©Copyright 2017, Stranger Devices. All rights reserved. All content and plugins are exclusively developed by Esteban Gómez

INTRODUCTION

Stranger Devices Performer Suite is a set of 4 Max for Live devices focused on improving workflow of musicians who are mixing traditional and virtual instruments during live performances. The effort behind the plugin design and programming is meant to encourage musicians to create with both kind of sources without thinking of them as two separate worlds.

All the plugins and associated content was planned, designed and executed by Esteban Gómez as his Culminating Experience project as a candidate for Master of Music in Music Production, Technology and Innovation in Berklee College of Music during 2016 and 2017.

For any further information, feel free to contact Esteban through his website: www.estebangomez.cl

ACKNOWLEDGEMENTS

Utmost gratitude is owed to all the people that supported this project, whether with guidance or collaboration during the different stages of its development:

Pablo Munguía: Program director
Pierce Warnecke: Culminating experience advisor
Ben Cantil: Prototype feedback and Ableton wizard
Dimitri Carabas: Cameraman and voice-over for demos
John Stewart: Drummer for demos
Tejaswi Gorti: Manual proofreading and support during recording sessions
Andrea Cazza: Support during recording sessions
Chris Heckman: Demo's plot revision
Dav Abrams: Great friend offering unselfish support during technological disasters

INDEX

OVERVIEW

Stranger Devices Performer Suite comprises 4 Max For Live plugins and their respective OSC layouts when communication with a mobile device is necessary. Users can still create their own OSC layouts as needed. The messages that each plugin and OSC layout is expected to send and receive is indicated in each of the plugin sections included in this manual.

In the next pages you will find all of the details on how to include and use every plugin in your session.

For further information, feel free to reach out the programmer, whose contact details are included in the introduction section of this manual.

GETTING STARTED

Before using **Stranger Devices Performer Suite** you will need to install **Ableton Live** on your computer. If you want to send OSC messages, you will need to load the free **TouchOSC** layouts included in the suite. It is highly recommended to create your own private network while working with OSC, otherwise you can experience a significant latency while sending and receiving messages from your mobile device to your computer and vice-versa.

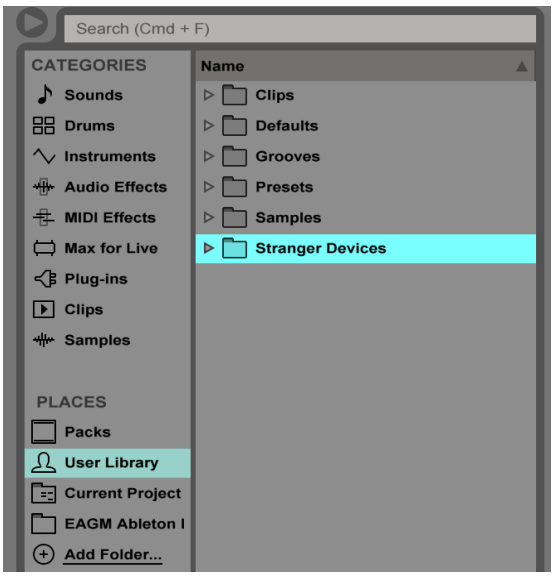
In order to make your plugins visible by **Ableton Live**, you must copy the **Stranger Devices** folder to your **User Library**. Depending on your operating system, the **User Library** can be found in the locations listed below:

Windows Vista, Windows 7, Windows 8, Windows 10:
\Users\[username]\Documents\Ableton\User Library

