Contraseña = 1000110001001
Prog Flipp = !!!!!!!!!!!!
```

4 ***Ejercicio.*** *Demuestre con un diagrama que Flipp es una implementación más del esquema general de comunicación verbal.* *Respuesta*

Nota. Sólo hay un programa Flipp que puede dar salida a una cadena dada. Tal programa puede ser calculado automáticamente como muestra el programa `FlippMaker.html`. El `Operador de signo de exclamación` puede ser llamado `Operador de salto` y el `Punto` el `Operador de identidad`. El programa Flipp puede entenderse como una derivada simbólica discreta de la cadena planteada.

9. EL COSTO DE LA PERFECCIÓN

Cuando un desarrollador es contactado, el cliente dice: `quiero un programa que en estas circunstancias produzca esto, y en aquellas otras, aquello`. La memoria escrita del contrato se consigna en un documento que se llama `Espec`. Nuestra preocupación aquí también trata de un problema `Espec`, de especificaciones, y del costo de encontrar una solución exacta.

9.1. ESPEC como una abstracción

¿Puede la Evolución hacer frente a la complejidad del diseño de los seres vivos? En el marco de la Genética, este problema se lee: ¿Puede la Evolución especificar el ADN de nuestras células? En el marco de la TE Verbal de Flipp se abstrae aún más de la siguiente manera:

ESPEC, el problema binario de Espec en PG sobre Flipp:
¿Puede la Evolución hacer coincidir la salida de un programa en Flipp con una cadena binaria aleatoria dada?

5 ***Ejercicio.*** *¿No sería suficiente limitarnos a cotejar cadenass binarias con cadenas binarias sin mencionar ningún programa?* *Respuesta*

Vemos que no se hace referencia a ninguna condición en la formulación de ESPEC por lo que nuestro problema es realmente sencillo, posiblemente el más simple. ¿Puede la PG sobre Flipp resolverlo? Por supuesto que sí. Pero, ¿a qué precio? Esta pregunta es importante para nosotros porque, en lo que respecta a las combinatorias, los recursos informáticos del Universo en su conjunto son fantásticamente bajos.

6 ***Ejercicio.*** *Demostrar o refutar: Flipp es el lenguaje más sencillo que produce cadenas binarias al menor coste. Por lo tanto, las estimaciones que encontraremos son las más baratas.* *Respuesta*

9.2. El costo de ESPEC

Para calcular el coste de ESPEC, adoptamos la metodología utilizada en Matemáticas e Informática para el cálculo de la complejidad algorítmica de una tarea (Arora y Barak, [2] 2007). La diferencia es que no utilizamos cálculos matemáticos, sino simulaciones. Un **desafío para todos** *es utilizar nuestras estimaciones para construir modelos matemáticos robustos.*

Esto es lo que haremos:

Damos una cadena binaria de longitud x, la emparejamos con PG sobre Flipp, medimos el coste en individuos como una función exponencial en x, y planteamos una nueva cadena aleatoria de longitud $x+1$. En nuestros experimentos, comenzamos con la longitud 1 y seguimos hasta que veamos una tendencia estable en la expresión exponencial. El coste crece rápidamente a medida que aumenta x, por lo que no podemos llegar a un número grande. En la mayoría de los casos, el número 30 es ya insoportable, pero las expresiones exponenciales se estabilizan en torno a 15.

¿Qué estamos haciendo? Estamos enfrentando a la Evolución contra la complejidad porque a medida que aumenta la longitud x de una cadena, también lo hace su contenido informativo, y también la dificultad para emparejarla.

El programa `PGwithFlipp.html` nos permite estimar el coste de emparejar una cadena binaria de longitud x como $49,7e^{0,3x}$. Esta estimación es el resultado de resolver ESPEC en el intervalo $[1,30)$, donde no se incluyó 30. Para las extrapolaciones, el exponente es lo que importa, a menos que las expresiones comparadas tengan el mismo exponente. El coste extrapolado de la Evolución es menor que el de la aleatoriedad, que a su vez es menor que el del peor caso de búsqueda.

Nuestros resultados nos permiten afirmar: El coste de ESPEC es exponencial en la longitud de la entrada, donde la longitud de la entrada es x.

Objeción: El procedimiento para extrapolar el coste como una función exponencial se obtuvo gracias a un número finito de puntos. Pero el ajuste natural en tal caso es un polinomio, por lo que una dependencia exponencial aquí no tiene lugar.

7 ***Ejercicio.*** *Resuelva la objección.* *Respuesta*

8 ***Ejercicio.*** *Comparar el coste de ESPEC en PG sobre Flipp con el de GAs.* *Respuesta*

Reto. Para un ser humano, es inmediato diseñar un programa Flipp cuya salida coincida con una cadena binaria dada. Sin embargo, para la Evolución, tiene un coste exponencial en la longitud de la entrada. Este desequilibrio dice que la Evolución no debe ejecutarse en el espacio de los programas, sino en el espacio de los sistemas de IA. Implemente ésto.

Reto. En el Volumen XVIII de *Java para el estudio de la evolución*, un programa mostró que uno puede arreglar las cosas para aumentar en uno el número de sitios emparejados por generación. Esto parece contradecir nuestros resultados actuales. Exprese claramente la contradicción en términos cuantitativos y resuélvala.

9.3. El surgimiento evolutivo de los genes

El coste exponencial de ESPEC se convierte fácilmente en una falsación de la TE Verbal:

Afirmación. Según nuestras simulaciones, la Evolución necesita un número exponencial de individuos para producir un programa Flipp cuya salida coincida con una

cadena binaria dada. Por lo tanto, la Evolución no puede ser la causa de nuestra existencia debido a su falta definitiva de recursos. De hecho, el coste exponencial de la ESPEC significa que no existe un universo con una velocidad finita de transferencia de información en el que la Evolución pudiera especificar nuestro ADN. Por ejemplo, el coste inocente $49,7e^{0,3x}$ evaluado para $x = 50$ da 162 millones y para $x = 500$ más de 10^{68}. Esto implica que el potencial informático de toda la humanidad es nula en comparación con lo que se necesita para simular la PG a largo plazo. En realidad, nuestros problemas suelen empezar alrededor de $x = 25$. Sin embargo, el programa `PGwithFlipp.html` también muestra que la Evolución es mejor que la aleatoriedad, que es mucho mejor que el peor caso de búsqueda. Por lo tanto, la Evolución puede ser útil para la PG más que aquellos y mirando a los problemas adecuados, la PG seguiría sorprendiéndonos. En general, predicamos: Su imaginación es el único límite al poder de la Evolución Aplicada.

9 ***Ejercicio.*** *Encuentre una salida evolucionista a la falsación dicha de la TE Verbal.* *Respuesta*

Tenemos la sospecha de que el coste exponencial, más que ser un rechazo de la TE Verbal, es uno más de sus matices que la hace interesante. Llegamos a la conclusión de que debemos explorar otras ideas. Presentemos una muy natural: el precio exponencial de la Perfeccción no es problema, precisamente porque la Evolución saca el mejor partido de la Imperfección.

10. EL COSTO DE LA IMPERFECCIÓN

Por nuestra experiencia con los AGs, sabemos que Evolución también debe pagar un coste exponencial para encontrar soluciones aproximadas. ¿Cuál es el caso aquí?

10.1. Nuestras simulaciones

El programa `FlippCostOfApprox.html` muestra que el coste de la imperfección se comporta como se muestra en la Tabla 1.

Decimos simplemente: aproximar con la salida de programas Flipp el 70 % de una cadena tiene un coste exponencial en la longitud de la entrada con exponente 0,06.

En todos los casos, el coste extrapolado de la Evolución fue menor que el de la aleatoriedad, que a su vez fue menor que el del peor caso de búsqueda.

