



Intercross: a Breeding Application for High-throughput Phenotyping

Chaney Courtney, Mitchell Neilsen

Department of Computer Science
Kansas State University
Manhattan, KS, USA

Abstract

Plant scientists around the world have expert-specified systems for dealing with sample breeding. The ability to track parents and their future progeny should be an intuitive and easy task for these systems. Surprisingly, breeders have a handful of various techniques that have little to no centralized protocol. This paper focuses on a novel implementation by Kansas State University's Cyber Physical Systems lab, the Android application, Intercross. Intercross is the newest addition to the PhenoApps organization's set of open source applications. This new application is a generic cross tracking system; however, because there are various methodologies for breeding this task is not trivial. This paper will expand on the non-trivial nature of crossing samples and why this system is needed. Previous systems that were used to track crosses are either outdated, do not exist, or are inefficient.

Keywords: Android, phenotypes, high-throughput, progeny, Cartesian product, threshing

1 Introduction

Classical approaches to systematic breeding and phenotype data acquisition have started to depopularize with the introduction of high-throughput mobile phenotyping applications [1, 2, 3, 4]. Intuitively, the use of mobile devices has been an evolutionary step in breeding and various phenotyping applications. The relatively cheap cost of Android devices and the ability to host software for free on the PlayStore creates a high availability for these mobile phones and software for users across the world. The ability to have a secure, handheld database opens the field to many different applications that otherwise could not have been done by pen and paper, or even stationary desktop computers. Intercross is one of such applications that has been developed by the PhenoApps organization. PhenoApps is aimed to increase overall productivity for breeders and plant scientists around the world by creating these various tools. This paper will describe the technical specifications, algorithms, and ideology of Intercross which was built by and for breeders. With the combined effort of Cornell University's Plant Science Department, the Boyce Thompson Institute, Kansas State University's Cyber Physical Systems Lab, and the Centro Internacional de la Papa (CIP) research center, Intercross was developed to capture the everyday workflow of breeders.

Intercross is essentially a genealogy tracker, where users can name the parents of a generated cross. Take this example where a breeder has two types of wheat A and B. The user will input the names of the parents and then the name of the cross, this relation will then be submitted into a local database for tracking. As the process continues, users can walk through the various parents and children to view their history and offspring. When the user inputs A for female, B for male, C for cross and clicks the submit button, this entry is submitted into the database. There are different conventions for naming parents and crosses that will be discussed throughout this paper, some of which utilize barcode scanning or automatic id generation. Breeding is a specified task for each crop; therefore, some features of Intercross may not be used by all breeders. Breed tracking is non-trivial as different plants have different requirements; a simple example being the need to track females before males, or the idea that bananas are self-fruitful. Features that will be discussed within this paper are cross block management, pollen management, automatic identifier generation, defining ZPL labels, breeding event tracking, and finally wish list tracking.

2 Background

The PhenoApps organization solely develops for the Android market. The Android Play Store is a public and free service that anyone can use with an Android phone. Android's open source nature, developer friendly ecosystem, and cheap devices, create a suitable development environment for targeting researchers throughout the world. Intercross uses the state-of-the-art Google Jetpack libraries for its architecture, utilizing LiveData and the Room persistence database [9]. Intercross uses four main tables in its database to track all information. The tables include: a main cross table that has male, female parent data and new cross event identifiers in a single row, a settings table that persists user settings and ID patterns, a parent table which simply stores parents that have been imported, and finally a wishlist table that tracks events that should be completed s.a crossing, threshing and harvesting. The jetpack architecture allows developers to define models, views and fragments that essentially reflect an MVC architecture. One difference being the ability to define shared view models, which is a way to share data between fragments of an application. For example, the barcode scanner is a separate fragment than the main fragment which displays crossing events. When the barcode scanner fragment is showing and the user scans an ID, a shared view model is used to save the last scanned ID. When the user transitions back to the events crossing page, the shared data from the barcode fragment can be used.

Because the act of creating a cross is non-trivial, Intercross has various ways to accept data. Intercross has three main categories that define a suitable cross that correspond to a given pollination type. When a male and female is present, the cross is bi-parental. If there is only one parent, the cross is considered self-pollinated; otherwise, the cross is open-pollinated. Other than the three main pollination types, the act of making crosses can vary between research facilities. Some researchers require that female parents of a cross be recorded before males or have other workflow specific requirements.

