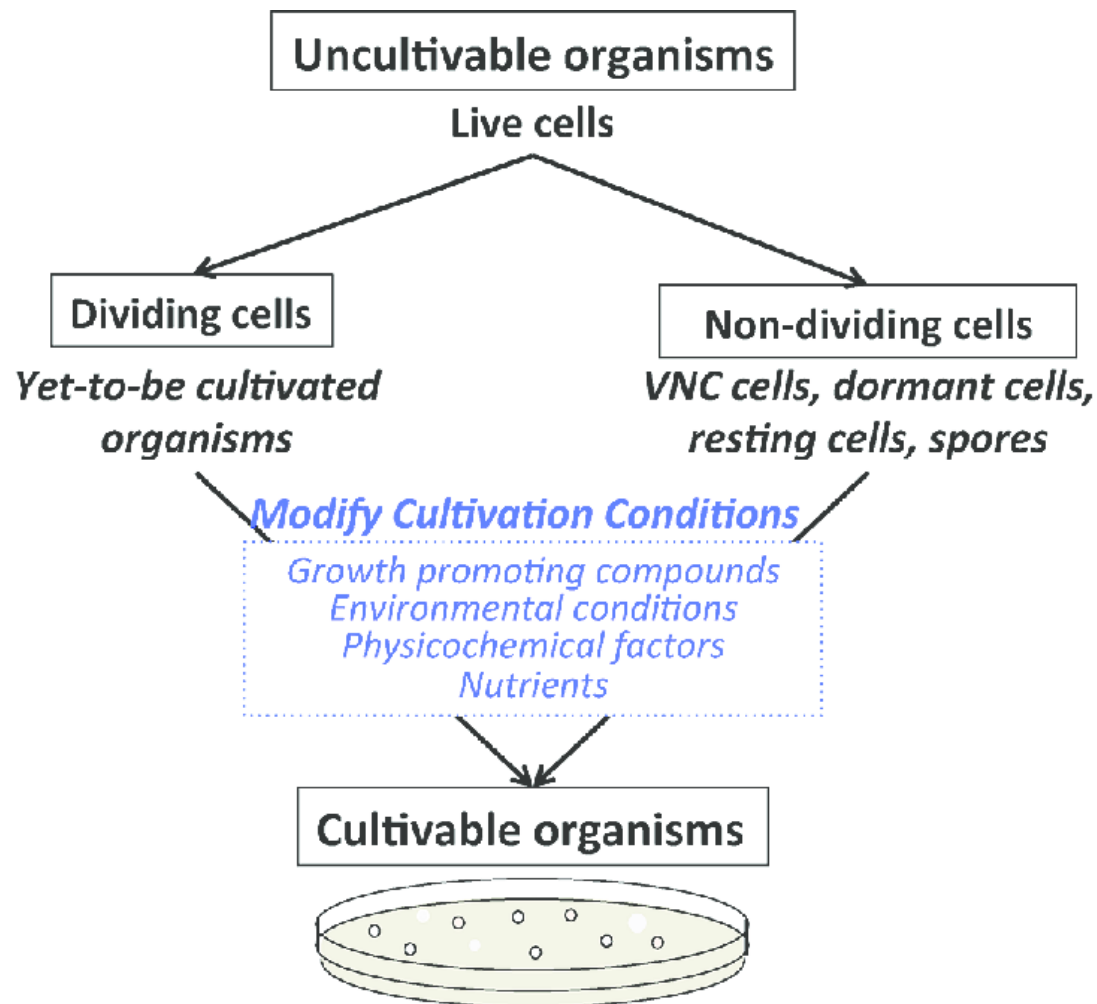
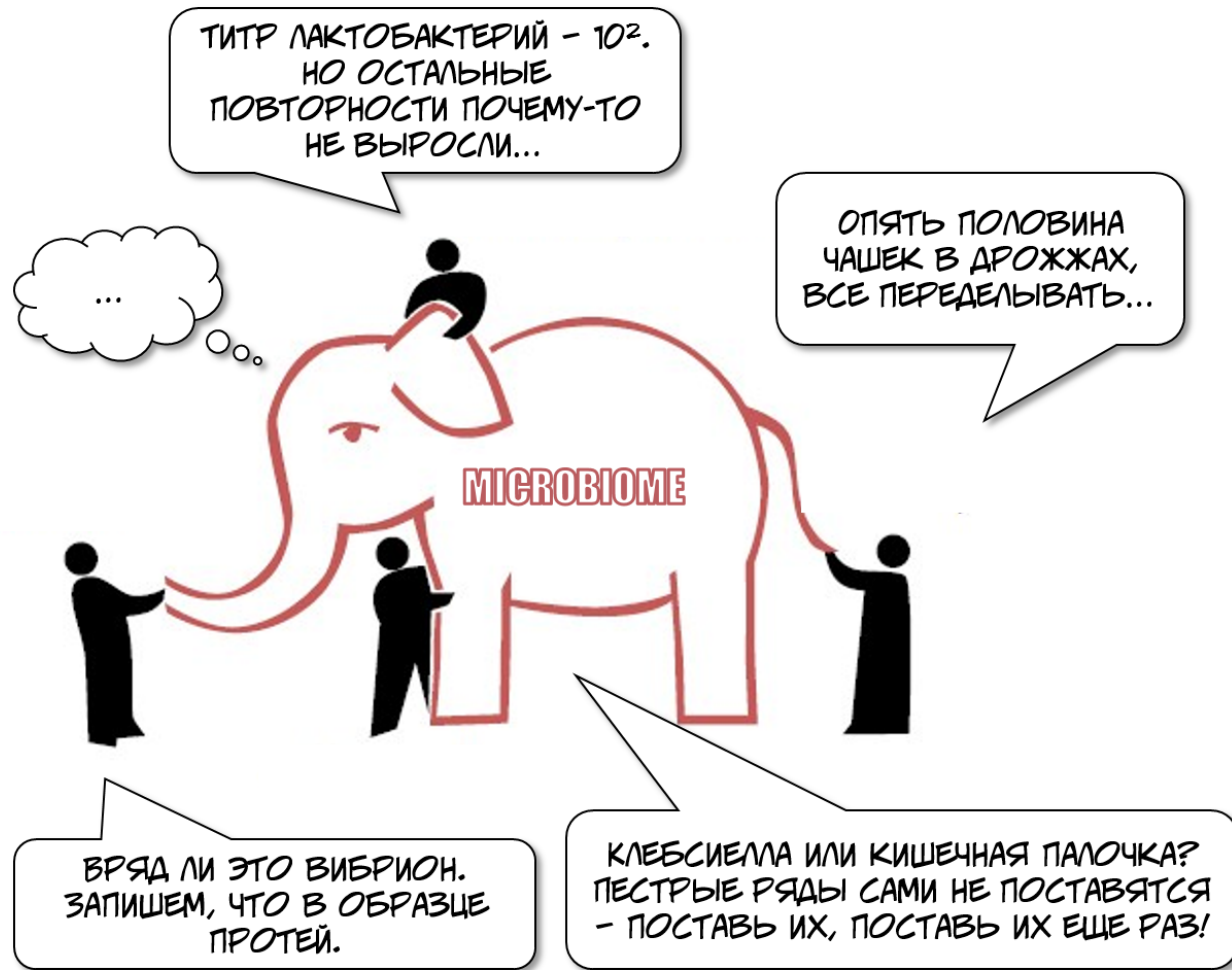


