Preventive medicine for epidemic outbreaks and risky behavior through mobile devices and ubiquitous computing

Felipe Besoain-Pino*1, Antoni Pérez-Navarro*2

1 Universitat Oberta de Catalunya, Rambla Poblenou 156, Barcelona, Spain
2 Estudis d’Informàtica, Multimèdia i Telecomunicació, Universitat Oberta de Catalunya, Rambla Poblenou 156, Barcelona, Spain

Email: Felipe Besoain-Pino* - fbesoain@uoc.edu; Antoni Pérez-Navarro* - aperezn@uoc.edu;

*Corresponding author

Abstract

Background: Advances in the development of information and communication technologies, as well as the increased use of social networks, have brought about a high rate of social interrelationships among different social strata. As a result, software has been developed to facilitate interrelations among people. Some of these applications facilitate contacts based on proximity. However, in spite of the software’s social benefits, some of the behavior that is associated with these applications could be considered as risky. How can we prevent this type of behavior without infringing on the character of these applications?

Methods: To respond to this question, we have followed the design and creation methodology, with an emphasis on ubiquitous computing. Software has been developed for the Android OS that communicates with a server to be able to get different viral alerts and/or medical centers. This server has a database with points of interest, which is distributed through a web service, appropriate to the principles of Representational State Transfer (REST).

Results: We have developed a software prototype for the Android operating system with the purpose of mitigating the number of contagions produced by risky behavior and/or a lack of information about outbreaks or massive virus contagions. The software has an interface that allows the user to see georeferenced alerts around him. Also, the software is capable of monitoring diverse risk areas and/or the use of programs that facilitate risk behavior, without the user ever being conscious of it. This way, the software can send instantaneous prevention messages.

Conclusions: In this work, we have developed the first free software for mobile devices with the Android operating system that can preventively mitigate the number of contagions of sexually transmitted infections (STI), associated with risk behavior. This software runs in two modes. The normal mode allows the user to see the alerts and nearby health centers. The second mode enables the service to work in the background. This software reports the health risks, as well as the location of different test centers. With these notifications, we hope that the user will become aware of the problem at the moment of maximum risk, in a way that is transparent and doesn’t affect the character of the applications that are at the center of this risk behavior.
Background

In the last few years, there has been a clear increase in the number of outbreaks of infectious diseases or other types of illnesses that had been previously categorized as under control. Epidemics have played a large role in the history of mankind; globalization and new technologies have contributed to the rapid movement of viral vectors, causing a large number of infections and increasing the level of incidences in different geographic sectors of the planet.

These infectious pathologies, which pose a great risk to the community, possess a high rate of morbidity and mortality. Among these diseases are: the pulmonary syndrome of the Hanta virus, tuberculosis, hepatitis, and HIV. The last of these examples is one of the principal mortal infectious diseases, according to the study published by the Worldwide Health Organization in 1999.

Given the viral character of this group of pathologies, epidemic outbreaks could emerge, which is a great danger for communities or individuals in contact with risk vectors, which can move about freely in determined geographic areas [1].

Currently, the prolific development of information and communication technologies, as well as the exploitation of social networks such as Facebook, Twitter, and My Space, has increased the rate of interrelations among distinct stratum of our society [2]. These changes have initiated the study of diverse behaviors among various groups of people. One of the most common behaviors is the sharing of personal information with other members of the social network, without consideration for essential confidentiality issues. In developed countries, these interactions have triggered diverse forms of risky behaviors. In other words, groups of people use social networks and the Internet to meet with sexual partners that are possibly infected with STIs or even HIV [3]. 98% of youth spends a minimum of 44 hours in front of a computer per week, which is significant since the computer is one of the principal enablers of risky behavior. In particular, one group of young people uses the Internet to search for sexual partners or research sex topics, publishing information on profiles or websites to enable the interaction among social groups from the same stratum [4]; at the same time, young people who are not involved in this behavior can be influenced by the open communication that social networks generate.

