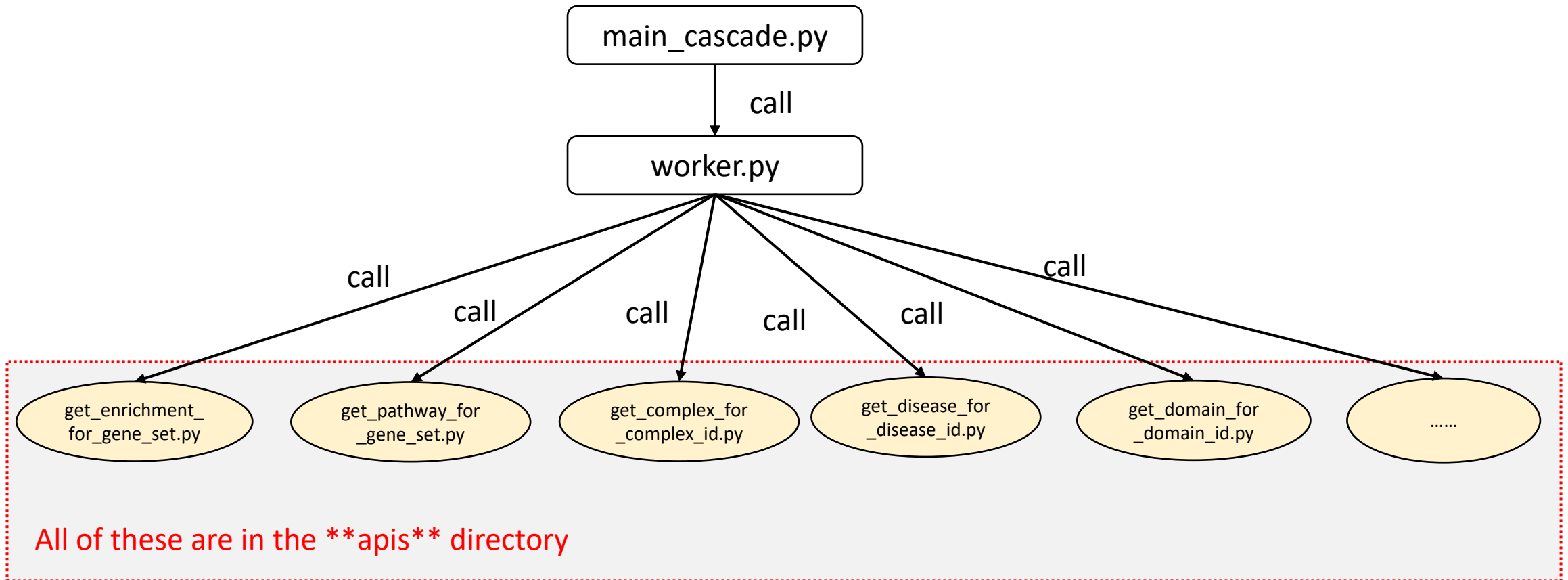
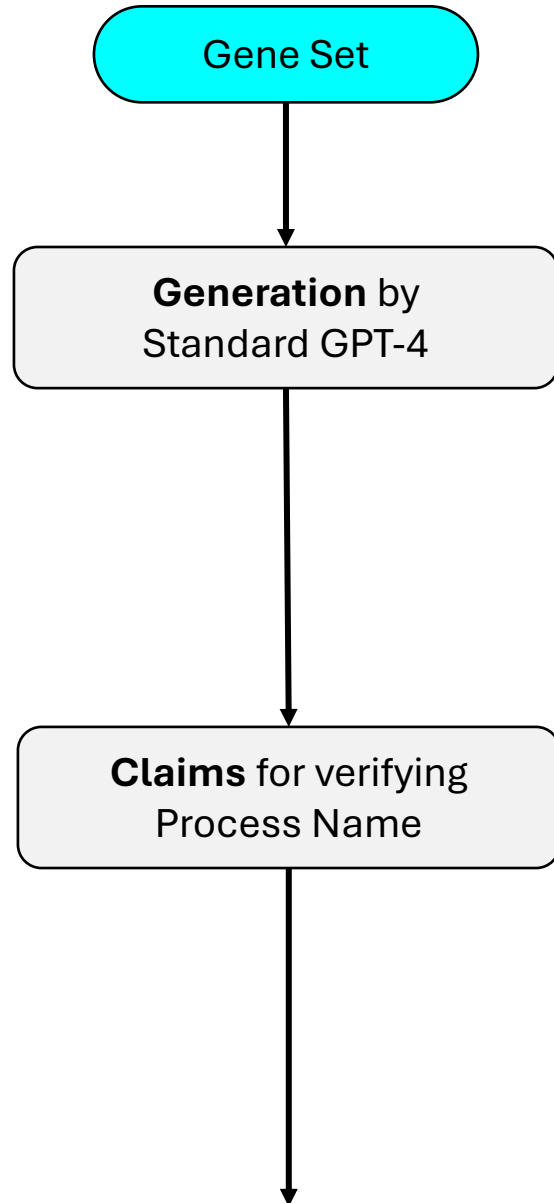