Si se repite la simulación, se observa una tendencia a la estabilidad, pero a valores ligeramente desplazados. Esto podría significar finalmente que para lograr la estabilidad y la convergencia al límite único se necesitan rangos más amplios. La opción de las fluctuaciones que nunca mueren es correcta, pero su magnitud disminuye al aumentar el rango. Por eso se espera que haya estabilidad.

Conclusión general. El coste de resolver ESPEC es exponencial en la longitud de la entrada, tanto para la perfección como para la imperfección. El grado de aproximación se refleja naturalmente en el exponente.

10 ***Ejercicio.*** *Proponga algunas implicaciones de nuestra conclusión con respecto a la Evolución Natural.* *Respuesta*

Cuadro 1: Costo exponential de las aproximaciones.

Grado	Evolución	Azar	Peor caso	Rango	Duración
0.70	$30{,}7e^{0{,}06x}$	$2{,}3e^{0{,}09x}$	$1{,}4e^{0{,}49x}$	[1-150)	7 min
0.80	$44{,}7e^{0{,}1x}$	$2{,}3e^{0{,}21x}$	$1{,}3e^{0{,}55x}$	[1-75)	12min
0.90	$28e^{0{,}22x}$	$1{,}8e^{0{,}41x}$	$1{,}3e^{0{,}63x}$	[1-50)	13 horas
0.95	$81{,}1e^{0{,}22x}$	$3{,}8e^{0{,}5x}$	$1{,}2e^{0{,}66x}$	[1-30]	1 min
0.95	$58{,}8e^{0{,}25x}$	$3{,}3e^{0{,}51x}$	$1{,}2e^{0{,}67x}$	[1-37]	31 horas
0.95	$47{,}5e^{0{,}27x}$	$2{,}9e^{0{,}51x}$	$1{,}2e^{0{,}67x}$	[1-38]	48 horas
0.95	$41{,}8e^{0{,}28x}$	$2{,}9e^{0{,}52x}$	$1{,}2e^{0{,}67x}$	[1-40)	54 horas
1.00	$49{,}7e^{0{,}3x}$	$0{,}2e^{0{,}74x}$	$1e^{0{,}69x}$	[1, 30)	20min
1.00	$47{,}5e^{0{,}28x}$	$0{,}3e^{0{,}74x}$	$1e^{0{,}69x}$	[1, 30)	55min
1.00	$35e^{0{,}3x}$	$0{,}6e^{0{,}71x}$	$1e^{0{,}69x}$	[1, 30)	30min

El programa `FlippCostOfApprox.html` *predice la dependencia exponencial del coste de aproximación de una cadena binaria aleatoria de longitud x según la PG sobre Flipp, la aleatoriedad y la búsqueda del peor caso. Esta tabla debe ser verbalizada como sigue: Para el grado de aproximación* $0{,}70$*, el coste medio esperado de encontrar por Evolución el primer programa Flipp que coincida con una cadena de longitud x es* $30{,}7e^{0{,}06x}$ *que se estableció en el rango [1-150). La última columna dice el tiempo necesario en un computador de escritorio que ejecuta exclusivamente el programa. La repetición hace que tanto el coeficiente como el exponente varíen como lo muestran los tres últimos renglones (obtenidos con el programa* `PGwithFlipp.html`*). Pero, las variaciones son menores cuanto más amplio es el rango de longitudes de las contraseñas investigadas.*

11 ***Ejercicio.*** *Nuestras simulaciones dicen que el coste tanto de la perfección como de la imperfección en PG sobre Flipp es exponencial en la longitud de la entrada. ¿Podría añadir el cálculo de los intervalos de confianza para hacer esa afirmación más seria?* *Respuesta*

Hasta ahora, hemos avanzado mucho. ¿Algún problema?

11. EL ROMPECABEZAS DE LA TELEONOMÍA

Lo que sigue es una terrible crítica a nuestro trabajo hasta aquí:

Objeción General. Todos nuestros tests de PG son más para el Lamarckismo que para el Darwinismo.

De hecho, Lamarck pensaba que la materia se reviste de una fuerza alquímica progresiva para la Evolución, mientras que Darwin proponía que el progreso es un subproducto de la lucha por la vida. Ahora bien, en nuestras simulaciones seguimos a Mendel, De Vries y Lamarck:

Seguimos a Mendel porque utilizamos elementos heredables discretos. Seguimos a de Vries porque nuestra mutación fue al azar. Seguimos a Lamarck porque todos nuestros programas tienen un objetivo que se impone desde el exterior y se hace cumplir

en cada paso de la reproducción. Así, comparamos permanentemente las cadenas evolucionadas con la declarada como objetivo. Por lo tanto, nuestra PG es, en efecto, una PG Dirigida porque siempre supone un objetivo.

Por el contrario, según la TE Verbal, que es una extensión materialista de la TE, la Evolución Natural no tiene ninguna meta. La fuerza de esta doctrina reside en la gran idea de Darwin de que la lucha por la vida en medio de hermanos señala al más apto que deja más descendencia. Este proceso conduce a la perfección cuando se itera generación tras generación.

Reto. Disfrute de la estupenda simulación de la Evolución que ha desarrollado Ingo Berg ([12] 2019) y que se ejecuta automáticamente en línea. Decida si sigue a Mendel, de Vries, Lamarck o Darwin. El código se puede descargar para un examen detallado. Podría preguntarse, por ejemplo: ¿hasta qué punto el procedimiento de selección está definido a mano y no por la autoorganización, por una propiedad emergente de la relación ambiente-organismo?

La situación en la que se persigue activamente un objetivo predefinido desde el exterior se denominará **teleonomía** (compárese Henning [10] 2011). Por lo tanto, tenemos una pregunta: ¿Cómo vamos a simular la Evolución sin teleonomía?

Creemos que la cuestión de la teleonomía es un enigma que nunca será suficientemente esclarecido por las simulaciones debido a la eterna falta de potencia de computación. Por lo tanto, debemos recurrir a un razonamiento indirecto. Nuestro primer punto es una invitación a mirar en retrospectiva:

El problema de la supervivencia es el único problema importante en el Darwinismo, el cual tiene posiblemente miríadas de soluciones correctas. Sin embargo, podemos observar que, en general, el éxito en la lucha por la vida se puede deconstruir, por ejemplo, de la siguiente manera: buena movilidad, alta inteligencia para entender el entorno, buen aprovechamiento de los recursos, comportamiento victorioso con los hermanos, excelente acercamiento con los compañeros. Todo esto exige una información constructiva adecuada. Lo que queremos decir es que esta información debe estar codificada en el ADN, ya sea de forma directa o de algún modo indirecto. Y cuando lo está, no es más que una cadena tan congelada como el hielo en los ojos de una persona no comprendida, que de todos modos conserva alguna porción de vida o variación. Esto parece implicar que el coste de igualar una cadena es el mismo en ambos casos, con o sin teleonomía. ¿Es realmente así?

12 ***Ejercicio.*** *Sabemos que un grado de aproximación en la PG Dirigida tiene un coste exponencial con exponentes consecuentemente crecientes. Demostrar que el coste correspondiente en la Evolución Auto-Organizada, sin teleonomía, no puede ser menor.* *Respuesta*

Objeción. La recombinación con experimentos extraños podría producir soluciones más rápidamente que la recombinación con más de lo mismo. Por lo tanto, el coste en la autoorganización podría ser menor que en la PG dirigida.

13 ***Ejercicio.*** *Resuelva la objeción anterior.* *Respuesta*

Hemos razonado guiados por el sentido común que parece enraizado en el gradualismo Darwiniano cuando el tamaño de la población es bastante grande. ¿Nos faltará algo más?