Методы идентификации бактерий

- Морфологические
- Тинкториальные
- Культуральные
- Физиолого-биохимические
- Серологические признаки

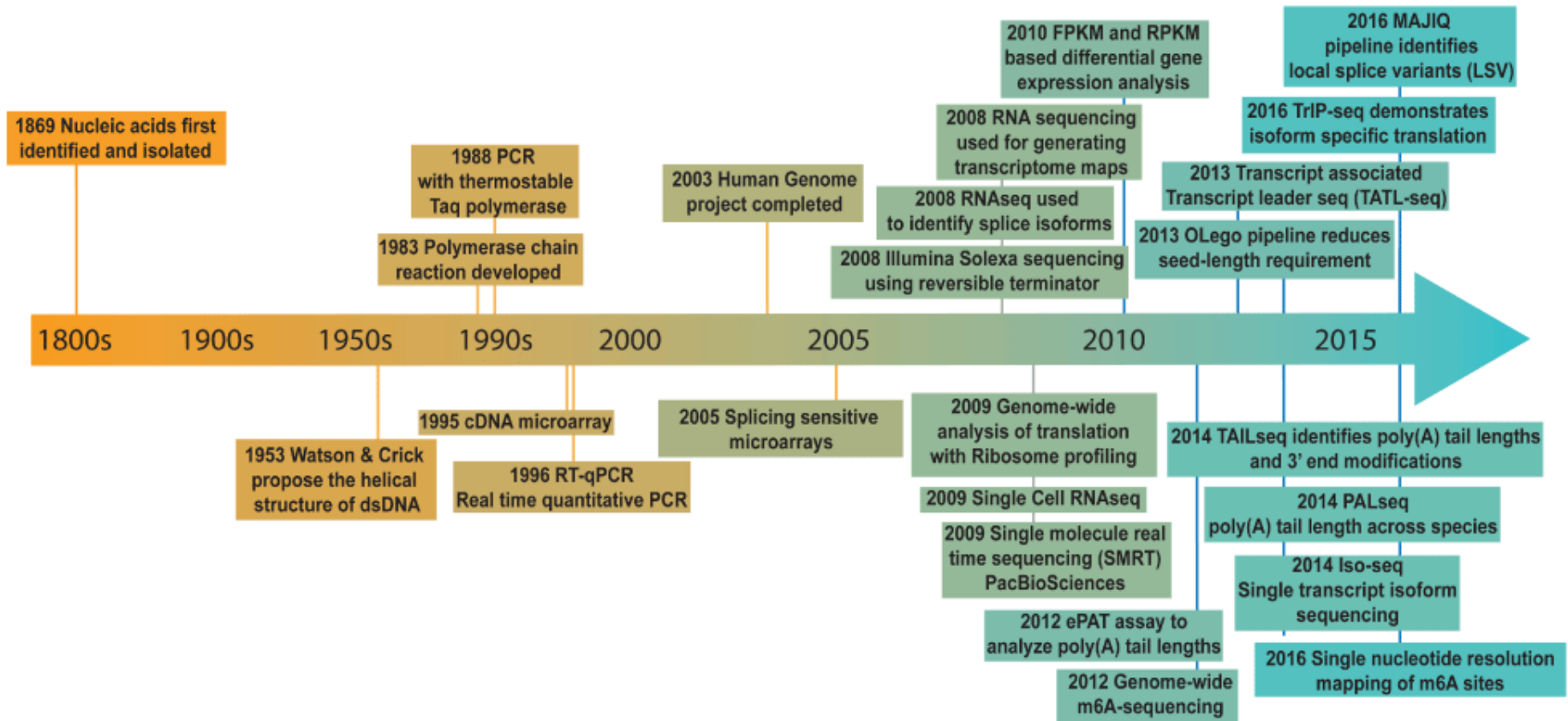
Проблема некультивируемых форм



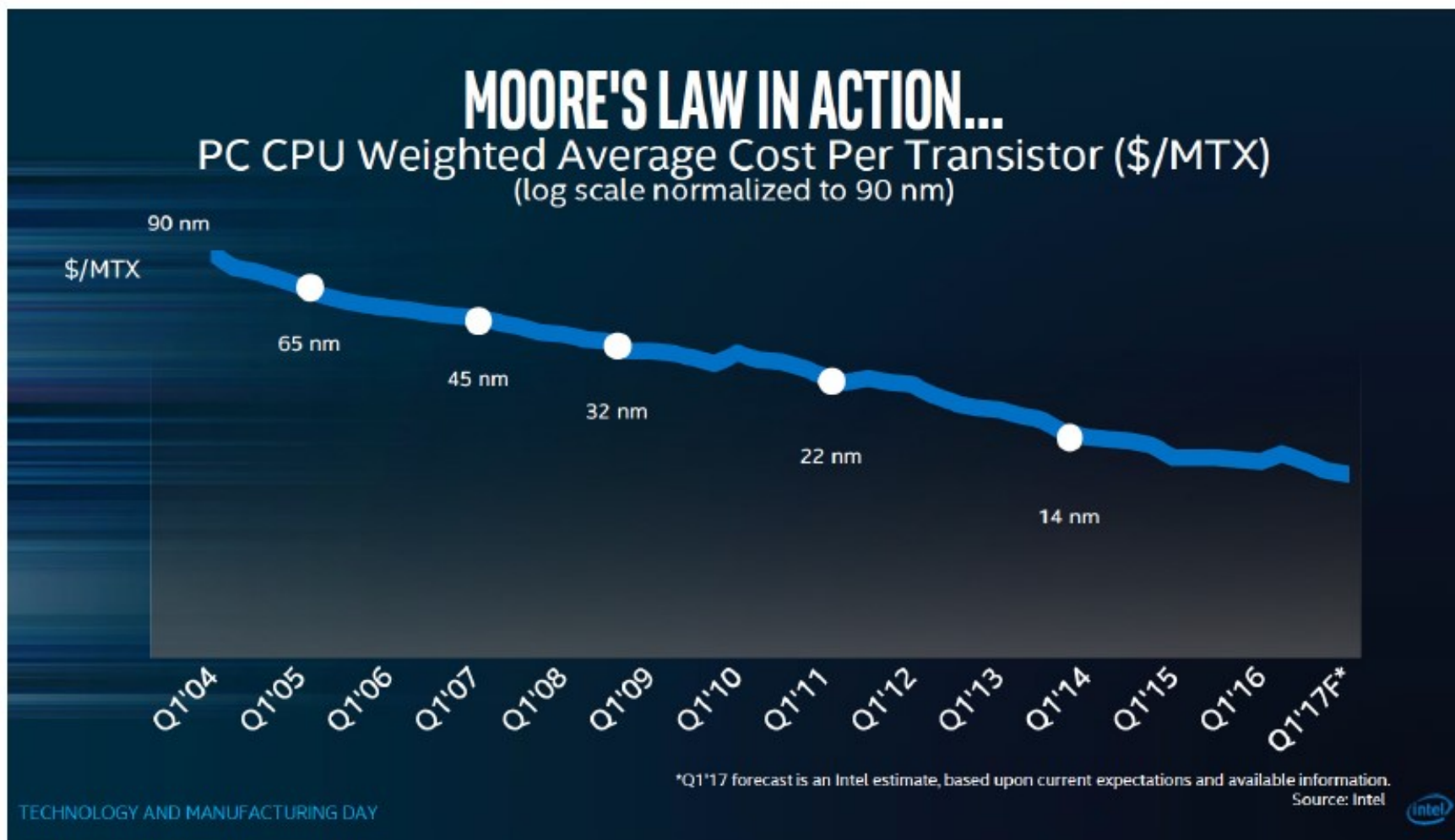


Классические методы оценки бактериальных популяций предполагают большое число рутинных операций

