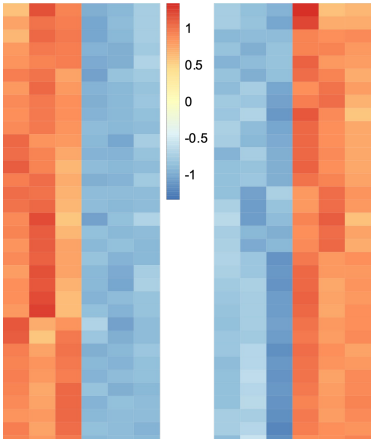
Сергій Науменко — вчений-біоінформатик, кандидат біологічних наук, що працював у дослідницьких відділах дитячих лікарень Торонто і Оттави (Канада), консультантом з біоінформатики в дослідницькій екосистемі медичної школи Гарвардського університету (Harvard School Of Public Health, Бостон, США) і в компанії «Великої Фарми» (Big Pharma). Виконав біоінформатичні аналізи в понад 50 проєктах, розробляв і підтримував біоінформатичні пайплайни, зробив внесок у дослідження менделівських хвороб і механізмів раку засобами геноміки і транскриптоміки, що були опубліковані в журналах «Американський журнал генетики людини», «Гени і розвиток», «Американський журнал медичної генетики».
Учасник професійної спільноти НГО «Геноміка ЮА» — https://genomics.org.ua/.

СЕРГІЙ НАУМЕНКО

БІОІНФОРМАТИКА

БІОІНФОРМАТИКА

ПОСІБНИК



СЕРГІЙ НАУМЕНКО

Біоінформатика
Посібник

Сергій НАУМЕНКО

14.07.2023

Біоінформатика
Посібник

Ідея посібника:

- Олекса Польшин;
- учасники НГО https://genomics.org.ua/.

Технічний редактор: Анатолій Науменко
Літературна редакторка: Ольга Васильєва
Обкладинка: Adam Burggraf
Зображення на обкладинці:
https://hbctraining.github.io/reproducibility-tools/activities/Rmd_exercise4.html
Видавець: Ryan Emmanuel Adam

Пам'яті моїх шкільних учителів:

- математики — Галини Іванівни Курочко;
- фізики — Володимира Ісайовича Якобі;
- української мови і літератури — Валентини Василівни Савчук;
- фізкультури — Анатолія Андрійовича Зайчука.

Посібник можна безкоштовно завантажити за посиланням:
https://github.com/naumenko-sa/bioinf_posibnyk_public/blob/main/bioinf_posibnyk.pdf.
Будь ласка, переказуйте гроші на підтримку українських снайперів:
https://www.facebook.com/profile.php?id=100087932614765.

Зміст

1 Вступ

1.1 Навіщо український посібник з біоінформатики

На перший погляд він не потрібен або взагалі може бути шкідливим. Річ у тім, що біоінформатик повинен якомога швидше приєднатися до світової професійної спільноти, навчитися не просто вільно читати документацію та наукові статті англійською, а й спілкуватися з колегами на форумах, у GitHub, email, на конференціях, співбесідах. Навчання виключно українською унеможливлює подібну інтеграцію, тобто навчання біоінформатика має бути двомовним (українською та англійською), де бажана пропорція — 50/50, якщо не 20/80 на користь англійської (вона не рідна, тому дається важче і потребує більше часу). Однак, незважаючи на те, що всі матеріали (підручники, tutorials (покрокові навчальні інструкції), відеолекції) доступні та англомовні, спочатку дуже важко зорієнтуватися, що саме треба вчити, в якому напрямку рухатись.

Тож мета посібника — дати основні посилання, зорієнтувати читача для самостійної освіти. Переклад tutorials, документації, наукових статей не має практичного сенсу: якісний переклад потребує часу, а всі ці джерела постійно оновлюються, тож будь-який переклад буде застарілим одразу, коли вийде. Але науково-популярні тексти українською, які поширюють фундаментальні знання та останні наукові новини, а також фундаментальні підручники (з математики, статистики, мов програмування, біології), звісно, потрібні. Приклади: *https://nauka.ua/*, *Молекулярна біологія клітини*, *https://zbruc.eu/nauka*.

На цю мить, на початку 2023 року, коли продовжується повномасштабне вторгнення РФ в Україну, науковці та викладачі перебувають на передовій або працюють волонтерами, лікарі не займаються дослідженнями, а працюють у військових шпиталях, студенти навчаються за допомогою зуму, тривають ракетні обстріли, відключення світла, холод, окупація, бойові дії, загибель близьких — здається, що біоінформатика зараз не на часі. Але, попри перешкоди, студенти навчаються, навіть захищають курсові та дипломи з окопів, а запитання з біоінформатики у профільних каналах ставлять щотижня, тож є надія, що цей посібник буде корисним. Утім, багато порад, які він містить, зокрема з тайменеджменту, неможливо виконати під час воєнного стану та відключень світла.

Цей посібник є своєрідним маніфестом надії: після перемоги України ми побачимо відновлення навчального та наукового процесу. Вже зараз біоінформатики українського походження працюють на світовому рівні у провідних наукових лабораторіях та підприємствах біотеху США і Європи, є віддані своїй справі викладачі та талановиті студенти в університетах. Сподіваюся, ми побачимо в Україні конкурентні програми з біоінформатики у вишах, гарні лабораторії, проєкти, біотех-компанії, міжнародні конференції, статті у провідних журналах, програми та бази даних, які будуть використовуватись у всьому світі.

Під біоінформатиками у посібнику так само мають на увазі й біоінформатикинь, яких чимало у професії.

1.2 Значення англійської мови

Володіння англійською — необхідна умова для освіти біоінформатика. Причому потрібно відразу починати спілкуватися та використовувати англійську, а не чекати досконалого оволодіння нею. Можливо, ви ніколи її не вивчите досконало, хоча й практикуватимете все життя. Англійське і взагалі англомовне суспільство толерантне до неідеального знання мови: є багато діалектів та вимов (навіть у самій Англії, не кажучи про різницю між американською, канадською, австралійською англійською), різні професійні словники. До того ж багато хто з технічно та науково обдарованих людей мають проблеми з вивченням мов. Усі ваші співрозмовники розуміють, що англійська для вас може бути не рідною, тому замість того, щоб комплексувати, треба спілкуватися та вдосконалюватися (слухати *https://esl-bits.net/*, радіо *https://www.npr.org/*, читати книжки, журнали, дивитися серіали). У навчанні треба створювати собі ситуації, коли англійська буде активно використовуватися (журнальний клуб). Не так вже й важко набрати рівень англійської, необхідний для навчання та наукового спілкування. Окремим пунктом є scientific writing (наукова письмова мова): уміння писати добрі тексти важливіше за ідеальну розмовну мову.

1.3 Про яку саме біоінформатику йдеться

10 грудня 2022 року я набрав у гуглі слово «біоінформатика» і розглянув перші десять посилань. Перше посилання було на вікіпедію: *Біоінформатика*. І справді, це та сама біоінформатика, про яку написано цей посібник. Але на українській вікісторінці дуже мало інформації. Докладніше про біоінформатику можна прочитати на англомовній сторінці: *Bioinformatics*. Сподіваюся, що хтось із читачів цього посібника витратить 10–20 годин, щоб покращити українськомовну сторінку.

Інші результати з першої сторінки пошуку тішили мене менше: два з них були похідними від сторінки у вікіпедії — один на якомусь учнівському сайті, інший — на спам-сайті, а три посилання були російськомовними.

Нарешті три посилання були на навчальні програми з біоінформатики: КПІ імені Ігоря Сікорського — кафедра біоінформатики, КНУ імені Тараса Шевченка — кафедра інтелектуальних технологій, спеціальність «Біологія (високі технології)». Останнє посилання було на відеолекції Інни Дубчак, організовані у співпраці Центру громадського здоров'я МОЗ України та американських партнерів (це дуже гарні чотири лекції). Усі ці чотири посилання теж є прикладами біоінформатики, про яку йдеться в цьому посібнику. Дуже обмежені результати, отримані за пошуком, додають мотивації щодо випуску цього посібника. На теперішньому етапі кожен внесок важливий.

Посібник стосується **біоінформатики як галузі науки, яка розробляє і використовує обчислювальні та статистичні методи, програмні інструменти, пайплайни і бази даних (інформатику) для вивчення біології**. Більш конкретно в посібнику йтиметься про базові компетенції NGS-біоінформатика (NGS — Next Generation Sequencing) та про аналіз даних секвенування (DNA-seq, RNA-seq).

Визначення можна дати за науковими журналами. **Біоінформатика** — це галузь науки, праці з якої публікуються в рецензованих журналах Bioinformatics, PLOS Computational Biology, BMC Bioinformatics та інших (*https://en.wikipedia.org/wiki/List_of_bioinformatics_journals*), а також методи якої використовують для отримання біологічно значущих результатів досліджень, що їх публікують у найвідоміших рецензованих журналах із біології та медицини (Nature, Science, Nature Reviews Genetics, Nature Genetics, eLife, Nature Protocols, Genome Biology, Nucleic Acids Research, Nature Communications, PLOS Genetics, JAMA, Lancet та багатьох інших).

Визначення за працевлаштуванням: **біоінформатика** — це галузь науково-технічної освіти, спеціаліст із якої може влаштуватися працювати за посадою Bioinformatician, Computational Biologist, Bioinformatics Software Developer, Biostatistician в академічних лабораторіях та біотех-компаніях (включно з компаніями Великої Фарми (Big Pharma) — Pfizer, Moderna, AstraZeneca, Novartis). Приклади позицій можна знайти в indeed.com: *https://ca.indeed.com/jobs?q=bioinformatics*, *https://bioinformatics.ca/jobs/*.

Визначення за конференціями: **біоінформатика** — це галузь науки, доповіді з якої подають на конференції ISMB — Intelligent Systems in Molecular Biology https://www.iscb.org/ismb2022, RECOMB — https://recomb2022.net/, Open Bioinformatics Foundation — https://www.open-bio.org/, а також на конференції з медичної генетики, геноміки, молекулярної еволюції, різних типів раку, вірусології, клітинної біології, системної біології.

Біоінформатик — це також перекладач між понятійними апаратами інформатики та біології. Він / вона може спілкуватись у спільних проєктах з лікарями-дослідниками, лабораторними та польовими біологами, програмістами й математиками.

У посібнику не йдеться про окультизм (це не наука), біоенергетику (іноді під нею мають на увазі окультизм, іноді зелену енергетику (green energy), іноді псевдонауку, а є справді наукові розділи — наприклад, обіг енергії у клітинах з погляду біохімії та клітинної біології), нанотехнології (трапляється як псевдонаука і як справжня наука в хімії та матеріалознавстві).

1.4 Інші підручники з біоінформатики

- Vince Buffalo. Bioinformatics Data Skills. *https://www.oreilly.com/library/view/bioinformatics-data-skills/9781449367480/* (якщо обирати один підручник, то це він).
- David W. Mount. Bioinformatics. Sequence and Genome analysis. *https://www.google.ca/books/edition/Bioinformatics/bvY21DGa1OwC?hl=en&gbpv=1&printsec=frontcover* (підручник застосовує мову Perl, тобто є трішки застарілим, але містить багато важливих фундаментальних концепцій).
- Bioinformatics. From Genomes to Therapies. *https://onlinelibrary.wiley.com/doi/book/10.1002/9783527619368*.

- Bioinformatics. 3d edition. *https://www.amazon.ca/Bioinformatics-Practical-Guide-Analysis-Proteins/dp/1119335582/* (є багато таких книг, де редактори запрошують 10–20–30 учених, кожну главу пишуть різні автори, а в результаті отримують дуже змістовну книгу).

Підручники, які стосуються молекулярної еволюції (тобто галузі науки, яка вивчає зміну послідовностей ДНК та білків у часі), філогенетики, обчислювальної біології також містять багато фундаментальних концепцій біоінформатики:

- Dan Graur and Wen-Hsiung Li. Fundamentals of molecular evolution. 2nd edition. 2000. *https://global.oup.com/academic/product/fundamentals-of-molecular-evolution-9780878932665?cc=ca&lang=en*.
- Bernhard Haubold. Introduction to Computational Biology. Evolutionary Approach. *https://www.amazon.com/Introduction-Computational-Biology-Evolutionary-Approach/dp/3764367008*.
- Paul G. Higgs and, Teresa K. Attwood. Bioinformatics and Molecular evolution. 2005. *https://onlinelibrary.wiley.com/doi/book/10.1002/9781118697078*.
- Ziheng Yang. Molecular Evolution. A statistical approach. *http://abacus.gene.ucl.ac.uk/MESA/2014Yang.MESA.CoverContents.pdf*.
- Dan Gusfield. Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology *https://www.amazon.ca/gp/product/B00AKE1UZU/ref=dbs_a_def_rwt_hsch_vapi_tkin_p1_i0*.
- Joseph Felsenstein. Inferring Phylogenies. *https://global.oup.com/ushe/product/inferring-phylogenies-9780878931774?cc=ca&lang=en&*.
- The Phylogenetic handbook. *https://www.cambridge.org/core/books/phylogenetic-handbook/A9D63A454E76A5EBCCF1119B3C56D766*.

1.5 Як працювати з посібником

Посібник написаний так, що читачеві необхідно переходити за посиланнями (їх зібрано чимало), читати книжки, статті, писати програми, опановувати аналізи, практикувати, робити помилки, виправляти їх, самостійно розбиратися у складних ситуаціях. Просто читати — не допоможе, адже це не підручник, а саме посібник для самостійної роботи. Тільки так можна навчитися біоінформатики.

2 Кар'єрний шлях

2.1 План кар'єри

У нашій культурі це поняття майже відсутнє через те, що завжди відбувається якась халепа, і підсвідомо ми впевнені, що довгострокові плани неможливі: чорнобильське покоління, злиденні 90-ті, війна 2014–2023. З іншого боку, кар'єрних можливостей в Україні з біоінформатики, як у академії, так і в індустрії, теж небагато, що обмежує планування. У розвинутих країнах (G7, EU, Австралія, Нова Зеландія, Південна Корея, Сінгапур, навіть Південна Африка, де біоінформатика розвинута досить добре — так добре, що вчені ПАР стали першими, хто сповістив про штам COVID-19 «Омікрон») життя більш стабільне, тож люди справді будують кар'єру, як будинок. Це довгий план на 5–10–20–30 років, який складається з навчання, ступенів, мети, бюджетів, ризик-менеджменту, плану Б.

2.2 Академія

Типова (північноамериканська) академічна кар'єра (*ступені*, *британська специфіка*):

- бакалавр;
- магістр;
- PhD (аспірант);
- постдок;
- ще постдок;
- assistant professor;
- associate professor;
- full professor.

Треба стрибати якомога вище на перших ступенях. Кожен ступінь потребує розсилання багатьох анкет-заяв (job application) та пересувань між містами і країнами. Що вищий рівень, то жорсткіша конкуренція. Тобто треба вчитися на магістра з біоінформатики в одному з найкращих центрів, який хоча б відомий у світі, їздити на конференції та шукати наступну позицію. На PhD подавати на 10 програм, студенти намагаються потрапити до найсильніших професорів у своїй галузі. PhD проходить енергійніше, тому що це не «пів години ганьби та хліб з маслом на все життя», які розтягуються на 10 років, а один з етапів — перепустка в наступний тур, коли треба зробити гарні публікації у провідних журналах.

На постдок треба потрапити в одну з трьох найкращих лабораторій у вузькій галузі, і щоб керівник був відомий тим, що його постдоки влаштовувалися на постійні позиції, а не тим, що він просто використав їхню працю. Включно до кінця постдока (одного чи двох), принаймні в США, це бідне життя, часто в борг (студенти після MIT та інших топ-університетів можуть мати до 100 тис. доларів боргу).

На постійну посаду професора треба подавати 50–100 анкет-заяв, їздити різними країнами (до 20 поїздок) на співбесіди (також проводити job talks — розгорнуті доповіді), отримати 1–2 пропозиції (job offer). Треба мати план досліджень, грантові успіхи, план викладання, план розвитку лабораторії. Здебільшого вчені йдуть туди, де отримали пропозицію, хоч у Саскачеван у Канаді, хоч у Нью-Мексико в США, а там вже проводять решту життя, будуючи основну частину кар'єри. Інколи буває, що всю лабораторію переводять до іншого міста, але це рідко.

Знання якоїсь європейскої мови та походження може полегшити пошук постійної посади — іспанці після навчання в США їдуть до Іспанії, італійці — до Італії. Є багато пропозицій у Європі, де створені спеціальні умови для висококваліфікованих кадрів, щоб вони повернулися. Сподіваюся, щось подібне буде колись і в Україні.

Коли змагання за постійну посаду вигране, вчений усвідомлює, що все тільки починається, — зараз треба змагатися за гранти з такими самими успішними професорами та ще успішнішими; викладати, наймати працівників, керувати людьми, грошима, часом, оновлювати обладнання, водночас триматися в темі й давати результати, видавати статті. За декілька років, коли стало зрозуміло, що людина справляється, вона отримує шлях до постійного контракту (tenure track), а згодом і сам постійний контракт (tenure), тобто вже зможе працювати до пенсії в цьому університеті й трохи менше напружуватися.

Кількість постійних посад в академії дуже обмежена — лише 1–2–5 від 100 аспірантів та постдоків можуть здобути професорські посади, а точний відсоток залежить від галузі, країни. Також розрізняють посади в топ-університетах та посади в університетах другого і третього ряду. За останні 30 років кількість постійних посад відносно кількості постдоків тільки зменшується, тобто конкуренція посилюється.

Окрім професорських, в академії є допоміжні постійні чи контрактні позиції — research associate, instructor, software developer, у великих та надвеликих лабах (є лабораторії до 100 осіб) потрібні керівники середньої ланки та інші кваліфіковані фахівці, які стабільно працюватимуть 10 років, а не стрибатимуть, як студенти та постдоки. Є також постійні посади в бюрократичних структурах університетів, державних органів, сервісних лабораторій (sequencing facility, bioinformatics core).

2.3 Індустрія

Ще 20 років тому неотримання постійної посади на шляху академічної кар'єри сприймалось як невдача та поразка, за якої необхідно було переходити до індустрії, щоб мати другий шанс (industry я перекладаю саме як індустрію, а не промисловість, тому що в українській мові «промисловіть» має «залізний» та виробничий відтінок, як-от важка промисловість, машинобудування або легка промисловість, у той час як industry — це взагалі технологічний бізнес, наприклад, фармацевтика, розробка ліків). Однак останніми роками перехід до біотех-

індустрії дедалі більше сприймається як нормальний розвиток кар'єри. Великі компанії й стартапи дедалі частіше мають всі умови для наукових досліджень та дуже швидкого доведення наукових результатів до практики.

Яскравий приклад — дуже швидка розробка, клінічні дослідження та виробництво вакцин від COVID-19, виготовлених за новітньою mRNA-технологією компаніями Pfizer та Moderna, які без перебільшення врятували мільйони життів. Тобто деякі біоінформатики одразу планують кар'єру в індустрії після аспірантури або після постдока.

В індустрії теж є різні варіанти: працювати біоінформатиком у біотех-стартапах чи великій фармі, бути консультантом. Фарма — більш стабільне положення, там теж треба будувати кар'єру. Біотех-стартапи — більш динамічно, більш ризиковано. У фармі можна працювати без PhD, зі ступенем магістра чи навіть бакалавра. Але це обмежуватиме ваш кар'єрний шлях. Скажімо, з бакалавром ви, може, будете у фармі програмістом і за 10–20 років дійдете до старшого програміста, а може, до керівника програмістів — теж добра кар'єра. Але що саме програмувати та куди рухатись, вирішуватимуть колишні магістри, PhD та постдоки.

До підбору на посади велика фарма ставиться дуже серйозно: наприклад, ви рухаєтесь в академічній колії, причому рухаєтесь добре, але після постдока йдете до фарми замість змагання за посаду професора. Фарма отримує перевірену людину, щонайменше з досвідом PhD та постдока. Інший шлях до фарми — через консалтинг — після бакалавра-магістра влаштовуєтесь у консалтингову фірму (такі є в Україні), яка продає ваш час клієнтам. Якщо ви добре впораєтесь, потрапите до найкращих консалт-фірм та найкращих клієнтів (Велика Фарма). Якщо й там добре попрацюєте — фарма може вас запросити до себе. Коли вже потрапили до фарми, там є не менше як 10 рівнів зростання (усі мають назви на кшалт senior scientist, principal scientist, VP R&D, supreme leader of the galaxy). У вас є мета на кожен рік та квартал, є керівники та підлеглі. На деяких рівнях можна переходити до іншої великої фармкомпанії або на вищі посади в компаніях меншого розміру.

Шлях до фарми та взагалі біотеху вже давно нормалізований. Але починаєте ви в академії — серед тих, хто в ній затримався. Звісно, вони часто вважають, що їхній шлях правильний. До того ж, коли вони отримували позиції 10–20–30 років тому, це було значно легше. Тому важливо мати загальний кар'єрний план, знати опції та не впадати у відчай, якщо щось не виходить, як ви планували. Індустрія зараз пропонує великі можливості, зокрема для досліджень.

Також багато хто чесно попереджає про проблеми та труднощі в академії, але про індустрію це менш відомо, тому що інформація часто закрита (або «все чудово», або нічого). Звісно, у фармі може бути дуже велика різниця між департментами. Взагалі у фармі дуже багато зустрічей (**meetings**), тому що компанії великі, на тисячі людей, їм треба тримати структуру. Працює дуже багато людей, ви майже ніколи не робите щось наодинці — завжди з кимось. Корпоративна культура: вам дають курси щодо цінностей компанії, які

можуть бути і непоганими, а можуть сприйматися як настирливі; контроль у всіх аспектах — що відбувається, коли, як, менеджмент. У фармі все прив'язано до **портфоліо** компанії та мети компанії. Іноді ваші дослідження можуть збігатися з цілями компанії, а іноді — ні, тоді знадобиться відкинути ваші цілі, бо **Compliance**. Менеджмент вирішив, що вашу улюблену тему треба закрити та зайнятись іншою, — саме так буде зроблено, а ви повинні з подвійним ентузіазом працювати над новою темою. Треба засвоювати багато різних IT-систем корпоративного рівня. Скажімо, чатитись у slack не вийде — вийшов наказ, і всі 10 000 співробітників перейшли у MS Teams і страждають там. Якщо запровадили JIRA в компанії — будете на JIRA з надвеликим ентузіазмом, хоча ви, може, терпіти її не можете. Видають комп'ютер, який ви не обирали, — всі однакові, встановити на нього теж нічого не можна — тільки через адмінів. Гра за бонуси, плани, виконання планів. Не можна базікати про свою роботу всюди, як в академії, — інформація про бізнес компанії тільки за пресрелізами. У компанії є секрети, які не можна розголошувати.

