This is a shared document to brainstorm project **ideas** for CoFest 2023 (https://www.open-bio.org/events/bosc-2023/obf-bosc-collaborationfest-2023/).

Everyone is welcome to contribute ideas, and we'll use these to coordinate and organize ourselves at the start of CoFest.

Feel free to also discuss ideas on our **Slack** space (https://join.slack.com/t/obf-bosc/shared_invite/zt-n5ur1gsj-z2C~69_4IYTFPg5tbWA8Ew).

The **Zoom** link for this meeting is https://us02web.zoom.us/j/87854497584?pwd=WW1iTGN2ZG9mM0lRZDNDVWlKUEU2UT09 (full dial-in details on the slack).

Please, **sign up** on the following spreadsheet: https://docs.google.com/spreadsheets/d/1BxvvMHieousC9Gl-15UIK5Igoo2pT4I-JHcA3l4xLhc/edit#gid=0

Smaller project ideas for new attendees

Open source LLMs and their applications

Participants: Tazro (@inutano)
This group aims to explore the applications and promotion of open source large language models (LLMs) in the field of bioinformatics. LLMs have gained popularity, and the community seeks models that offer accessibility and transparency. Strategies for collaboration and community engagement, including the use of shared repositories and benchmarking frameworks, will be discussed. Ethical considerations such as data privacy, bias, and interpretability will also be explored.

Sharing resources and ideas: https://docs.google.com/document/d/1fxA3JCtPkScm7vXSFQqPJtgP6OEZgAwN7-o_pKtoH5c/edit?usp=sharing

An Open Source platform to manage (training) events

Participants: Lisanna (@Lisanna Paladin)
Managing events is often not our job, but we end up doing it. When faced with the task, we often rely on a stack of paid or suboptimal tools (e.g. EventBrite, manual emailing, Google Sheets with notes on personal data on someone's account, SurveyMonkey, etc.) because it's not our job and we don't have time to develop the full platform. But what if we did it collaboratively, once

for all, and share it as an open community project? I bet several Open Research communities will benefit from it.
P.S. Lisanna will only join later in the first day and might have other commitments, looking for a co-lead.

Updating the “Blast book” for Blast+

Participants: Jérémy Just (RDP/ENS Lyon/CNRS)
The Blast book was published in 2003. It provides a lot of “recipes” for Blast searches in different contexts. Since then, however, the syntax of most tools from the Blast suite has changed, with the introduction of Blast+ in 2009. The wrapper currently distributed with Blast+ to convert the old syntax (`blastall`) has quite limited features. My idea is to update the command examples in the book to the new syntax, explain new options, and identify places that need more in-depth updating.
I’ve contacted the main author (Ian Korf), and he’s not against a new book about Blast, but he doesn’t want to be involved. I have two hard copies of the book, and I've also scanned it (please do not redistribute the files).

Correspondence table :
https://docs.google.com/document/d/1Vb65V_jgNc_IpLa08VcBzIzcvVM2pC3Kn2LyUAjnKIE/edit

General project ideas

Biopython

Participants: Peter Cock (@p.j.a.cock on slack)
Peter will be attending remotely, but as one of the regular Biopython contributors will try to match any newcomers with suitable projects/issues. He hopes to work on migrating the Tutorial documentation from LaTeX to RST to simplify and automate it for each release (see PR 4371). See the #cofest-biopython channel on the slack.

MultiK main function parallelization

Participants: Quentin Cavallini-Speisser (RDP/ENS Lyon/CNRS)
MultiK is a R package built upon Seurat that objectively selects multiple insightful numbers of clusters (K) in a single-cell RNA-seq dataset.
However the main function of the package is very expensive both in computing and in time since it is not parallelized.
The idea would be to attempt to parallelize it using the future package (which is already used by Seurat) in order to speed up scRNA-Seq workflows making use of it.