Smartphones, tablets and rapid channels of software distribution, such as Android’s Google Market and Iphone’s OS Apple Store [5], have increased the level of interaction among these stratum, providing specialized tools that lead to “risky behavior” [6]. Some examples of this software are: Grindr [7], Man-Hunt [8], Hawt [9], etc. They make use of georeference techniques, such as localization by towers or cells and GPS, to facilitate the process of localization among individuals that are looking for sexual encounters, regardless to whether they live in a large metropolitan region or its surrounding areas [10].

Minimizing the risk of contagion

Organizations have tried to mitigate or minimize the risks of contagion of the aforementioned pathologies in a variety of ways. Today’s youth has grown and matured alongside technology. In particular, this group doesn’t view technology as a tool, but rather as an entity that is indispensable for their lives [11]. As a result, there have been innumerable efforts to use TICs with preventative health objectives. [4], in the article “Using Technology, New Media, and Mobile for Sexual and Reproductive Health”, creates a compendium of the most commonly used TIC solutions for the prevention of sexual pathologies in 2010, highlighting, for example, Wiki, eCards, Podcast, VideoSharing, Games, etc [4]. These last examples succeed at reaching a large percentage of youth; nevertheless, the solutions only establish communication in one direction, without interacting with the users on the basis of their necessities. Furthermore, they are exclusively entities that distribute information to interested parties; consequently, it is very important to define the population in which the intervention will occur, so that the appropriate tool could be selected for distributing information correctly.

One case study in the use of TIC tools for the prevention of diseases, was carried out through Facebook: “The Just Us study” [11] is a representation of developing sexual education through a social network. The study analyzed the online behavior of 1,588 young students, showing how the youths could communicate freely, without shame, about
Figure 1: **Models of ubiquitous computing**: This figure represents the different models of use for ubiquitous computing in daily life. This omnipresence and imperceptibility can be used as principal axes in the paradigm of the development of software or ubiquitous procedures.

Topics associated with the prevention of diseases, such as HIV. The young people represented themselves in the network through a single and unreal image, known as the user, which allows the youths to detach themselves from interpersonal communication barriers, including shame. This is harmful because it makes vulnerable several confidentiality factors that are associated with the person.

**Trend from personal computers to mobile devices**

One of the principal problems of using personal computers and websites for prevention is that they do not cover the entire possible spectrum of the target population. The Internet and the services it offers have been evolving day to day. The users of the e-world have migrated to mobile devices, which are currently seen as pocket computers, because of their capacity for processing, physical memory, and network connection, among other features. Therefore, mobile devices, instead of computers, should be used in response to communities’ health care needs.

The advances in technology have permitted the integration of other types of hardware, for example, movement sensors (accelerometers), diverse types of connectors, etc. One example that is particularly important is GPS (Global Position System), because of its potential for controlling contagion. It can be used to track geographical movement, thus contributing relevant information to prevent or contain outbreaks.

**How does the use of mobile devices influence our lives?**

The increased use of mobile devices does not only affect youths, but rather the population as whole, as new models of human-computer interaction emerge. Computers interact significantly with our lives. Nowadays, it is normal to have at least one technological device, which abstracts us from the real world to carry us to a fictional world, as Mark Weiser described in 1991 in his work, “The Computer for the Twenty-First Century”.

The interoperability with different connected devices gives us a sensation that these devices do not exist. In other words, when a person sits at his desk to write a paper, he uses the keyboard, screen and the computer’s resources; however, as he is writing, in the precise moment when he is putting down every thought, the actor is not conscious that he is using a computer that has an operating system [12][14]. See Figure 1. This new model of interaction, known as ubiquitous computing, will be applied in this study to solve problems related to contagion and infectious diseases.
The architecture of the solution: The architecture has two sections: the first one is the data entry in the information system through a web client, the second is the communication of data through a restful web service and the mobile clients.