12. BUGS

Hace unos 15 años llegué a la conclusión de que para falsar la TE se necesita el Método Científico. Y, que además, las simulaciones eran la forma más fácil de calcular las predicciones. En ese momento Java se convirtió en código abierto, y entonces, decidí aprender Java para programar simulaciones de la Evolución. Así, después de unos 10 años sin programar, empecé a trabajar y me asombré de la inmensa cantidad de errores que cometía. Al mismo tiempo, también me di cuenta de que el Genoma es software. En el oficio, a los errores de programación se les llama **bugs (bichos)**. Por eso me pregunté: **Si la Evolución me creó, ¿dónde están sus bugs?** Por una razón desconocida y desde mis veinte años, suelo mirarme la mano izquierda cuando me pregunto cosas así. La abro y la cierro y percibo lo contento que me siento con ella. Y de mi corazón surge una llama fulgurante que quema cualquier deseo de creer que estoy aquí por la Evolución. Y florece en mi mente y en mi alma el deseo de glorificar a Dios mi Creador.

Soy consciente de que Dios Creador no es una cuestión de Ciencia, es decir, de experimentos con el Método Científico, por lo que me clasifico como irracional. Pero, ¿qué pasa con la TE? El Método Científico exige calcular predicciones. Para ello implementé simulaciones en Java. Sin embargo, me di cuenta de que ejecutar programas en Java era imposible para la mayoría de la gente. Entonces, aprendí a programar en JavaScript para que todo el mundo pudiera ejecutar programas con un solo clic si uno los descomprime. De hecho, los programas en JavaScript pueden ser abiertos con cualquier navegador web, y corren inmediatamente: sobre un nuevo tab en su navegador, presione `Ctrl + O`, haga click sobre el nombre del programa, y ábralo. Si no funciona, revise la configuración del navegador para activar la posibilidad de correr programas en JS.

Nuestra tarea consiste ahora en transformar una experiencia subjetiva de rechazo de la TE a causa de los bugs en algo objetivo y transparente que pueda ser simulado con fidelidad. Para empezar, el concepto de bug está irremediablemente ligado a la teleonomía porque los bugs son errores que se cometen contra un objetivo. En segundo lugar, preferimos preguntas operativas:

¿Por qué son tan frecuentes los bugs?

Eso depende tanto del desarrollador como de la naturaleza del Software. Nos olvidamos del desarrollador porque buscamos facilidad para detectar objetividad. En lo que respecta al Software, la causa es que es discontinuo, es decir, que no obedece al gradualismo, a sumar pequeños progresos. De hecho, el Software por ser verbal no puede ser gradualista. Consideremos, por ejemplo, la siguiente expresión que es válida en la mayoría de lenguajes de alto nivel:

$$s = g(t)$$

Dado que t puede ser cualquier cosa procesable por la función g, se puede variar t para obtener resultados posiblemente muy diversos y disímiles. Por lo tanto, el gradualismo no es la regla en la PG. ¿Cómo es la situación con Flipp? Para saberlo, estudiemos las salidas de las caminatas aleatorias en el espacio de los programas. Las caminatas aleatorias son el epítome mismo de la acción sin objetivo alguno.

12.1. Caminatas aleatorias en PG sobre Flipp

Una **caminata aleatoria en PG** comienza con la generación de un programa al azar, que se muta recurrentemente para estudiar cómo cambia la salida después de cada mutación al azar. Ahora bien, ¿qué esperamos en un caminata aleatoria?

Flipp admite mutaciones que producen cambios suaves.

Ejemplo. El programa ..... imprime 000000000 que se interpreta como el número 0 y el mutante .....! produce 00000001 que en interpretación decimal es 1. El mutante es lo más parecido posible a su padre.

Sin embargo, cambios bruscos también son posibles:

Ejemplo. El programa ..... imprime 000000000 que se interpreta como el número 0 y el mutante !..... produce 111111111. Si sumamos 1 a este número, obtenemos 100000000 que en notación decimal es 2^9. Así que el mutante es $2^9 - 1$. Vemos que una mutación es suficiente para provocar el mayor cambio posible. Así pues, Flipp es discontinuo por su propia naturaleza.

¿Qué ocurre en general en las caminatas aleatorias ?

El programa `FlippPGRndmWalk.html` presenta un estudio sistemático de los cambios a lo largo de las caminatas aleatorias. Concluimos que los saltos de todos los tamaños son la dinámica normal en las caminatas aleatorias de los programas Flipp.

Ahora, ¿qué efectos tiene la discontinuidad de Flipp sobre la PG ordinaria?

12.2. La dinámica en PG sobre Flipp

El programa `FlippDynamics.html` nos muestra cómo la PG trata a lo largo del tiempo el problema de cazar una contraseña que representa un tablero de ajedrez. Una casilla en gris se representa por 0, y una en negro por 1. Así, un tablero se representa por una cadena binaria. El programa traza la dinámica a través del reporte del mejor puntaje de cada generación. Aquí no vemos agujas porque no prestamos atención a todos los tipos de saltos sino sólo a los del mejor puntaje de una generación a otra. El programa nos dice que la dinámica en la PG es muy similar a la de los AGs: tras un valor inicial que se produce por azar y que se clasifica claramente como mediocre, la mejor solución asciende un poco formando un hombro que se nivela en un valor de equilibrio. Observemos que ese valor es del puntaje, pero las cadenas correspondientes podrían tener una dinámica fluida. La razón es que muchas cadenas pueden tener el mismo grado de coincidencia del objetivo elegido. Las fluctuaciones existen y son más amplias que en los AGs. Estas son mayores para una complejidad baja, pero disminuyen a medida que aumenta la longitud de las contraseñas.

Ahora tenemos una pregunta:

¿Es correcta la extrapolación del costo exponencial? Esa extrapolación supone que la PG puede llegar efectivamente a la Perfección. Pero aquí vemos que la Evolución crea un valor de equilibrio bastante cercano a la mediocridad y que las fluctuaciones que se apartan de él son suficientes para resolver los problemas de pequeña complejidad, pero cuando ésta aumenta, el tamaño de las fluctuaciones disminuye. Así que, desde esta perspectiva, no está nada claro que la perfección sea alcanzable en PG sobre Flipp.

Nuestra interpretación es que las fluctuaciones nunca mueren y que éstas no tienen ningún misterio, ni para crear milagros ni para impedir su aparición. Más bien, obedecen a las leyes estadísticas y aquí diríamos que entra en acción la ley de los grandes números. Así pues, creemos que la Evolución sí puede alcanzar la Perfección aunque su costo sea exponencial en la longitud de la entrada, como fue extrapolado por un programa anterior.

Observemos que el valor de equilibrio es estable: si una fluctuación sube, lo más probable es que descienda. Y si la fluctuación baja, la Evolución hará subir el puntaje hasta nivelarlo. Por eso identificamos el estado estable con el efecto de un proceso determinista, que recibe el nombre de Selección. Por lo tanto, inferimos que la Selección es en general una fuerza antievolutiva, mientras que las fluctuaciones pueden ser tanto reaccionarias como progresivas. Sin embargo, siga leyendo.

Reto. Hemos dicho que el valor de equilibrio es estable porque observamos que las fluctuaciones bailan alrededor de él. ¿Podría hacer una prueba directa de esto programando condiciones iniciales adecuadas y siguiendo su Evolución posterior?

Reto. ¿Cuál es finalmente la causa del estancamiento? Sugerimos que las mutaciones con efectos leves son responsables del componente determinista de la PG, que es estancada y frustrante en términos de perfección. Demuestre o refute esto.

Reto. ¿Cuál es la causa de alcanzar el éxito en PG sobre Flipp? Demostrar o refutar: La PG alcanza el éxito gracias a las fluctuaciones, pero no por sí solas, sino que las fluctuaciones y la selección trabajan juntas de alguna manera para acelerar la Evolución hacia el éxito. Pensamos esto porque la PG es más rápida que el azar. Por eso, la PG tiene un lugar seguro en la Ciencia y en la Ingeniería.