2.1 Workflow

As mentioned in the Introduction, there is no generic workflow for breeders around the world. For this reason, Intercross has settings and different options for attuning to various workflows. One of the more basic settings is the ability to choose which is named first, the male or female. The reasoning for this is that some breeders only track individual female plants and not

necessarily the male plant. In these situations the male seed comes from a bulk of different male plants, and the breeder cares more about the female being tracked. A certain feature in Intercross, Pollen manager, can create a hash array list of male id keys that relate to a set of male ids. This feature will be later discussed in the pollen manager section and allows the ability to associate a set of ids to one id. Some breeders have no use of male parents at all. For these breeders, there is a setting to disable male id, which will be saved as 'blank' in their database. Breeders do have a specification for different types of parental pairs. These classes are open-pollinated, bi-parental, and self-pollinated. These pollination types define how the plant was bred, for example a plant that requires a male and a female is labeled as bi-parental. Open-pollinated plants have a female but no unique male id. Finally, self-pollinated is a plant where the male and female ids are equal.

Users of Intercross can easily visualize the pollination type of their crosses within the app. Other than specific breeder tasks, there will always be parental data entry and cross naming throughout the workflow. A daily-first-step for breeders is to check whats called a cross block manager, which is a matrix populated by the Cartesian product of the sets of males and females. A certain coordinate in the cross block represents the physical location of the plants to be crossed in a greenhouse. Classically, breeders have used paper logs or excel spreadsheets to track how many crosses have been made of certain parental combinations. Intercross has developed a simple and intuitive visualization for managing cross blocks which will be discussed in a further section. During the crossing workflow, users have the options of manually inputting data or scanning via the mobile phones camera. Typically, parental barcodes must be printed before any crosses can be made, that way their barcodes can be printed in bulk and scanned when they are to be crossed. Barcode printing and scanning will be covered in the later sections. Once breeders have scanned their daily crosses, they can export a cross block summary which is a checkpoint for the days work. Users can then import this checkpoint to visualize their cross block summary and continue breeding for that next day. A similar, daily-management feature of Intercross is the wish list manager. This manager can delegate work by creating a list of males and females that should be crossed. Users that oversee breeding processes can create multiple wish lists to parallelize work for their breeders. This way each breeder has a guided day of work, meaning they will know exactly which plants must be crossed for that day.

3 Barcoding

Another unique and useful feature of the mobile pocket PC devices is the integrated barcode scanner and ability to print barcodes via bluetooth to a connected printer. Barcode scanning is essential for breeders to have an efficient workflow; otherwise, work days are inundated with manual data entry. Classically, parents would have a certain name or ID written in a field book which would have to be manually scanned by reading the text and comparing it to a cross block list. More recent technological advancements with barcode scanning allows users to simply print labels for their parents, and when it comes time to breed them the researchers can scan the printed labels for entry into their application. Android devices cannot only connect to bluetooth or USB bar code scanners and printers, but with modern software libraries can scan barcodes by solely using the embedded camera on the front or back of the phone. The ability to read barcodes simply by using a phone's camera opens these research capabilities to everyone with an Android phone, and simplifies the process of buying and connecting a scanning device. The scanning capability of Android phones has proved efficient and useful in other PhenoApps applications [6]. Android phones have similar functionality when it comes to connecting to a printer. Previous pocket PC, software, CIPCross, does not give the user any power to change

or create custom labels. Intercross uses the Zebra SDK [7] which allows users to import their own .ZPL (Zebra Printing Language) for specifying a label format. Intercross has a unique, continuous-crossing mode that allows breeder workers to easily and quickly scan data from barcodes utilizing the following algorithm:

```

val callback = object : BarcodeCallback {
    override fun barcodeResult(result: String) {
        when (mode) {
            "single" -> {
                lastScan = result
                popBackStack()
            }
            "search" -> navigate(result)
            "continuous" -> {
                lastScan = result
                handler.postDelayed({
                    barcodeScanner.decodeSingle(callback)
                }, 2000)
            }
        }
    }
}
barcodeScanner.decodeSingle(callback)