Тобто все це не фатально: середовище може бути дуже здоровим, люди приємні, гроші, задачі, досягнення, але це зовсім інший світ, до якого треба адаптуватися після навчання та праці в академічному світі.

Один яскравий приклад: у корпорації робочі календарі можуть бути відкритими для всіх. Якщо людині потрібно з вами зустрітися (комусь справді потрібно, а хтось просто любить призначати наради, тому що не хоче працювати або любить бути менеджером та показувати свою владу), то вона просто тицяє у ваш календар і проводить нараду, не питаючи, чи вам це зручно. Так у вас може бути 3–4–5 годинних нарад щодня, а працювати треба десь між нарадами. В академії теж багато нарад, особливо на вищих ступенях, але все ж таки свободи більше (а грошей менше).

Якщо ви плануєте прямувати до індустрії, треба приділяти увагу своєму *LinkedIn* профілю, розширювати свою мережу, шукати колег, які вже влаштувалися у фармі, спілкуватися з ними, відвідувати конференції, презентації, більше спрямовані на бізнес-світ, практику. LinkedIn — це соціальна мережа професійних контактів із жахливим інтерфейсом, а по суті — це база даних для рекрутерів. Критерій гарного профілю у LinkedIn — це описати свої досягнення так, щоб вам було страшенно соромно за цей опис. Такі правила гри!

2.4 Критерії підбору наукового керівника для PhD (аспірантури)

Ідеальної лабораторії не буває, але перед тим, як ви підписуєтесь на декілька років життя, приєднуючись до лабораторії, важливо зрозуміти, наскільки вона гарна. Є абсолютні red flags (насильство), є просто проблеми, які можуть бути розв'язані або з якими можна жити. Однак важливо, щоб керівник та лабораторія мали позитивну динаміку, — рухались у напрямку нормальної (ідеальної) лабораторії. Керівник — це дорослiша людина, ніж аспірант. Якщо він / вона одразу не розуміє, що має бути нормальною лабораторією, то вже ніколи і не зрозуміє, а перевиховати керівника ви не зможете — треба одразу шукати іншого.

Основні вимоги до керівника лабораторії такі:

- достатній науковий рівень (див. індекс Гірша в розділі «Журнальний клуб»), публікації в міжнародних журналах;
- налаштована лабораторія (є сайт лабораторії, співробітники, публікації, гранти, проєкти);
- GitHub, якщо лабораторія випускає та підтримує програми;
- у лабораторії є життя: журнальний клуб, доповіді, проєкти, статті;
- у лабораторії є приміщення і воно не завалене сміттям;
- люди приходять на работу в лабораторію і працюють, а не ганяють каву, чаї чи коньяк у творчій атмосфері або ведуть псевдоінтелектуальні бесіди;
- керівник більшість часу перебуває в лабораторії, на роботі, а не бозна-де;
- керівник займається саме наукою, лабораторією, а не бозна-чим (бізнесом, політикою);
- лабораторія має обладання, яке працює (якщо немає експериментів — щонайменше комп'ютерний парк, принтер, доступ до статей, баз даних, HPC);
- є перспективні теми робіт, якийсь напрацьований напрямок (а не «може, потім щось вигадаємо» чи «займаємось усіма напрямками одразу»);
- попередні аспіранти публікують статті, захищаються та рухаються далі, а не зникають у нікуди (див. розділ alumni на сайті лабораторії);
- керівник вміє отримувати фінансування на дослідження і готовий навчати цього вас;
- керівник розуміє, що аспірантам треба щось їсти, і дає їм змогу заробити (гранти або викладання);
- фінансовий бік відрізняється в різних країнах: десь аспірантура — це робота, за яку платять, а десь — це навчання, за яке платить PhD-студент; десь це навчання, але є стипендія, на яку можна жити (невелика), однак у будь-якому разі важливо обирати керівника, який розуміє, що людям треба їсти, а не такого, який «думає тільки про науку» (сам / сама при цьому добре влаштовані);
- керівник має міжнародні контакти з провідними вченими у галузі, тобто він / вона «у грі», а не у власній бульці;
- лабораторія виїжджає на міжнародні конференції, співробітники мають усні доповіді (а не тільки постери);
- керівник не має великих особистих проблем (див. розділ «Психологічне здоров'я»): не п'є, не вживає наркотиків, не вдається до насильства щодо працівників (фізичного, сексуального, психологічного).

2.5 Критерії підбору наукового керівника для постдока

- Ті самі, що і в аспірантурі: науковий рівень, публікації, гранти, жива лабораторія;

- тематика постдока збігається і є продовженням PhD, але більш спеціалізована і на більш високому рівні (якщо ви не змінюєте тему);
- відомо, що попередні постдоки цього керівника влаштовуються як PI (principal investigator) або в індустрії, а не просто працюють до повного виснаження 5 років і поповнюють лави пацієнтів психіатра;
- постдок — це низькооплачувана посада в будь-якому разі, але керівник не повинен бути жадібним («гроші маю, але платити не буду»). Жадібний керівник і відгук вам не напише у той момент, коли він буде життєво необхідний для здобуття наступної посади. З'ясувати це дуже легко: на останніх стадіях співбесіди йдеться про гроші, і ось тут вже видно, чи керівник платить так мало, що менше не можна, або так багато, скільки дозволяють гранти лабораторії (теж може бути небагате життя, але важливо, як керівник ставиться до грошей, що для нього / неї на першому місці — люди чи гроші). Red flag (застереження) — якщо керівник має мільйонні гранти (цю інформацію легко знайти, а можна й прямо запитати — які гранти у лабораторії), але жадібно давиться за додаткові $ 5 тис. на рік (US, Canada);
- науковий керівник має рекомендації щодо керівника постдока і були якісь позитивні взаємодії під час PhD;
- постдок — це джокер, коли можна вирватись із колії, але можна й потрапити з калюжі в болото;
- постдок-позицій дуже багато, обирати треба ретельно, найчастіше проводять 5–10 співбесід під час конференцій, щоб визначитися, — дивляться, хто є хто у вашій галузі (див. розділ «Networking»);
- зі свого боку, керівники лабораторій також шукають постдоків, проводять багато співбесід і намагаються набрати найкращих працівників, хоча на ретельний вибір у них часто немає часу, тож для них постдок теж є джокер, вони намагаються страхуватися співбесідою з усіма членами лабораторії;
- є великі та успішні керівники з великими лабораторіями, де постдоків дуже багато і вони залишені напризволяще (але буває й так, що їм дають усі можливості майже на рівні PI, щоб просувати свій проєкт);
- є керівники, які стали PI лише вчора (буквально), а вже наймають постдоків (це правила гри). Такі керівники можуть бути й хорошими (в них є свіжі ідеї, багато сил, вони готові працювати з постодоком), але й погані теж можуть трапитись (у яких немає анінайменшого розуміння, чим займатись, лабораторії теж немає — мовляв, «усе тільки починається»);
- добре, якщо є комбінація керівників: молодий керівник працює в тандемі з дуже досвідченим керівником іншої лабораторії;
- керівник у середині кар'єри має вже стабільну лабораторію, гарні публікації та час працювати разом із постдоками;
- тема грошей взагалі не повинна бути табу: хороший керівник розповість, які в нього / неї гранти, в які гранти постдок робить внесок (з яких оплачується його робота), на які гранти можна буде подати заяву — колективно або

персонально;

- одна з головних відмінностей постдока від аспірантури — за ним вже йде постійна посада в академії чи індустрії. Хороший керівник сприятиме вашому успіхові: допомагатиме розробляти план нової лабораторії, навчати отримувати грантове фінансування на академічному шляху, знаходити проєкти, де ви будете консультантом, щоб придбати необхідні контакти в індустрії, якщо це кінцева мета. Звісно, керівник насамперед переслідує свої цілі: наукова робота лабораторії, звіти за грантами, статті, але хороший керівник побудує працю так, що обидві мети (його лабораторії та вашої особистої кар'єри) будуть досягнуті. Для поганого керівника постдок — це просто дешевий інструмент, який буде замінений за 4–5 років, адже подальша кар'єра постдока йому нецікава;
- звісно, треба мати особистий план кар'єри і його просувати. Якщо плану немає, то й найкращий керівник не допоможе, а якщо він є, то й поганий керівник не стане на заваді.

3 Складники освіти біоінформатика

3.1 Формальна освіта

В ідеалі освіту біоінформатика можна здобути за програмою в університеті: наприклад, *бакалаврська програма університету Каліфорнії в Лос-Анджелесі*.

Є магістерські програми з біоінформатики, на які можна потрапити після біологічних програм або інших наукових чи технічних:

- програма в університеті Bath, UK: *https://www.bath.ac.uk/courses/postgraduate-2023/taught-postgraduate-courses/msc-molecular-biosciences-bioinformatics/#course-structure*;
- програма в університеті провінції Британська Колумбія, Канада: *https://www.grad.ubc.ca/prospective-students/graduate-degree-programs/master-of-science-bioinformatics*.

Біоінформатиками також стають під час аспірантури (PhD) чи навіть постдока, коли дослідник-біолог опановує засоби біоінформатики для свого проєкту, або дослідник з обчислювальної галузі (computer science) потрапляє до біологічного проєкту.

3.2 Чотири складники

Основні складники освіти біоінформатика:

- біологія (класичні біологічні курси, щонайменше молекулярна біологія клітини, генетика, біологічна еволюція та знання біології проєкту чи конкретного напрямку);
- статистика (R, Rstudio, основи статистики, розподіли, помилки, візуалізація, пакети аналізу);
- програмування (python / perl, структури даних, алгоритми, дискретна математика, контроль версій програмного коду, debug, тести);
- Linux (командний рядок, робота на сервері чи кластері, робота з хмарними системами).

Я тут навіть не зачіпаю хімію, яка дуже важлива для білкових біоінформатиків, які працюють над дослідженнями ліків, а також для тих, хто розробляє нові молекулярні аналізи.

Насправді це вже дуже складна суміш знань, навіть якщо кожен із компонентів опанований не дуже глибоко:

- зазвичай математики та програмісти мають проблеми з біологією: після логіки та визначеності абстракцій біологічне знання здається їм невизначеним, складним, заплутаним, дуже об'ємним;

- біологи та медики натомість можуть мати проблеми з програмуванням та математикою: абстракції їм здаються несуттєвими, тому що вони безпосередньо не відповідають жодній біологічній реальності;
- програмісти, біологи, медики можуть мати проблеми зі статистикою;
- математики можуть сприймати статистику з фундаментального боку, тоді як для проєкту може знадобитися адаптована та спеціально розроблена біологічна статистика (реально я бачив математика, який буквально докладав T-тест для аналізу диференційної експресії генів, ігноруючи всю статистику, спеціально розроблену в пакетах DEseq, edgeR);
- в усіх можуть бути проблеми з командним рядком Linux, особливо після сучасних інтерфейсів у браузері та смартфонах, а також проблеми з програмуванням графіки (побудовою якісних наукових ілюстрацій).

Дайте, наприклад, дефініцію поняття гена. Програміст очікує на структуру даних і координати, математик — на точне визначення, а біолог розповідатиме про алелі, локуси, ознаки, закони Менделя, ДНК, білок-кодувальні гени, трансляцію, транскрипцію, РНК-гени, малі РНК, екзони, сплайсинг, ортологи, паралоги тощо, доки його / її не зупинять. Біоінформатик повинен буде визначитись щодо гена в конкретній ситуації: можливо, це будуть усі білок-кодувальні гени людини, координати яких записані у форматі GFF, версії Ensembl-104, для геномного референсу hg38.

3.3 Крива самовпевненості

Коли біологи опановують інформатику або інформатики — біологію, треба пам'ятати про криву самовпевненості: *https://en.wikipedia.org/wiki/Dunning%E2%80%93Kruger_effect*, *https://en.wikipedia.org/wiki/Four_stages_of_competence*.

На першій (найменш досконала — дилетант) і останній (найбільш досконала — професіонал) стадії опанування знання чи навички люди психологічно можуть почуватися однаково, тобто біолог, який написав великий скрипт на Python на 1 000 рядків (або навіть маленький на 100 рядків), може почуватися досвідченим програмістом, хоча він таким не є (а наступні роки не вдосконалю-ватиме своїх навичок), а математик чи програміст може розібратися в таблиці генетичного коду та кодуванні ДНК буквами АТГЦ і провести залишок життя за розробкою теорем щодо властивостей генетичного коду, які не мають жодного біологічного значення, і буде цілком впевнений, що робить унікальний внесок у біологію.

Тобто звідки б ви не прийшли до біоінформатики, вам доведеться полишити зону комфорту: програмістам усе життя потрібно цікавитися біологією (хоча вони ніколи не наздоженуть біологів), біологам усе життя потрібно вдосконалювати програмування (хоча вони ніколи не наздоженуть програмістів). Отже, біоінформатик — це нелегка професія, яка потребує психологічної стійкості.

3.4 Чому біоінформатиків бракує

Програміст може розвивати свою кар'єру в програмуванні як архітектор або головний технічний спеціаліст або лідер команди: у нього вистачає проблем і без того, щоб вчити незрозумілу та неприємну біологію. До того ж зарплатні досягнення можуть і не зростати в напрямку кар'єри біоінформатика, а саме зростати на головному шляху програміста, особливо якщо рухатись до інвестбанку (програмісти, поцікавтесь, де зараз автор C++). У цьому напрямку також часто рухаються біоінформатики, які прийшли з біології: відкинути біологію, вивчити добре якусь нову платформу — і можна остаточно «перейти в IT», тобто перестати займатись біологією і «нарешті почати жити».

З боку біолога, якщо він навіть опанував методи біоінформатики на стадії аспіранта (PhD) чи постдока і просунувся в кар'єрі до професора, директора лабораторії в академії чи менеджера-дослідника в індустрії, найпривабливіший розвиток кар'єри — це інвестувати саме в менеджмент, гранти, дослідження або знання портфоліо компанії, тобто знову вийти з практики біоінформатика.

Окрема історія з лікарями-дослідниками. Багато хто з них безпосередьо працює з пацієнтами, тобто добре, якщо в них є час на медико-біологічні дослідження. Дуже рідко в них є час, щоб зайнятися програмуванням та біоінформатикою, хоча, наприклад, у системі Harvard Medical School багато дослідників опановують аналіз RNA-seq у Rstudio та інші типи аналізу даних.

Тобто люди приходять до біоінформатики зусібіч, але й вимиваються з неї теж урізнобіч. Останнім часом додався новий тренд: біоінформатики вчать Machine Learning (у всіх можливих дефініціях) і полишають біоінформатику (як визначено в цьому посібнику).

Саме тому біоінформатик — це дуже дефіцитний спеціаліст. Багато керівників у кулуарах конференцій та бізнес-зустрічей скаржиться, що біоінформатика дуже важко знайти. Існує ціла індустрія, де рекрутери шукають біоінформатиків через LinkedIn, наймають їх як контракторів і перепродають іншим компаніям.

3.5 Наукові проєкти виконують командою

Як же взагалі можливі наукові проєкти, якщо вони потребують таких різноманітних знань? Сучасна наука давно перестала бути справою самотнього геніального вченого, який сидить у своєму кабінеті чи лабораторії. У реальному та успішному медико-біологічному проєкті можуть співпрацювати:

- польові біологи, які виїжджають в експедиції та навіть пірнають на дно озера чи в печери і добувають зразки;
- експериментальні біологи, які займаються експериментами в лабораторії, на зразках чи клітинних лініях;
- молекулярні біологи, які допомагають з молекулярними методами;
- спеціалісти з секвенування;
- біоінформатики;

- статистики;
- програмісти (якщо у проєкті є вебсайти, портали, бази даних);
- системні адміністратори (які забезпечують роботу серверів, кластерів, систем зберігання даних);
- менеджери, які координують роботу;
- фінансові керівники.

Тобто не дивно, що у 3- або 5-річному проєкті, який закінчується якісною публікацією, беруть участь 5–20 дослідників різного профілю. У розвинених наукових спільнотах деякі спеціалісти (Sequencing Core, Bioinformatics Core) можуть бути зовнішніми експертами для лабораторії, запрошеними за контрактом.

В українських реаліях трапляється так, що всі ці функції виконують 2 людини — аспірант та науковий керівник, кожен з яких ще має «основну» роботу, як-от викладання та виживання, тому проєкт може затягнутися років на 5–10. А поділити функції теж не завжди можливо, бо:

- «нікого немає» — треба більше кваліфікованих науковців, більше зв'язків між ними;
- не можна довіряти — треба вирощувати репутації;
- немає фінансування;
- делегувати кому-небудь означає зізнатись, що я не все можу, але тягти ілюзію всемогутності можна аж до пенсії.

Якщо порівняти з українською, західна наукова спільнота дуже відрізняється: це справді мережа дослідників різного профілю, дуже спеціалізованих, насичена контактами, які в різних комбінаціях дають ефективні робочі групи для проєкту. Приблизно за таким самим принципом працює й біотех, тільки там взаємодіють підрозділи, компанії, консультанти, а проєкти можуть бути більш масштабними, наприклад, спрямованими на розробку нових ліків, проведення клінічних досліджень та вихід на ринок.

4 Біологія

Якщо ви прийшли до біоінформатики з біології, вам не потрібно розповідати, що саме являє собою біологія і який обсяг знань треба мати, щоб базово розуміти біологічні наукові статті. Але якщо ви прийшли з computer science, програмістів, математиків і особливо не вирізнялися знанням біології навіть у школі, то привчайте себе регулярно покращувати знання та вирощувати експертність і зацікавленість протягом усього життя. Зрештою, біоінформатика не існує сама по собі, а займається здебільшого дослідженнями біологічних об'єктів.

4.1 Початок

Почати можна з:

- Wiki — Biology;
- шкільних підручників (товсті американські підручники для старших класів (High School) дуже хороші, навіть ті, що продаються в супермаркеті підійдуть для початку: Everything you need to ace biology, аж до гарних підручників Biology: self-teaching guide by Garber, Campbell Biology;
- Biology for Dummies (у серії «для чайників» завжди дуже гарні та професійно написані книжки);
- освітнього ресурсу від журналу Nature — Scitable, особливо з клітинної біології;
- вступу до біології від MIT — 35 лекцій на YouTube.

Пройти тест на знання біології дуже легко — дивіться розділ «Журнальний клуб». Якщо ваших знань вистачає, щоб розуміти та обговорювати сучасні статті, то й добре, якщо ні — треба читати підручники, адже просто так від нуля до Nature Genetics ви не дотягнетесь. Також деякі біологи можуть бути здивовані, що сучасна біологія пішла дуже далеко від того, на чому зупинилось викладання на деяких кафедрах та факультетах. У такому разі теж доведеться наздоганяти, і сучасний підручник допоможе в цьому.

4.2 Мікроскоп

Я не буду всоте переповідати «ДНК для програмістів», про це написано будь-де. При вивченні біології для програміста головне, на мій погляд, не поспішати і мати терпіння: легко не буде. Засвойте шкільні знання, наприклад, рівні організації живої матерії, перш ніж лізти в нетрі. Біологія — це наука з мікроскопом, навіть тоді, коли ви ніколи не відвідаєте лабораторій чи польових досліджень. Під час будь-якого аналізу чи обговорення треба розуміти, на якому «збільшенні мікроскопа» ви перебуваєте: молекулярному (більш точно — ДНК, транкриптомі), клітинному, тканинному, рівні органів, організмів, популяційному, екологічному.

Відомий приклад, коли вчений не зауважив цього, — антиеволю-ційна книга видатного біохіміка Майкла Біхе Чорна скринька Дарвіна. У ній автор відкидає біологічну еволюцію (натомість пропонує ідею intelligent design) саме тому, що зачинив себе на молекулярному рівні, а еволюційний підхід потребує міркувати на рівні популяції та виду. Приблизно так само матеріалознавець не зміг би пояснити, як були розроблені сучасні авто, — для цього треба було б дивитися схеми та порівнювати їх у часі, а не дивитися в молекулярний мікроскоп на зразок гуми.

4.3 Біологічна еволюція

Другий важливий аспект біології після мікроскопу — це еволюційний підхід: «nothing in biology makes sense except in the light of evolution (ніщо в біології не має сенсу без опертя на еволюцію)», як писав видатний американський біолог українського походження XX сторіччя Феодосій Добжанський (див. вікі — стаття українською, Wiki — English, Britannica — English).

Еволюційний підхід: що він означає на практиці? «Еволюціо» латинською — це «розгортання». Біологічний об'єкт (на будь-якому рівні) ніколи не статичний, а радше є частиною якогось процесу, що розгортається в часі (прогрес хвороби або деградація РНК). Біологічні процеси ніколи не бувають ізольованими, найчастіше потрібно оцінити роль декількох процесів чи факторів. Наприклад, час, наявність щеплення, наявніть мутації, застосування ліків у клінічному дослідженні. Або безпосередьо в еволюційній біології: природний відбір, мутації, генетичний дрейф, міграція, які впливають на розподіл частот алелів у популяції. Дослідник найчастіше має тільки деякі окремі виміри біологічного об'єкта (наприклад, транскриптоми на 1-й, 5-й, 10-й день експерименту з клітинними лініями), і за цими вимірами намагається оцінити хід усього процесу.

Окрім принципу динамізму та факторів, еволюційний підхід майже завжди дає цікаві ідеї, якщо просто подивитися на об'єкт вашого дослідження на еволюційному дереві: ось ви вивчаєте Malignant hyperthermia — рідкісне захворювання людини, за якого загальна анестезія під час операції викликає підвищену температуру з ризиком для життя, потім з'ясовується, що ті ж самі мутації посилення функції (gain of function) в гені *RYR1* псують м'ясо свиней, що відразу дає вам ідеї щодо пошуку консервативних сайтів у цьому гені для усіх ссавців, а також animal model для вивчення хвороби.