Windows XP
\Documents and Settings\[username]\My Documents\Ableton\User Library

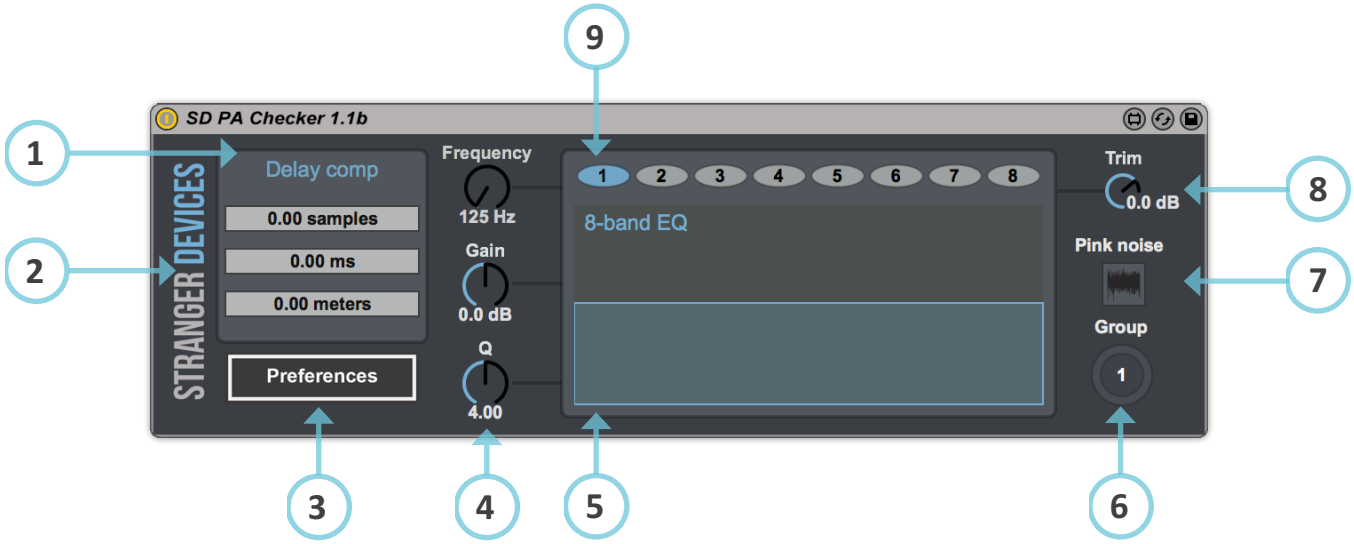
Mac OS X
Macintosh HD/Users/[username]/Music/Ableton/User Library

After you have copied the folder, the plugins should be visible in an **Ableton Live** session. Now you can drag and drop any plugin you want to use into a MIDI track or an audio track.



PA CHECKER

If you need to adjust your PA system before your performance, but you want to hear how it sounds from the audicence's perspective, then **PA Checker** is your friend. This plugin features three main sections in order to accomplish this task: Delay compensation, equalization and trim. You can also group different instances of **PA Checker** under the same settings. This is especially useful if you are in a venue with more than one stereo pair or if you want to modify the settings of several speakers at once.



1. Delay compensation: Delays the audio output. The amount can be adjusted in samples, miliseconds and meters.

2. About window: Click the logo to display the about window of the plugin.

3. Preferences: Allows you to adjust the OSC settings of the plugin.

4. Frequency/Gain/Q: Active band parameters in the equaliser can be adjusted by moving these knobs.

5. EQ frequency response: Visual display of the current overall magnitude response of the equaliser.

6. Group: The plugin group can be set from 1 to 8. All instances of PA Checker within the same group will share their settings.

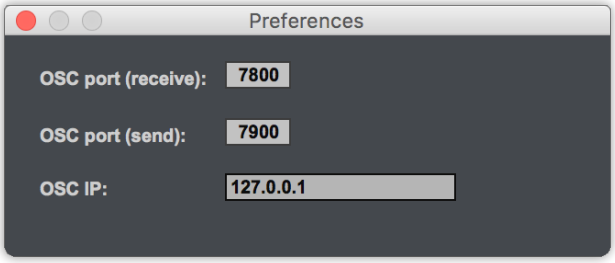
7. Pink noise: Pink noise signal for testing purposes. Bypasses channel input.