The real need

These trends in technology create opportunities to respond to communities’ health needs, confronting the problem of contagion and viral pathologies while allowing the community to make informed decisions with respect to their behavior and geographical movement.

For instance, as a first security measure in an epidemic, it is necessary to reduce the social contacts between infected individuals and the community. Consequently, mass gatherings, where the probability of being infected increases considerably, must be avoided [13]. Using mobile devices to gather statistics and report new cases of contagion, enables interested parties to store epidemiological data in real time and thus prevent contagion [14].

To clarify how technology could minimize the risk of contagion, we will briefly consider an example: HIV. Certain social groups use networks and the Internet to find potential sexual partners who could be infected with STIs or HIV. Therefore, it is necessary to mitigate the number of contagions. This is most effectively done through the use of multiple communication media and specifically mobile devices because they are the same instruments that enable the risky behavior. At the very least, it is important to establish a way of communication among people that want to stay informed about recently reported cases of HIV, or the location of a HIV test center, near to the user or to a determined place.

In summary, the objective of using mobile devices is to inform users, at the moment when they engage in risky behavior, of the epidemics present in the zone and the location of test centers. Thus, mobile devices could be excellent tools for intervening and distributing information related to protecting our health [3, 15].

Many people that suffer from a disease or sporadically partake in risky behavior are afraid of being rejected socially or psychologically. Ubiquitous computing offers a solution, focusing on the interactions between the virtual and real world, through the omnipresent connectivity for collectives of people that use TICs as a way of communication to find sexual partners without thinking about the associated risks.

Methods

In this section, we will describe the architecture of the solution and its system, along with its crucial components and their functions.
In order to develop a comprehensive solution we must consider the following aspects:

- Creation and storage of geodata sources.
- Development of a web service for distributed systems.
- Integration with mobile devices.

The solution consists of three principal parts (See Figure 2):

- Web server: The web server carries out the main function of storing geodata. It also serves as a standard interface of communication through the implementation of a Restful web service with the mobile clients.
- Web client: The principal function of the web client is adding georeferenced points to the database for viral alerts and medical centers.
- Mobile client: The mobile client communicates with the web server with the aim of getting points of interest, whether they are alerts or medical centers.
the business section. This second section executes a search algorithm of geopoints that interact directly with the data layer in our system. It is important to point out that the service subdivision is the section that ultimately interacts with the diverse clients associated with the service.

The data layer, or persistence data layer, is a database that stores the distinct geopoints, together with their associated characteristics.

**Android client**

The presentation layer consists of the integration of the OpenStreetMap with Android that shows the different alerts that exist in a determined area. It is also the interface for accessing the configurations and different modes of execution of the application, which has been designed to be simple for the user.

The domain layer is subdivided in two sections:

- **Service section:** This section implements two internal monitoring services that have the responsibility of reviewing and comparing certain mobile states. These states relate to the localization and the applications that are running at a determined moment. At the same time, it interacts with the business section.

- **Business section:** This section contains the core of the application that interacts with JSONParser, which has the responsibility of communicating with the RESTful service explained previously.

The data layer, or persistence data layer, corresponds to the application’s preferences. Additionally, this layer contains the strings associated with the application, which are independent of all the other layers. This independence allows us to be able to re-use diverse components of our application and at the same time work on important internationalization aspects.

**Entering data into the information system**

To enter the data into the information system, we have developed a web client (See Figure 4), with which any user could add alerts and/or medical centers to our database. In order to achieve this, we have used PHP as a control language [19], OpenStreetMap [20] as a free map framework and the
OpenLayers libraries \[21\] to administer the distinct layers of maps.

To enter an alert, the service provider must enter the name, the affected area, and then instinctively select the point on the map. Automatically, the system will store the geoposition of the alert (see Figure 4).

In Figure 4, we can see the different risks stored together with their risk radius of contagion. This is particularly interesting since the respective health organizations would be able to obtain important statistics regarding the frequency of viral outbreaks of a determined type.