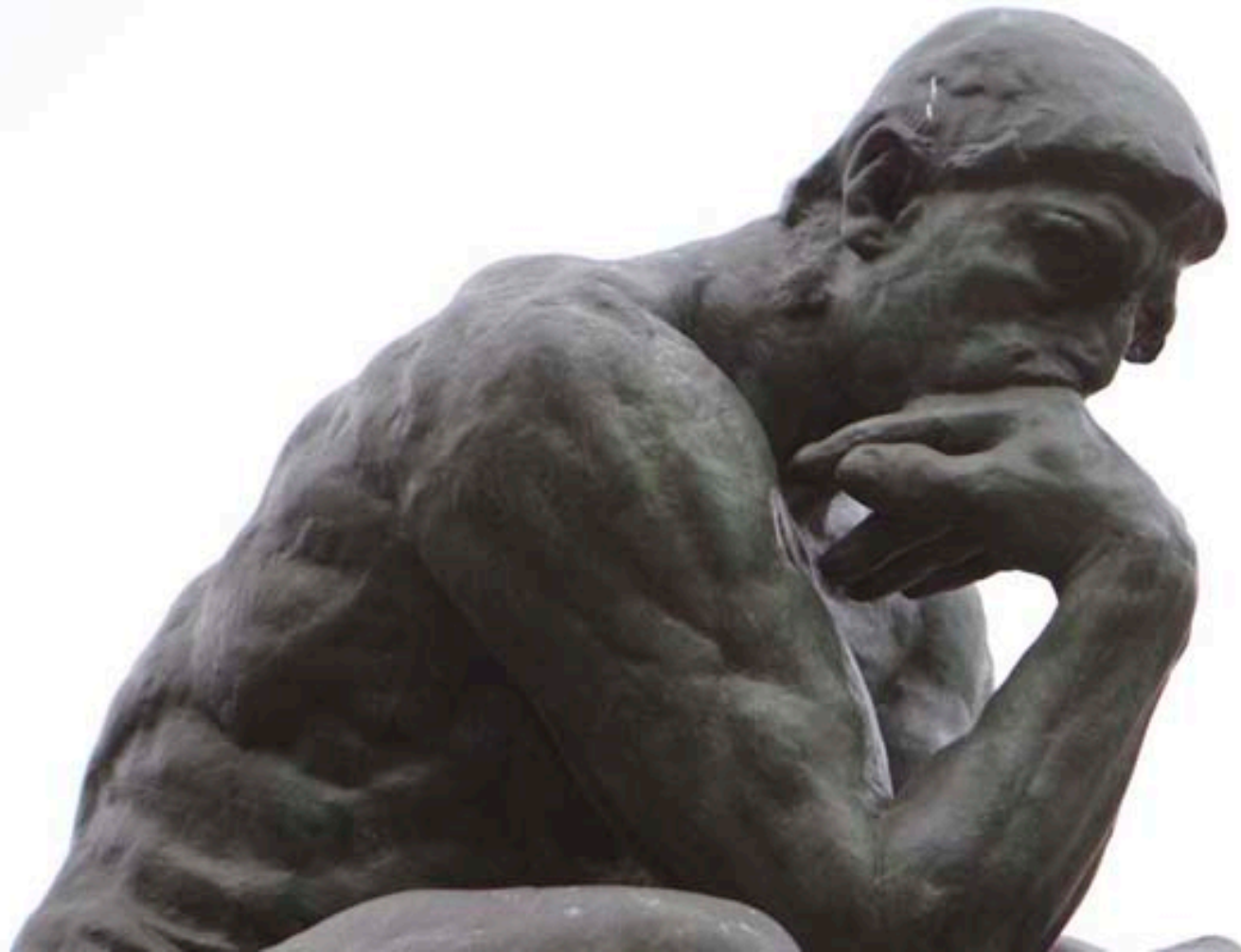
Доступность технологий секвенирования ДНК позволила получать больше достоверных данных



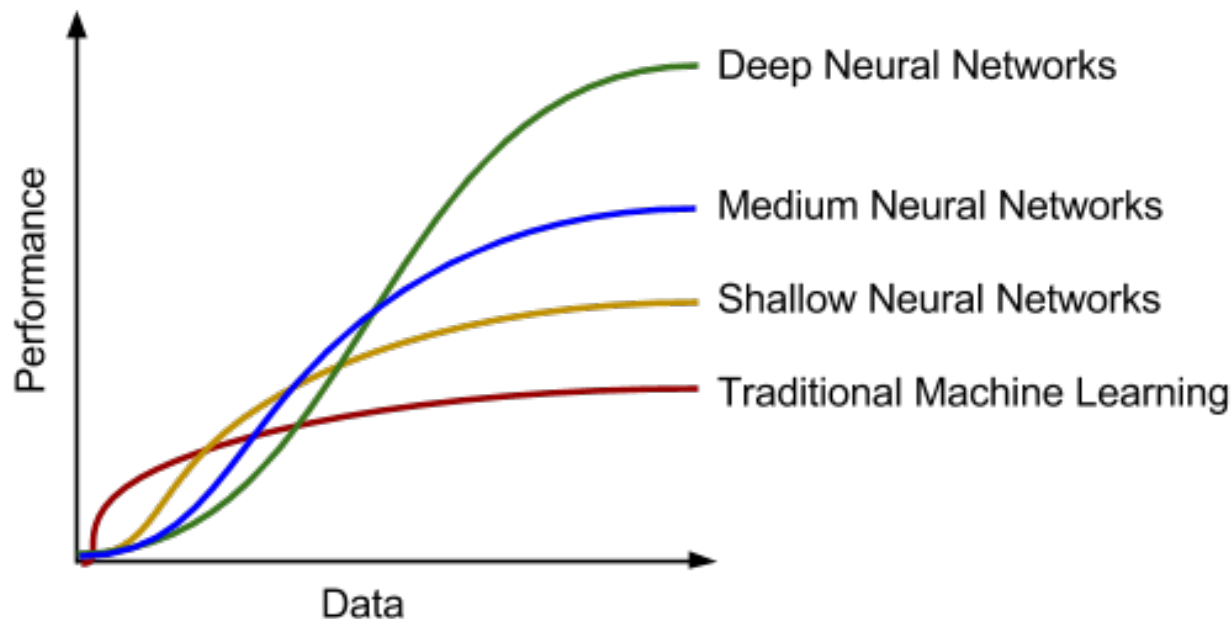
Доступность высокопроизводительной электроники позволила обрабатывать больше экспериментальных данных