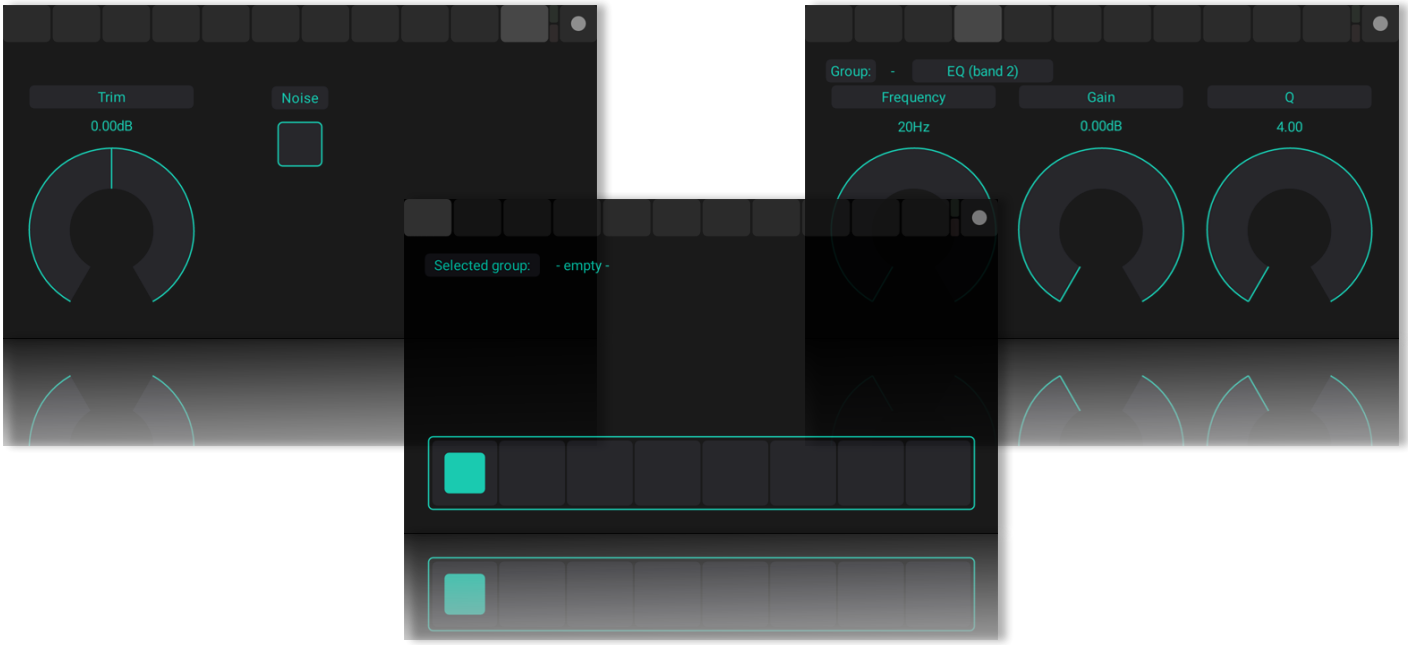
8. Trim: Adjusts the gain of the output independently from the channel gain.

9. EQ bands: The active band of the equaliser will be highlighted in a blue background.

PA CHECKER

In order to communicate **PA Checker** with your mobile device, you must be sure that both are connected to the same network. Later on, you must set the correct IP and port of your mobile device in the **Preferences** window of **PA Checker** and set the IP and port of your computer in the OSC application of your choice. It is highly recommended to keep your network isolated from the venue's public network where you are using **PA Checker** in order to avoid undesirable latency in OSC communication.