Development of the RESTful service

The client applications carry out requests through GET methods from HTTP protocol about certain resources of the server. These resources are identified through a URI. HTTP protocol is used in RESTful services to interact with the server’s resources. This is extensively beneficial for mobile clients since they have limited resources. Through this design we will lighten the client load by turning it over to the server, which has more resources. Moreover, we must consider engineering aspects about the uniformity of access to data. Through this interface, any client could obtain the results. This removes the interaction between client-server and provides more independence to each component of the design. Therefore, when faced with an eventual transformation in the server or client, it will not be affected thereby providing a solution with greater interoperability and modularity.

We developed two services: the first is devoted to the alerts and/or viral outbreaks, and the second, clearly to the medical centers. Both of these services are set from JsonParser to Restful service with a determined URI (See Figure 3).

The alert service receives a request from the alerts, together with a set of important data in the URI, which are: geolocalization of the user and search radius. With this information, the service communicates with the previously mentioned persistent data layer. More specifically, it communicates with the database in order to get all the points
stored within it and thereby calculate the distance between the point at which the user is located and the point from the database. If the distance is less or equal to the radius sent from the mobile, then the service responds with the geopoints, the name of the type of risk and the radius of the affected area. All of this data is parsed with Json-encoded. It is important to point out that the service could eventually not find any alert in the sent radius.

The medical centers service receives a request from the mobile, along with the geolocalization of the user in the URI.

With this information, the service communicates with the previously mentioned persistent data layer. Specifically, it communicates with the database in order to get all the points stored within it. Then, this service calculates the distance from the client's point to the point stored in the database to choose the point with the shortest distance. This point corresponds to the medical center that is closest to the user. The service thus responds with: the geopoints and the name of the medical center. All of this data is parsed with Json-encoded.

Development of the android client v0.01

We have developed a software prototype that is able to recognize when each risk application is running, with the purpose of notifying the user of the possible risks that he is subject to. It would be, in a certain sense, a preventative medicine system, indicating the nearest medical center so that the corresponding tests could be carried out. In addition, the software is capable of notifying the user when he nears a determined risk zone. This way, it prevents contagion of viral diseases due to the user’s lack of information.

To begin to achieve this, we have divided our prototype in three parts:

• Notification of zones at risk of contagion.
• Background notification of zones at risk of contagion.
• Background notification of the use of software that facilitates risky behavior.

With the aim of developing an application with reusable components, modular and independent, we have based our development on General Responsibility Assignment Software Patterns (GRASP). These patterns allow us to have high cohesion and low coupling among objects [22,23]. To analyze our architecture, we will base our model on the general black box sequence diagram (See Figure 5). In this diagram, we can see the relationship that exists between the different software packages:

• cl.tfm.Map: This package has the necessary classes to be able to visualize the affected areas at risk of contagion, using the OSMdroid libraries [24].
• cl.tfm.JsonParser: It has the necessary classes to be able to make different requests to the Restful service and process data, thereby allowing its later representation with other components of the software.
• cl.tfm.services: This package stores the internal services that are responsible for notifying the user when he nears a risk area or runs a program that facilitates risky behavior.

Visualizing risk areas

In order to visualize the risk areas (See Figure 6), we created an object instance of the Activity type [18], which follows the life cycle of a Activity defined by the Android operating system. This object gets the stored preferences, such as the radius of the search of the risk areas, in the layer of persistence data defined by an XML, instantiating a PreferenceManager object of the Android API. Three values exist by defect, which are: 3, 5 y 10 kilometers around. With this data, we can instantiate the class that communicates with the service Restful, JsonParser. This class sends a request with a URI that has the longitude, latitude of the current position of the mobile device and the radius of the search. The answer is saved in an ArrayList of the OverlayItem type, defined in the API of OSMdroid [24]. Inside of every list, each OverlayItem object contains the latitude, longitude, the name of the risk, the affected radius and the icon to be drawn on the map.

