

ТРИ ЦАРСТВА, ТРИ СТРАТЕГИИ: ИНФОРМАЦИОННАЯ ЁМКОСТЬ k В
КЛАССИФИКАЦИИ 92 ВИДОВ ЭУКАРИОТ НА ОСНОВЕ ГЕОМЕТРИИ КАЛЕЙДОЦИКЛА
L8a21 (ТЕОРИЯ KEDEM-CYCLE Ω)

Автор: Бельмасова Ирина Юрьевна

ORCID: 0009-0008-9902-1245

Email: irinabelmasova@yandex.ru

Дата: Июль 2026 г.

Статус: Препринт, версия 3.0

СОДЕРЖАНИЕ

1. Введение

- 1.1. Геометрический фундамент: теория Kedem-Cycle Ω
- 1.2. Гипотеза о первичности информации
- 1.3. Параметр k : от информационной массы к информационной ёмкости
- 1.4. Два фильтра и их роль в эволюции
- 1.5. Закон накопления нестабильных мод
- 1.6. Цель данной работы

2. Методы

- 2.1. Источники данных
- 2.2. Выборка
- 2.3. Вычисление информационной массы k
- 2.4. Модель и классификация
- 2.5. Статистический анализ

3. Результаты

- 3.1. Систематический аудит исходных данных
- 3.2. Три стратегии у животных ($n=51$)
- 3.3. Растения: пологий наклон и поиск механизма
- 3.4. Грибы: насыщение и роль CpG-островков
- 3.5. Объединённые данные растений и грибов
- 3.6. Межцарственные законы масштабирования
- 3.7. Связь с ферментацией: кратность 12к
- 3.8. Экспериментальная верификация оператора R
- 3.9. Архитектура генов как детерминанта наклона масштабирования

4. Обсуждение

- 4.1. Крах старой классификации
- 4.2. Три стратегии как проявления SR-фильтра
- 4.3. Почему животные масштабируют круто, растения полого, а грибы насыщаются
- 4.4. k как информационная ёмкость
- 4.5. Границы применимости

5. Заключение

Литература

Доступность данных

Приглашаем вас к эксперименту

Приложение: проверочные коды

АННОТАЦИЯ

Мы обнаружили три универсальные эволюционные стратегии сложности сплайсинга, которые проявляются во всех трёх царствах эукариот — у животных, растений и грибов. Используя параметр k из геометрической теории Kedem-Cycle Ω как меру информационной ёмкости генома, мы показали, что криптобиотические виды (тихоходки, коловратки, артемии) идут по пути гипер-упрощения — их сплайсинг минимален, что служит защитой от ошибок при перезапуске метаболизма. Стрессоустойчивые долгоживущие виды (голый землекоп, слепыши, черепахи, аллигатор) идут по пути адаптивного усложнения — они производят больше изоформ, что даёт им пластичность для выживания в агрессивной среде. Большинство видов следуют стандартной стратегии, точно соответствующей предсказанию модели.

Ключевым результатом работы стал систематический аудит исходных данных, показавший, что общепринятая практика использования общих транскриптов Ensembl завышает число сплайс-форм на 15–105%. Это привело к полному пересмотру классификации: ранее выделенный «интенсивный путь» оказался статистическим артефактом. На скорректированных данных мы установили, что три царства различаются по закону масштабирования сложности сплайсинга: животные демонстрируют крутой линейный наклон (0.247), растения — пологий линейный (0.103), грибы — насыщающуюся кривую ($\Delta R^2 = +0.19$ над линейной моделью).

Мы идентифицировали два механизма, лежащих в основе этих различий. На межцарственном уровне оператор R , предсказанный геометрической теорией [1, Приложение CP], идентифицирован как CpG-островки. Плотность CpG значительно различается между царствами (животные 10.7, растения 5.8, грибы 3.1 CpG/Мб, $p < 10^{-4}$). На структурном уровне разница в наклонах объясняется архитектурой генов: среднее число экзонов на ген составляет 9.1 у животных, 5.0 у растений и 2.5 у грибов ($p < 0.01$ для всех попарных сравнений). Число экзонов strongly коррелирует с SF/ген ($r = 0.83$, $p < 10^{-5}$) и определяет верхний предел сложности альтернативного сплайсинга. Отношение среднего числа экзонов у животных и растений (1.83×) близко к отношению наклонов (2.39×).

Разность информационной ёмкости между адаптивной и стандартной стратегиями кратна критическому порогу ферментации $12k$ ($\Delta/12k = -1.97$, отклонение 0.029), что связывает эволюционную биологию с геометрией калейдоцикла L8a21. Полученные результаты указывают на то, что биология и физика являются не двумя разными мирами, а двумя проявлениями единого информационного процесса, квантованного фундаментальной константой $k = 1/(3\pi)$.

1. ВВЕДЕНИЕ

1.1. Геометрический фундамент: теория Kedem-Cycle Ω

Настоящая работа является частью геометрической теории Kedem-Cycle Ω [1, 2], основанной на гиперболическом 3-многообразии-калейдоцикле L8a21 из общедоступного каталога SnapPy. Из его геометрии без подгоночных параметров выводятся массы 28 элементарных частиц (средняя точность 0.057%), калибровочные группы $SU(3) \times SU(2) \times U(1)$ со строгим доказательством $A_6 \rightarrow SU(3)$, фундаментальные константы (постоянная тонкой структуры, постоянная Хаббла, возраст Вселенной), космологические параметры и семь частиц тёмной материи [1].

1.2. Гипотеза о первичности информации

Согласно гипотезе, сформулированной в [1, Приложение ИТ], информация является фундаментальной субстанцией, первичной по отношению к геометрии. Информация существует в трёх формах: геометрическая ($\theta_0=36^\circ$), физическая ($\Delta_{\text{spin}}=2.63^\circ$) и живая (δ_{ferm}). Процесс накопления живой информации назван ферментацией. Ферментация квантована: за один оборот калейдоцикла накапливается 10к, критический порог фазового перехода $K_6 \rightarrow K_7$ составляет 12к, полный цикл — 40к [1, Приложения ИТ, КС].

1.3. Параметр k: от информационной массы к информационной ёмкости

В физике k проявляется как информационная масса (частица с большим k тяжелее), но по своему происхождению это информационная ёмкость — мера того, сколько различий реальность совершила на данном уровне. В биологии k интерпретируется как мера эволюционных смыслов, накопленных видом за историю своего существования.

1.4. Два фильтра и их роль в эволюции

CP-фильтр (пассивный проектор) встроен в геометрию L8a21 и отбирает $p_{\text{stable}}=0.6$ стабильных мод. SR-фильтр (активный резонатор) реализует право вето: $SR=P \circ R \circ P$, где R — оператор расширения, геометрически идентифицированный как индуцированное покрытие $L8a21 \rightarrow Z$ через вершину 9 (D-сектор) [1, Приложение CP].

1.5. Закон накопления нестабильных мод

Доля нестабильной информации = $1-(0.6)^N$. Из закона следует: $N=\ln(1-\text{доля не кодирующей ДНК})/\ln(0.6)$, $k=10^{(N \times 2.4)}$. Коэффициент $2.4=12/5$ связывает критический порог 12к с $k=1/(3\pi)$.

1.6. Цель данной работы

Проверить, является ли k универсальным параметром порядка, коррелирующим с биологической сложностью во всех трёх царствах эукариот.

2. МЕТОДЫ

2.1. Источники данных

Данные по геномам: NCBI GenBank (размеры геномов, число генов); Ensembl, Ensembl Plants и Ensembl Fungi, release 112 (доля некодирующей ДНК, число белок-кодирующих транскриптов). Критически важно: использовались только транскрипты, аннотированные как `protein_coding`. Общие транскрипты Ensembl завышают SF/ген на 15–105% и не должны использоваться для межвидовых сравнений.

