Lessons from the NESCent-sponsored series of software development hackathons

Authors: blah blah blah

Affiliations: blah blah blah

Corresponding author: blah blah blah

**Table of Contents**

[Abstract](#h.nbl254ez4fbw)

[Introduction](#h.rpe0lsy8adeq)

[Methods](#h.z2b8tp3208d)

[Data collection and encoding](#h.dboxuvdxpjod)

[Format and availability](#h.zchlkkmrlis2)

[The NESCent model for extended, carefully scoped hackathons](#h.7xobi1ead8cd)

[Overall Structure and Roles](#h.24vqn0wllrdp)

[Pre-planning](#h.pmq14c3xoow2)

[Resources](#h.g9huin7r9bii)

[Assembling the Leadership Team](#h.h2mah3hresev)

[Setting Scope and Goals](#h.50sqnibzbn9d)

[Planning](#h.blsg4g6wwobi)

[Logistical Considerations](#h.la7amf6eo00u)

[Facilities and equipment](#h.o66vz3gimdyo)

[Communication Technology](#h.4eegpu7rrab4)

[Travel](#h.rdigpjd2xuou)

[Identifying Participants](#h.qtnmu2brcbae)

[The Application Process](#h.c1t1rpkkfuon)

[Criteria for Choosing Participants](#h.dwa9372g4ff1)

[Process for Choosing Participants](#h.8q0crtwgextl)

[Pre-hackathon Engagement](#h.yvqbuonp13fr)

[The Hackathon](#h.enow34e8qncg)

[Day 1](#h.12n19h4rutzw)

[Welcome and introductions](#h.w76sp3ax1zol)

[Technical presentations](#h.6awgcv82vfkf)

[Open discussion](#h.w6d827sfcm56)

[Initial pitches and feedback](#h.7u1np5lnldvz)

[Team formation and pitch refinement](#h.z2xhjta5kq3t)

[Intermediate Days](#h.9e8x4yag2jqe)

[Last Day](#h.naoxa3rbcc6a)

[Post-hackathon Wrap-Up](#h.u41ew6xs57rv)

[Results and Discussion](#h.ydkllzty593c)

[We begin with a discussion of our analyses to attempt to measure/quantify impact, and then use this data to explore several specific questions we have about effectiveness of certain features of hackathons.](#h.hn9tvthsrjso)

[Understanding and assessing hackathon impacts](#h.ls4dfth0o4wk)

[Measuring hackathon impact](#h.x70b5o9rcxbz)

[Lessons learned](#h.gs72advo11ah)

[Challenges of building a leadership team](#h.ozneng93bmlu)

[Make people feel welcome; encourage people](#h.ijaxiwp7a136)

[If you have a great hackathon, don't do it again](#h.ta4sogtlc55f)

[Managing expectations for follow up](#h.pwt92muulxvg)

[Diversifying and growing the community](#h.pr63x2bbwdg6)

[Scoping to Balance Flexibility and Specificity](#h.xq45k7r8imit)

[Remote participation in hackathons](#h.w8tl56ambmsv)

[Using face-to-face time productively](#h.olfl15m9n7hq)

[Going further](#h.kw73foprnaux)

[unclassified stuff for "going further"](#h.kys4cl6vymfh)

[hacking med example](#h.qjqsny942qbo)

[less preconception](#h.f05tg8snv6uk)

[issue trackers and email lists vs. wikis](#h.skjx8gdciggi)

[How do we quantify the impact of code?](#h.ynnwltkw88kf)

[What makes the NESCent model different?](#h.gfrk8w45bhw7)

[Career stage of participants](#h.g5flg8p0g4di)

[Data we wish we had collected](#h.akcnkoi79u0q)

[Conclusions](#h.tiw5d3aup7km)

[Possible data figures/tables](#h.acc04gg89ls9)

[Diversity variation between meetings](#h.of3jks61fr62)

[Variation in hackathon parameters](#h.qj900hz22wsl)

[Sustainability of technology adoption](#h.wlzcwkwnspoz)

[Sustainability of projects initiated during hackathons](#h.n1bot73jqu0a)

[Impact of hackathons on building community and collaborations](#h.3b7gkjqtxx2o)

[Resources](#h.hig2ijl6g607)

# Abstract

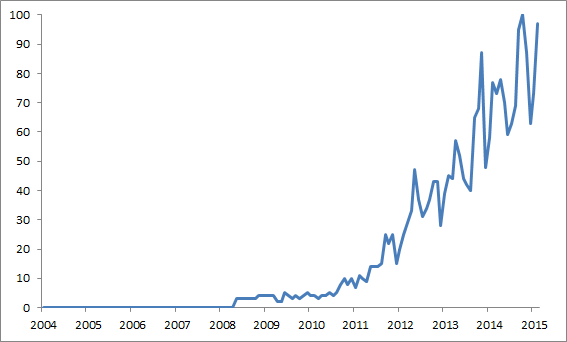
Hackathons have become popular instruments to stimulate creative problem solving, engender engagement with a sponsor's objectives, and build and develop communities and social networks. The kinds of problems being targeted, the people whose engagement is being sought, and the types of objectives sponsors aim to promote all vary widely. Models, processes, and formats based on which hackathon events are run also vary widely. Unfortunately, there is scant data, and even fewer scientific studies, on which way to conceive and execute a hackathon is most conducive to, or tends to preclude which kinds of outcomes. Here we report on the experience of planning and running a cumulative total of 9 hackathons aimed at fostering and nurturing a domain science community of practice, and its capacity to collaboratively develop and sustain its informatics resources. We define the model and process we used throughout these hackathons, and describe the motivations and goals that underlie them. To enable future more comprehensive studies, we present the data and evidence we collected, and discuss which conclusions about the effectiveness of the events they suggest. We end with recommendations on which data could be collected at future hackathons to eventually enable an evidence-based approach to the various decision-making points involved in planning and running a hackathon.

# 

# 

# Introduction

Since the early 2000's, *hackathons* (also called *hackfests* or *codefests*) have become increasingly popular in academic, non-profit, corporate, and government sectors (Figure 1). Hackathons are software development events that emphasize spontaneity and collaboration, bringing together developers (sometimes also users) with the goal of achieving innovation in software development, often coupled with other objectives such as fostering community or drawing attention to a sponsor's data or services. Despite the increasing interest in hackathons, there has been minimal formal literature about these events: a Web of Science search for "hackathon", "hackfest", and "codefest" reveals only 11 papers through 2014 (1 each from 2007, 2012, and 2013, with the remaining 8 from 2014).



*Figure 1: Google trends results for the term "hackathon" from January 2004 through February 2015. Values are relative to the highest point on the chart, thus the month with the greatest search interest in the term receives 100% and other months are scaled accordingly.*

As currently practiced, hackathons vary in many ways, including duration (1 to 5 days), size (from dozens to thousands of participants), criteria for admission (from invitation-only to completely open), goals, incentives, and meeting structure. Some of them are planned for months in advance, and others emerge much more spontaneously. A variety of brief guides and "how to" documents describing hackathons can be found online {link to a list}, and there are several peer-reviewed publications that represent reports on a hackathon or its products {e.g., Lapp, et al 2007, phylotastic}. Thus, prospective organizers face an array of choices when considering how best to organize and execute a hackathon. Such choices presumably shape the technical, social and community impacts of the hackathon.

Yet, it is difficult to see how best to make such choices. Strangely, in spite of the growing enthusiasm for hackathons—which suggests that they have valuable impacts—there is no clear consensus on *precisely why* hackathons are valuable. For instance, the most obvious impact of hackathons is to produce useful code, but a common complaint of organizers is that hackathon teams typically generate prototype software that remains unused after the hackathon ends {ref}. Apparently hackathons may be considered worthwhile, even when they often fail to deliver on their most frequently stated goal.

Of course, hackathons have many other potential impacts. For instance, when a hackathon is focused on utilizing a sponsor’s newly released API, the event may reveal bugs and bring valuable exposure to the sponsor, even if it doesn't yield sustainable client software. There are many benefits to participants. Even if a team of hackers abandons a prototype, team members leave the event with a unique experience that enables them to team up with others and build a related-but-better prototype, *e.g.*, for a different platform. They benefit from specific technical skills picked up during the event, from sharing best practices, and from making personal connections with colleagues. Furthermore, there are potential benefits to a larger **community** from discussions and interactions that spread technical knowledge and grow a community-wide awareness of domain-specific challenges, opportunities, and best practices. Is the main strategic objective of a hackathon to produce working code, to have the enabling experience of producing working code, to augment participants' technical skills, to grow community, to practice teamwork, or something else? Is it all of these things together?

This uncertainty regarding the value of hackathons adds an extra layer of difficulty to planning and executing an event in such a way as to advance the goals of participants, organizers, and sponsors. For instance, one wishes to know, not only how important is teamwork, but whether there are particular practices (*e.g.*, regarding scoping, seating arrangements, or meeting format) that tend to improve the prospects for effective teamwork.

A systematic study of the organizational practices of hackathons, and their effect on measures of impact, would be valuable as a source of guidance to organizers. However, in spite of some initial work on hackathon impacts {trainer, et al}, no such systematic study is available.

To address the need for evidence-based guidance, and to lay the foundations for more systematic studies in the future, we offer here an intensive case-analysis of hackathon practices and impacts, based on a series of well documented hackathons that followed what we call the "NESCent model", referring to the National Evolutionary Synthesis Center, an academic research center. Over a 8-year period from 2006 to 2014, NESCent organized 8 hackathons, each hosting ~30 participants for ~5 days. Each event was focused on scientific software development in the domain of evolutionary biology (comparative analysis, phylogenies, evolutionary modeling, population analysis, etc). Each event was planned by a different leadership team whose membership intersects with the set of authors of this work (*i.e.*, each leadership team included at least one of us, and most included several of us). Due to the nature of the sponsoring organization, intangible outcomes such as promoting diversity and building a "community of practice" were as important as producing code.

Because the organizers made an early commitment to public sharing of resources online, the NESCent hackathons are well (albeit inconsistently) documented, with a trail of wikis and other resources that provide participant lists, technical presentations, pitches, code repositories, team reports, and even notes from meetings of the organizers. From these sources, we have gathered a systematic set of data on NESCent hackathons (8 events, 48 teams, 206 participants, and XXX products) that aims to provide a foundation for systematic research on hackathon impacts that may be undertaken in the future.

Below, we summarize the documented impacts of 8 hackathons that follow the NESCent model, and reflect on some of the lessons we have learned as organizers and participants. We begin with a detailed guide to the NESCent hackathon model, describing key steps in planning and execution, illustrating key points by referring to past hackathons, and explaining the motivations behind chosen practices. Note that, because we have studied only one type of hackathon, we cannot draw conclusions about which kinds of hackathons are best. However, we can provide evidence on the impacts of the NESCent model, and in some cases, we can provide evidence for the effectiveness of specific practices.