Overall Structure of the GeneAgent Code

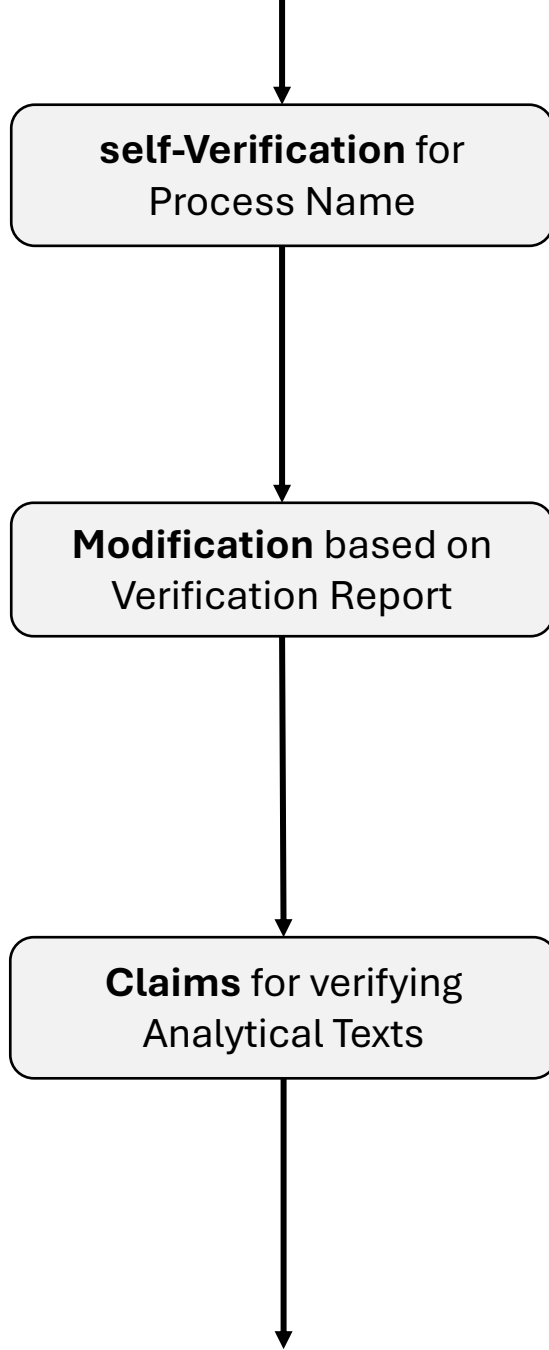


Process of main_cascade.py



```
prompt_baseline = baseline(genes)
messages = [
    {"role": "system", "content": system},
    {"role": "user", "content": prompt_baseline}
]
summary = openai.ChatCompletion.create(
    engine="gpt-4-32k",
    messages=messages,
    temperature=0,
)
messages.append(summary.choices[0]["message"])
summary = summary.choices[0]["message"]["content"]
```

```
process = summary.split("\n")[0].split("Process: ")[1]
prompt_topic = topic(genes, process) + topic_instruction
message_topic = [
    {"role": "system", "content": system_verify},
    {"role": "user", "content": prompt_topic}
]
claims_topic = openai.ChatCompletion.create(
    engine="gpt-4-32k",
    messages=message_topic,
    temperature=0,
)
claims_topic = json.loads(claims_topic.choices[0]["message"]["content"])
```