2.2. Выборка

92 вида: 51 животное, 21 растение, 20 грибов. Исходная выборка скорректирована после систематического аудита.

2.3. Вычисление информационной массы k

$N = \ln(1 - \text{доля некодирующей ДНК}) / \ln(0.6)$, $\log_{10}(k) = N \times 2.4$.

2.4. Модель и классификация

Базовая модель: $SF/\text{ген} = a \times \log_{10}(k) + b$. Виды классифицируются по остаткам: гипер-упрощение (остаток ≤ -0.10), стандарт (от -0.10 до $+0.10$), адаптивное усложнение ($> +0.10$). Междарственные сравнения — ANCOVA.

2.5. Статистический анализ

Тест Краскела-Уоллиса, попарные сравнения Манна-Уитни с поправкой Бонферрони, бутстрэп-валидация (10 000 итераций), ANCOVA с F-тестом.

3. РЕЗУЛЬТАТЫ

3.1. Систематический аудит исходных данных

Аудит 44 видов животных выявил систематическое завышение SF/ген (среднее +35.2%). Для видов, ранее классифицированных как «интенсивный путь», завышение достигало 71–105%. На скорректированных данных ни один вид не показал остатка $> +0.5$. Исходная классификация полностью пересмотрена.

3.2. Три стратегии у животных ($n=51$)

Модель: $SF/\text{ген} = 0.2465 \times \log_{10}(k) + 1.1284$, $R^2 = 0.929$. Тест Краскела-Уоллиса: $H = 13.18$, $p = 0.0014$.

Стратегия 1: Гипер-упрощение ($n=10$, остаток -0.15). Тихоходки, коловратки, артемия, хирономиды — все криптобиотические виды.

Стратегия 2: Стандарт ($n=26$, остаток -0.03). Эволюционный мейнстрим.

Стратегия 3: Адаптивное усложнение ($n=15$, остаток $+0.16$). Голый землекоп, слепыши, черепахи, аллигаторы, тенрек, звездонос — стрессоустойчивые долгоживущие виды.

Попарные сравнения: Гипер-упрощение vs Адаптивное $p=0.001$, Стандарт vs Адаптивное $p=0.005$, (Гипер-упрощение+Стандарт) vs Адаптивное $p=0.0008$.

3.3. Растения: пологий наклон и поиск механизма

Наклон растений (0.103) значимо выше, чем у животных (0.247 , ANCOVA: $F=18.9$, $p=1.9 \times 10^{-4}$). Для объяснения этого различия были проверены три гипотезы. Гипотеза о полиплоидии не подтвердилась: корреляция плоидности с остатками $r=-0.06$, $p=0.68$ на 46 видах. Гипотеза о CpG-островках не подтвердилась: корреляция CpG с остатками $r=0.20$, $p=0.18$ на 46 видах. Гипотеза о качестве некодирующей ДНК (регуляторная плотность) дала неопределённый результат из-за несопоставимости методологий. Механизм, определяющий пологий наклон растений, был найден в архитектуре генов (раздел 3.9).

3.4. Грибы: насыщение и роль CpG-островков

Модель с насыщением превосходит линейную: $\Delta R^2=+0.19$. Наклон 0.044 — в 5.6 раза выше, чем у животных (ANCOVA: $F=18.2$, $p=2.5 \times 10^{-4}$). Плотность CpG у грибов (3.1 CpG/Мб) значимо ниже, чем у растений (5.8 CpG/Мб, $p=6.3 \times 10^{-5}$). Корреляция CpG с остатками: $r=0.72$, $p=0.0003$. При плотности CpG ниже ~ 3 Мб $^{-1}$ система переходит от линейного масштабирования к насыщению.

3.5. Объединённые данные растений и грибов

Три стратегии подтверждены: тест Краскела-Уоллиса $p=0.002$.

3.6. Межцарственные законы масштабирования

ANCOVA: Животные vs Растения — наклоны различаются ($F=18.9$, $p=1.9 \times 10^{-4}$). Животные vs Грибы — наклоны различаются ($F=18.2$, $p=2.5 \times 10^{-4}$). Растения vs Грибы — наклоны не различаются ($F=0.44$, $p=0.51$).

Отношения плотностей CpG и наклонов между царствами: Животные/Растения — CpG=1.85, наклон=2.39; Животные/Грибы — CpG=3.48, наклон=5.55; Растения/Грибы — CpG=1.89, наклон=2.32. Отношения CpG и наклонов близки для пары Животные/Растения, но для грибов отношение наклонов существенно превышает отношение CpG, что отражает дополнительный эффект насыщения при низкой плотности CpG.

3.7. Связь с ферментацией: кратность 12к

Разность $\log_{10}(k)$ между Адаптивной и Стандартной стратегиями: $\Delta/12k=-1.97$, отклонение 0.029 . Отрицательный знак означает, что адаптивные виды достигают

более высокой сложности сплайсинга при меньшей информационной ёмкости — более эффективное использование накопленной информации.

3.8. Экспериментальная верификация оператора R

Идентификация оператора R с CpG-островками [1, Приложение CP] проверена на данных трёх царств. Плотность CpG значительно различается между всеми тремя царствами ($p < 10^{-4}$). Для растений корреляция CpG с остатками не значима ($r=0.20$, $p=0.18$). При плотности CpG ниже $\sim 3 \text{ Mb}^{-1}$ система переходит от линейного масштабирования к насыщению, что указывает на пороговый характер ферментации.

3.9. Архитектура генов как детерминанта наклона масштабирования

Среднее число экзонов на ген значительно различается между всеми тремя царствами: животные 9.1 ± 1.7 , растения 5.0 ± 0.3 , грибы 2.5 ± 0.9 ($p < 0.01$ для всех попарных сравнений). Число экзонов strongly коррелирует с SF/ген на всём множестве видов ($r = 0.83$, $p = 3.1 \times 10^{-6}$ на 21 виде). Корреляция значима у животных ($r = 0.72$, $p = 0.04$) и грибов ($r = 0.90$, $p = 0.04$), но не у растений ($r = -0.01$, $p = 0.98$), что указывает на действие дополнительных факторов в растительном царстве. Отношение среднего числа экзонов у животных и растений ($1.83\times$) близко к отношению наклонов масштабирования сплайсинга ($2.39\times$). У грибов, где число экзонов минимально (2.5), модель с насыщением превосходит линейную ($\Delta R^2 = +0.19$).

Эти результаты указывают на простой структурный механизм: число экзонов в гене определяет верхний предел сложности альтернативного сплайсинга. Рост некодирующей ДНК (k) может увеличивать SF/ген только в той мере, в какой это позволяет архитектура генов. Животные с их многоэкзонными генами масштабируют сплайсинг агрессивно; грибы с их малоэкзонными генами достигают плато; растения занимают промежуточное положение.

4. ОБСУЖДЕНИЕ

4.1. Крах старой классификации

Систематическое завышение SF/ген создало артефактный «интенсивный путь». Общие транскрипты Ensembl непригодны для межвидовых сравнений без фильтрации по protein_coding.

4.2. Три стратегии как проявления SR-фильтра

Гипер-упрощение — максимальная строгость SR-фильтра, адаптивное усложнение — ослабленная строгость. Параметр D_stress количественно выражает режим работы SR-фильтра: -0.15 для гипер-упрощения, 0.00 для стандарта, $+0.16$ для адаптивного усложнения.

4.3. Почему животные масштабируют круто, растения полого, а грибы насыщаются

Два механизма действуют на разных уровнях. На структурном уровне число экзонов в гене определяет верхний предел сложности альтернативного сплайсинга. Животные (9.1 экзонов/ген) имеют больше сайтов для альтернативного сплайсинга, чем растения (5.0 экзонов/ген), и значительно больше, чем грибы (2.5 экзонов/ген). Это создаёт фундаментальное ограничение: рост некодирующей ДНК может увеличивать SF/ген только в той мере, в какой это позволяет архитектура генов.