В еру COVID-19 усі вже бачили еволюційні (філогенетичні) дерева: початкова лінія ковіду змінювалась, ми пройшли штами Alpha, Beta, Gamma, Delta (найнебезпечніший штам), почалося масове щеплення проти Alpha, з'явився менш летальний Omicron. Майже всі засвоїли базові еволюційні поняття популяції, мінливості, мутації, відбору, але буде не зайве їх відновити за допомогою підручника.

Еволюційна концепція адаптивного ландшафту — це велика тема (див. fitness landscape). Вона дозволяє розглядати біологічні процеси як рух у просторі, де фенотип визначається як функція на просторі всіх можливих генотипів. Метафора адаптивного ландшафту дуже продуктивна і докладається до багатьох тем — від

популяційної генетики до генетики видоутворення, екології, біології розвитку. Для математиків адаптивний ландшафт цікавий тим, що деякі біологічні проблеми можна формалізувати як проблеми оптимізації.

Інша підказка — поповнювати знання з сучасної медицини, яка дуже часто перетинається з сучасною біологією. WebMD та Mayo-Clinic вам у поміч!

4.4 Книжки

Далі просто посилання на книжки з біології, які я читав і читаю або які хоча б погортав. Вони мені допомагають у роботі (корисно мати невеличку бібліотеку на кафедрі, в лабораторії, на факультеті, в компанії):

- Unraveling DNA by Maxim D Frank Kamenetskii;
- What is Life by Erwin Schrodinger;
- Genetics and Genomics in Medicine by Tom Strachan;
- Evolution by Bergstrom and Dugatkin;
- Essential Genetics and Genomics by Daniel Hartl;
- Lewin's Genes XII+ — це підручник, який існує років 50 і з часом оновлюється;
- Origin of Genome Architecture by Michael Lynch;
- Fundamentals of Molecular Evolution by Graur and Li;
- Population Genetics by Hamilton;
- Crumbling Genome by A. Kondrashov;
- Molecular Biology of the Cell — є переклад українською;
- Molecular Biology of the Gene;
- The Neutral Theory of Molecular Evolution by Kimura;
- On the origin of species by Darwin;
- Elements of Evolutionary Genetics by B and D Charlesworth;
- She Has Her Mother's Laugh, а також інші книжки Карла Циммера;
- Популярні книжки з біології від goodreads.com;
- Oxford IB Diploma Programme: Biology Course Companion.

5 R, Rstudio, візуалізація, статистика

5.1 R/Rstudio/Tidyverse

Якщо є одна технологія, за допомогою якої можна швидко просунутися в біоінформатиці, а потім вже надолужити інші, то це буде R/Rstudio/tidyverse/Bioconductor. Отже, не одна, а цілих чотири технології. Якщо потрібна одна, то це буде Rstudio. Нещодавно її назву змінили на Posit, щоб підкреслити можливість працювати на R та Python.

R — функціональна мова, де майже будь-яка конструкція — це функція з параметрами. Це мова високого рівня, близька за рівнем абстракції до математичних пакетів Mathematica, Matlab, Maple, але спеціально сфокусована на статистиці. Тому її не люблять програмісти, бо насправді це не мова програмування, а мова опису статистичних функцій. З іншого боку, R/Rstudio легко засвоїти буть-якому науковцю й інженеру і необов'язково мати освіту чи здібності програміста.

Безкоштовні R/Rstudio можуть стати пропуском у світ професійного програміста-статистика, який також володіє SAS and SPSS.

R має прості базові структури даних: вектор, матриця (всі дані однакового типу), датафрейм (таблиця зі стовпчиками різного типу). На цьому базисі побудовано цілий світ: спеціальні матриці для зберігання великих обсягів даних, спеціальні об'єкти, подібні до датафреймів для зберігання даних секвенування SummarizedExperiment.

Для початкового засвоєння R/Rstudio пропрацюйте уроки за datacarpentry: R-ecology-lesson, R from HBC або R crash course.

Tidyverse — це декілька бібліотек, що надали мові R «людського обличчя»: концепцію tidy data (охайні дані), комбінацію функцій у пайплайни, розвинені бібліотеки для маніпулювання даними, візуалізацію ggplot2. Немає сенсу окремо вчити R та відкладати вивчення tidyverse, бо можна відразу почати з безкоштовної книги R for data science.

Розвиватись у напрямку R/tidyverse теж можна. Хоча спочатку здається, що R — мова проста, але насправді для того, щоб писати пакети з тестами, debug, документацією, підтримувати колекції пакетів, вписувати їх до екосистеми, потрібно багато знань, і це може стати професією.

Деякі з advanced R topics:

- R packages by Hadley Wickham and Jenny Brian;
- Advanced R by Hadley Wickkam;
- Testing R code by Richard Cotton;
- R programming for Bioinformatics by R.Gentleman.

5.2 Bioconductor

R/Bioconductor містить понад 2 000 спеціалізованих R-пакетів для біоінформатиків. Це велика екосистема разом з документацією та навчальними курсами.

Пакети можна подивитися за рангом. Серед найпопулярніших — пакети аналізу диференційної експресії генів: limma, edgeR, DESeq2. Цей аналіз, імовірно, найбільш поширений зі стандартних у NGS-біоінформатиці.

Екосистема Python теж має багато пакетів, але дуже велика кількість пакетів у R/Bioconductor не має аналогів у системі Python. Деякі пакети Bioconductor дуже спеціалізовані: найкращі спеціалісти у світі у вузькій галузі вклали понад 10 років у розробку та підтримку пакета, наприклад PureCN для copy number analysis у ракових зразках, що секвеновані за допомогою high coverage gene panels (генні панелі високого покриття). Звичайно, буває і навпаки, коли пакет у екосистемі Python не має аналогів у R/Bioconductor: наприклад, пакет Hotpot для визначення інформативних генів в Single Cell RNA-seq датасетах. Тому біоінформатики часто балансують між двома екосистемами, експортують дані з одних пакетів до інших, вибудовують пайплайни.

5.3 Візуалізація

Програмування графіки за допомогою R/ggplot2 — це стандартний засіб для створення ілюстрацій до наукових публікацій, звітів, семінарів. Створення якісної графіки потребує наполегливого вивченння основ:

- книга Клауса Вілке Fundamental of Data Visualization;
- ggplot2: Elegant graphics for data analysis by Hadley Wickham;
- https://r-graph-gallery.com/ — містить багато прикладів типових графіків;
- R graphics cookbook by Winston Chang.

Графіки та аналіз у R можна поєднувати в документи Markdown (див. також https://www.markdownguide.org/cheat-sheet/) та Bookdown. Їх можна компілювати в pdf чи html-документи. Markdown-документи — це не тільки інструмент оформлення звітів та лабораторних журналів, а й потужний засіб відтворюваності (reproducibility), наразі з Jupyter Notebook для аналізів з Python. Ноутбуки також часто використовують для розробки навчальних матеріалів.

Найчастіше при підготовці ілюстрацій для статті треба комбінувати декілька зображень у панелі, для чого є пакети cowplot, patchwork.

Іноді навіть стандартних засобів ggplot бракує для спеціальних ілюстрацій, тоді використовують спеціалізовані пакети:

- heatmaps для експресії генів: pheatmap;
- oncoprints для ракових мутацій (діаграми «пацієнт-мутація-ефект»): ComplexHeatmap;
- lattice — коли треба намалювати багато точок.

5.4 Статистика

Не є таємницею, що багато хто з біоінформатиків не знає статистики, а покладається на пакети, в яких імплементовані потужні статистичні методи. Але

все ж таки треба покращувати знання зі статистики, хоча б для того, щоб розуміти, які методи треба застосовувати до яких даних та як інтерпретувати результати. Особливо це стосується теорії похибок та p-values (значущість).

Покращувати ці знання легко за допомогою Stat Quest from Josh Starmer. Це коротенькі відео, в яких пояснюються часом дуже складні статистичні концепції.

Корисно й перечитувати підручники з біологічної статистики. Ось деякі гарні:

- Statistics for the Life Sciences;
- Susan Holmes and Wolfgang Humber — Modern Statistics for Modern Biology;
- Rosner — Fundamentals of Biostatistics;
- Chap T.Le, Lynn E. Eberly — Introductory Biostatistics;
- Pagano, Gauvreau — Principles of biostatistics;
- An introduction to statistical learning with Applications in R;
- Introduction to data science by Rafael A. Irizarry;
- Richard McElreath — Statistical Rethinking (не пов'язаний з біологією).

Якщо вас цікавить саме статистика, то вона відкриває дорогу до популярного Machine Learning.

6 Python та програмування

6.1 Програмування — це професія

Повна освіта програміста приблизно охоплює:

- дискретну математику: https://en.wikipedia.org/wiki/Discrete_mathematics, тобто логіку, теорію графів, теорію інформації, комбінаторику, теорію множин та ін.;
- середовища розробки (IDE): https://en.wikipedia.org/wiki/Integrated_development_environment;
- добре знання декількох мов програмування (C/C++, Java, Python): https://www.oreilly.com/search/?q=python%20c%252B%252B%20java;
- алгоритми: https://mitpress.mit.edu/9780262533058/introduction-to-algorithms/;
- структури даних: https://www.geeksforgeeks.org/data-structures/;
- методології розробки, керування проєктом: https://www.tutorialspoint.com/software_engineering/software_project_management.htm;
- проєктування: https://en.wikipedia.org/wiki/Software_design;
- тестування: https://www.ibm.com/topics/software-testing;
- розробку візуальних інтерфейсів;
- веброзробку;
- бази даних;
- мережі, протоколи;
- машинне навчання, штучний інтелект;
- операційні системи;
- системне програмування;
- програмування на асемблері;
- розробку для мобільних платформ;
- компіляцію, синтаксичний аналіз;
- архітектуру комп'ютерів та обчислювальних систем;
- файлові системи;
- історію комп'ютерних наук;
- функціональне програмування;
- об'єктно-орієнтоване програмування;
- паралельне програмування, multithreading;
- математичну теорію складності обчислень;
- криптографію;
- теорію кодів;
- чисельні методи;
- комп'ютерну графіку;
- хмарні платформи, мікросервіси.

Більшість цих курсів можно прослухати від світового лідера технічної освіти Massachusetts Institute of Technology (MIT): https://github.com/Developer-Y/cs-

video-courses.

6.2 Мінімум для біоінформатика

Біоінформатик, якщо він не приходить з програмістів, не має змоги опанувати всі ці курси. Мінімум для біоінформатика з програмування такий:

- основи написання програм: скриптові мови, оформлення коду програм, документація, GitHub, debug;
- основи комп'ютингу: процесори, процеси, файли, файлові системи, користувачі, пам'ять, мережі;
- добре знання однієї скриптової мови, сьогодні це переважно python;
- знання бібліотек: biopython, scipy, numpy, matplotlib, scanpy.

6.3 Learning Python

Такі ресурси та книги допомагають опанувати python:

- https://docs.python.org/3/tutorial/ — офіційна документація;
- https://www.w3schools.com/python/ — курс від W3C;
- https://ocw.mit.edu/courses/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/ — відеокурс від MIT;
- https://www.oreilly.com/library/view/learning-python-5th/ — Learning Python;
- https://www.oreilly.com/library/view/python-cookbook-3rd/9781449357337/ — Python Cookbook;
- https://www.oreilly.com/library/view/python-for-data/9781491957653/ — Python for Data Analysis;
- https://effectivepython.com/ — Effective Python by M. Slatkin;
- https://biopython.org/;
- https://scipy.org/;
- https://numpy.org/;
- https://matplotlib.org/;
- Mastering Python for Bioinformatics: https://www.oreilly.com/library/view/mastering-python-for/9781098100872/.

Головне у навчанні програмування — це практика. Будь-який гарний підручник проводить вас повз написання великої кількості скриптів на різні теми. Добре, якщо ви вже маєте проєкт, над яким працюєте, і можете використовувати в ньому приклади скриптів.

6.4 Онлайн-тренування

Дуже розвиває будь-якого junior-програміста розв'язання 100 простих задач на одному з серверів ACM: https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8.

Кожна задача вимагає написати невелику програму, яка повинна пройти валідацію з 10 тестів. У співбесіди з обіймання багатьох посад, особливо в індустрії, впроваджують подібні задачі, наприклад, у системі HackerRank.

6.5 GitHub

Git — це система керування версіями програмного коду (технологія, api) й водночас GitHub — це сервер, на якому можна зберігати свій програмний код (у приватних чи публічних репозиторіях). Репозиторій — це одиниця зберігання програмного коду, зазвичай одного проєкту, який може містити сотні файлів. GitHub є безкоштовним (дякуючи спонсорству Microsoft).

Потрібно задіяти акаунт у GitHub, засвоїти базові операції git і комітити свій код, знаходити проєкти з біоінформатики, в які можна зробити свій внесок. По-перше, необхідно знаходити та розв'язувати проблеми (issues), а потім розробити code contribution. GitHub — це соціальна мережа для програмістів, де можна знаходити проєкти й окремих розробників, які вас цікавлять. Профіль GitHub — це частина портфоліо біоінформатика. На співбесіді вас обов'язково запитають, чи є у вас такий профіль, чи можна подивитись на ваш програмний код і чи зробили ви внесок у якийсь відомий проєкт.

GitHub може здаватися дуже складним, але насправді для початку достатньо засвоїти створення репозиторія програмного коду (онлайн у браузері) та прості операції: git clone, git status, git remote, git add, git commit, git push, git pull:

- https://docs.github.com/en/get-started/quickstart/hello-world;
- https://git-scm.com/book/en/v2 — Pro Git book;
- https://github.com/git-tips/tips;
- https://www.gitkraken.com/.

GitHub також корисний для керівника на рівні лабораторії. Якщо всі проєкти у лабораторії зберігаються в GitHub, то дуже легко стежити за кодом, навіть коли співробітники залишають лабораторію і приєднуються нові. Також, якщо зберігати всі скрипти аналізів проєкту в приватному репозиторії GitHub, то буде дуже легко відкрити код при публікації статті.

Великі компанії створюють свої git-сервери, наприклад, за допомогою GitLab.

6.6 IDE, Editor

Професійні програмісти зазвичай працюють у середовищі розробки (IDE), що підвищує якість роботи та продуктивність: PyCharm, Eclipse. Середовища розробки також мають інтеграцію з Git та візуальний інтерфейс для роботи з комітами. Біоінформатики, які переважно працюють над скриптами, часто обирають просунуті текстові редактори Visual Studio Code, Sublime, mc, Emacs, або Jupyter notebooks.

7 Linux, HPC, Cloud

7.1 Навіщо Linux

Дані NGS мають великий обсяг (гігабайти, терабайти). Програми, які їх обробляють, теж часто не можна запустити на лептопі, тому NGS-біоінформатики значну частину часу проводять у командному рядку (Command Line Interface, CLI) на серверах, кластерах, у хмарі. Базове знання Linux допомагає розібратися з будь-яким термінальним інтерфейсом.

Один зі способів швидкого навчання Linux — це встановити Fedora Linux або Ubuntu Linux на десктопі чи лептопі. У 2023 році інтерфейси та програми вже не дуже відрізняються за якістю порівняно з MacOS та Windows (звісно, системи Open Source часто мають гірші інтерфейси та менше функцій, якщо порівняти з корпоративними розробками).

У Linux треба зрозуміти базові поняття терміналу, файлу, процесу, користувача, прав доступу (див. digital ocean), вивчити базові команди (cd, ls, pwd, readlink, du, df, mv, mkdir, rm, rmdir, cat, gunzip, zcat, diff, grep, ps, top, head, tail, |, man, touch, tar, gzip, ssh, comm, kill, chmod, chown, wget, sudo), навчитися писати прості скрипти (це корисно, навіть якщо ваша основна мова програмування — Python), навчитися працювати на кластері разом із сотнями користувачів: bash intro from HBC.

7.2 Термінальні редактори

Треба навчитися працювати в одному з термінальних текстових редакторів, щоб редагувати текстові файли (скрипти, конфігураційні файли) на серверах:

- mc — це одночасно редактор, файл-браузер та ftp-клієнт;
- emacs;
- vi.

7.3 One-liners

Робота над текстовими файлами у Linux часто потребує написання one-liner (однорядкових команд) з використанням awk, sed, perl, regular expressions:

- https://www.theunixschool.com/p/awk-sed.html;
- mastering regular expressions;
- https://learnbyexample.github.io/learn_perl_oneliners/cover.html.

7.4 Книги та підручники

- https://www.w3schools.io/terminal/bash-tutorials/ — вступ до bash від W3C;
- https://mywiki.wooledge.org/BashPitfalls — часті помилки в bash-скриптах;
- https://hbctraining.github.io/Intro-to-Shell/ — вступ до bash від HBC;

- https://www.amazon.com/UNIX-Linux-System-Administration-Handbook/dp/0134277554 — підручник системного адміністратора Linux;
- https://www.youtube.com/@MissingSemester — «загублений семестр» від MIT;
- https://www.digitalocean.com/community/tutorials/how-to-set-up-ssh-keys-2 — робота з SSH та ключами;
- https://www.oreilly.com/search/?q=linux — книжки від видавництва О'Рейлі;
- Fedora Linux: https://fedoramagazine.org/, https://docs.fedoraproject.org/, https://getfedora.org/;
- Ubuntu Linux: https://ubuntu.com/;
- RHEL derivatives — Linux-дистрибутиви для промислових систем;
- The Linux Command Line by William Shotts.

7.5 Сім CLI-інструментів та чотири формати даних

Кожен із цих інструментів містить декілька функцій, та ще й з параметрами, а gatk/picard містить пару сотень функцій:

- blast — найстаріший інструмент біоінформатики з пошуку послідовностей у базі даних. Можна запускати вебінтерфейс blast для малих пошуків, для великого обсягу даних треба розгорнути blast на кластері чи сервері. Жартують, що не можна стати біоінформатиком, аж доки ви не розробите свій парсер результатів бласт;
- EMBOSS — інструменти від EMBL;
- JKent Utils — CLI-інструменти від розробників UCSC genome browser;
- samtools — для роботи з BAM-файлами;
- bcftools — для роботи з VCF-файлами;
- bedtools — для роботи з BED-файлами;
- gatk та picard — для роботи з VCF-, BED-, BAM-файлами.

Відповідні формати даних теж треба вивчати — див. вікі-сторінки:

- https://en.wikipedia.org/wiki/Binary_Alignment_Map;
- https://en.wikipedia.org/wiki/BED_(file_format);
- https://en.wikipedia.org/wiki/Variant_Call_Format;
- https://en.wikipedia.org/wiki/FASTQ_format.

7.6 HPC

Зазвичай біоінформатики працюють на комп'ютерних кластерах (High Performance Computer), серверах, центрах обробки даних (data center), хмарних системах. На такому обчислювальному ресурсі можуть працювати десятки і навіть тисячі дослідників. Приклад великого центру — Harvard Medical School Research Computing: O2, FAS RC. Для того, щоб працювати на кластері, треба засвоїти

batch job submission system (slurm, pbs, torque — залежно від кластера). Slurm — найпопулярніша система:

- https://harvardmed.atlassian.net/wiki/spaces/O2/pages/1586793632/Using+Slurm+Basic;
- https://slurm.schedmd.com/quickstart.html.

Для роботи на видалених системах потрібні термінальні програми:

- ssh-термінал у Linux;
- https://www.nomachine.com/;
- https://www.putty.org/ — для Windows.

Корпоративні системи можуть використовувати VPN-клієнти та Citrix, які дозволяють приєднатися до корпоративної мережі.

7.7 Хмарні системи (cloud)

Cloud — це велика ферма серверів та систем зберігання даних, доступ до якої продається за обсягом даних, часом обчислень, кількістю віртуальних машин, тобто platform as a service, compute as as service, storage as a service.

Є 3 головні провайдери хмарних послуг:

- Amazon Web Services — AWS;
- Google Cloud Platform — GCP;
- Microsoft Azure.

Кожна платформа — це цілий світ зі своїми правилами, архітектурами, API, але Billing об'єднує всі платформи, і за послуги треба сплачувати. Спочатку ви приєднуєте billing account, після чого рахунки за зберігання даних, завантаження та розрахунки можуть коштувати до сотень доларів на місяць за прості обчислення і до мільйонів доларів на місяць — у разі роботи великого підрозділу. Робота у хмарі потребує точного розуміння, скільки кожна операція коштуватиме. Оптимізація витрат — одне із завдань хмарного архітектора. До команди біоінформатиків, розробників та адміністраторів даних у великому хмарному проєкті додаються архітектори, спеціалісти безпеки, спеціалісти dev ops.

З одного боку, хмарні обчислення ускладнюють і без того складний процес. З іншого — хмарні платформи дозволяють отримати якісне обчислювальне середовище без вкладання грошей у фізичну інфраструктуру. Під час війни саме хмарні сервіси відіграють значну роль в Україні для підтримання банківських та державних послуг, коли фізична інфраструктура зазнає руйнувань під час обстрілів.

Хмарні системи легко масштабуються, якщо ваш проєкт росте за рік у рази чи десятки разів або має сильні коливання обчислень, обсягів даних. У хмарному середовищі можна адаптуватися, в той час як масшбатувати фізичну інфраструктуру не так легко.

7.8 Залізо чи хмара?

Використовувати чи не використовувати хмару в науковому проєкті чи компанії і яку саме хмарну платформу обрати — надважливі питання.