14 ***Ejercicio.*** *Cuando la PG lucha por cazar una contraseña, encuentra coincidencias parciales, primero de longitud 1, luego de longitud 2, y así sucesivamente hasta que por fin, hace diana. ¿Se trata de una tendencia constructiva en la que la PG utiliza los resultados obtenidos? ¿O es una tendencia destructiva en la que la PG no puede utilizar los resultados obtenidos? Si se grafica el coste de alcanzar la coincidencia parcial de longitud k, ¿cuál sería la expresión funcional? Haga una conjetura y pruébala.* *Respuesta*

12.3. El Tapiz de la Perfección

La discusión sobre el papel de las fluctuaciones y de la fuerza de estancamiento de la Selección alrededor del estado de equilibrio es para nosotros fascinante. Con la espe-

ranza de entender un poco mejor esa interacción, el programa `FlippEvoOfPerfection.html` representa la Evolución de la Perfección en PG sobre Flipp en la lucha contra ESPEC.

El programa presenta 3 gráficos que nos muestran la expresión funcional del costo de alcanzar la coincidencia parcial de longitud k cuando la PG resuelve una contraseña dada. En segundo lugar, nos muestra un tapiz de pequeños cuadrados coloreados. El color representa la aptitud o puntaje relativo del programa Flipp correspondiente. Cada renglón representa una generación, que está numerada. La aptitud aumenta desde el verde oscuro (clase cero) al verde claro (clase ocho), luego al violeta, que representa la cuasi-perfección, y después al rojo, que es la perfección. El Genoma perfecto está unido por dos segmentos a sus progenitores, uno justo debajo de él y el otro en algún lugar del renglón inferior. Cada columna denota la descendencia directa.

Observamos que casi todas las generaciones tienen cuadros de color verde claro, lo que significa que para la PG es bastante fácil construir una buena calidad. La cuasi-Perfección, de color violeta, se ve raramente. Entonces, ¿cómo se fabrica la Perfección? El cuadrado rojo casi nunca procede de los violeta, sino que suele proceder de progenitores que producen soluciones más bien incipientes.

Concluimos que el tapiz nos muestra que la Evolución hace que los programas Flipp produzcan salidas cuya aptitud cambia preferentemente por saltos.

Objeción. Los resultados del programa `FlippEvoOfPerfection.html` se han utilizado para afirmar que la discontinuidad de PG sobre Flipp se debe a la naturaleza discontinua del Software. Pero esto es falso: lo que es discontinuo es la recombinación. Precisamente, la recombinación es la causa de que los AGs, que no contienen software, puedan escapar de los máximos locales atascadores.

15 ***Ejercicio.*** *Resuelva la objeción.* *Respuesta*

Nuestra conclusión es que en la PG Dirigida sobre Flipp, la Perfección es precedida por largas rachas de sub-perfección. Extendemos esta conclusión a la Evolución Natural debido al surgimiento de la Selección Auto-Organizada, y así obtenemos otra falsación de la TE Verbal: Antes de la perfección, la imperfección debe encontrarse abundantemente en el registro fósil. Como esto es falso, la TE Verbal ha sido falsada.

16 ***Ejercicio.*** *Demostrar que la TE Verbal predice que los efectos de los bugs deben ser abundantemente visibles en las poblaciones actuales. Compare la predicción con lo que observa en su propio cuerpo. Decida si la TE Verbal ha sido falsada.* *Respuesta*

13. CONTEXTUALIZACIÓN

¿Cómo se relaciona nuestra falsación de la TE con nuestro mundo y nuestra vida? Veamos:

13.1. Relación con el creacionismo

Nuestro trabajo es una falsación científica de la TE. De ello no se puede deducir que Dios nos haya creado. Sin embargo, si alguien tiene fe en que Dios nos creó y que Él juzgará cada pensamiento oculto que tengamos, entonces la Ciencia no puede usar la Evolución para rechazar esa creencia.

13.2. Relación con la Ciencia

La Filosofia Materialista forza a la Ciencia a afirmar en contra de los Cristianos: `No se puede detectar su alma, por tanto no existe, y todas sus creencias son erróneas.` No se defienda, más bien disfrute su fe, y trate a las personas de menor rango como iguales a Usted porque ellas también son la imagen de Dios. Y no se le olvide de que sus enemigos tienen una lucha permanente contra la depresión.

A nivel operacional, tenemos un cambio de paradigma:
"Nada en la Biología tiene sentido si no es a la luz de la Evolución". Esto es lo que todo el mundo creía hasta ahora. Todo se explicaba en la Ciencia por la Evolución, y para cada caso de especiación, sólo había que inventar un mecanismo aceptable.
Pero aquí tenemos una falsación de la TE. ¿Qué será lo que sigue?
La falsación de la TE es similar a la explosión inducida de un meteorito porque para salvar nuestras vidas debemos cuidarnos después de sus fragmentos. De la misma manera, el enfoque de las hipótesis evolutivas debe cambiar. De ahora en adelante, el dicho que iluminará nuestro trabajo será:

Nada tiene sentido en Biología si no es a la luz de la Complejidad. Así, ninguna relación entre especies puede ser explicada por la Evolución si no se puede demostrar mediante el Método Científico, cuyo primer requisito es la replicabilidad. No hay excusa para la Ciencia porque el número de especies de microorganismos oscila en el orden de los millones o, ¿quizás trillones? (Kenneth et al. [15] 2016).

Podemos poner todo en un eslogan: **La TE ha muerto, ¡Viva la TE!** Dado que la TE está muerta, el proyecto de la Evolución como una cosmogonía se desvanece. Esto significa que cada teoría cosmológica que utiliza la Evolución como bloque de construcción fundamental también está muerta. Por lo tanto, se le invita a inyectar nueva vida a la Cosmología inventando un nuevo comienzo.

Nuestras exigencias sobre la especiación son extremadamente difíciles y muy costosas en todos los sentidos. Posiblemente toda la humanidad deba aunar esfuerzos para hacer un solo punto más allá de los fáciles, digamos, el origen de algunas especies de moscas.

17 ***Ejercicio.*** *Proponga realizaciones concretas del Método Científico que pongan a prueba la hipótesis de que dos especies dadas están relacionadas por descendencia.* *Respuesta*

13.3. Relación con las personas

Nos gusta tener dogmas. Pero una persona podría decidir cuáles son sus dogmas y cuáles creencias son cuestionables. Precisamente, eso lo hace humano porque esto siempre se hace por motivos irracionales, digamos que es parte de la propia religión, cultura o herencia familiar. Aquí hemos exhibido nuestra irracionalidad al hacer de la TE una teoría cuestionable que ha demostrado ser falsa. Sin embargo, recordemos que la Evolución está en todas partes y que nuestra imaginación es la barrera más alta para una aplicación poderosa de la Evolución.

Cada persona tiene ahora una pregunta: si la Evolución no es la causa de mi existencia, ¿qué es y qué quiere de mí? Toda respuesta a esta pregunta es irracional: ¡Bienvenido al mundo de los seres humanos!

13.4. ¡Jesus viene!

Este trabajo obedece la orden dada por el Señor por medio del Profeta Isaías: "Preparad el camino del Señor, enderezad sus sendas."