To draw the risk areas on the map, three layers were made:

• Map layer or mapview: In the main Activity, specifically in its onCreate() state, we load a map view, which is defined by a XML
file. We made the layer and added several characteristics defined by the API of OSM-droid, such as `setBuiltInZoomControls(); enableCompass(); enableFollowLocation();`.

- Point layer: It gets the latitude, longitude and the icon of the `ArrayList` made by the Json-Parser object and adds `OverlayItem` objects to the layer.

- Area layer: A new `Overlay` object was instantiated, overriding the `draw()` method. In this method, we review the `ArrayList` of the `OverlayItem` type list with the geopoints. Also, we make a point in the same localization but with a circle in the middle drawn with `Canvas`, specifically with `drawCircle()`. In order to measure the radius accurately, we get the projection of the map and we transform it with the geopoint. Similarly, to transform the distance in meters, we use the API method from OSM-droid called `metersToEquatorPixels()`. Thus, each time that a `ZoomIn` or `ZoomOut` is performed, the method will re-draw the radius of the circle according to the map’s projection and `Zoom`.

Each `Activity` in Android has its own life cycle. Therefore, it is important to point out that in each of the states in the cycle, they interact with the operating system. In this section, we worked with the following states: `onCreate()`, `onStart()`, `onDestroy()`, in which different interactions with the respective components of the operating system were instantiated, run and killed.

### Service development

Android services have their own life cycles, and like the Activities, the life cycle of a service is between `onCreate()` and `onDestroy()`. Similar to an `Activity`, a service’s initial configuration is created in `onCreate()` and frees all the resources that it uses in `onDestroy()`. The active life of a service begins with a call to the `onBind()` method.

### Alert service

In Figure 5, we can see that the alert service is communicated with 2 essential components: `JsonParser` and `NotificationManager`. The algorithm for detecting if we are near an area stored in the web server.
Figure 7: States diagram, alert service  This diagram represents the different states of the alert service. This service will be running between different intervals of time. Thus, we can save the mobile devices' limited resources, like battery.

database is based on the implementation of a LocationListener, which has the following states (see Figure 7). Once the service is initialized, in the onCreate() state, we define some resources that will be used, such as the LocationManager of the Android API. As a provider of the service, we use the NETWORK_PROVIDER. The accuracy of the localization has noise that is no more than 10 to 15 meters, and the search radius is larger than 3 kilometers. Therefore, there is greater cost-benefit to using the NETWORK_PROVIDER than there is using the GPS_PROVIDER, because it saves the mobile's resources. Once the provider is selected, we call a requestLocationUpdates(), defining its parameters: the provider, the quantity of time and the distance that we want to update. After the update, if the location has changed the onLocationChanged() method, we get the current position, passing it to the notification class. This instantiates JsonParser with the current location of the mobile with the aim of getting a risk area near the radius previously defined by the user. If there is an alert, it is communicated with the operating system’s NotificationManager. To alert the user, we have developed sound and vibration notifications using the message bar of the mobile device with a preventive alert. Also, an Intent is activated when the user selects the alert in the bar. If there is no alert, the service will continue functioning until the user or the operating system kills the service.

Risk service

Similar to the previously described service, the monitorization service of risky applications is communicated with two essential components: JsonParser and NotificationManager. The algorithm for detecting a risk application instantiates ActivityManager of the Android API. This allows us to access information about the processes that the operating system is running, so we can compare the running application list in a determined moment (TimerTask time intervals) with our list of risky applications. If the program is being run, then JsonParser is instantiated. It immediately makes a request to the medical center nearest the current location in order to notify the user through the NotificationManager with an alert message that contains location of the nearest center. If a risky program is not being run, the ser-
vice will continue functioning until the user or the system kills it.