# Methods

*Table 1. The 8 hackathons which form the basis of this study. More detail on individual events can be found in Supplemental Document X.*

|  |  |  |
| --- | --- | --- |
| **Date** | **Nickname** | **Title** |
| 12/2006 | phylohack | NESCent Phyloinformatics Hackathon: Lowering the Barrier |
| 12/2007 | comparativer | NESCent Hackathon on Comparative Methods in R |
| 3/2009 | vocamp | Phyloinformatics VoCamp |
| 11/2009 | dbinterop | Evolutionary Database Interoperability Hackathon |
| 11/2010 | gmodtools | GMOD Tools for Evolutionary Biology |
| 6/2012 | phylotastic 1 | Phylotastic: the Tree of Life, as you need it |
| 1/2013 | phylotastic 2 | Phylotastic 2 |
| 9/2014 | treeforall | Tree-for-all Hackathon |

An early commitment by the organizers of these hackathons to open science has led to the persistence of a large amount of public information on NESCent hackathons (Table 1) and their products. For instance, agendas and other documents were developed and shared on NESCent's public wikis; event rosters were shared publicly, so the composition of all the teams is public knowledge; teams composed reports using public wikis, and were encouraged to share code in public source-code repositories (SourceForge, Google Code, GitHub, BitBucket). From these sources, we have gathered a systematic set of data on NESCent hackathons, including (1)events (names dates, and locations for hackathon events); (2) projects (names and descriptions of hackathon team projects); (3) products (outcomes or products of hackathons and follow-up activities); and (4) participants: names and affiliations of participants.

## Data collection and encoding

Because the set of authors of this manuscript overlaps with the organizers of each of the NESCent hackathons, we began this project knowing exactly how many events were held, and where to look for information. Most of the NESCent hackathon information is discoverable by following links outward from the hackathon page of NESCent's bioinformatics division (need URL). We also had access to NESCent's NEAD database of participants, programs, and outcomes (though this is only a small subset of outcomes reported here). The majority of information on team projects (roster, goals, repositories, team reports) is readily available from these resources.

In several cases, gathering data required other methods than simply following links and copying content.

* In a number of cases in which teams did not provide a succinct and easily readable statement of purpose or goals, we constructed a statement based on the materials provided.
* To determine the involvement of remote participants in 3 of the hackathons, in the context of poor documentation of remote involvement, we consulted our memories and provided this information to the best of our recollection.
* To find github handles (which were requested only for 3 hackathons) for all participants, we searched github using names or email addresses (because names are not always unique, we only accepted a name-based match in the case of confirmatory evidence such as a matching affiliation, or involvement in a known hackathon repository).

The collection of information on products (outcomes) was the most haphazard. While code repositories and reports generated at the hackathon typically are linked into the hackathon web page at the time of the hackathon, and thus are easily discoverable, it would not be easy for an outside observer to identify outcomes that emerged after the event, *e.g.*, conference presentations, blogs, or grant proposals. In this case we simply relied on the NEAD database and our familiarity with the hackathons and their outcomes. However, we cannot make a claim to be complete. Our data on persons, teams and hackathon projects is relatively complete, but the information on subsequent impacts and outcomes is likely incomplete.

All of the data, with the exception of diversity data collected by NESCent and stored in NEAD (which must be kept private), is available at (repository).

# The NESCent model for extended, carefully scoped hackathons

The following is a step-by-step description of our approach to instigating, planning, executing and following up a hackathon. This approach has evolved over a decade. The description below largely reflects the final model, which incorporates some lessons learned over the years (see Discussion). Unless otherwise specified, references to hackathons below specifically refer to the NESCent model and should not be taken to reflect hackathons in a more general sense.

## Overall Structure and Roles

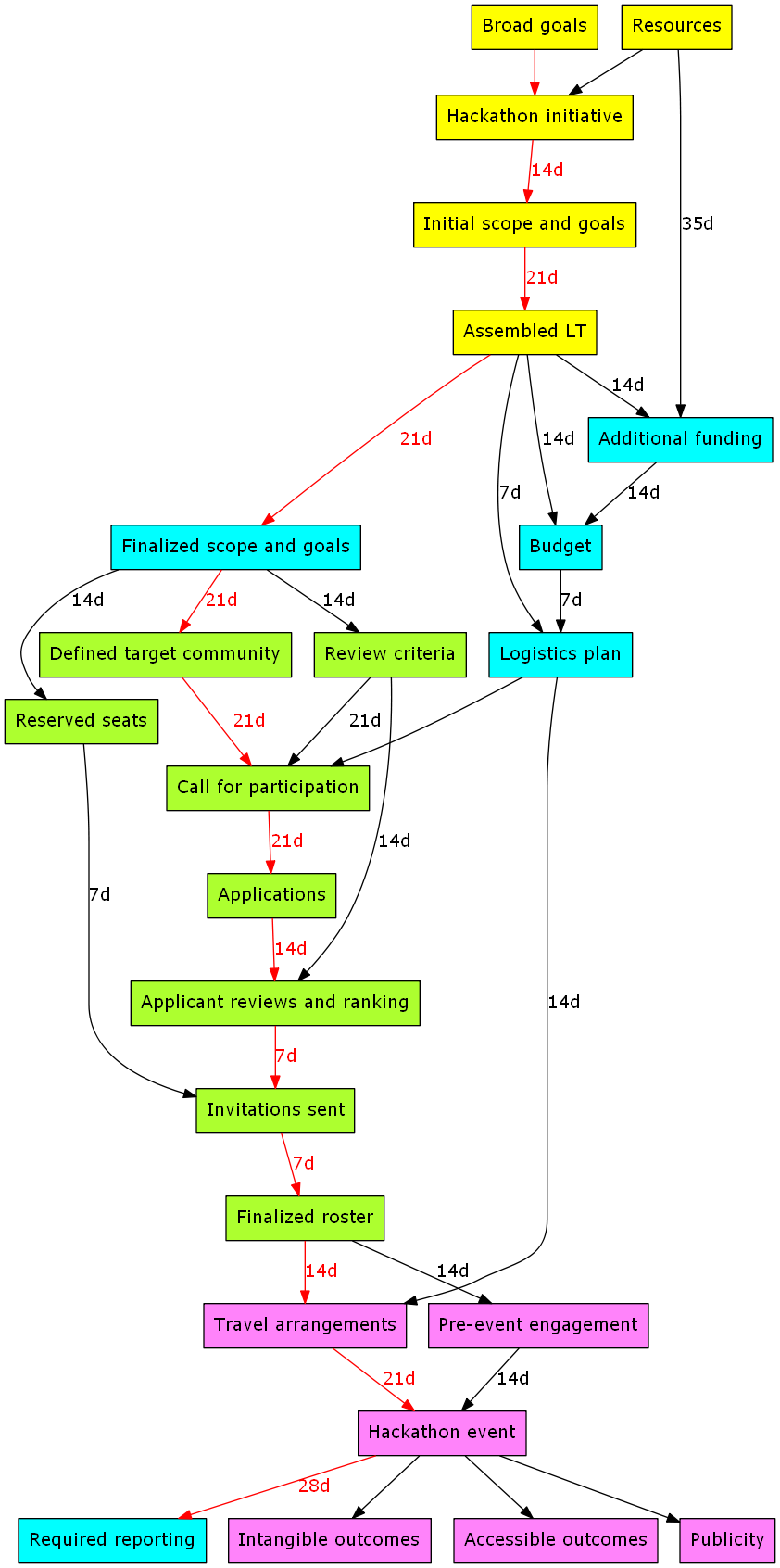
The process below makes reference to a number of roles. Planning typically begins with **instigators** (or a single instigator) who solicit support from **sponsors**, and then assemble a Leadership Team (**LT**) of **organizers** who carry forward the planning process and eventually recruit **participants**. Hackathon activities are managed by **facilitators**. The roles may overlap. The instigator may be an agent of the sponsor; organizers often attend the hackathon as facilitators and participants.

For participants, the week-long hackathon event is the main focus of activities. For organizers, the hackathon is the culmination of a planning process that begins months before the event, in which a scope is determined, the event is advertised, participants are recruited, and preparations are made to ensure a successful event. Table 2 illustrates an estimated timeline of events, to allow for better planning. While certain planning activities can be accelerated, one might view these as minimal estimates to allow proper preparation.

*Table 2. An approximate timeline of major planning milestones for a hackathon. The timeline works backwards from the hackathon itself (at week 0) to give potential organizers an idea of how far out they may need to start planning.*

|  |  |  |
| --- | --- | --- |
| **Week** | **Milestone** | **Description** |
| +3 | Report | Develop a report to the sponsoring organization |
| 0 | Hackathon start | Begin the actual hackathon event |
| -2 | Engagement | Initiate pre-event engagement with participants |
| -3 | Travel planning complete | Finalize travel plans for all participants |
| -4 | Finalize Invites | Finalize invitations |
| -5.5 | Applicant Decisions | Finalize decision on whom to invite |
| -6 | Review Deadline | Reviewers to finish all applicant reviews |
| -7 | Application Deadline | Deadline for applicants to apply |
| -8 | Secord RFP | Re-issue the solicitation for applicants |
| -11 | Issue RFP | Disseminate the solicitation for applicants |
| -12 | Recruitment Strategy | LT finalizes its recruitment materials and timeline |
| -14 | Final Scope | LT finalizes scope and goals of hackathon |

Figure 2 illustrates in greater detail the flow of the planning process. Certain elements may occur out of order, in parallel, or be skipped, as appropriate to the nature of the conceived hackathon. The figure is meant as a guide to the major sections below.



*Figure 2. A flowchart illustrating the various planning steps for a hackathon. Colors indicate ???.*

## Pre-planning

Instigators typically begin with an idea that they believe will inspire broader participation. This idea may become a formal proposal for funding, authored by the instigators. In the case of NESCent hackathons, the instigators came from NESCent's bioinformatics staff, or from two working groups focused on evolution-related bioinformatics, the EvoInfo (evolutionary informatics) and HIP (hackathons, interoperability, phylogenies) working groups, which organized 3 hackathons. The goals of the instigators overlap on themes such as facilitating the sharing and re-use of data, and facilitating interoperability among resources.

### Resources

Identifying resources is a critical phase of the process and involves identifying funding entities that are willing to provide the funds, facilities, and staff support (administrative and technical) necessary to execute a successful hackathon event. There are a number of possible funding sources for a hackathon. These include:

* Major funding agencies, where a hackathon could be either part of a larger project (*e.g.*, as part of outreach for an NSF or EU grant) or the main focus of the funding request
* Organizations that are dedicated to support working group initiatives (*e.g.*, NESCent)
* Organizations that support the community of interest (*e.g.*, the Society of Molecular Biology and Evolution expressed interest in supporting a hackathon as a satellite meeting)
* Non-profit or for-profit organizations that have an interest in the broad scope of the activity (*e.g.*, iPlant)
* Other stakeholders (*e.g.*, universities, research labs)

In the case of the hackathons described here, the main sponsor was NESCent. However, other sponsors were involved in several cases, *e.g.*, the 2009 VoCamp was sponsored partly by TDWG, while the HIP working group obtained support from 2 other sponsors (iPlant and BioSync) in order to expand its budget sufficiently to run 3 hackathons. It was understood that well qualified applicants from the sponsor organizations would be included in the hackathons, either by direct invitation or through the application-and-review process. A key part of the agreement between iPlant and HIP was that one of the hackathons would be held at the academic home of the iPlant project, to facilitate participation of iPlant personnel. Some of the authors also obtained funding from a professional society for a hackathon that never took place due to the inability to establish a functioning LT (see Lessons Learned).