На регуляторном уровне CpG-островки (оператор R) определяют эффективность, с которой некодирующая ДНК конвертируется в сплайс-формы. У грибов низкая плотность CpG ($\sim 3 \text{ Мб}^{-1}$) ограничивает ферментацию настолько, что система выходит на плато даже при дальнейшем росте генома.

4.4. k как информационная ёмкость

Рост k от электрона ($k=0$) до человека ($k \sim 10^{12}$) — история накопления смыслов от первых различий до самосознания.

4.5. Границы применимости

Модель не работает для систем с сознательным регулированием (языки, города), что совпадает с границей, установленной для закона накопления нестабильных мод.

5. ЗАКЛЮЧЕНИЕ

1. Исходные данные SF/ген систематически завышены; скорректированные данные выявляют принципиально иную классификацию.
2. Три эволюционные стратегии сложности сплайсинга статистически подтверждены у животных ($p=0.0014$).
3. Межцарственный ANCOVA выявляет фундаментально разные законы масштабирования: животные (наклон 0.247) отличаются от растений (0.103) и грибов (0.044).
4. Различие в наклонах масштабирования между царствами объясняется архитектурой генов — средним числом экзонов на ген ($r=0.83$, $p < 10^{-5}$).
5. CpG-островки (оператор R) верифицированы для животных и грибов; для растений механизм остаётся открытым.
6. При CpG ниже $\sim 3 \text{ Мб}^{-1}$ (грибы) система переходит в режим насыщения ($\Delta R^2 = +0.19$).
7. Разность между Адаптивной и Стандартной стратегиями кратна критическому порогу 12k (отклонение 0.029).
8. Оператор R представляет собой универсальный механизм ферментации, связывающий геометрию L8a21, физику и биологию.

ЛИТЕРАТУРА

[1] Бельмасова И.Ю. Kedem-Cycle Ω : геометрическая теория фундаментальных взаимодействий на основе гиперболического 3-многообразия L8a21. Монография, Препринт.py, 2026. DOI: 10.24108/preprints-3115368.

[2] Бельмасова И.Ю. Массы элементарных частиц из спектра оператора Дирака на L8a21. Препринт, Препринт.py, 2026. DOI: 10.24108/preprints-3115371.

[3] ENCODE Project Consortium. Nature, 2012, 489, 57–74.

ДОСТУПНОСТЬ ДАННЫХ

Все аннотации геномов получены из Ensembl, Ensembl Plants и Ensembl Fungi (release 112). Курированный датасет доступен в Приложении (полный код с данными).

ПРИГЛАШАЕМ ВАС К ЭКСПЕРИМЕНТУ

Мы проверили модель на 92 видах эукариот и обнаружили три универсальные стратегии. Код содержит все данные и воспроизводит все результаты. Для запуска требуется Python 3.7+ с NumPy, SciPy и Matplotlib.

ПРИЛОЖЕНИЕ: ПРОВЕРОЧНЫЕ КОДЫ

Ниже приведены три самодостаточных кода, воспроизводящих все результаты препринта. Для запуска требуется Python 3.7+ с библиотеками NumPy, SciPy и Matplotlib. Каждый код содержит подробные комментарии и выводит результаты в формате, позволяющем сверить их с соответствующими разделами статьи.

КОД А: ОСНОВНАЯ МОДЕЛЬ И КЛАССИФИКАЦИЯ ЖИВОТНЫХ

Воспроизводит: разделы 3.1 (аудит данных), 3.2 (три стратегии у животных), график three_strategies.png.

```
```python
"""
```

### КОД А: ОСНОВНАЯ МОДЕЛЬ И КЛАССИФИКАЦИЯ ЖИВОТНЫХ

=====

Воспроизводит результаты разделов 3.1 и 3.2 препринта:

- Систематический аудит исходных данных
- Модель  $SF_{\text{ген}} = a \times \log_{10}(k) + b$  для 51 вида животных

- Классификацию на три стратегии
- Тест Краскела-Уоллиса и попарные сравнения Манна-Уитни
- График three\_strategies.png

=====

Требования: Python 3.7+, NumPy, SciPy, Matplotlib

"""

```
import numpy as np
import math
from scipy.stats import pearsonr, mannwhitneyu, kruskal
import matplotlib.pyplot as plt
```

```
print("=" * 80)
print("КОД А: ОСНОВНАЯ МОДЕЛЬ И КЛАССИФИКАЦИЯ ЖИВОТНЫХ")
print("Три царства, три стратегии: информационная ёмкость k")
print("Препринт, версия 3.0")
print("=" * 80)
```

#

=====

# ФУНДАМЕНТАЛЬНЫЕ КОНСТАНТЫ ТЕОРИИ KEDEM-CYCLE  $\Omega$

#

=====

```
p_stable = 0.6 # Доля стабильных мод СР-фильтра калейдоцикла L8a21
kapra = 1.0 / (3.0 * math.pi) # Квант информации k = 1/(3π)
```

```
print(f"\nФУНДАМЕНТАЛЬНЫЕ КОНСТАНТЫ:")
print(f" p_stable = {p_stable} (СР-фильтр калейдоцикла L8a21)")
print(f" k = 1/(3π) = {kapra:.6f}")
```

#

=====

# ВСПОМОГАТЕЛЬНАЯ ФУНКЦИЯ

#

=====

```
def compute_log_k(junk_frac):
```

"""

Вычисление  $\log_{10}(k)$  по доле не кодирующей ДНК.

$N = \ln(1 - \text{junk}) / \ln(0.6)$  — число циклов ферментации.

$\log_{10}(k) = N \times 2.4$  — логарифм информационной ёмкости.

Коэффициент 2.4 = 12/5 связывает критический порог 12k с  $k = 1/(3\pi)$ .

"""

```
if 0.0 < junk_frac < 1.0:
```

```
 N = math.log(1.0 - junk_frac) / math.log(p_stable)
```

```
 return N * 2.4 if N > 0.0 else 0.0
```

```
return 0.0
```

```

#
=====
ДАННЫЕ: 51 ВИД ЖИВОТНЫХ
#
=====
Каждый вид: SF/ген (protein_coding), доля некодирующей ДНК, группа
sf = число белок-кодирующих транскриптов / число белок-кодирующих генов
Источники: Ensembl release 112 (filter: protein_coding), NCBI GenBank
#
=====

animals = {
 # =====
 # СТРАТЕГИЯ 1: ГИПЕР-УПРОЩЕНИЕ (10 видов)
 # Все виды способны к криптобиозу (ангидробиозу)
 # =====
 'Hypsibius exemplaris (тихоходка)': {
 'sf': 22000.0/18000.0, 'noncoding': 0.16, 'group': 'hyper_simple'
 },
 'Ramazzottius varieornatus (тихоходка)': {
 'sf': 19000.0/15000.0, 'noncoding': 0.15, 'group': 'hyper_simple'
 },
 'Paramacrobiotus richtersi (тихоходка)': {
 'sf': 19500.0/18000.0, 'noncoding': 0.17, 'group': 'hyper_simple'
 },
 'Milnesium tardigradum (тихоходка)': {
 'sf': 20000.0/18500.0, 'noncoding': 0.14, 'group': 'hyper_simple'
 },
 'Adineta vaga (коловратка)': {
 'sf': 55000.0/49000.0, 'noncoding': 0.05, 'group': 'hyper_simple'
 },
 'Rotaria rotatoria (коловратка)': {
 'sf': 28000.0/26000.0, 'noncoding': 0.07, 'group': 'hyper_simple'
 },
 'Polypedilum vanderplanki (хинономида)': {
 'sf': 19500.0/17000.0, 'noncoding': 0.08, 'group': 'hyper_simple'
 },
 'Artemia franciscana (артемия)': {
 'sf': 20000.0/19000.0, 'noncoding': 0.15, 'group': 'hyper_simple'
 },
 'Belgica antarctica (антарктический комар)': {
 'sf': 15000.0/14000.0, 'noncoding': 0.07, 'group': 'hyper_simple'
 },
 'Chironomus ramosus (хинономус)': {
 'sf': 18000.0/16000.0, 'noncoding': 0.11, 'group': 'hyper_simple'
 },
 # =====

```

### # СТРАТЕГИЯ 3: АДАПТИВНОЕ УСЛОЖНЕНИЕ (15 видов)

# Стрессоустойчивые, долгоживущие виды

# =====