Деякі фактори, що можна розглянути:

- які реальні обсяги обчислень і даних вам потрібні на наступні 1–3–5–10 років? Можливо, вистачить просто сервера з дисковою поличкою?
- Яка динаміка попиту вашої лабораторії на обчислення за роками?
- Які ваші фінанси на обладнання чи хмару (буває, що є великі фінанси цьогоріч, а на наступні роки не буде — тоді можна інвестувати саме в обладнання)?
- Чи є якісь пільги від університету (інколи є майже безкоштовне підвальне приміщення, електрика)?
- Які ваші ресурси на підтримання інфрастуктури (хмара та сервери-кластери потребують зусиль до адміністрування, однак це різні компетенції)?
- Чи можна об'єднатися з кимось та розділити витрати на інфрастуктуру?
- Чи можна знайти спонсора (іноді хмарні провайдери можуть надати доступ безкоштовно, якщо ви використаєте їхню платформу і сприятимете розвитку їхнього API, а для них ваші обсяги обчислень можуть бути незначними)?
- Якщо ви розділяєте інфраструктуру разом з обчислювальним центром чи іншою лабораторією, майте на увазі, що ваші потреби і потреби інших користувачів можуть бути протилежними (вам куплять, наприклад, кластер з дорогою високошвидкісною мережею та сховищем лише на 100T, а насправді вам не потрібна швидкісна мережа, натомість потрібне сховище на 1PB, або вам потрібні сервери з великим обсягом пам'яті — 512G, 2T, які не є стандартними, або вам запропонують велике сховище даних, проте яке неспроможне витримувати одночасно читання-запис від багатьох серверів, тоді як вам потрібне велике сховище, можливо, з меншою швидкістю доступу, але з широким bandwidth).

8 Три геномних браузери

Візуалізація в біоінформатиці дуже важлива. Найчастіше (чи не завжди?) проєкт складається з багатьох ітерацій: порахували — подивились. Окрім засобів візуалізації, розглянутих у розділі про R, часто потрібно подивитися на вирівнювання, покриття чи варіанти або розглянути детально анотацію того чи того локусу генома.

- Integrative Genomics Viewer (IGV) — десктопна програма на Java, в яку можна завантажувати файли bam, vcf, bed, bigWig і розглядати окремі гени та локуси. Є реалізації браузера на JavaScript для використання у вебзастосунках.
- Ensembl genome browser — геномний браузер для різних видів, зокрема людини та миші.
- UCSC genome browser — геномний браузер від UCSC, містить багато референсних геномів та повногеномні множинні вирівнювання.

9 Пайплайни

9.1 Що таке пайплайн

Будь-яка реальна обробка NGS-даних потребує багатьох обчислювальних кроків, які становлять пайплайн, або workflow. Класичний приклад — це пайплайн для визначення малих germline-варіантів та інделів від Broad Institute: https://gatk.broadinstitute.org/hc/en-us/articles/360035535932-Germline-short-variant-discovery-SNPs-Indels-.

Під час навчання можна виконати всі кроки пайплайну в командному рядку, але щоб обраховувати велику кількість зразків постійно та вносити корективи у процес, треба імплементувати пайплайн щонайменше як bash- або python-скрипт. Біоінформатики проводять велику кількість часу, створюючи та підтримуючи пайплайни від маленьких пайплайнів в одному проєкті до production-пайплайнів, які обслуговують команди біоінформатиків та програмістів. Для досконалішої імплементації пайплайнів застосовують workflow management systems, тобто дозволяють відділити власне опис біоінформатичних кроків пайплайну від технічних подробиць завантаження задач на кластері чи у хмарі.

Типові інструменти та стадії NGS-пайплайнів щодо аналізу варіантів:

- https://genomemedicine.biomedcentral.com/articles/10.1186/s13073-020-00791-w/tables/2;
- https://genomemedicine.biomedcentral.com/articles/10.1186/s13073-020-00791-w/figures/1.

9.2 Broad Institute: GATK, Terra

Broad Institute of MIT and Harvard https://www.broadinstitute.org/ — це одна з провідних наукових організацій у галузі біомедицини, що розташована в Бостоні. Під час пандемії COVID-19 Broad Institute швидко запустив платформу тестування на ковід та протестував понад 37 млн зразків (https://covid19-testing.broadinstitute.org/).

Екосистема GATK (Genome Analysis Toolkit) — один із найвідоміших продуктів Broad Institute, який забезпечив лідерство в галузі визначення малих варіантів. GATK підтримує як соматичні варіанти (mutect2), так і гермінальні (germline) (gatk-haplotype):

- https://gatk.broadinstitute.org/hc/en-us;
- https://github.com/broadinstitute/gatk.

Крім окремих інструментів, як-от mutect, Broad Institute відкрив свої перевірені практики (Best Practices), які дозволяють маленьким лабораторіям обробляти дані за стандартами Broad Institute. Для розуміння типового біоінформатичного пайплайну можна взяти одну з Best Practices та написати скрипт bash, python або snakemake, який імплементує всі кроки, наприклад,

https://github.com/naumenko-sa/bioscripts/blob/master/scripts/go.gatk.all.pbs.
Реальні пайплайни, особливо ті, які використовуються у production великими компаніями, набагато складніші й можуть включати сотні кроків та тисячі утиліт.

Спочатку GATK був доступний тільки для академічних проєктів, а для індустрії була потрібна досить дорога ліцензія, яка обмежувала використання, але, починаючи з GATK4, всі утиліти стали open source. Також проєкт picard (здебільшого утиліти обробки bam-файлів) було інтегровано до GATK.

GATK підтримується великим колективом програмістів, біоінформатиків, учених. Щоб бути в курсі вдосконалення інструментів, треба стежити за форумами та новими статтями.

Terra — це хмарна платформа від Broad Institute, що побудована на базі Google Cloud, яка містить основні gatk-based workflows. Terra потребує billing account у Google Cloud, тобто, як і зазвичай у хмарних платформах, треба стежити за коштами. Аналіз даних може бути не таким дорогим: 1–2 тис. доларів за 300 T/N (tumor / normal) ракових WES-зразків, але сума може зрости швидко, якщо «забути» дані у хмарі (за кожен день перебування великого обсягу даних у хмарі доведеться платити). Також треба враховувати кошти на завантаження даних у хмару та вивантаження з хмари (що може бути дорого) до сховища даних на постійне зберігання. Vcf-файли з варіантами не такі великі, але збереження даних bam, cram щодо сотні зразків може призвести до великих витрат. Щоб оцінити витрати на великий проєкт, завжди виконують pilot — маленький проєкт з реальними даними, дивляться на витрати і прогнозують їх у великому проєкті. У великих компаніях, коли процесинг даних постійно відбувається у хмарі, витрати оптимізують та беруть під контроль.

- https://terra.bio/;
- (login) https://app.terra.bio/#library/showcase.

Terra має дуже простий вебінтерфейс — одна людина може обробити великі масиви даних. Необхідно провести такі кроки для Tumor/Normal WES пайплайну (приблизно):

- створити bucket для вхідних даних, якщо вони ще не у Google Cloud (https://console.cloud.google.com, gsutil);
- створити рахунки для проєкту та налаштувати білінг;
- клонувати workspace fastq to uBam, створити таблицю, яка описує дані, виконати workspace для таблиці;
- виконати uBam to Bam workflow (alignment);
- виконати Somatic-SNV-Indels-GATK4 workflow.

Виконання будь-якого workflow у Terra потребує таких кроків:

- клонувати його (він починає використовувати вашу квоту в google-хмарі й починається білінг);

- створити таблицю-датафрейм, яка описує дані (наприклад, вхідні fastq, вихідні bam);
- завантажити таблицю до Terra;
- налаштувати параметри workflow (їх може бути чимало й треба їх зрозуміти, щоб потім не виконувати аналіз ще раз. Якщо не впевнені — запустіть workflow для одного-двох зразків та подивіться на результат, запитуйте tech support — вони добре відповідають);
- запустити workflow для всіх зразків у таблиці;
- стежити за виконанням, деякі розрахунки потребують перезапуску з більшим обсягом пам'яті;
- стежити за витратами;
- видалити дані, які не потрібні, щоб оптимізувати білінг;
- передати дані на інший workflow, якщо треба, або зберегти дані в постійне сховище;
- звільнити місце в Google Cloud, підсумувати витрати.

9.3 Illumina Dragen

- https://support-docs.illumina.com/SW/DRAGEN_v310/Content/SW/FrontPages/DRAGEN.htm;
- https://github.com/naumenko-sa/bioscripts/tree/master/dragen.

Dragen — це FPGA-процесор, який дуже потужно прискорює read mapping та визначення варіантів. Dragen обробляє один повний генотип людини (30X WGS) за 20–30 хвилин порівняно з годинами чи десятками годин на GATK-based пайплайнах. Dragen існує багато років, але спочатку його застосування було обмеженим, аж поки компанію не придбала Illumina і почалася співпраця з Broad Institute, яка спрямована на досягнення еквівалентності пайплайнів GATK and Dragen.

У 2021–2022 роках Dragen став визнаним інструментом, який був валідований багато разів для визначення варіантів, зокрема в лабораторіях клінічної діагностики.

Dragen — це не відкрита платформа: можна купувати розрахунки у хмарі AWS, можна купувати сервери з платами Dragen та встановлювати їх у дата-центрах. При формуванні бюджету треба знати, що навіть купівля Dragen не дасть вам можливості необмежених розрахунків — кожна ліцензія має обмеження щодо обсягу обробки даних, наприклад, 600 зразків генотипів людини на рік.

9.4 NextFlow, NF-Core collection

- https://www.nextflow.io/
- https://nf-co.re/pipelines

NextFlow — це спеціалізована мова програмування та середовище для розробки пайплайнів, яку підтримує компанія SeqeraLabs. В екосистемі дуже

багато гарних open source пайплайнів, які підтримують фахівці провідних лабораторій. Технологічно платформа NextFlow розроблена дуже якісно. Бізнес компанії побудований навколо NextFlow Tower — продукту, що дозволяє будувати production environment у хмарі. Тобто для маленької наукової лабораторії чи компанії можна просто використовувати пайплайни NF-core безкоштовно.

9.5 Академічні пайплайни

Порівняно з компаніями, які мають стабільні команди та професійних програмістів, академічні пайплайни зазвичай менш конкурентоспроможні, але вони теж мають свої сильні сторони, наприклад, містять унікальні вузькоспеціалізовані пайплайни.

Деякі з академічних пайплайнів та систем:

- https://snakemake.readthedocs.io/en/stable/ — це найпростіший, але дуже потужний замінник bash- чи python-скриптів.
- https://snakemake.github.io/snakemake-workflow-catalog/ -— каталог snakemake-пайплайнів.

Пайплайни на snakemake дозволяють відокремити опис технічних параметрів від біоінформатичних кроків пайплайну. Також середовище завантаження задач відокремлене від логіки пайплайну.

- Bcbio (US, Boston): https://bcbio-nextgen.readthedocs.io/en/latest/, https://github.com/bcbio/bcbio-nextgen/, Bcbio intro (30 хв): https://docs.google.com/presentation/d/1rhE5T8_kDTQo96fZQqXsdwFvoy4SuSk6wSjYBL3_AhA/edit?usp=sharing;
- GenPipes (Canada, Montreal): https://bitbucket.org/mugqic/genpipes/src/master/, https://genpipes.readthedocs.io/en/genpipes-v4.1.0/;
- BPipe (Australia): http://docs.bpipe.org/.

9.6 CWL, WDL

Розглядаючи пайплайни, не можна уникнути теми workflow languages. Окрім NextFlow та Snakemake, це щонайменше:

- CWL: https://doc.arvados.org/rnaseq-cwl-training/01-introduction/index.html, https://www.youtube.com/watch?v=JYo0_kE3_L8;
- WDL (Broad Institute, Terra): https://support.terra.bio/hc/en-us/articles/360037117492-Overview-Getting-started-with-WDL;
- OpenWDL: https://openwdl.org/.

Головна ідея така сама: зробити спеціалізовану мову опису пайплайнів, щоб відокремити технічні подробиці виконання пайплайну на HPC або у хмарі від власне біоінформатичного опису кроків пайплайну.

На практиці треба мати якусь систему пайплайнів у лабораторії та підтримувати її, а не писати пайплайни щоразу спочатку новими мовами. Яку саме систему обрати, залежить від лабораторії, і це важливий вибір, який залежить від обсягу даних, бюджетів, наявності спеціалістів та інших чинників. Краще прагнути до використання upstream-систем — систем, які вже були розроблені більш досвідченими фахівцями у компаніях чи академічних лабораторіях, щоб виконувати аналіз даних на загальноприйнятому світовому рівні.

10 Журнальний клуб

10.1 Навіщо ЖК і типи ЖК

Журнальний клуб (ЖК) необхідний, щоб триматися в курсі сучасної наукової літератури. Зазвичай значно легше читати та обговорювати статті гуртом: декілька учасників приносять одразу багато статей, оскільки одна людина неспроможна прочитати їх самотужки, а також напрацьовується звичка зрозумілого викладу статті.

Формати ЖК можуть бути дуже різними, наприклад:

- за фіксованим часом за ланчем — 1 година;
- за фіксованим часом без ланчу — 1–2 години;
- з необмеженим часом — може розтягуватися до 2–3 годин, якщо вистачає матеріалу;
- за ставками: кожен учасник приносить статтю і ставить 1 долар, потім по колу кожен рекламує статю за 1–2 хв., керівник семінару записує коротенько на дошці, потім учасники голосують, яку статтю обрати для детального розгляду наступного разу. Переможець отримує гроші й наступного разу доповідає статтю, яку всі вже прочитають;
- за системою: секретар семінару веде записи, які статті було розглянуто протягом року (секретарі можуть змінюватися раз на рік), записує все в журнал, який доступний усім;
- спеціалізований ЖК — розглядаються тільки статті за напрямком лабораторії;
- загальний ЖК — розглядаються будь-які цікаві статті;
- розгляд статей з основних журналів: на початку сезону (семестру) кожен учасник обирає 1–2 журнали, за якими він буде стежити та доповідати, що в них було цікавого.

Відсутність ЖК в науковій лабораторії — це поганий маркер, який свідчить, що людям не цікава наука або вони неспроможні зрозуміти сучасні статті. Якщо маєте вибір, то краще обирати лабораторію, де є ЖК та регулярні обговорення.

10.2 Paywall та open access

Де брати статті? Зараз постійно збільшується кількість статей у відкритому доступі на сайтах журналів, у вигляді препринтів. Якщо ви не можете дістати статтю через paywall, то можна пошукати у препринт-серверах, а також написати авторам (обирайте молодших — вони отримують менше імейлів), більшість учених просто надішле вам pdf. На Заході університет чи інша наукова установа має доступ до журналів через бібліотеку. Інколи керівник лабораторії повинен підписати заяву, щоб надати своїм співробітникам доступ.

Все ж таки багато статтей ще за paywall. Постійно тривають дискусії, що це неправильно (більшість статей — результат проєктів, які виконано на державні

кошти, тобто гроші громадян, які сплачують податки, а отже, результати повинні бути відкритими для всього суспільства, а не приносити прибуток комусь). Деякі журнали вже повністю відкриті (щоправда, вони беруть плату з авторів): Elife, PLOS.

У топ-журналах, як-от журнали Nature, за відкритий доступ авторам доводиться платити досить дорого (див. ціни). Зазвичай отримати необхідну кількість статей для ЖК — це не проблема, особливо якщо мати бібліотеку статей на кафедрі / у підрозділі. Якщо стаття потрібна, а її ніяк не вдається дістати, вчені інколи крадуть статтю на всім відомому ресурсі, який зберігає тексти статей і має велику кількість логінів до бібліотечних систем, запроваджених прихильниками вільного доступу до журналів. Усе ж таки цивілізований доступ до текстів може бути організований на рівні кафедри, факультету, університету за рахунок грантів, спільної праці разом із західними університетами.

Останніми роками доступ до 300 журналів Elsevier був заблокований на декілька місяців у Німеччині та Нідерландах, тому що мережа університетів не хотіла платити за дорогі підписки. Тиск на видавців, які отримують надприбутки за рахунок праці вчених та безкоштовної праці рецензентів, зростає щороку, й немає сумніву, що проблему доступу до статей буде розв'язано.

Не треба зловживати завантаженням статей: якщо ви тільки завантажуєте сотні статей, а не читаєте їх, то це не має сенсу. Краще прочитати докладно одну статтю, ніж завантажити 20 і не прочитати. Часто читання однієї статті займає тиждень або більше (поміж іншими справами), тому що найчастіше у статті містяться результати багаторічної праці, що спирається на десятиріччя попередніх досліджень у галузі.

10.3 Наукові журнали

Наукові журнали сортуються за імпакт-фактором (коефіцієнтом впливовості), тобто що частіше статті журналу цитують в інших статтях, то більший вплив у журналу. Звісно, якісну статтю прочитають й у «маленькому» журналі, а погану не прочитають навіть у «великому», якщо вона туди раптом потрапила. Але журнали намагаються відібрати найкращі манускрипти, тоді як автори намагаються опублікуватися в якнайкращому журналі. Цю систему всі лають, тому що часто авторам доводиться подавати статтю в один журнал — її відкидають, змінювати формат, подавати в інший журнал — там теж відкидають, далі йти по журналах, аж поки статтю не приймуть на рецензію, яка може затриматись (треба чекати на відповідь трьох рецензентів). Зрештою, статтю приймають до публікації або відкидають — і все починається знов. Рецензії можуть бути справедливими та професійними (деякі рецензенти навіть підписують свої рецензії), а можуть бути й відвертим знущанням (анонімно) або рецензент може просто не зрозуміти роботу чи вимагати розмістити його статті в переліку літератури.

З боку рецензента проблема полягає в тому, що йому надсилають багато статей на рецензування з різноманітних журналів, а рецензування — це безкоштовна і досить важка праця: деякі статті цікаві, але їх важко зрозуміти, треба вчитуватись,

щоб написати якісну рецензію, деякі статті низької якості, інколи треба запускати програми, які написали автори, а вони не завжди працюють.

Спостерігаються деякі покращення в системі рецензування: постпублікаційні рецензії, публікація тексту рецензії разом зі статтею, але загалом система рецензування залишається такою самою, як і в епоху паперових журналів.

Головні журнали з біоінформатики та обчислювальної біології (повний список від Google Scholar):

- Bioinformatics: https://academic.oup.com/bioinformatics;
- PLOS Computational Biology: https://journals.plos.org/ploscompbiol/;
- Briefings in Bioinformatics: https://academic.oup.com/bib;
- BMC bioinformatics: https://bmcbioinformatics.biomedcentral.com/.

Журнали з біології та медицини (https://scholar.google.com/citations?view_op=top_venues&hl=en):

- Nature Reviews Genetics;
- Nature;
- Science;
- The Lancet;
- Nature Communications;
- Cell;
- JAMA;
- PNAS;
- Nucleic Acids Research (NAR);
- Journal of Clinical Oncology;
- Scientific Reports;
- Nature Genetics;
- Nature Biotechnology;
- Proceedings of The Roya Society B;
- Moledular Biology and Evolution;
- Genome Biology and Evolution;
- Genome Research.

У кожної конкретної галузі є своя лінійка з 5–10 топ-журналів (у вірусології, молекулярній еволюції, імунології тощо).

Треба мати на думці «свої» журнали, тобто які здебільшого читають у лабораторії та в яких публікуються співробітники, прагнути до кращих публікацій. Також треба уникати predatory journals — журналів, які намагаються відкусити свою частину від бізнесу наукового видавництва шляхом розсилання спам-повідомлень, які містять запрошення надіслати статтю. Ці журнали за гроші публікують що завгодно (див. https://en.wikipedia.org/wiki/Predatory_publishing). Якщо ви публікуєте якісну роботу в подібному журналі, вона буде скомпрометована, її не читатимуть і не цитуватимуть. Часто дебатується питання: чи відносити журнали https://www.mdpi.com/ до категорії predatory publishing.

Вони справді десь посередині між legit і predatory залежно від галузі, але якщо у лабораторії достатньо потенціалу щодо публікацій, краще грати в лізі вищій, ніж MDPI.

10.4 Інструменти пошуку літератури

Google Scholar дозволяє ознайомитися з публікаціями окремого дослідника. Варто мати свій профіль, його не складно створити. Критерій якісної аспірантури — це коли команда «дослідник + науковий керівник + лабораторія» здатна виконати проєкт і опублікувати 1–3 статті у провідних 10–20 журналах за 5 років.

Потенційний науковий керівник уже повинен мати статті в таких журналах, але якщо він / вона їх не має, то зненацька публікувати з новими аспірантами їх не почне.

Одна з метрик наукової продуктивності — це Гірш-індекс. Не треба перебільшувати його значення (займатися гіршометрією), але один погляд на цей індекс у Google Scholar чи Scopus дасть вам приблизну оцінку науковця. Гірш-індекс 100 означає, що дослідник опублікував понад 100 статей, які були процитовані щонайменше 100 разів (тобто 100 разів і більше). Ось, наприклад, один із видатних сучасних дослідників у біології, який працює дуже давно, — George Church — має індекс H194. Інший приклад — дослідниця, яка нещодавно стала PI (Principal Investigator) і явно йде вгору, має якісні публікації — Kamila Naxerova з індексом H24. Для порівняння профіль українського академіка біології, директора інституту біохімії НАН України Сергія Комісаренка з індексом H23 — це гарний результат, але він показує, де перебуває українська наукова спільнота відносно лідерів. А «заслужені» керівники без профілю, які мають H-індекс <10, навряд чи стануть бустером вашої академічної або індустріальної кар'єри.

Scopus має багато функцій і може використовуватись як інструмент наукометрії, але потребує платної підписки та розробляється видавництвом Elsevier, яке останніми роками вважається токсичним через свою ненаситну бізнес-модель. Але у Scopus Author можна швидко і безкоштовно ознайомитися з автором, якого немає в Google Scholar.

PubMed — це безкоштовна та дуже потужна система пошуку публікацій від Національної Бібліотеки з Медицини NCBI USA. В пабмеді можна знайти статтю за назвою, подібні статті й відразу отримати текст (якщо він доступний і потрапив до PubMed Central). Кожна стаття має ID у системі PubMed — PMID та стабільне посилання, наприклад, Cummings 2017.

Connected Papers — унікальний інструмент, який будує граф літератури. Ви починаєте з однієї статті й бачите, з якими статтями вона пов'язана тематично, а також які статті у цій галузі найбільш впливові (найбільш цитовані відображаються більшими колами).

10.5 Книжковий клуб

До Журнального клубу можна додавати Книжковий клуб — читати та обговорювати наукову книгу раз на місяць, раз на два місяці, на семестр чи влітку замість ЖК, проводити огляди нових чи класичних цікавих книжок, «ретро»-статей. Книжки та класичні статті є противагою зневажливому ставленню до науки, яке, на превеликий жаль, є дуже поширеним у сучасній біології, мовляв, «ця стаття дуже стара — вийшла вже 10 років тому, тож немає сенсу її розглядати».