### Assembling the Leadership Team

The use of a leadership team (LT) to plan and execute the hackathon arises in part due to the desire to broaden decision-making authority, and in part due to the lack staff to carry out plans (with the exception of travel arrangements, for which we traditionally have had administrative support, as described below). The LT makes a plan, and also enacts its plan. For the LT to take true ownership of the project, it must be allowed to re-think the scope and goals (see below). Therefore, assembling the LT is a critical step in which the instigators attempt to diversify the sources of authority and responsibility. Difficulty in assembling a committed LT, or in getting an LT to reach closure on scope and goals, is an indication that the instigators' plan may be flawed or premature (see Lessons Learned).

In recruiting organizers, the instigators may consider:

* **Expertise:** the broad topic of the hackathon will inform a set of expertise that is necessary to formulate the appropriate scope of the event; that expertise will also facilitate the identification of potential participants to invite to the event.
* **Diversity:** diversity in the LT is critical to provide the connections and the perspectives necessary to facilitate the recruitment of a diverse group of participants.
* **Vision:** members of the LT are in charge of establishing the scope of the hackathon. As such, they need to be capable of providing a general vision that is both realistic/feasible as well as groundbreaking.
* **Management Experience:** the members of the LT will provide the overall supervision and management of the hackathon event. Over several months, LT members will be relied on to accomplish various milestones, and must be reliable performers. Experience in team management, group formation, event management, and other relevant management skills.
* **Cultivating new leaders:** the instigators periodically recruit individuals who are young or inexperienced, but show promise as organizers.

When recruiting, we ask prospective LT members to make an explicit time commitment and to be available during the crucial period of application review. During their tenure, the organizers will participate in perhaps 5 to 8 LT meetings (typically teleconferences). For every hour of meeting time, LT members can expect to spend 1 to 2 hours executing the LT's plan (*e.g.*, reviewing applications, making meeting arrangements. Thus, the total time commitment is roughly 1 to 2 hours per week over 4 months.

### Setting Scope and Goals

A critical aspect of the hackathon event is a decision on its precise scope and goals. The choice of the hackathon theme should reflect (1) the overall objective of the project, aligned with the commitment to the funding entities; (2) the feasibility of accomplishing the objectives within the timeframe of the hackathon; (3) the needs of a community; and (4) be general enough to enable participants to recognize themselves, feel engaged, and see a value for their participation in the event.

How a hackathon is scoped should be directed by its specific objective and goals as there most likely will be significant tradeoffs among all potential objectives. A well-defined scope can ease the decision making process regarding those tradeoffs as well as manage the expectations of potential participants. A scope should be flexible enough to allow participants to employ creativity and discover group goals thus enabling projects of common interest, yet limited enough that it meets the overall objectives as determined by the LT.

See "Controlling Scope to Balance Flexibility and Specificity" in the discussion for further detail and examples.

## Planning

Once there is a leadership team in place, and the scope and goals of the hackathon have been determined, planning the event and recruiting participants can begin.

### Logistical Considerations

#### Facilities and equipment

Some aspects of choosing a facility are the same for any meeting. Universities and other institutions often have event coordinators to provide advice on such matters.

The typical hackathon setup includes the flexibility to seat everyone in a single room, and also to divide participants into teams of ~3–6 individuals. Ideally it will be possible for participants to organize tables and chairs as they see fit. Some teams will prefer to work at a single table within the larger room, while others may prefer a smaller separate workspace in an adjoining room or hallway. Ideally, these smaller rooms will be close enough that teams can check in with each other, make plans for meals and breaks, and so on.

The facilities must provide a room with a sufficiently open floor plan for pitching. Pitches (~10 of them) will be displayed on easels, on available mounted whiteboards, or on large adhesive notes stuck to the walls. Participants must be able to cluster around one pitch without bumping into another cluster.

Some alternative space configurations are inadvisable. A room with fixed stadium seating, for instance, is unsuitable, no matter how large. Other room configurations tend to create or amplify inequalities, *e.g.*, a room with a single table large enough for most, but not all, participants would leave some participants, literally, without a seat at the table.

*Anecdote: during the opening session of one hackathon of about 30 participants, the room was arranged with chairs in standard rows. It was observed that, with the exception of a single person who was on the leadership team, every woman at the hackathon had chosen a seat in the back two rows of the room. The LT noticed this and made a point of ensuring that this lack of integration did not continue throughout the meeting (it did not).*

Hackathons do not require much in the way of equipment or supplies. Participants are expected to bring their own laptops or tablets. The meeting rooms must have excellent wireless access, numerous electrical outlets, and extension cords, and at least one projector for full group presentations and bootcamps. For the pitching session, there must be marking pens, along with a suitable number of easels or whiteboards, or a suitable amount of accessible (and flat) wall space to post adhesive notes.

#### Communication Technology

The nature of a hackathon usually makes heavy use of a variety of communication technologies. We recommend for organizers to designate a set of preferred technologies in advance and encourage all participants to use this set. **Consistency among participants is extremely important,** both for general communication and for archiving. As an example, in one hackathon involving remote participants, a particular subgroup was using one communication technology before lunch and then switched, without announcement, to another technology after lunch. Remote participants had no way of knowing this and were left out of discussion until this was discovered. In general, we encourage technologies which are free, open, and multi-platform, although the needs of a particular project may preclude some of these factors. Again, consistency across hackathon participants is key. Major categories of communication technology to consider include:

**Version Control:** There are many types of version control available. Among the most popular, and the one we have used in recent years, is Git and Github. SVN is also very popular. Earlier hackathons used GoogleCode and SourceForge. What is most important is that everyone at a particular hackathon use the same system, because it is critical to tracking production and outputs and helps no one if the data and output is spread across multiple systems. The last three hackathons primarily used github, because the social layer allows individuals to connect in a direct manner and the decentralized nature of git, encourages "forking" projects for experimental development and submitting successful experiments for merging back into a project's code base by way of "pull requests".

An additional benefit of using github is that it is relatively easy, via the github API, to access information about contributions to code of individuals involved (*e.g.,* their extent of usage of github before, during, and after the event), the development of the repos worked on (the history of commits to the repos before, during, and after commits), and collaborations (looking at degree to which hackathon participants worked in github on the same repos before, during, and after the event). This makes assessment of certain kinds of impact easier and more readily allows automated and systematic data collection.

**Document Editing:** Shared document editing capabilities is essential, whether for note taking, concept planning, documentation, etc. There are many such technologies we have used, including wikis, Google Docs, Mozilla's Etherpad, Github documents, and others. The most important thing is that everyone use the same technologies, otherwise documents get scattered across too many systems in too many different formats. The ability to use an environment for the collaborative development of documents, which is persistent, is also important to provide a common point for sharing information before, during, and after the hackathon. For example, the wiki pages created during several hackathon events (*e.g.*, http://www.evoio.org/wiki/Phylotastic1#Phylotastic\_pages, https://www.nescent.org/wg\_evoinfo/Database\_Interop\_Hackathon) represent a permanent record of project ideas, bootcamp materials, participants, and directions for follow-ups.

**Communication:** There are many systems for direct communication, including Skype, Google Hangouts, IRC, etc. Even telephones. These will be primarily used pre- and post-hackathon, but can be used during, either to communicate with remote participants or sometimes when teams are scattered throughout different rooms. Again, the most important thing is to decide in advance which protocols will be used and adhere to them, otherwise some members may not know where or how to tune into important conversations. Social networks may play a role in a hackathon through drumming up broader interest and exposure, but usually do not serve as a primary means of communication among participants during the event.

For many of the hackathons we have used wikis for completely open document planning and note taking; this has created a historical record of many of the events that would otherwise be lost if we had stuck with documents on laptops or other closed sources. Despite this, we still find that basic data about the hackathons can be hard to compile or is missing because it never occurred to anyone to record these data in a systematic way. For this reason, we strongly encourage organizers to be systematic about recording data about their events. See "Data we wish we had collected" below for more discussion.

#### Travel

The presence of a central organization (*e.g.*, NESCent) that can handle the logistical aspects of the hackathon is strongly encouraged. Travel arrangements should be consistent with the duration of the hackathon, ensuring that participants are at the location on time for the beginning of day 1 of the event (which is critical to the overall event). The travel arrangements should also emphasize that the participation to the complete event is strongly encouraged, as often the projects start shaping only after the first 2 days of the event. Locating all participants in the same residential venue (*e.g.*, the same hotel) will also facilitate interactions outside of the daily hackathon activities and develop a social infrastructure to the hackathon.

### Identifying Participants

The basic approaches to recruiting participants are (1) directly inviting from a list of named individuals, (2) issuing an open call for applications, or (3) a mixture of the two. Direct invitations are useful to ensure that the hackathon participant pool covers essential expertise, e.g., if the scope names a particular data resource, the organizers may wish to invite experts on that resource. The danger with direct invitations is that it will perpetuate privileged relationships and limit the opportunities to build new professional relationships among participants. An open call is better for community-building and outreach. Also, those who respond to an open call and fill out an application clearly are motivated to participate. The danger here is not receiving enough applications (though our experience did not include this problem).

Generally, we have used a mixed model, where we target invitations (1) to people who have particular domain expertise and (2) as a means of increasing diversity, but always including an open application process to reach out more broadly and engage unexpected members of the targeted community. One needs to be careful in this model because if you target too many people with direct invitations, there may not be enough space for those responding to the open call. Generally, if you directly invite someone to apply it is difficult to later not accept them as an event participant.

Our advertising has been predominantly through popular mailing lists in the appropriate academic discipline, as well as spreading word through emails to colleagues. Generally the first call for applications would be sent about 3 weeks before the deadline, with a second call/reminder sent at 1 week. Direct invitations would go out at the time of the first call, but additional invites can be sent later as other names/suggestions are found. An example of such a call can be found in Supplemental Document X.

#### The Application Process

We have used a fairly simple application form created through Google Docs for screening participants (Supplemental Document X). Beyond typical fields such as name and contact information, we were interested in the following major fields: (1) availability: can the person attend the entire event; (2) a general checklist of skills/expertise that were applicable to the hackathon theme; (3) personal goal for attending the hackathon: why do they want to attend and what do they want to get from it; (4) demographics (lesson learned: we did not ask for demographic info in earlier applications, but found that diversity suffered when we did not have information to help balance decisions); and (5) where they are coming from (for budgeting purposes) and whether they have support to attend without our funding. We also specify that applying to the hackathon is a commitment to open source licensing of software outcomes generated from the event.

At one point we asked for proof of expertise, but it was a bit off-putting to participants and proved ineffective.