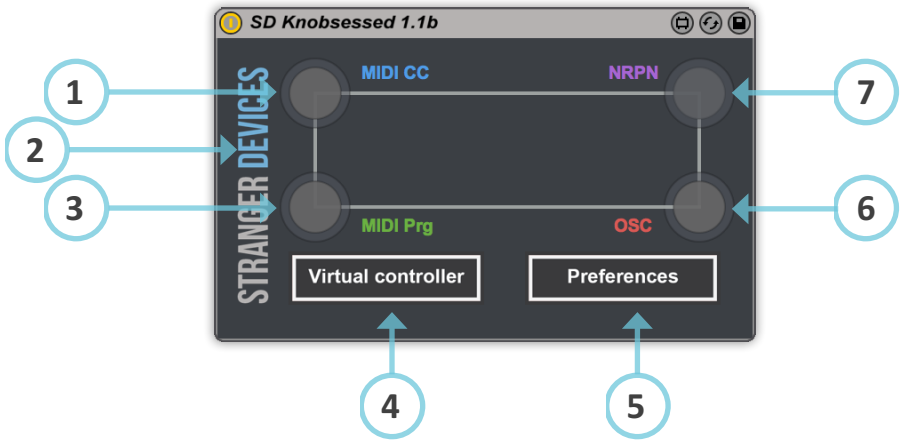


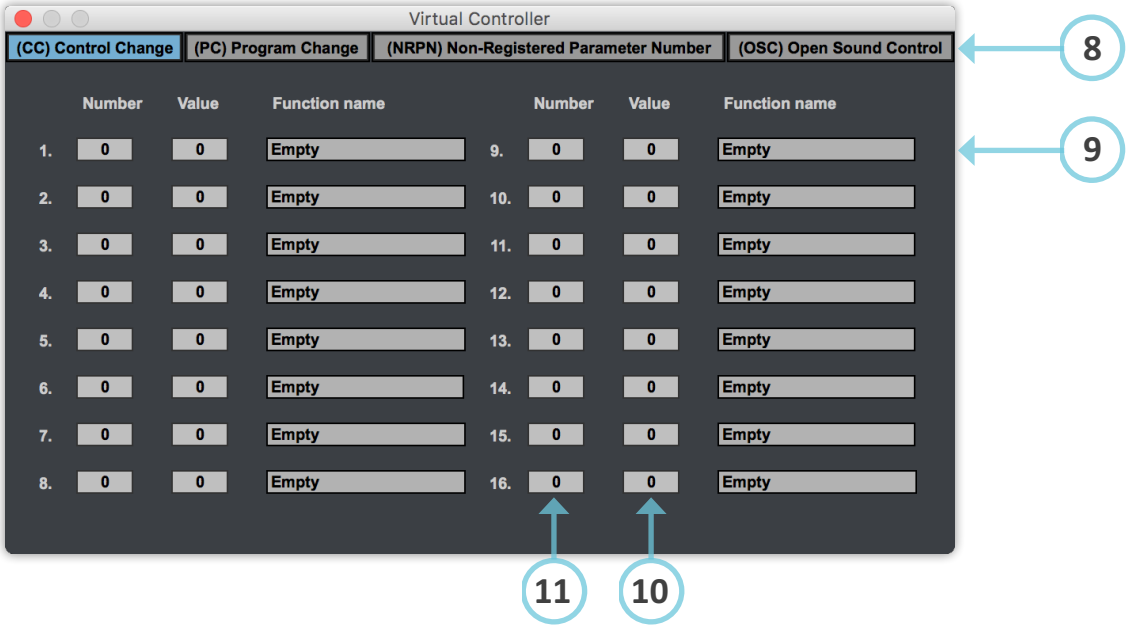
PA CHECKER (OSC CHART)

Address pattern	Values	Function
/sd/pach/external/group/1/{x} {y}	{x}: 1-8; {y}: 0-1	Changes target group
/sd/pach/public/comp {x}	{x}: 0.-1.	Delay compensation amount
/sd/pach/public/eq/{x}/frequency {y}	{x}: 1-8; {y}: 0.-1.	Target frequency of band {x}
/sd/pach/public/eq/{x}/gain {y}	{x}: 1-8; {y}:0.-1.	Gain of band {x}
/sd/pach/public/eq/{x}/q {y}	{x}: 1-8; {y}:0.-1.	Q of band {x}
/sd/pach/public/trim/left/gain {x}	{x}: 0.-1.	Trim gain
/sd/pach/public/trim/left/noise {x}	{x}: 0-1	Toggles pink noise generator
/sd/pach/external/label/group_number {x}	{x}: 1-8	Selected group number label
/sd/pach/external/label/comp_time {x}	{x}: 0-1000	Compensation time (ms) label
/sd/pach/external/label/comp_samples {x}	{x}: 0-Sample rate	Compensation samples label
/sd/pach/external/label/comp_meters {x}	{x}: 0.-343.	Compensation meters label
/sd/pach/external/label/eq/{x}/frequency {y}	{x}: 1-8;{y}:20-20000	Target frequency label of band {x}
/sd/pach/external/label/eq/{x}/gain {y}	{x}: 1-8;{y}:-12.-12.	Gain label of band {x}
/sd/pach/external/label/eq/{x}/q {y}	{x}:1-8;{y}:0.-8.	Q label of band {x}
/sd/pach/external/label/trim/left/gain {x}	{x}:-70.-12.	Trim gain label
/sd/pach/external/instances {x}	{x}: 1	Select instances page
/sd/pach/external/page/comp {x}	{x}: 1	Select compensation page
/sd/pach/external/page/eq/band_{x}	{x}: 1-8	Select {x} band page
/sd/pach/external/page/trim {x}	{x}: 1	Select trim page