14. CONCLUSIÓN

Hemos especificado completamente la `Teoría Evolutiva (TE)` formulando la `TE Verbal`, que es mecanicista, moderna, definitiva, transparente, sencilla y clara. Primero: `El genoma es software`. Es un hecho de la Naturaleza. Segundo: *La Evolución es un desarrollador de software*. Es un hecho de la Naturaleza y de la ingeniería como Programación Genética (PG), que es el desarrollo de software por medio de la Evolución. En tercer lugar, la `TE Verbal` dice: *La Evolución es el único desarrollador de software que es responsable de la creación de todas las especies en la Tierra.* Para poner a prueba esa teoría con el **Método Científico**, hemos construido el banco de pruebas más sencillo: `PG sobre Flipp`, donde `Flipp` es un lenguaje de programación especialmente adecuado para imprimir cadenas binarias. Resulta que la `Evolución` es vencida por su enemigo natural, la `Complejidad`. Es decir, la `TE Verbal` predice obligatoriamente que debe existir una **Evolución de la Perfección y que la sub-perfección debe ser inmanente a la Evolución**. Operacionalmente, esto significa que las malformaciones deben ser abundantes tanto en el registro fósil como en nuestros propios cuerpos, y las disfunciones deben llenar nuestra propia fisiología. No se encuentra tal cosa en nuestra especie. Por lo tanto, la `TE` es falsa, ya no es un dogma, pero vive como investigación científica: `La TE está muerta, ¡viva la TE!`

RESPUESTAS

Ejercicio 1, page 9. El Código Genético se ajusta a la estructura general de la comunicación verbal, tal y como se muestra en la Figura 2.

Ejercicio 3, page 16. Me parece que la perfección es algo etéreo, es decir, una mezcla de arte y diseño, de complejidad y función. Así que trabajaremos con una simplificación operativa: la perfección es la coincidencia exacta de un conjunto de instrucciones y restricciones. Como todo texto puede codificarse siempre en formato binario, terminamos con una definición de la perfección como la coincidencia exacta de una cadena binaria. En PG, la cadena debe ser el producto de un programa. Esto aparecerá explícitamente en nuestros códigos, donde la cadena a comparar puede ser cualquiera.

Ejercicio 4, page 18. El diagrama de la figura 3 muestra que Flipp permite hacer programas que se ajustan al esquema general de la comunicación verbal.

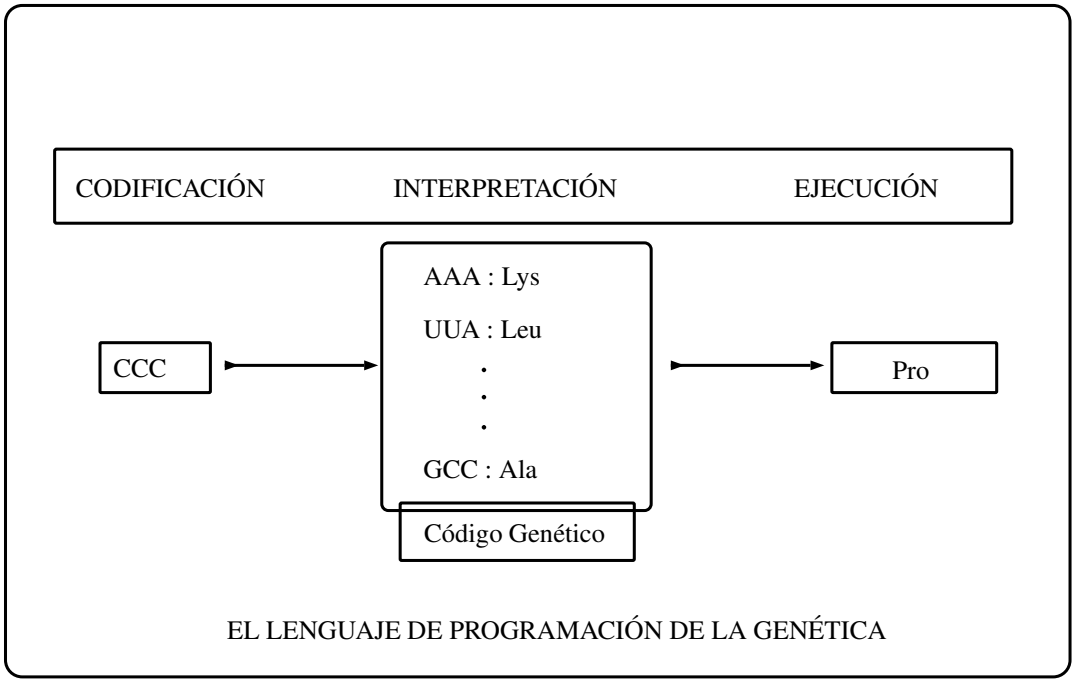


Figura 2: *El ADN contiene codones que son tripletas de símbolos. Cada tripleta es una instrucción verbal que se interpreta según el Código Genético, y la instrucción es ejecutada por un ribosoma, en el citoplasma, que añade el aminoácido correspondiente a una cadena polipeptídica.*

Ejercicio 5, page 18. El ADN no se selecciona por sí mismo, sino por sus frutos, es decir, por el fenotipo. Como primera y posible aproximación burda, el fenotipo de un gen estructural podría ser la enzima o el polipéptido que codifica. En nuestra abstracción, el fenotipo de un programa Flipp es una cadena binaria. Por lo tanto, el puntaje de un programa Flipp se define sobre la cadena que produce.

Ejercicio 6, page 18. Flipp no es el lenguaje más simple que puede producir cadenas binarias de cualquier contenido. El más simple es el lenguaje que toma la propia cadena binaria y la interpreta inmediatamente como un programa. Así, supongamos que la cadena es 0101. Ese lenguaje, el `Lenguaje de la Identidad`, diría: el primer 0 es un comando que significa `Imprimir un cero`. El siguiente 1 significa `Imprimir un 1`. El segundo 0 significa `Añadir un cero`. El siguiente 1 significa `Imprimir un 1`. Y así sucesivamente. Sin embargo, ambos lenguajes, el de la Identidad y Flipp, pagan el mismo costo sobre una misma cadena, el cual coincide con la longitud de la misma. Así, para la cadena 0101, ambos necesitan la ejecución de 4 instrucciones.

Ejercicio 7, page 19. Calculamos en nuestras simulaciones la regresión exponencial para todos los valores de la longitud de las palabras a adivinar, uno tras otro en el rango considerado, y observamos la tendencia: si tras las fluctuaciones iniciales la dependencia exponencial tiende a estabilizarse, la afirmación exponencial es robusta frente a las pretensiones polinómicas. En caso contrario, es débil. Este criterio se debe

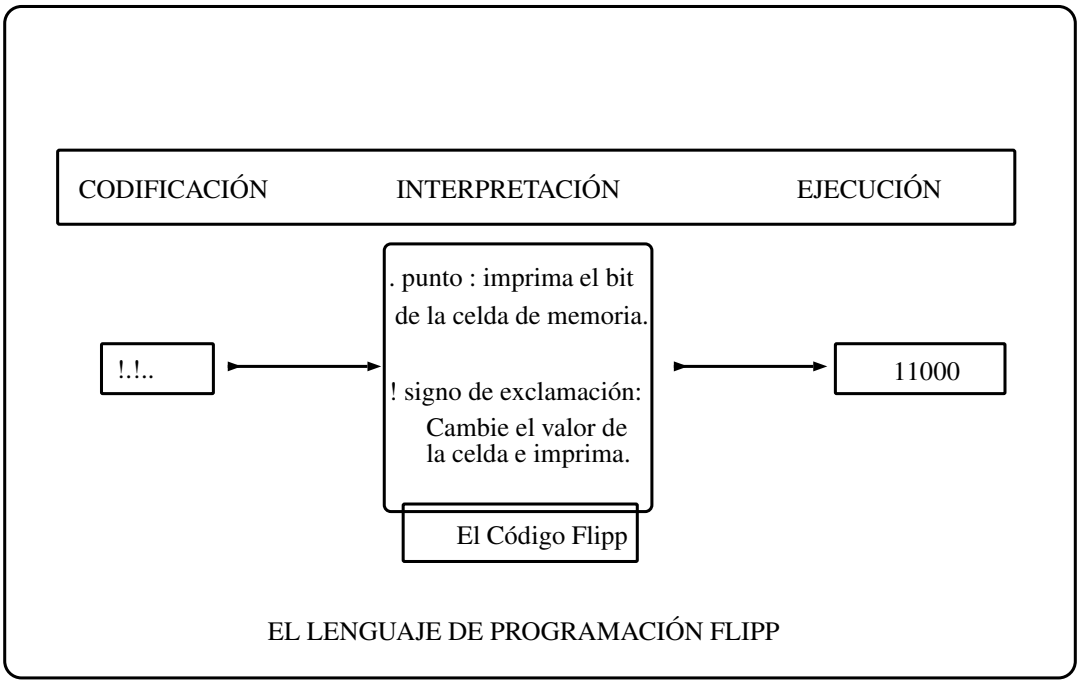


Figura 3: *Flipp es un lenguaje de programación con dos comandos, el punto, y el signo de exclamación. El punto . significa:* `imprimir el contenido de la celda de memoria`*. El signo de exclamación ! significa* `cambiar el valor de la celda de memoria e imprimir`*.*

a que ningún polinomio puede ajustarse a una exponencial en rango largo, sino que el grado del polinomio aumentará a medida que se incorporen más puntos. Por el contrario, nuestras regresiones exponenciales tienden a estabilizarse.