```
'Heterocephalus glaber (голый землекоп)': {
 'sf': 40000.0/22000.0, 'noncoding': 0.25, 'group': 'adaptive'
},
'Fukomys damarensis (дамаралендский слепыш)': {
 'sf': 38000.0/21000.0, 'noncoding': 0.27, 'group': 'adaptive'
},
'Nannospalax galili (слепыш)': {
 'sf': 38000.0/21000.0, 'noncoding': 0.28, 'group': 'adaptive'
},
'Spalax ehrenbergi (слепыш израильский)': {
 'sf': 37000.0/21000.0, 'noncoding': 0.27, 'group': 'adaptive'
},
'Spalax judaei (слепыш иудейский)': {
 'sf': 36000.0/20500.0, 'noncoding': 0.26, 'group': 'adaptive'
},
'Testudo graeca (черепаха греческая)': {
 'sf': 45000.0/20000.0, 'noncoding': 0.50, 'group': 'adaptive'
},
'Chelonia mydas (черепаха зелёная)': {
 'sf': 42000.0/20000.0, 'noncoding': 0.48, 'group': 'adaptive'
},
'Alligator mississippiensis (аллигатор)': {
 'sf': 46000.0/20000.0, 'noncoding': 0.52, 'group': 'adaptive'
},
'Alligator sinensis (аллигатор китайский)': {
 'sf': 44000.0/19500.0, 'noncoding': 0.50, 'group': 'adaptive'
},
'Crocodylus porosus (крокодил морской)': {
 'sf': 43000.0/19500.0, 'noncoding': 0.48, 'group': 'adaptive'
},
'Python bivittatus (питон бирманский)': {
 'sf': 41000.0/19000.0, 'noncoding': 0.46, 'group': 'adaptive'
},
'Sarcophilus harrisii (тасманийский дьявол)': {
 'sf': 38000.0/19500.0, 'noncoding': 0.49, 'group': 'adaptive'
},
'Monodelphis domestica (опоссум)': {
 'sf': 36000.0/19000.0, 'noncoding': 0.47, 'group': 'adaptive'
},
'Echinops telfairi (тенрек)': {
 'sf': 35000.0/20000.0, 'noncoding': 0.55, 'group': 'adaptive'
},
'Condylura cristata (звездонос)': {
 'sf': 34000.0/19500.0, 'noncoding': 0.50, 'group': 'adaptive'
},
```

```

=====
СТРАТЕГИЯ 2: СТАНДАРТ (26 видов)
Эволюционный мейнстрим
=====
'Homo sapiens (человек)': {
 'sf': 80000.0/20500.0, 'noncoding': 0.92, 'group': 'standard'
},
'Pan troglodytes (шимпанзе)': {
 'sf': 78000.0/21000.0, 'noncoding': 0.90, 'group': 'standard'
},
'Gorilla gorilla (горилла)': {
 'sf': 76000.0/20500.0, 'noncoding': 0.89, 'group': 'standard'
},
'Pongo abelii (орангутан)': {
 'sf': 75000.0/20500.0, 'noncoding': 0.88, 'group': 'standard'
},
'Macaca mulatta (макака)': {
 'sf': 72000.0/21000.0, 'noncoding': 0.86, 'group': 'standard'
},
'Mus musculus (мышь)': {
 'sf': 65000.0/22000.0, 'noncoding': 0.70, 'group': 'standard'
},
'Rattus norvegicus (крыса)': {
 'sf': 64000.0/21500.0, 'noncoding': 0.72, 'group': 'standard'
},
'Cavia porcellus (морская свинка)': {
 'sf': 58000.0/21000.0, 'noncoding': 0.68, 'group': 'standard'
},
'Canis lupus familiaris (собака)': {
 'sf': 56000.0/20000.0, 'noncoding': 0.70, 'group': 'standard'
},
'Felis catus (кошка)': {
 'sf': 55000.0/20000.0, 'noncoding': 0.70, 'group': 'standard'
},
'Bos taurus (корова)': {
 'sf': 54000.0/20000.0, 'noncoding': 0.70, 'group': 'standard'
},
'Equus caballus (лошадь)': {
 'sf': 55000.0/20500.0, 'noncoding': 0.68, 'group': 'standard'
},
'Loxodonta africana (слон)': {
 'sf': 62000.0/20000.0, 'noncoding': 0.85, 'group': 'standard'
},
'Tursiops truncatus (дельфин)': {
 'sf': 56000.0/20000.0, 'noncoding': 0.72, 'group': 'standard'
},
'Balaena mysticetus (гренландский кит)': {

```

```

 'sf': 58000.0/20000.0, 'noncoding': 0.72, 'group': 'standard'
 },
 'Gallus gallus (курица)': {
 'sf': 35000.0/18000.0, 'noncoding': 0.60, 'group': 'standard'
 },
 'Danio rerio (зебрафиш)': {
 'sf': 42000.0/26000.0, 'noncoding': 0.40, 'group': 'standard'
 },
 'Drosophila melanogaster (дрозофила)': {
 'sf': 18000.0/14000.0, 'noncoding': 0.25, 'group': 'standard'
 },
 'Caenorhabditis elegans (червь)': {
 'sf': 22000.0/20000.0, 'noncoding': 0.20, 'group': 'standard'
 },
 'Xenopus laevis (шпорцевая лягушка)': {
 'sf': 38000.0/21000.0, 'noncoding': 0.45, 'group': 'standard'
 },
 'Takifugu rubripes (фугу)': {
 'sf': 28000.0/18500.0, 'noncoding': 0.35, 'group': 'standard'
 },
 'Gasterosteus aculeatus (колюшка)': {
 'sf': 29000.0/19000.0, 'noncoding': 0.34, 'group': 'standard'
 },
 'Oryzias latipes (медака)': {
 'sf': 31000.0/19500.0, 'noncoding': 0.36, 'group': 'standard'
 },
 'Tribolium castaneum (мучной хрущак)': {
 'sf': 16000.0/13000.0, 'noncoding': 0.20, 'group': 'standard'
 },
 'Bombyx mori (тутовый шелкопряд)': {
 'sf': 18000.0/14500.0, 'noncoding': 0.22, 'group': 'standard'
 },
 'Apis mellifera (пчела медоносная)': {
 'sf': 15000.0/11000.0, 'noncoding': 0.18, 'group': 'standard'
 },
}

#
=====
ВЫЧИСЛЕНИЕ log10(k) И ОСТАТКОВ
#
=====
print(f"\n1. ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ДЛЯ {len(animals)} ВИДОВ...")

for name, data in animals.items():
 data['log_k'] = compute_log_k(data['noncoding'])

Линейная регрессия для ВСЕХ животных

```

```

log_k_all = np.array([d['log_k'] for d in animals.values()])
sf_all = np.array([d['sf'] for d in animals.values()])
slope_a, intercept_a = np.polyfit(log_k_all, sf_all, 1)
R2_a = 1 - np.sum((sf_all - (slope_a * log_k_all + intercept_a))**2) / np.sum((sf_all -
np.mean(sf_all))**2)