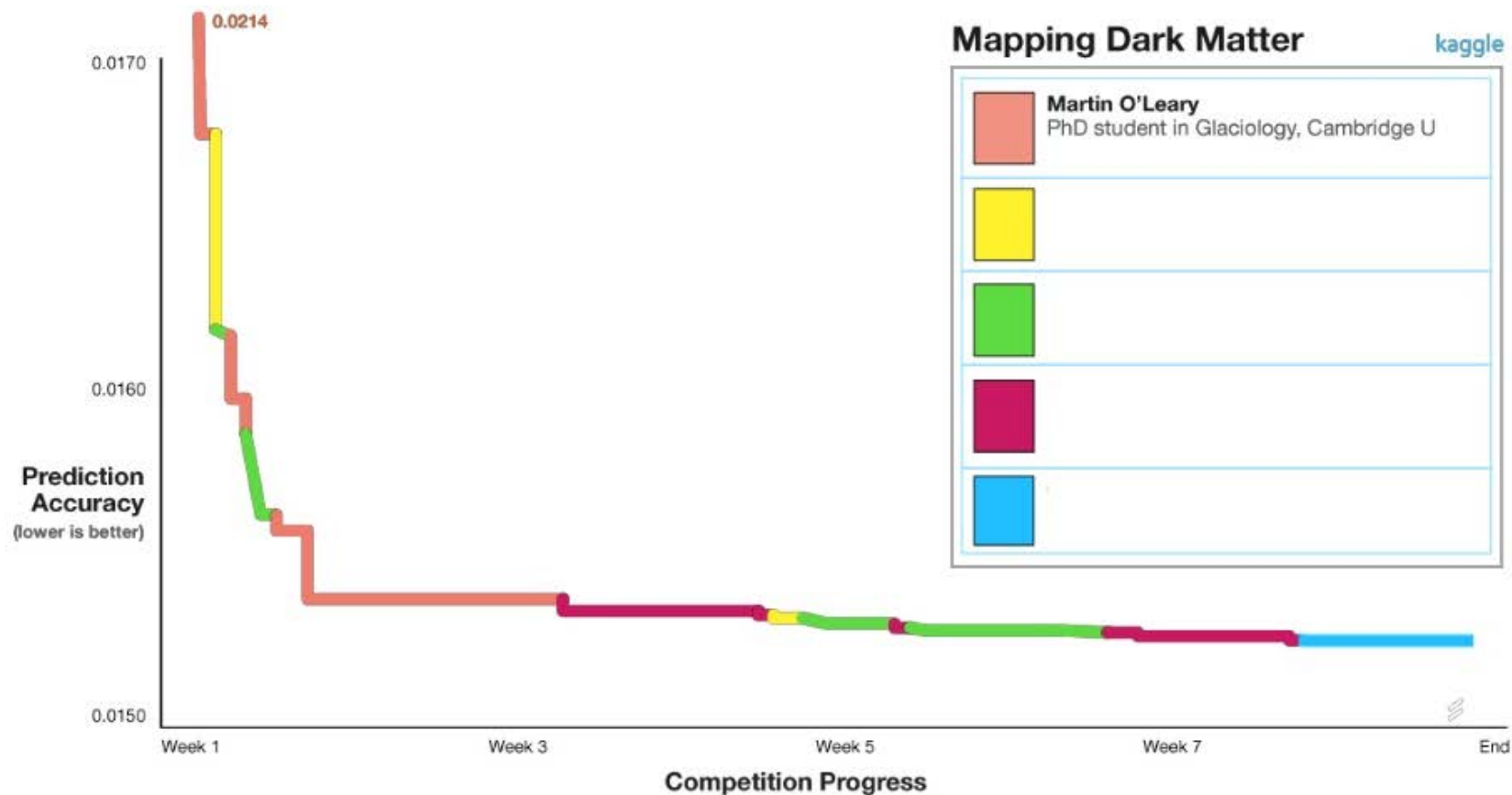
Быстро растущие потоки данных требуют интенсификации процессов их обработки и постоянного пересмотра существующих алгоритмов



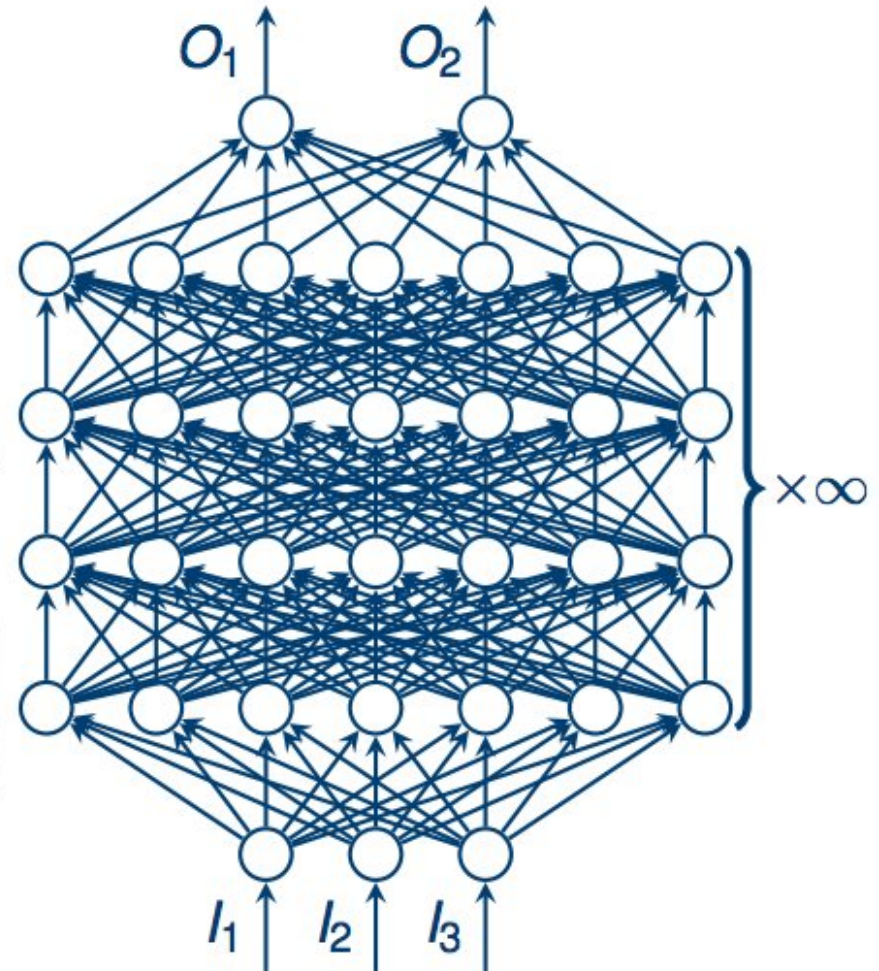