```

The above algorithm describes the three different scanning modes available in Intercross. First of all the single scanning mode simply uses the shared view model to send data back to the events fragment. The method 'popBackStack()' is an android navigation function that returns the user to the previous fragment (saved within a stack). The search mode will simply navigate to a fragment which contains more information about that resulting ID. Finally, the continuous fragment will loop through scans by saving the last scanned ID, and calling the scanner to decode again. Because this fragment uses a shared view model with the main events fragment, the continuous mode can submit data to the database as it scans data. In the main fragment, as scans are accepted, it will check if the input is valid with the settings and default requirements and submit the cross to the database.

4 Identifiers

Breeders around the world have no generalized pattern for crop naming conventions. Although it is not uncommon for groups to create internal patterns; therefore, Intercross has an accumulation of different naming functions. This section will describe two of the naming functions in detail. Intercross by default has the ability to manually name samples with any Unicode characters.

4.1 Universally Unique Identifiers

Intercross has the option to name crosses by automatically generating a universally unique identifier. The code to generate this type of string is in the segment below.

```
mCross.text = UUID.randomUUID().toString()
```

In the code above, the view for user-input cross IDs is being set by generating a random UUID. When breeders don't need a certain naming convention for their crosses, this is a good replacement which will speed up their workflow.

4.2 Pattern Creation

Another naming convention option is the ability to create patterns. Intercross supports a subset of regular expressions, where the user chooses a prefix, and suffix string along with a middle number with optional padding. Pattern creation is a setting within Intercross that will dynamically visualize the pattern as the user inputs these values. For example if the users were to choose their prefix as "Inter", their suffix as "Cross", their middle number as 108, and their padding as three, the generated pattern would be "Inter108Cross". If the padding was set to four, it would be "Inter0108Cross". As users generate crosses the middle number is used as the unique identifier and is incremented, similar to automatic int content values in SQL databases. Continuing the example, once two crosses are made there will be two entries in the database, F1, M1, "Inter108Cross", and F2, M2, "Inter109Cross". While the user interface will already have a populated cross entry which will be "Inter110Cross".

```
mCross.text = "$prefix${num.toString().padStart(pad, '0')} $suffix"
```

Pattern creation is an optional setting within Intercross. If this setting is true, whether the user is using a barcode scanner or manual input, there will only be two required scans (unless blank male is set as well) one for the male and one for the female.

5 Cross Block Manager

With a greenhouse full of plants to be crossed, users need an easy way to manage which samples should be crossed in a given day. A simple, tournament style, matrix is created where the rows are females and the columns are males. The user provides Intercross with a comma separated sequence of male and female names. Intercross then creates a Cartesian plot of the names and accesses the local database. To fill the matrix, the database will query which males and females have been crossed already, and how many times. Each cell in the matrix will be color-coded which is dependent on a given max-cross value. If a male and female have passed the max-cross value then the cell will be colored in red, otherwise the cell will be blank. This creates a visualization for the breeders to easily tell which samples should be crossed for that day.

6 Wish List

Similar to the cross block manager, the wish list import and management allows a 'manager' to create workloads for other devices. The wish list simply imports a text file full of rows that represent crosses. When the wish list is imported, a table is populated to represent how many of each parental combination should be crossed for that list. The wish list management allows managers to create a definitive workload for its users while still being able to visualize the project as a whole with the cross block manager. As crosses are made, the wish list is updated and if the requirements are met the user is prompted with an audio notification to let them know their work is complete.