#### Criteria for Choosing Participants

Choosing participants for a hackathon is more complicated than simply choosing expert programmers. A number of criteria need to be considered in trying to optimize productivity. These include (1) The ability to work with others: some people may be great at working alone, but unable to work in a group setting, which makes them a poor choice for such a group project; (2) Balancing coders and users: having people who will be using the product work directly with the people who are programming them can better facilitate understanding how the two halves go together. Programmers can learn about the way users will want to use a product and the ways in which they will be capable of using it (a user interface), while users will learn more about what features are feasible for a product; (3) Diversity: diversity includes gender, ethnicity, experience, and research discipline. If one of the main goals of a hackathon is to increase the community engagement in a product, system, or concept, it is important to recruit from the broader representative community. This is not always easy. In a general call for applications, one often finds that those who view themselves on the fringes of the “hacker” community will not apply or feel that they cannot contribute, so effort needs to be made to directly encourage and/or invite them.

Successful participants to invite are motivated people who will gain a long-term advantage of the experience, and possibly already have connections to other participants or projects.

It is generally advantageous to mix participants of different career stages as it tends to increase networking and community engagement significantly more than a hackathon of all senior-level participants. Mixing newcomers with those who have more experience will also increase community size. One incentive to participate in a hackathon for younger participants may be the opportunity to interact with established researchers and programmers. When one is concerned about tangible products as part of the outcome (as defined earlier), a balance of career stages may be of particular importance. Younger participants whose future career and job success may be more dependent on tangible outputs will be more motivated to continue with and finish projects after the hackathon has ended; senior participants for whom the output is more of a bonus than a necessity are more likely to fail in post-hackathon engagement due to busy schedules (note: there is evidence of this for working groups...should apply to hackathons as well). Our experience is predominantly with hackathons in academic settings, however our hackathons have been a mix of faculty, postdocs, students, and research staff. Research staff generally have greater limitations on post-hackathon time commitments and thus will be less likely to be able to commit after the hackathon is over; related to the earlier point, they are also less likely to generate a career benefit from a product. The major exception to this may be when the focus of the hackathon is explicitly on the technology or codebase that directly supports the staff; in this case, continued support and development post-hackathon may be a direct expectation of their job.

Generally, we have found that the entire LT does not have to participate in individual hackathons. A minimum number of members of the LT is critical to ensure that the discussion and activities of the hackathon are consistent with the original scope and that teams are formed and operate in a consistent and positive manner. It is our experience that participating LT members actively contribute to individual teams, and occasionally they have provided pitches for projects.

#### Process for Choosing Participants

Applications were divided among LT members in a staggered fashion such that each applicant was reviewed by three different members. Applications were scored on a 3 point scale (3 = top, 2 = solid, 1 = poor; half point intervals were sometimes used), based on the criteria described above, with a textual explanation expected for the specific ranking.

Applicants were ranked from high to low based on their average score, with participants chosen during a conference call. In order to allow the LT to best balance the needs of expertise, experience, and diversity, the ranking was used as a general guide, but not as an absolute decision making criterion. Alternates were identified in case participant slots became available after invitations were extended.

### Pre-hackathon Engagement

Pre-event engagement is an opportunity for

* participants to overcome initial shyness and begin communicating
* organizers to identify gaps in domain knowledge or technical expertise
* participants to begin discussing possible projects
* expectation management

The identification of gaps in technologies and methodologies can lead to a plan for technical talks (aka "bootcamps") on hackathon day 1.

Several remote technologies are available for pre-event engagement. We have used email lists, teleconferences, issue trackers (OpenTree hackathon), and collaborative wikis. To use project-specific issue trackers, it is necessary to cajole everyone into joining the project (*e.g.*, via github).

Generally, we focus on just getting people to communicate, and we keep expectations low. With repeated reminders, organizers can get nearly everyone to make an online introduction on an email list (and this is much easier at a live telecon). However, pre-event discussions are likely to involve only a subset of participants. In our most extreme efforts at pre-event engagement (dbhack1 and phylotastic), we had a mailing list and organized several telecons, with the result of engaging the majority of participants.

We are confident that pre-event engagement is highly useful *for organizers* to identify gaps, and to get a sense of the opportunities and challenges presented by a specific group of participants. We are less sure about whether pre-event engagement is beneficial to participants or to project teams.

## The Hackathon

### Day 1

The goal of the first day is to organize a set of teams, where each team has a membership of 3 to 6 participants committed to a clearly defined set of goals that will result in tangible products ("deliverables") by the end of the hackathon. Achieving this goal is critical for the success of the hackathon, and requires active facilitation.

#### Welcome and introductions

The first activity is usually a brief introduction by the organizers, typically aimed at explaining the scope and goals of the hackathon. This may be followed by a chance for participants to introduce themselves (*e.g.*, 3 sentences on background, reasons for attending, hopes and expectations).

#### Technical presentations

In advance of the hackathon, the organizers will have identified speakers and arranged for technical talks (sometimes called "bootcamps") to address gaps exposed during pre-event engagement activities. For instance, at dbhack1, there was a talk on a relatively new and unfamiliar file format called NeXML {ref} that was an emerging standard for data exchange. The set of technical talks may take from as little as 90 minutes to several hours.

Some flexibility is useful. In some cases, it is helpful to arrange for a technical talk that may or may not be delivered, depending on needs expressed at the event. In a few cases, an unanticipated need has arisen during the hackathon. Typically these are of interest only to a subset of participants and can be arranged on an ad hoc basis.

#### Open discussion

The next major activity is a relatively open discussion about how to interpret the scope of the hackathon, what kinds of opportunities exist to address goals of the hackathon, and what technologies are available to address key challenges. Also, pre-event engagement (above) may have exposed specific topics of uncertainty or confusion that would benefit from discussion at this time. The organizers should allow at least an hour. The facilitator of the discussion should solicit questions and take other steps to ensure that there is broad participation.

At this stage, ideas for pitches will emerge. The facilitators should encourage discussions of scope, goals and technical approaches of specific pitches. Attention to relevance (scope) is important at this stage. However, detailed technical discussions should be discouraged unless they are critical to establish feasibility. If a participant knows that a technical challenge can be solved, the exact solution does not need to be discussed at this time.

The appropriate time to end the discussion is when participants are ready to proceed to the next stage, which is pitching (*e.g.*, the facilitator may ask if anyone would like more time before going on to pitches).

#### Initial pitches and feedback

To "pitch" an idea is to propose it for wider adoption. Proposed hackathon projects are called "pitches", and the proposers are pitch-makers or champions. Anyone can make a pitch, but in practice, they tend to be made by more senior people with a more confident sense of what projects would have an impact. In our hackathons, there are often a few pitches from organizer-participants, but this is not necessary.

In the first stage of pitching, anyone who wants to make a pitch (a “champion”) is given a minute or two to make a proposal, ideally with (1) goals linked to deliverables or tangible products; (2) an approach to reach the goals (and likely challenges); and (3) what expertise or knowledge is needed to fill out the team. Each pitch is followed by an opportunity for questions. Questions should focus on clarifying what the project aims to do, and how it aligns with the scope.

Champions may decide to drop their pitch (*e.g.*, if feedback suggests it is infeasible, or if the champion finds another pitch more attractive). The facilitator should ask if any pitches will be dropped. If a large number of pitches (>1/3 the number of participants) remain in contention, the facilitators may wish to start a discussion about how different pitches could be combined.

#### Team formation and pitch refinement

What follows is a spontaneous process of team-formation and pitch-refinement, the goal of which is to end with a set of teams (each with 3 to 6 participants), each with an exciting, tractable, well scoped project. This stage typically takes an hour.

First, champions should be given materials to post a sketch of a pitch (*e.g.*, on an easel or whiteboard) in a room large enough for each pitch to be posted separately with space for people to gather around it. Pitches that might be combined should be clustered. Large adhesive notes (stuck to walls) work best for this process, because they are easy to move, and can be supplemented with further notes as needed.

Then, participants are free to wander around the room, discussing pitches, offering suggestions, and deciding how to fit in. This is a spontaneous, interactive process. The fit of participant to proposed project is not like the fit of a key to a pre-built lock, because the definition of the project is still in flux. Instead, it is an "induced fit", *e.g.*, the interaction of participant and champion may result in a modified plan that incorporates the participant's unique skills and interests. Champions should be instructed to focus on matching a team with a strategy, not on solving technical problems that can be solved on later days.

A champion may abandon a pitch to join another team: the pitch will be abandoned if no one else steps in to champion it.

At this stage, facilitators may wish to intervene to ensure that (1) newbies and relative outsiders are actively engaged in creating-finding a place for themselves; (2) projects remain aligned with the scope; (3) experienced individuals are finding a place on a team rather than planning a project on their own (this occasionally happens, and our experience is that it is not productive). Newbies who feel that they have little to contribute should be encouraged to find the place where they have the most to **learn**, *i.e.*, each participant should maximize contributing, or maximize learning.

If there is any time left in the day, participants may use it to refine their plans or begin execution.

### Intermediate Days

Most of the rest of the days of the hackathon are spent with teams working on their individual projects, with occasional time taken to reassemble as a whole group to update progress. These stand-ups are generally meant to be short and, we have found, generally only need to be performed about twice per day (to avoid wasting too much time on updates): once before breaking for lunch and once at the end of the day. Stand-ups are an opportunity for projects to crow about their accomplishments, and to share challenges in the hopes of receiving useful suggestions.

### Last Day

At the end of the hackathon, a general assembly to wrap things up, discuss overall progress, future plans, etc., takes place. Typical components of this final event include

* Final reports (presentations) from the different teams
* Identification of needed followups from each project
* Identification of expected tangible products that can be achieved in a short time after the hackathon (*e.g.*, publications, presentations at conferences, commitments to codebases).

Hackathons have also included more general discussions about long-term followups (*e.g.*, identification of potential funding sources that would enable scaling up some of the development efforts). It is recommended that participants assume responsibility for the completion of components of the various projects (*e.g.*, see http://www.evoio.org/wiki/Phylotastic1#Projected\_tangible\_outcomes as an example).

## Post-hackathon Wrap-Up

After the hackathon, there are a number of activities that are either required or desirable (we address follow-on projects separately below). The most obvious *required* activities include:

* **Logistical Activities:** wrapping up the event requires providing reimbursements to the participants for the expenses incurred during the event. This is typically handled through the sponsoring organization.
* **Reports:** Sponsoring organizations will typically require the development of a final report that summarizes the activities and accomplishments achieved during the hackathon. The development of such report is typically main responsibility of the instigators of the event (even though other members of the LT may actively participate).

# Results and Discussion

## We begin with a discussion of our analyses to attempt to measure/quantify impact, and then use this data to explore several specific questions we have about effectiveness of certain features of hackathons.

## Understanding and assessing hackathon impacts

There are many possible outcomes and impacts of a hackathon (Table 3): some of these are tangible (*e.g.*, code, standards, test data sets, publications, etc.) and some are intangible (*e.g.*, increased community diversity or community awareness of technologies or best practices). The procedures for holding a hackathon that we have developed were generated specifically based on the value we give to these outcomes. Decisions on participant pools, for example, may be heavily driven on the relative weight one gives to tangible vs. intangible impacts. Organizers need to make decisions based on maximizing the outcomes they value the most.