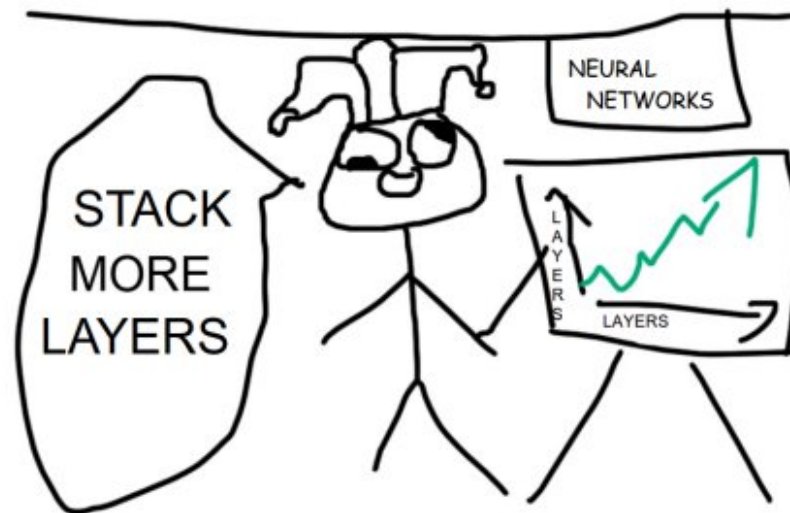
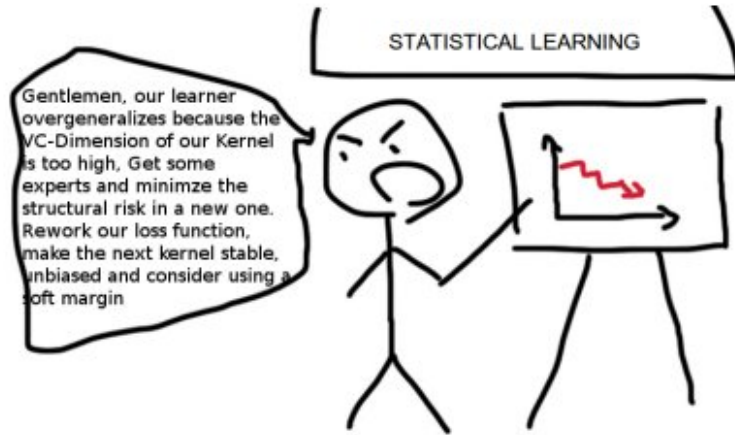
Использованные ранее алгоритмы статистического анализа могут не подходить для обработки выросших объемов данных