Ejercicio 8, page 19. El programa `GAcost.html` muestra que el costo de ESPEC para los AGs es también exponential pero mucho menor que el de la PG, dado por el programa `PGwithFlipp.html`. ¿Por qué? Nuestra opinión es que se debe a que el comando `Exclamación` crea interferencias. De hecho, invierte el valor de la memoria y eso cambia lo que el comando `Punto` imprime a continuación. Por lo tanto, es más difícil separar los caracteres en PG que en GA.

Ejercicio 9, page 20. El resultado exponencial ha sido convertido por la mencionada afirmación en un teorema de `Evolución sin futuro`. Esto contrasta con lo que la Ciencia nos muestra cada día: La Evolución es una herramienta poderosa tanto para el diseño como para encontrar soluciones aproximadas a problemas de optimización. Por lo tanto, la postura de un teorema de no-futuro es inapropiada: tiende a cegar nuestra imaginación. Más concretamente, consideremos el siguiente escenario: La Evolución puede hacer coincidir cadenas cortas, pero no puede hacer coincidir cadenas grandes con exactitud. La solución evolutiva a esta tensión es sencilla: los trozos pequeños de ADN adquirirán una función y evolucionarán en consecuencia hasta convertirse en los genes modernos de los organismos actuales. El análisis de complejidad de esta idea es el siguiente: Mientras que las cadenas binarias, así como el lenguaje Flipp, tienen un alfabeto con dos símbolos, el ADN tiene cuatro. Pero una vez que consideramos los genes, entramos en otro lenguaje con un número ilimitado de símbolos. En un entorno evolutivo, podrían preferirse genes más cortos. Así pues, podríamos considerar un alfabeto con un número finito pero grande de genes que eventualmente cumplan diferentes funciones. Con este nuevo alfabeto, volvemos a encontrarnos con el problema de especificar una función dada de alto orden, digamos de supervivencia, y esperamos un coste exponencial, igual que antes. Por lo tanto, la aparición de genes no es una solución asegurada a la falsación de la TE planteada por el coste exponencial de la PG. Pero queda una duda.

Ejercicio 10, page 20. Nos encontramos con la misma tensión que antes: es demasiado costoso desarrollar cadenas binarias largas imperfectas, por lo que surgirán genes a medias que en combinaciones aleatorias coevolucionarán en grandes unidades funcionales. Y, como el coste de ascender por la escalera de la Perfección es exponencial, la Imperfección es obligatoria: debe estar presente tanto en el registro fósil como en mi propio cuerpo. Esto es falso. La perfección de los esqueletos de perros recientemente muertos despierta en mí una admiración tan grande como la de los dinosaurios y la de los fósiles de hojas en la roca. Y con respecto a mí, siendo consciente de todos mis defectos, me declaro perfecto: me maravilla continuamente que mi corazón todavía trabaje y trabaje y trabaje después de 67 años de servicio sin descanso. ¡Simplemente maravilloso!

Ejercicio 11, page 21. Los intervalos de confianza predicen las observaciones bajo repeticiones y más allá de los datos reales. Entre más estrechos, más convincentes son las estimaciones. Sucede que los intervalos de confianza son tanto más amplios cuanto más alejados están del rango de los datos registrados. Precisamente, nuestro propósito aquí es extrapolar valores del rango [1, 30] al reino de los miles de millones (nuestro ADN contiene 3.000 millones de bases). Por lo tanto, los intervalos de confianza no

tienen sentido para esa extrapolación porque serían demasiado amplios para ser de interés. Incluso la afirmación más simple de un coste exponencial carecería de valor para algunos puristas. Entonces, nos preguntamos: ¿Es inútil nuestro trabajo? No, no lo es. En primer lugar, las críticas serían importantes en el único caso en el que el coste de la PG binaria resultase ser inferior al exponencial, es decir, polinómico de bajo grado. Tal resultado impulsaría una revolución en la industria del software y la mayoría de los ingenieros se quedaría sin trabajo. Pero esto es inesperado para la mayoría de la gente, incluso si se emplea la computación cuántica. En segundo lugar, nuestros resultados guiarían la formulación de modelos matemáticos que producirían por sí mismos límites inferiores del costo para cualquier rango. Por muy desalentadora que sea esta tarea, programe su mente con la convicción de que puede hacer algo serio. Y permítase intentarlo. En este trabajo se ofrecen varias pistas.

Ejercicio 12, page 22. Cuando la Evolución transcurre sin objetivos externos, todas las funciones posibles van apareciendo hasta que algunas pocas comienzan a ser lo suficientemente perfectas como para generar individuos más aptos que el resto. Es así como surge una fuerza selectiva direccional que guía el Proceso Evolutivo. Y mientras evolucionan hacia la perfección, de todos modos, otros muchos experimentos paralelos están en marcha. Así, la Evolución Auto-Organizada es, con mucho, más cara que la Dirigida, que premia a los más aptos. Pero, si miramos bien, nuestra discusión muestra que la Evolución sin teleonomía no es buena para producir secuencias específicas, pero que sí es buena en la búsqueda de organismos complejos. El problema es que de todas maneras, se necesitarían 7 eternidades para empezar a ver algo interesante.

Ejercicio 13, page 22. La Cultura Científica reconoce que maravillosas sorpresas pueden venir de ese mundo que no es el centro de nuestra atención. Se ha dado un nombre a esta situación: serendipia. Pero está claro que las universidades no esperan que esas sorpresas les den categoría. Más bien trabajan ordenadamente sobre programas de investigación con un objetivo claro. Esto me invita a pensar que el coste de la Evolución por pura serendipia es tremendamente caro en comparación con la Evolución Dirigida y la PG. Así, el coste tanto de la Perfección como de la Imperfección en la Evolución sin objetivo es de todos modos exponencial en la longitud de la entrada. Esto implica que en el Darwinismo, que no impone ninguna meta, la mediocridad y la Imperfección deben abundar para dejar su lugar a la eterna sub-perfección hasta la eventual llegada de la Perfección. Esto no se encuentra en la Naturaleza.

Ejercicio 14, page 25. Dado que la PG es la Evolución de software, esperamos que las discontinuidades desempeñen un papel dominante y, por tanto, que la tendencia a la perfección sea destructiva. Por otro lado, nuestra conjetura sobre el coste de alcanzar la coincidencia parcial de longitud k es que es exponencial y con exactamente la misma expresión que el coste de coincidir con una contraseña de longitud k encontrada previamente. Ponemos a prueba nuestra creencia con un programa que aparece un poco más adelante.