KNOBSESSED

Sometimes, musicians want to include the sound of a specific piece of external gear inside of an Ableton Live session, but recreating everything with exact detail every time you turn on your computer and open your session could be a tedious task. **Knobsessed** can help you to do this. It does not matter if the external gear communicates through MIDI, NRPN or OSC messages. You can still program everything that is happening inside it and keep track of what parameter you are moving and use it not only inside the studio, but also during live performances.





1. MIDI CC monitor: Turns on when a Control Change message is sent to an external device.

2. About window: Click the logo to display the about window of the plugin.

3. MIDI Prg monitor: Turns on when a Program Change message is sent to an external device.

4. Virtual controller: Click this button to open the virtual controller window.

5. Preferences: Allows you to adjust the OSC settings of the plugin.

6. OSC monitor: Turns on when an Open Sound Control message is sent to an external device.

7. NRPN monitor: Turns on when a Non-Registered Parameter Number message is sent to an external device.

8. Message panel tabs: Click the type of message you want to send to your external device and the message settings for that type will be displayed.

9. Function name*: The name to idenfity the modified parameter in the external device.

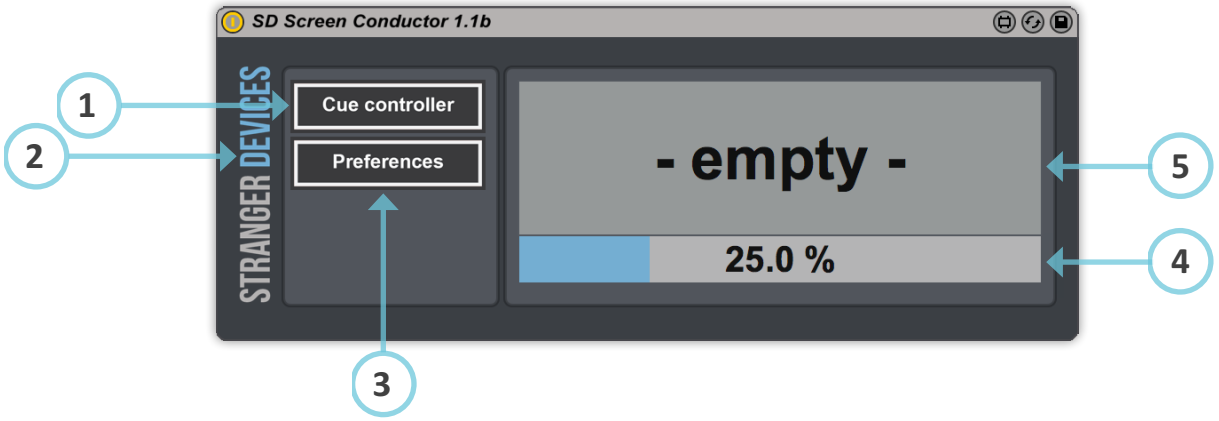
10. Value*: The current value of the modified parameter in the external device.

11. Number*: The number of the current modified parameter in the external device.

*The settings displayed in the window will vary depending on what type of you message is being sent to the external device.

SCREEN CONDUCTOR

While performing under certain circumstances, ears are not enough for interpreting a specific cue correctly, and the visual aspect is needed. It can be about the next song in the setlist, a direction of an improvisation or a message you must deliver quickly to the sound engineer or road crew. **Screen Conductor** was specifically designed to address these situations. Write text messages to the mobile devices of band members or whoever is in the same network where **Screen Conductor** is, and it will be delivered in the specific time and with the specific duration you programmed. Additional information about bar, beats, time signature and tempo can also be sent.