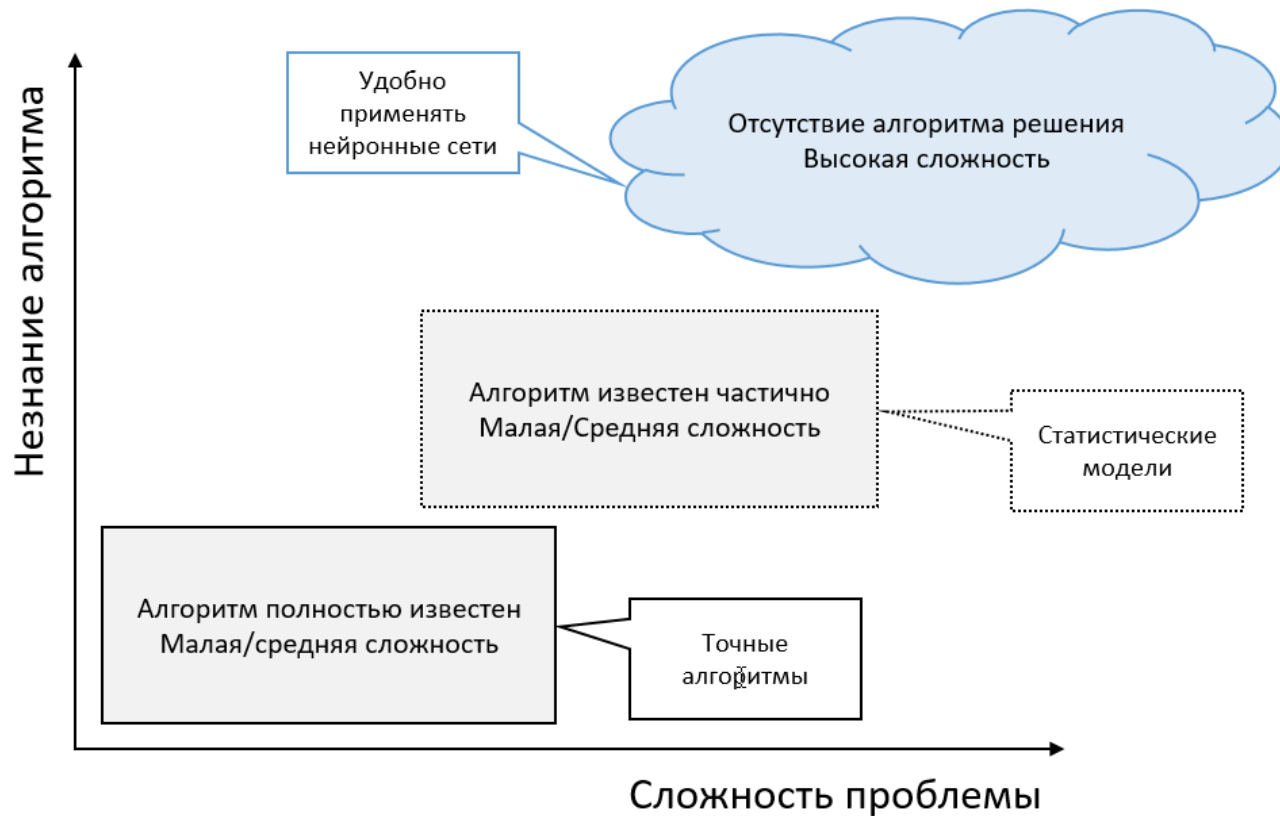
Анализ больших данных позволяет
получить более достоверные результаты