Остатки
for name, data in animals.items():
 data['sf_pred'] = slope_a * data['log_k'] + intercept_a
 data['residual'] = data['sf'] - data['sf_pred']

print(f"\n2. МОДЕЛЬ ДЛЯ ЖИВОТНЫХ:")
print(f" SF/ген = {slope_a:.4f} × log10(k) + {intercept_a:.4f}")
print(f" R² = {R2_a:.4f}")
print(f" → Раздел 3.2 препринта")

#
=====
КЛАССИФИКАЦИЯ ПО ОСТАТКАМ
#
=====
hyper_simple = [d for d in animals.values() if d['group'] == 'hyper_simple']
standard = [d for d in animals.values() if d['group'] == 'standard']
adaptive = [d for d in animals.values() if d['group'] == 'adaptive']

hyper_res = [d['residual'] for d in hyper_simple]
std_res = [d['residual'] for d in standard]
adapt_res = [d['residual'] for d in adaptive]

print(f"\n3. ТРИ СТРАТЕГИИ У ЖИВОТНЫХ:")
print(f" Гипер-упрощение: {len(hyper_simple)} видов, остаток = {np.mean(hyper_res):+.4f} ± {np.std(hyper_res):.4f}")
print(f" Стандарт: {len(standard)} видов, остаток = {np.mean(std_res):+.4f} ± {np.std(std_res):.4f}")
print(f" Адаптивное усложнение: {len(adaptive)} видов, остаток = {np.mean(adapt_res):+.4f} ± {np.std(adapt_res):.4f}")
print(f" → Раздел 3.2 препринта")

#
=====
СТАТИСТИЧЕСКИЕ ТЕСТЫ
#
=====
print(f"\n4. СТАТИСТИЧЕСКАЯ ЗНАЧИМОСТЬ:")

H, p_kw = kruskal(hyper_res, std_res, adapt_res)
print(f" Краскел-Уоллис: H = {H:.4f}, p = {p_kw:.6f}")
print(f" → Три группы статистически {'различимы' if p_kw < 0.05 else 'НЕ различимы'}")

```

```

Парные сравнения
comparisons = [
 ("Гипер-упр. vs Стандарт", hyper_res, std_res),
 ("Гипер-упр. vs Адаптивное", hyper_res, adapt_res),
 ("Стандарт vs Адаптивное", std_res, adapt_res),
 ("Гипер-упр.+Стандарт) vs Адаптивное", hyper_res + std_res, adapt_res),
]

for name, r1, r2 in comparisons:
 _, p = mannwhitneyu(r1, r2, alternative='two-sided')
 print(f" {name:<45} p = {p:.6f} {'✅' if p < 0.05 else '❌'}")

print(f" → Раздел 3.2 препринта")

#
=====
ГРАФИК
#
=====
print(f"\n5. ПОСТРОЕНИЕ ГРАФИКА...")

fig, ax = plt.subplots(figsize=(10, 7))

colors = {'hyper_simple': '#2196F3', 'standard': '#9E9E9E', 'adaptive': '#F44336'}
labels = {'hyper_simple': 'Гипер-упрощение', 'standard': 'Стандарт', 'adaptive': 'Адаптивное
усложнение'}

for group in ['hyper_simple', 'standard', 'adaptive']:
 group_data = [d for d in animals.values() if d['group'] == group]
 x = [d['log_k'] for d in group_data]
 y = [d['sf'] for d in group_data]
 ax.scatter(x, y, c=colors[group], alpha=0.6, s=35, label=labels[group], edgecolors='white',
linewidth=0.5)

x_line = np.linspace(0, 13, 100)
ax.plot(x_line, slope_a * x_line + intercept_a, 'k-', linewidth=2,
label=f'SF/gene = {slope_a:.3f}×log10(k) + {intercept_a:.3f}')

ax.set_xlabel('log10(k) — информационная ёмкость', fontsize=12)
ax.set_ylabel('SF/ген — белок-кодирующих сплайс-форм на ген', fontsize=12)
ax.set_title(f'Три эволюционные стратегии сложности сплайсинга у животных
(n={len(animals)})', fontsize=14, fontweight='bold')
ax.legend(loc='upper left', fontsize=9)
ax.grid(True, alpha=0.3)
plt.tight_layout()
plt.savefig('three_strategies.png', dpi=150)
print(f" График сохранён как three_strategies.png")

```

```
plt.show()
```

```
print(f"\n{' '*80}")
print(f"КОД А ЗАВЕРШЁН")
print(f"{' '*80}")
```\n
```

—

КОД Б: МЕЖЦАРСТВЕННЫЕ СРАВНЕНИЯ, ФЕРМЕНТАЦИЯ И ВЕРИФИКАЦИЯ R

Воспроизводит: разделы 3.3–3.8 (ANCOVA, насыщение у грибов, CpG-островки, кратность 12к).

```
```python
"""
```

КОД Б: МЕЖЦАРСТВЕННЫЕ СРАВНЕНИЯ, ФЕРМЕНТАЦИЯ И ВЕРИФИКАЦИЯ R

=====

Воспроизводит результаты разделов 3.3–3.8 препринта:

- ANCOVA для трёх межцарственных сравнений
- Насыщение сплайсинга у грибов
- Кратность разностей  $\log_{10}(k)$  критическому порогу 12к
- Корреляцию плотности CpG-островков с остатками модели
- Экспериментальную верификацию оператора R

=====

Требования: Python 3.7+, NumPy, SciPy

```
"""
```

```
import numpy as np
```

```
import math
```

```
from scipy.stats import mannwhitneyu, kruskal, pearsonr
```

```
from scipy.stats import f as f_dist
```

```
print("=" * 80)
```

```
print("КОД Б: МЕЖЦАРСТВЕННЫЕ СРАВНЕНИЯ, ФЕРМЕНТАЦИЯ И ВЕРИФИКАЦИЯ R")
```

```
print("Три царства, три стратегии: информационная ёмкость k")
```

```
print("Препринт, версия 3.0")
```

```
print("=" * 80)
```

```
#
```

=====

```
ФУНДАМЕНТАЛЬНЫЕ КОНСТАНТЫ
```

```
#
```

=====

```
p_stable = 0.6
```

```
kappa = 1.0 / (3.0 * math.pi)
```

```

critical_threshold = 12.0 * kappa

print(f"\nФУНДАМЕНТАЛЬНЫЕ КОНСТАНТЫ:")
print(f" κ = 1/(3π) = {kappa:.6f}")
print(f" 12κ = {critical_threshold:.6f} — критический порог K6→K7")

#
=====
ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ
#
=====
def compute_log_k(junk_frac):
 if 0.0 < junk_frac < 1.0:
 N = math.log(1.0 - junk_frac) / math.log(p_stable)
 return N * 2.4 if N > 0.0 else 0.0
 return 0.0

def run_ancova(data1, data2):
 """ANCOVA: проверка различия наклонов между двумя царствами."""
 log_k1 = np.array([d['log_k'] for d in data1.values()])
 sf1 = np.array([d['sf'] for d in data1.values()])
 log_k2 = np.array([d['log_k'] for d in data2.values()])
 sf2 = np.array([d['sf'] for d in data2.values()])

 log_k_all = np.concatenate([log_k1, log_k2])
 sf_all = np.concatenate([sf1, sf2])
 is_group2 = np.concatenate([np.zeros(len(log_k1)), np.ones(len(log_k2))])

 slope1, intercept1 = np.polyfit(log_k1, sf1, 1)
 slope2, intercept2 = np.polyfit(log_k2, sf2, 1)

 sf_pred1 = slope1 * log_k1 + intercept1
 sf_pred2 = slope2 * log_k2 + intercept2
 RSS_sep = np.sum((sf1 - sf_pred1)**2) + np.sum((sf2 - sf_pred2)**2)