SCREEN CONDUCTOR

1. Cue controller: Click this button to open the cue controller window.

2. About window: Click the logo to display the about window of the plugin.

3. Preferences: Allows you to adjust the OSC settings of the plugin.

4. Progress bar: Shows the progress of the current bar.

5. Message monitor: Monitors the messages that are sent to an external mobile device.

6. Start bar: Sets the first bar where the message is sent to an external mobile device.

7. Start beat: Sets the first beat where the message is sent to an external mobile device.

8. End bar: Sets the last bar where the message is sent to an external mobile device.

9. End beat: Sets the last beat where the message is sent to an external mobile device.

10. Cue message: Sets the text message that is sent to an external mobile device.

11. State: If the button is checked, the message will be delivered to a mobile device during the specified time, otherwise it will be skipped.

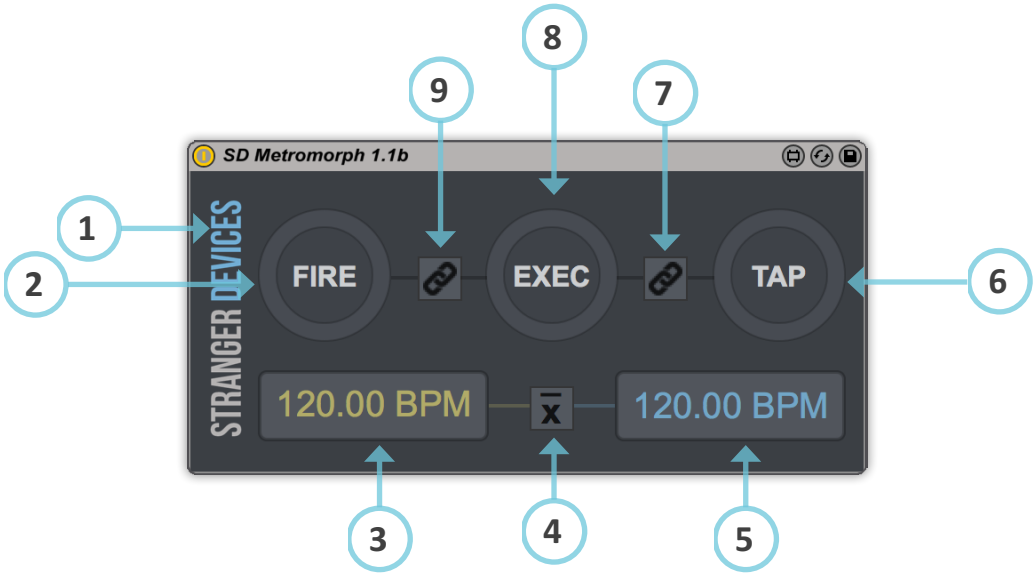
12. Cue tab: Click to select the page with the cue you want to edit.

SCREEN CONDUCTOR (OSC CHART)

Address pattern	Values	Function
/sd/scco/external/label/instruction {x}	{x}: String	Changes displayed message
/sd/scco/external/label/progress {x}	{x}: 0.-1.	Progress of the current bar
/sd/scco/external/label/tempo {x}	{x}: 20-999	Tempo of the current song
/sd/scco/external/label/time_signature/numerator {x}	{x}: 1-99	Numerator of time signature
/sd/scco/external/label/time_signature/denominator {x}	{x}: 1-16	Denominator of time signature
/sd/scco/external/label/bar {x}	{x}: 1-9999	Current bar
/sd/scco/external/label/beat {x}	{x}: 1-99	Current beat

METROMORPH

When a band incorporates electronic sequences and software-generated sounds during live performances, the common way of doing it is through a click track that the drummer or the rhythm section follows with the help of monitoring systems. This could make the performance sound exactly the same every time and thereby limits the natural character of it. With **Metromorph** you can tap tempo the next sequence you plan to launch before doing it. You will be able to adapt the tempo of your clips to the current band tempo, then when you launch them they will follow the tempo of the band.



1. About window: Click the logo to display the about window of the plugin.

2. Fire button: Click here to fire the selected clip. It can be assigned to a MIDI controller.

3. Average tempo: The average tempo of all the recent tap tempo actions.

4. Average: If activated, will assign the average tempo to the selected clip, otherwise the instantaneous tempo will be assigned.