*Table 3 presents a sample of potential hackathon outcomes, divided into tangible and intangible, to illustrate the spectrum of impacts.*

|  |  |  |
| --- | --- | --- |
| **Tangible Outcome** | **Possible Measures** | **Challenges** |
| Source code | Draft versus working quality sustained post-hackathon development activity, which could be assessed by tracking the extent of commit and other activity associated with a repo post-hackathon; community interest gathered, for example counting shares of a repo link via social media; adoption by non-participants (forks, downloads, citations). | Metrics for code writing productivity (lines of code, number of commits) are often confounded and not useful. The specific impact of contributions to larger projects is nearly impossible to track. The academic attribution system falls short for unpublished software. |
| Publications | Number of and impact metrics for publications (citations, altmetrics). | Impact and value of non-scholarly publications can be significant yet difficult to quantify. |
| Fund raising, Grant funding | Number of proposals; amount funded; increased funding rate | Measuring change of funding rate requires long time windows and disentanglement of the many confounding factors. The value of unfunded proposals is hard to quantify but likely non-zero. |
| Documentation | Amoung of text written; number of tools or methods documented; access and citation statistics for online documentation | Quality, comprehensiveness, and being up-to-date can be difficult to measure. Metrics for offline documentation are few and inadequate (e.g., downloads). |
| Data products (data, ontologies, benchmarks, etc) | Size, number of and impact metrics for data products (citations, downloads, altmetrics); | The specific impact of contributions to larger datasets is nearly impossible to track. Tracking of scholarly attribution and impact is still in its infancy, and nearly impossible for unpublished products. |
| Community standards and best practices | As publications if published, and as documentation otherwise. | Tracking of scholarly attribution and impact is still in its infancy, and nearly impossible for unpublished products. |
| **Intangible Outcome** | **Possible Measures** | **Challenges** |
| Increased Collaboration | # of novel collaborators (e.g. contributors to same github or other repo/co-publications publications) from participants post-hackathon | Tracking co-authorships without using unique author identifiers such as ORCIDs can be highly innacurate. Yet such tools are not used by all researchers, and are themselves still being developed and could be considered in their infancy. Changing handle-names, multiple handle-names can make tracking software collaboration difficult. |
| Broadening Communities / Networks | Increased participation/friends/followers of mailing lists, social networks, etc. | None of the social networks is used by everyone, and some people don't use them at all, so such analyses can (and indeed are likely) to miss interactions. |
| Community Awareness / Training in Technologies, Standards, or Best Practices | # new users of a technology |  |
| Increased Diversity | Diversity includes demographic concepts (gender, ethnicity, experience) and disciplinary concepts | Such data is sensitive, comes with many legal constraints on its collection, storage, and sharing. |
| Publicity / Brand Awareness | Conference presentations/posters, Press releases (and where they are picked up), Social media impressions |  |
| Broadening Perspectives | Idea inspiration for projects/work unrelated to hackathon |  |
| Connecting Cultures | Bridging the culture gap between "coders" and "users" |  |

The scope of the hackathon can have a large effect on the likely sustained production of tangible outcomes. The broad objective of many hackathons fall into two general categories: those that aim to enhance an existing product or codebase (*e.g.*, the R-???? hackathon was designed to enhance the XXXX in R, while the TreeForAll Hackathon was designed to build integration with the Open Tree of Life API) and those focused more on the creation of novel technologies (*e.g.*, Phylotastic I and II). Hackathons in the first category are a little more likely to have products carry forward because the existing community of the technology may be more likely to pick up on hackathon projects, even when the primary participants do not.

Beyond the direct code produced at the hackathon, tangible outcomes may continue to develop after it is over through publications, presentations, grant proposals, and documentation.. One interesting example is that the comparative methods in r hackathon led directly to the creation of a discussion-based mailing list (r-sig-phylo) which X years later has XXX subscribers and generates approximately XXX messages per month.

The intangible outcomes of hackathons tend to involve concepts such as community knowledge, structure, and adoption of technologies. A critical contribution of hackathons is represented by raising awareness of existing and new technologies/methodologies. There are several aspects and examples of this.

The hackathon event itself is an opportunity for members of a community to learn about upcoming technologies and experience them first-hand. The technologies can be either already in use by part of the community or are inherited from other communities but valuable to the audience target by the hackathon. Examples of this include (1) promoting the use of GitHub and other code repositories; (2) focusing on the use of specific toolkits and expanding their reach (*e.g.*, R, BioPerl); (3) expanding the generality and applicability of existing tools (*e.g.*, the generalization of Phylomatic achieved during the Phylotastic I hackathon); and (4) semantic Web technologies (*e.g.*, VoCamp).

In addition to this, hackathons can be developed to create support for novel technologies and methodologies that have been specifically developed to meet a perceived need of the target audience. An example of this is represented by the EvoIO stack, composed of an exchange data format (NeXML), an ontology (CDAO), and a web-service API (PhyloWS), previously developed by a working group. Two hackathons contributed either to the development of tools supporting this infrastructure (the Database Interoperability hackathon) or to the deployment application of the stack (the Phylotastic I hackathon).

### Measuring hackathon impact

We would like to be able to use a range of different metrics to assess the impact of these hackathons, as we feel they would be useful to inform evidence-based decisions made by various potential stakeholders of hackathons, including (1) organizers; (2) participants; (3) supervisors of organizers and participants; and (4) funders. Unfortunately, measuring the impact of these events is difficult. Measuring the number of lines of code or number of commits as a metric can be meaningless since these counts are very fungible based on the style and language in which individuals write code. Number of programs or scripts produced is barely better; is the code rough draft or a polished product? How error-free does it need to be? Does it need to work at all or can it be conceptual? Does what happens to the code after the hackathon matter? Does code that is used widely by the community (download counts? citations? github forks?) count the same as code that is completely ignored? There are no simple answers to these questions and the matter is even more challenging for intangible outcomes where direct measurement may be more difficult.

We have carried out some analyses using the data we have available. In most cases these analyses do not allow us to draw strong conclusions, but we include them anyway as examples of analyses that could become more powerful in the face of more complete data, and from data from more events. One clear lesson learned is that hackathon data and meta-data need to be collected more deliberately by the LT, both before, during, and after the event. This is discussed in more detail below.

## Lessons learned

### Challenges of building a leadership team

Leadership team members are often selected among more established and senior researchers. The benefit is that they have a greater level of awareness of the community and may provide better guidance in the scoping of the problem and in the identification of effective participants. On the other hand, more senior researchers tend to have an extensive agenda of commitments, which detracts from the necessary dedication required by the organization of a hackathon event which is highly focused, intense over an extended period of time, and guided by rigid deadlines. LT members need to provide availability in their schedules for regular meetings (*e.g.*, weekly or bi-weekly teleconferences) and for participation in the preparatory activities (*e.g.*, attend pre-hackathon working group meetings; prepare, disseminate and evaluate applications).

Commitments of LT members tend to change rapidly, leading to shift in focus and in the level of engagement. There have been instances where LT members had to abandon the LT due to sudden lack of time and availability. Major shifts in the LT composition may endanger the success of the event, and even lead to the cancellation of a hackathon before its start (as happened in 2014 with a planned Phylostars hackathon).

LT members need to be fully engaged and invested in the event, fully appreciating the value for the contributions that may arise from the hackathon and embracing such contributions for their importance and the benefit they may accomplish. The lack of conviction in the benefits and values of the hackathon can be causes that lead to the failure of the event.

Just as the initial success of the hackathon event is dependent on the time and effort dedicated by members of the LT, the members of the LT are also often the individuals that guide the post-hackathon activities necessary to summarize results, guide follow-up efforts (*e.g.*, development of manuscripts that present the achievements of the hackathon projects), and ensure that the hackathon outcomes are made fully available to the broad community.

### Make people feel welcome; encourage people

Hackathons have earned a regrettable reputation as unwelcoming events catering to insiders, and to men. The leadership teams of NESCent hackathons made a deliberate effort to increase diversity and create a welcoming atmosphere for women, minorities, and junior scientists (for whom the hackathon may be more of a learning experience than an opportunity to make important technical contributions).

Procedures and strategies to create a more welcoming environment were revised over the course of 9 hackathons. Ultimately they were applied at several stages of the hackathon process. First, we designed recruitment materials to appeal to a wide audience, avoiding technical language except as necessary. Our requests for participation stated that women and minorities are encouraged to apply (see also “Diversifying the Community” below). We explicitly specified non-programmer roles (*e.g.*, "domain expert", "use-case consultant"), and avoided implicitly equating participants with programmers (*e.g.*, we don't refer to them as "programmers" or "coders"). Women tend to undervalue their skills relative to men {ref} and are less likely to self-identify as "gurus", "wizards", or "power users", therefore we learned to avoid this kind of language.

Secondly, we took steps to be welcoming during the event, particularly on day 1, which typically ends with a group dinner (which, to be welcoming, should not be priced so as to discourage anyone). The facilitators pay attention to whether everyone is participating in discussion. The practice of instructing participants that they always should be "either learning or doing" makes the event more welcoming to novices who sometimes have a hard time finding a fit. During team formation, facilitators may intervene to discourage teams from unintentionally closing ranks around a pitch (some participants will commit early to a pitch and begin a deep technical discussions, sometimes with their backs to everyone else, which discourages others from approaching or getting involved).

### If you have a great hackathon, don't do it again

The first Phylotastic hackathon (event 6) was considered a major success, and the leadership team decided to follow it with another hackathon on the same theme. The scope was to build on the results of the previous hackathon, which had prototyped various parts of a system of web services, but not gone very far toward assembling them into a coordinated system. Although progress occurred at the second hackathon, there were indications that the scope was too narrow: some participants had a hard time finding teams, and one project was out of scope (soft persuasion by the facilitators failed to discourage it). What the Phylotastic project really needed was major funding for full-time staff, and the first task for such a project would not have been further development of existing prototypes, but a serious design process starting from nothing.

In retrospect, repeating a hackathon with little modification was a mistake. The organizers had a prioritized list of hackathon targets, and it would have been better to focus on the next item on their list, rather than repeat the first item. Hackathons are mainly about the chance for impact via novelty. The first hackathon was sufficiently productive that a manuscript on it was already submitted by the time the second hackathon took place. A grant proposal submitted after the second hackathon (and recommended for funding 1.5 years later, after a re-submission) was based mainly on results of the first hackathon.

### Managing expectations for follow up

A common complaint of hackathon organizers is that there is little follow up on projects after the hackathon ends. Indeed, when hackathon teams are working energetically, organizers and team members may have enthusiastic discussions of follow-ups, yet, when the hackathon ends, the team dynamics and the energy typically dissipate rapidly, and team members return to other responsibilities.

Yet, this thinking is unreasonable. The nature of hackathons is that we steal talented people from their day jobs for a limited time. We should not expect a team dynamic to persist beyond the face-to-face conditions that fostered the team. We should not expect participants to work toward hackathon goals when we are no longer supporting them to do it.

This situation encourages 2 strategies to manage expectations. Because of the low prospects for follow-up, organizers should instruct participants, especially pitch-makers, to focus on producing tangible products \*within the space of the hackathon\*. Participants should focus efforts with the expectation that tasks unfinished on the last day will never be finished. No team should commit to a goal that depends on the cooperation of outside parties or that cannot be completed within the space of the hackathon.