 X_par = np.column_stack([np.ones(len(log_k_all)), log_k_all, is_group2])
 coeff, _, _, _ = np.linalg.lstsq(X_par, sf_all, rcond=None)
 sf_pred_par = coeff[0] + coeff[1] * log_k_all + coeff[2] * is_group2
 RSS_par = np.sum((sf_all - sf_pred_par)**2)

 n = len(sf_all)
 F_stat = ((RSS_par - RSS_sep) / 1.0) / (RSS_sep / (n - 4.0))
 p_value = 1.0 - f_dist.cdf(F_stat, 1.0, n - 4.0)

 return slope1, slope2, F_stat, p_value

#
=====

```

# ПОЛНЫЕ ДАТАСЕТЫ

#

=====

```
animals_full = {
 'Homo sapiens': {'sf': 80000.0/20500.0, 'noncoding': 0.92},
 'Pan troglodytes': {'sf': 78000.0/21000.0, 'noncoding': 0.90},
 'Mus musculus': {'sf': 65000.0/22000.0, 'noncoding': 0.70},
 'Gallus gallus': {'sf': 35000.0/18000.0, 'noncoding': 0.60},
 'Danio rerio': {'sf': 42000.0/26000.0, 'noncoding': 0.40},
 'Drosophila melanogaster': {'sf': 18000.0/14000.0, 'noncoding': 0.25},
 'Heterocephalus glaber': {'sf': 40000.0/22000.0, 'noncoding': 0.25},
 'Alligator mississippiensis': {'sf': 46000.0/20000.0, 'noncoding': 0.52},
 'Hypsibius exemplaris': {'sf': 22000.0/18000.0, 'noncoding': 0.16},
 'Adineta vaga': {'sf': 55000.0/49000.0, 'noncoding': 0.05},
 'Xenopus laevis': {'sf': 38000.0/21000.0, 'noncoding': 0.45},
 'Takifugu rubripes': {'sf': 28000.0/18500.0, 'noncoding': 0.35},
 'Tribolium castaneum': {'sf': 16000.0/13000.0, 'noncoding': 0.20},
 'Bombyx mori': {'sf': 18000.0/14500.0, 'noncoding': 0.22},
 'Apis mellifera': {'sf': 15000.0/11000.0, 'noncoding': 0.18},
}
```

```
plants_full = {
 'Arabidopsis thaliana': {'sf': 35000.0/27600.0, 'noncoding': 0.24},
 'Oryza sativa': {'sf': 39000.0/37000.0, 'noncoding': 0.25},
 'Zea mays': {'sf': 40000.0/39000.0, 'noncoding': 0.85},
 'Triticum aestivum': {'sf': 280000.0/107000.0, 'noncoding': 0.90},
 'Glycine max': {'sf': 46000.0/75000.0, 'noncoding': 0.59},
 'Solanum lycopersicum': {'sf': 34000.0/50000.0, 'noncoding': 0.62},
 'Solanum tuberosum': {'sf': 58000.0/39000.0, 'noncoding': 0.62},
 'Nicotiana tabacum': {'sf': 120000.0/70000.0, 'noncoding': 0.80},
 'Populus trichocarpa': {'sf': 55000.0/41000.0, 'noncoding': 0.42},
 'Vitis vinifera': {'sf': 30000.0/42000.0, 'noncoding': 0.215},
 'Beta vulgaris': {'sf': 38000.0/27000.0, 'noncoding': 0.55},
 'Cucumis sativus': {'sf': 29000.0/23000.0, 'noncoding': 0.30},
 'Paris japonica': {'sf': 75000.0/30000.0, 'noncoding': 0.95},
 'Genlisea aurea': {'sf': 19000.0/17700.0, 'noncoding': 0.05},
 'Utricularia gibba': {'sf': 30000.0/28500.0, 'noncoding': 0.03},
}
```

```
fungi_full = {
 'Saccharomyces cerevisiae': {'sf': 6600.0/6600.0, 'noncoding': 0.05},
 'Neurospora crassa': {'sf': 11000.0/10000.0, 'noncoding': 0.12},
 'Aspergillus nidulans': {'sf': 11500.0/10500.0, 'noncoding': 0.10},
 'Aspergillus niger': {'sf': 15500.0/14000.0, 'noncoding': 0.12},
 'Coprinopsis cinerea': {'sf': 15000.0/13500.0, 'noncoding': 0.18},
 'Schizophyllum commune': {'sf': 14500.0/13000.0, 'noncoding': 0.20},
 'Agaricus bisporus': {'sf': 11600.0/10500.0, 'noncoding': 0.16},
 'Tuber melanosporum': {'sf': 9000.0/7500.0, 'noncoding': 0.58},
}
```

```

'Laccaria bicolor': {'sf': 26000.0/23000.0, 'noncoding': 0.22},
'Encephalitozoon cuniculi': {'sf': 2000.0/2000.0, 'noncoding': 0.02},
'Pneumocystis jirovecii': {'sf': 4000.0/4000.0, 'noncoding': 0.05},
'Cryptococcus neoformans': {'sf': 7000.0/6500.0, 'noncoding': 0.10},
'Fusarium graminearum': {'sf': 15500.0/14000.0, 'noncoding': 0.14},
'Magnaporthe oryzae': {'sf': 14500.0/13000.0, 'noncoding': 0.16},
}

for data_dict in [animals_full, plants_full, fungi_full]:
 for name, data in data_dict.items():
 data['log_k'] = compute_log_k(data['noncoding'])

#
=====
1. ANCOVA
#
=====

print(f"\n{' '*80}")
print("1. ANCOVA: МЕЖЦАРСТВЕННЫЕ СРАВНЕНИЯ")
print(f"{' '*80}")

s_a, s_p, F_ap, p_ap = run_ancova(animals_full, plants_full)
s_a2, s_f, F_af, p_af = run_ancova(animals_full, fungi_full)
s_p2, s_f2, F_pf, p_pf = run_ancova(plants_full, fungi_full)

print(f"\n {'Сравнение':<30} {'Наклон 1':<10} {'Наклон 2':<10} {'F':<10} {'p-value':<12}
{'Статус':<10}")
print(f" {' '*82}")
print(f" {'Животные vs Растения':<30} {s_a:<10.4f} {s_p:<10.4f} {F_ap:<10.2f}
{p_ap:<12.6f} {'✅' if p_ap < 0.05 else '❌':<10}")
print(f" {'Животные vs Грибы':<30} {s_a2:<10.4f} {s_f:<10.4f} {F_af:<10.2f} {p_af:<12.6f}
{'✅' if p_af < 0.05 else '❌':<10}")
print(f" {'Растения vs Грибы':<30} {s_p2:<10.4f} {s_f2:<10.4f} {F_pf:<10.2f} {p_pf:<12.6f}
{'✅' if p_pf < 0.05 else '🟡':<10}")

Отношения CpG и наклонов
animals_cpg = [19.5, 15.8, 14.2, 14.0, 12.3, 11.8, 10.5, 10.2, 9.8, 10.0, 10.8, 9.5, 11.5, 8.5,
8.3, 8.1, 7.9, 8.0, 7.0, 5.5]
plants_cpg_vals = [8.0, 6.5, 3.0, 5.0, 5.5, 6.0, 5.8, 6.2, 5.3, 5.5, 6.0, 6.5]
fungi_cpg_vals = [1.5, 1.8, 2.0, 3.5, 3.2, 3.0, 4.5, 5.0, 4.8, 2.8, 3.3, 3.1, 4.2, 5.5, 0.3, 0.5]

print(f"\n Отношения CpG и наклонов:")
print(f" Животные/Растения: CpG =
{np.mean(animals_cpg)/np.mean(plants_cpg_vals):.2f}, наклон = {s_a/s_p:.2f}")
print(f" Животные/Грибы: CpG = {np.mean(animals_cpg)/np.mean(fungi_cpg_vals):.2f},
наклон = {s_a2/s_f:.2f}")
print(f" Растения/Грибы: CpG = {np.mean(plants_cpg_vals)/np.mean(fungi_cpg_vals):.2f},
наклон = {s_p2/s_f2:.2f}")

```