11 Bulk RNA-seq

11.1 Hitchhiker's guide

Аналіз даних bulk RNA-seq (диференційна експресія генів) — це найпоширеніший стандартний аналіз даних у NGS-біоінформатиці: https://en.wikipedia.org/wiki/Gene_expression_profiling. Стаття RNA sequencing data: hitchhiker's guide to expression analysis впроваджує відмінний вступ до цього аналізу та типу даних.

Головна відмінність даних bulk RNA-seq від Single Cell RNA-seq — це здатність «мікроскопу»: у bulk RNA-seq РНК видобувається з багатьох клітин, тобто сигнал експресії усереднюється на рівні зразка, а саме частки тканини, в той час як single cell технологія зберігає інформацію, в якій саме клітині вимірюється експресія.

При секвенуванні bulk RNA-seq бібліотек на інструментах Illumina секвенується cDNA, тобто продукт зворотної транскрипції ізольованого транскриптома.

Окрім диференційної експресії, дані bulk RNA-seq можна використати для інших, складніших типів аналізу:

- збірки транскриптомів *de-novo*, яка актуальна для немодельних видів;
- аналізу мутацій сплайсингу, також і в менделівських захворюваннях;
- аналізу експресії ізоформ генів;
- визначення ДНК-варіантів;
- аналізу A>I редагування;
- аналізу gene fusions;
- аналізу алель-специфічної експресії.

Головні стадії аналізу диференційної експресії:

- планування експерименту — метадані, глибина секвенування, покриття (sequencing depth), репліки (replicates);
- quality control — контроль якості даних;
- отримання counts рівня генів чи транскриптів — матриця експресії (примітка: влучного варіанту перекладу слова counts ми не знайшли, counts — це кількість фрагментів або ридів РНК-секвенування);
- аналіз диференційної експресії — набір up-regulated і down-regulated генів;
- функціональний аналіз груп генів.

11.2 Планування експерименту

Дуже важливо спланувати експеримент заздалегідь, а не намагатися виправити поганий експериментальний дизайн на етапі аналізу. Часто це неможливо: якщо в експерименті недостатньо реплік, то в аналізі нічого зробити не можна. Додаткове секвенування зразків теж може бути проблемним, бо комбінування датасетів призводить до batch effect. На batch effect можна внести поправку, але тільки якщо категорії зразків подані в кожній batch, інакше неможливо відокремити ефект категорії від ефекту групи зразків (batch).

Детальну інформацію щодо планування експерименту дивіться в покрокових інструкціях (tutorials) та статтях нижче. Головне — запросити для обговорення проєкту біоінформатика, який відповідатиме за аналіз ще на етапі планування експерменту (якщо ви біоінформатик, наполягайте на такому обговоренні до того, як почалася робота зі зразками). Треба мати таблицю метаданих, які описують експеримент, мати чітке уявлення того, які фактори впливають на експресію генів у експерименті (важливо перелічити BCI фактори). Експеримент повинен мати достатньо реплік для аналізу всіх факторів (у простішому випадку одного фактора це 3 + 3 біологічних зразки). Треба також відрізняти технічні репліки від біологічних: технічні репліки — повторне секвенування однієї бібліотеки — не додають до статистичної потужності (statistical power) щодо визначення біологічного сигналу, хоча вони можуть бути корисними для валідації роботи лабораторії та збільшення глибини покриття (важливо при роботі з генами, рівень експресії яких малий).

11.3 Контроль якості даних

Здебільшого контроль якості базується на мапуванні ридів (bam-файл), навіть якщо цей файл не використовується для quantification (отримання counts). Мапування ридів RNA-seq з урахуванням сплайсингу здійснюється найчастіше програмами STAR та hisat2 як один з кроків у складі RNA-seq пайплайну. Вихідний файл мапування — bam-файл, що дозволяє підрахувати багато quality metrics, таких як % мапованих ридів, % ридів у екзонах, інтронах, GC-контент та багато інших метрик (див. RNA-SEQC2, RSeQC, RNA-Seq QC Overview).

Якісний RNA-seq пайплайн обов'язково включає модуль QC і QC-звіт.

11.4 Матриця експресії

Існує декілька методів отримати counts. Перше покоління методів використовувало short read alignment (bam-файл) та анотацію генів, щоб порахувати, скільки counts випадає на кожен ген — FeatureCounts. Друге покоління методів базувалось на pseudoalignment, які аналізують kmer frequencies (DNA substrings) — унікальні сигнатури, що характеризують транскрипти — Patcher et al. Нарешті третє покоління використовує transcriptome alignment та kmer-based підхід (https://genomebiology.biomedcentral.com/articles/10.1186/s13059-020-02151-8).

Велика кількість матриць експресії зберігається у GEO — Gene Expression Omnibus. Зараз вважається обов'язковим завантажувати у GEO як сирі дані, так і матриці експресії разом із подаванням статті до публікації. Recount3 пропонує швидко отримати матриці експресії для великої кількості зразків.

11.5 Нормалізація: TPM, RPKM, CPM

Спочатку матриця експресії не є нормалізованою. Однак для того, щоб порівнювати рівні експресії між зразками та генами, потрібна нормалізація.

Є декілька засобів нормалізації:

- https://www.youtube.com/watch?v=TTUrtCY2k-w;
- https://hbctraining.github.io/Training-modules/planning_successful_rnaseq/lessons/sample_level_QC.html.

11.6 Диференційна експресія (DE)

Диференційна експресія — це аналіз, що визначає групу генів, які up-regulated (експресуються сильніше, експресія підвищена) або down-regulated (експресуються слабше, експресія нижча) у групі зразків, об'єднаних у категорію. Наприклад, генів, експресія яких підвищена внаслідок умов експерименту порівняно з контрольною групою зразків. Важливо, щоб різниця в експресії була статистично значущою після multiple testing correction (тобто FDR або adjusted P-value < 0.05 або < 0.01), а метод містив нормалізацію даних (на покриття та довжину транскрипту). Розроблені методи DE-аналізу мають потужний статистичний апарат, щоб виділити біологічний сигнал за даними RNA-seq, в яких багато технічних шумів.

Основні методи:

- Deseq2;
- edgeR;
- limma.

Порівняння: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-14-91

Результатом DE-аналізу зазвичай є excel-таблиця, яка містить статистично значущі гени. Корисно додати до неї стовпчики значень експресії (нормалізовані) для всіх зразків або середні значення за категоріями — це допомагає інтерпретувати результати, дивитись, у яких зразках і генах експресія підвищена чи нижча.

11.7 Функціональний аналіз

Біологічний сигнал DE можна оцінити за кількістю значущих генів так:

- 10–20 генів — слабкий сигнал;
- понад 100 генів — досить сильний сигнал, може дати статистично значущу біологічну інтерпретацію;
- понад 1000 генів — дуже сильний сигнал, трапляється в експериментах біології розвитку (коли типи клітин та профілі експресії змінюються повністю).

Після отримання таблиці DE-генів постає питання, як можна інтерпретувати цей результат функціонально. Пакети, які дозволяють це зробити:

- https://bioconductor.org/packages/release/bioc/html/clusterProfiler.html — імплементує багато типів функціонального аналізу, також і enrichGO;
- http://yulab-smu.top/biomedical-knowledge-mining-book/index.html;
- https://www.gsea-msigdb.org/gsea/index.jsp — GSEA — стандартний метод який має велику базу Gene Sets, один із найбільш цитованих методів у історії біоінформатики;
- https://davetang.github.io/muse/go_enrichment.html;
- https://bioconductor.org/packages/release/bioc/vignettes/simplifyEnrichment/inst/doc/simplifyEnrichment.html;
- https://bioconductor.org/packages/release/bioc/html/topGO.html;
- http://www.webgestalt.org.

11.8 Візуалізації

Класичними візуалізаціями, які підходять до статей, є:

- heatmaps — показують експресію генів для панелі з 25, 50, 100 генів та кластеризують зразки за профілями експресії;
- volcano plots — відображають гени як точки у просторі logFC/-logPvalue, тобто можна побачити, які гени мають статистично значущий сигнал і яка його амплітуда (Fold Change);
- enrichment plots — відображають біологічні категорії enriched у групах up- і down- regulated генів, тобто який біологічний зміст має DE-сигнал.

11.9 Бази даних: GTEx, GEO

GTEx є найбільшим проєктом, який запровадив значення експресії генів для великої кількості тканин post-mortem зразків людини (>1 000). GTEx counts як на рівні генів, так і на рівні екзонів можна використовувати як контрольні дані. Дані GTEx доступні як у сирому вигляді, так і після обробки (counts) (додатково GTEx містить геноми на екзоми для деяких зразків).

Geo — це сховище даних RNA-seq, ATAC-seq, результати аналізу яких опубліковані в статтях. Можна шукати та завантажувати RNA-seq датасети, які стосуються будь-якої галузі біології.

Bgee — порівняна експресія у нормальних тканинах між різними видами.

11.10 Tutorials (покрокові навчальні інструкції)

Ось деякі (існує дуже багато якісних) tutorials, за якими можна вивчити цей тип аналізу:

- http://bioconductor.org/packages/devel/bioc/vignettes/DESeq2/inst/doc/DESeq2.html — аналіз за допомогою DESeq-2;

- https://hbctraining.github.io/Training-modules/planning_successful_rnaseq/ — усе, що треба знати для планування RNA-seq експерименту від HBC;
- https://chagall.med.cornell.edu/RNASEQcourse/ — workshop (коротенький навчальний курс) від Weil Cornell Medical College;
- https://biocorecrg.github.io/RNAseq_course_2019/differential_expression.html — від CRG, Barcelona;
- https://www.youtube.com/results?search_query=differential+expression+analysis — YouTube.

11.11 Статті

- https://www.annualreviews.org/doi/abs/10.1146/annurev-biodatasci-072018-021255 — RNA sequencing data: hitchhiker's guide to expression analysis;
- https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0881-8 — a survey of best practices for RNA-seq data analysis;
- https://genomebiology.biomedcentral.com/articles/10.1186/s13059-014-0550-8 — moderated estimation of fold change and dispersion for RNA-seq data with DESeq2;
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4670015/ — RNA-Seq workflow: gene-level exploratory analysis and differential expression;
- https://pubmed.ncbi.nlm.nih.gov/26925227/ — differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences;
- https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-14-91 — a comparison of methods for differential expression analysis of RNA-seq data;
- https://www.biorxiv.org/content/10.1101/2022.12.14.520412v1 — DE analysis in python;
- https://www.nature.com/articles/nrg3244 — studying and modelling dynamic biological processes using time-series gene expression data;
- https://pubmed.ncbi.nlm.nih.gov/28424332/ — improving genetic diagnosis in Mendelian disease with transcriptome sequencing;
- https://www.nature.com/articles/nature24277 — genetic effects on gene expression across human tissues;
- https://www.science.org/doi/10.1126/science.1261877 — effect of predicted protein-truncating genetic variants on the human transcriptome;
- https://www.nature.com/articles/nrg.2015.3 — RNA mis-splicing in disease;
- https://www.pnas.org/doi/10.1073/pnas.0506580102 — gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles;
- https://www.cell.com/trends/genetics/fulltext/S0168-9525(23)00018-5 — interpreting omics data with pathway enrichment analysis;
- https://www.nature.com/articles/nbt.3519 — near-optimal probabilistic RNA-seq quantification;
- https://www.nature.com/articles/nmeth.4197 — salmon provides fast and

bias-aware quantification of transcript expression;
- https://pubmed.ncbi.nlm.nih.gov/25158696/ — a genome-wide map of hyper-edited RNA reveals numerous new sites;
- https://horvath.genetics.ucla.edu/html/CoexpressionNetwork/Rpackages/WGCNA/ — gene co-expression analysis;
- https://www.nature.com/articles/s41467-020-15816-6 — cell composition;
- https://github.com/zhangyuqing/ComBat-seq — batch effect correction.

12 Single Cell RNA-seq

Останні 5–10 років дуже швидко розвивається галузь Single Cell RNA-seq профілювання транскриптомів на рівні окремих клітин. Це надвелика тема, наведу лише деякі посилання.

Single Cell RNA-seq (SC) аналіз нагадує bulk RNA-seq, де спочатку генерується матриця експресії (counts). У SC ця матриця значно більша, вона містить не тільки зразки та гени, а ще й ідентифікатори клітин, тому аналізи SC потребують великої кількості RAM. Аналізи до 100 тисяч клітин у Seurat можна запускати на потужних лептопах, але більші датасети потребують серверів. Після контролю якості генерують кластери клітин — цей крок відсутній у bulk RNA-seq. Клітини з подібними профілями експресії кластеризуються разом.

Алгоритми такої кластеризації та взагалі проблема dimension reduction — це питання для дебатів і досі (див. дискусію 2022 року навколо UMAP — https://www.biorxiv.org/content/10.1101/2021.08.25.457696v4). Після кластеризації виконують аналіз маркерів, який є різновидом DE-аналізу. Маркери визначають, які гени відрізняють кластери. За маркерами також визначають тип клітин, які складають кластер (cluster identity, cluster annotation). Після анотації кластерів виконують DE-аналіз між категоріями клітин у кластері (експеримент vs контроль). Іноді даних бракує, тоді використовують pseudobulk approach — це підрахунок середнього значення експресії гена у кластері. На основі цих значень робиться DE-аналіз. Ітерації аналізів фільтрація-кластеризація-маркери може проводитися декілька разів, на кожній ітерації видаляються клітини, які не формують кластери (doublets, пошкоджені клітини), або клітини, що не є об'єктом аналізу (наприклад, астроцити у нейронному проєкті).

Single Cell технології розвиваються дуже швидко.

Одним із напрямків є spatial transcriptomics — це коли експресія вимірюється не тільки в окремих клітинах, а ще й фіксується положення цих клітин у просторі. SpatialTranscriptome profiling вже близький до гістології та патології. Якщо поєднати аналіз зображень тканини зі spatial transcriptomics profiling, то можна отримати дані для автоматизації роботи патолога.

Інший великий напрямок розвитку Single Cell технолоії — це multimodal (багатомодальні технології). У них для кожної клітини отримується декілька різних вимірювань, наприклад, експресія РНК і стан хроматину (ATAC-seq). Ці модальності аналізують разом, що дає більш наочну картину.

- https://satijalab.org/seurat/ — основний пакет аналізу в R;
- https://www.10xgenomics.com/products/single-cell-gene-expression — головна молекулярна технологія секвенування SC RNA-seq;
- https://scanpy.readthedocs.io/en/stable/ — SC-аналіз в Python;
- https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/what-is-cell-ranger — Cell Ranger Pipeline;
- https://www.kallistobus.tools/tutorials/scrna-seq_intro/python/scrna-seq_intro/ — bus tools і курс лекцій у CalTech;

- https://hbctraining.github.io/scRNA-seq/lessons/pseudobulk_DESeq2_scrnaseq.html — pseudobulk;
- https://www.nature.com/articles/s41596-021-00534-0 — методи анотації кластерів;
- https://threadreaderapp.com/thread/1577714756047278080.html — якісний список статей для початку;
- https://portals.broadinstitute.org/harmony/ — інтеграція багатьох датасетів;
- https://satijalab.org/seurat/articles/multimodal_vignette.html — multimodal data;
- https://pachterlab.github.io/LP_2021/index.html — Museum of Spatial Transcriptomics;
- https://www.nature.com/articles/s41592-020-01033-y — Method of the year: spatial resolved transcriptomics;
- https://genomemedicine.biomedcentral.com/articles/10.1186/s13073-022-01075-1 — spatial transcriptomics for biomedical research;
- https://www.nature.com/articles/s41592-019-0691-5 — Single-cell multimodal omics: the power of many;
- https://github.com/seandavi/awesome-single-cell — колекція програмних пакетів;
- https://github.com/theislab/scCODA — аналіз диференційної композиції;
- https://satijalab.org/scgd23/ — Single Cell Genomics Day;
- https://www.sc-best-practices.org/preamble.html — найкращі практики;
- https://azimuth.hubmapconsortium.org/ — автоматична анотація;
- https://yoseflab.github.io/Hotspot/ — визначення інформативних груп генів.

13 DNA-seq

Секвенування ДНК може використовуватися для визначення малих варіантів (SNV, indels), CNV (copy-number variants), SV (structural variants), LOH (loss of heterozygosity regions) та інших аналізів мінливості ДНК.

У проєктах із визначенням мінливості на рівні ДНК, особливо у великих популяційних проєктах, працюють фахівці, які мають різний бекграунд: лікарі-дослідники, програмісти, біологи (популяційні, з медичної генетики, молекулярні), лікарі, статистики, аналітики даних, генетичні консультанти, біоінформатики, тому дуже важливо у проєкті визначитися щодо словника термінів, щоб розуміти одне одного. Наприклад, популяційний генетик вживатиме терміни алель, локус, поліморфізм, а лікар вживатиме термін мутація. Зазвичай сходяться на термінології медичної генетики, яку використовують генетичні консультанти (https://www.thriftbooks.com/w/genetics-and-genomics-in-medicine_tom-strachan_judith-goodship/10933430/item/28247557/). Корисно зробити словник у проєкті та проговорити основні терміни: варіант, мутація, алель, SNP, SNV, CNV, DNM, генотип, геном, особливо для того, щоб програмісти зрозуміли, про що йдеться. Цікава стаття про уточнення термінів: https://www.nature.com/articles/gim2016139.

13.1 Типи мінливості ДНК

- Малі варіанти (SNV = Single nucleotide variant, indel < 50 bp);
- структурні варіанти (SV) > 500 bp;
- варіанти у кількості копій (CNV);
- тандемні повтори (STR = short tandem repeats).

Докладніше див. Aaron Quinlan, lecture 5 on genetic variation.

13.2 Технології секвенування

- https://www.nature.com/articles/nrg2626 — Metzker 2010 review — стаття, що відкрила еру NGS (next generation sequencing);
- Sanger: https://www.youtube.com/watch?v=KTstRrDTmWI;
- Illumina: https://www.youtube.com/watch?v=fCd6B5HRaZ8;
- PacBio HIFI: https://www.pacb.com/technology/hifi-sequencing/;
- Oxford Nanopore: https://nanoporetech.com/resource-centre/using-ultra-long-reads-fully-characterise-genomes;
- Центр секвенування: https://www.youtube.com/watch?v=KfyAwAtyUQE.

У розвинених наукових центрах є центри секвенування, які мають усе необхідне обладнання, зокрема секвенатори (це дуже дорогі прилади, яких в одній кімнаті може бути на $ 10 млн). Фахівці центру секвенування найкраще знаються на технологіях, якими володіє центр, тому при плануванні експерименту важливо

обговорити параметри та кошти разом з ними. Якщо немає центру секвенування, то можна надіслати зразки в один із найближчих центрів. Завжди можна запросити розцінки, щоб визначити бюджет.

Найбільші центри секвенування в екосистемі Harvard Medical School:

- https://singlecellcore.hms.harvard.edu/;
- https://genome.med.harvard.edu/;
- http://genomics.broadinstitute.org/;
- https://corefacilities.hms.harvard.edu/sequencing.

13.3 Типи даних, покриття

- WES = whole exome sequencing, WGS = whole genome sequencing: https://pmbio.org/module-03-align/0003/04/01/PostAlignment_Visualization/;
- панелі високого покриття: https://www.youtube.com/watch?v=iC0BLBL8g4g, https://genomemedicine.biomedcentral.com/articles/10.1186/s13073-020-00791-w/tables/1;
- амплікони: https://www.biostars.org/p/112000/;
- секвенування із UMI: https://dnatech.genomecenter.ucdavis.edu/faqs/what-are-umis-and-why-are-they-used-in-high-throughput-sequencing/;
- Duplex UMI: https://www.youtube.com/watch?v=zGhUmGeSKTI;
- Комбінації WES, WGS;
- T/N (пухлина / норма) пари — секвенування для визначення соматичних мутацій;
- трійки, родини — секвенування батьків та дітей, які мають менделівські фенотипи, або для вивчення частот гермінальних (germline) мутацій;
- когорти — секвенування сотень або тисяч зразків пацієнтів на одну хворобу;
- популяції — секвенування тисяч нормальних зразків для визначення мінливості в популяції за етнічним похожденням чи популяції у країні: AllofUS, Dutch Genomes, Ukrainian genomes, UK biobank.

13.4 Референсний геном

Референсний геном — це умовний генотип, який репрезентує цей вид (наприклад, у проєкті «Геном людини» було багато донорів ДНК, але більшість генома — це ДНК від одного з донорів. Докладніше: https://en.wikipedia.org/wiki/Human_Genome_Project#Genome_donors).

Останнім часом науковці підкреслюють важливість мати референсні геноми для різних етносів та популяцій — пангеном людини (https://humanpangenome.org/).

У проєкті дуже важливо зафіксувати, за яким референсом визначатимуться варіанти. Вибір залежить зокрема від уже наявних даних (якщо проєкт має варіанти 10K пацієнтів, визначених за hg19 за останні 10 років, і додається ще 1K зразків, то переходити на новий референс немає сенсу). Новіші,

вдосконалені референси також мають кращі анотації генів і за рахунок високої якості референса — кращі параметри визначення варіантів. Найпоширеніші референси для *Homo sapiens* такі:

- https://www.ncbi.nlm.nih.gov/assembly/GCF_000001405.40 ;
- https://www.ncbi.nlm.nih.gov/assembly/GCF_000001405.13 ;
- https://sites.google.com/ucsc.edu/t2tworkinggroup?pli=1 .

Більшість проєктів у 2022 році виконувалась на основі hg38/GRCh38, але саме у 2022 році з'явився унікальний референс від консорціуму T2T — від теломери до теломери, у якому використовували дані long read sequencing з метою отримання повного референсу. Найімовірніше, що використання цього референсу буде поширюватися: https://www.genome.gov/about-genomics/telomere-to-telomere.

Велика кількість референсних геномів (не тільки людини) зібрана в базах даних UCSC та Ensembl.