Show Exons and Isoforms in iCn3D

Participants: Jiyao Wang, NCBI/NLM/NIH, iCn3D developer and Ravinder Abrol (CSUN)

Abstract:
iCn3D is a web-based 3D structure viewer synchronizing 1D, 2D, and 3D view, e.g., https://www.ncbi.nlm.nih.gov/Structure/icn3d/?mmdbid=1TUP&showanno=1&show2d=1. The 1D sequence view shows all kind of annotations (e.g., domains, SNP, etc) in tracks. This project will show the start and end positions of exons in the sequence, and show the sequence of isoforms of the protein and their exons. Thus users can clearly see the exon skipping and potentially relate the exon skipping to the protein functions.

Create a bioconda recipe for Bionano Solve software

Participants: Clément Bellot (RDP/ENS Lyon/INRAE)

Bionano is a technology to create optical maps from HMW DNA. One major problem is the lack of tools options for analysis since there are only tools provided by Bionano Genomics : Bionano Access (server and GUI) and Bionano Solve (analysis software).
The installation of Bionano Solve is clumsy : it uses a docker image to install dependencies. My idea is to retrieve all the dependencies used from the docker image to create a bioconda recipe in order to easily install and maintain Bionano Solve software dependencies.

Making biomedical research software reusable

Participants: Bhavesh Pate (FAIR Data Innovations Hub, California Medical Innovations Institute)

Abstract:

Most would agree that making biomedical research software (code, scripts, desktop software, Jupyter Notebooks, etc.) reusable is essential to prevent duplicate effort, enable building on top of existing work, and ultimately increase the pace of discoveries and innovations for improving human health. The question then is, how do we make biomedical research software reusable? The Findable, Accessible, Interoperable, and Reusable principles for Research Software (or FAIR4RS principles) published in 2022 provide high-level instructions to achieve that. It is the result of a large-scale effort and is backed by a large community of research software developers. However, just like the original FAIR principles, the FAIR4RS principles remain aspirational and do not provide clear actionable instructions. To address this, we have established the FAIR Biomedical Research Software (FAIR-BioRS) guidelines, that provide clear, actionable step-by-step instructions for making biomedical research software reusable in line with the FAIR4RS principles. Our idea here is to discuss the current version (v2.0.0) of the FAIR-BioRS guidelines, identify if/how they can be improved, brainstorm on how they can be maintained going forward, etc. so as a community we can start adhering consistently with the FAIR4RS principles to make our software reusable and also provide clear guidelines to do so especially for the next generation of biomedical software developers.

Common Workflow Language 1.2.1 release work

M. Crusoe will be writing the changelog for the planned 1.2.1 point release of CWL.
Work in progress is at https://github.com/common-workflow-language/cwl-v1.2/pull/262
Previews

- https://deploy-preview-262--cwl-v1-2-dev.netlify.app/commandlinetool#Introduction_to_the_CWL_Command_Line_Tool_draft_standard_v1.2.1
- https://deploy-preview-262--cwl-v1-2-dev.netlify.app/workflow#Changelog_for_v1.2.1

Adding native RO-Crate export to the CWL reference runner (cwltool)

M. Crusoe & R. de Wit

The Common Workflow Language reference runner (cwltool) has built in support for emitting CWLProv RO Bundles via the --provenance command line option. This packaging format is being superseded by the Workflow Run RO-Crate profile, which is heavily based upon CWLProv but is more generic. There is already a CWLProv RO Bundle to RO-Crate converter (written by Simone Leo); a direct export will offer more customization and provenance pass through options. Ideally the work will be transferred to a reusable Python library for any CWL runner that wishes to use it.

Making it easier to update bio.tools records on software release & jalview 2.11.3.0 bio.tools record

Hervé Menager and Jim Procter

Original objective - small achievable goal of updating bio.tools records for a new software release with changes to command line interface. Challenge is to how to prepare for making the update without actually updating the record. One approach - use the JSON representation available from the Update record site and store it in your project's repository - eventually this could be posted automatically (after validation) to have records automatically updated on release.

Progress made: Jalview git commit https://source.jalview.org/gitweb/?p=jalview.git;a=commit;h=c2bb08e5dc01cc9de193f19b625c3f1abd0488c5 and issue on Jalview bugtracker