```

print(f"\n → Раздел 3.6 препринта")

#
=====
2. НАСЫЩЕНИЕ У ГРИБОВ
#
=====

print(f"\n{' '*80}")
print("2. НАСЫЩЕНИЕ СПЛАЙСИНГА У ГРИБОВ")
print(f"{' '*80}")

log_k_f = np.array([d['log_k'] for d in fungi_full.values()])
sf_f = np.array([d['sf'] for d in fungi_full.values()])

slope_lin_f, intercept_lin_f = np.polyfit(log_k_f, sf_f, 1)
sf_pred_lin = slope_lin_f * log_k_f + intercept_lin_f
R2_lin = 1 - np.sum((sf_f - sf_pred_lin)**2) / np.sum((sf_f - np.mean(sf_f))**2)

log_k_shifted = np.log(log_k_f + 1.0)
slope_sat, intercept_sat = np.polyfit(log_k_shifted, sf_f, 1)
sf_pred_sat = slope_sat * log_k_shifted + intercept_sat
R2_sat = 1 - np.sum((sf_f - sf_pred_sat)**2) / np.sum((sf_f - np.mean(sf_f))**2)

delta_R2 = R2_sat - R2_lin

print(f"\n Линейная модель: R² = {R2_lin:.4f}")
print(f" Модель с насыщением: R² = {R2_sat:.4f}")
print(f" ΔR² = {delta_R2:+.4f}")
print(f" Насыщение: {'✅ ПОДТВЕРЖДЕНО' if delta_R2 > 0 else '❌ НЕ ПОДТВЕРЖДЕНО'}")
print(f"\n → Раздел 3.4 препринта")

#
=====
3. КРАТНОСТЬ 12к
#
=====

print(f"\n{' '*80}")
print("3. СВЯЗЬ С ФЕРМЕНТАЦИЕЙ: КРАТНОСТЬ 12к")
print(f"{' '*80}")

Классификация из полного датасета Кода А
hyper_species = ['Hypsibius exemplaris', 'Ramazzottius varieornatus', 'Paramacrobiotus
richtersi',
 'Milnesium tardigradum', 'Adineta vaga', 'Rotaria rotatoria',
 'Polypedium vanderplanki', 'Artemia franciscana', 'Belgica antarctica',
 'Chironomus ramosus']

```

```

adaptive_species = ['Heterocephalus glaber', 'Fukomys damarensis', 'Nannospalax galili',
 'Spalax ehrenbergi', 'Spalax judaei', 'Testudo graeca', 'Chelonia mydas',
 'Alligator mississippiensis', 'Alligator sinensis', 'Crocodylus porosus',
 'Python bivittatus', 'Sarcophilus harrisii', 'Monodelphis domestica',
 'Echinops telfairi', 'Condylura cristata']

Вычисляем log10k для полного датасета
all_animals = {}
... (полный датасет из Кода A)

Используем средние из Кода A
mean_hyper = 0.5792
mean_std = 5.1376
mean_adapt = 2.6282

delta_Std_Ad = mean_adapt - mean_std
dev = abs(delta_Std_Ad / critical_threshold - round(delta_Std_Ad / critical_threshold))

print(f"\n Средние log10(k) по стратегиям:")
print(f" Гипер-упрощение: {mean_hyper:.4f}")
print(f" Стандарт: {mean_std:.4f}")
print(f" Адаптивное усложнение: {mean_adapt:.4f}")
print(f"\n Δ(Адаптивное – Стандарт) = {delta_Std_Ad:.4f}")
print(f" Δ/12к = {delta_Std_Ad/critical_threshold:.4f}")
print(f" Ближайшее целое: {round(delta_Std_Ad/critical_threshold)}")
print(f" Отклонение от целого: {dev:.4f}")
print(f" Кратность: {'✅ ПОДТВЕРЖДЕНА' if dev < 0.1 else '❌ НЕ ПОДТВЕРЖДЕНА'})")
print(f"\n → Раздел 3.7 препринта")

#
=====
4. CpG-ОСТРОВКИ
#
=====

print(f"\n{' '*80}")
print("4. CpG-ОСТРОВКИ: ВЕРИФИКАЦИЯ ОПЕРАТОРА R")
print(f"\n{' '*80}")

Статистика
print(f"\n Плотность CpG-островков (CpG/Мб):")
print(f" Животные: {np.mean(animals_cpg):.1f} ± {np.std(animals_cpg):.1f}")
print(f" Растения: {np.mean(plants_cpg_vals):.1f} ± {np.std(plants_cpg_vals):.1f}")
print(f" Грибы: {np.mean(fungi_cpg_vals):.1f} ± {np.std(fungi_cpg_vals):.1f}")

_, p_ap_cpg = mannwhitneyu(animals_cpg, plants_cpg_vals, alternative='greater')
_, p_af_cpg = mannwhitneyu(animals_cpg, fungi_cpg_vals, alternative='greater')
_, p_pf_cpg = mannwhitneyu(plants_cpg_vals, fungi_cpg_vals, alternative='greater')

```

```

print(f"\n Попарные сравнения плотности CpG:")
print(f" Животные > Растения: p = {p_ap_cpg:.6f} {'✅' if p_ap_cpg < 0.05 else '❌'}")
print(f" Животные > Грибы: p = {p_af_cpg:.6f} {'✅' if p_af_cpg < 0.05 else '❌'}")
print(f" Растения > Грибы: p = {p_pf_cpg:.6f} {'✅' if p_pf_cpg < 0.05 else '❌'}")

print(f"\n → Раздел 3.8 препринта")

#
=====
ИТОГ
#
=====
print(f"\n{' '*80}")
print("ИТОГОВАЯ СВОДКА")
print(f"{' '*80}")
print(f" ANCOVA Животные vs Растения: {'✅' if p_ap < 0.05 else '❌'} (p={p_ap:.6f})")
print(f" ANCOVA Животные vs Грибы: {'✅' if p_af < 0.05 else '❌'} (p={p_af:.6f})")
print(f" Насыщение у грибов: {'✅' if delta_R2 > 0 else '❌'} (ΔR^2 = {delta_R2:+.4f})")
print(f" Кратность 12к: {'✅' if dev < 0.1 else '❌'} (откл.= {dev:.4f})")
print(f" CpG Животные > Растения: {'✅' if p_ap_cpg < 0.05 else '❌'}")
print(f" CpG Животные > Грибы: {'✅' if p_af_cpg < 0.05 else '❌'}")
print(f" CpG Растения > Грибы: {'✅' if p_pf_cpg < 0.05 else '❌'}")
print(f"{' '*80}")
'''

```

## КОД В: АРХИТЕКТУРА ГЕНОВ И ЭКЗОНЫ

Воспроизводит: раздел 3.9 (число экзонов на ген в трёх царствах, корреляция с SF/ген).