Yet, follow-on projects are possible. In 2 cases, hackathon projects have provided proof-of-concept and specifications that were important for obtaining funding for further development. In 2 cases, hackathons resulted directly in a publication {ref, ref}.

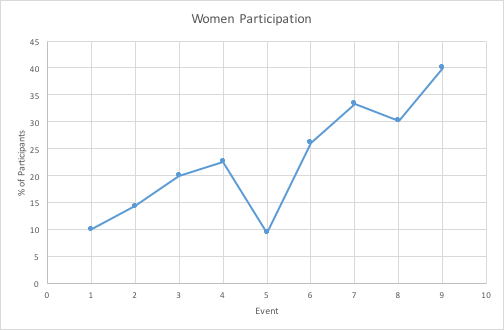
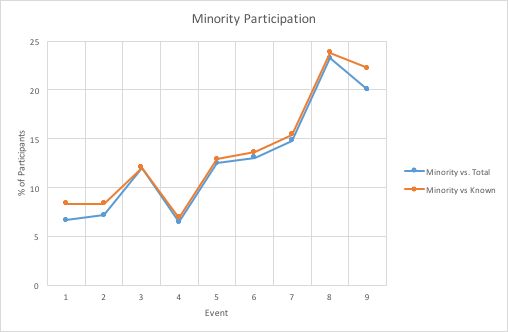
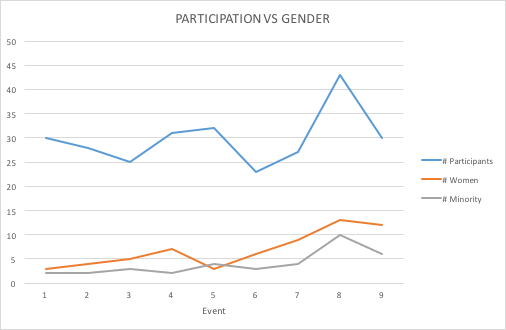
Because the cases of successful follow-up are small in number, it is hard to make generalizations. However, it seems obvious that the potential for follow-up increases when a hackathon project aligns with interests of participants, particularly when it aligns with a project that has funding, and with a leader who has the time to manage a follow-up effort.

### Diversifying and growing the community

As a group, we made a conscious decision to declare that we valued community diversity and thus challenged ourselves to increase diversity when possible through the hackathons. Diversification does not happen without a focused effort; simply being demographically neutral when making participant decisions does not help if the pool of applicants lacks diversity, as we discovered in earlier hackathons. To increase this pool, a number of strategies were implements. These include (1) targeting selection of diverse members for the leadership team who will serve as role models for potential participants, as well as potentially having greater contacts or networks with underrepresented subsets of the community; (2) targeting potential applicants through direct invitation; (3) targeting mentors or role models of potential applications who may themselves not wish to participate in the hackathon but who could suggest appropriate individuals to invite and/or encourage them to apply.

Direct invitation to a hackathon serves an important function, not only in increasing diversity, but also often in targeting expertise. However, we found it to be too limiting as a general strategy. Choosing participants, at least in part, from an open pool is extremely important if one of the goals of the hackathon is to grow the community. Much like Gordon conferences, invitation-only hackathons (*e.g.*, the BioHackathons {ref,ref,ref} organized by DBCLS, Japan) have a danger of stratifying individuals into the invited in-crowd and the ignored out-crowd.

As the hackathons progressed, we developed a strategy for increasing the diversity of the applicant pool. Soon after the open call is distributed, each organizer contacts a few qualified women or minority candidates and urges them to apply. Whereas only a tiny fraction of people respond to the open call, roughly half of people (in our experience) apply when they are encouraged to do so personally by an organizer. The applicants recruited in this manner are roughly similar in qualifications as other applicants, and have roughly the same chance of being accepted. The impact of these steps on the diversity of participants is apparent in Fig. X and Fig Y (photo of phylotastic 2).



*Figure X. Trends in diversity across the events. The top graph shows absolute numbers of participants, women, and minorities at each event, while the bottom two figures show the relative proportions of minorities (left) and women (right). In both cases, our deliberate initiatives to increase diversity at the hackathons led to a steady increases in both minority and women.*

### Scoping to Balance Flexibility and Specificity

As already mentioned, controlling the scope of a hackathon is a balancing act between being flexible enough to allow participants the freedom to pursue their own interests and creativity, but focused enough to meet the broad objectives as determined by the LT.

For example, NESCent's Phylotastic II hackathon had a very open scope with the greatest talent diversity. However, it produced fewer tangible products than typical, which may be due to its loose structure. In contrast, Phylotastic I was also open-scoped however with far less talent diversity, but had stronger tangible results. On the other hand, a loosely-defined scope can also lead people to work increasingly on their own rather than in groups, which was seen in the VoCamp hackathon. It can also lead to intangible topics (*e.g.*, “Big Data” or “The Semantic Web”) which might result in a lot of discussion, but little in the way out concrete outcome.

As a result, there may exist a tension between the goals of producing tangible products and increasing talent diversity. We have found that having two disparate goals of producing large amounts of useful code while also focusing on community building is challenging, particularly during application evaluation and facilitation of the actual hackathon.

It is also important to decide on the technologies used. The LT must specify the technological and thematic limits to ensure that conformant projects advance the goals of the sponsors, while giving maximum flexibility to participants to leverage their own unique technology skills. NESCent hackathons typically have technology limitations (or technological focus) as part of the scope. For Comparative Methods in R, the technology limit was to use R, and the thematic limit was the academically defined area of comparative evolutionary methods. For the Database Interoperability Hackathon, we encouraged users to apply a specific interoperability "stack" developed by the EvoInfo working group (NeXML file format, PhyloWS web service schema, and CDAO, the comparative data analysis ontology) to the problem of building links between data resources, or linking resources to users. For the TreeForAll hackathon, the technology limitation was to make use of OpenTree's web services API, and there was no restriction on themes. For Phylotastic I and II the thematic limit was the development of a workflow system to facilitate extraction and use of parts of large trees; we encouraged participants to explore the components of the workflow, the underlying architectural variations to make the workflow feasible, and possible applications of the system.

Our practice is to encourage and enable, but not rigidly enforce, conformity to scope. In practice, projects that do not conform to the scope rarely make it out of the pitching stage. The scope is encouraged by the LT, first by the communicating the scope as part of the advertising, and thereafter by interactions with participants. The opening talks on Day 1 of the hackathon should reiterate the scope and explain what it signifies. When a potentially out-of-scope idea arises, rather than asking a negatively framed question ("Isn't that out of scope?"), a more welcoming approach is simply to ask the proposer to flesh out possible ways in which the idea aligns with the thematic and technical scope of the hackathon ("what are some ways that this project will . . .?"). This type of question should be asked both during pre-event engagement, and during pitching (by member of the LT present at the hackathon).

### Remote participation in hackathons

Though we believe that face-to-face interaction is crucial, we also have explored the use of remote methods. In several cases we have welcomed a number of individuals to participate remotely, and few have done so. In one case, there was a kind of satellite mini-hackathon in which a remote group formed a team of its own.

The remote participation of a single individual is challenging. The individual needs to be included in one of the teams, which requires the extended use of communication technology (e.g., Google Hangouts) for the entire duration of the hackathon and in a continuing manner. The interaction through cameras and microphones is not ideal, and makes it challenging to share documents (e.g., written notes, drawings on a whiteboard) and brainstorm. Further challenges to remote participation (and continuous interaction) are the possible differences in timezones, which makes it hard to schedule stand-ups and breaks, and the unavoidable distractions caused by the daily activities of the remote participants (e.g., receiving phone calls in their home office). One individual who has experienced success in remote participation closes his office door and hangs up a sign indicating that he is not to be disturbed.

Remote participation of a single individual has worked best when the individual serves as the project lead and the project is amenable of being clearly partitioned in loosely related components. For example, in the Phylotastic I hackathon, this was the model that successfully led to the development of the DateLife project.

The remote participation of an entire team was experimented with during the Phylotastic I hackathon. The team was located at a remote site, with several researchers spending an equal amount of time as the on-site participants in a dedicated remote space (a lab reserved for the event). The remote team focused entirely on one of the hackathon projects, with no participation in such projects from the main site participants. The remote team did not require continuous video/audio connection to the main site, with interactions limited to the initial introductions and the regular stand-up meetings. While easier to implement than the individual remote participants, the remote team participation does not benefit from the knowledge exchanges that occur among participants at the main site; the infrequent connections and interactions remove many of the described benefits of the hackathon.

The challenge with remote participation is not just remoteness, but maintaining aligned schedules. This can be practically impossible if remote participants are more than 3 time zones away. Even if remote participants are in the same time zone, it is necessary to ensure that they are aware of schedule modifications and agenda changes. Using generalized VTC technology, and ensuring that important on-site activities are always viewable and audible via the VTC system, is almost never feasible as it would require 5 to 6 VTC rooms when participants are split into teams. We find that a better strategy is to identify an on-site participant who will agree to maintain continuous video contact with a remote participant. The remote participant may then ask questions about what is happening, and receive clarification (via text chat), without disrupting the meeting. If a separate "avatar laptop" is available for each remote participant, the avatar can simply sit at the table with the rest of the team, maintaining constant video and audio contact.

### Using face-to-face time productively

The hackathon event has a limited time window. This face-to-face time should be managed such that it maximizes activities which benefit from direct interaction and minimizes activities which can be performed alone outside of the event. A large portion of the hackathon will necessarily be individuals working, and writing code is necessarily an individualized process for many, but the purpose of the event is to work collaboratively and thus time and effort should be facilitated to take advantage of the group setting whenever possible. The serial nature of out-of-step communications (*e.g.*, email) make efficient discussion difficult; hard discussions and questions should be prioritized for the face-to-face setting where time-lags are minimized and in-person consensus can be reached. This also motivates the introduction of a structure in the activities of each team at the hackathon, where team discussions are frequent and the face-to-face time is positively used to achieve consensus on shared goals and resolve design challenges.

## Going further

This is the section for

* suggestions for how to move forward with research on hackathons
* ideas, speculation, new hypotheses
* other stuff?

### unclassified stuff for "going further"

#### hacking med example

Generally, however, building sustainable products out of hackathons is very difficult.

* Example: Hacking medicine hackathons (http://hackingmedicine.mit.edu/) are large events, specifically geared toward products that could become startups. The focus is on commercially viable products. While the raw number of products is impressive, the success % may not be higher than we have achieved.

#### less preconception

* The fewer products that are pre-conceived, the better they have tended to turn out
* Ex: R hackathon: very little source code was available before the event, all code open-source after event

#### issue trackers and email lists vs. wikis

wikis tend to get ignored or taken over by a few people editing them. email lists keep everyone involved. issue trackers with notification are more specific and customizable ways to keep everyone involved post-hackathon. have we learned that using issue-trackers is better?

### How do we quantify the impact of code?

Computer code has a direct impact when it is used in a production environment. For scientific programming, this means that the code is used to generate or analyze results for publications, presentations, and grant proposals. A similar standard could be applied to artifacts other than code, i.e., data sets and schemata have an impact when they are used in a production environment.