Не каждый алгоритм анализа данных является оптимальным с позиции расхода ресурсов



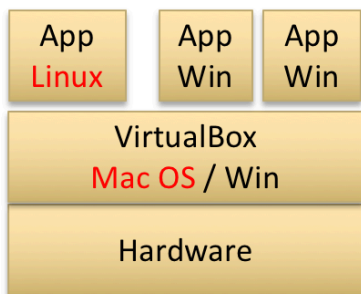
Однозначные проблемы требуют последовательного решения и точного ответа



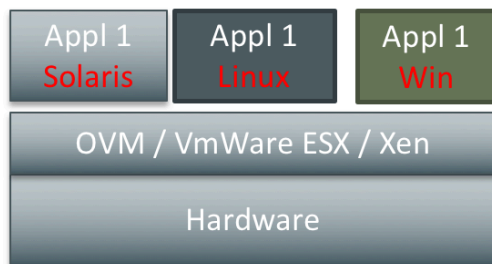
Разнообразие идей, задач и входных данных определяет разнообразие инструментов



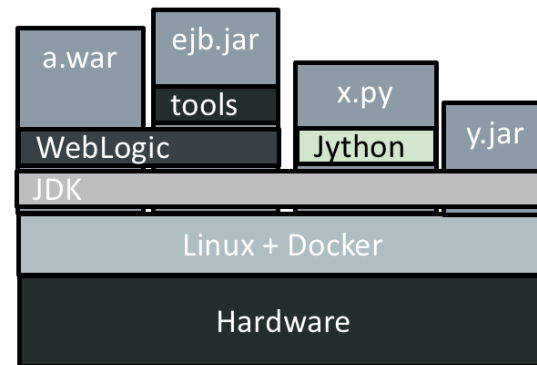
Разнообразие одновременно используемого ПО и его версий требует пересмотра стратегии обновления на системном уровне



Desktop Virtualization:
type 2 hypervisor
= with host OS

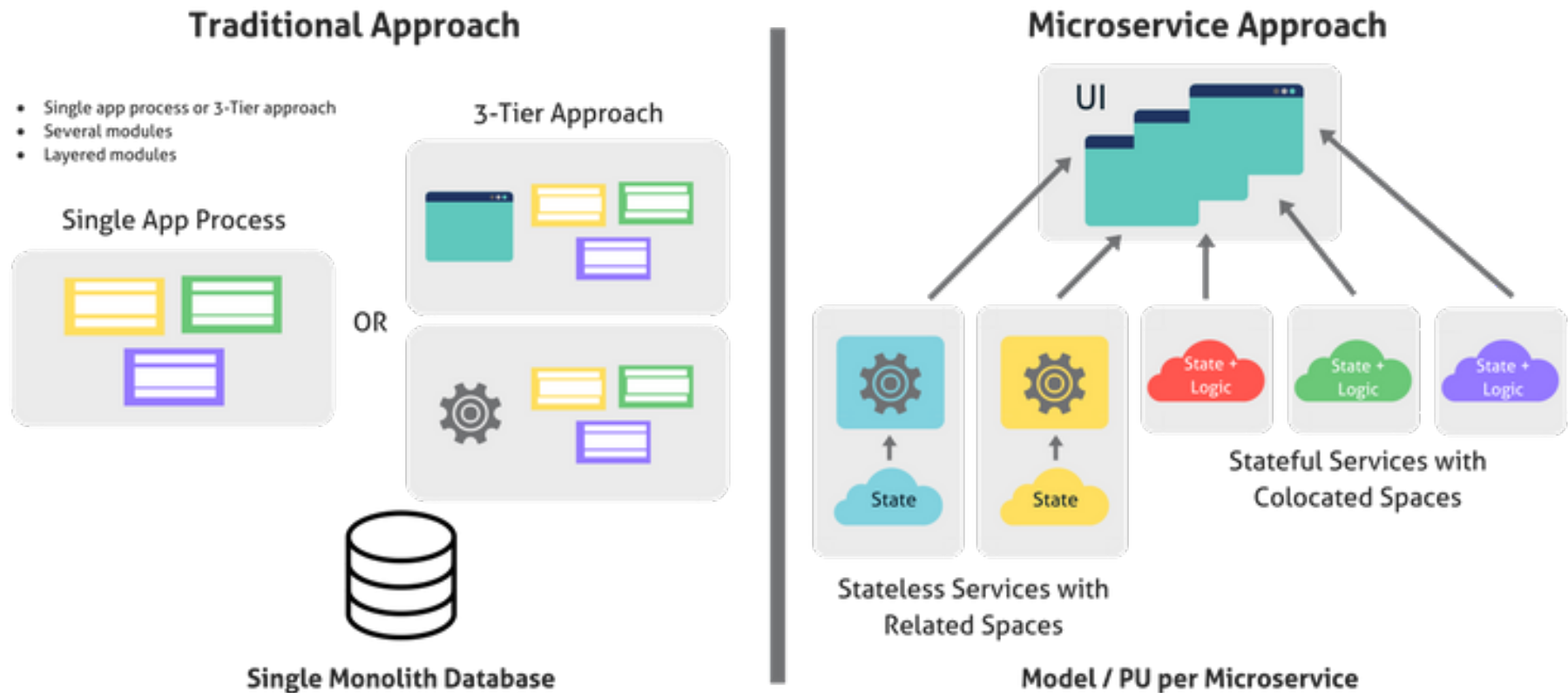


Server Virtualization
type 1 hypervisor
= on bare metal



Docker **container** in Linux
with own FS, network stack /
IP address, process space and
resource limits