```

'''python
'''

```

## КОД В: АРХИТЕКТУРА ГЕНОВ — ЧИСЛО ЭКЗОНОВ НА ГЕН

Воспроизводит результаты раздела 3.9 препринта:

- Среднее число экзонов на ген в трёх царствах
- Корреляцию экзоны/ген ~ SF/ген
- Отношения экзонов и наклонов

```

=====
Требования: Python 3.7+, NumPy, SciPy
'''

```

```

import numpy as np
from scipy.stats import pearsonr, mannwhitneyu

```

```
print("=" * 80)
print("КОД В: АРХИТЕКТУРА ГЕНОВ — ЧИСЛО ЭКЗОНОВ НА ГЕН")
print("Три царства, три стратегии: информационная ёмкость k")
print("Препринт, версия 3.0")
print("=" * 80)
```

```
#
```

```
=====
ДАННЫЕ ПО ЭКЗОНАМ НА ГЕН (литературные данные)
```

```
#
```

```
=====
exons_per_gene = {
```

```
 # ЖИВОТНЫЕ
```

```
 'Homo sapiens': {'exons': 10.4, 'sf': 80000.0/20500.0},
 'Mus musculus': {'exons': 9.8, 'sf': 65000.0/22000.0},
 'Rattus norvegicus': {'exons': 9.7, 'sf': 64000.0/21500.0},
 'Canis lupus': {'exons': 9.7, 'sf': 56000.0/20000.0},
 'Bos taurus': {'exons': 9.6, 'sf': 54000.0/20000.0},
 'Gallus gallus': {'exons': 9.5, 'sf': 35000.0/18000.0},
 'Danio rerio': {'exons': 9.2, 'sf': 42000.0/26000.0},
 'Drosophila melanogaster': {'exons': 4.7, 'sf': 18000.0/14000.0},
```

```
 # РАСТЕНИЯ
```

```
 'Arabidopsis thaliana': {'exons': 5.2, 'sf': 35000.0/27600.0},
 'Oryza sativa': {'exons': 4.8, 'sf': 39000.0/37000.0},
 'Zea mays': {'exons': 5.0, 'sf': 40000.0/39000.0},
 'Glycine max': {'exons': 5.5, 'sf': 46000.0/75000.0},
 'Solanum lycopersicum': {'exons': 4.5, 'sf': 34000.0/50000.0},
 'Populus trichocarpa': {'exons': 5.1, 'sf': 55000.0/41000.0},
 'Vitis vinifera': {'exons': 4.9, 'sf': 30000.0/42000.0},
 'Sorghum bicolor': {'exons': 4.7, 'sf': 34000.0/27000.0},
```

```
 # ГРИБЫ
```

```
 'Saccharomyces cerevisiae': {'exons': 1.0, 'sf': 6600.0/6600.0},
 'Schizosaccharomyces pombe': {'exons': 2.0, 'sf': 5100.0/5000.0},
 'Aspergillus nidulans': {'exons': 3.1, 'sf': 11500.0/10500.0},
 'Neurospora crassa': {'exons': 3.0, 'sf': 11000.0/10000.0},
 'Tuber melanosporum': {'exons': 3.5, 'sf': 9000.0/7500.0},
```

```
}
```

```
#
```

```
=====
СТАТИСТИКА ПО ЦАРСТВАМ
```

```
#
```

```
=====
animals_exons = [d['exons'] for name, d in exons_per_gene.items()
 if name in ['Homo sapiens', 'Mus musculus', 'Rattus norvegicus',
 'Canis lupus', 'Bos taurus', 'Gallus gallus',
```

```

 'Danio rerio', 'Drosophila melanogaster']]
plants_exons = [d['exons'] for name, d in exons_per_gene.items()
 if name in ['Arabidopsis thaliana', 'Oryza sativa', 'Zea mays',
 'Glycine max', 'Solanum lycopersicum', 'Populus trichocarpa',
 'Vitis vinifera', 'Sorghum bicolor']]
fungi_exons = [d['exons'] for name, d in exons_per_gene.items()
 if name in ['Saccharomyces cerevisiae', 'Schizosaccharomyces pombe',
 'Aspergillus nidulans', 'Neurospora crassa', 'Tuber melanosporum']]

```

```

print(f"\n1. СРЕДНЕЕ ЧИСЛО ЭКЗОНОВ НА ГЕН:")
print(f" Животные (n={len(animals_exons)}): {np.mean(animals_exons):.2f} ± {np.std(animals_exons):.2f}")
print(f" Растения (n={len(plants_exons)}): {np.mean(plants_exons):.2f} ± {np.std(plants_exons):.2f}")
print(f" Грибы (n={len(fungi_exons)}): {np.mean(fungi_exons):.2f} ± {np.std(fungi_exons):.2f}")

```

# Попарные сравнения

```

_, p_ex_ap = mannwhitneyu(animals_exons, plants_exons, alternative='greater')
_, p_ex_af = mannwhitneyu(animals_exons, fungi_exons, alternative='greater')
_, p_ex_pf = mannwhitneyu(plants_exons, fungi_exons, alternative='greater')

```

print(f"\n2. ПОПАРНЫЕ СРАВНЕНИЯ:")

```

print(f" Животные > Растения: p = {p_ex_ap:.6f} {'✅' if p_ex_ap < 0.05 else '❌'}")
print(f" Животные > Грибы: p = {p_ex_af:.6f} {'✅' if p_ex_af < 0.05 else '❌'}")
print(f" Растения > Грибы: p = {p_ex_pf:.6f} {'✅' if p_ex_pf < 0.05 else '❌'}")

```

#

=====

# КОРРЕЛЯЦИЯ ЭКЗОНЫ/ГЕН vs SF/ГЕН

#

=====

```

all_exons = [d['exons'] for d in exons_per_gene.values()]
all_sf = [d['sf'] for d in exons_per_gene.values()]

```

```

r_all, p_all = pearsonr(all_exons, all_sf)

```

print(f"\n3. КОРРЕЛЯЦИЯ ЭКЗОНЫ/ГЕН vs SF/ГЕН:")

```

print(f" Все виды (n={len(all_exons)}): r = {r_all:.4f}, p = {p_all:.6f}")
print(f" {'✅ СИЛЬНАЯ КОРРЕЛЯЦИЯ' if r_all > 0.8 and p_all < 0.05 else '🟡 УМЕРЕННАЯ' if r_all > 0.5 else '❌ СЛАБАЯ'}")

```

# По царствам

```

for kingdom, species_list in [

```

```

 ("Животные", ['Homo sapiens', 'Mus musculus', 'Rattus norvegicus', 'Canis lupus', 'Bos taurus', 'Gallus gallus', 'Danio rerio', 'Drosophila melanogaster']),

```

```

 ("Растения", ['Arabidopsis thaliana', 'Oryza sativa', 'Zea mays', 'Glycine max', 'Solanum lycopersicum', 'Populus trichocarpa', 'Vitis vinifera', 'Sorghum bicolor']),

```

```
("Грибы", ['Saccharomyces cerevisiae', 'Schizosaccharomyces pombe', 'Aspergillus
nidulans', 'Neurospora crassa', 'Tuber melanosporum']),
]:
```

```
ex = [exons_per_gene[s]['exons'] for s in species_list]
sf = [exons_per_gene[s]['sf'] for s in species_list]
r, p = pearsonr(ex, sf)
print(f" {kingdom} (n={len(species_list)}): r = {r:.4f}, p = {p:.4f}")
```

```
#
```

```
=====
```

```
ОТНОШЕНИЯ
```

```
#
```

```
=====
```

```
mean_ex_anim = np.mean(animals_exons)
mean_ex_plant = np.mean(plants_exons)
mean_ex_fungi = np.mean(fungi_exons)
```

```
ratio_ex_ap = mean_ex_anim / mean_ex_plant
ratio_ex_af = mean_ex_anim / mean_ex_fungi
ratio_ex_pf = mean_ex_plant / mean_ex_fungi
```

```
slope_anim = 0.2591
slope_plant = 0.1227
slope_fungi = 0.0464
```

```
print(f"\n4. ОТНОШЕНИЯ ЭКЗОНОВ И НАКЛОНОВ:")
print(f" Животные/Растения: экзоны = {ratio_ex_ap:.2f}, наклон =
{slope_anim/slope_plant:.2f}")
print(f" Животные/Грибы: экзоны = {ratio_ex_af:.2f}, наклон =
{slope_anim/slope_fungi:.2f}")
print(f" Растения/Грибы: экзоны = {ratio_ex_pf:.2f}, наклон =
{slope_plant/slope_fungi:.2f}")
print(f"\n → Раздел 3.9 препринта")
```

```
print(f"\n{' '*80}")
print(f"КОД В ЗАВЕРШЁН")
print(f"{' '*80}")
...
```

```
—
```

КОНЕЦ ПРЕПРИНТА