Ejercicio 15, page 26. El tema es difícil. Por decir lo menos, los AGs también contienen software: cuando se codifican las soluciones de un problema como cadenas, se evolucionan y se decodifican en soluciones, utilizamos reglas de traducción, es decir, estamos tratando con software, aunque de baja expresividad. Por lo tanto, también debe haber discontinuidad en los AGs. En efecto, un AG puede mutar la cadena 0000 en 1000, lo cual es un salto enorme. Con recombinación sucede algo muy parecido. ¿Cuál

es el efecto de esa discontinuidad sobre la dinámica de una simulación de un AG? El programa `GAevoOfPerfection.html` presenta el tapiz de la perfección para los AGs. Encontramos que el gradualismo es definitivamente más visible para los AGs que para la PG sobre Flipp. No obstante, también se observan casos de saltos en los que el cuadrado rojo no está conectado a uno violeta. Así pues, no se puede ignorar el papel perturbador ni de la mutación, ni de la recombinación, ni de la interpretación o semántica de las cadenas al traducirlas a números. Pero nuestro programa no da ni una pista sobre cómo separar dichos efectos.

Ejercicio 16, page 26. Las fluctuaciones son el combustible que impulsa la Evolución hacia la perfección. Las fluctuaciones favorables son multiplicadas por la selección y, eventualmente, otra mejora puede ser hecha por mutación o recombinación. Pero la magnitud de las fluctuaciones disminuye a medida que crece la longitud de la entrada. Esto implica que la perfección es tanto más costosa cuanto mayor es la complejidad. Más concretamente, su coste es exponencial. Por lo tanto, la Evolución es a la vez abierta y permanentemente revestida de productos imperfectos. Es abierta porque nadie puede predecir qué milagro ocurrirá a continuación. Por otra parte, la Evolución deshace en un instante lo que posiblemente tardó siglos en conseguirse. Así pues, la imperfección reina. Eternamente. Por eso mi propio cuerpo debe estar lleno de todo tipo de imperfecciones anatómicas y funcionales. Al abrir y cerrar mi mano izquierda, descubro que eso es falso. La TE Verbal ha sido falsada en mi propio cuerpo. ¡Que maravilloso!

Ejercicio 17, page 27. Dado que a partir de ahora la Evolución es científica, todas las afirmaciones de eventos evolutivos pertenecen al ámbito de la Ciencia. Esto significa que ahora pueden ser sometidas a prueba. Presentamos algunos ejemplos para ilustrar cómo salir de la Mitología para celebrar la potenciación del Método Científico en el ámbito de la Evolución Teórica:

- Consideremos el caso de los chimpancés y los humanos. Tenemos más de mil millones de razones para considerar que están emparentados por descendencia, es decir, que tienen un ancestro común. Por lo tanto, es una hipótesis razonable y merece una investigación científica. Para fijar ideas, consideremos que el ancestro común es el chimpancé. Ahora, usemos la selección artificial para hacer evolucionar al chimpancé para que camine erguido. Esto debe ser un reto muy fácil, porque los chimpancés caminan así ocasionalmente. Un zoológico es el lugar más apropiado para llevar a cabo este experimento. Puede ocurrir que el chimpancé no sea el ancestro común, así que, calcúlelo con precisión, haga una regresión del chimpancé a ese ancestro, y vuelva a la Evolución hacia lo humano.
- Utilice la Evolución Dirigida para hacer evolucionar la mano del chimpancé a una humana y prepárese para registrar los cambios en su cerebro y comportamiento. Éste es un programa de Evolución de la anatomía, pero puede cambiarse a la evolución de las funciones: seleccione el comportamiento, podría ser el uso de herramientas de manera humana, y registre la Evolución de la mano y de los circuitos neuronales del cerebro y de la médula espinal.
- Ponga a prueba la idea de que la mitocondria y el cloroplasto son el resultado del engullimiento de una célula por otra (Scitable [30] 2014). Para ello, seleccione bacterias engullidas que proporcionen al huésped energía, azúcares y otras sustancias. La simbiosis se promete tan fructífera que la célula anfitriona sacrifica su

integridad en favor de una organización más poderosa en la que la célula invitada ya no es un parásito sino una parte integrada y productiva de todo el sistema. No se espera que aparezcan mitocondrias reales, sino sólo sustitutos útiles.

- El código genético tiene una variación natural (Kooning y Novoshilov, [16] 2009) y el análisis teórico parece implicar que hay muchas posibilidades para su Evolución (Xu y Zhang [37] 2021) aunque no tanto como se desea (Terres et al [33] 2000). Ahora, utilice la selección artificial para conectar el código genético estándar de las bacterias con el de las mitocondrias: inyecte una bacteria en el plasma de otra y seleccione a continuación el código genético de una mitocondria y continúe con la selección de una mitocondria por completo. Elija sabiamente sus especímenes y, por supuesto, los microbios elegidos deben no tener mitocondrias (Martin [21] 2010). Y asegúrese de que parte del ADN de las mitocondrias nacientes debe migrar al núcleo de la célula anfitriona para que todo el conjunto sea altamente funcional.

- La naturaleza es tan extraña que permite maravillas tan asombrosas como el ADN Hachimoji, que está construido con ocho letras en lugar de las cuatro del Código Genético Natural (Hirao et al, [11] 2002). Reinvente esto o algo mejor mediante la Evolución Dirigida. Un programa como éste ya está en marcha porque cubre una necesidad urgente de la biotecnología (DeBenedictis et al, [6] 2021).

- Un espermatozoide puede entrar en un óvulo porque el primero tiene la llave de la puerta cifrada del segundo. El cambio en la llave o en la puerta es suficiente para interferir la fecundación (Lobov et al, [20] 2019). Pero puede ocurrir que ambos mutantes se reconozcan entre sí. El producto es una nueva especie. Utilice esta estrategia para crear dos especies y seguir su Evolución posterior. O también, deshaga los cambios que causan la divergencia de dos especies y reporte las posibilidades de cruce de las nuevas versiones.

Referencias

[1] ADAMS JILL 2008
Paternity Testing: Blood Types and DNA Nature Education 1(1):146
https://www.Nature.com/scitable/topicpage/paternity-testing-blood-types-an
11

[2] SANJEEV ARORA AND BOAZ BARAK 2007
Computational Complexity: A Modern Approach
https://theory.cs.princeton.edu/complexity/book.pdf
18

[3] BERKELEY: UNDERSTANDING EVOLUTION 2010?
Using trees for classification
https://evolution.berkeley.edu/evolibrary/article/phylogenetics_04
4

[4] DARWIN C 1859
The origin of species Murrap, Londres.
Darwin Online

http://darwin-online.org.uk/content/frameset?itemID=F373&viewtype=side&page
Online Variorum of Darwin's Origin of Species: first British edition (1859),
page 186
http://darwin-online.org.uk/Variorum/1859/1859-187-c-1861.html
3, 4

[5] DAVIES NAHLA 2020 *Neural networks could help computers code themselves: Do we still need human coders?* The Overflow, sept 16/2020
https://stackoverflow.blog/2020/09/16/neural-networks-could-help-computers
10

[6] DEBENEDICTIS, E.A., CARVER, G.D., CHUNG, C.Z. ET AL. 2021
Multiplex suppression of four quadruplet codons via tRNA directed evolution.
Nature Communications volume 12, Article number: 5706 (2021)
https://www.Nature.com/articles/s41467-021-25948-y 34

[7] S J FREELAND 1 , L D HURST 1998
The genetic code is one in a million. J Mol Evol 1998 Sep;47(3):238-48.
doi: 10.1007/pl00006381.
https://pubmed.ncbi.nlm.nih.gov/9732450/ 13

[8] GENETIC PROGRAMMING GROUP
Official website https://geneticprogramming.com/ 14

[9] A. D. HALL 1935
HUGO DE VRIES 1848-1935
https://royalsocietypublishing.org/doi/pdf/10.1098/rsbm.1935.0002
5

[10] HENNIG, BORIS. 2011
Teleonomy.