We suspect that most hackathon code is not used in a production environment. However, it may be used by the authors, without anyone else knowing it.

The best prospects for direct code impact is when a hackathon team adds to an existing scientific software package that continues to be used (ape, brownie, NeXML, etc). However, it is difficult to track the importance of individual lines of code generated at a hackathon for 3 reasons. First, use of the package in a production environment may not be documented. Second, even if package use is documented (*e.g.*, by citations), we may not be able to tell how much it depends on specific lines of code. Third, even if we could assign a usage value to each part of the code, it may be difficult to tell which parts (in an evolving body of code) came from a hackathon.

### What makes the NESCent model different?

The NESCent Model for hackathons distinguishes itself in certain salient characteristics from the many other models being practiced. These characteristics are motivated by two foundational objectives: (1) to allow tackling the often difficult problems of making existing tools and resources work together more seamlessly; and (2) to foster sustained, cohesive communities of practice

Include here a quick discussion and reference to a table/figure describing variation of what we see as relevant parameters between the 8 hackathons, *e.g.*, as described in the section at the end of this document 'Variation in hackathon parameters'.

The distinguishing characteristics designed to accomplish these objectives can be reduced to a combination of the following three.

1. **A participant group that balances diversity with homogeneity, and that is approximately 20-30 people in size.** The idea is a group small and homogeneous enough in size and composition that every participant will still interact with everyone else in at least a meaningful social capacity, yet large enough to allow for diversity in a variety of aspects, including perspectives, skills, career stage, field of study, and demographics.
2. **A well-defined scope that everyone buys into by virtue of participating.** A well-defined scope gives functional as well as technological targets, but balances between being open enough to allow for emergent outcomes and being specific enough to define the event by requiring. Defined thus, the scope signifies buy-in from prospective participants to shared goals and desired outcomes.
3. **A duration of 4–5 days.** This has shown to be still short enough to not preclude people from participating who are otherwise well suited for an event i.e. it is demonstrably not too long that it prevents the people we want to attend from showing up, i.e. such people are prepared to accept the absence from home, and removal from their daily work, that is linked to travelling to attend the event. It is also long enough to allow participants to attempt solutions to data and software interoperability issues, which are often challenging yet become tractable with the right set of collaborators in the room. And it is long enough to allow ample room for participant bonding over common obstacles and shared goals, which is key to fostering and sustaining community growth and cohesion.

We have used this model exclusively for hackathons in academia, for bringing together academic research projects, and to foster communities of practice that consist largely of scientists and research staff. The model is potentially well applicable to non-profit and government settings as well, but in the industry realm in particular the duration will prevent most people from participating.

The following additional characteristics have also emerged as common across NESCent Model-hackathons, but they can be found frequently in other types of hackathon events as well.

**A 'destination' event.** With one exception, NESCent hackathons have been stand-alone events not tied to a conference or other event. Each hackathon has a minority of local participants, but the majority travel to the site of the hackathon to be present at the hackathon, and for no other reason.

**Leadership team for collaborative well-informed decision making.** There are many points in the planning phase of a NESCent Model-hackathon at which decisions have to be made that co-determine the direction and outcomes of the event, but for which right or wrong answers do not necessarily exist. To ensure that these decisions are made in a well-informed and deliberate manner, a leadership team rather than a single individual is responsible for them. The leadership team is put in charge early on the process, and is assembled from people that are invested in the event's outcomes and that between them represent a wide umbrella of perspectives.

**Structured first day, unstructured rest.** Most of the first day of hackathons in this model is taken up by structured activities that start with welcoming everyone and end with participants breaking up into teams with self-chosen target projects. The rest of the time is largely unstructured, with teams determining their schedule on their own. The whole event concludes with show-and-tell of each team, followed by identifying actions the sponsor can take, or resources the sponsor could help establish, to aid post-hackathon completion and, ideally, sustainability of nascent products.

**Daily succinct and brief report-outs ("standups").** To ensure that related or dependent pieces of work across teams remain coordinated, each team briefly reports its status, plans for continuation, and current roadblocks once on each day after the first. This format is borrowed from the daily "standups" in agile software development methodologies. Asking participants to include in these standups a description of current roadblocks is intended to allow tapping into the collective expertise in the room to overcome these.

**All products are public and open-source, immediately.** This normally implies that all software tools or resources targeted by work at the hackathon must be open-source projects. While code produced at the event may be merged into a larger target codebase only with a delay, or after further modification, there is no embargo on any of the code produced at the hackathon.

**No competition.** There are no prizes awarded at the end of the hackathon, and there is no judging panel. Supporting additional participants has remained a higher priority for using the sponsors' funding than awarding a prize. The removal of a competition aspect is also motivated by the desire of creating a spirit of collaboration and enabling the inclusion of participants with different skill levels.

### Career stage of participants

An observation which comes from working groups at some of the national synthesis centers (NCEAS, NESCent, NIMBios) is that one driver of success of many working groups is having a good mix of junior and senior participants {ref?}. While senior participants are essential for their experience and knowledge of problems and systems, their schedules are often busier post event and following up on a particular outcome for the event has less impact on their career. In contrast, a junior participant (graduate student, postdoc, untenured professor) may be much more driven to follow up and drive production after an event because the tangible product may have a large impact on progression to their next career stage. Hackathons are likely to follow this same pattern, with an even greater complication: research staff. Many of the hackathons we have conducted have included large numbers of research staff, often chosen for their programming/domain expertise. While invaluable during the hackathon, these participants may be the least likely to see a direct career benefit from continued production of a tangible product after the hackathon (the personal benefits to them may follow the intangible benefits) and, more importantly, they often lack the job flexibility in time and effort that someone in a more traditional academic position can devote to following up on details.

Thus to achieve tangible working products, one should consider whether (1) deliverables can be achieved in the time slotted for the hackathon, or (2) whether choosing participants who may lack certain skills or experience may serve the overall goal better by their likelihood of being able to dedicate extra time after the event to follow-up and complete the project.

Another critical aspect in the achievement of tangible results and sustainable continuation of the projects is the ability to coalesce a community around the activities performed at that hackathon. The development of open data repositories and the contribution to widely accessed code repositories are aspects that facilitate the community "buy-in" and enable the long-term sustainability of the hackathon products. An example of this is represented by the contributions to the NeXML code base achieved during the Database Interoperability hackathon.

### Data we wish we had collected

* Ask participants who they know beforehand (and who they've collaborated with beforehand), then who they know afterward = measurement of connection and assess new collaborations
* Publications
  + if they have a complete publication list, ask them for that (or maybe insist they give it to us)
  + if they have a curated ORCID account, ask them for that - or maybe insist they create one before coming
* Citations
* Social media participation:
  + Determine a unique HashTag before the meeting that can be used to trace event-specific Twitter interaction
  + collect their Twitter handles, LinkedIn URLs, Facebook things (whatever they are, I'm not on Facebook), github handles
* Query participants on familiarity with key technologies of interest prior to hackathon (particularly those which form the core of the project or which are introduced in bootcamps). Query them afterwards to see how many use these technologies after the event has completed.
* Keep track of who does what when. Which individuals make the pitches? Which pitches are chosen to move forward?
* Feedback forms from all paritcipants after each event, with the same set of questions asked in all of them (or at least that all forms contained the same set of core questions), allowing us to track subjective measures of success of the hackathons direct from participants e.g. by answers to the question "how useful was this event for you for building your professional network, on a scale of 1-5" etc.

# Conclusions

A key conclusion is that there is a particular set of data that we encourage people to collect associated with Hackathons to enable them to much more easily be assessed for impact. Using basic analyses of some of this kind of data, we've been able to demonstrate xxxx.

# Possible data figures/tables

## Diversity variation between meetings

Motivation of including this figure is as a way of visualising impact in terms of various different kinds of diversity, while also giving some information of the extent to which overall changes in diversity were reflected in diversity of arguably more 'influential' roles (leadership team, pitchers, accepted pitchers).

However, I would be interested in plotting on the same graph the value for several different categories of people. e.g. when plotting gender diversity, we could look at it in:

- applicants

- participants

- leadership team

- pitchers

- accepted pitches

Other diversity measures could be:

- proportion of minorities

- proportion of people holding an academic position at the time of the hackathon

- proportion of senior career-stagers (junior professor and up...?)

We could also look to see the extent of correlation between some of these diversity measures, and other measures of different kinds of impact (development of repos that continue to be worked on, for example).

## Variation in hackathon parameters

Motivation is just to provide a graphical impressions of variation (I expect, but am not sure, it will rather be a story of relative stability) of various features of the hackathon series.

Suggestion is to do this using a chart plotting:

- participant numbers

- applicant numbers

- remote participant numbers

- accepted pitch numbers

- cost/budget

against hackathons ordered by time on the x axis.

## Sustainability of technology adoption

Motivation is to assess the impact of the hackathons in terms of encouraging participants both to try out for the first time, and to adopt for use in later work/projects, "important" tools, for which I suggest we focus on github.

To do this, I suggest we collect via the github API a set of all commits for all particiapants witha github handle. If possible, this would be the total list of all commits these people have ever made on all public github repos (some people will have many 1000s, others 0 or 1), recording the date of the commit, the repo name, whether or not they are owners of the repo, and the date of creation of their github account.

With this, we could ask the following questions:

- for the hackathons where all development was done using github, how many participants made their first commit, how many created their github account, within e.g. the event itself and maybe 2 weeks before it; this is a count of the people who were actively introducted to this new technology as a result of participating in the hackathon

- how many of the people who started using github as a result of the hackathon are still using it 6 months, and 12 months after the event (i.e. who have incorporated it as part of their working life)

We could plot these on a chart with x axis the github-focused hackathons in order of occurance, and for each of them the numbers described above on the y axis (number of people using github for the first time in that meeting, number of those still using it after 6 months, still using it after 12 months)

## Sustainability of projects initiated during hackathons

Motivation is to come up with some kind of measure of the extent to which projects initiated during the hackathon have continued to be developed, and used, and of interest to the wider community, after the hackathon ends. I'm suggesting we do that focused on github given the (I hope) relative ease of interacting with it to collect this kind of information programatically. I suggest we only focus on those projects that are initiated specifically for the hackathon, rather than those that incorporate code into existing repos, as it is then easier to trace actigvity specifically to the hackathon.

For each github repo created new for a project in the hackathon, we could collect the date of each commit since the repo was created. We could plot the distribution of these commits through time, lining up the charts like this from all such projects on the 5 days of the hackathon, then we see how this varies between the different hackathons, and we see easily which ones have been further developed, which not.

We could also show forks, maybe in a differnet colour.

## Impact of hackathons on building community and collaborations

Motivation is to check to see the extent to which participating in hackathons develops longer-term interaction and collaboration amongst people within the hackathon (perhaps also within the same team?)

We would like to collect, for this, an as-complete-as-possible list of Twitter and github handles for all participants of all hackathons.