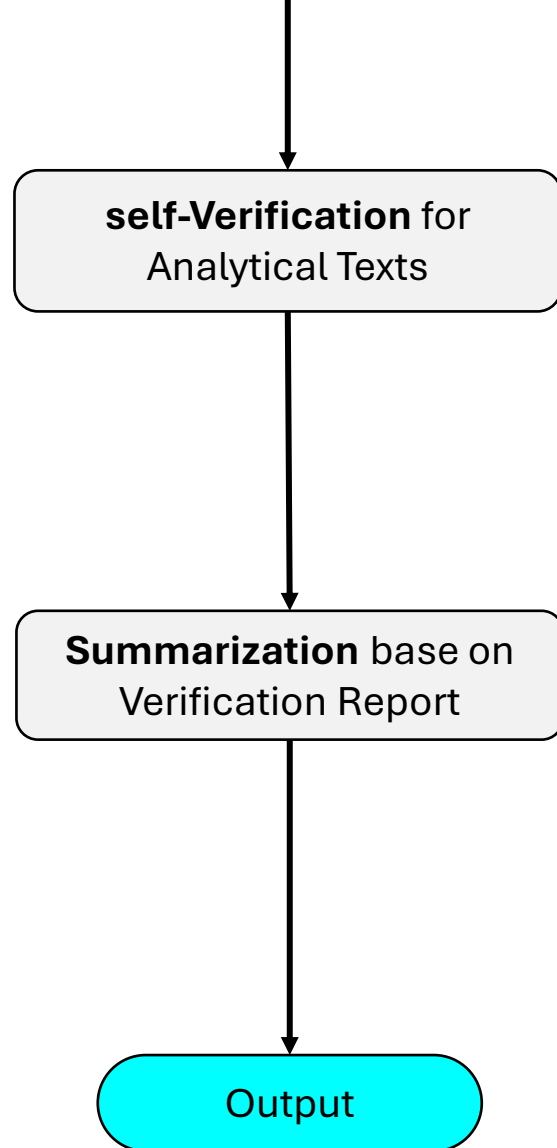
```
verification_topic = ""
for claim in claims_topic:
    claim_result = agentphd.inference(claim)
    verification_topic += f"Original_claim:{claim}"
    verification_topic += f"Verified_claim:{claim_result}"
```

Call **worker.py**

```
messages.append(
    {"role": "user", "content": f"I have finished the verification for process name."
}
)
updated_topic = openai.ChatCompletion.create(
    engine="gpt-4-32k",
    messages=messages,
    temperature=0,
)

messages.append(updated_topic.choices[0]["message"])
updated_topic = updated_topic.choices[0]["message"]["content"]
```

```
prompt_analysis = analysis(genes, summary) + analysis_instruction
analysis_message = [
    {"role": "system", "content": system_verify},
    {"role": "user", "content": prompt_analysis}
]
claims_analysis = openai.ChatCompletion.create(
    engine="gpt-4-32k",
    messages=analysis_message,
    temperature=0,
)
claims_analysis = json.loads(claims_analysis.choices[0]["message"]["content"])
```



```
verification_analysis = ""
for claim in claims_analysis:
    claim_result = agentphd.inference(claim)
    verification_analysis += f"Original_claim:{claim}"
    verification_analysis += f"Verified_claim:{claim_result}"
```

Call **worker.py**

```
messages.append(
    {"role": "assistant", "content": f"I have finished the verification for the revised summary."}
)
updated = openai.ChatCompletion.create(
    engine="gpt-4-32k",
    messages=messages,
    temperature=0,
)
update = updated.choices[0]["message"]["content"]
```