13.5 Соматичні та гермінальні (germline) варіанти

Між цими типами є принципова різниця. Гермінальні варіанти визначити легше — за допомогою даних WES, WGS, тому що ми очікуємо на ці варіанти або в гомозиготі (тоді всі риди, скажімо, у 30X WGS підтримуватимуть такий варіант), або в гетерозиготі (тоді приблизно половина ридів підтримає цей варіант). Для визначення варіанта мінімально потрібно 10 ридів (Illumina). Звісно, цю межу зменшують навіть до 5 ридів, але що менше ридів підтримує варіант, то вища ймовірність помилки визначення (false positive). Тобто більшість гермінальних варіантів визначається навіть невеликим покриттям у 30X.

Соматичні варіанти (ракові) визначаються у 5–20 % клітин, тобто при покритті у 30X 10 % VAF варіант матиме в середньому три риди, чого недостатньо для визначення варіанта технологіями NGS. Тому секвенування соматичних варіантів здійснюється значно вищим покриттям, таким як 100X, 150X, або використовуються UMI-технології. Підвищення глибини покриття призводить до зменшення ширини покриття, тому ракові варіанти зазвичай секвенують за допомогою high coverage panels (100–500 генів), а не WGS. Визначення соматичних варіантів також потребує panel of normals (3+ зразків нормальної тканини або варіанти з проєкту 1000 геномів) або matched normals — зразка нормальної тканини (найчастіше крові) того самого пацієнта, в якого взяли зразок на аналіз соматичних варіантів.

13.6 Визначення варіантів, валідація

Для визначення варіантів використовуються best practices пайплайни (див. розділ «Пайплайни»). На практиці, особливо в медичній генетиці, важливо, щоб пайплайн був валідованим (навіть якщо ви його не розробляли — пайплайн має бути валідований у лабораторії власноруч на вашому обладнанні й середовищі).

Валідація — це коли здійснюють секвенування відомого зразка (NA12878, Ashkenazim trio, Chinese trio), а результати визначення варіантів у пайплайні порівнюють з уже визначеним набором варіантів (валідаційні зразки вивчені дуже добре, секвеновані багато разів). Результат валідації — це precision (точність) та sensitivity (чутливість), які можуть розраховуватись за False Discovery Rate (FDR) та False Negative Rate (FNR). Для сучасного WES, WGS секвенування FDR, FNR < 1% для малих варіантів. Зазвичай WGS дає кращі результати для інделів, ніж WES. Валідація соматичних варіантів складніша. Можна використовувати обчислювальний підхід, тобто змішувати риди двох зразків у відомій пропорції, щоб моделювати соматичні мутації, або секвенувати реальні соматичні зразки з відомими мутаціями.

13.7 Анотація

Анотація — це додавання інформації про *naked variant* (який є просто алелем із координатою відносно референса): у якому гені цей варіант, яке його функціональне значення (місенс, нонсенс), яка частота варіанта в популяції, чи трапляється цей варіант у базах даних патогенних варіантів тощо. Є десятки корисних анотацій, які допомагають з'ясувати біологічне та медичне значення варіанта.

VEP — Ensembl Variant Effect Predictor, система анотації варіантів, яку можна використовувати онлайн (для невеликої кількості варіантів) чи встановити локально на сервері, кластері та анотувати великі датасети.

Анотація онлайн: http://grch37.ensembl.org/Tools/VEP, оберіть Assembly GRCh37, спробуйте анотувати варіант: 6 26093141 26093141 G/A +. Простий скрипт для VEP: https://github.com/naumenko-sa/cre/blob/master/cre.vep.sh.

snpEff — інструмент анотації, створений видатним біоінформатиком Пабло Цинголані:

- http://pcingola.github.io/SnpEff/se_running/;
- http://pcingola.github.io/SnpEff/se_inputoutput/#ann-field-vcf-output-files;
- snpEff databases — list of available databases;
- SarsCOV2 is available in snpEff 5.0!: snpEff databases | grep SARS.

Vcfanno (див. у лекції Aaron Quinlan). Ця програма допомагає анотувати один vcf-файл за допомогою інформації з іншого vcf- чи bed-файла.

OpenCravat — це потужна система анотації насамперед соматичних (ракових) мутацій:

- Modules: https://open-cravat.readthedocs.io/en/latest/Home.html#available-modules;
- https://open-cravat.readthedocs.io/en/latest/quickstart.html;
- CLI usage: https://open-cravat.readthedocs.io/en/latest/2.-Command-line-usage.html.

Annovar — ще одна популярна система анотації.

13.8 Пріоритезація

Пріоритезація — це знаходження candidate causal варіантів (варіантів, які ймовірно пов'язані з фенотипом). Основа пріоритезації — це якісна анотація. Далі пріоритезація здійснюється за стандартними правилами ACMG чи правилами для ракових варіантів. Стандартизація правил пріоритезації пов'язана з тим, що результати пріоритезації надходять до молекулярного діагнозу.

Див. Aaron Quinlan Lecture12. Приклад звіту варіантів у родині: https://docs.google.com/spreadsheets/d/1teY0rkE0TF4yvx7ra1nphzq6xGC2OiiS/edit?usp=sharing&ouid=107988514150849896790&rtpof=true&sd=true

Приблизний алгоритм пріоритезації такий:

- фільтр за частотою в популяції (<1 %);
- патогенні мутації у Clinvar;
- знаходження de-novo варіантів (призначити Zygosity до Het,-,-);
- рецесивне успадкування;
- домінантне успадкування;
- настанови ACMG: https://pubmed.ncbi.nlm.nih.gov/25741868/#&gid=article-figures&pid=figure-1-uid-0.

Інструменти пріоритезації:

- R/tidyverse: https://raw.githubusercontent.com/naumenko-sa/tutorials/master/2019-03-22_variant_prioritization/tutorials.prioritization.R;
- gemini/sql DB: https://gemini.readthedocs.io/en/latest/content/querying.html, https://gemini.readthedocs.io/en/latest/content/database_schema.html;
- cBioPortal — cancer mutations: https://www.cbioportal.org/oncoprinter, https://www.cbioportal.org;
- SolveBio: https://www.solvebio.com/, https://docs.solvebio.com/ ;
- Oncoprints with complex heatmaps: https://jokergoo.github.io/ComplexHeatmap-reference/book/oncoprint.html.

13.9 Бази даних

За останні 20 років накопичено велику кількість даних щодо ДНК-варіантів, серед яких:

- https://gnomad.broadinstitute.org/ — варіанти у великій популяції зразків;
- https://www.clinicalgenome.org/data-sharing/clinvar/ — ClinVar та ClinGen — мутації у менделівських захворюваннях;
- https://www.hgmd.cf.ac.uk/ac/index.php — патогенні мутації;
- https://cancer.sanger.ac.uk/cosmic — ракові мутації;
- https://www.ncbi.nlm.nih.gov/snp/ — анотовані малі варіанти;
- https://www.snpedia.com/ — література за кожним варіантом, енциклопедія;
- https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga — атлас мутацій різних типів раку.

13.10 Статті

- Див. також мою лекцію на GenomicsUA: Біоінформатика малих геномних варіантів, slides;
- Koboldt 2020. Best practices for variant calling in clinical sequencing. https://pubmed.ncbi.nlm.nih.gov/33106175/ — рекомендації щодо клінічного секвенування;
- Zook et all. 2016. Extensive sequencing of seven human genomes to characterize benchmark reference materials. https://www.nature.com/articles/sdata201625 — дані для валідації;
- https://www.nature.com/articles/s41587-019-0054-x — принципи валідації;
- https://precision.fda.gov/ — бенчмарки від US Food and Drug administration;
- https://pubmed.ncbi.nlm.nih.gov/25741868/ — інтерпретація варіантів, ACMG guidelines;
- https://pubmed.ncbi.nlm.nih.gov/34859531/ — інтерпретація за допомогою GnomAD;
- https://www.nature.com/immersive/d42859-020-00002-x/index.html — база GnomAD;
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9160216/ — висновки інтерпретації за допомогою GnomAD;
- https://pubmed.ncbi.nlm.nih.gov/29625052/ — гермінальні мутації в ракових зразках;
- https://www.nature.com/articles/s41467-022-33351-4 — порівняння ракових мутацій у популяціях США та Китаю;
- https://www.nature.com/articles/s41586-020-1943-3 — ракові мутаційні сигнатури;
- https://www.nature.com/articles/nrg.2017.52 — принципи пріоритезації варіантів;
- https://www.nature.com/articles/nrg.2017.116 — геномна діагностика;
- https://pubmed.ncbi.nlm.nih.gov/30559314/ — що робити, коли екзомна діагностика не допомагає;
- https://pubmed.ncbi.nlm.nih.gov/28008009/ — когорта 50 тис. екзомів + health records;
- https://pubmed.ncbi.nlm.nih.gov/30609409/ — BabySeq, секвенування новонароджених;
- https://pubmed.ncbi.nlm.nih.gov/27903644/ — секвенування родини з 17 осіб;
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1852721/ — OMIM database;
- https://www.nature.com/articles/nature14962 — UK10K;
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3459641/ — дослідження асоціацій рідкісних варіантів із фенотипом;
- https://www.nature.com/articles/nature22034 — популяція високого рівня спорідненості.

14 Керування часом

Культурна різниця: на Сході часу скільки завгодно, поспішати нема куди — якщо не можна відкласти на місяць, то відкладімо хоч на тиждень. «Помри ти сьогодні, а я — завтра» (В. Шаламов «Колимські оповідання»). Якщо ніхто не питає, як справи, то можна й не працювати. Російська імперія взагалі анахронічна — веде війну за те, чого давно вже нема, засобами минулих війн. «Politics of eternity», як це визначає історик Тімоті Снайдер.

Західний світ: людина існує в часі, у всіх є 24 години на добу, в Північній Америці час — це гроші у прямому сенсі, оплата праці погодинна і може бути від мінімальної 10–15–20 доларів за годину до 50–100 й більше для кваліфікованих фахівців. Робочий тиждень має 35 (Канада) чи 40 (США) робочих годин. На рік — 52 тижні, 1800–2080 годин. Активна кар'єра — 30–40 років, 60–80 тис. годин. Великий проєкт — 1 000 годин. Ось десь 20–50 великих проєктів за життя, якщо пощастить, або розміняєте час на дрібниці.

На Заході, якщо треба щось зробити, починають за можливості одразу, а якщо не можна одразу — планують, коли саме можна почати (але якомога раніше). Очікування: якщо «немає новин», отже, праця просувається, просто ще немає чого показати. У східному, російському світі навпаки — якщо новин немає, то ніхто нічого й не починав робити, треба 10 разів нагадати, тоді, може, почнуть. З боку працівника на Сході: якщо не нагадують, то й робити не треба, робота не вовк — у ліс не втече, не поспішай виконувати наказ, бо надійде інший, протилежний, який все скасує.

Керувати часом треба навчатися — спочатку своїм, потім допомагати організовувати розклад для підлеглих, колег у лабораторії, компанії. Є дуже багато ідей, методик, книг щодо таймменеджменту. Головна ідея — знати точно, на що витрачається ваш час у масштабі дня, тижня, місяця, року. Але надмірна турбота про час теж його з'їдає — така його природа.

Добре мати щоденник, лабораторний журнал проєкту, у якому вказувати, скільки часу пішло на виконання тієї чи тієї роботи. Якщо проєкти типові, то згодом з'являються оцінки, скільки часу займає той чи той аналіз. Добре мати календар (гугл-календар) — особистий та спільний для лабораторії, підрозділу. Треба охороняти свій робочий час від переривань, тому що кожне відволікання, кожна нарада чи імейл ділять ваш день навпіл.

Наприклад, замість продуктивного дня:

- працювали 5 годин поспіль — завершили та надіслали звіт проєкту.

30-хвилинна нарада посеред робочого дня розділяє його на 2 год. + 2 год. 30 хв., тобто:

- 120 хв. праці;
- нарада 30 хв.;
- 150 хв. праці.

Додайте 10 відволікань на імейл по 10 хв.:

- 10 хв. праці;
- імейл;
- 20 хв. праці;
- імейл;
- 20 хв. праці;
- імейл;
- 10 хв. праці;
- імейл;
- 10 хв. праці;
- нарада 30 хв.;
- 20 хв. праці;
- імейл;
- 20 хв. праці;
- імейл;
- 20 хв. праці;
- імейл;
- 20 хв. праці;
- імейл;
- 20 хв. праці.

Додайте сюди декілька проєктів одночасно, слак, месенджери у телефоні, новини, перекуси, колег, які до вас підійшли та посмикали заради «just a quick question» — і ваш день зруйновано, результату не досягнено, звіт не надіслано.

Homo sapiens, на відміну від комп'ютера з тактовою частотою в гігагерцах, істота інертна, кожен її робочий відрізок потребує часу на зосередження, як літак потребує часу на зліт і набір висоти. Кожне переривання — це збитий літак думки. Популярні open space, де люди ходять туди-сюди, снідають, п'ють каву, ведуть бесіди, руйнують усім роботу. Треба встановлювати «library rules», тишу.

Пріоритезація: мати цілі, великі проєкти, розбивати великі проєкти на етапи, робити внесок у великі проєкти та просувати їх щодня (нарізати «слона» по шматках і «з'їдати»). Є справи важливі та неважливі, термінові й нетермінові (уявіть собі квадрат: важливі-термінові, важливі-нетермінові, неважливі-термінові, неважливі-нетермінові). Ваше завдання — приділяти увагу важливим нетерміновим справам (великим проєктам) і не давати повсякденним (особливо неважливим) з'їдати ваші дні. Водночас не давати маленьким неприємним, але обов'язковим справам накопичуватись. Можна назвати їх жабами, записати у список та «з'їдати» по жабі щоранку, аж доки їх не поменшає.

Інколи просто допомагає стежити за часом у Harvest чи іншому інструменті тайменеджменту. Ці інструменти швидко допомагають позбутися прокрастинації (за рік гарантовано), але довго сидіти під стрілкою дуже стресово, на наступному етапі це може не приносити жодної користі, а тільки фрустрацію. (Керівники, якщо ви будете зловживати time tracking, знайте, що працівники шукатимуть засобів, як правдоподібно фальсифікувати трекінг).

Стежити за ритмами: як спливає рік (quarterly objectives, yearly objectives), як сформований тиждень, яка структура дня. Стежити за мотивацією: якщо вам цікаво і необхідно те, що ви робите, і цю справу ви чесно вважаєте важливою, то зазвичай ви це робите дуже ефективно та швидко.

Дивіться більше про організацію роботи у книжках Rework, Remote, Calm.

15 Психологічне здоров'я (mental health, wellness)

Сучасне західне суспільство, насамперед наукове, зняло стигму з психічного захворювання. Від нього ніхто не застрахований на будь-якому етапі життя, творчі люди з розвинутим мозком, які працюють під великим стресом, дуже часто опиняються в межових психічних станах чи просто в патологічній зоні. Основи психології (емоції, депресія, тривожність, вигоряння та ін.) треба знати і стежити за власним психічним станом, станом колег і підлеглих. Коли виникають проблеми — шукати допомоги. У західному суспільстві є дуже багато механізмів допомоги: від простої обізнаності — it is ok to be not ok — до розгалуженої системи психологічних консультантів, літератури, груп допомоги та ін.

Яскравий приклад людини, яка зробила велику кар'єру, маючи таку серйозну особливість, як біполярний розлад (маніакально-депресивний психоз) і яку бачили всі — це принцеса Лея зі Star Wars, яку зіграла акторка Carry Fisher.

Важлива також здорова психологічна атмосфера в лабораторії, підрозділі. Великі корпорації теж за цим стежать, а інколи у них виходить навіть краще, ніж в академії: легше запровадити однакові правила та тренінги у всіх підрозділах компанії на 10 тис. працівників, ніж зробити це у 1 000 наукових лабораторіях, кожна з яких контролюється однією, часто екстраординарною людиною.

Психологічна атмосфера в наукових лабораторіях буває дуже різною: від дуже комфортної, де всі люди позитивні та підтримують одне одного, до дискомфортної — де інтриги, конкуренція, зневага, амбіції, стрес, трудоголізм, навіть наркотики. Психологічна атмосфера не завжди корелює з результатами та науковим рівнем: психологічні катастрофи можуть бути в найкращих у своїй галузі лабораторіях, а «комфортні» лабораторії можуть пасти задніх. Але психологічну атмосферу треба враховувати, бо навіть найкраща у світі наука не буде в радість при постійному стресі з ризиком депресій та інших патологій.

Більшість сучасних керівників та науковців, які їх оточують, створює комфортні позитивні середовища, які водночас є продуктивними. Навіть якщо вам кілька разів траплялися патологічні винятки, це не означає, що треба припиняти пошук.

Оцінити психологічну атмосферу під час співбесід легко: зверніть увагу на реакції людей під час вашого job talk (запишіть його та перегляньте, якщо зосередженість на презентації заважає вам оцінювати аудиторію), ставте запитання під час бесід сам на сам (що є найскладнішим у вашій роботі, який life / work balance?). Більшість співбесід передбачає зустріч з усіма членами лабораторії чи хоча б ключовими співробітниками. Якщо керівник не допускає вас до підписання

контракту спілкуватись із лабораторією — це red flag. Як людина більш досвідчена керівник може вами зманіпулювати, але співбесіда з лабораторією завжди покаже справжній стан речей.

Найчастіша психологічна девіація, яка трапляється в наукових лабораторіях, — це взаємини за моделлю «діти–батьки». Звісно, студенти, аспіранти, постдоки проходять фази «відкладеного зростання», тобто інтенсивно зростають в особистому та професійному плані, хоча давно є дорослими людьми. Керівники задають умови для такого зростання, мають набагато більше досвіду і знань, тобто ієрархічні взаємини закладаються автоматично. Лабораторія — це також дуже мала спільнота, що відкриває як простір для зростання, так і для зловживань.

Лабораторія не повинна нагадувати (і замінювати) родину, а ставлення керівників до підлеглих — ставлення батьків до дітей, і навпаки, «піднесене» ставлення підлеглих до керівників. Психологічно здорова атмосфера в лабораторії — це коли професійна спільнота має контрактні зобов'язання, і відносини сфокусовані на праці та проєктах, а не на співзалежності й почуттях; людей цінують, але не тримають насильно чи маніпулятивно.

Інша девіація, притаманна українському науковому суспільству, — це «кріпацтво». Попри заявлений перехід на європейскі рейки в освіті, студенти потрапляють до факультету й кафедри на 5 років і не дуже часто коригують свій кар'єрний шлях. Навіть на модерних кафедрах із вибором курсів часто читають те, що можуть, а студенти не можуть обирати курси. Аспіранти — такі самі кріпаки наукових керівників, а науковці та викладачі — кріпаки на кафедрах та факультетах. Вихід із культури «кріпацтва» — тільки через прозорість і справжню конкуренцію на всіх рівнях.

Багато написано про психологічне та сексуальне насильство в наукових лабораторіях (abuse of power). Треба знати, що воно не є нормою, і щонайменше в американських університетах з насильством борються дуже жорстко. Якщо ви стали жертвою насильства, треба звертатися по допомогу, а не терпіти мовчки роками.

Україна ще не має розгалуженої системи практичної психології, хоча її елементи з'являються. В академічному середовищі трапляються «генії не при тямі», а насправді просто нещасні люди, які потребують кваліфікованої допомоги, яка б дозволила їм функціонувати на продуктивнішому рівні. Також трапляються токсичні керівники, від яких залежить кар'єра молодших співробітників.

Дуже багато науковців приходить до того, що основа психологічного здоров'я для працівників, які здебільшого сидять за комп'ютером чи лабораторним столом, — це:

- фізична активність: біг (від прогулянок до марафонів та тріатлону), піший туризм (hiking), велопрогулянки (biking), плавання;
- позитивні емоції;
- здорове харчування;
- здорові психологічні бар'єри;

- розумний ритм праці (контроль напруженості, вміння казати «ні», сон, відпочинок, час для себе, родини, друзів).

Якщо цих простих речей виявляється недостатньо для підтримання стабільного психологічного стану, то регулярні заняття з психологом або психотерапевтом можуть допомогти.

Див. огляд у Nature, Mental health — Wiki, Mental health in Academia (paywall).

16 Робота над текстами. Препринт. Проти культу записок

Робота над текстами дуже важлива як для кар'єри в академії, так і в індустрії. У західному (англомовному, франкомовному, німецькомовному, італомовному тощо) світі культура нарисів (essays), уміння висловити та сформулювати свою думку в тексті має глибоке коріння (ще від давньогрецької та римської культур), її навчаються зі школи. А в нас — диктанти, перекази, записки, реферати, автореферати, де головне — форма, а не зміст.

Треба усвідомити, що диктатура форми — це насамперед вплив російської імперської культури, російської літератури, де навіть літературознавці визнають: словесні пишноти, об'єм, форма — скільки завгодно, попри те, що зі змістом, сюжетом — проблеми. Тож треба прагнути змісту, щоб насамперед виправити наявний дисбаланс.

Ключові елементи тексту: зміст, достовірність, форма.

Зміст. По-перше, треба щось напрацювати, щоб було що записати. «Половину дисертації з океанології становила "риба", а решта дисертації була заповнена "водою"». Комп'ютери зробили кожного письменником. Написано так багато текстів, що кожен повинен замислитися: а чи справді мій текст потрібен? Можеш не писати — не пиши, пиши тільки тоді, коли не можна не писати. Якщо нема чого писати, то почитай спочатку, що інші написали з цієї теми, але зроби щось, щоб це було варте опису.

Достовірність (authenticity). Треба чесно писати так, як можеш, від себе. Не треба красти чужих текстів і видавати їх за свої. У системі, де багато хто некоректно запозичує (перекладає сторінки вікіпедії, російської чи англійської, собі в текст, передирає з науково-популярних ресурсів, книжок, дипломних робіт, дисертацій попередніх років), усе ж таки можна піти іншим шляхом. Існує цілий науковий світ, де принципово не брешуть та не передирають текстів, і приєднатись до нього якомога раніше буде корисно.

Форма (шрифт, відступи, нумерація сторінок, позначення ілюстрацій, таблиці в науково-технічних текстах, «толстовщина» в описах) важлива, але не може бути самоціллю. Не можна ставити форму поперед змісту, особливо в наукових текстах.

Щоб проілюструвати ці досить абстрактні думки, розгляну дві магістерські дипломні роботи з біоінформатики. Дуже перепрошую їхніх авторів та керівників. Немає сумніву, що вони працювали чесно, просто ці роботи хоча б знаходяться в гуглі та потрапили до мене, а тисячі інших десь заховалися. Авторефератів дисертацій з біоінформатики українською мовою я просто не знайшов.