We could, for each hackathon (perhaps for each team also) count the number of twitter interactions (RTs, favourites, mentions) each person in that team made via Twitter before, during, and after the hackathon. We could break "after" down into 2 weeks after hackathon, and then 6 months and a year afterwards, to see if/whether itneractions were sustatined over a long time.

We could do the same thing with github handles, looking for the occurance of two people who met at a hackathon making commits to the same repos as others form the same one, both before, during,a nd after, the hackathons.

<!-- also, people can become each other's followers on github, so you could look at social graph density before and after. RAV -->

Include here also something based on the data Jonathon used in his description of network/community impact on the basis of detailed quesionarires returned by all participants from one of the hackathons.

# Resources

Ten Simple Rules to Achieve Conference Speaker Gender Balance http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003903

Work Groups, Structural Diversity, and Knowledge Sharing in a Global Organization, Jonathon N. Cummings, Management Science, Vol. 50, No. 3 (Mar., 2004), pp. 352-364.

StitchFest: Diversifying a College Hackathon to Broaden Participation and Perceptions in Computing. Gabriela T. Richard, Yasmin B. Kafai, Barrie M. Adleberg, Orkan Telhan. Proceedings of the 46th ACM Technical Symposium on Computer Science EducationProceedings of the 46th ACM Technical Symposium on Computer Science Education, 2015.

Ones directly based on NESCent hackathons:

[Phylotastic! Making tree-of-life knowledge accessible, reusable and convenient.](http://www.biomedcentral.com/1471-2105/14/158) BMC Bioinformatics. 2013 May 13;14:158. doi: 10.1186/1471-2105-14-158.

Stoltzfus A, Lapp H, Matasci N, Deus H, Sidlauskas B, Zmasek CM, Vaidya G, Pontelli E, Cranston K, Vos R, Webb CO, Harmon LJ, Pirrung M, O'Meara B, Pennell MW, Mirarab S, Rosenberg MS, Balhoff JP, Bik HM, Heath TA, Midford PE, Brown JW, McTavish EJ, Sukumaran J, Westneat M, Alfaro ME, Steele A, Jordan G.

**REFERENCES WE NEED TO SOMEHOW INCLUDE**

Community-driven development for computational biology at Sprints, Hackathons and Codefests. Steffen Möller, Enis Afgan, Michael Banck, Raoul J. P. Bonnal, Timothy Booth, John Chilton, Peter J. A. Cock, Markus Gumbel, Nomi L. Harris, Richard C. G. Holland, Matús Kalas, László Kaján, Eri Kibukawa, David R. Powell, Pjotr Prins, Jacqueline Quinn, Olivier Sallou, Francesco Strozzi, Torsten Seemann, Clare Sloggett, Stian Soiland-Reyes, William Spooner, Sascha Steinbiss, Andreas Tille, Anthony J. Travis, Roman Guimera, Toshiaki Katayama, Brad A. Chapman. BMC Bioinformatics 15(S-14): S7 (2014).

**PAPERS describing individual events**

[Meeting report from the first meetings of the Computational Modeling in Biology Network (COMBINE).](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3235518/) Stand Genomic Sci. 2011 Nov 30;5(2):230-42. doi: 10.4056/sigs.2034671. Le Novère N, Hucka M, Anwar N, Bader GD, Demir E, Moodie S, Sorokin A.

This article gives a more detailed description of the relevant meetings than in "Meeting Report: Hackathon-Workshop on Darwin Core and MIxS Standards Alignment", and also follows up some more reflective sections on

- "What could have been done differently?"

- "What worked well and should be repeated?"

[Crowdsourcing knowledge discovery and innovations in medicine.](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4180345/) J Med Internet Res. 2014 Sep 19;16(9):e216. doi: 10.2196/jmir.3761.

Celi LA, Ippolito A, Montgomery RA, Moses C, Stone DJ.

As the title suggests, it's a discussion of application of ways of opening up discovery in medicine outside of clinicians. The authors have been involved in organising several hackathons, as described in the quote below

> The authors of this article have helped organize numerous hackathons and data marathons that have brought together engineers, data scientists, and clinicians (including nurses, pharmacists, and other allied health personnel) to address problems and questions identified during routine clinical practice, including the Critical Data Marathon held at the Massachusetts Institute of Technology (MIT) in January 2014 (see Multimedia Appendix 1). To date, the MIT Hacking Medicine has organized 17 events in the United States, India, Uganda, and Spain

There then follow some anecdotal examples of successes/benefits of such hackathons, but no detailed description of what makes them work particularly well

[Less noise, more hacking: how to deploy principles from MIT's hacking medicine to accelerate health care.](http://www.mghcgh.org/assets/files/news/Less\_Noise\_More\_Hacking.\_IJTAHC.2014Olson.pdf)

Int J Technol Assess Health Care. 2014 Jul;30(3):260-4. doi: 10.1017/S0266462314000324. Epub 2014 Aug 6. DePasse JW, Carroll R, Ippolito A, Yost A, Santorino D, Chu Z, Olson KR.

A more detailed description of the format and results of the hackathons described in the "Crowdsourcing knowledge discovery and innovations in medicine" article mentioned above.

Meeting Report: Hackathon-Workshop on Darwin Core and MIxS Standards Alignment. Éamonn Ó Tuama, John Deck, Gabriel Dröge, Markus Döring, Dawn Field, Renzo Kottmann, Juncai Ma, Hiroshi Mori, Norman Morrison, Peter Sterk, Hideaki Sugawara, John Wieczorek, Linhuan Wu, Pelin Yilmaz. Standards in Genomic Sciences (2012) 7:166-170.

Enriched biodiversity data as a resource and service. Rutger Aldo Vos, Jordan Valkov Biserkov, Bachir Balech, Niall Beard, Matthew Blissett, Christian Brenninkmeijer, Tom van Dooren, David Eades, George Gosline, Quentin John Groom, Thomas D. Hamann, Hannes Hettling, Robert Hoehndorf, Ayco Holleman, Peter Hovenkamp, Patricia Kelbert, David King, Don Kirkup, Youri Lammers, Thibaut DeMeulemeester, Daniel Mietchen, Jeremy A. Miller, Ross Mounce, Nicola Nicolson, Rod Page, Aleksandra Pawlik, Serrano Pereira, Lyubomir Penev, Kevin Richards, Guido Sautter, David Peter Shorthouse, Marko Tähtinen, Claus Weiland, Alan R. Williams, Soraya Sierra. Biodiversity Data Journal 2: e1125 (2014).

OCData Hackathon @ CSCW 2014: online communities data hackathon. Sean P. Goggins, Andrea Wiggins, Susan Winter, Brian Butler. Computer Supported Cooperative Work, CSCW '14, ACM Press (2014).

TUIO Hackathon. Martin Kaltenbrunner, Florian Echtler. Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces (2014).

Industrial Experiences of Organizing a Hackathon to Assess a Device-centric Cloud Ecosystem. Mikko Raatikainen, Marko Komssi, Vittorio dal Bianco, Klas Kindstom, Janne Järvinen. 37th Annual IEEE Computer Software and Applications Conference (2013).

BioHackathon series in 2011 and 2012: penetration of ontology and linked data in life science domains. Toshiaki Katayama, Mark D Wilkinson, Kiyoko F Aoki-Kinoshita, Shuichi Kawashima, Yasunori Yamamoto, Atsuko Yamaguchi, Shinobu Okamoto, Shin Kawano, Jin-Dong Kim, Yue Wang, Hongyan Wu, Yoshinobu Kano, Hiromasa Ono, Hidemasa Bono, Simon Kocbek, Jan Aerts, Yukie Akune, Erick Antezana, Kazuharu Arakawa, Bruno Aranda, Joachim Baran, Jerven Bolleman, Raoul JP Bonnal, Pier Luigi Buttigieg, Matthew P Campbell, Yi-an Chen, Hirokazu Chiba, Peter JA Cock, K Bretonnel Cohen, Alexandru Constantin, Geraint Duck, Michel Dumontier, Takatomo Fujisawa, Toyofumi Fujiwara, Naohisa Goto, Robert Hoehndorf, Yoshinobu Igarashi, Hidetoshi Itaya, Maori Ito, Wataru Iwasaki, Matúš Kalaš, Takeo Katoda, Taehong Kim, Anna Kokubu, Yusuke Komiyama, Masaaki Kotera, Camille Laibe, Hilmar Lapp, Thomas Lütteke, M Scott Marshall, Takaaki Mori, Hiroshi Mori, Mizuki Morita, Katsuhiko Murakami, Mitsuteru Nakao, Hisashi Narimatsu, Hiroyo Nishide, Yosuke Nishimura, Johan Nystrom-Persson, Soichi Ogishima, Yasunobu Okamura, Shujiro Okuda, Kazuki Oshita, Nicki H Packer, Pjotr Prins, Rene Ranzinger, Philippe Rocca-Serra, Susanna Sansone, Hiromichi Sawaki, Sung-Ho Shin, Andrea Splendiani, Francesco Strozzi, Shu Tadaka, Philip Toukach, Ikuo Uchiyama, Masahito Umezaki, Rutger Vos, Patricia L Whetzel, Issaku Yamada, Chisato Yamasaki, Riu Yamashita, William S York, Christian M Zmasek, Shoko Kawamoto and Toshihisa Takag. Journal of Biomedical Semantics 2014, 5:5

The DBCLS BioHackathon: standardization and interoperability for bioinformatics web services and workflows.

T Katayama, K Arakawa, M Nakao, K Ono… - Journal of biomedical semantics, 2010 [doi:10.1186/2041-1480-1-8]

The 2nd DBCLS BioHackathon: interoperable bioinformatics Web services for integrated applications

T Katayama, MD Wilkinson, R Vos, T Kawashima… - Journal of biomedical semantics, 2011 [doi:10.1186/2041-1480-2-4]

The 3rd DBCLS BioHackathon: improving life science data integration with Semantic Web technologies

T Katayama, MD Wilkinson, G Micklem, S Kawashima… - Journal of biomedical semantics, 2013 [doi:10.1186/2041-1480-4-6]

\*Communities of pratice\*

Smith MK (2003) Jean Lave, Etienne Wenger and communities of practice. Encyclopedia of Informal Education Available: http://infed.org/mobi/jean-lave- etienne-wenger-and-communities-of-practice/ via the Internet.

Wenger E (2009) Communities of Practice: A Brief Introduction. Available: http://wenger-trayner.com/theory/ via the Internet.

Lave J, Wenger E (1991) Situated Learning: Legitimate Peripheral Participation. Cambridge, UK: CUP. 138 p.

\*Communities in general\*

Wenger E, Trayner B, de Laat M (2011) Promoting and assessing value creation in communities and networks: a conceptual framework. Ruud de Moor Centrum, Open University of the Netherlands.

Bacon J (2012) The Art of Community: Building the new age of participation (theory in practice). O'Reilly Media. 527 p.

Budd A, Corpas M, Brazas MD, Fuller JC, Goecks J, Mulder NJ, et al. (2015) A Quick Guide for Building a Successful Bioinformatics Community. PLoS Comput Biol 11(2): e1003972. doi:10.1371/ journal.pcbi.1003972

\*Measuring impact of and growth of communities\*

Millington R (2012) Buzzing communities.