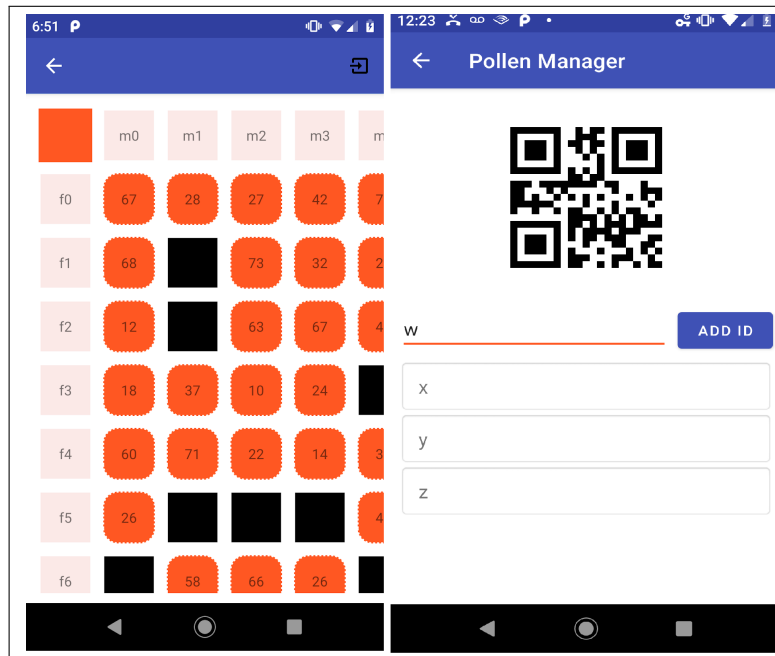


Figure 1: Cross block visualization on the left and pollen manager on the right within Intercross.

7 Pollen Manager

Most breeders will always have a male and female when making crosses. In some scenarios the information of the male is not as important as the female. For this situation, breeders typically create a bag of bulk-males. When it comes time to breed the females, a random male is chosen from the bulk and is mated with the female. For this reason, breeders need a way to print a label for a set of males, which would be attached to the bulk-bag. The breeders would then simply scan the female barcode on the female flower, and then scan the barcode on the bulk male bag. The pollen manager activity of Intercross creates these bulk barcodes by allowing the user to enter an array of male ids. The barcode is dynamically generated and visualized as the user adds IDs. These male IDs are tracked in a wish list table to preserve which males are bulked together.

8 Related Work

Intercross has been in beta since July 2018 when it was developed during a hackathon at Cornell University, #phenoAppsHackathon [5]. At that time, developers were able to interface with the needs of plant scientists and develop a first pass application for their workflow. Features have been constantly worked on and reviewed at a second hackathon that was hosted by CIP in Lima, Peru. Previously, the researchers at CIP used handheld Motorola Pocket PC devices. Each device is about \$1500, more than three times the amount of a standard Android phone. Developers at CIP created the software, CIPCross, to be used on these pocket PCs, which Intercross aims at replacing. The figure below shows the current state of CIPCross on its main crossing page. Researchers at CIP noted that using the resistive touch screens of the