Переход от монолитной сервис-ориентированной архитектуры к микросервисной снижает нагрузку на управление всей системой



Большое число запущенных приложений влияет на баланс нагрузки и отказоустойчивость

INFRASTRUCTURE
LAYER



DATABASE



VIRTUAL MACHINES



NETWORK



LOAD BALANCERS

CONTAINER
LAYER

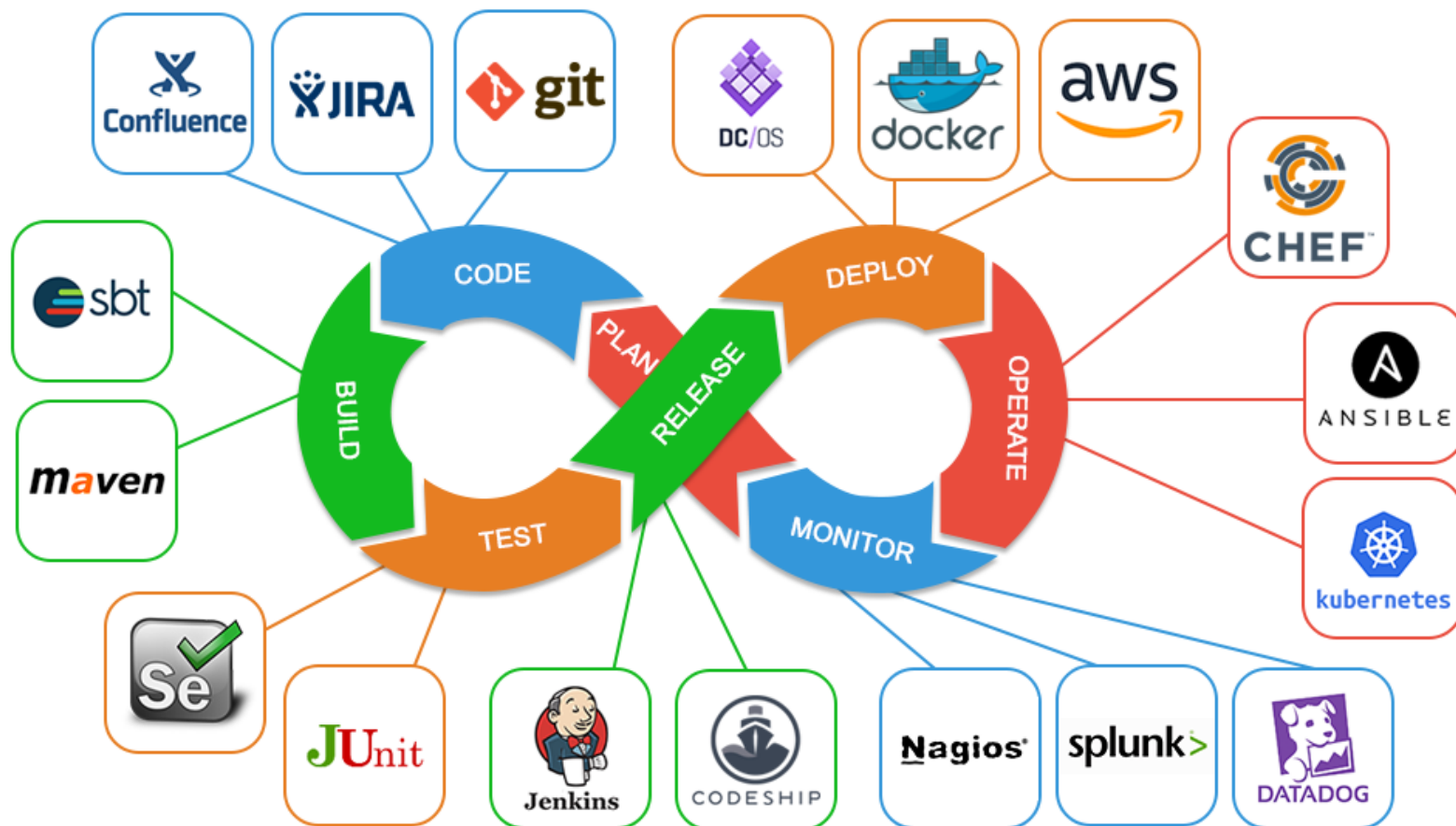


APPLICATION
LAYER



PODS & SERVICES

Ускорение и автоматизация разработки
элементарных сервисов ведет к усложнению
тестирования и разработки в целом



Проект “Kubernetes Cluster on Bare Metal”: продолжение

- Скачать и извлечь файл *nucleotide_fasta_protein_homolog_model.fasta* с референсной БД CARD
 - <https://card.mcmaster.ca/latest/data>
- Извлечь первые 500 последовательностей с заголовками в директорию на node0 */data/reference/CARD*
- Проиндексировать полученный референс с использованием скрипта *cook_the_reference.py* Docker-образа *ivasITyev/bwt_filtering_pipeline_worker*
- Поднять Redis-сервер и его службу на кластере
- Создать чарты *config.yaml*, *master.yaml* и *worker.yaml*
- Вылить *master.yaml* в продакшен и отследить его завершение по команде *kubectl get pods --show-all*
- Вылить *worker.yaml* в продакшен
 - При проблемах с производительностью уменьшить размер референса и повторить заново
- После успешного завершения работы воркеров определить число полных очередей с помощью команды *kubectl logs <имя пода>*