Харків–2021:
https://openarchive.nure.ua/bitstream/document/16951/1/2021_M_ShI_Baranov_YeO.pdf

- Форма «пояснювальної записки» збереглася точнісінько така сама, яка була, мабуть, в 70-х роках XX ст.

- «Значимість роботи оцінюється як висока, що обумовлено поточним станом в світовій системі охорони здоров'я і поточною пандемією короновірусної хвороби.» — ніби автор сподівається врятувати світ.
- Стор. 14, порівняйте із https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BA%D0%B2%D0%B5%D0%BD%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%A0%D0%9D%D0%9A: «Технологическая платформа для быстрого широкомасштабного секвенирования была создана в 2005 году фирмами 454 Life Sciences[6] и Illumina (ранее Solexa)[7], и сначала использовалась для секвенирования геномов[en]. Первые работы по секвенированию транскриптомов появились в 2008 году. В числе первых были секвенированы транскриптом дрожжей[8], арабидопсиса[9] и мыши[10]. В настоящее время РНК-секвенирование осуществляется в основном с использованием трех инструментальных платформ широкомасштабного секвенирования: Illumina, 454 Life Sciences и SOLiD[11]. В 2019 году удалось секвенировать РНК из кожи, хрящей, печени и скелетных мышц щенка Тумата (волка или собаки) возрастом 14300 лет[12].» — цей текст автор запозичив з російської вікіпедії, просто переклавши на українську, тобто це є некоректне запозичення тексту. Далі виникає запитання: чи всі «гладкі» частини огляду літератури та гарні ілюстрації з scRNA-seq було некоректно запозичено? Якщо виникає така підозра, далі вже повинна працювати автоматична програма, яка шукатиме запозичення замість розгляду роботи за суттю.
- Швидкий пошук за зображенням 2.7 дав дуже багато посилань в інтернеті, зокрема https://bolster.ai/blog/gans-in-real-world-can-bad-actors-use-gans-to-beat-ai. Тобто підозра, що більшість тексту та ілюстрацій некоректно запозичено, підкріплюється.
- «Огляд літератури» йде аж до стор. 47 (всього 65), далі починається щось схоже на саму роботу, стор. 48–59. Що саме було зроблено, які саме датасети взяті («датасет з назвами груп клітин, які мали під собою біологічне обґрунтування»), де програмний код (десь у додатку) — не дуже зрозуміло. Автор наводить 5 гістограм одного кольору з одним покажчиком «accuracy». Звісно, кожен експеримент проводився один раз, тобто жодного статистичного значення результати не мають. Висновок: усі мережі працюють по-різному на різних задачах, тобто ясно, що нічого не ясно.
- Ось де автор був щирим: «Як можна побачити з рисунку 3.1, необроблені вхідні дані в форматі FASTQ займають 23 гігабайти дискового простору, що може ускладнювати обробку цього датасету. На щастя, результуючі матриці після обробки займають менший об'єм даних та можуть бути оброблені в цілому». Тобто автор ніколи не мав нагоди проаналізувати реальний scRNA-seq dataset, починаючи з сирих даних. Чи він проаналізував матричний датасет хоча б у Seurat?
- І швидко-швидко автор переходить до висновків, стор. 60: «В цілому,

використання вищезазначеного процесу має зменшити кількість ручної роботи, яку треба зробити для аналізу матриці експресії генів та частково автоматизувати процес аналізу даних, що дозволить швидше отримувати корисні результати та можливо пришвидшити розробку ліків від таких недугів, як рак, ВІЛ/СНІД та коронавірусна хвороба COVID-19». Wishful thinking, detached from reality.

- Попри досить яскраву та актуальну тему (застосування нейромереж для аналізу даних scRNA-seq), ця робота є черговою імітацією. Приклад справжньої роботи з подібними мотивами — застосування deep learning для визначення геномних варіантів у Google привело до дуже успішного проєкту deepvariant, який став поруч із GATK HaplotypeCaller та Dragen — світовими лідерами у визначенні варіантів (variant calling).

Київ–2020: https://ela.kpi.ua/bitstream/123456789/40067/1/Yevdoshchenko_bakalavr.pdf

- «Немає запозичень», але перекладено абзац за абзацом з російських ресурсів постнауки, біомолекули та навіть лєнта.ру. Здебільшого це робота з перекладу, а не біоінформатики;
- переклад у роботі налічує аж 53 сторінки із загальних 82, тобто це гігантська праця саме з перекладу;
- сама робота стосовно моделювання сигналів викладена на стор. 54–77, здебільшого в малюнках;
- незрозуміло, як саме промодельовані сигнали, у якому середовищі;
- яким чином робота пов'язана саме з аналізом даних Nanopore sequencing?

Система боротьби з плагіатом уже розроблена, гроші освоєні, тож проблема розв'язана? https://uk.nure.info/novyny/145-borotba-z-plahiatom.html

Обидві роботи були виконані дуже старанно. Викладачі намагалися знайти цікаві теми, студенти витратили дуже багато часу. Чому ж виходить імітація?

Такі самі записки вимагають від викладачів на наступних ступенях наукової «кар'єри»: дисертації та автореферати, монографії, навчальні посібники та навчальні програми. Ось приклад такої роботи: http://iht.univ.kiev.ua/wp-content/uploads/2021/10/02-%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%BD%D0%B0-%D1%82%D0%B0-%D1%84%D1%83%D0%BD%D0%BA%D1%86%D1%96%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0-%D0%B3%D0%B5%D0%BD%D0%BE%D0%BC%D1%96%D0%BA%D0%B0.pdf.

Знов за формою не видно змісту.

Для порівняння: абсолютно вільна за формою, але багата за змістом програма курсу прикладної обчислювальної геноміки (Applied Computational Genomics — прикладна обчислювальна геноміка) від одного з лідерів сучасної біоінформатики — Aaron Quinlan: GitHub.

Звісно, щось змінити майже неможливо, хоча багато хто розуміє всю хибність культу записок. Коли я навчався, вимагали «хоч 50 сторінок». Зараз бачимо 70, мабуть, за 30 років буде вже 100 за допомогою ChatGPT.

Але все ж таки що можна зробити, коли хтось захоче:

- поставити текст на своє місце: головне — сама робота, її суть, яку текст супроводжує. Нема роботи — нема про що писати.
- Текст теж є роботою (записка надає йому якогось псевдостатусу, мовляв, ми тут просто граємось у текст, насправді це не текст). Текст можна називати препринтом — в ідеалі статтею, готовою до публікації. Дивіться препринти у відкритому архіві: https://www.biorxiv.org/, https://www.medrxiv.org/. Сервери препринтів існують для того, щоб розповсюджувати текст статті ще до публікації в журналі. Проходження рецензії ще в декількох журналах, аж поки статтю приймуть, може займати декілька років, тому добре вже «напівофіційно» опублікувати текст, щоб надсилати колегам, які змогли б його прочитати. Деякі автори взагалі не подають статей до журналів, щоб не марнувати часу, а залишають роботу у вигляді препринта (якісно зробленого).
- Текст треба писати так, щоб його можна було опублікувати в науковому журналі чи подати на конференцію (який саме журнал чи конференцію, залежить від проєкту. Ідеально, якщо це реальна західна англомовна конференція та реальний, хоч і не вищого рівня, науковий журнал).
- Треба порушувати реальні теми, а не вдавати, що рятуємо людство. Наприклад, тема курсової для програміста: пофіксити баг у Seurat (головний пакет аналізу Single Cell даних) так, щоб фікс узяли в реліз. Тобто опанувати аналіз у Seurat, стежити за GitHub, знайти баг, який треба, щоб був зафіксений, та пофіксити. Необхідно буде декілька спроб, щоб це вдалося. Код там не такий вже й легкий — R, можливо R/Cpp. Це буде справжній внесок, але це не буде щось величне, на кшталт «нейромереж у SingleCell для спасіння людства», і записка буде лише на одну сторінку з посиланням на issue та commit у GitHub. Але чи це приймуть на кафедрі? Почуєте щось таке: «Наші студенти пишуть по 70 сторінок, а тут одна. Як це можна захищати? Це не робота!».
- Проблема полягає також у кількості студентів. Велика кафедра може випускати й 100 бакалаврів + 100 магістрів на рік. Неможливо взяти 200 «справжніх» тем на рік, та ще й кожен рік нових, свіжих і актуальних. Але якщо не гратися в «науку», а подивитися, на що реально спроможна кафедра та студенти, то можна знайти вихід. Наприклад, кафедра програмування може підтримувати два реальних open source програмних продукти. Хай усі теми будуть закриттям багів чи про нові функції, а препринти — на 5–10 сторінок, із посиланням на issue та код у GitHub.
- Реальні наукові проєкти потребують багато зусиль, одна стаття може містити десятки аналізів, кожен аналіз може бути темою диплому, а разом вони — становити статтю. За пів року (типовий термін виконання дипломної роботи)

майже неможливо зробити роботу на високому рівні. Тож можна робити ітерації аналізів, покращуючи результати наступним поколінням студентів.

- Якщо з'явиться раз на 5 років студент/-ка, який / яка напише справжню велику роботу за новим методом з гарним текстом, то й добре.
- Інше нескінченне джерело відносно простих тем — вікіпедія. Погляньте на вашу галузь науки, технології, і ви побачите, які бідні та неповні українські сторінки. Робота може бути перекладом і доповненням однієї великої сторінки. «Записка» — одна сторінка з посиланням на результат праці. Це буде також робота з перекладу, але її результат буде на користь спільноті, а не піде на смітник.

17 Усе тече, все змінюється: reproducibility

17.1 Постійні зміни

Програмні інструменти, бібліотеки, мови програмування, фреймворки, пайплайни змінюються дуже швидко. Ви вже не можете один раз опанувати мову ANSI C 90 та провести наступні 40–50 років, користуючись лише цим одним програмним інструментом для розв'язання задач. Здається, розробники цілком упевнені, що їхня екосистема є найважливішою у світі, а користувачі будуть щасливі присвятити своє життя тому, щоб стежити за всіма змінами.

Звичайно, розробники сфокусовані на своїй системі, намагаються покращити саме її, звідси постійні зміни. Але мало хто бере до уваги, що користувач може одночасно стежити за 10–20 різними системами і кожну з них використовувати потроху. Існує ще й такий аспект, що кожен колектив розробників намагається створити свій програмний продукт. Ідея полягає в тому, щоб користувачі проводили все своє життя за цим продуктом, тобто треба розробити продукт якомога детальніше. Це суперечить базовій філософії Unix-систем «одна утиліта — одна функція», але така філософія погано монетизується. Деякі екосистеми нагадують тоталітарні секти. Особливо це стосується систем, розроблених великими корпораціями. В одному тільки AWS (Amazon Web Services) є понад 10 сертифікацій, тобто ви можете провести 10–15 років або навіть усю кар'єру в біоінформатичних системах обробки даних у хмарі від Amazon.

Гнучкі методології розробки додають жвавості, але й занадто збільшують кількість маленьких змін, релізів. Існують команди розробників, які завжди щось фіксять у продакшні, й це для них норма стресу.

Прикладів змін дуже багато, легше сказати, які програми не змінюються.

Приклади стабільних програм:

- GSEA: https://www.gsea-msigdb.org/gsea/index.jsp, десктопна програма на Java для функціонального аналізу наборів генів, існує вже років 20 (з 2003 року). Це так довго, що молоді спеціалісти вважають її інтерфейс доісторичним (він справді нагадує десктопні програми епохи LRT-моніторів);
- bwa: https://github.com/lh3/bwa, найпоширеніша програма на C з мінімумом залежностей, для мапування коротких ридів, написана Heng Li, підтримується з 2011 року, тобто вже понад 10 років.

Приклади дуже динамічних систем:

- сама мова Python: перехід від Python 2 на Python 3, постійні вдосконалення та покращення. Наприклад, оператор := у Python 3.8, який нарешті дозволяє зробити цикл while not eof замість нескінченного циклу while True (https://peps.python.org/pep-0572/#sysconfig-py);
- мова R більш стабільна, майже не змінюється, але для аналізу даних переважно використовують зручнішу екосистему бібліотек tidyverse

(https://www.tidyverse.org/), яка змінюється дуже швидко, і навіть деякі функції виводяться з обігу (з'явилось таке поняття, як життєвий цикл функції — користувачі в захваті, тепер треба не тільки вичати самі функції, їх використання, які в них параметри, але й у якій фазі життєвого циклу перебуває функція, коли починати її використовувати, коли закінчувати та переходити на нові функції, переписувати код);

- екосистема GATK/Mutect/Picard: https://gatk.broadinstitute.org/hc/en-us. Вона дуже потужна, містить сотні утиліт, десятки workflows, цілу хмарну платформу, що побудована на базі Google Cloud — Broad Terra — https://terra.bio/. Але щоб професійно нею володіти, треба сидіти на форумах, спілкуватися з програмістами, стежити за релізами.

Багато програмних інструментів з біоінформатики розробляється, навіть публікується, підтримується декілька років, а потім аспірант захищається, постдок знаходить посаду, після чого програму ніхто вже не підтримує. Таких програм безліч. Існують цілі лабораторії, де завжди розробляються якісь нові програми, а потім їх кидають, і за 10–15 років жодна з них уже не підтримується. Це трапляється не тільки через злу волю та wishful thinking, а найчастіше внаслідок кар'єрних траєкторій та недостатнього фінансування. В академії важко дістати гроші на підтримку програми, у грантах треба завжди обіцяти щось нове, а не підтримку 10-річної програми. В індустрії корпорація має гроші, але не бачить сенсу публікувати програму open source та нести відповідальність за неї. У співробітників вистачає власних проєктів усередині корпорації.

Ситуація потрохи покращується, з'являється більше стабільних програм, є спеціальні фонди, які дають гроші на підтримку наукових open source програм, наприклад, https://chanzuckerberg.com/eoss/, https://summerofcode.withgoogle.com/.

Тут є велика ніша для лабораторій з біоінформатики в Україні. Можна взяти якийсь якісний пакет, який усім потрібен, але який нема кому підтримувати, і підтримувати його для світової спільноти. Таких кинутих (дуже гарних) пакетів — сотні.

Знову ж таки, внаслідок постійних змін у програмах професія біоінформатика потребує психологічної стійкості. Якщо ви підтримуєте пайплайни в лабораторії, запускаєте старі аналізи, то навіть удосконалювати програми й бази даних до сучасних версій — це вже велика праця, до якої додається version control та reproducibility й валідація. Аж тут виявляється, що пайплайн з оновленими програмами дає інші результати, ніж трирічної давнини з тими самими даними.

Reproducible-аналіз — це такий аналіз, який інша людина в іншій лабораторії може запустити й на тих самих даних отримати такий самий результат. Здається, просто, але більшість опублікованих аналізів не є reproducible, що називається reproducibility crisis. Причин дуже багато: або використали новий метод, який самі ж не опублікували, або не опублікували скрипти аналізу, або не зафіксували версії, або все опублікували та зафіксували, але аналіз такий складний, що в іншій лабораторії треба провести декілька місяців, щоб налаштувати потрібне програмне оточення.

Зараз reproducibility — це важлива тема. Лідери наукової спільноти наполягають на публікації програмного коду і даних разом із текстом статті, а краще — повних пайплайнів, контейнерів, програмних оточень (environments).

17.2 Інструменти reproducibility

- GitHub — створення репозиторіїв програмного коду дозволяє керувати кодом та впроваджувати version control;
- docker та singularity контейнери пропонують завантажити програму, скрипт і всі залежності (бібліотеки) в невеликий контейнер, який потім можна запускати. Багато програм зараз розповсюджується у вигляді контейнерів;
- RMarkdown та Jupyter ноутбуки — інтерактивні аналізи, які можна запускати та змінювати;
- використання стандартних інструментів аналізу, пайплайнів (не треба писати свій пакет, якщо вже є стабільний пакет для цього аналізу);
- якісна документація аналізу без оминання жодного кроку;
- нумерація скриптів в аналізі: 01.get_data.sh, 02.quality_control.Rmd, 03.cluster_analysis.Rmd тощо;
- створення environments: conda/bioconda, python venv, tutorial — тобто створюють середовище з усіма необхідними пакетами для аналізу, його зберігають, і за рік або п'ять років можна запустити той самий аналіз за тими самими версіями програмних пакетів;
- https://davetang.org/muse/2019/12/04/reproducible-bioinformatics/.

17.3 FAIR data

Не тільки аналізи, скрипти та пайплайни повинні бути reproducible та reusable, а й дані, сгенеровані в експерименті чи проєкті, повинні відповідати подібним стандартам. Якщо скрипти, які використані для аналізів у статті, доступні, а дані — ні, то наукова спільнота не може скористатися результатами проєкту (і гроші платників податків витрачені марно).

Принципи FAIR data:

- F = findable — дані повинні бути розташовані так, щоб їх можна було знайти (наприклад, у репозиторіях GEO, SRA);
- A = accessible — можна отримати доступ до даних. Навіть секвенування зразків людей можна розмістити в репозиторії dbGAP, доступ до якого є контрольованим;
- I = Interoperable — дані є в такому форматі, а метадані створені так, що їх можна використовувати в інших аналізах у комбінації з іншими даними;
- R = Reusable — дані можна використовувати багато разів, не тільки в одному аналізі.

18 Networking: наукове спілкування

Networking — це діяльність кожного науковця для створення мережі контактів. Саме наукова спільнота, ком'юніті, а не конкретний учений, є носієм знань та методів і здатна розв'язувати великі та складні задачі, робити якісні публікації, впроваджувати експертні оцінки. Науковці у своїй галузі знають одне одного — це сотні контактів. Це велика частина роботи — підтримувати контакти на конференціях, семінарах, через гранти, колаборації. Багате наукове середовище сприяє науковому спілкуванню (networking): коли постійно організовуються семінари, доступні тревел-гранти, науковці багато подорожують. Якщо такого середовища немає, усе ж таки можна займатись networking: приєднуватись до зум-семінарів, писати колегам просто щоб познайомитись, запросити відомого науковця зробити доповідь в інституті.

Гірше за все — замикатися на роки в лабораторії з науковим керівником та кількома співробітниками і не виходити у «великий світ». «Замикання» в лабораторії — це шлях до багатьох зловживань — від науки, що не пов'язана ні з чим реальним, до токсичних взаємин у лабораторії. Інколи networking дається важко вченим-інтровертам. Треба знати свої психологічні властивості та використовувати спеціально створені ситуації для наукового спілкування: конференції, семінари. Завдання аспірантури та постдоку — не тільки виконати проєкти та написати статті, а й набрати необхідну кількість контактів.

Job interviews є важливою частиною культури networking. Коли ви приходите або приїздите на співбесіду, ви робите job talk — більш формальний виступ-доповідь, а також chalk talk — неформальну сесію запитань і відповідей, зустрічаєтеся з науковцями лабораторії чи департаменту поодинці чи з декількома особами за один раз, снідаєте разом. Навіть якщо ви не отримуєте позицію, ці професійні контакти будуть корисними.

Окрім контактів у вузькій спеціалізованій галузі, важливо підтримувати контакти у своїй науковій спільноті — в дослідницькому центрі, у шпиталі, в університеті. Для цього організовують семінари — для всього кампусу, школи, окремо для постдоків, аспірантів, research associates, PIs, керівників лабораторій. Ніколи не знаєш, який контакт спрацює в новому проєкті чи на новому ступені кар'єри, тому науковці серйозно ставляться до всіх контактів — це частина роботи.

19 Видатні науковці

Багато чого можна навчитися, якщо стежити за лідерами в галузі — які статті вони публікують. За допомогою https://scholar.google.com дуже легко розібратися, «хто є хто» у вашій галузі.

Найбільш цитовані автори:

- всі за Гірш-індексом >=100;
- науковиці;
- біоінформатики;
- фахівці з геноміки;
- фахівці з транскриптоміки;
- фахівці з single cell аналізу;
- популяційні генетики.

Деякі науковці, близькі до біоінформатики:

- Gad Getz — ракова геноміка;
- Daniel MacArthur — варіанти та експресія у великих когортах (GnomAD, GTEx);
- Richard Durbin — біоінформатика;
- Aaron Quinlan — інструменти для аналізу варіантів у рідкісних (менделівських) хворобах;
- Xiaole Shirley Liu — CHIP-seq аналіз, ракова біоінформатика.

20 Bioinformatics Core

Bioinformatics Core (BC) — це спеціалізований підрозділ в екосистемі великого дослідницького центру, який викладає тренінги з біоінформатики для лікарів-дослідників чи для лабораторних біологів та виконує біоінформатичні аналізи у співпраці з дослідниками. Наявність BC та інших допоміжних facilities (Sequencing Core, Single Cell Core, Spatial Transcriptomics Core, Metabolomics Core) — це ознака багатої дослідницької культури центру.

Приклади:

- https://bioinformatics.sph.harvard.edu/ — BC у системі Harvard Medical School, Boston, US;
- https://lsi.ubc.ca/resources/facilities/bioinformatics/ — BC at University of British Columbia, Vancouver, Canada;
- https://www.well.ox.ac.uk/research/scientific-cores/bioinformatics-statistical-genetics — Bioinformatics&Statistics Genetics at Wellcome Centre For Human Genetics, Oxford, UK;
- https://abc.med.cornell.edu/ — Applied Bioinformatics Core at Weill Cornell Medicine, NY.

20.1 Навчальна компонента

Зазвичай BC має одного чи навіть цілу команду викладачів-тренерів, які розробляють короткі практичні навчальні курси з біоінформатики. Ця діяльність нагадує будь-який навчальний процес: розробка та підтримка курсів, запис на курси, реклама курсів у науковій спільноті, проведення курсів, зворотний зв'язок, години «відчинених дверей». Залежно від попиту курси можуть бути кожного місяця чи навіть тижня, кожного кварталу, семестру або щороку.