**Internationalization**

With the aim of having a modular and independent architecture, Android [22, 23], allows us to store the data and strings in XML files. This information on the system’s strings is referenced with tags. These strings correspond to a specific language. This language depends on the locales defined by the mobile. To store the information in different languages, we used the model determined by the ISO 639-2 in the alpha-3 version. This represents the names of the languages as Datos-Es for Spanish and Datos-En for English.

**Results and Discussion**

After introducing the problem, in this section we will see that the proposed application is an effective and efficient form of preventing contagions, and even risky behavior, through mobile devices.

When a user is in an unknown area, he is more likely to act without considering the risk, for example: camping in the mountains without considering the possible risk of Hanta contagion; visiting zones where there is a high rate of tuberculosis contagion; or using an application such as Manhunt or Android, looking for sexual partners without thinking about the possible risks of contagion associated with this behavior. Therefore, as a measure for mitigating contagion, we used the distribution of information about risk areas so that users can use this software prototype to make informed decisions.

When the user opens the software prototype, all of the risk areas that are close to the user’s geoposition are gotten and shown on the map screen (See Figure 6). Each red circle corresponds to an area at risk of contagion. These have been drawn based on the positions, radius measurements and specified map projections. If the user touches the icon over the risky area, he will be able to see what the alert corresponds to (See Figure 8). Each one of the points is gotten from the central data server. The user can configure the distance to search around him. Our objective was to develop a simple interface so that the user can access information quickly and configure the software, from the minimal functionalities, such as ZoomIn and ZoomOut, to functions such as seeing different locations of the map.

The application works in two modes:

- **Voluntary mode**: In this mode, the user runs applications with the purpose of getting in-
This sequence shows from when the user launches the notification service, to when the service detects a risk zone near the location of the user. The service also notifies the user in the message bar. When the user selects that notification, the software opens a new window with the location of the risk and the user.

formation about the different risks that exist around him.

- Ubiquitous mode: This option must be configured in the software. In this mode, the user doesn’t need to run the application for notifications to arrive about risks associated to his geoposition and the use of risky software. This is because the application runs constantly in the background, so that it is able to give the user maximum liberty in using the mobile for whatever type of action, while keeping the user informed.

The software is configured through the data configuration screen (See Figure 9), under the Android Guidelines:

- Show instructions: Shows a brief summary of how the software works.

- Radius distance: Limits the distance of the search radius for risky areas.

- Monitorization time: Defines how often the user wants the services to monitor the situation.

- Ubiquitous mode: Initializes the risk application service each time that the mobile device is turned on, without the user needing to start the process.

- Vibration: Sets the mobile to vibrate when an alert is shown.

- Sound: Sets the mobile to make a sound when an alert is shown.

When the user runs any of the notification services, he can close the application and the services will function in the background of the mobile. This allows the user to have a completely different experience, since he will able to use his device normally, for example, to talk on the phone, navigate the web or simply leave it idle.

The monitorization services will work according to the user’s configuration of, for example, the check time and radius of the considered area. Figure 10 shows how the user can manually turn on the service. Once the software detects that the user is at risk, it will notify him with an alert in the message bar of the mobile. Once the user reads the message, he will instinctively touch the notification. This will open a map visualization software installed on his
mobile, which could be OSMdroid or Googlemaps, among others, showing his current position and the affected area.

When the user runs the monitorization service for the risk application, or selects the ubiquitous mode, it will be constantly monitoring the list of programs that are running in the operating system (this service checks this list every 3 minutes). When it finds a match, for example when it detects the ManHunt program, (See Figure [10,11]), it will alert the user in the same manner described previously.

Once the alert is made, our software has the capability of interacting with other software installed on the operating system. Thus, the user will not only be able to visualize the alert, but will also be able to share it through email, text messages, social networks, etc. In summary, the user will have a fully-connected experience.

Upon seeing the different alerts, the user will be able to make informed decisions regarding the possible consequences of entering the risk zones. This will significantly reduce the number of contagions due to a lack of knowledge about risky zones or zones in quarantine. In the same way, we believe that the alerts about risky behavior, along with the localization of nearby medical centers, could make people more aware of their actions and guide them towards taking necessary measure to avoid contagion and the spread of diseases.