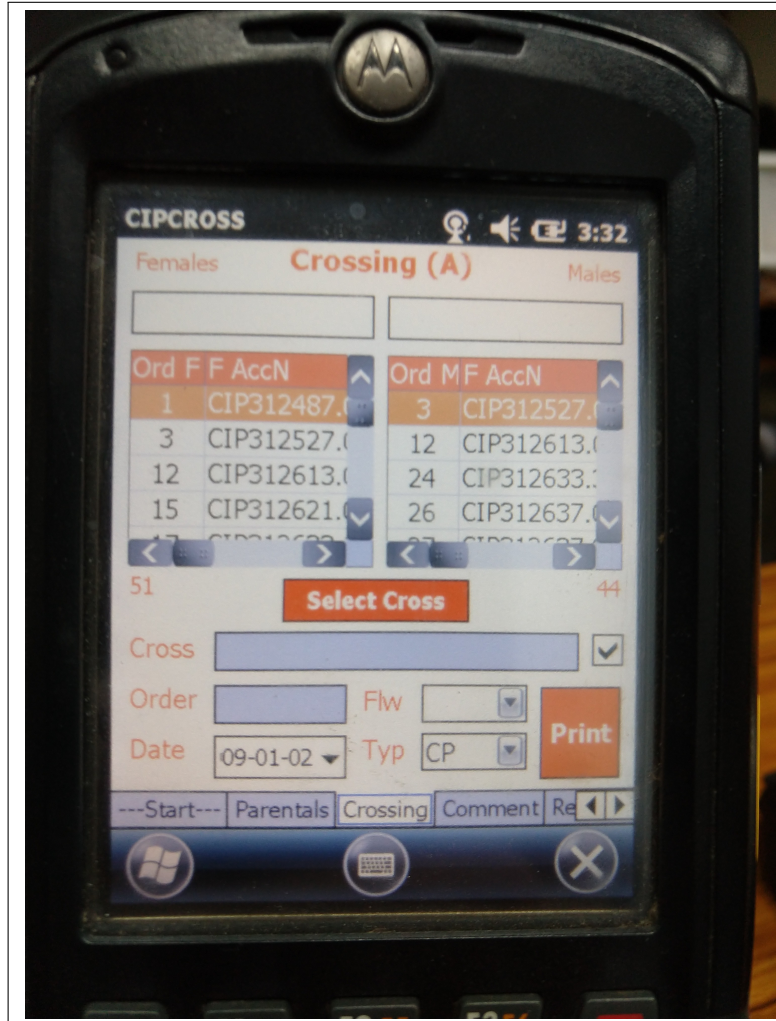


Figure 2: A picture of CIPCross

Motorola devices takes longer than modern phone screens. These Windows PC's are no longer supported by Microsoft and were once preferred by these researchers for their rugged design for field work and internal barcode scanner. Therefore, the combination of CIPCross being run on outdated devices, and having less features than Intercross submits that Intercross is a welcome and much-needed replacement at these facilities. As far as the authors' knowledge there are no other mobile applications for cross tracking on the Android market. Developing Intercross for Android gives researchers the opportunity to reduce cost of equipment, while also receiving complementary software support as this project is completely open source. All of Intercross' code is hosted on Github, which allows users to post questions, along with pull requests, which are features they have implemented and want to be added to the code. Developers can accept pull requests or suggest contributors to fork the project if the pull request is too specific. A mantra of Intercross development is to not over generalize the workflow for one certain crop.

9 Conclusion

With mobile devices becoming increasingly affordable there is an opportunity to transfer workloads from antiquated devices. PhenoApps aims at making breeding a simple task by introducing the app, Intercross. The features and ideologies discussed within this paper describe the every day tasks and problems that breeders are faced with. Intercross is currently in beta and is the only Android application on the market that handles these tasks in cross tracking. Future developments include the improvement of progeny visualization. Being the first of its kind, Intercross is setting the stage for ease-of-use cross tracking. As Intercross gains more users and traction, more features and abilities will be implemented that cohere to daily breeder tasks. PhenoApps' overarching goal is to help breeders and plant scientists around the world build tools to replace hefty-workloads and older methodologies with new technology. Intercross is one exceeding example that fills that gap of breeders trying to cross samples in the field and in the lab. By replacing older technologies and opening up the breeding environment to Android devices, Intercross is setting a new standard for high-throughput data acquisition.

10 Acknowledgements

This material is based upon work supported by the National Science Foundation (NSF) under NSF-BREAD Grant No. 1543958. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

- [1] PhenoApps Organization Webpage <http://phenoapps.org/>
- [2] T.W. Rife and J. A. Poland, "*Field Book: An open-source application for field data collection on Android*". In *Crop Science* 54, 1624-1627, 2014. DOI: 10.2135/cropsci2013.08.0579.
- [3] M.L. Neilsen, S.D. Gangadhara, T. Rife, "*Extending watershed segmentation algorithms for high throughput phenotyping*". in *Proceedings of the 29th International Conference on Computer Applications in Industry and Engineering*, Denver, CO, Sept. 26-28, 2016.
- [4] Chaney Courtney (2017) *Open source application development for phenotypical data acquisition*, Kansas State University, Manhattan, KS, USA
- [5] PhenoApps Github Hackathon Repository <https://github.com/trife/Hackathon>
- [6] Verify Github Repository <https://github.com/PhenoApps/Verify>
- [7] Zebra Android SDK <https://www.zebra.com/us/en/products/software/scanning-systems/scanner-drivers-and-utilities/scanner-sdk-for-android.html>
- [8] Android Developers Documentation "*Documentation website*" <https://developer.android.com/>
- [9] Android Jetpack "*Jetpack website*" <https://developer.android.com/jetpack>