Philosophiegeschichte und Logische Analyse. 14. 185-202. 10.30965/26664275-01401012.
https://www.researchgate.net/publication/288275673_Teleonomy
22

[11] HIRAO, ICHIRO & OHTSUKI, TAKASHI & FUJIWARA, TSUYOSHI & MITSUI, TSUNEO & YOKOGAWA, TOMOKO & OKUNI, TAEKO & NAKAYAMA, HIROSHI & TAKIO, KOJI & YABUKI, TAKASHI & KIGAWA, TAKANORI & KODAMA, KOICHIRO & YOKOGAWA, TAKASHI & NISHIKAWA, KAZUYA & YOKOYAMA, SHIGEYUKI, 2002.
An unnatural base pair for incorporating amino acid analogs into proteins.
Nature biotechnology. 20. 177-82. 10.1038/nbt0202-177.
https://www.researchgate.net/publication/11538018_An_unnatural_base_pair_fo
34

[12] INGO BERG
Simulated Evolution - An algorithm for simulating evolution in a web browser
https://beltoforion.de/en/simulated_evolution/index.php#idStart
Simulierte Evolution - Simulieren von Evolutionsprinzipien mit dem Computer
https://beltoforion.de/de/simulierte_evolution/index.php
22

[13] Caifu Jiang, Aziz Mithani, Xiangchao Gan, Jiannis Ragoussis, Richard Mott, Nicholas P. Harberd and others 2011
Regenerant Arabidopsis Lineages Display a Distinct Genome-Wide Spectrum of Mutations Conferring Variant Phenotypes
Current Biology, Report— Volume 21, ISSUE 16, P1385-1390, August 23, 2011
https://www.cell.com/current-biology/fulltext/S0960-9822(11)00771-8 11

[14] Lynn Jorde, Ph.D. 2016
NHGRI Current Topics in Genome Analysis 2016 Week 7: Introduction to Population Genetics April 6, 2016
https://www.genome.gov/sites/default/files/media/files/2019-03/Introduction 5

[15] Kenneth J. Loceya, and Jay T. Lennon 2016
Scaling laws predict global microbial diversity. PNAS, 2016 DOI: 10.1073/pnas.1521291113
https://www.pnas.org/content/113/21/5970 27

[16] Eugene V. Koonin and Artem S. Novozhilov 2009
Origin and evolution of the genetic code: the universal enigma
IUBMB Life. 2009 Feb; 61(2): 99–111.
doi: 10.1002/iub.146
PMCID: PMC3293468
NIHMSID: NIHMS78555
PMID: 19117371
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3293468/ 34

[17] John R. Koza, Riccardo Poli 2005
Genetic Programming. Search Methodologies (pp.127-164)Publisher: Springer US, New York, NY, USAEditors: Burke, Edmund K., and Kendall, Graham.
https://www.researchgate.net/publication/229091441_Genetic_Programming 14

[18] Kurzgesagt – In a Nutshell (2018)
The Deadliest Being on Planet Earth – The Bacteriophage
https://www.youtube.com/watch?v=YI3tsmFsrOg 9

[19] Jean-Baptiste Pierre Antoine de Monet, chevalier de Lamarck 1809
PHILOSOPHIE ZOOLOGIQUE 1809
1er août 1744, Bazentin, Somme, 18 décembre 1829, Paris.
https://ia801004.us.archive.org/28/items/LamarckPZ/Lamarck_PZ.pdf 4

[20] Lobov AA, Maltseva AL, Mikhailova NA, Granovitch AI. 2019
The Molecular Mechanisms of Gametic Incompatibility in Invertebrates. *Acta Naturae. 2019 Jul-Sep;11(3):4-15. doi: 10.32607/20758251-2019-11-3-4-15. PMID: 31720011; PMCID: PMC6826153.*
https://pubmed.ncbi.nlm.nih.gov/31720011/ 34

[21] MARTIN WILLIAM, MENTEL MAREK , 2010
The Origin of Mitochondria
Nature Education 3(9):58.
https://www.Nature.com/scitable/topicpage/the-origin-of-mitochondria-14232
34

[22] NHGRI , 2021
Genome
https://www.genome.gov/genetics-glossary/Genetic-Code
9

[23] MCLEOD SAUL 2020
Karl Popper - Theory of Falsification.
SimplyPSychology
https://www.simplypsychology.org/Karl-Popper.html 4

[24] HELEN PEARSON 2003
250 years of Linnaeus' plant names celebrated.
Nature News— doi:10.1038/news030825-7
https://www.Nature.com/news/2003/030825/full/news030825-7.html
4

[25] VINCENT RACANIELLO 2013
How many viruses on Earth? Virology blog
https://www.virology.ws/2013/09/06/how-many-viruses-on-earth/
9

[26] JOSÉ RODRÍGUEZ 2010 *The genome is software and evolution is a software developer*. Arxiv 1003.0575
https://arxiv.org/abs/1003.0575 11

[27] JOSÉ RODRÍGUEZ 2017 *EASY JAVA GENETIC PROGRAMMING. JAVA FOR THE STUDY OF EVOLUTION VOLUME XVII version 2*
https://ejristos.com/dl/javaevol17v2main.pdf 14

[28] JOSÉ RODRÍGUEZ 2019 *Christianity, evolution, complexity, simulations, and passwords v2*.
Text without programs:
https://ejristos.com/dl/PasswordsText.html
Text and programs:
https://ejristos.com/dl/PasswordsHtml.zip 5

[29] JOSÉ RODRÍGUEZ 2021 *The Darwinian Evolutionary Theory at its face value*.
Text without programs:
https://ejristos.com/dl/DarwinAtFaceValue.pdf
https://ejristos.com/dl/DarwinEnSerio.pdf (Spanish)
Text and programs:
https://ejristos.com/dl/DarwinAtFaceValue.zip 5, 6

[30] SCITABLE 2014
The origin of mitochondria and chloroplasts
https://www.Nature.com/scitable/content/the-origin-of-mitochondria-and-chl
33

[31] SCIENCE LEARNING HUB – POKAPŪ AKORANGA PŪTAIAO 2011.
Mendel's experiments
https://www.sciencelearn.org.nz/pages/about 4

[32] SPECTOR, L.,H. BARNUM,AND H.J. BERNSTEIN.1998.
Genetic Programming for Quantum Computers. In Genetic Programming 1998: Proceedings of the Third Annual Conference, edited by J.R.Koza, W.Banzhaf, K.Chellapilla, K.Deb, M.Dorigo, D.B.Fogel, M.H.Garzon, D.E.Goldberg, H.Iba, and R.L.Riolo. pp.365-374. San Francisco, CA: Morgan Kaufmann.
http://faculty.hampshire.edu/lspector/pubs/GP-quantum-GP98-with-cite.pdf 15

[33] TERRES A. RONNEBERG, LAURA F. LANDWEBER, AND STEPHEN J. FREELAND 2000
Testing a biosynthetic theory of the genetic code: Fact or artifact?
PNAS December 5, 2000 97 (25) 13690-13695;
https://doi.org/10.1073/pnas.250403097 34

[34] THALER 2018 DAVID S. *Why should mitochondria define species? 2018.*
Human Evolution 33(n1-2) May 2018
https://www.researchgate.net/publication/325270582_Why_should_mitochondria_ 11

[35] W3SCHOOLS 2020
The code game
https://www.w3schools.com/codegame/index.html 10

[36] CONTRIBUTEURS DE WIKIPÉDIA 2021.
Jean-Baptiste de Lamarck [Internet]. Wikipédia, l'encyclopédie libre; 2021 fév. 15, 00:13 UTC [cited 2021 fév. 15]. Disponible à l'adresse: http://fr.wikipedia.org/w/index.php?title=Jean-Baptiste_de_Lamarck&oldid=1 4

[37] HAIQING XU AND JIANZHI ZHANG 2021.
On the Origin of Frameshift-Robustness of the Standard Genetic Code. Mol Biol Evol. 2021 Oct; 38(10): 4301–4309.
PMCID: PMC8476161
PMID: 34043802:
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8476161/ 34

All cited links were active by 14/XII/2021