Conclusions
In this work, we have developed a system of alerts with the objective of making the user aware of risk zones, situations and behavior.

In the first place, there exists a need to develop new preventive medicine methods. Mobile devices are currently part of the life of a majority of people. As a result, we believe that these devices can contribute to research in the area of preventative medicine.

When a user is vulnerable to risk of contagion due to a lack of information, this contagion is one hundred percent preventable. After all, people make decisions based on information that this prototype will help to provide about the areas that they pass through.

If a user runs one of the mentioned risky applications, he will be vulnerable to taking actions that put not only his health at risk, but also the health of
others. The greatest indexes of contagious venereal
diseases, including HIV, are the product of actors’
carelessness. These acts of licentiousness must be
carried out with the appropriate measures of health
control. We believe that the alert notifications’ non-
 invasive interaction and the references to medical
centers could help mitigate the risks on contagion
through these acts, since an alert designed to in-
crease awareness arrives at the moment of maximum
risk. Considering that each time someone uses his
mobile to contact other users and participate in risky
activities, he will be notified of his action, taken by
surprise, at his moment of greatest vulnerability.

The addition of new functionalities in this proto-
type is technically easy to implement, given that we
present a modular architecture with high cohesion
and low coupling among classes.

Future work will be centered on measuring the
real impact of this proposal in terms of statistically
significant study groups with the goal of improving
technologies, interfaces and interaction with the
user. Another challenge will be the integration with
world data banks such as HealthMap [25], in order
to distribute more georeferenced information in the
field of health.

Authors contributions
FB has developed the applications, the architecture
of the solution to the initial problem and has written
this article. He also has developed with AP the
proposal of the solution. This research is a product
of his Master’s thesis of the Masters in Free Soft-
ware at the University Oberta of Catalonia. AP
has participated as a thesis advisor, coordinated its
execution, and provided comments and suggestions
on the manuscript.

Acknowledgements
This research has been overseen by the UOC. The author
FB, deeply thanks the institution for the support and
advice in this process. He also thanks the free software
community that makes possible the use and distribution
of these technologies, which are knowledge for humanity.

References
1. Gonzalez MC, Hidalgo CA, Barabasi AL: Understanding
Cited: 287 Cited Reference Count: 30]

2. Lazer D, Pentland A, Adamic L, Aral S, Barabasi AL, Brewer D, Christakis N, Contractor N, Fowler J, Gut-
mann M, Jebra, King G, Macy M, Roy D, Van Alstyne M: Social Science Computational Social
Count: 17 Lazer, David Pentland, Alex Adamic, Lada Aral, Sinan Barabasi, Albert-Laszlo Brewer, Dev-
on Christakis, Nicholas Contractor, Noshir Fowler, James Gutmann, Myron Jebra, Tony King, Gary Macy, Michael Roy, Deb Van Alstyne, Marshall AMER ASSOC ADVANCEMENT SCIENCE WASHINGTON].

3. Rietmeijer CA, McFarlane M: Web 2.0 and beyond: risks for sexually transmitted infections and opportunities

Count: 22 Levine, Deb SPRINGER NEW YORK].


6. Margolis AD, Joseph H, Belcher L, Hirshfield S, Chias-
on MA: ‘Never Testing for HIV’ Among Men Who Have Sex with Men Recruited from a Sexual Net-

7. Grindr - Find gay, bi, curious guys for free near you [http://grindr.com/].


15. Kaplan WA: Can the ubiquitous power of mobile phones be used to improve health outcomes in developing countries? *Globalization and Health* 2006, 2. [Times Cited: 39].

16. OSI, Open source Initiative [http://www.opensource.org/].

17. Free Software Foundation [http://www.fsf.org/].


20. Open Street Map project [http://www.openstreetmap.org/].