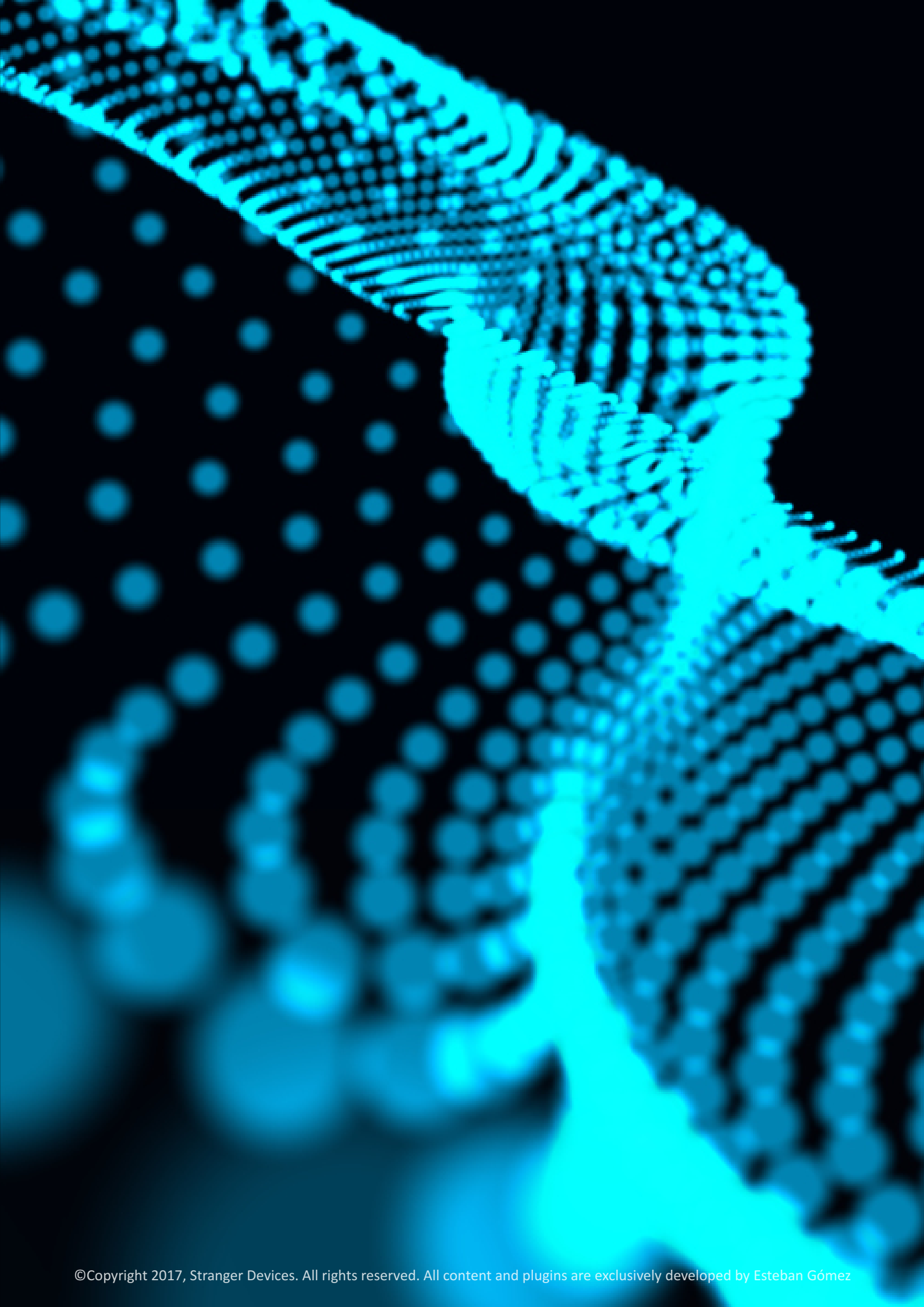
5. Instantaneous tempo: The tempo of the last tap tempo action.

6. Tap button: When clicked twice, calculates the tempo based on time-difference. It can be assigned to a MIDI controller.

7. Tap&Morph: Once the tempo has been calculated, it will be automatically assigned to the selected clip.

8. Morph button: Copies the current tempo to the selected clip. It can be assigned to a MIDI controller.

9. Morph&Fire: Once the tempo has been copied to the selected clip, it will be launched.



©Copyright 2017, Stranger Devices. All rights reserved. All content and plugins are exclusively developed by Esteban Gómez