20.2 Життєвий цикл проєкту

Типовий біоінформатичний аналіз у BC проходить такі стадії:

- контакт — запрошення від дослідника до BC;
- перша відповідь, стандартне уточнення інформації щодо проєкту (експеримент, тип даних, тип аналізу, обсяг проєкту, наскільки терміново);
- перша консультація: більш детальне обговорення проєкту, терміни, складність, призначається аналітик;
- фінансові подробиці;
- передавання даних та метаданих до BC;
- аналіз «сирих» даних за допомогою пайплайнів;
- quality control, повідомлення досліднику про якість даних, подальші перспективи аналізу (garbage in — garbage out, або чи гарні дані);
- декілька ітерацій аналізу даних (аналіз — звіт — обговорення);

- підготовка публікації;
- документація проєкту;
- передавання даних від ВС до дослідника;
- виставлення рахунку (щомісяця або один раз);
- видалення даних.

20.3 Програмні системи

ВС може використовувати такі системи для організації діяльності:

- Trello — картки для життєвого циклу проєкту: кожен проєкт — це картка, до якої заноситься стандартна інформація (назва проєкту, бюджет, коротенький опис, контактна інформація дослідника, аналітик). Картки організовані у стовпчики згідно зі стадією проєкту і пересуваються праворуч, від початкової стадії до завершення;
- Harvest — для таймменеджменту;
- Basecamp — для лабораторних журналів, щоб витягнути інформацію з імейлів; публікація звітів, ілюстрацій;
- Dropbox — для передавання та збереження невеликих файлів (звіти, презентації, документи, таблиці);
- Globus — для передавання великих файлів (fastq, bam).

20.4 Критерії якісного проєкту

Проєкти надходять до ВС у великій кількості. За своєю якістю вони можуть бути дуже різними: від тих, які приносять цікаві біологічні результати та публікації в топових журналах, до тих, які приносять тільки розпач та вигорання аналітикам. Залежно від фінансової моделі ВС може мати нагоду не брати всі проєкти, які до неї надходять.

Існує декілька критеріїв, за якими можна оцінити якість проєкту: люди, наукова проблема, дані, методи. Дуже зручно організовувати проєкти в таблиці та досліджувати цей датасет. Можна швидко оцінити кожен із компонентів як 0 чи 1. Якісний проєкт має всі чотири складники, тобто оцінку 4 в підсумку. Якщо один складник провисає (0), то проєкт може бути проблемним. Якщо 2–4 складники проблемні, проєкт приречений із самого початку.

Таблиця, яку можна заповнювати для проєктів:

- назва проєкту;
- хто є PI (Principal Investigator), який у нього / неї Гірш-індекс, чи публікував цей дослідник якісні статті раніше, чи публікував конкретно з цієї тематики, чи взагалі бере участь у цьому проєкті (в обговореннях) або бере участь тільки researcher/postdoc/phd student. Якщо PI не бере участі, це одразу red flag — проєкт може не мати жодного реального значення або бути дуже ризикованим (PI не зацікавлений/-на);

- хто є основним дослідником, який Гірш-індекс. Якщо в дослідника немає публікацій до проєкту, то найімовірніше, вони й не з'являться у процесі цього проєкту;
- що саме хочуть дослідники від ВС (наприклад, вони можуть розповідати про якусь цікаву наукову задачу, а насправді просто бажають, щоб ви їм запустили програму, яку вони не можуть запустити, а тільки-но ви надішлете результати, ваша участь у цікавій науці закінчиться);
- яке ставиться наукове питання, чи воно взагалі має сенс? Інколи дослідники, особливо медичні доктори, настільки заклопотані, що бувають випадки, коли науковий складник не проходить навіть базової перевірки (sanity check). Звісно, як біоінформатик ви не завжди можете оцінити науковий складник проєкту, який до вас приносить спеціаліст у вузькій галузі, але можна трішки переглянути літературу, подивитися та скласти враження;
- які дані є у проєкту? Чи аналіз цих даних взагалі дозволяє дати відповідь на наукове запитання? Яка якість даних? Пам'ятайте принцип «garbage in — garbage out». Значна частка проєктів закінчується на етапі ретельного аналізу якості даних: виявляється, що дані поганої якості або експеримент узагалі не спрацював. Наприклад, 3 роки тому були переплутані клітинні лінії, і зараз це видно на PCA-діаграмі, побудованій на основі даних RNA-seq. Або взагалі дослідники думали, що вони експериментували з культурою клітин людини, а виявилося, що в них була лінія миші, яку вони позичили в колег. (Обидва випадки були реальні в моєму досвіді). Завданням біоінформатика тут є довести інформацію щодо проблем до PI якомога раніше, а не сподіватись «виправити» результати за рахунок аналізу. Якщо є якісні дані, це підвищує шанси на успіх, але його не гарантує. Витрачати час на аналіз поганих даних — це неетично насамперед щодо дослідника (який наприкінці сплачуватиме за цей аналіз);
- який саме метод чи методи біоінформатики будуть докладати до аналізу даних? Найкраще, якщо метод уже існує, він валідований у багатьох дослідженнях, а результати аналізу опубліковані багато разів (DESeq, GATK). Буває, що метод опублікований, але тільки один раз, і працював на одному датасеті. Це може стати проблемою: чи взагалі метод спрацює на іншому датасеті? Буває так, що метод опублікували, а підтримувати його для інших користувачів немає кому (прикладів безліч). Може бути так, що метод добре працює, але датасет трохи інший, і застосувати метод неможливо. Тут можна написати авторам у Github, на email. Деякі автори методів дуже допомагають, деякі — не допомагають, тоді ви можете залишитися без методу й потрібно буде щось винаходити. Можливо, є інші методи, які можна знайти в літературі, можливо, адаптувати якийсь метод. Але це вже фактор ризику для проєкту: чи є бюджет на розробку методу, чи взагалі ви зможете розробити метод, чи вистачить у вас кваліфікації. Якщо фактично методу немає, треба повідомити PI якомога раніше та обговорити альтернативи. PI може думати, що у вас все готово, вам тільки «натиснути

кнопку», а насправді для цієї задачі навіть розробленого методу не існує. Навіть для добре розроблених та опрацьованих методів часто буває так, що запустити програму (декілька програм) і отримати результати може бути дуже складно. Інколи це потребує знаходження та фіксення багів у програмі (з допомогою автора чи самотужки) або написання додаткових програм для конвертації даних чи побудови пайплайну. Знання методів, нових та якісних пакетів — ключове в роботі біоінформатика. Треба тримати руку на пульсі щодо головних методів, стежити, чим займаються лідери в галузі (bulk RNA-seq, CRISPR, SC RNA-seq), читати статті, які порівнюють методи (benchmarks).

Проєкт із усіма якісними чотирма компонентами теж може бути неуспішним — слабкий біологічний сигнал, невдалий експеримент. Протягом років можна побачити, які експерименти й датасети випускають дослідники в екосистемі, і вже з самого початку буває зрозуміло, який потенціал у проєкту.

21 Production informatics у великій компанії

У цьому розділі розглянемо підрозділ біоінформатики у великій компанії (такій, як Big Pharma) та визначимо відмінності від академічної лабораторії біоінформатики.

Компанія зосереджена на розробці ліків та їх клінічних досліженнях у своєму портфоліо. Бізнес-світ дуже спеціалізований: є великі фармкомпанії, які займаються виробництвом ліків та розробляють нові ліки, вони можуть купувати та інтегрувати менші компанії; є компанії, які організовують клінічні дослідження (фармкомпанія може їх замовити); є компанії, які збирають когорти клінічних зразків.

Припустимо, що в компанії є підрозділ секвенування, який на замовлення біоінформатиків здійснює секвенування великої кількості біологічних зразків, і щотижня дані секвенування потрапляють до відділу PI (Production Informatics).

Приклади систем, які можна використовувати у Production Informatics:

- JIRA — на кожен датасет відкривається ticket, у якому він описується за стандартною формою, датасет обробляється, і ticket надає можливість стежити, хто відповідальний, де збережені дані, які проблеми виникли, як їх було розв'язано;
- GitHub — є команда розробників, система репозиторіїв, можливо, з аутсорс-учасниками, які підтримують систему пайплайнів та orchestration (створення віртуальних машин, конфігурації, завантаження, логіювання);
- SolveBio — варіанти завантажують у систему, в якій можна писати запити та аналізувати когорти;
- cBioPortal — ще одна система зберігання варіантів;
- LIMS — база даних зразків;
- MSTeams — спілкування працівників, організація зустрічей;
- HPC — команда системних адміністраторів підтримує локальний обчислювальний кластер для експериментів, валідації та розрахунків;
- Dragen — підрозділ може мати декілька серверів з платами Dragen;
- AWS або GCP — команда, яка відповідає за збереження даних у хмарі та хмарні бюджети;
- безпека даних — спеціалісти, які стежать за безпекою даних.

Відмінність від академічного підрозділу:

- значно більше людей (у рази);
- функції більш спеціалізовані (програміст є програмістом, а не програмістом + біоінформатиком + системним адміністратором + викладачем + менеджером, як буває в академії);
- вертикальна інтеграція (всі співробітники є співробітниками компанії, у них єдиний інтерфейс — пошта, календар, месенджери);
- організація в команди з чітким менеджментом;
- більш стандартизована праця;

- великі обсяги даних;
- великі можливості щодо використання комерційних систем, продуктів.

22 Біоінформатика в Україні

Професійна спільнота GenomicsUA збирає словник та посилання на ресурси з біоінформатики українською тут: https://github.com/naumenko-sa/awesome-ukrainian-omics-words. Додавайте інформацію, будь ласка.

22.1 Критика програми КНУ імені Тараса Шевченка

Прошу зауважити, що моя критика — не заради критики. Зробити нову програму — це дуже складно, та й ресурси будь-якої кафедри обмежені. Тому першопрохідці заслуговують на всіляку повагу та заохочення.

Водночас для студента буде корисно знати, що саме в нього питатимуть з біоінформатики при влаштуванні в аспірантуру або міжнародну компанію, тобто які білі плями є в його освіті. Також я сподіваюся, що моя критика буде сприйнята як порада і привід для подальшого покращення освітніх програм з біоінформатики.

Розгляну якісну (мабуть, найкращу натепер в Україні) програму КНУ — https://iht.knu.ua/navchannja/navchalni-disciplini/.

По-перше, я не бачу (може, вони є, але я не знайшов) відео лекцій та матеріалів практичних занять, тобто, щоб оцінити програму, треба дивитись нормативні документи. Майже всі біоінформатики публікують матеріали відкрито: https://github.com/hbctraining, https://github.com/hbc/knowledgebase/blob/master/scrnaseq/tutorials.md.

По-друге, незрозуміло, яка структура, послідовність курсів у програмі. Є NGS-технології, програмування, бази даних, філогенетика, фізика твердого тіла, наноплазмоніка, нанофізика, навіть щось із теорії поля, машинне навчання, навіть ПЛІС — логічні схеми, C/C++, хімія. Тобто тут тем на цілий університет. Але чи дає програма знання студенту, щоб йому стати біоінформатиком та пройти співбесіду? Тобто чи дає ця магістерська програма добру спеціалізацію або ж це черговий сферичний кінь у вакуумі? Зрозуміло, що кафедри завжди обмежені у викладанні й викладають «що можуть» замість того, «що треба» для програми світового рівня, але все ж таки повинен бути баланс між «можемо» і «треба».

Я провів багато співбесід із біоінформатики в США та Канаді (і як претендент, і як представник лабораторії чи компанії). Ось найменший перелік запитань, які ставлять кандидату, що претендує обійняти практичну посаду NGS-біоінформатика:

- 1) Linux/bash;
- 2) базові знання молекулярної біології (ДНК / РНК), загальне розуміння та цікавість до біології, медицини;
- 3) Python — практичні навички, GitHub. Базові алгоритми, структури даних. «Співбесіда програміста»;

- 4) R/Rstudio/tidyverse/dataframe та статистика.

Зазвичай запитують щодо практичного досвіду аналізу даних.
З огляду на ці вимоги програма КНУ не має:

- 1) статистики — трішки торкаються, R — теж 2–4 години;
- 2) Linux/bash — тільки для самостійної роботи;
- 3) практичних аналізів NGS — не побачив.

Тобто головна проблема програми зі структурної біології та біоінформатики КНУ полягає у відсутності структури програми. Якщо переглянути всі курси, то C/C++ там більше, ніж R/Python, а фізики більше, ніж статистики.

По-третє, відокремлю ось такі курси (знову прошу вибачення, я тільки можу дивитися на формальну записку — робочу програму. Може, там насправді все набагато краще викладають, а записку пишуть, тому що мусять). Також я змінив порядок курсів, щоб згрупувати їх у блоки, з'ясувати структуру програми.

22.1.1 Біонформатика

- Технології аналізу даних: теорія аналізу даних. Ентропія. Трішки статистики (але і нейрони, і навчання, і сусіди, і регресія — все це за 4 години). Є Python — але теж лише 4 години. Хоча, може, це магістри і вони вже всі знають Python?
- Структурна та функціональна геноміка — все дуже загально, є NGS, є бази даних. Але що конкретно? Чи навчають там хоча б простої функціональної анотації (які гени, транскрипти, pathways, як цю інформацію здобувати). Список літератури трохи дивний.
- Програмування в біоінформатиці: Python, scipy, numpy, R, Rstat, C++. Якісний курс, можливо, центральний у програмі. Але дивно: як можна все це вивчити в одному курсі, тобто на якому рівні воно виходить? Чи здобувають студенти справді практичні знання чи це просто стрибання по верхівках? Список літератури теж дивний — є набагато кращі ресурси з кожної теми.
- Інженерія програмного забезпечення: Agile, .NET, MS Visual Studio, IDEF, UML, WEB, PHP, Java, JSON, python, CMS, G, LabView, Jira, git. Здається, це жарт — усі ці теми вмістили у 20 годин лекцій + 20 годин лабораторних.
- Комп'ютерна практика: UNIX + Python — самостійна праця.
- Науково-виробнича практика — незрозуміло, який її зміст.
- Молекулярна філогенія: bioedit, blast, clustal, mafft, MEGA. Досить базовий курс філогенетики за російськомовним (хоча й гарним — автор працює в Нідерландах) підручником Лукашова. Тільки desktop-програми, а зараз філогенетика робиться здебільшого на серверах за допомогою Raxml, Phyml, PAUP, Paml. Курс може бути значно

посилений до рівня https://www.cambridge.org/core/books/phylogenetic-handbook/A9D63A454E76A5EBCCF1119B3C56D766, Felsenstein: https://felsenst.github.io/.

- NGS DNA: секвенування, маркери, праймери, аж раптом — OMIM! Лекція про секвенатори Roche 454 and ABI Solid, які вже не використовуються. NCBI, EMBL-EBI. Книжки Рєбрікова, які не слід використовувати. Дещо дивний курс. Не бачу, щоб хоча б розповіли різницю між WGS / WES / Panels / Amplicon sequencing / WGBS, RNA-seq.
- Програмування мовами C. C++ за п'ять лекцій (!). Сучасний C, здається, не зачепили. Найбільш важливе — які там завдання, з програми це незрозуміло.
- Проєктування баз даних: SQL, noSQL, MongoDB. Маленький курс SQL. Немає згадки про «Understanding SQL» Мартіна Ґрубера. Звісно, це «Трішки про SQL», а не «Проєктування баз даних».
- Вебпрограмування: PHP, javascript, css, mysql. На жаль, вибір PHP не можна вважати вдалим, ця технологія використовується, але здебільшого для legacy-проєктів, і вона безпосередьо не пов'язана з біоінформатикою. Використання Python у цьому курсі підсилило б узагалі складник програмування. Інша ідея — R/Shiny — розробка веб-застосунків за допомогою R.
- Машинне навчання: scikit-learn, regression, neural nets, pyTorch, NLP, RDkit. Здається, якісний, але дуже коротенький. Незрозуміло щодо практики.
- Тенденції сучасної біоінформатики: доповіді, письмовий іспит. Незрозуміло, що конкретно.

22.1.2 Protein science, drug design

- Загальна та прикладна біоінформатика: білки, докінг, динаміка.
- Обчислювальна хімія.
- Біохімія — здається, якісний, але не дуже великий курс (20 лекцій).
- Структурна біологія: біохімія, ДНК, взаємодія білок-ДНК. Читання статей.
- Визначення структури біологічних макромолекул: спектроскопія, ЯМР, PDB.
- Молекулярна динаміка.
- Computational drug discovery — єдиний курс англійською назвою. Здається, якісний курс. Незрозуміло щодо практики.
- Біоінформатика білків: фолдинг. Незрозуміло щодо практики.

22.1.3 Гуманітарний блок

- Організація наукових досліджень — трохи філософії науки, як писати резюме.
- Професійна етика на 90 годин: добре, що є гуманітарний курс із достатнім часом на написання есеїв.
- Логіка.

- Психологія: цікавий курс насправді — за Е. Берном, малі групи, психологія спілкування.
- Англ. мова — щось є, але дуже мало. Тільки writing можна викладати 10–20–30 занять. Див. про роль англ. мови у біоінформатиці на початку цього посібника.
- Історія культури.

22.1.4 Математична фізика

- Математичні методи в сучасній біології: математичні курси за російським підручником Різниченка (МДУ). Тобто диф. рівняння, чисельні методи, коливання. До біоінформатики в широкому значенні слова це не має стосунку, це ближче до математичних моделей біофізики.
- Моделювання складних систем — знов просувається синергетика, трішки статистики, система реакція-дифузія, порядок-хаос, вейвлети, клітинні автомати.
- Фізика твердого тіла. Як вона стосується біоінформатики? Може, цей курс пов'язаний із молекулярною динамікою, ЯМР?
- Наноплазмоніка: наноматеріали, наномедицина, «самоорганізація наночастинок». Знов трішки теорії поля з фізики. Тут я не фахівець, але до біоінформатики у визначенні цього посібника цей курс недотичний.
- Наноматеріали — щось дивне на російських джерелах. Матеріалознавство?
- Взаємодії в наносистемах: знову фізика твердого тіла? Наночастинки? Знову функція Гріна? Тобто у спеціалізації з біоінфоматики та структурної біології є величезний блок hardcore математичної фізики з 5–6 курсів. Звісно, фізичне мислення ще нікому не завадило, але чи не забирають ці курси час від саме основ біоінформатики?

22.1.5 Мікроелектроніка

- ПЛІС: Xilinx / Verilog. Узагалі цей курс дуже якісний, але до чого він тут? Цей курс для інженерів-системотехніків-електроніків, які розробляють плати чи БПЛА. Як це стосується біоінформатики?

Узагалі ідея гарна: бакалавр біології спеціалізується з біоінформатики, вивчає більше програмування. Сподіваюся, студенти не проходять усіх цих курсів, а обирають, що треба. Помітна дуже сильна хімічна школа — найкращі курси тут protein science, тобто це більш хемоінформатика, ніж біоінформатика, можливо, Drug Design specialty. Помітна кількість просто незрозумілих курсів. IT-курси несистематичні, не пов'язані один з одним. Великий провал у статистиці. Якщо її якісно не виклали бакалаврам, то біда. Подивився курс, який вони викладають бакалаврам, — не дуже якісний. Незрозуміло, чи є журнальні клуби. На одному курсі бачив статті для розгляду. Незрозуміло, наскільки гарна англійська програма. В ідеалі десь половина навчання має бути англійською, щоб набрати словник та практику.

Програма КНУ — це найкраща програма з тих, що є в Україні. Тобто якщо хтось хоче зробити конкурентну програму, то це цілком можливо. Не слід лягати у прокрустове ліжко магістерської програми, можна вирощувати її еволюційно: розробити дуже якісний курс аналізу даних bulk RNA-seq, запустити, потім — дуже якісний курс SC RNA-seq, рухатись поступово.

22.2 Курс «Біоінформатика»: Івано-Франківськ, Львів, Харків

Привертає увагу нещодавно організований онлайн-курс біоінформатики: http://lifesciencescourse.org/. Це ще тільки один курс, а не програма, але це справді приклад сучасно організованого курсу з біоінформатики: він цілеспрямований (єдина концепція), інтегрований (різні теми викладають різні викладачі, але теми органічно пов'язані), містить як фундаментальні поняття біоінформатики, так і сучасні інструменти, дуже практичний. Здається, на сучасному етапі жодна кафедра чи факультет в Україні не мають достатніх ресурсів для викладання повноцінної програми з біоінформатики, тож залучення найкращих спеціалістів з різних університетів до спільної програми цілком виправдане.

22.3 Як розвивати навчальну програму з біоінформатики: ідеї

Висловлювати загальні поради дуже легко, на відміну від їх втілення, але все ж таки:

1) адаптувати найкращі курси аналізу біологічних даних, які добре розроблені в англомовній біоінформатиці та доступні безкоштовно (bulk RNA-seq, single cell RNA-seq). Стежити за тим, щоб у групі був високий рівень розуміння базових аналізів — постійно їх обговорювати;
2) організувати щотижневий журнальний клуб (50 зустрічей на рік);
3) стежити за балансом 4-х опорних складників: біологія, R/статистика, Python/програмування, Linux;
4) підтягувати статистику за допомогою StatQuest і підручника;
5) організувати обчислювальний сервер Linux (хоча б за допомогою одного потужного комп'ютера) та вчитися на ньому працювати (conda environments, installations, resource sharing, job submission, monitoring);
6) мати реальний науковий проєкт, який веде до публікації. Якщо такого немає — шукати спільний проєкт у лабораторіях, де є дані та задачі, але немає біоінформатиків. У крайньому разі — взяти опублікований аналіз даних у топ-журналі, завантажити сирі дані з GEO та повторити аналіз власноруч. Може, будуть цікаві винаходи;
7) підтягувати культуру програмування та знання алгоритмів і структур даних через розв'язання простих ACM-задач (+Python, git-практика).

23 Післямова

Посібник не є ідеальним, він не претендує на повноту охоплення теми, адже був написаний дуже швидко. Також текст перенасичений англізмами через відсутність понятійного апарату в українській мові. Дякую всім, хто долучився до його видання!

Зауваження, доповнення та пропозиції щодо посібника можна надсилати за допомогою GitHub https://github.com/naumenko-sa/bioinf_posibnyk_public або на імейл: serhiy крапка naumenko @ yahoo крапка com.

Питання з біоінформатики українською можна порушувати у професійній спільноті «Геноміка ЮА»: https://genomics.org.ua і в чаті телеграм-групи: https://t.me/GenomicsUA.

Багато додаткових посилань можна знайти в базі знань.